Firepower eStreamer Integration Guide

Version 6.0
May 31, 2016

Cisco Systems, Inc.
www.cisco.com
Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Introduction

The Cisco Event Streamer (also known as eStreamer) allows you to stream Firepower System events to external client applications. You can stream host, discovery, correlation, compliance white list, intrusion, user activity, file, malware, and connection data from a Management Center and you can stream intrusion data from 7000 and 8000 series devices.

Note that eStreamer is not supported on NGIPSv, Firepower Services, Firepower Threat Defense Virtual, and Firepower Threat Defense. To stream events from these devices, you can configure eStreamer on the Management Center that the device reports to.

eStreamer uses a custom application layer protocol to communicate with connected client applications. As the purpose of eStreamer is simply to return data that the client requests, the majority of this guide describes the eStreamer formats for the requested data.

There are three major steps to creating and integrating an eStreamer client with a Firepower System:

1. Write a client application that exchanges messages with the Management Center or managed device using the eStreamer application protocol. The eStreamer SDK includes a reference client application.
2. Configure a Management Center or device to send the required type of events to your client application.
3. Connect your client application to the Management Center or device and begin exchanging data.

This guide provides the information you need to successfully create and run an eStreamer Version 6.0 client application.

Major Changes in eStreamer Version 6.0

If you are upgrading your Firepower System deployment to Version 6.0, please note the following changes, some of which may require you to update your eStreamer client:

- New request message Domain Streaming Request Message Format, page 2-31 allows clients to request events by domain.
- Added the following blocks:
  - Added Name Description Mapping Data Block, page 3-59 to map ID numbers to names and descriptions.
  - Added SSL Rule ID, page 3-106 to provide information about SSL rules.
  - Added User Record, page 3-19 to provide information on user names and detection.

  - Added Endpoint Profile Data Block for 6.0+, page 3-66 to provide information about connection endpoints.
  - Added Access Control Policy Name Data Block, page 3-70 to provide information about access control policy names.
- Added the following metadata records:
  - Realm Metadata for 6.0+, page 3-65
  - Security Group Metadata for 6.0+, page 3-67
  - Sinkhole Metadata for 6.0+, page 3-68
  - Netmap Domain Metadata for 6.0+, page 3-69
  - Filelog Storage Metadata for 6.0+, page 3-100
  - Filelog Sandbox Metadata for 6.0+, page 3-100
  - Filelog Spero Metadata for 6.0+, page 3-101
  - Filelog Archive Metadata for 6.0+, page 3-102
  - Filelog Static Analysis Metadata for 6.0+, page 3-103
  - File Policy Name for 6.0+, page 3-104
- Replaced the following blocks:
  - Replaced Malware Event Data Block 5.4.x, page B-77 with Malware Event Data Block 6.0+, page 3-83 to add an HTTP Response field.
  - Replaced User Information Data Block for 5.x, page B-104 with User Information Data Block for 6.0+, page 4-176 to add endpoint profile, Security Intelligence, and IPv6 fields.
  - Replaced User Login Information Data Block 5.1-5.4.x, page B-102 with User Login Information Data Block 6.0+, page 4-178 to add endpoint profile and Security Intelligence fields.
  - Replaced File Event for 5.4.x, page B-198 with File Event for 6.0+, page 3-73 to add fields for file analysis, local malware analysis, and capacity handling statuses.
  - Replaced Connection Statistics Data Block 5.4.1, page B-165 with Connection Statistics Data Block 6.0+, page 4-109 to add HTTP response, DNS, sinkhole, and Security Intelligence fields.
  - Replaced Access Control Policy Rule Reason Data Block, page B-260 with Access Control Policy Rule Reason Data Block for 6.0+, page 3-69 to increase the Reason field from 16 bits to 32.
  - Replaced Intrusion Event Record 5.4.x, page B-36 with Intrusion Event Record 6.0+, page 3-7 to add an HTTP Response field.

Using this Guide

At the highest level, the eStreamer service is a mechanism for streaming data from the Firepower System to a requesting client. The service can stream the following categories of data:

- Intrusion event data and event extra data
- Correlation (compliance) event data
- Discovery event data
- User event data

- Metadata for events
- Host information
- Malware event data

Descriptions of the data structures returned by eStreamer make up the majority of this book. The chapters in the book are:

- Understanding the eStreamer Application Protocol, page 2-1, which provides an overview of eStreamer communications, details some of the requirements for writing eStreamer client applications, and describes the four types of messages used to send commands to and receive data from the eStreamer service.
- Understanding Intrusion and Correlation Data Structures, page 3-1, which documents the data formats used to return event data generated by the intrusion detection and correlation components and the data formats used to represent the intrusion and correlation events.
- Understanding Discovery & Connection Data Structures, page 4-1, which documents the data formats used to return discovery, user, and connection event data.
- Understanding Host Data Structures, page 5-1, which documents the data formats that eStreamer uses to return full host information data when it receives a host information request message.
- Configuring eStreamer, page 6-1, which documents how to configure the eStreamer on a Management Center or managed device. The chapter also documents the eStreamer command-line switches and provides instructions for manually starting and stopping the eStreamer service and for configuring the Management Center or managed device to start eStreamer automatically.
- Data Structure Examples, page A-1, which provides examples of eStreamer message packets in binary format.
- Understanding Legacy Data Structures, page B-1, which documents the structure of legacy data structures that are no longer in use by the currently shipping product but may be used by older clients.

Prerequisites

To understand the information in this guide, you should be familiar with the features and nomenclature of the Firepower System and the function of its components in general, and with the different types of event data these components generate in particular. Definitions of unfamiliar or product-specific terms can frequently be obtained from the *Firepower eStreamer Integration Guide*.

Product Versions for Firepower System Releases

Version numbers are used throughout this guide to describe the data format for events generated by the Management Center and managed devices. The Firepower System Product Versions table lists versions for each product by major release.

Table 1-1 Firepower System Product Versions

Release	Management Center Version	Master Management Center Version	Intrusion Sensor Version	Sensor Version	Managed Device Version
IMS 3.0	Management Console 3.0	N/A	Network Sensor 3.0	N/A	N/A
IMS 3.1	Management Console 3.1	N/A	Network Sensor 3.1	RNA Sensor 1.0	N/A
IMS 3.2	Management Console 3.2	N/A	Network Sensor 3.2	RNA Sensor 2.0	N/A
3D System 4.0	Management Center 4.0	N/A	Intrusion Sensor 4.0	RNA Sensor 3.0	N/A
3D System 4.5	Management Center 4.5	N/A	Intrusion Sensor 4.5	RNA Sensor 3.5	N/A
3D System 4.6.1	Management Center 4.6.1	Master Management Center 4.6.1	N/A	N/A	4.6.1
3D System 4.7	Management Center 4.7	Master Management Center 4.7	N/A	N/A	4.7
3D System 4.8	Management Center 4.8	Master Management Center 4.8	N/A	N/A	4.8
3D System 4.8.0.2	Management Center 4.8.0.2	Master Management Center 4.8.0.2	N/A	N/A	4.8.0.2
3D System 4.9	Management Center 4.9	Master Management Center 4.9	N/A	N/A	4.9
3D System 4.9.1	Management Center 4.9.1	Master Management Center 4.9.1	N/A	N/A	4.9.1
3D System 4.10	Management Center 4.10	Master Management Center 4.10	N/A	N/A	4.10
3D System 4.10.1	Management Center 4.10.1	Master Management Center 4.10.1	N/A	N/A	4.10.1
3D System 4.10.2	Management Center 4.10.2	Master Management Center 4.10.2	N/A	N/A	4.10.2
3D System 4.10.3	Management Center 4.10.3	Master Management Center 4.10.3	N/A	N/A	4.10.3

Table 1-1 Firepower System Product Versions (continued)

Release	Management Center Version	Master Management Center Version	Intrusion Sensor Version	Sensor Version	Managed Device Version
3D System 5.0	Management Center 5.0	N/A	N/A	N/A	5.0
3D System 5.1	Management Center 5.1	N/A	N/A	N/A	5.1
3D System 5.1.1	Management Center 5.1.1	N/A	N/A	N/A	5.1.1
3D System 5.2	Management Center 5.2	N/A	N/A	N/A	5.2
3D System 5.3	Management Center 5.3	N/A	N/A	N/A	5.3
Firepower System 5.3.1	Management Center 5.3.1	N/A	N/A	N/A	5.3.1
Firepower System 5.4	Management Center 5.4	N/A	N/A	N/A	5.4
Firepower System 6.0	Management Center 6.0	N/A	N/A	N/A	6.0

Document Conventions

The eStreamer Message Data Type Conventions table lists the names used in this book to describe the various data field formats employed in eStreamer messages. Numeric constants used by the eStreamer service are typically unsigned integer values. Bit fields use low-order bits unless otherwise noted. For example, in a one-byte field containing five bits of flag data, the low-order five bits will contain the data.

Table 1-2 eStreamer Message Data Type Conventions

Data Type	Description
nn-bit field	Bit field of nn bits
byte	8-bit byte containing data of arbitrary format
int8	Signed 8-bit byte
uint8	Unsigned 8-bit byte
int16	Signed 16-bit integer
uint16	Unsigned 16-bit integer
int32	Signed 32-bit integer
uint32	Unsigned 32-bit integer
uint64	Unsigned 64-bit integer
string	Variable length field containing character data
[n]	Array subscript following any of the above data types to indicate n instances of the indicated data type, for example, uint8[4]

Table 1-2 eStreamer Message Data Type Conventions (continued)

Data Type	Description
variable	Collection of various data types
BLOB	Binary object of unspecified type, typically raw data as captured from a packet

IP Addresses

The Cisco database stores IPv4 and IPv6 addresses in the same fields in a BINARY format. To get IPv6 addresses, convert to hex notation, for example: `20010db80000000000000000000004321`. The database follows the RFC for storing IPv4 addresses by filling in bits 80-95 with 1's, which yields an invalid IPv6 address. For example, the IPv4 address 10.5.15.1 would be stored as `00000000000000000000FFFF0A050F01`.

Understanding the eStreamer Application Protocol

The Firepower System Event Streamer (eStreamer) uses a message-oriented protocol to stream events and host profile information to your client application. Your client can request event and host profile data from a Management Center, and intrusion event data only from a managed device. Your client application initiates the data stream by submitting request messages, which specify the data to be sent, and then controls the message flow from the Management Center or managed device after streaming begins.

Throughout this document, the eStreamer service on the Management Center or a managed device may be referred to as the eStreamer server or eStreamer.

The following sections describe requirements for connecting to the eStreamer service and introduce commands and data formats used in the eStreamer protocol:

- Connection Specifications, page 2-1 describes the communication flow between the eStreamer service and your client and describes how the client interacts with it.
- Understanding eStreamer Communication Stages, page 2-2 describes the communication protocol for client applications to submit data requests to the eStreamer server and for eStreamer to deliver the requested information to the client.
- Understanding eStreamer Message Types, page 2-6 describes the message types used in the eStreamer protocol; discusses the basic structure of data packets used by eStreamer to return intrusion event data, discovery event data, metadata, and host data to a client; and provides other information to help you write a client that can interpret eStreamer messages.

Connection Specifications

The eStreamer service:

- Communicates using TCP over an SSL connection (the client application must support SSL-based authentication).
- Accepts connection requests on port 8302.
- Waits for the client to initiate all communication sessions.
- Writes all message fields in network byte order (big endian).
- Encodes text in UTF-8.

Understanding eStreamer Communication Stages

There are four major stages of communication that occur between a client and the eStreamer service:

1. The client establishes a connection with the eStreamer server and the connection is authenticated by both parties.

   See Establishing an Authenticated Connection, page 2-2 for more information.

2. The client requests data from the eStreamer service and specifies the types of data to be streamed. A single event request message can specify any combination of available event data, including event metadata. A single host profile request can specify a single host or multiple hosts.

   Two request modes are available for requesting event data:

   - Event Stream Request — The client submits a message containing request flags that specify the requested event types and version of each type, and the eStreamer server responds by streaming the requested data.
   - Extended Request — The client submits a request with the same message format as for Event Stream requests but sets a flag for an extended request. This initiates a message interaction between client and eStreamer server through which the client requests additional information and version combinations not available via Event Stream requests.

   For information on requesting data, see Requesting Data from eStreamer, page 2-3.

3. eStreamer establishes the requested data stream to the client.

   See Accepting Data from eStreamer, page 2-5 for more information.

4. The connection terminates.
   See Terminating Connections, page 2-5 for more information.

Establishing an Authenticated Connection

Before a client can request data from eStreamer, the client must initiate an SSL-enabled TCP connection with the eStreamer service. The client can request on any configured management interface on the Management Center or managed device. Client connections do not enforce traffic channel configuration for management interfaces so that configuration can be ignored when choosing an interface for your connection. When the client initiates the connection, the eStreamer server responds, initiating an SSL handshake with the client. As part of the SSL handshake, the eStreamer server requests the client's authentication certificate, and verifies that the certificate is valid (signed by the Internal Certifying Authority [Internal CA] on the eStreamer server).

Note Cisco recommends that you also require your client to verify that the certificate presented by the eStreamer server has been signed by a trusted Certifying Authority. This is the Internal CA certificate included in the PKCS#12 file that Cisco provides when you register a new eStreamer client with the Management Center or managed device. See Adding Authentication for eStreamer Clients, page 6-3 for more information.

After the SSL session is established, the eStreamer server performs an additional post-connection verification of the certificate. This includes verifying that the client connection originates from the host specified in the certificate and that the subject name of the certificate contains the appropriate value. If either post-connection check fails, the eStreamer server closes the connection. If necessary, you can configure the eStreamer service so that it does not perform a client host name check (see eStreamer Service Options, page 6-4 for more information).

While the client is not required to perform post-connection verification, Cisco recommends that the client perform this verification step. The authentication certificate contains the following field values in the subject name of the certificate:

Table 2-1 Certificate Subject Name Fields

Field	Value
title	eStreamer
generationQualifier	server

After the post-connection verification is finished, the eStreamer server awaits a data request from the client.

Requesting Data from eStreamer

Your client performs the following high-level tasks in managing data requests:

- Initializing the request session — See Establishing a Session, page 2-3.
- Requesting events from the eStreamer event archive — Using Event Stream Requests and Extended Requests to Initiate Event Streaming, page 2-3.
- Requesting host data — See Requesting Host Data, page 2-4.
- Changing a request — See Changing a Request, page 2-5.

Establishing a Session

The client establishes a session by sending an initial Event Stream request to the eStreamer service.

In this initial message, you can either include data request flags or submit the data requests in a follow-on message. This initial Event Stream request message itself is a prerequisite for all eStreamer requests, whether for event data or for host data. For information about using the Event Stream request message, see Event Stream Request Message Format, page 2-10.

Note The eStreamer client can request on any configured management interface on the Management Center or managed device. Client connections do not enforce traffic channel configuration for management interfaces so that configuration can be ignored when choosing an interface for your connection.

Using Event Stream Requests and Extended Requests to Initiate Event Streaming

The eStreamer service provides two modes of requests for event streaming. Your request can combine modes. In both modes, your client starts the request with an Event Stream request message but sets the request flag bits differently. For details about the Event Stream message format, see Event Stream Request Message Format, page 2-10.

When eStreamer receives an Event Stream request message, it processes the client request as follows:

- If the request message does **not** set bit 30 in the request flag field, eStreamer begins streaming any events requested by other set bits in the request flag field. For information, see Submitting Event Stream Requests, page 2-4.

- If bit 30 **is** set in the Event Stream request, eStreamer provides extended request processing. Extended request flags must be sent if this bit is set. For information, see Submitting Extended Requests, page 2-4. Note that eStreamer resolves any duplicate requests. If you request multiple versions of the same data, either by multiple flags or multiple extended requests, the highest version is used. For example, if eStreamer receives flag requests for discovery events version 1 and 6 and an extended request for version 3, it sends version 6.

Submitting Event Stream Requests

Event stream requests use a simple process:

- Your client sends a request message to the eStreamer service with a start date and time and a request flag field that specifies the events and their version level to be included in the data stream.
- eStreamer streams events beginning at the specified time. For information about the streaming protocol, see Accepting Data from eStreamer, page 2-5.

For information on the format and content of the client's Event Stream request message, see Event Stream Request Message Format, page 2-10.

For information on the event types and versions of events that the client can request, see Table 2-6 on page 2-12.

Submitting Extended Requests

If you set bit 30 in the request flags field of an Event Stream Request message, you initiate an extended request, which starts a negotiation with the server. Extended request flags must be sent if this bit is set. For the event types available by extended request, see Table 2-21 on page 2-33.

The steps for extended requests are as follows:

- Your client sends an Event Streaming Request message to eStreamer with the request flags bit 30 set to `1`, which signals an extended request. See Event Stream Request Message Format, page 2-10 for message format details.
- eStreamer answers with a Streaming Information message that advertises the list of services available to the client. For details about the Streaming Information message, see Streaming Information Message Format, page 2-28.
- The client returns a Streaming Request message that indicates the service it wants to use, with a request list of event types and versions available from that service. The request list corresponds to setting bits in the request flag field when making a standard event stream request. For details about how to use the Streaming Request message to request events, see "Sample Extended Request Messages" section on page 2-35.
- eStreamer processes the client's Streaming Request message and begins streaming the data at the time specified in the message. For information about the streaming protocol, see Accepting Data from eStreamer, page 2-5.

Requesting Host Data

Once you have established a session, you can submit a request for host data at any time. eStreamer generates information for the requested hosts from the Firepower System network map.

Changing a Request

To change request parameters for an established session, the client must disconnect and request a new session.

Accepting Data from eStreamer

Note The eStreamer server does not keep a history of the events it sends. Your client application must check for duplicate events, which can inadvertently occur for a number of reasons. For example, when starting up a new streaming session, the time specified by the client as the starting point for the new session can have multiple messages, some of which may have been sent in the previous session and some of which were not. eStreamer sends all message that meet the specified request criteria. Your application should detect any resulting duplicates.

During periods of inactivity, eStreamer sends periodic null messages to the client to keep the connection open. If it receives an error message from the client or an intermediate host, it closes the connection.

eStreamer transmits requested data to the client differently, depending on the request mode.

Event Stream Requests

If the client submits an event stream request, eStreamer returns data message by message. It may send multiple messages in a row without waiting for a client acknowledgment. At a certain point, it pauses and waits for the client. The client operating system buffers received data and lets the client process it at its own pace.

If the client request includes a request for metadata, eStreamer sends the metadata first. The client should store it in memory to be available when processing the event records that follow.

Extended Requests

If the client submits an extended request, eStreamer queues up messages and sends them in bundles. eStreamer may send multiple bundles in a row without waiting for a client acknowledgment. At a certain point, it pauses and waits for the client. The client operating system buffers received data and lets the client read it off at its own pace.

The client unpacks each bundle, message by message, and uses the lengths of the records and the blocks to parse each message. The overall message length in each message header can be used to calculate when the end of each message has been reached, and the overall bundle length can be used to know when the end of the bundle is reached. The bundle requires no index of its contents to be correctly parsed.

For information about the message bundling mechanism, see Message Bundle Format, page 2-36.

For information about the null message that the client can use for additional flow control, see Null Message Format, page 2-7.

Terminating Connections

The eStreamer server attempts to send an error message before closing the connection. For information on error messages, see Error Message Format, page 2-8.

The eStreamer server can close a client connection for the following reasons:

- Any time sending a message results in an error. This includes both event data messages and the null keep-alive message eStreamer sends during periods of inactivity.
- An error occurs while processing a client request.
- Client authentication fails (no error message is sent).
- eStreamer service is shutting down (no error message is sent).

Your client can close the connection to eStreamer server at any time and should attempt to use the error message format to notify the eStreamer server of the reason.

Understanding eStreamer Message Types

The eStreamer application protocol uses a simple message format that includes a standard message header and various sub-header fields followed by the record data which contains the message's payload. The message header is the same in all eStreamer message types; for more information, see eStreamer Message Header, page 2-7.

Table 2-2 eStreamer Message Types

Message Type	Name	Description
0	Null message	Both the eStreamer server and the client send null messages to control data flow. For information, see Null Message Format, page 2-7.
1	Error message	Both the eStreamer server and the client use error messages to indicate why a connection closed. For information, see Error Message Format, page 2-8.
2	Event Stream Request	A client sends this message type to the eStreamer service to initiate a new streaming session and request data. For information, see Event Stream Request Message Format, page 2-10.
4	Event Data	The eStreamer service uses this message type to send event data and metadata to the client. For information, see Event Data Message Format, page 2-17.
5	Host Data Request	A client sends this message type to the eStreamer service to request host data. A session must be started already via an Event Stream Request message. For information, see Host Request Message Format, page 2-25.
6	Single Host Data	The eStreamer service uses this message type to send single host data requested by the client. For information, see Host Data and Multiple Host Data Message Format, page 2-28.
7	Multiple Host Data	The eStreamer service uses this message type to send multiple host data requested by the client. For information, see Host Data and Multiple Host Data Message Format, page 2-28.

Table 2-2 eStreamer Message Types (continued)

Message Type	Name	Description
2049	Streaming Request	A client uses this message type in extended requests to specify which of the advertised events from the Stream Information message it wants. For information, see Sample Extended Request Messages, page 2-35.
2051	Streaming Information	The eStreamer service uses this message type in extended requests to advertise the list of services available to the client. For information, see Streaming Information Message Format, page 2-28.
4002	Message Bundle	The eStreamer service uses this message type to package messages that it streams to clients. For information, see Message Bundle Format, page 2-36.

eStreamer Message Header

All eStreamer messages start with the message header illustrated in the graphic below. The following table explains the fields.

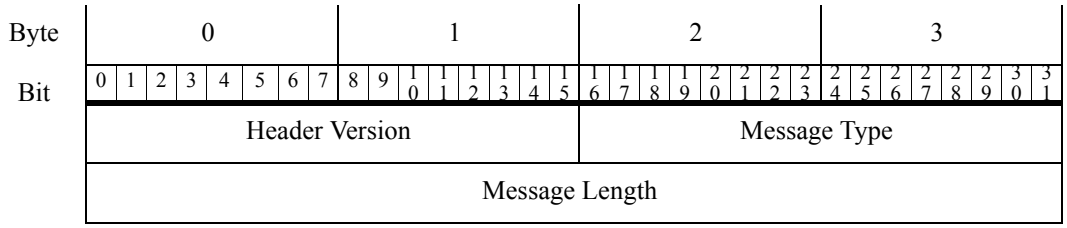


Table 2-3 Standard eStreamer Message Header Fields

Field	Data Type	Description
Header Version	uint16	Indicates the version of the header used on the message. For the current version of eStreamer, this value is always `1`.
Message Type	uint16	Indicates the type of message transmitted. For the list of current values, see Table 2-2 on page 2-6.
Message Length	uint32	Indicates the length of the content that follows, and excludes the bytes in the message header itself. A message with a header and no data has a message length of zero.

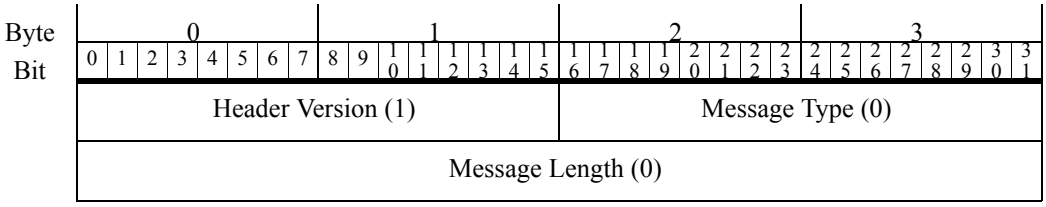
Null Message Format

Both the client application and the eStreamer service send null messages. The null message has a type of 0 and contains no data after the message header.

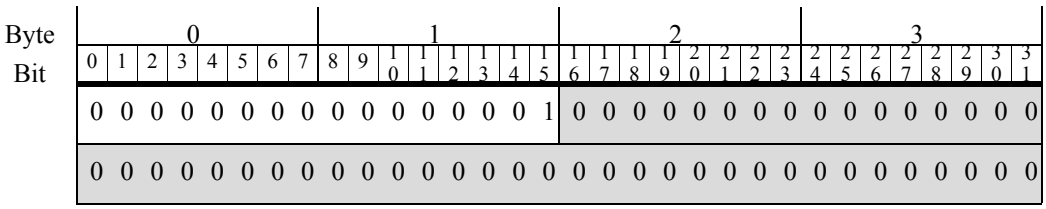
The client sends a null message to the eStreamer server to indicate readiness to accept more data. The eStreamer service sends null messages to the client to keep the connection alive when no data is being transmitted. The message length value for null messages is always set to `0`.

Tip In data structure diagrams in this book, integers in parentheses such as (1) or (115) represent constant field values. For example, Header Version (1) means that the field in the data structure under discussion always has a value of 1.

The Null message format is shown below. The only non-zero value in the message is the header version.



An example of a null message in binary format follows. Notice that the only non-zero value is in the second byte, signifying a header version value of 1. The message type and length fields (shaded) each have a value of 0.



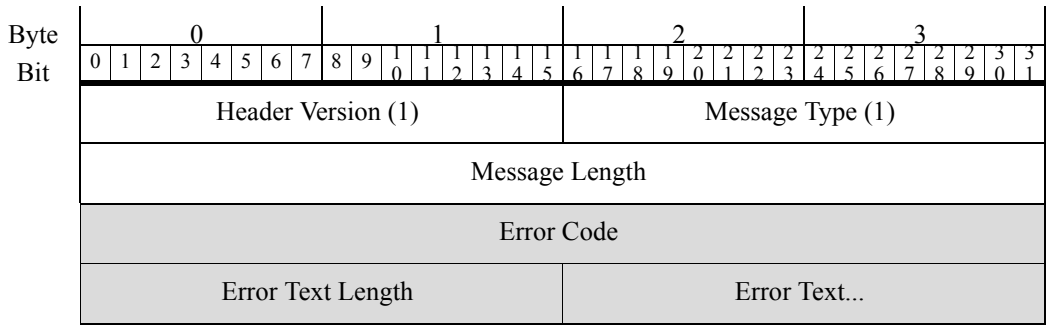
Tip Examples in this guide appear in binary format to clearly display which bits are set. This is important for some messages, such as the event request message and event impact fields.

Error Message Format

Both the client application and the eStreamer service use error messages. Error messages have a message type of 1 and contain a header, an error code, an error text length, and the actual error text. Error text can contain between 0 and 65,535 bytes.

When you create custom error messages for your client application, Cisco recommends using -1 as the error code.

The following graphic illustrates the basic error message format. Shaded fields are specific to error messages.

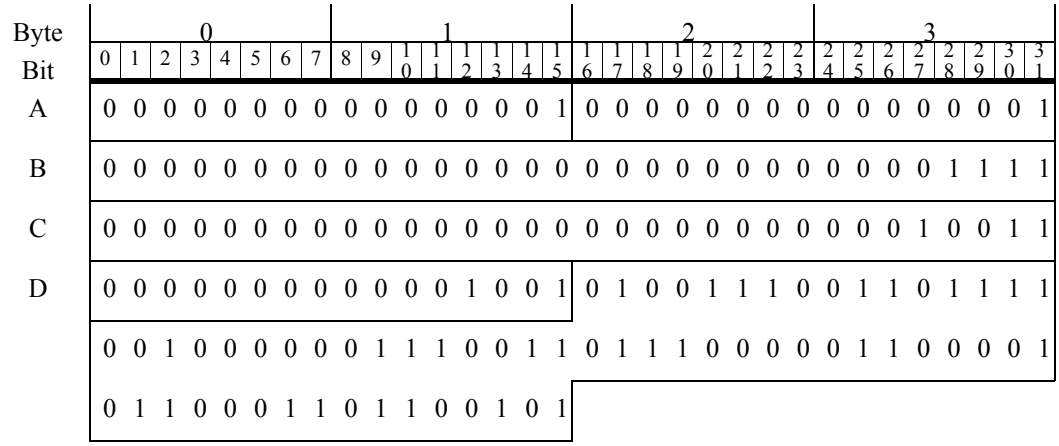


The following table describes each field in error code messages.

Table 2-4 Error Message Fields

Field	Data Type	Description
Error Code	int32	A number representing the error.
Error Text Length	uint16	The number of bytes included in the error text field.
Error Text	variable	The error message. Up to 65,535 bytes.

The following diagram shows an example error message:



In the preceding example, the following information appears:

Letter	Description
A	The first two bytes indicate the standard header value of `1`. The second two bytes show a value of 1, which signifies that the transmission is an error message.
B	This line indicates the amount of message data that follows it. In this example, 15 bytes (in binary, `1111`) of data follow.

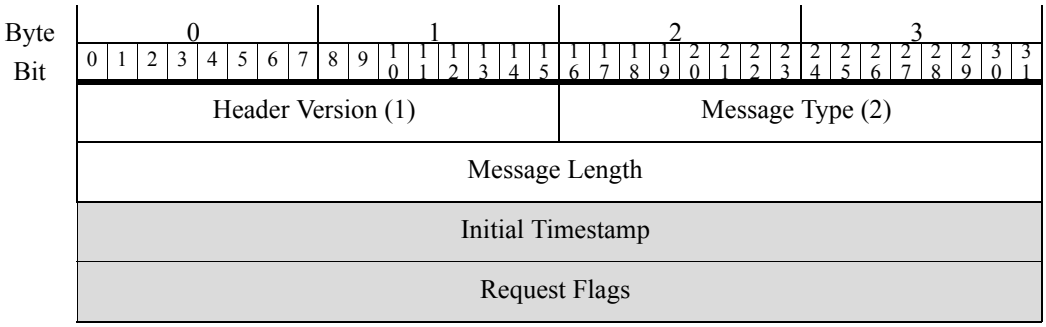
Letter	Description
C	This line displays the error code. In this example, the message contains a value of `19` (`10011`). Therefore, error number 19 is transmitted in the message.
D	This line contains the number of bytes in the error message (`1001`, or nine bytes), and the error message itself follows in the next nine bytes. The error message value, when converted to ASCII text, equals “No space,” which is the error message that accompanies error code 19.

Event Stream Request Message Format

eStreamer clients use the Event Stream Request message to start a streaming session. The request message includes a start time and a bit flag field to specify the data the eStreamer service should include, which can be any combination of events, as well as intrusion event extra data and metadata. The Event Stream Request message can initiate both event stream requests and extended requests. The message type is 2.

You must submit an Event Stream Request message for all data requests, including a request exclusively for host profile information. In such a case, you first submit an Event Stream Request message, then a Host Request message (type 5) to specify the host data.

The following graphic illustrates the Event Stream Request message format. The message uses the standard header. The shaded fields are specific to the request message and are described in the following table.



The following table describes each field in Event Stream Request messages.

Table 2-5 Event Stream Request Message Fields

Field	Data Type	Description
Initial Timestamp	uint32	Defines the start of the session. To start at: • the time the client connects to eStreamer, set all timestamp bits to 1. • the oldest data available, set all timestamp bits to zero. • a given date and time, specify the UNIX timestamp (number of seconds since January 1, 1970). See Initial Timestamp, page 2-11 below for important information.
Request Flags	bits[32]	Specifies the types and versions of events and metadata to be returned in event stream requests. See Request Flags, page 2-11 for flag definitions. Setting bit 30 initiates an extended request, which can co-exist with event stream requests in the same message.

Initial Timestamp

Note Your client application should use the archival timestamp in the Initial Timestamp field when submitting an event stream request, as explained below. This ensures that you do not inadvertently exclude events. Devices transmit data to the Management Center using a "store and forward" mechanism with transmission delays. If you request events by the generation timestamp assigned by the device that detects it, delayed events may be missed.

When starting a session, a best practice is to start up from the archival timestamp (also known as the "server timestamp") of the last record in the previous session. It is not a technical requirement but is strongly recommended. Under certain circumstances, if you use the generation timestamp you can inadvertently exclude events from the new streaming session.

To include the archival timestamp in your streamed events, you must set bit 23 in the request flag field.

Note that only time-based events have archival timestamps. Events that eStreamer generates, such as metadata, have zero in this field when extended event headers have been requested with bit 23 set.

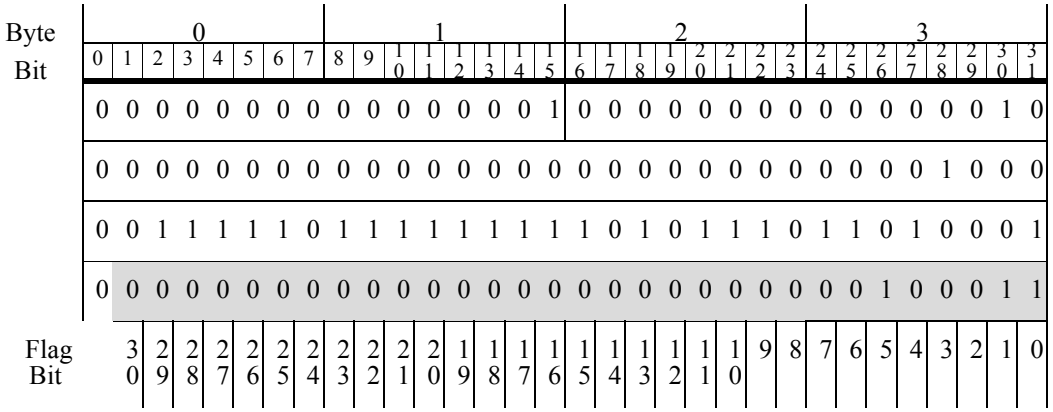
Request Flags

You set bits 0 through 29 in the event data request flag field to select the types of events you want eStreamer to send. You set bit 30 to activate the extended request mode. Setting bit 30 does not directly request any data. Extended request flags must be sent if this bit is set. Your client requests data during the server-client message dialog that follows submission of the Event Stream Request message. For information on extended requests, see Requesting Data from eStreamer, page 2-3.

See Table 2-6 on page 2-12 for definitions of the bit settings in the Request Flags field. Different flags request different versions of the event data. For example, to obtain data in Firepower System 4.9 format instead of 4.10 format you set a different flag bit. For specific information on the flags to use when requesting data for particular product versions, see Table 2-7 on page 2-15.

Note that you request metadata by version, not by the individual metadata record. For information about each supported version of metadata, see Request Flags, page 2-11.

The following diagram shades the bits in the flags field that are currently used:



For information on each request flag bit, see the following table.

Table 2-6 Request Flags

Bit Field	Description
Bit 0	Requests the transmission of packet data associated with intrusion events. If set to `1`, packet data is transmitted with intrusion events. If set to `0`, packet data is not transmitted.
Bit 1	Requests the transmission of version 1 metadata associated with intrusion, discovery, correlation, and connection events. If set to `1`, version 1 metadata is transmitted with events. If set to `0`, version 1 metadata is not transmitted. You can use metadata to resolve coded and numeric fields in events. See Understanding Metadata, page 2-37 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 2	Requests the transmission of intrusion events. If bit 2, bit 6, or both bit 2 and 6 are set to `1`, but the extended request flag, bit 30, is set to `0`, the system interprets this as a request from a Version 4.x client and record type 104/105 is sent. If no event type is specified when bit 2, bit 6, or both bit 2 and 6 are set to `1`, and bit 30 is set to `1`, the system interprets this as a request from a Version 5.0-5.1 client and record type 207/208 is sent. If bit 30 is set to `1`, and a specific event type is requested, intrusion events are sent regardless of bits 2 and 6. For details on requesting record types, see Submitting Extended Requests, page 2-4. If bit 2, bit 6, and bit 30 are all set to `0`, intrusion events are not sent. Bit 6 is used in a manner identical to bit 2. Either bit can be set to request intrusion events. Setting one of these bits to `0` will not override the other bit; setting bit 2 to `0` and bit 6 to `1`, or setting bit 2 to `1` and bit 6 to `0`, will be interpreted as a request for intrusion events.
Bit 3	Requests the transmission of discovery data version 1 (Management Center 3.2). If set to `0`, discovery data version 1 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures, page 4-1.
Bit 4	Requests the transmission of correlation data version 1 (Management Center 3.2). If set to `0`, correlation data version 1 is not transmitted.

Table 2-6 Request Flags (continued)

Bit Field	Description
Bit 5	Requests the transmission of impact correlation events (intrusion impact alerts). If set to 1, intrusion impact alerts are transmitted. If set to 0, intrusion impact alerts are not transmitted. See Intrusion Impact Alert Data 5.3+, page 3-16 for more information about intrusion impact alerts.
Bit 6	Bit 6 is used in a manner identical to bit 2. See Bit 2, page 2-12.
Bit 7	Requests the transmission of discovery data version 2 (Management Center 4.0 - 4.1) if set to 1. If set to 0, discovery data version 2 is not transmitted.
Bit 8	Requests the transmission of connection data version 1 (Management Center 4.0 - 4.1) if set to 1. If set to 0, connection data version 1 is not sent.
Bit 9	Requests the transmission of correlation data version 2 (Management Center 4.0 - 4.1.x) if set to 1. If set to 0, correlation policy data version 2 is not transmitted.
Bit 10	Requests the transmission of discovery data version 3 (Management Center 4.5 - 4.6.1) if set to 1. If set to 0, discovery data version 3 is not transmitted. For more information about legacy discovery events, see Legacy Discovery Data Structures, page B-87.
Bit 11	Disables transmission of events.
Bit 12	Requests the transmission of connection data version 3 (Management Center 4.5 - 4.6.1) if set to 1. If set to 0, connection data version 3 is not sent.
Bit 13	Requests the transmission of correlation data version 3 (Management Center 4.5 - 4.6.1). If set to 0, correlation data version 3 is not transmitted.
Bit 14	Requests the transmission of version 2 metadata associated with intrusion, discovery, correlation, and connection events. If set to 1, version 2 metadata is transmitted with events. If set to 0, version 2 metadata is not transmitted. See Understanding Metadata, page 2-37 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 15	Requests the transmission of version 3 metadata associated with intrusion, correlation, discovery, and connection events. If set to 1, version 3 metadata is transmitted with events. If set to 0, version 3 metadata is not transmitted. See Understanding Metadata, page 2-37 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 16	Unused
Bit 17	Requests the transmission of discovery data version 4 (Management Center 4.7 - 4.8.x). If set to 0, discovery data version 4 is not transmitted.
Bit 18	Requests the transmission of connection data version 4 (Management Center 4.7 - 4.9.0.x) if set to 1. If set to 0, connection data version 4 is not sent. See Connection Chunk Message, page 4-47 for more information.
Bit 19	Requests the transmission of correlation data version 4 (Management Center 4.7). If set to 0, correlation data version 4 is not transmitted. See Legacy Correlation Event Data Structures, page B-209 for information about correlation events transmitted in Management Center 4.7 format.

Table 2-6 Request Flags (continued)

Bit Field	Description
Bit 20	Requests the transmission of version 4 metadata associated with intrusion, discovery, user activity, correlation, and connection events. If set to `1`, version 4 metadata is transmitted with events. If set to `0`, version 4 metadata is not transmitted. Version 4 metadata includes the following: • correlation (compliance) rule information • correlation (compliance) policy information • fingerprint records • client application records • client application type records • vulnerability records • host criticality records • network protocol records • host attribute records • scan type records • user records • service detection device (version 2) records • event classification (version 2) records • priority records • rule information (version 2) • malware information If you request bit 20 with bit 22, user metadata is also sent. See Understanding Metadata, page 2-37 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 21	Requests the transmission of version 1 user events. For more information on user events, see User Record, page 4-18.
Bit 22	Requests the transmission of correlation data version 5 (Management Center 4.8.0.2 - 4.9.1). If set to `0`, correlation data version 5 is not transmitted. If you request bit 20 with bit 22, user metadata is also sent. For more information about legacy correlation (compliance) events, see Legacy Correlation Event Data Structures, page B-209.
Bit 23	Requests extended event headers. If set to `1`, events are transmitted with the timestamp applied when the event was archived for the eStreamer server to process and four bytes reserved for future use. If this field is set to `0`, events are sent with a standard event header that only includes the record type and record length. See eStreamer Message Header, page 2-7 for information about the event message header.
Bit 24	Requests the transmission of discovery data version 5 (Management Center 4.9.0.x). If set to `0`, discovery data version 5 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures, page 4-1.

Table 2-6 Request Flags (continued)

Bit Field	Description
Bit 25	Requests the transmission of discovery data version 6 (Management Center 4.9.1+). If set to `0`, discovery data version 6 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures, page 4-1.
Bit 26	Requests the transmission of connection data version 5 (Management Center 4.9.1 - 4.10.x) if set to `1`. If set to `0`, connection data version 5 is not sent. See Connection Chunk Message, page 4-47 for more information.
Bit 27	Requests event extra data associated with an intrusion event in an Extra Data record. For more information about event data, see Table 3-11Intrusion Event Extra Data Data Block Fields, page 3-27.
Bit 28	Requests the transmission of discovery data version 7 (Management Center 4.10.0+). If set to `0`, discovery data version 7 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures, page 4-1.
Bit 29	Requests the transmission of correlation data version 6 (Management Center 4.10 - 4.10.x). If set to `0`, correlation policy data version 6 is not transmitted. If you request bit 20 with bit 29, user metadata is also sent. For more information about correlation events, see earlier versions of the product.
Bit 30	Indicates an extended request to eStreamer. Extended request flags must be sent if this bit is set. For information about extended requests, see Submitting Extended Requests, page 2-4.

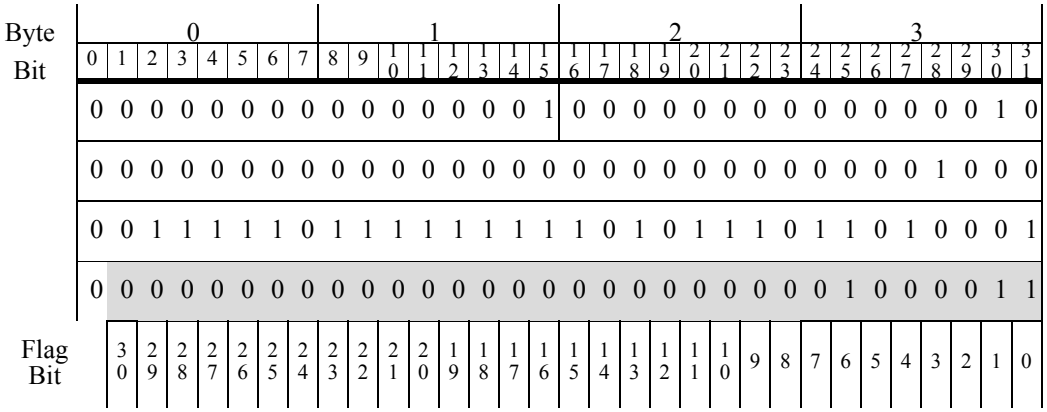
To help you decide which flags to use to request data for a particular version, see the following table. For Version 5.0 and later, see Submitting Extended Requests, page 2-4 for more information about using Bit 30.

Table 2-7 Event Request Flags by Product Version

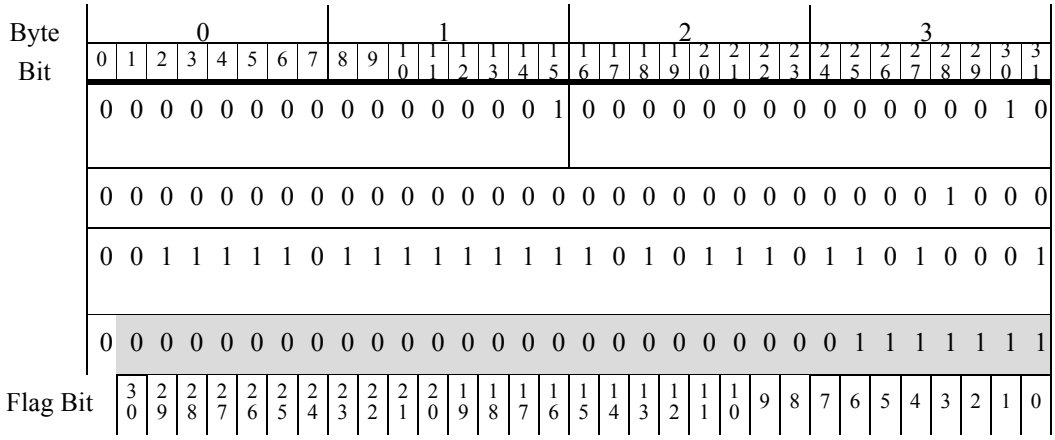
Type of Requested Data	4.9.0.x	4.9.1.x	4.10.x	5.0+	5.1	5.1.1+
packet data	Bit 0	Bit 0	Bit 0	Bit 0	Bit 0	Bit 0
intrusion events	Bit 2	Bit 2	Bit 2	Bit 2	Bit 2	Bit 30
metadata	Bit 20	Bit 20	Bit 20	Bit 20	Bit 20	Bit 20
discovery events	Bit 24	Bit 25	Bit 28	Bit 30	Bit 30	Bit 30
correlation events	Bit 22	Bit 22	Bit 29	Bit 30	Bit 30	Bit 30
event extra data	—	—	Bit 27	Bit 27	Bit 27	Bit 27
impact event alerts	Bit 5	Bit 5	Bit 5	Bit 5	Bit 5	Bit 5
connection data	Bit 18	Bit 26	Bit 26	Bit 30	Bit 30	Bit 30
user events	Bit 21	Bit 21	Bit 21	Bit 30	Bit 30	Bit 30
malware events	—	—	—	—	—	Bit 30
file events	—	—	—	—	—	Bit 30

Caution In all event types, prior to version 5.x, the reference client labels `detection engine ID` fields as `sensor ID`.

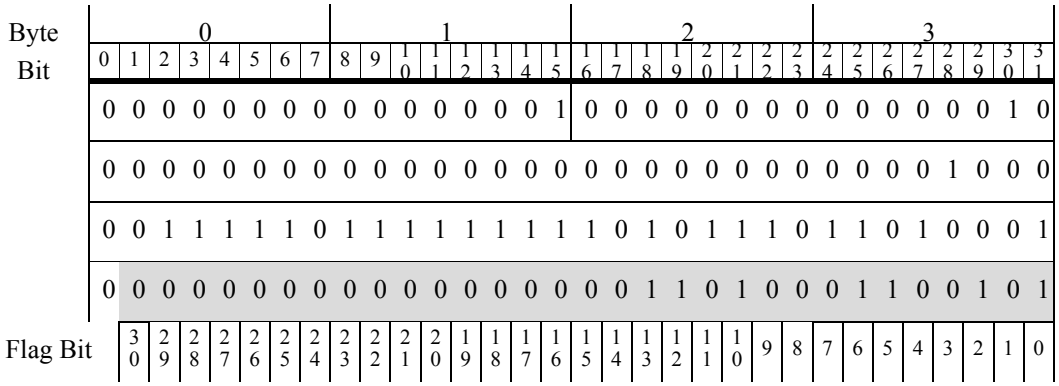
The following example requests intrusion events of type 7 (compatible with Firepower System 3.2+) with both version 1 metadata and packet flags:



To request only data compatible with Firepower System 3.2 (including intrusion events, packets, metadata, impact alerts, policy violation events, and version 2.0 events), use the following:



To request intrusion impact alerts, correlation events, discovery events, connection events, and intrusion events of type 7 with packets and version 3 metadata in Management Center 4.6.1+ format, use the following:



Event Data Message Format

The eStreamer service transmits event data and related metadata to clients when it receives an event request. Event data messages have a message type of 3. Each message contains a single data record with either event data or metadata.

Note that type 3 messages carry only event data and metadata. eStreamer transmits host information in type 6 (single-host) and type 7 (multiple-host) messages. See Host Data and Multiple Host Data Message Format, page 2-28 for information on host message formats.

Understanding the Organization of Event Data Messages

The event data and metadata messages that eStreamer sends contain the following sections:

- eStreamer message header — The standard message header defined at eStreamer Message Header, page 2-7.
- Event-specific sub-headers — Sets of fields that vary by event type, with codes that describe additional event details and determine the structure of the payload data that follows.
- Data record — Fixed-length fields and a data block.

Note The client should unpack all messages on the basis of field length.

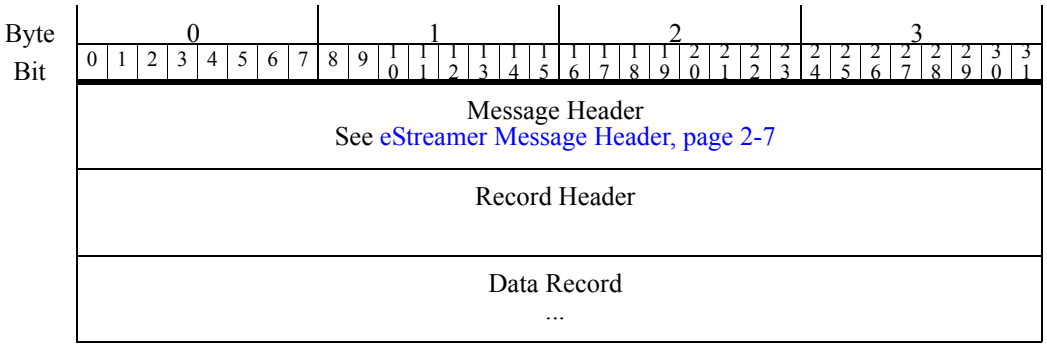
For the event message formats by event type, see the following:

- Intrusion Event and Metadata Message Format, page 2-18 for intrusion event data records and all metadata records. These messages have fixed-length fields.
- Discovery Event Message Format, page 2-19 for messages with discovery event or user event data. In addition to the standard eStreamer message header and a record header similar to the intrusion event message, discovery messages have a distinctive discovery event header with an event type and subtype field. The data record in discovery event messages is packaged in a series 1 block that can have variable length fields and multiple layers of encapsulated blocks.
- Connection Event Message Format, page 2-21 for messages with connection statistics. Their general structure is identical to discovery event messages. Their data block types, however, are specific for connection statistics.

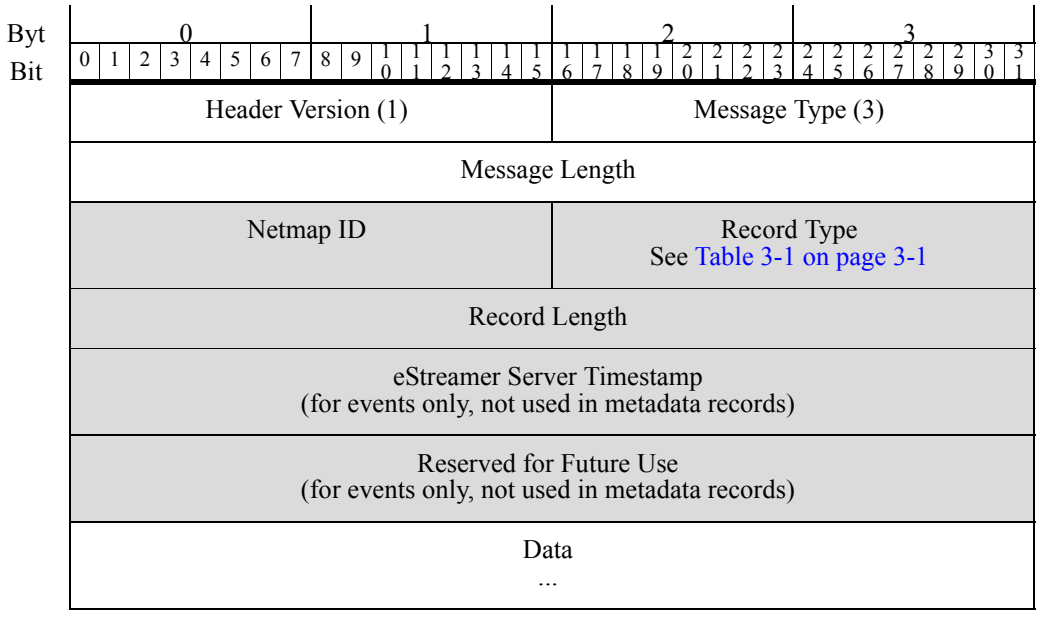
- Correlation Event Message Format, page 2-21 for messages with correlation (compliance) event data. The headers in these messages are the same as in intrusion event messages but the data blocks are series 1 blocks.
- Event Extra Data Message Format, page 2-23 for a series of messages that deliver intrusion-related record types with variable-length fields and multiple layers of nested data blocks such as intrusion event extra data. See Event Extra Data Message Format, page 2-23 for general information on the structure of this message series. See Data Block Header, page 2-24 for information about the structures of this series of blocks which are similar to series 1 blocks but numbered separately.

Intrusion Event and Metadata Message Format

The graphic below shows the general structure of intrusion event and metadata messages.



The following graphic shows the details of the record header portion of the intrusion event and metadata message format. The record header fields are shaded. The table that follows defines the fields.



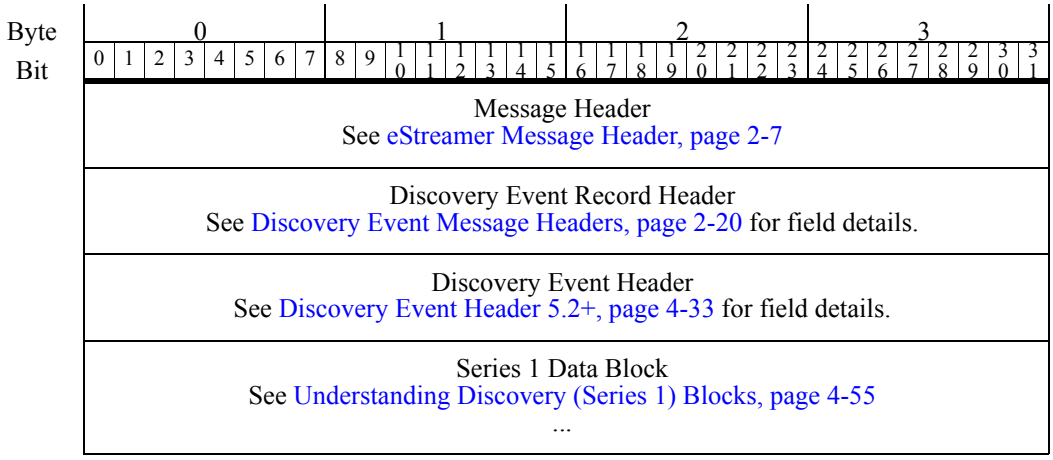
The following table describes each field in the header of intrusion events and metadata messages.

Table 2-8 Intrusion Event and Metadata Record Header Fields

Field	Data Type	Description
Netmap ID	uint16	The first bit of this field is a flag indicating whether the header is an extended header containing an archive timestamp. The remaining 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. If this field is not used, it is left empty. Netmap IDs map to domains as provided in metadata.
Record Type	uint16	Identifies the data record content type. See Table 3-1Intrusion Event and General Metadata Record Types, page 3-1 for the list of record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.

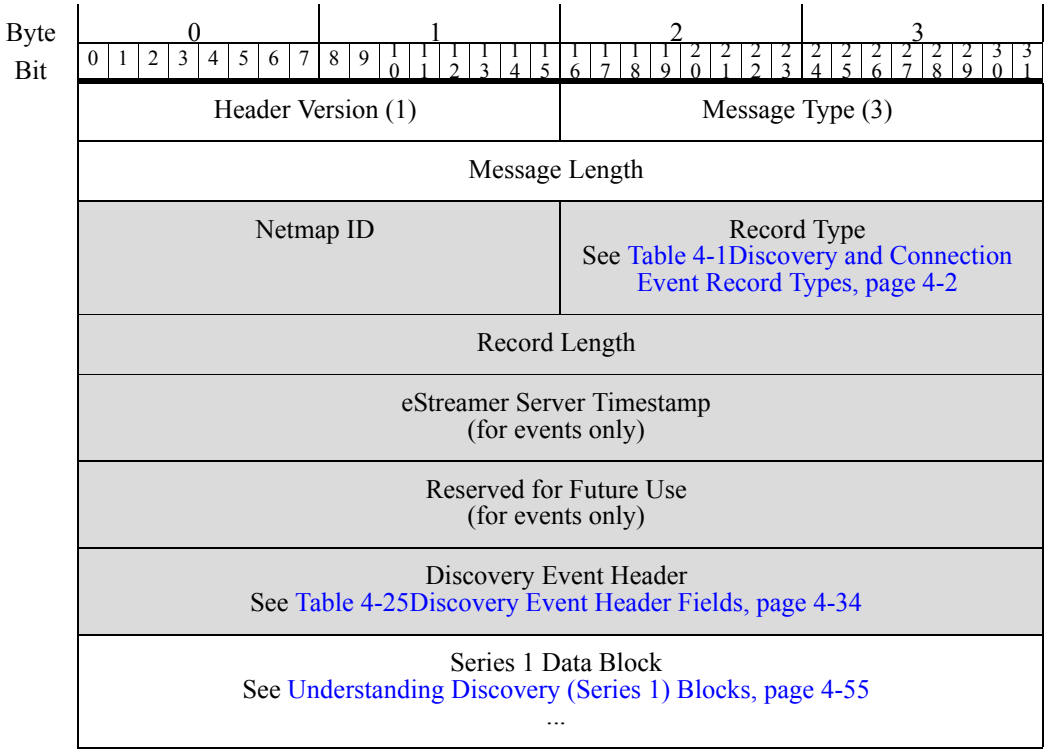
Discovery Event Message Format

The graphic below shows the structure of discovery event messages. The standard eStreamer message header and event record header are followed by a discovery event header used only in discovery and user event messages. The discovery event header section of the message contains the discovery event type and subtype fields, which together form a key to the data block that follows. For the current discovery event types and subtypes, see Table 4-26Discovery and Connection Events by Type and Subtype, page 4-35.



Discovery Event Message Headers

The shaded section in the following graphic shows the fields of the record header in the discovery event data message format, and shows the location of the event header that follows it. The following table defines the fields of the discovery event message headers.



The following table describes the fields in the record header and the event header of the discovery event message.

Table 2-9 Discovery Event Message Header Fields

Field	Data Type	Description
Netmap ID	uint16	The first bit of this field is a flag indicating whether the header is an extended header containing an archive timestamp. The remaining 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. If this field is not used, it is left empty. Netmap IDs map to domains as provided in metadata.
Record Type	uint16	Identifies the data record content type. See Table 4-1Discovery and Connection Event Record Types, page 4-2 for the list of record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)

Table 2-9 Discovery Event Message Header Fields (continued)

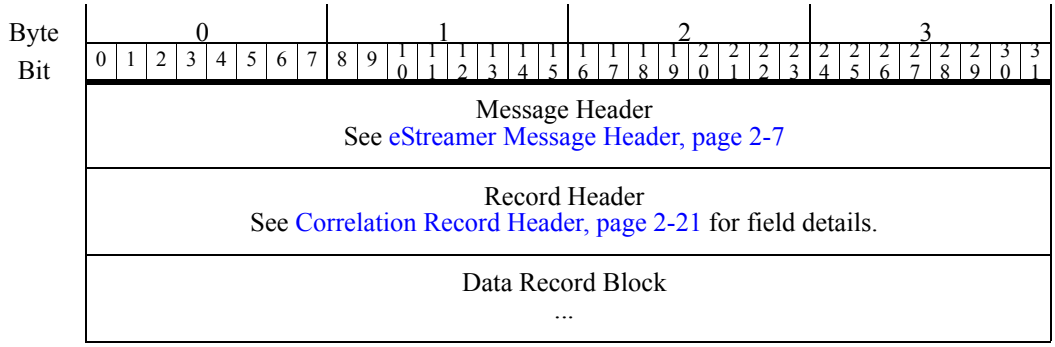
Field	Data Type	Description
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request flags field of the event stream request.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.
Discovery Event Header	Varied	Contains a number of fields, including the event type and subtype, which together form a unique key to the data structure that follows. See Discovery Event Header 5.2+, page 4-33 for definitions of fields in the discovery event header.

Connection Event Message Format

Messages with connection statistics have a structure identical to discovery event messages. See Discovery Event Message Format, page 2-19 for general message format information. Connection event messages are distinct in terms of the data block types they incorporate.

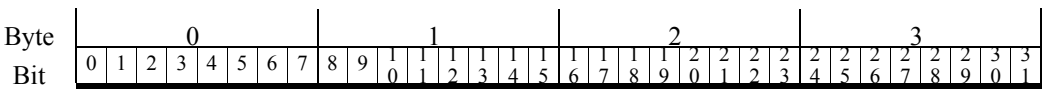
Correlation Event Message Format

The graphic below shows the general structure of correlation (compliance) event messages. The standard eStreamer message header and record header are followed immediately by a data block in the data record section of the message. Correlation messages use Series 1 data blocks.



Correlation Record Header

The shaded section of the following graphic shows the fields of the record header in correlation event messages. Note that correlation messages use series 1 data blocks; however, they do not have the discovery header that appears in discovery event messages. Their header fields resemble those of intrusion event messages. The table that follows the graphic below defines the record header fields for correlation events.



Header Version (1)	Message Type (3)
Message Length	
Netmap ID	Record Type See Table 3-1Intrusion Event and General Metadata Record Types, page 3-1
Record Length	
eStreamer Server Timestamp (for events only, not used in metadata records)	
Reserved for Future Use (for events only, not used in metadata records)	
Data Record Block Uses Series 1 block, see Understanding Discovery (Series 1) Blocks, page 4-55 ...	

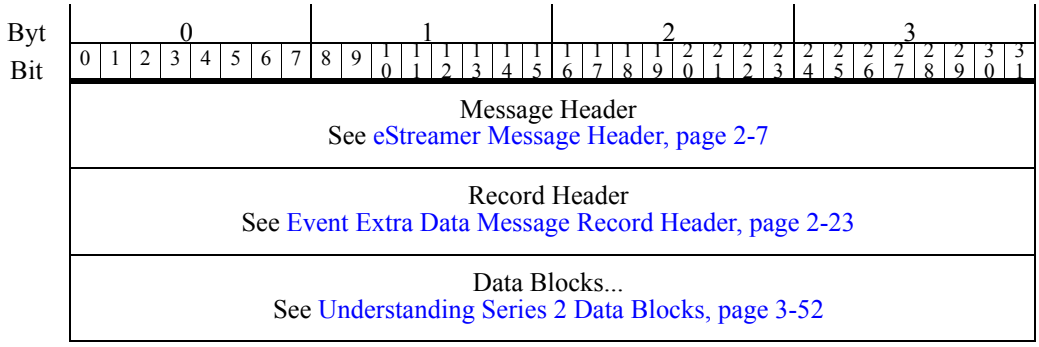
The following table describes each field in the record header of correlation event messages.

Table 2-10 Correlation Event Message Record Header Fields

Field	Data Type	Description
Netmap ID	uint16	The first bit of this field is a flag indicating whether the header is an extended header containing an archive timestamp. The remaining 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. If this field is not used, it is left empty. Netmap IDs map to domains as provided in metadata.
Record Type	uint16	Identifies the data record content type. See Table 3-1 on page 3-1 for the list of intrusion, correlation, and metadata record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags. Field is zero for data generated by the Management Center such as host profiles and metadata.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.

Event Extra Data Message Format

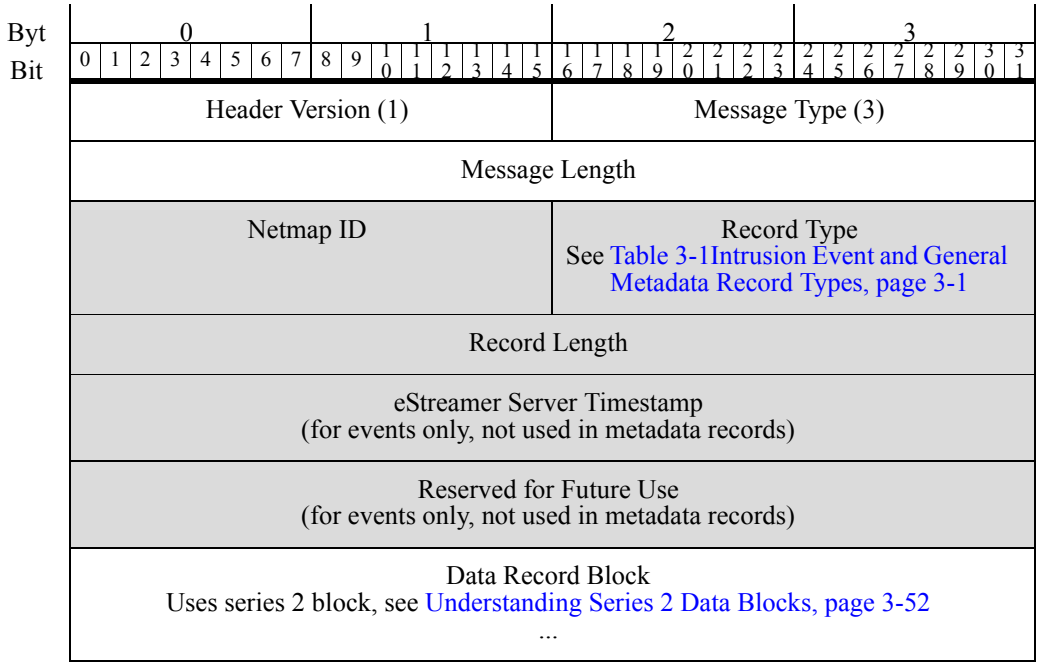
The graphic below shows the structure of event extra data messages. The Intrusion Event Extra Data message is an example of this message group.



Event extra data messages have the same format as correlation event messages, with a data block directly after the record header. Unlike correlation messages, they use series 2 data blocks, not series 1 data blocks, which have a separate numbering sequence. For information about series 2 block types, see Understanding Series 2 Data Blocks, page 3-52.

Event Extra Data Message Record Header

The shaded section of the following graphic shows the fields of the record header in event extra data messages. The table that follows defines the record header fields for event extra data messages.



The following table describes each field in the record header of event extra data messages.

Table 2-11 Event Extra Data Message Record Header Fields

Field	Data Type	Description
Netmap ID	uint16	The first bit of this field is a flag indicating whether the header is an extended header containing an archive timestamp. The remaining 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. If this field is not used, it is left empty. Netmap IDs map to domains as provided in metadata.
Record Type	uint16	Identifies the data record content type. See Table 3-1Intrusion Event and General Metadata Record Types, page 3-1 for the list of event extra data record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags. Field is not present for events generated by the Management Center.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags. Field is not present for events generated by the Management Center.

Data Block Header

Series 1 blocks and series 2 blocks have similar structures but distinct numbering. These blocks can appear anywhere in the data portion of a discovery, correlation, connection, or event extra data message. These blocks encapsulate other blocks at multiple levels of nesting.

The data blocks in both the first and second series begin with the header structure shown in the graphic below. The following table provides information about the header fields. The header is followed immediately by the data structure associated with the data block type.

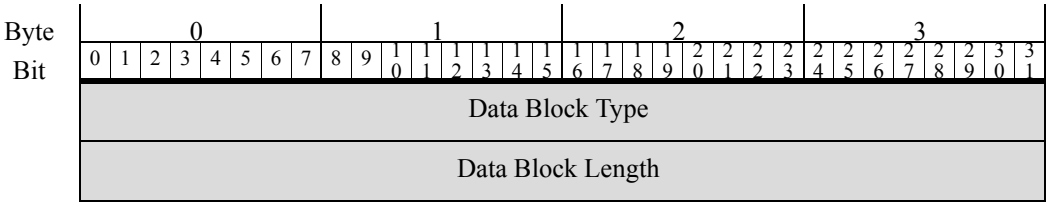


Table 2-12

Field	Data Type	Description
Data Block Type	uint32	For series 1 block types, see Understanding Discovery (Series 1) Blocks, page 4-55. For series 2 block types, see Table 3-26Series 2 Block Types, page 3-52.
Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.

Host Request Message Format

To receive host profiles, you submit Host Request messages. You can request data for a single host or multiple hosts defined by an IP address range.

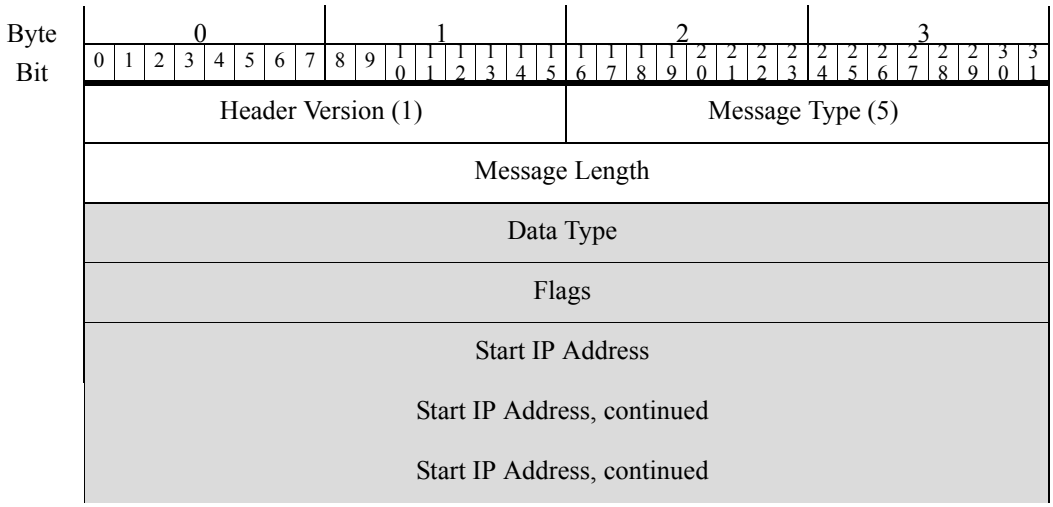
Note that it is mandatory for all data requests, including requests for host profile information, to first initialize the session by submitting an Event Stream Request message. To set up for streaming host data only, you can use any of the following request flag settings in your initial Event Stream Request message:

- set the bit for the appropriate version of metadata (this can be beneficial when streaming host data)
- set no request flags
- set bit 11 (to suppress any default event streaming if using legacy versions of eStreamer)

After the initial message, you then use a Host Request message (type 5) to specify the hosts.

Note For legacy eStreamer versions with default event streaming, if you want to stream only host profile data, you need to suppress the default event messages. First send the server an Event Stream Request message with bit 11 in the Request Flags field set to `1`; then, send the Host Request message.

The graphic below shows the format for the Host Request message. The shaded fields are specific to the Host Request message format and are defined in the following table. The preceding three fields are the standard message header.



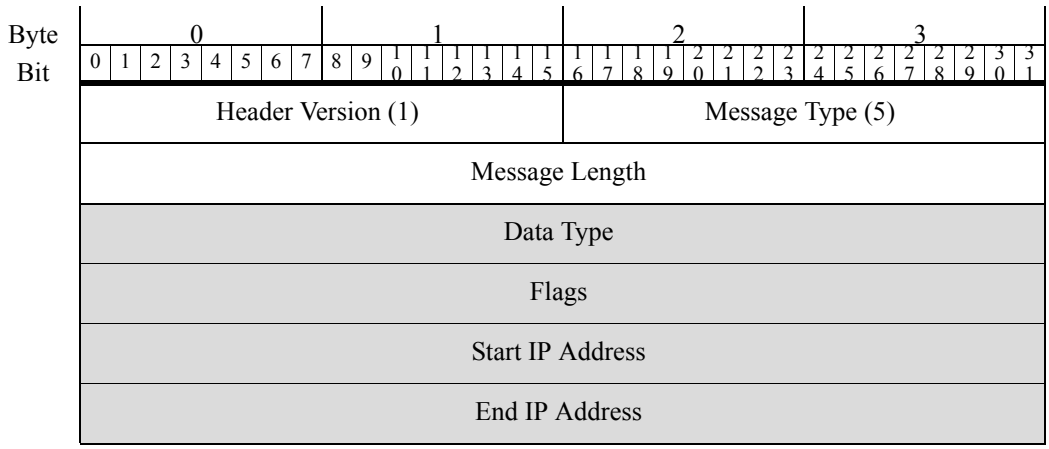
Start IP Address, continued
End IP Address
End IP Address, continued
End IP Address, continued
End IP Address, continued

The following table explains the message fields.

Table 2-13 Host Request Message Fields

Field	Data Type	Description
Data Type	uint32	Requests data for a single host or multiple hosts, using the following codes: • `0` — Version 3.5 - 4.6 for a single host. • `1` — Version 3.5 - 4.6 for multiple hosts (uses block 34). • `2` — Version 4.7 - 4.8 for a single host (uses block 47). • `3` — Version 4.7 - 4.8 for multiple hosts (uses block 47). • `4` — Version 4.9 - 4.10 for a single host (uses block 92). • `5` — Version 4.9 - 4.10 for multiple hosts (uses block 92). • `6` — Version 5.0+ data for a single host (uses block 111, see Full Host Profile Data Block 5.3+, page 5-1). • `7` — Version 5.0+ data for multiple hosts (uses block 111, see Full Host Profile Data Block 5.3+, page 5-1).
Flags	32-bit field	• `0x00000001` — Causes the Notes field of the host profile to be populated (with user-defined information about the host stored in the Firepower System). • `0x00000002` — Causes the Banner field of the service block to be populated (with the first 256 bytes of the first packet detected for the service). Banners are disabled by default and available only if configured.
Start IP Address	uint8[16]	IP address of the host whose data should be returned (if request is for a single host), or the starting address in an IP address range (if request is for multiple hosts). Can be either an IPv4 or IPv6 address.
End IP Address	uint8[16]	Ending address in an IP address range (if request is for multiple hosts), or the Start IP Address value (if request is for single host). Can be either an IPv4 or IPv6 address.

The graphic below shows the format for the legacy Host Request message. eStreamer will still respond to this request. The only difference from the current request is the smaller IPv4 address fields. The shaded fields are specific to the Host Request message format and are defined in the following table. The preceding three fields are the standard message header.



The following table explains the message fields.

Table 2-14 Host Request Message Fields

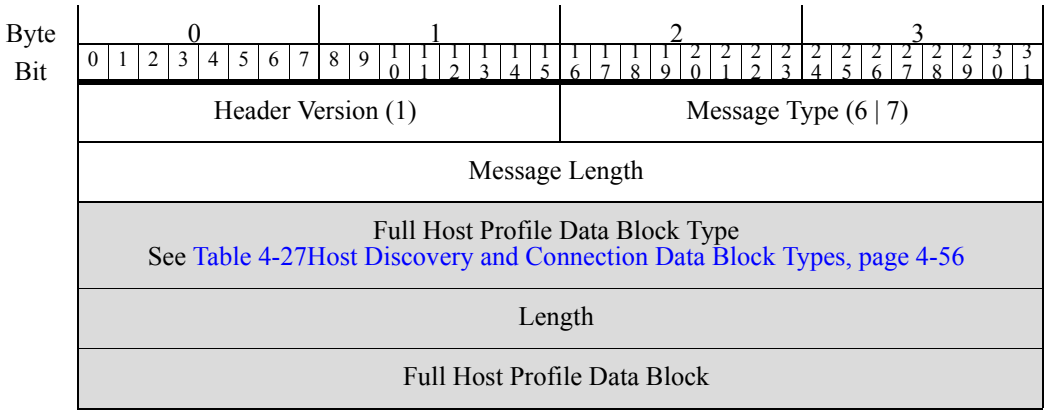
Field	Data Type	Description
Data Type	uint32	Requests data for a single host or multiple hosts, using the following codes: • `0` — Version 3.5 - 4.6 for a single host. • `1` — Version 3.5 - 4.6 for multiple hosts (uses block 34). • `2` — Version 4.7 - 4.8 for a single host (uses block 47). • `3` — Version 4.7 - 4.8 for multiple hosts (uses block 47). • `4` — Version 4.9 - 4.10 for a single host (uses block 92). • `5` — Version 4.9 - 4.10 for multiple hosts (uses block 92). • `6` — Version 5.0+ data for a single host (uses block 111, see Full Host Profile Data Block 5.3+, page 5-1). • `7` — Version 5.0+ data for multiple hosts (uses block 111, see Full Host Profile Data Block 5.3+, page 5-1).
Flags	32-bit field	• `0x00000001` — Causes the Notes field of the host profile to be populated (with user-defined information about the host stored in the Firepower System). • `0x00000002` — Causes the Banner field of the service block to be populated (with the first 256 bytes of the first packet detected for the service). Banners are disabled by default and available only if configured.
Start IP Address	uint8[4]	IP address of the host whose data should be returned (if request is for a single host), or the starting address in an IP address range (if request is for multiple hosts). Specify the address in IP address octets.
End IP Address	uint8[4]	Ending address in an IP address range (if request is for multiple hosts), or the Start IP Address value (if request is for single host).

Host Data and Multiple Host Data Message Format

eStreamer responds to host requests by sending host data messages, each with a full host profile data block. eStreamer sends one host data message for each host specified in the request. eStreamer uses the type 6 message to respond to requests for a single host profile, and uses the type 7 message to respond to requests for multiple hosts. The formats of the type 6 and type 7 messages are identical, only the message type is different.

Host data messages do not have a record type field. The structure of the message is communicated by the message type and the data block type of the full host profile included in the message. Full host profile data blocks are in the series a group of blocks.

The graphic below shows the format of the host data message and the table that follows defines the shaded fields:



The fields specific to the Host Request message are:

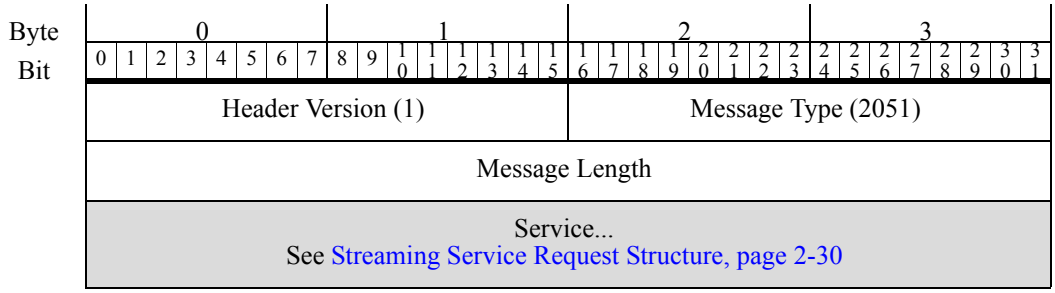
Table 2-15

Field	Data Type	Description
Full Host Profile Data Block Type	uint32	Specifies the block type for the full host profile data included in the message. See Table 4-27Host Discovery and Connection Data Block Types, page 4-56.
Length	uint32	Length of the full host profile data in the message.
Full Host Profile Data Block	variable	The host data. For links to the definitions of current full host profile data blocks, see Table 4-27Host Discovery and Connection Data Block Types, page 4-56.

Streaming Information Message Format

When the eStreamer service receives a request for an extended request, it sends the client the Streaming Information message described below. This message advertises the server's list of available services. Currently, the only relevant option is the eStreamer service (6667), although the message can list other services, which should be ignored. Each advertised service is represented by a Streaming Service Request structure described in Streaming Service Request Structure, page 2-30.

The graphic below illustrates the format for the Streaming Information message. The shaded field is specific to this message type. The preceding three fields are the standard message header.



The fields of the Streaming Information message are:

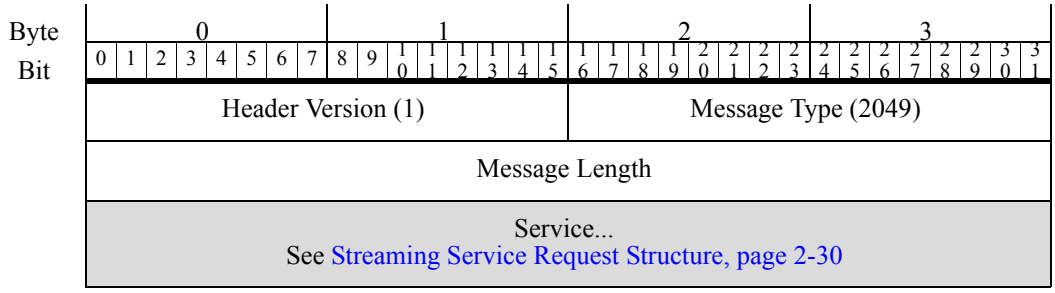
Table 2-16 Streaming Information Message Fields

Field	Data Type	Description
Header Version	uint16	Set to `1`.
Message Type	uint16	eStreamer message type. Set to `2051` for Streaming Request messages.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the Header Version, Message Type, and Message Length fields.
Service[]	array	List of available services. See Streaming Service Request Structure, page 2-30.

Streaming Request Message Format

The client uses the Streaming Request message to specify to eStreamer the service in the Streaming Information message that it wants to use, followed by a set of requests for event types and versions to be streamed. The graphic below shows the message structure and the following table defines the fields. The requested service is represented by a Streaming Service Request structure described in Streaming Service Request Structure, page 2-30.

The graphic below illustrates the format for the Streaming Information message. The shaded field is specific to this message type. The preceding three fields are the standard message header.



The fields of the Streaming Request message are:

Table 2-17 Streaming Request Message Fields

Field	Data Type	Description
Header Version	uint16	Set to `1`.
Message Type	uint16	eStreamer message type. Set to `2049` for Streaming Request messages.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the Header Version, Message Type, and Message Length fields.
Service[]	array	List of requested service structures. See Streaming Service Request Structure, page 2-30.

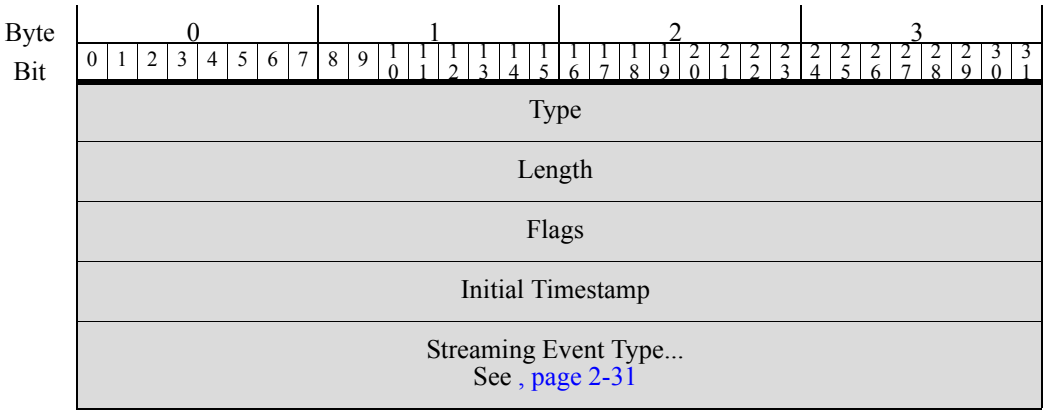
Streaming Service Request Structure

The eStreamer service sends one Streaming Service Request data structure in the Streaming Information message for each service it advertises. The eStreamer service does not use the last field of the Streaming Service Request, which provides for a list of event types to be included.

The client processes the Streaming Service Request structure from eStreamer and uses the same structure in the response it returns to the server. In the Streaming Service Request that the client sends to the server, it includes, first, a request for the service advertised by eStreamer, and, second, a list of Streaming Event Type structures, which specify the requested event types the client wants to receive.

Each Streaming Event Type structure contains two fields to specify the event type and version for each requested event type. For information on the Streaming Event Type structure, see , page 2-31.

The graphic below shows the fields of the Streaming Service Request structure. The table that follows defines the fields.



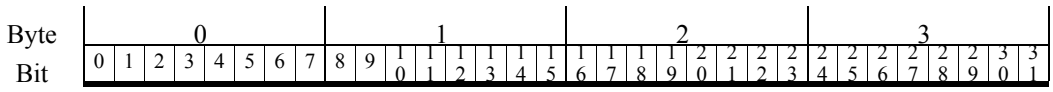
The fields of the Streaming Service Request structure are:

Table 2-18 Streaming Service Request Fields

Field	Data Type	Description
Type	uint32	Service ID. In eStreamer server messages, this advertises an available service. In client messages, it specifies a requested service. Current valid options: • `6667` (for eStreamer service)
Length	uint32	Service request length. Describes the length of the service request, including Type and Length. Note that Length must include all the Streaming Event Type records in the message, plus the terminating one.
Flags	uint32	In eStreamer's Streaming Information messages: Always `0`. In client's Streaming Request message: replicates the flag settings in the original Event Stream Request message.
Initial Timestamp	uint32	In eStreamer's Streaming Information messages: Always `0`. In client's Streaming Request message: replicates the timestamp in the original Event Stream Request message.
Streaming Event Type	array	In eStreamer's Streaming Information message: • Reserved for future use. Has `0` length. In client's Streaming Request message: • One Streaming Event Type entry for each requested event type. See , page 2-31. • Terminate the request list with a `0` Event Type entry, with both Event Type and Version set to `0`. See , page 2-31.

Domain Streaming Request Message Format

The client uses the Domain Streaming Request message to request events from a specific domain from eStreamer. The graphic below shows the message structure and the following table defines the fields. The shaded fields are specific to this message type. The preceding three fields are the standard message header.



Header Version (1)	Message Type (2052)
Message Length	
String Block Type (0)	
String Block Length	
Domain...	

The fields of the Domain Streaming Request message are:

Table 2-19 Domain Streaming Request Message Fields

Field	Data Type	Description
Header Version	uint16	Set to `1`.
Message Type	uint16	eStreamer message type. Set to `2052` for Domain Streaming Request messages.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the Header Version, Message Type, and Message Length fields.
String Block Type	uint32	Initiates a String data block containing the domain. This value is always `0`.
String Block Length	uint32	The number of bytes included in the domain String data block, including eight bytes for the block type and header fields plus the number of bytes in the domain.
Domain	string	Domain from which streaming events are requested. If left blank, the service will stream events for all domains to which the client has access.

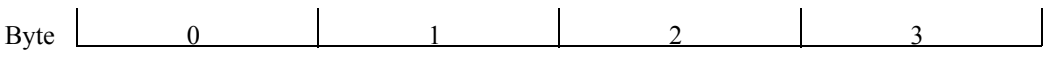
Streaming Event Type Structure

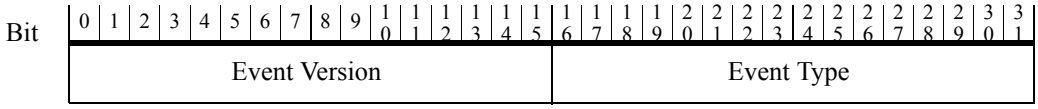
eStreamer clients use the Streaming Event Type structure to specify an event's version and version. Each event version/type combination is a request for an event stream.

Lists of Streaming Event Type structures must be terminated with a structure with all fields set to zero. That is:

```
Event Version = 0
Event Type = 0
```

The following diagram illustrates the format for the Streaming Event Type structure.





The fields of the Streaming Event Type structure are:

Table 2-20 Streaming Event Type Fields

Field	Data Type	Description
Event Version	uint16	Version number of event type. For list of versions supported for each event type, see Table 2-21Event Types and Versions for Extended Request, page 2-33.
Event Type	uint16	Code for requested event type. For the current list of valid event types and version codes, see Table 2-21Event Types and Versions for Extended Request, page 2-33. List of event types should be terminated with a zero event type and zero event version.

The following table lists the event types and versions that clients can specify in extended requests. The table indicates the Management Center software versions that correspond to each event type version. For example, to request the correlation events that were supported by the Management Center in version 4.8.0.2 - 4.9.1, you should request Event Type 31, Version 5. If an event was recorded with a different event type, it will be upgraded or downgraded to match the format of the requested event type.

Table 2-21 Event Types and Versions for Extended Request

To request...	Use this event version number...	And this event code
intrusion events	1 — 4.8.x and earlier 2 — 4.9 - 4.10.x 3 — 5.0 - 5.1 4 — 5.1.1.x 5 — 5.2.x 6 — 5.3 7 — 5.3.1 8 — 5.4.x 9— 6.0+	12
metadata	1 — 3.2 - 4.5.x 2 — 4.6.0.x 3 — 4.6.1 - 4.6.x 4 — 4.7+	21
correlation and compliance white list events	1 — 3.2 and earlier 2 — 4.0 - 4.4.x 3 — 4.5 - 4.6.1 4 — 4.7 - 4.8.0.1 5 — 4.8.0.2 - 4.9.1.x 6 — 4.10.0 - 4.10.x 7 — 5.0 - 5.0.2 8 — 5.1 - 5.3.x 9 — 5.4+	31

Table 2-21 Event Types and Versions for Extended Request (continued)

To request...	Use this event version number...	And this event code
discovery events	1 — 3.2 and earlier 2 — 3.0 - 3.4.x 3 — 3.5 - 4.6.x 4 — 4.7 - 4.8.x 5 — 4.9.0.x 6 — 4.9.1 - 4.9.x.x 7 — 4.10.0 - 4.10.x 8 — 5.0.x 9 — 5.1.x 10 — 5.2 - 5.3 11 — 5.3.1+	61
connection events	1 — 4.0 - 4.1 3 — 4.5 - 4.6.1 4 — 4.7 - 4.9.0.x 5 — 4.9.1 - 4.10.x 6 — 5.0.x 7 — 5.1.0.x 8 — 5.1.1.x 9 — 5.2.x 10 — 5.3 11 — 5.3.1 12 — 5.4 13 — 5.4.0.1-5.4.0.2 14 — 6.0+	71
user events	1 — 4.7 - 4.10.x 2 — 5.0.x 3 — 5.1-5.1.x 4 — 5.2+	91
malware events	1 — 5.1.0.x 2 — 5.1.1.x 3 — 5.2.x 4 — 5.3 5 — 5.3.1 6 — 5.4.x 7— 6.0+	101
file events	1 — 5.1.1 - 5.1.x 2 — 5.2.x 3 — 5.3 4 — 5.3.1 5 — 5.4.x 6— 6.0+	111
impact correlation events	1 — 5.2.x and earlier 2 — 5.3+	131
terminating event type in a list	0	0

Sample Extended Request Messages

Streaming Information Message

In the sample below, the server advertises two services, the first type 6667 (eStreamer) and the second type 5000. In Streaming Information messages from the server, the flags field and initial timestamp fields are zero, and the message specifies no event types.

Table 2-22

Header Version:	1	/*always 1*/
Message Type:	2051	/*streaming info msg*/
Message Length	32	/*bytes of msg content*/
Service[1].Type	6667	/*eStreamer service ID*/
Service[1].Length	8	
Service[1].Flags	0	/*no flags from server*/
Service[1].Initial Timestamp	0	/*always 0*/
Service[2].Type	5000	/*service-2 ID*/
Service[2].Length	8	
Service[2].Flags	0	/*no flags from server*/
Service[2].Initial Timestamp	0	/*always 0*/
Header Version:	1	/*always 1*/
Message Type:	2051	/*streaming info msg*/

Streaming Request Message

Below is a Streaming Request message where the client requests service type 6667 (eStreamer) and specifies two event types: version 6 of connection events (event type 71) and version 4 of metadata (event type 21).

Table 2-23

Header Version:	1	/*always 1*/
Message Type:	2049	/*stream request msg*/
Message Length	28	/*payload bytes*/
Service[1].Type	6667	/*eStreamer service ID*/
Service[1].Length	20	
Service[1].Flags	30	/*original flags value*/
Service[1].Initial Timestamp	0	/*original timestamp*/
Service[1].Event[1].Version	6	/*version 6*/
Service[1].Event[1].Type	71	/*connection events*/
Service[1].Event[2].Version	4	/* version 4*/

Table 2-23

Service[1].Event[2].Type	`21`	/*metadata*/
Service[1].Event[3].Version	`0`	/*terminate event list*/
Service[1].Event[3].Type	`0`	/*terminate event list*/

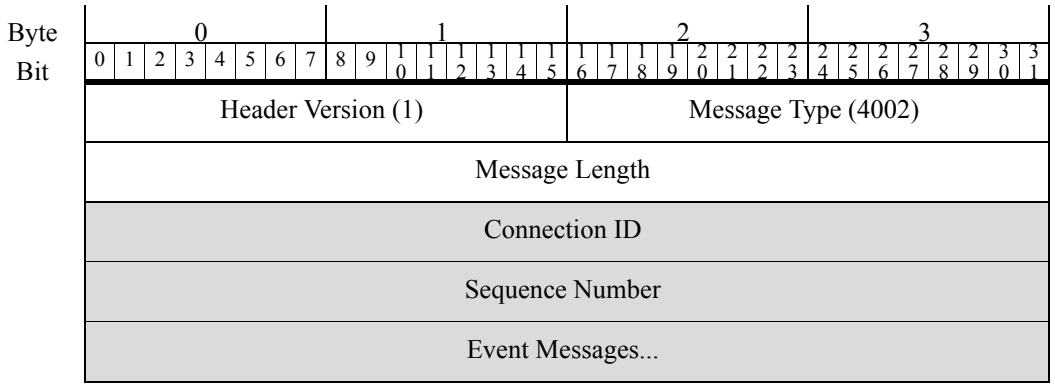
Message Bundle Format

The eStreamer server sends messages in a bundle format when the client submits an extended request.

The client responds with a null message to acknowledge receipt of an entire bundle. The client should not acknowledge receipt of individual messages in a bundle.

Message bundles have a message type of `4002`.

The graphic below shows the structure of a message bundle. The shaded fields are specific to the bundle message type. The following table describes the content of the fields and data structures.



The fields of a message bundle message are:

Table 2-24 Message Bundle Message Fields

Field	Data Type	Description
Header Version	uint16	Always `1`.
Message Type	uint16	Always `4002`.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the bundle's Header Version, Message Type, and Message Length fields. As the client loads a message from the bundle, it can subtract the message's total length (including header) from the length in this field. As long as the remainder is positive, there are more messages to process.
Connection ID	uint32	A unique identifier for the connection with the server.

Table 2-24 Message Bundle Message Fields (continued)

Field	Data Type	Description
Sequence Number	uint32	Starts at `1` and increments by one for each bundle sent by the eStreamer server.
Event Messages []	array	The events streamed by the server in the bundle. Each message has a full set of headers, including message version number (1), archive timestamp if requested, and so forth.

Understanding Metadata

The eStreamer server can provide metadata along with requested event records. To receive metadata, you must explicitly request it. See Table 2-6Request Flags, page 2-12 for information on how to request a given version of metadata. The metadata provides context information for codes and numeric identifiers in the event records. For example, an intrusion event contains only the internal identifier of the detecting device, and the metadata provides the device's name.

Metadata Transmission

If the request message specifies metadata, eStreamer sends the relevant metadata record before it sends any related event records.

eStreamer keeps track of the metadata it has sent to the client and does not resend the same metadata record. The client should cache each received metadata record. eStreamer does not keep a history of metadata transmissions from one session to the next, so when a new session starts and a request message specifies metadata, eStreamer restarts metadata streaming from scratch.

Understanding Intrusion and Correlation Data Structures

The eStreamer service transmits a number of data record types to deliver requested events and metadata to the client. This chapter describes the structures of data records for the following types of event data:

- intrusion events data and event extra data generated by managed devices
- correlation (compliance) events generated by the Management Center
- metadata records

The following sections in this chapter define the event message structures:

- Intrusion Event and Metadata Record Types, page 3-1.

For a general overview eStreamer's message format for transmitting data records, see Event Data Message Format, page 2-17.

Intrusion Event and Metadata Record Types

The table that follows lists all currently supported record types for intrusion events, intrusion event extra data, and metadata messages. The data for these record types is in fixed-length fields. By contrast, correlation event records contain one or more levels of nested data blocks with variable lengths. The table below provides a link to the chapter subsection that defines the associated data record structure.

For some record types, eStreamer supports more than one version. The table indicates the status of each version (current or legacy). A current record is the latest version. A legacy record has been superseded by a later version but can still be requested from eStreamer.

Table 3-1 Intrusion Event and General Metadata Record Types

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
2	N/A	N/A	Packet Data (Version 4.8.0.2+)	Current	Packet Record 4.8.0.2+, page 3-5
4	N/A	N/A	Priority Metadata	Current	Priority Record, page 3-6
9	20	1	Intrusion Impact Alert	Legacy	Intrusion Impact Alert Data, page B-44
9	153	1	Intrusion Impact Alert	Current	Intrusion Impact Alert Data 5.3+, page 3-16
62	57	2	User Metadata	Current	User Record, page 3-19

Table 3-1 Intrusion Event and General Metadata Record Types (continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
66	N/A	N/A	Rule Message Metadata (Version 4.6.1+)	Current	Rule Message Record for 4.6.1+, page 3-20
67	N/A	N/A	Classification Metadata (Version 4.6.1+)	Current	Classification Record for 4.6.1+, page 3-22
69	N/A	N/A	Correlation Policy Metadata (Version 4.6.1+)	Current	Correlation Policy Record, page 3-23
70	N/A	N/A	Correlation Rule Metadata (Version 4.6.1+)	Current	Correlation Rule Record, page 3-25
104	N/A	N/A	Intrusion Event (IPv4) Record 4.9 - 4.10.x	Legacy	earlier versions of the product
105	N/A	N/A	Intrusion Event (IPv6) Record 4.9-4.10.x	Legacy	earlier versions of the product
110	4	2	Intrusion Event Extra Data (Version 4.10.0+)	Current	Intrusion Event Extra Data Record, page 3-26
111	5	2	Intrusion Event Extra Data Metadata (Version 4.10.0+)	Current	Intrusion Event Extra Data Metadata, page 3-28
112	128	1	Correlation Event for 5.1-5.3.x	Legacy	Correlation Event for 5.1-5.3.x, page B-217
112	156	1	Correlation Event for 5.4+	Current	Correlation Event for 5.4+, page 3-41
115	14	2	Security Zone Name Metadata	Current	Security Zone Name Record, page 3-29
116	14	2	Interface Name Metadata	Current	Interface Name Record, page 3-31
117	14	2	Access Control Policy Name Metadata	Current	Access Control Policy Name Record, page 3-32
118	15	2	Intrusion Policy Name Metadata	Current	Intrusion Policy Name Record, page 4-21
119	15	2	Access Control Rule ID Metadata	Current	Access Control Rule ID Record Metadata, page 3-33
120	N/A	N/A	Access Control Rule Action Metadata	Current	Access Control Rule Action Record Metadata, page 4-22
121	N/A	N/A	URL Category Metadata	Current	URL Category Record Metadata, page 4-23
122	N/A	N/A	URL Reputation Metadata	Current	URL Reputation Record Metadata, page 4-23
123	N/A	N/A	Managed Device Metadata	Current	Managed Device Record Metadata, page 3-34
N/A	64	2	Access Control Name Data Block	Current	Access Control Policy Name Data Block, page 3-70
124	59	2	Access Control Policy Rule Reason Data Block	Current	Access Control Policy Rule Reason Data Block for 6.0+, page 3-69
125	N/A	2	Malware Event Record (Version 5.1.1+)	Current	Malware Event Record 5.1.1+, page 3-35

Table 3-1 Intrusion Event and General Metadata Record Types (continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
125	24	2	Malware Event (Version 5.1.1+)	Current	Malware Event Data Block 5.1.1.x, page B-50
125	33	2	Malware Event (Version 5.2.x)	Legacy	Malware Event Data Block 5.2.x, page B-56
125	35	2	Malware Event (Version 5.3)	Legacy	Malware Event Data Block 5.3, page B-63
125	44	2	Malware Event (Version 5.3.1)	Legacy	Malware Event Data Block 5.3.1, page B-70
125	47	2	Malware Event (Version 5.4.x)	Current	Malware Event Data Block 5.4.x, page B-77
125	62	2	Malware Event (Version 6.0+)	Current	Malware Event Data Block 6.0+, page 3-83
127	14	2	Cisco Advanced Malware Protection Cloud Name Metadata (Version 5.1+)	Current	Cisco Advanced Malware Protection Cloud Name Metadata, page 3-35
128	N/A	N/A	Malware Event Type Metadata (Version 5.1+)	Current	Malware Event Type Metadata, page 3-37
129	N/A	N/A	Malware Event Subtype Metadata (Version 5.1+)	Current	Malware Event Subtype Metadata, page 3-38
130	N/A	N/A	AMP for Endpoints Detector Type Metadata (Version 5.1+)	Current	AMP for Endpoints Detector Type Metadata, page 3-39
131	N/A	N/A	AMP for Endpoints File Type Metadata (Version 5.1+)	Current	AMP for Endpoints File Type Metadata, page 3-39
132	N/A	N/A	Security Context Name	Current	Security Context Name, page 3-40
140	27	2	Rule Documentation Data Block for 5.2+	Current	Rule Documentation Data Block for 5.2+, page 3-97
207	N/A	N/A	Intrusion Event (IPv4) Record 5.0.x - 5.1	Legacy	Intrusion Event (IPv4) Record 5.0.x - 5.1, page B-2
208	N/A	N/A	Intrusion Event (IPv6) Record 5.0.x - 5.1	Legacy	Intrusion Event (IPv6) Record 5.0.x - 5.1, page B-6
260	19	2	ICMP Type Data Data Block	Current	ICMP Type Data Block, page 3-62
270	20	2	ICMP Code Data Block	Current	ICMP Code Data Block, page 3-63
282	N/A	2	Security Intelligence Category Metadata for 5.4.1+	Current	Security Intelligence Category Metadata for 5.4.1+, page 3-64
300	N/A	N/A	Realm Metadata for 6.0+	Current	Realm Metadata for 6.0+, page 3-65
301	58	2	Endpoint Profile for 6.0+	Current	Endpoint Profile Data Block for 6.0+, page 3-66
302	N/A	N/A	Security Group Metadata for 6.0+	Current	Security Group Metadata for 6.0+, page 3-67
322	N/A	N/A	Sinkhole Metadata for 6.0+	Current	Sinkhole Metadata for 6.0+, page 3-68

Table 3-1 Intrusion Event and General Metadata Record Types (continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
350	N/A	N/A	Netmap Domain Metadata for 6.0+	Current	Netmap Domain Metadata for 6.0+, page 3-69
400	34	2	Intrusion Event Record 5.2.x	Legacy	Intrusion Event Record 5.2.x, page B-12
400	41	2	Intrusion Event Record 5.3	Legacy	Intrusion Event Record 5.3, page B-17
400	42	2	Intrusion Event Record 5.3.1	Legacy	Intrusion Event Record 5.3.1, page B-29
400	45	2	Intrusion Event Record 5.4.x	Legacy	Intrusion Event Record 5.4.x, page B-36
400	60	2	Intrusion Event Record 6.0+	Current	Intrusion Event Record 6.0+, page 3-7
500	32	2	File Event (Version 5.2.x)	Legacy	File Event for 5.2.x, page B-182
500	38	2	File Event (Version 5.3)	Legacy	File Event for 5.3, page B-186
500	43	2	File Event (Version 5.3.1)	Legacy	File Event for 5.3.1, page B-192
500	46	2	File Event (Version 5.4+)	Current	File Event for 6.0+, page 3-73
502	32	2	File Event (Version 5.2.x)	Legacy	File Event for 5.2.x, page B-182
502	38	2	File Event (Version 5.3)	Legacy	File Event for 5.3, page B-186
502	43	2	File Event (Version 5.3.1)	Legacy	File Event for 5.3.1, page B-192
502	46	2	File Event (Version 5.4.x)	Current	File Event for 5.4.x, page B-198
502	56	2	File Event (Version 6.0+)	Current	File Event for 6.0+, page 3-73
510	N/A	N/A	File Type ID Metadata for 5.3+	Current	File Type ID Metadata for 5.3+, page 3-96
511	26	2	File Event SHA Hash for 5.11-5.2.x	Legacy	File Event SHA Hash for 5.1.1-5.2.x, page B-208
511	40	2	File Event SHA Hash for 5.3+	Current	File Event SHA Hash for 5.3+, page 3-94
515	N/A	N/A	Filelog Storage Metadata for 6.0+	Current	Filelog Storage Metadata for 6.0+, page 3-101
516	N/A	N/A	Filelog Sandbox Metadata for 6.0+	Current	Filelog Sandbox Metadata for 6.0+, page 3-101
517	N/A	N/A	Filelog Spero Metadata for 6.0+	Current	Filelog Spero Metadata for 6.0+, page 3-102
518	N/A	N/A	Filelog Archive Metadata for 6.0+	Current	Filelog Archive Metadata for 6.0+, page 3-103
519	N/A	N/A	Filelog Static Analysis Metadata for 6.0+	Current	Filelog Static Analysis Metadata for 6.0+, page 3-104
520	28	2	Geolocation Data Block for 5.2+	Current	Geolocation Data Block for 5.2+, page 3-104
530	N/A	N/A	File Policy Name for 6.0+	Current	File Policy Name for 6.0+, page 3-105
600	N/A	N/A	SSL Policy Name	Current	SSL Policy Name, page 3-106
601	51	2	SSL Rule ID	Current	SSL Rule ID, page 3-107

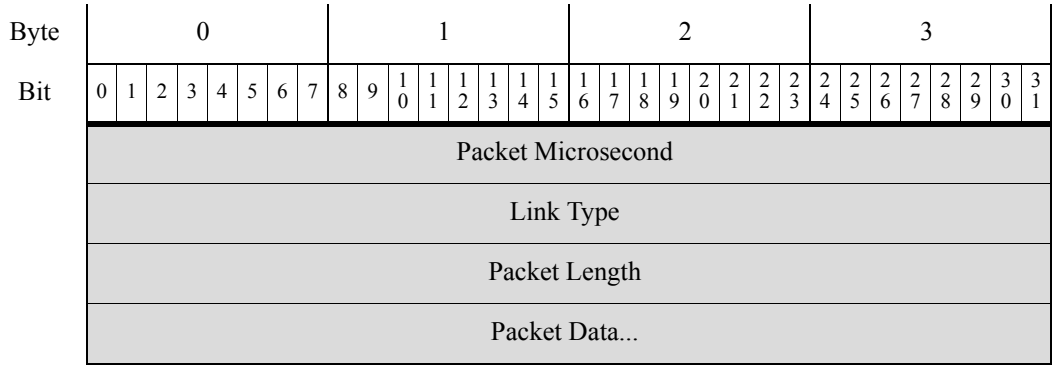
Table 3-1 Intrusion Event and General Metadata Record Types (continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
602	N/A	N/A	SSL Cipher Suite	Current	SSL Certificate Details Data Block for 5.4+, page 3-114
604	N/A	N/A	SSL Version	Current	SSL Version, page 3-109
605	N/A	N/A	SSL Server Certificate Status	Current	SSL Server Certificate Status, page 3-110
606	N/A	N/A	SSL Actual Action	Current	SSL Actual Action, page 3-111
607	N/A	N/A	SSL Expected Action	Current	SSL Expected Action, page 3-112
608	N/A	N/A	SSL Flow Status	Current	SSL Flow Status, page 3-112
613	N/A	N/A	SSL URL Category	Current	SSL URL Category, page 3-113
614	50	2	SSL Certificate Details Data Block for 5.4+	Current	SSL Certificate Details Data Block for 5.4+, page 3-114
700	N/A	N/A	Network Analysis Policy Record	Current	Network Analysis Policy Name Record, page 3-118

Packet Record 4.8.0.2+

The eStreamer service transmits the packet data associated with an event in a Packet record, the format of which is shown below. Packet data is sent when the Packet flag—bit 0 in the Request Flags field of a request message—is set. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record. Note that the Record Type field, which appears after the Message Length field, has a value of 2, indicating a packet record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (2)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Device ID																																
Event ID																																
Event Second																																
Packet Second																																



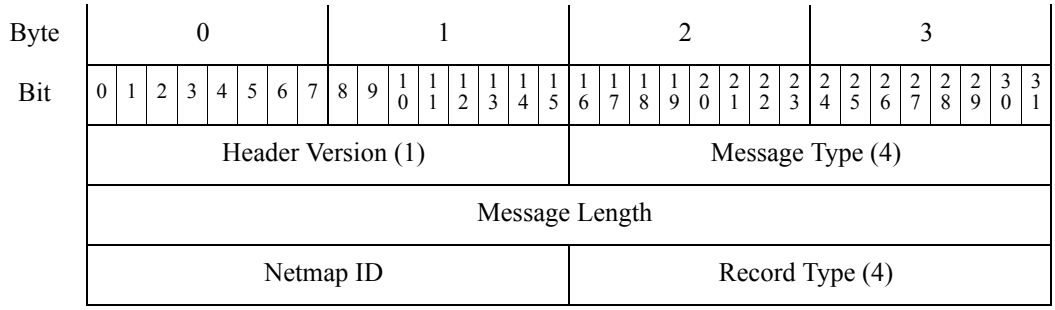
The following table describes the fields in the Packet record.

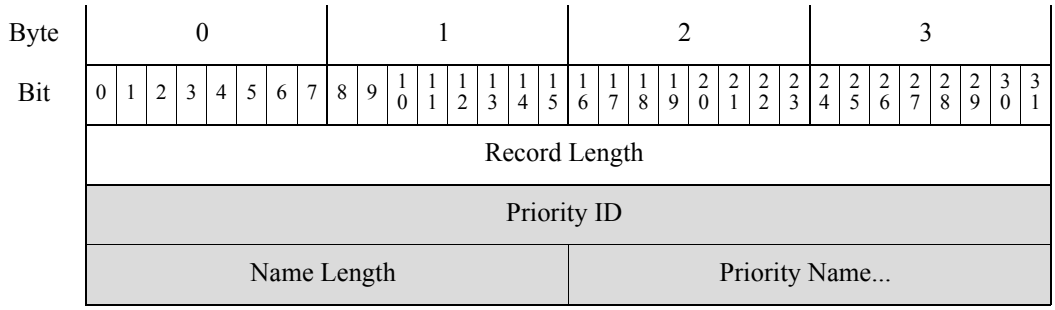
Table 3-2 Packet Record Fields

Field	Data Type	Description
Device ID	uint32	The device identification number. You can obtain device names that correlate to them by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	The event identification number.
Event Second	uint32	The second (from 01/01/1970) that the event occurred.
Packet Second	uint32	The second (from 01/01/1970) that the packet was captured.
Packet Microsecond	uint32	Microsecond (one millionth of a second) increment that the packet was captured.
Link Type	uint32	Link layer type. Currently, the value will always be `1` (signifying the Ethernet layer).
Packet Length	uint32	Number of bytes included in the packet data.
Packet Data	variable	Actual captured packet data (header and payload).

Priority Record

The eStreamer service transmits the priority associated with an event in a Priority record, the format of which is shown below. (Priority information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `4`, indicating a Priority record.





The following table describes each priority-specific field.

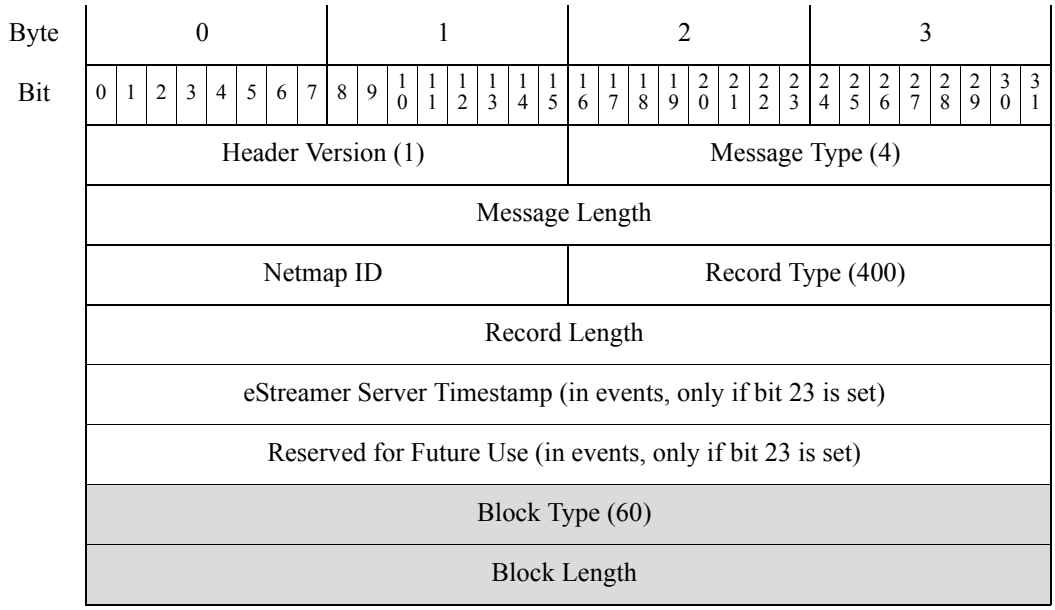
Table 3-3 Priority Record Fields

Field	Data Type	Description
Priority ID	uint32	Indicates the priority identification number.
Name Length	uint16	Number of bytes included in the priority name.
Priority Name	variable	Priority name that corresponds with the priority ID (1 - high, 2 - medium, 3 - low).

Intrusion Event Record 6.0+

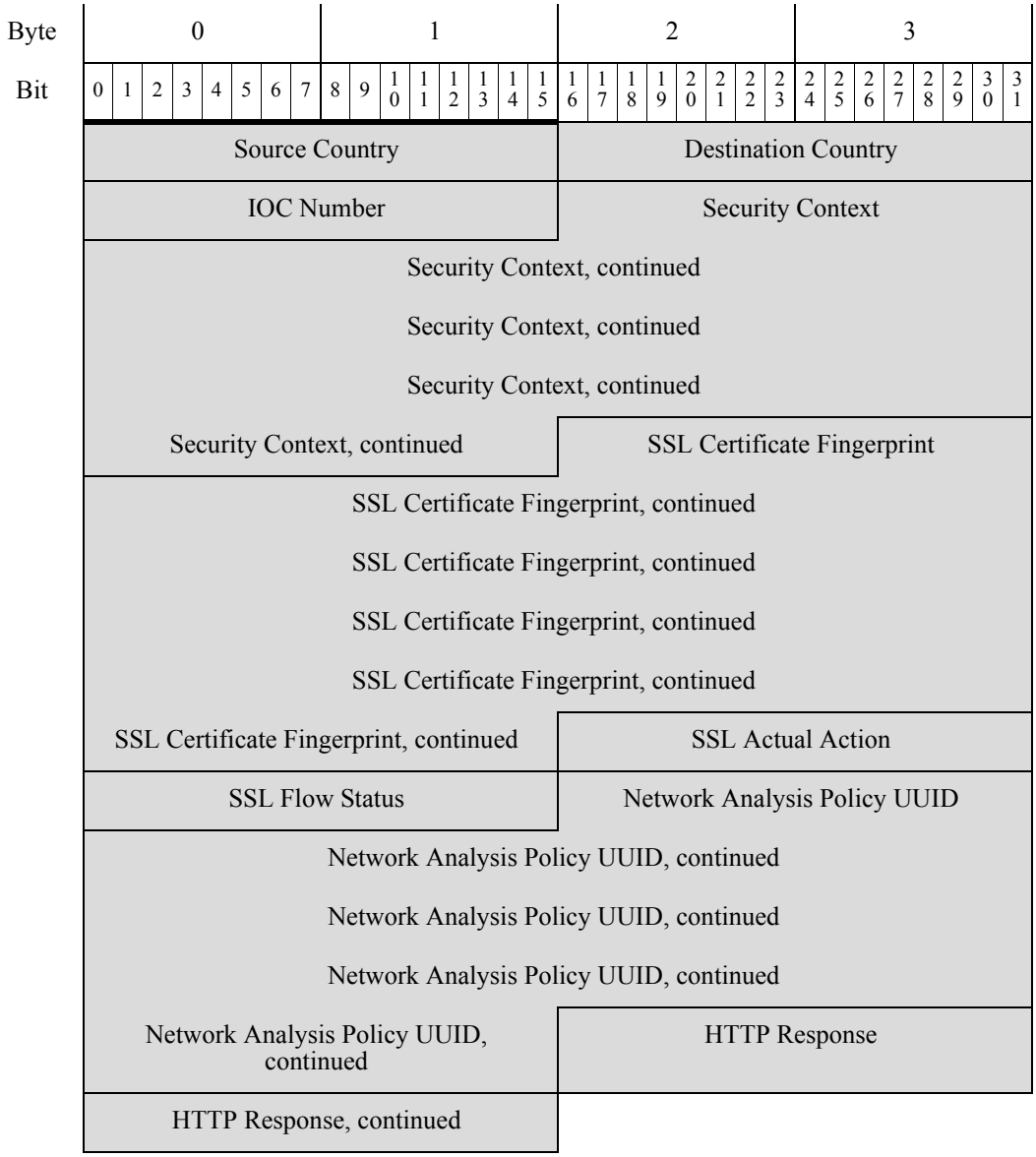
The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 60 in the series 2 set of data blocks. It supersedes block type 45. An HTTP Response field has been added.

You can request 6.0+ intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 9 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Source Port or ICMP Type																	Destination Port or ICMP Code														
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued																																
Interface Egress UUID Interface Egress UUID, continued Interface Egress UUID, continued Interface Egress UUID, continued																																
Security Zone Ingress UUID Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued																																
Security Zone Egress UUID Security Zone Egress UUID, continued Security Zone Egress UUID, continued Security Zone Egress UUID, continued																																
Connection Timestamp																																
Connection Instance ID																	Connection Counter															



The following table describes each intrusion event record data field.

Table 3-4 Intrusion Event Record 6.0+ Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 60.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.

Table 3-4 Intrusion Event Record 6.0+ Fields (continued)

Field	Data Type	Description
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol ID	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP

Table 3-4 Intrusion Event Record 6.0+ Fields (continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Management Center. An `x` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — Gray (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • `0` — Not blocked • `1` — Blocked • `2` — Would be blocked (but not permitted by configuration)

Table 3-4 Intrusion Event Record 6.0+ Fields (continued)

Field	Data Type	Description
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Interface Ingress UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Interface Egress UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Security Zone Ingress UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Security Zone Egress UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8[16]	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.

Table 3-4 Intrusion Event Record 6.0+ Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table 3-4 Intrusion Event Record 6.0+ Fields (continued)

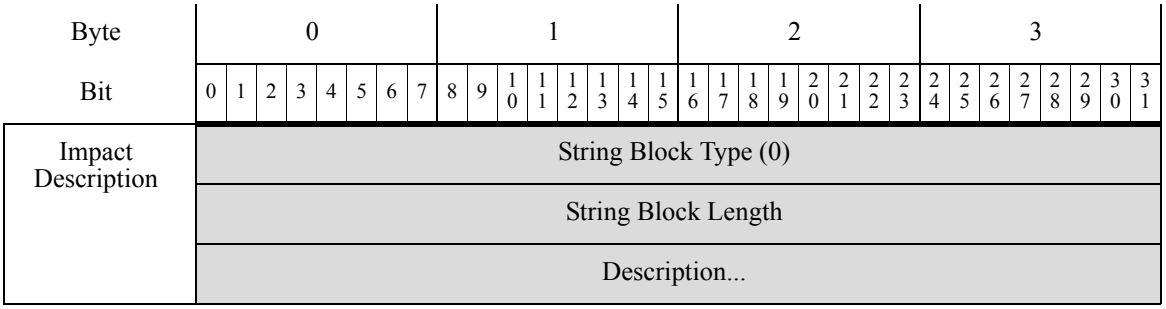
Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
Network Analysis Policy UUID	uint8[16]	The UUID of the Network Analysis Policy that created the intrusion event.
HTTP Response	uint32	Response code of the HTTP Request.

Intrusion Impact Alert Data 5.3+

The Intrusion Impact Alert 5.3+ event contains information about impact events. It is transmitted when an intrusion event is compared to the system network map data and the impact is determined. It uses the standard record header with a record type of 9, followed by an Intrusion Impact Alert data block with a series 1 data block type of 153 in the series 1 group of blocks. (The Impact Alert data block is a type of series 1 data block. For more information about series 1 data blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.)

You can request that eStreamer only transmit intrusion impact events by setting bit 5 in the Flags field of the request message. See Event Stream Request Message Format, page 2-10 for more information about request messages. Version 1 of these alerts only handles IPv4. Version 2, introduced in 5.3, handles IPv6 events in addition to IPv4.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (9)															
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Intrusion Impact Alert Block Length																																
Event ID																																
Device ID																																
Event Second																																
Impact																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																



The following table describes each data field in an impact event.

Table 3-5 Impact Event Data Fields

Field	Data Type	Description
Intrusion Impact Alert Block Type	uint32	Indicates that an intrusion impact alert data block follows. This field will always have a value of `20`. See Intrusion Event and Metadata Record Types, page 3-1.
Intrusion Impact Alert Block Length	uint32	Indicates the length of the intrusion impact alert data block, including all data that follows and 8 bytes for the intrusion impact alert block type and length.
Event ID	uint32	Indicates the event identification number.
Device ID	uint32	Indicates the managed device identification number.
Event Second	uint32	Indicates the second (from 01/01/1970) that the event was detected.

Table 3-5 Impact Event Data Fields (continued)

Field	Data Type	Description
Impact	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Management Center. An `x` indicates the value can be 0 or 1: • gray (0, unknown): `00x00000` • red (1, vulnerable): `xxxx1xxx`, `xxx1xxxx`, `x1xxxxxx`, `1xxxxxxx` (version 5.0+ only) • orange (2, potentially vulnerable): `00x0011x` • yellow (3, currently not vulnerable): `00x0001x` • blue (4, unknown target): `00x00001`
Source IP Address	uint8[16]	IP address of the host associated with the impact event. This can contain either an IPv4 or IPv6 address. See IP Addresses, page 1-6 for more information.
Destination IP Address	uint8[16]	IP address of the destination IP address associated with the impact event (if applicable). This can contain either an IPv4 or IPv6 address. See IP Addresses, page 1-6 for more information. This value is `0` if there is no destination IP address.
String Block Type	uint32	Initiates a string data block that contains the impact name. This value is always set to `0`. For more information about string blocks, see String Data Block, page 4-64.

Table 3-5 Impact Event Data Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the event description string block. This includes the four bytes for the string block type, the four bytes for the string block length, and the number of bytes in the description.
Description	string	Description of the impact event.

User Record

When you request metadata, you can retrieve information about the users referenced in events generated by components in your Firepower System. The eStreamer service transmits metadata containing user information for an event within a User record, the format of which is shown below. The user metadata record can be used to determine a user name associated with an event by correlating the metadata with the user ID value from a User Vulnerability Change Data Block, User Host Deletion Data Block, User Service Deletion Data Block, User Criticality Change Blocks, Attribute Definition Data Block, User Attribute Value Data Block, or Scan Result Data Block. (User information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of , indicating a User record. The User Record contains a User data block, block type 57 in series 2 data blocks.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (62)															
	Record Length																															
	Block Type (57)																															
	Block Length																															
	User ID																															
	Protocol																															
Name	String Block Type (0)																															
	String Block Length																															
	Name...																															

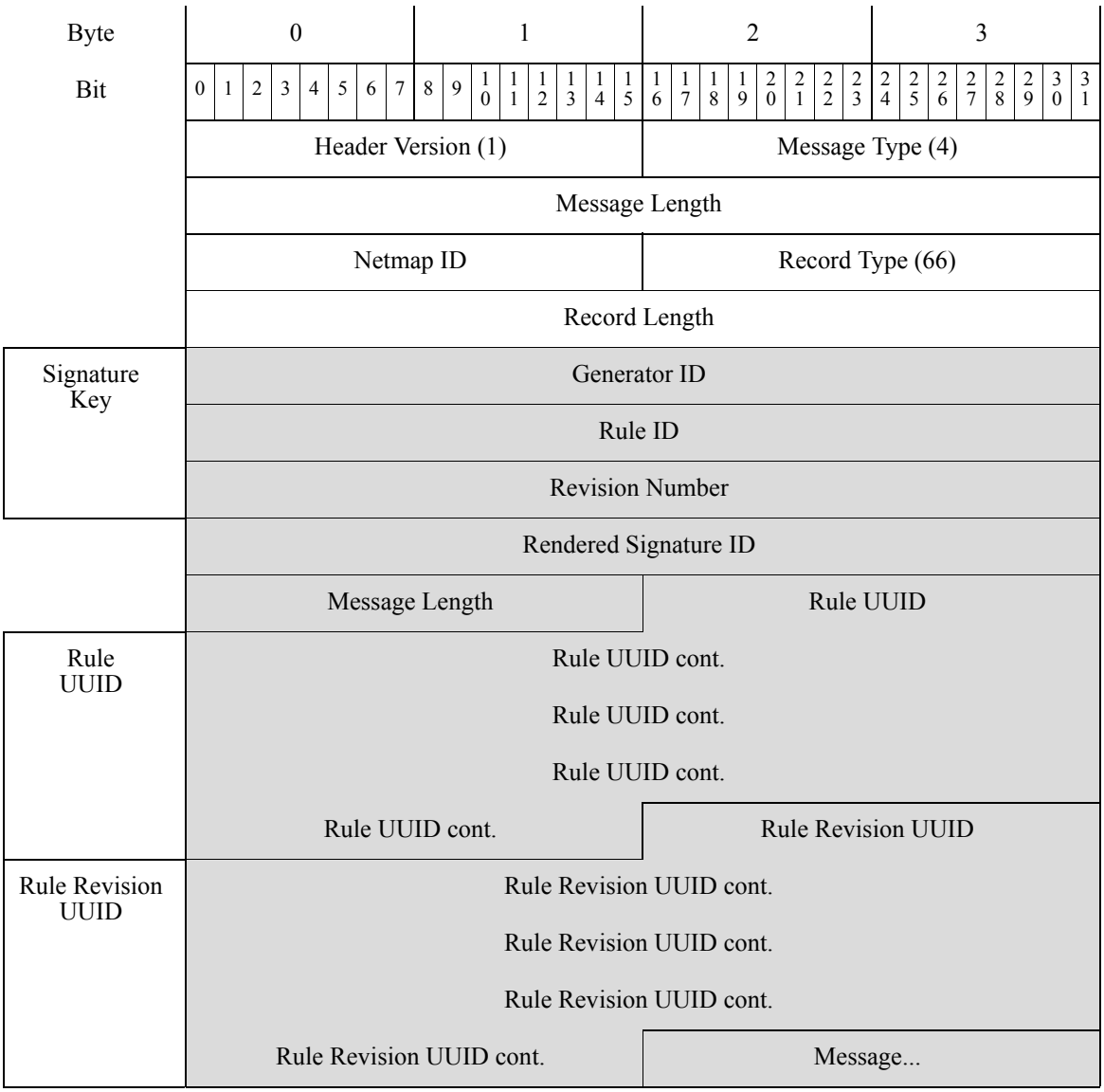
The following table describes the fields in the User record.

Table 3-6 User Record Fields

Field	Data Type	Description
Block Type	unint32	Initiates an User data block. This value is always 57.
Block Length	unint32	Total number of bytes in the User data block, including eight bytes for the User block type and length fields, plus the number of bytes of data that follows.
User ID	uint32	The user ID number.
Protocol	uint32	Protocol used to detect or report the user. Possible values are: • 165 - FTP • 426 - SIP • 547 - AOL Instant Messenger • 683 - IMAP • 710 - LDAP • 767 - NTP • 773 - Oracle Database • 788 - POP3 • 1755 - MDNS
String Block Type	uint32	Initiates a string data block that contains the name. This value is always set to 0. For more information about string blocks, see String Data Block, page 4-64.
Name Length	uint32	The number of bytes included in the user name.
Name	string	The name of the user.

Rule Message Record for 4.6.1+

Rule message information for an event is transmitted within a Rule Message record, the format of which is shown below. The eStreamer service transmits the Rule Message record for 4.6.1+ when you request Version 2 or Version 3 metadata. The Rule Message record for 4.6.1+ contains the same fields as the Rule Message record for 4.6 and lower but also has new UUID and Revision UUID fields. (Version 2, Version 3, or Version 4 metadata information is sent when the appropriate metadata flag—bit 14 for Version 2, bit 15 for Version 3, or bit 20 for Version 4 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 66, indicating a Rule Message Version 2 record.



The following table describes each rule-specific field.

Table 3-7 Rule Message Record Fields

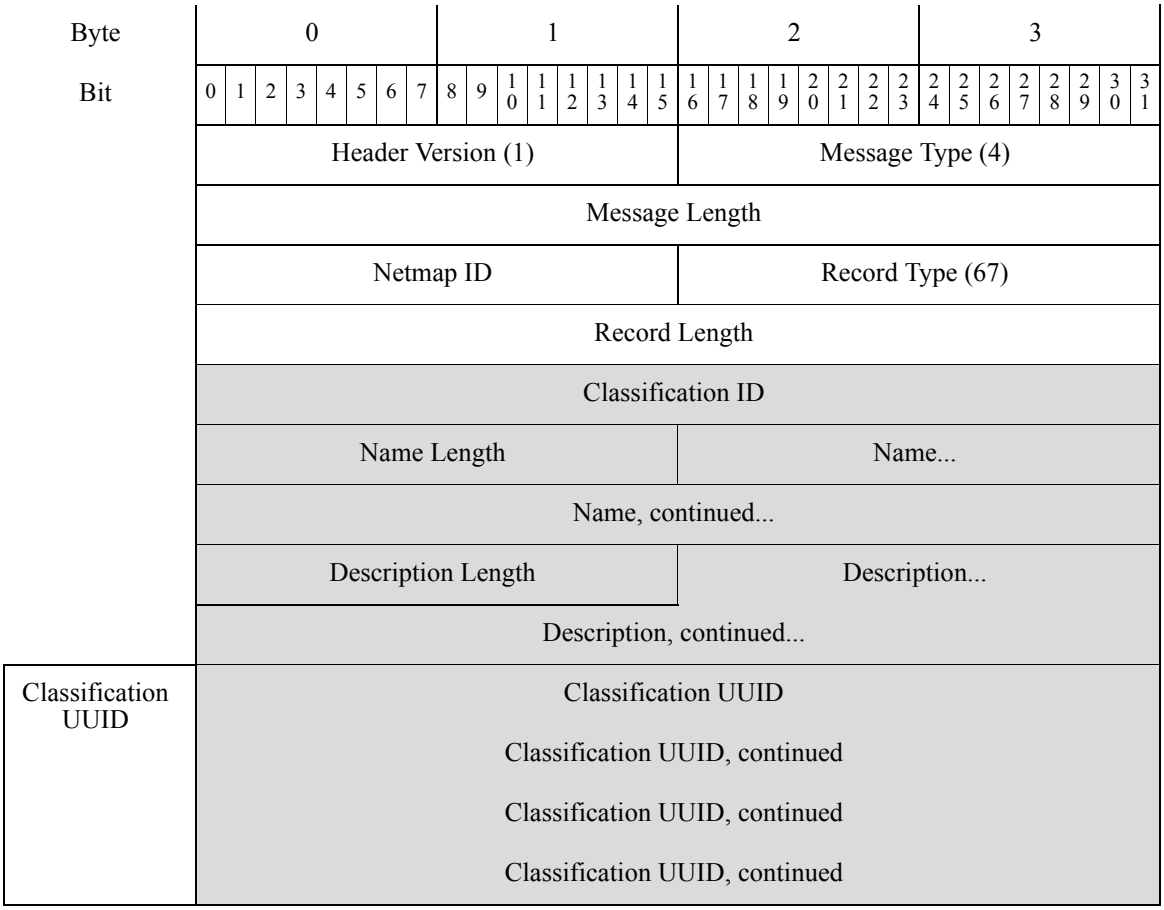
Field	Data Type	Description
Generator ID	uint32	The generator identification number.
Rule ID	uint32	The rule identification number for the local computer.
Rule Revision	uint32	The rule revision number. This is currently set to `0` for all rule messages.
Rendered Signature ID	uint32	The rule identification number rendered to the Firepower System interface.
Message Length	uint16	The number of bytes included in the rule text.

Table 3-7 Rule Message Record Fields (continued)

Field	Data Type	Description
UUID	uint8[16]	A rule ID number that acts as a unique identifier for the rule.
Revision UUID	uint8[16]	A rule revision ID number that acts as a unique identifier for the revision.
Message	variable	Rule message that triggered the event.

Classification Record for 4.6.1+

The eStreamer service transmits the classification information for an event in a Classification record for 4.6.1+, the format of which is shown below. The Classification record for 4.6.1+ contains the same fields as the Classification record for 4.6 and lower but also has new UUID and Revision UUID fields. (Classification information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 67, indicating a Classification Version 2 record.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Classification Revision UUID	Classification Revision UUID Classification Revision UUID, continued Classification Revision UUID, continued Classification Revision UUID, continued																															

The following table describes the fields in the Classification record.

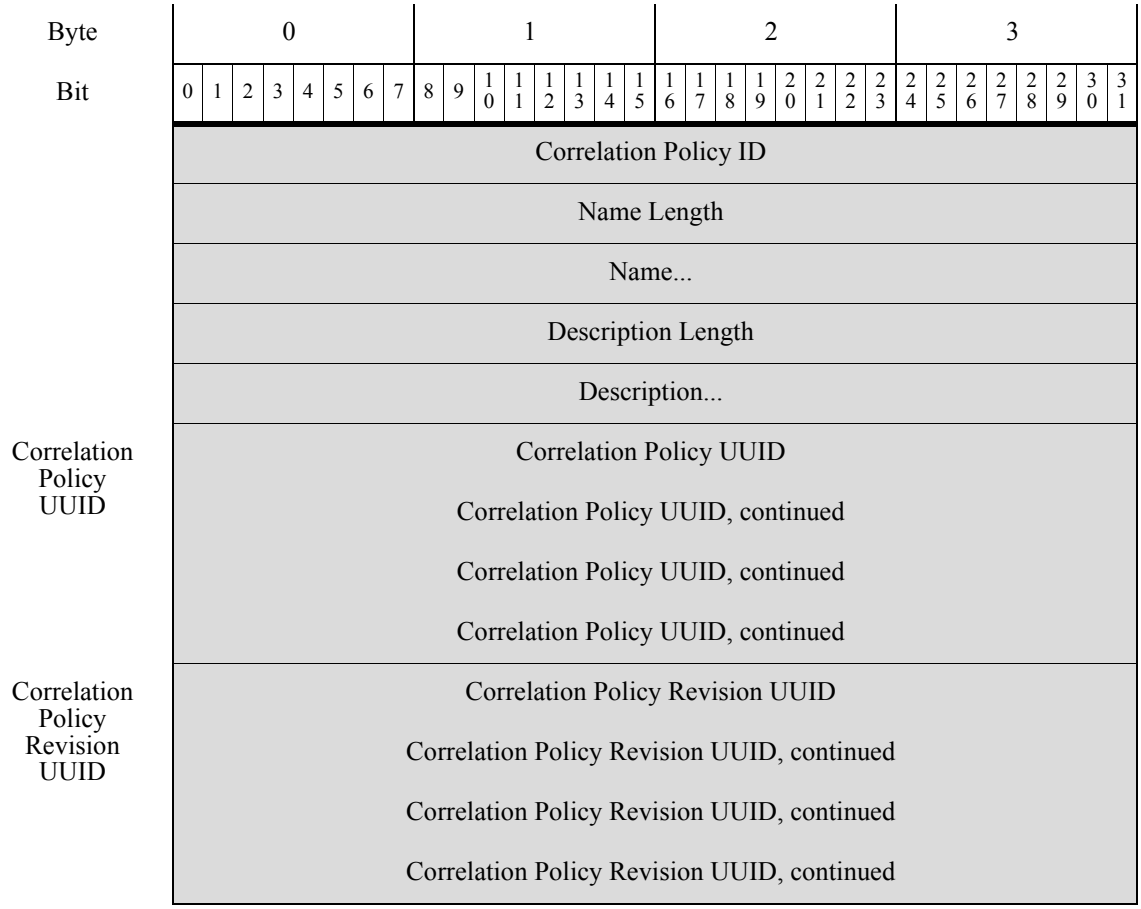
Table 3-8 Classification Record Fields

Field	Data Type	Description
Classification ID	uint32	The classification ID number.
Name Length	uint16	The number of bytes included in the name.
Name	string	The classification name.
Description Length	uint16	The number of bytes included in the description.
Description	string	The classification description.
UUID	uint8[16]	A classification ID number that acts as a unique identifier for the classification.
Revision UUID	uint8[16]	A classification revision ID number that acts as a unique identifier for the classification revision.

Correlation Policy Record

The eStreamer service transmits metadata containing the correlation policy for a correlation event within a Correlation Policy record, the format of which is shown below. (Correlation policy information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `69`, indicating a Correlation Policy record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (69)															
	Record Length																															



The following table describes the fields in the Correlation Policy record.

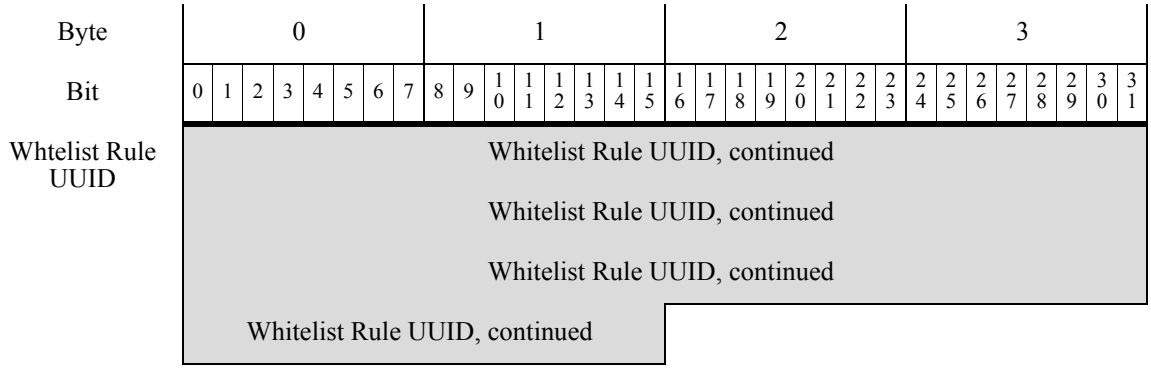
Table 3-9 Correlation Policy Record Fields

Field	Data Type	Description
Correlation Policy ID	uint32	The correlation policy ID number.
Name Length	uint16	The number of bytes included in the correlation policy name.
Name	string	The name of the correlation policy that triggered the event.
Description Length	uint16	The number of bytes included in the correlation policy description.
Description	string	The description of the correlation policy that triggered the event.
UUID	uint8[16]	A correlation policy ID number that acts as a unique identifier for the correlation policy.
Revision UUID	uint8[16]	A correlation policy revision ID number that acts as a unique identifier for the correlation policy.

Correlation Rule Record

The eStreamer service transmits metadata containing information on the correlation rule that triggered a correlation event within a Correlation Rule record, the format of which is shown below. (Correlation rule information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `70`, indicating a Correlation Rule record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (70)															
	Record Length																															
	Correlation Rule ID																															
	Name Length																Name...															
	Name...																Description Length															
	Description...																															
	Event Type Length																Event Type...															
	Event Type...																Correlation Rule UUID															
Correlation Rule UUID	Correlation Rule UUID, continued																															
	Correlation Rule UUID, continued																															
	Correlation Rule UUID, continued																															
	Correlation Rule UUID, continued																Correlation Revision UUID,															
Correlation Rule Revision UUID	Correlation Rule Revision UUID, continued																															
	Correlation Rule Revision UUID, continued																															
	Correlation Rule Revision UUID, continued																															
	Correlation Rule Revision UUID, continued.																Whitelist Rule UUID															



The following table describes the fields in the Correlation Rule record.

Table 3-10 Correlation Rule Record Fields

Field	Data Type	Description
Correlation Rule ID	uint32	The correlation rule ID number.
Name Length	uint16	The number of bytes included in the correlation rule name.
Name	string	The name of the correlation rule that triggered the event.
Description Length	uint16	The number of bytes included in the correlation rule description.
Description	string	The description of the correlation rule that triggered the event.
Event Type Length	uint16	The number of bytes included in the event type description.
Event Type	string	The description of the event that triggered the correlation rule.
UUID	uint8[16]	A correlation rule ID number that acts as a unique identifier for the correlation rule.
Revision UUID	uint8[16]	A correlation rule revision ID number that acts as a unique identifier for the correlation rule revision.
Whitelist UUID	uint8[16]	A correlation ID number that acts as a unique identifier for the event sent as a result of a whitelist violation.

Intrusion Event Extra Data Record

The eStreamer service transmits the event extra data associated with an intrusion event in the Intrusion Event Extra Data record. The record type is always `110`.

The event extra data appears in an encapsulated Event Extra Data data block, which always has a data block type value of `4`. (The Event Extra Data data block is a series 2 data block. For more information about series 2 data blocks, see Understanding Series 2 Data Blocks, page 3-52.)

The supported types of extra data include IPv6 source and destination addresses, as well as the originating IP addresses (v4 or v6) of clients connecting to a web server through an HTTP proxy or load balancer. The graphic below shows the format of the Intrusion Event Extra Data record.

If bit 27 is set in the Request Flags field of the request message, you receive the event extra data for each intrusion event. If you set bit 20, you also receive the event extra data metadata described in Intrusion Event Extra Data Metadata, page 3-28. If you enable bit 23, eStreamer will include the extended event header. See Request Flags, page 2-11 for information on setting request flags.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (110)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Event Extra Data Data Block Type (4)																																
Event Extra Data Data Block Length																																
Device ID																																
Event ID																																
Event Second																																
Type																																
BLOB Block Type (1)																																
BLOB Length																																
Event Extra Data																																

Note that the Event Extra Data block structure includes a BLOB block type, which is one of several variable length data structures introduced in Version 4.10 of the Firepower System.

The following table describes the fields in the Intrusion Event Extra Data record.

Table 3-11 Intrusion Event Extra Data Data Block Fields

Field	Data Type	Description
Event Extra Data Data Block Type	uint32	Initiates an Event Extra Data data block. This value is always `4`. The block type is a series 2 block; for information see Understanding Series 2 Data Blocks, page 3-52.
Event Extra Data Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Device ID	uint32	The managed device identification number.
Event ID	uint32	The event identification number.
Event Second	uint32	UNIX timestamp of the event (seconds since 01/01/1970).

Table 3-11 Intrusion Event Extra Data Data Block Fields (continued)

Field	Data Type	Description
Type	uint32	Identifier for the type of extra data; for example: • 1 — XFF client (IPv4) • 2 — XFF client (IPv6) • 9 — HTTP URI
BLOB Block Type	uint32	Initiates a BLOB data block containing extra data. This value is always 1. The block type is a series 2 block.
Length	uint32	Total number of bytes in the BLOB data block.
Extra Data	variable	The content of the extra data. The data type is indicated in the Type field.

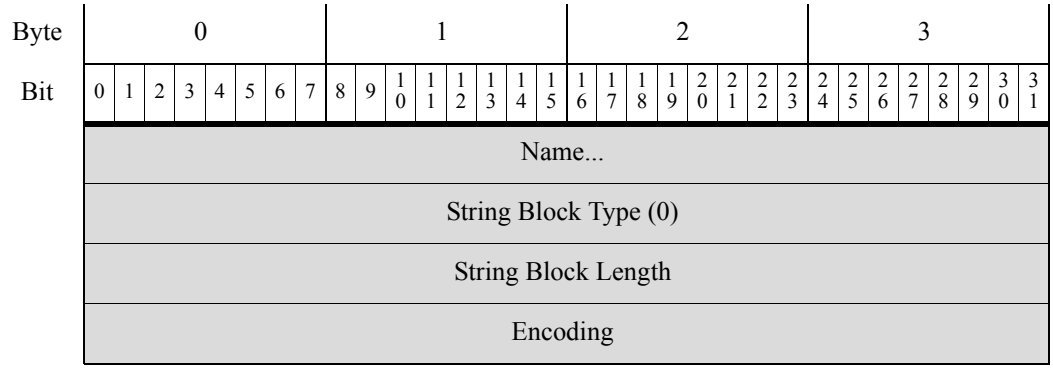
Intrusion Event Extra Data Metadata

The eStreamer service transmits the event extra data metadata associated with intrusion event extra data records in the Intrusion Event Extra Data Metadata record. The record type is always 111.

The event extra data metadata appears in an encapsulated Event Extra Data Metadata data block, which always has a data block type value of 5. The Event Extra Data data block is a series 2 data block.

If bit 20 is set in the Request Flags field of a request message, you receive the event extra data metadata. If you want to receive both intrusion events and event extra data metadata, you must set bit 2 as well. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (111)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Event Extra Data Metadata Data Block Type (5)																																
Data Block Length																																
Type																																
String Block Type (0)																																
String Block Length																																



Note that the block structure includes encapsulated String block types, one of several series 2 variable length data structures introduced in Version 4.10 of the Firepower System.

The following table describes the fields in the Event Extra Data Metadata record.

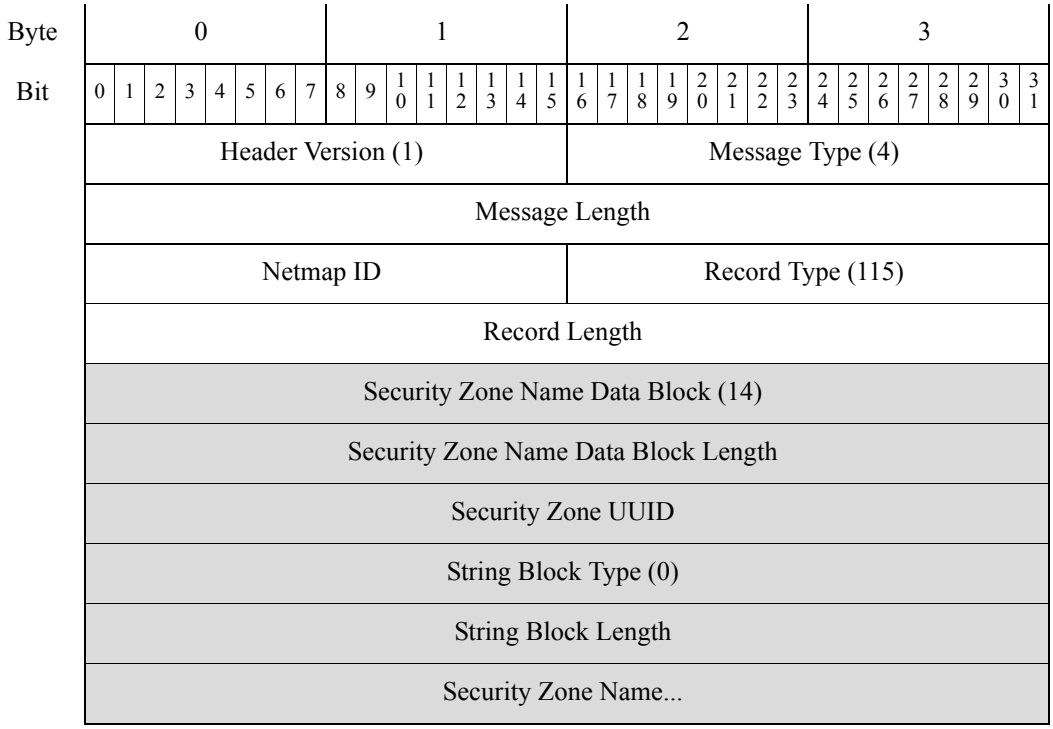
Table 3-12 Event Extra Data Metadata Data Block Fields

Field	Data Type	Description
Event Extra Data Metadata Data Block Type	uint32	Initiates an Event Extra Data Metadata data block. This value is always `5`. This block type is a series 2 block.
Event Extra Data Metadata Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Type	uint32	The type of extra data. Matches the Type field in the associated Event Extra Data record.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`. This block type is a series 2 block.
String Block Length	uint32	Number of bytes in the client application version String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the version string.
Name	string	Name of the type of event extra data, for example, XFF client (IPv6), and HTTP URI.
String Block Type	uint32	Initiates a string data block for the client application URL. This value is always `0`. This block type is a series 2 block.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the URL string.
Encoding	string	Encoding used for the event extra data, for example, IPv4, IPv6, or string.

Security Zone Name Record

The eStreamer service transmits metadata containing information on the name of the security zone associated with an intrusion event or connection event within a Security Zone Name record, the format of which is shown below. (Security zone information is sent when the Version 4 metadata flag—bit 20

in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `115`, indicating a Security Zone Name record. It contains a UUID String data block, block type 14 in the series 2 set of data blocks.



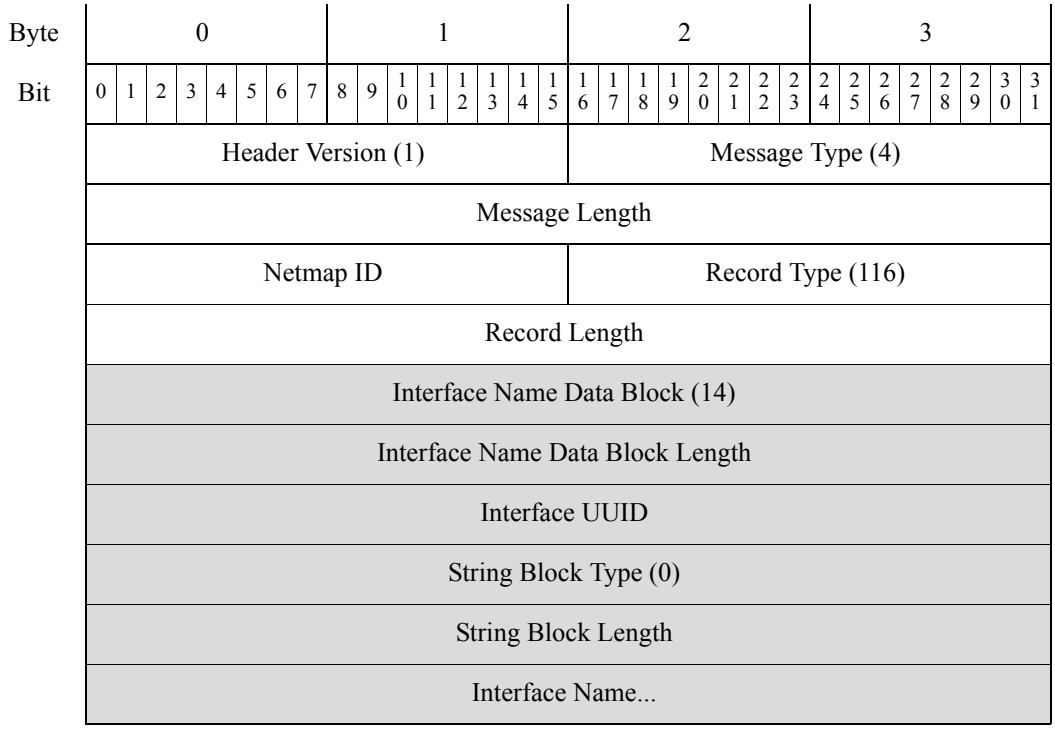
The following table describes the fields in the Security Zone Name data block.

Table 3-13 Security Zone Name Data Block Fields

Field	Data Type	Description
Security Zone Name Data Block Type	uint32	Initiates a Security Zone Name data block. This value is always `14`. The block type is a series 2 block.
Security Zone Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Security Zone UUID	uint8[16]	The unique identifier for the security zone associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the security zone. This value is always `0`.
String Block Length	uint32	The number of bytes included in the security zone name String data block, including eight bytes for the block type and header fields plus the number of bytes in the name.
Security Zone Name	string	The security zone name.

Interface Name Record

The eStreamer service transmits metadata containing information on the name of the interface associated with an intrusion event or connection event within an Interface Name record, the format of which is shown below. (Interface name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `116`, indicating an Interface Name record. It contains a UUID String data block, block type 14 in the series 2 set of data blocks.



The following table describes the fields in the Interface Name data block.

Table 3-14 Interface Name Data Block Fields

Field	Data Type	Description
Interface Name Data Block Type	uint32	Initiates an Interface Name data block. This value is always `14`. The block type is a series 2 block.
Interface Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the interface associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the interface. This value is always `0`.
String Block Length	uint32	The number of bytes included in the interface name String data block, including eight bytes for the block type and header fields plus the number of bytes in the interface name.
Interface Name	string	The interface name.

Access Control Policy Name Record

The eStreamer service transmits metadata on the name of the access control policy that triggered an intrusion event or connection event within an Access Control Policy Name record, the format of which is shown below. (Access control policy name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `117`, indicating an Access Control Policy Name record. It contains a UUID String data block, block type 14 in the series 2 set of data blocks.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (117)															
Record Length																																
Access Control Policy Name Data Block (14)																																
Access Control Policy Name Data Block Length																																
Access Control Policy UUID																																
String Block Type (0)																																
String Block Length																																
Access Control Policy Name...																																

The following table describes the fields in the Access Control Policy Name data block.

Table 3-15 Access Control Policy Name Data Block Fields

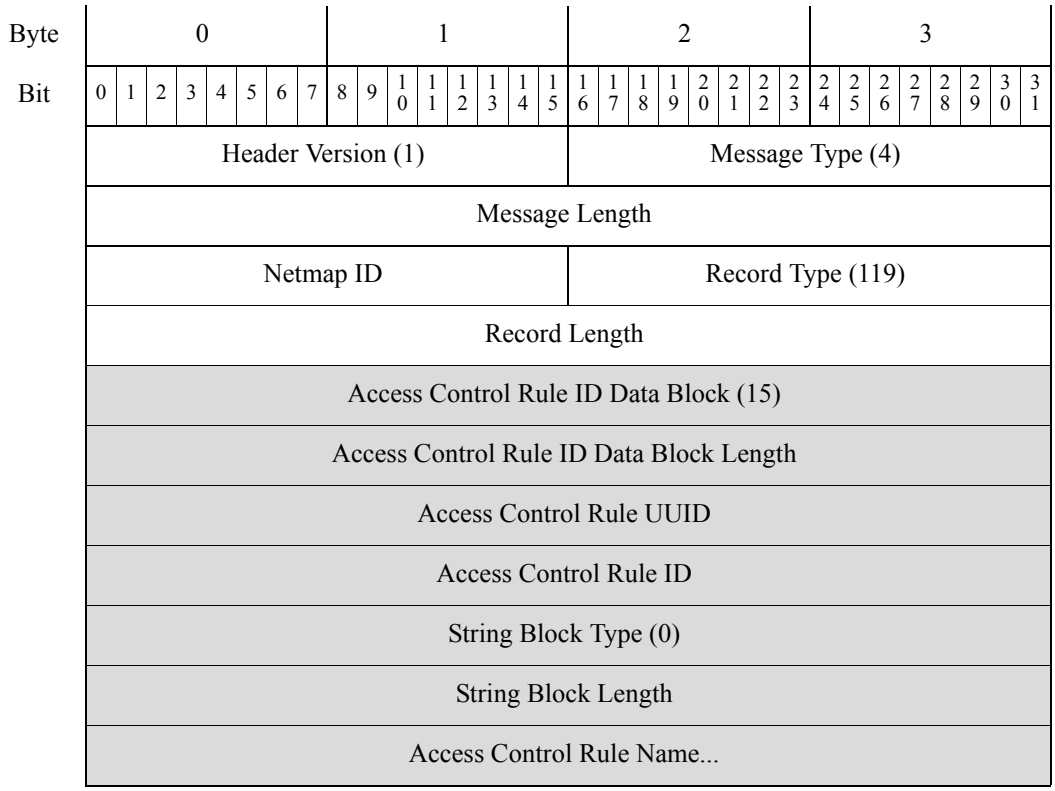
Field	Data Type	Description
Access Control Policy Name Data Block Type	uint32	Initiates an Access Control Policy Name data block. This value is always `14`. The block type is a series 2 block.
Access Control Policy Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Access Control Policy UUID	uint8[16]	An ID number that acts as a unique identifier for the access control policy associated with the intrusion event or connection event
String Block Type	uint32	Initiates a String data block containing the name of the access control policy. This value is always `0`.

Table 3-15 Access Control Policy Name Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the access control policy name String data block, including eight bytes for the block type and header fields plus the number of bytes in the access control policy name.
Access Control Policy Name	string	The access control policy name.

Access Control Rule ID Record Metadata

The eStreamer service transmits metadata containing information about the access control rule that triggered an intrusion event or connection event within an Access Control Rule ID record, the format of which is shown below. Access control rule metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `119`, indicating an Access Control Rule ID record. It contains a Rule ID data block, block type 15 in the series 2 set of data blocks.



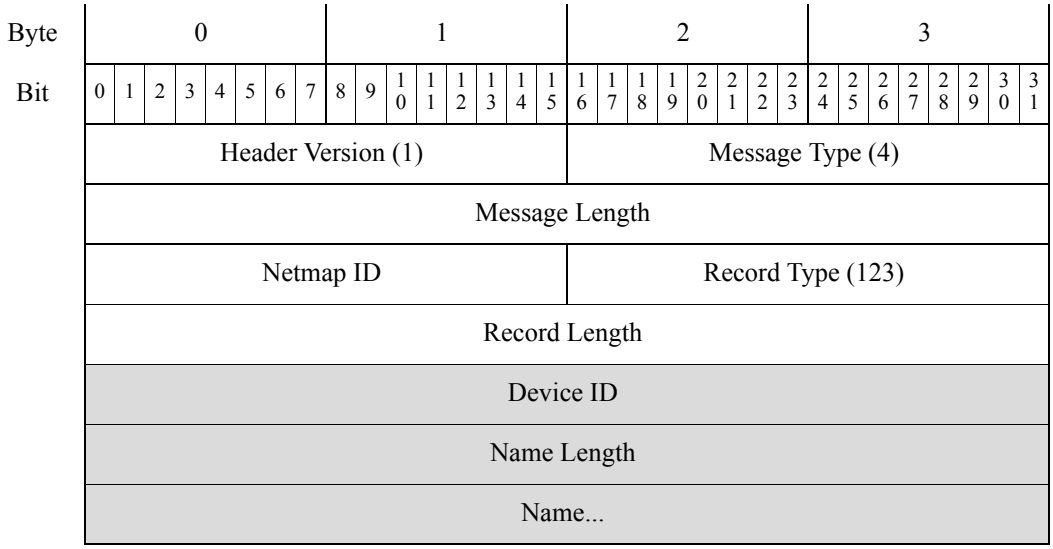
The following table describes the fields in the Access Control Rule ID data block.

Table 3-16 Access Control Rule ID Data Block Fields

Field	Data Type	Description
Access Control Rule ID Data Block Type	uint32	Initiates an Access Control Rule ID data block. This value is always `15`. The block type is a series 2 block.
Access Control Rule ID Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Access Control Rule UUID	uint8[16]	A rule ID that acts as the unique identifier for the rule in the access control policy associated with the connection event.
Access Control Rule ID	uint32	The internal identifier for the rule in the access control policy associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the access control rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the String data block, including eight bytes for the block type and header fields plus the number of bytes in the rule name.
Access Control Rule Name	string	The access control rule name.

Managed Device Record Metadata

The eStreamer service transmits metadata containing information on the managed device associated with an intrusion event within a Managed Device record, the format of which is shown below. Managed device metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `123`, indicating a Managed Device record.



The following table describes the fields in the Managed Device record.

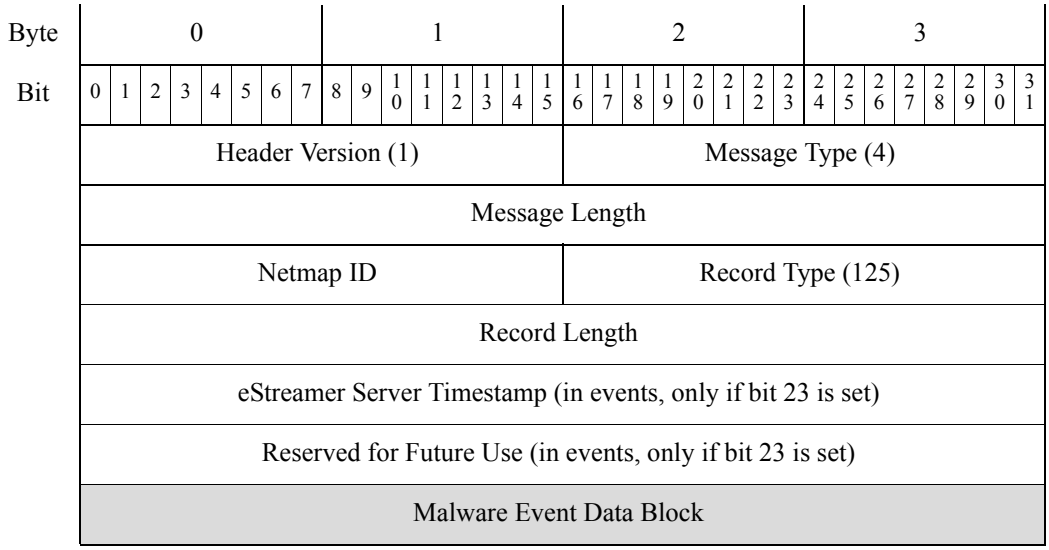
Table 3-17 Managed Device Record Fields

Field	Data Type	Description
Device ID	uint32	ID number of the managed device.
Name Length	uint32	The number of bytes included in the name.
Name	string	The managed device name.

Malware Event Record 5.1.1+

The fields in the malware event record are shaded in the following graphic. The record type is 125.

You request malware event records by setting the malware event flag—bit 30 in the Request Flags field—in the request message with an event version of `2` and an event code of `101`. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record. It contains a Malware Event data block, one of block types 24, 33, 35, 44, 47, or in the series 2 set of data blocks.



The following table describes each malware event record data field.

Table 3-18 Malware Event Record Fields

Field	Data Type	Description
Malware Event Data Block	variable	Indicates a malware event data block. See Malware Event Data Block 6.0+, page 3-83 for more information.

Cisco Advanced Malware Protection Cloud Name Metadata

The eStreamer service transmits metadata containing information on the name of the Cisco Advanced Malware Protection cloud (referred to as the AMP cloud or simply cloud) associated with an intrusion event or connection event within a Cisco Advanced Malware Protection cloud Name record, the format

of which is shown below. (AMP cloud name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `127`, indicating a Cisco Advanced Malware Protection cloud Name record. It contains a UUID String data block, block type 14 in the series 2 set of data blocks.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (127)															
Record Length																																
Cisco Advanced Malware Protection cloud Name Data Block (14)																																
Cisco Advanced Malware Protection cloud Name Data Block Length																																
Cisco Advanced Malware Protection cloud UUID																																
Cisco Advanced Malware Protection cloud UUID, cont.																																
Cisco Advanced Malware Protection cloud UUID, cont.																																
Cisco Advanced Malware Protection cloud UUID, cont.																																
String Block Type (0)																																
String Block Length																																
Cisco Advanced Malware Protection cloud Name...																																

The following table describes the fields in the Cisco Advanced Malware Protection cloud Name data block.

Table 3-19 Cisco Advanced Malware Protection cloud Name Data Block Fields

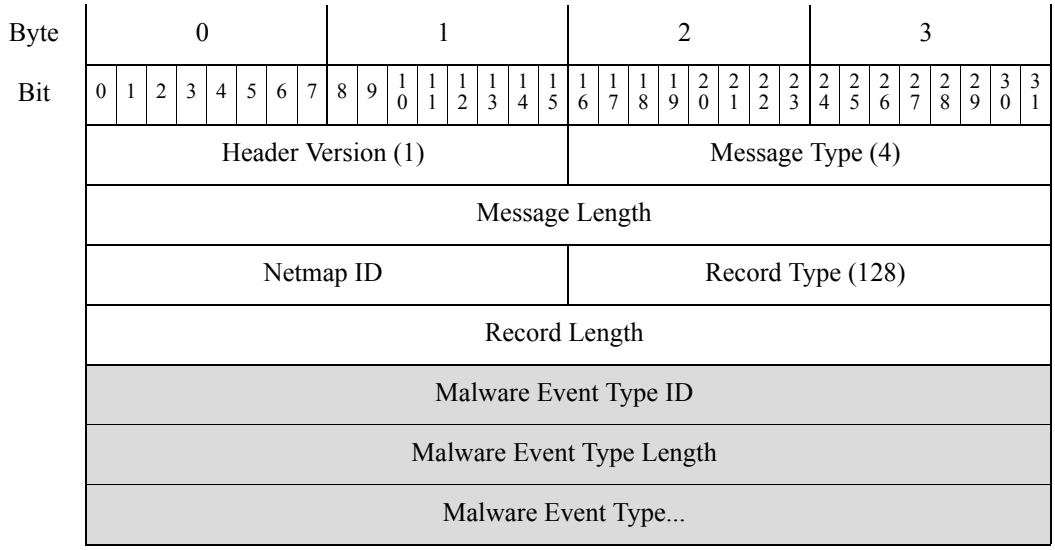
Field	Data Type	Description
Cisco Advanced Malware Protection cloud Name Data Block Type	uint32	Initiates a Cisco Advanced Malware Protection cloud Name data block. This value is always `14`. The block type is a series 2 block.
Cisco Advanced Malware Protection cloud Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.

Table 3-19 Cisco Advanced Malware Protection cloud Name Data Block Fields (continued)

Field	Data Type	Description
Cisco Advanced Malware Protection cloud UUID	uint8[16]	A Cisco Advanced Malware Protection cloud ID number that acts as a unique identifier for the Cisco Advanced Malware Protection cloud associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the Cisco Advanced Malware Protection cloud. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Cisco Advanced Malware Protection cloud Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Cisco Advanced Malware Protection cloud name.
Cisco Advanced Malware Protection cloud Name	string	The Cisco Advanced Malware Protection cloud name.

Malware Event Type Metadata

The eStreamer service transmits metadata containing malware event type information for an event within a malware event type record, the format of which is shown below. (Malware event type information is sent when the metadata flag, bit 20 in the request flags field of a request message, is set. See Request Flags, page 2-11.) Note that the record type field, which appears after the message length field, has a value of `128`, indicating a malware event type record.



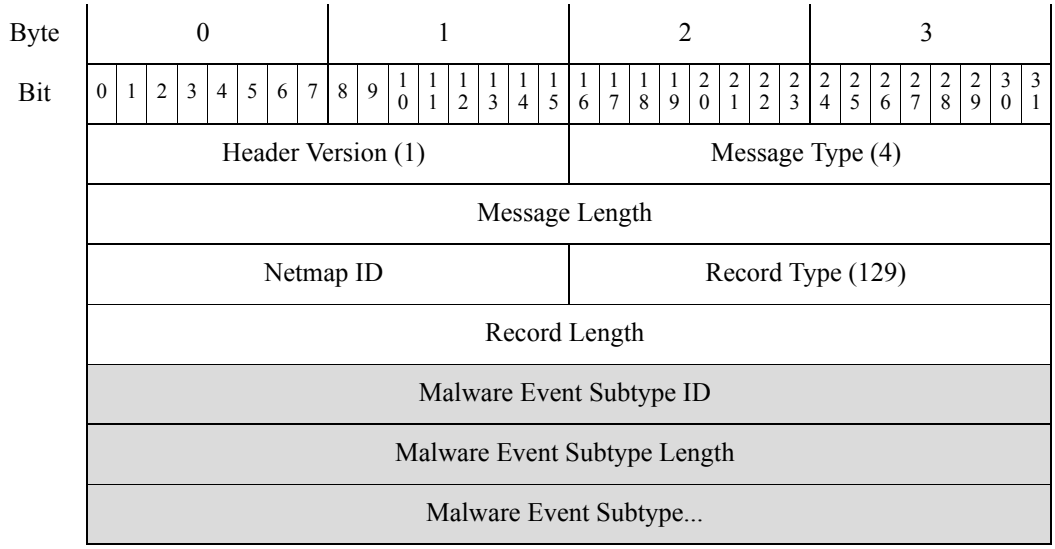
The following table describes the fields in the malware event type record.

Table 3-20 Malware Event Type Record Fields

Field	Data Type	Description
Malware Event Type ID	uint32	The malware event type ID number.
Malware Event Type Length	uint32	The number of bytes included in the malware event type.
Malware Event Type	string	The type of malware event.

Malware Event Subtype Metadata

The eStreamer service transmits metadata containing malware event subtype information for an event within a malware event subtype record, the format of which is shown below. (Malware event type information is sent when the metadata flag, bit 20 in the request flags field of a request message, is set. See Request Flags, page 2-11.) Note that the record type field, which appears after the message length field, has a value of `129`, indicating a malware event subtype record.



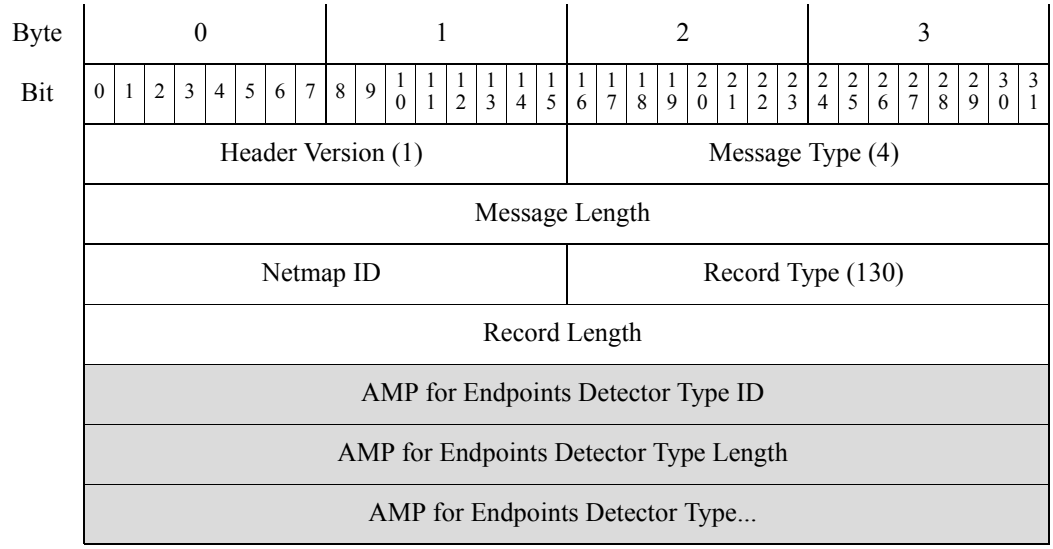
The following table describes the fields in the malware event subtype record.

Table 3-21 Malware Event Subtype Record Fields

Field	Data Type	Description
Malware Event Subtype ID	uint32	The malware event subtype ID number.
Malware Event Subtype Length	uint32	The number of bytes included in the malware event subtype.
Malware Event Subtype	string	The malware event subtype.

AMP for Endpoints Detector Type Metadata

The eStreamer service transmits metadata containing AMP for Endpoints detector type information for an event within a AMP for Endpoints Detector Type record, the format of which is shown below. (AMP for Endpoints detector type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `130`, indicating a AMP for Endpoints detector type record.



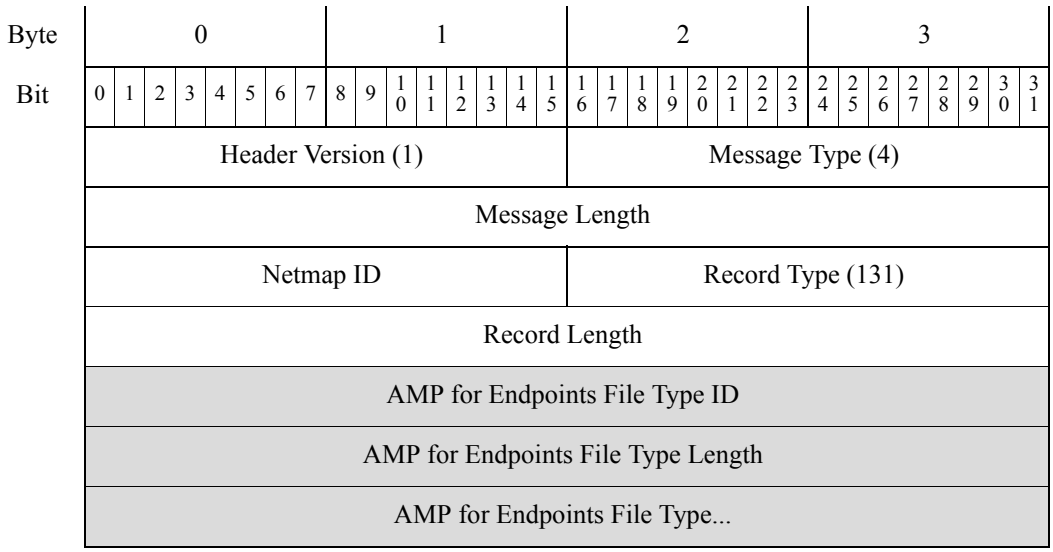
The following table describes the fields in the AMP for Endpoints Detector Type record.

Table 3-22 AMP for Endpoints Detector Type Record Fields

Field	Data Type	Description
AMP for Endpoints Detector Type ID	uint32	The AMP for Endpoints detector type ID number.
AMP for Endpoints Detector Type Length	uint32	The number of bytes included in the AMP for Endpoints detector type.
AMP for Endpoints Detector Type	string	The type of AMP for Endpoints detector.

AMP for Endpoints File Type Metadata

The eStreamer service transmits metadata containing AMP for Endpoints file type information for an event within a AMP for Endpoints File Type record, the format of which is shown below. (AMP for Endpoints file type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `131`, indicating a AMP for Endpoints file type record.



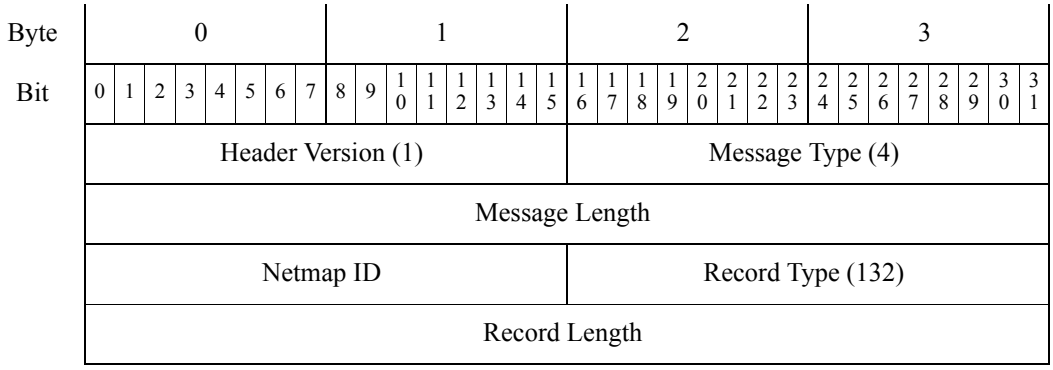
The following table describes the fields in the AMP for Endpoints File Type record.

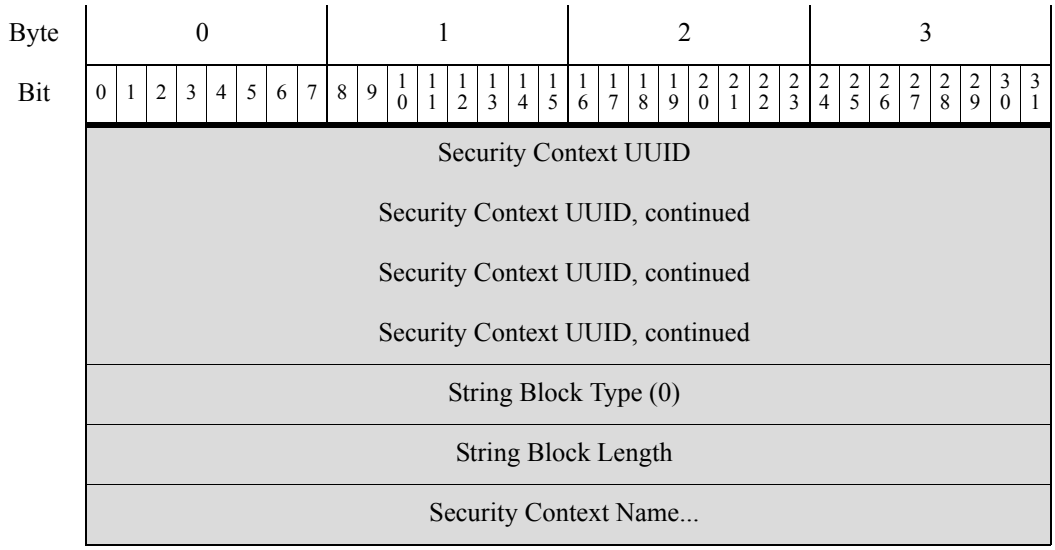
Table 3-23 AMP for Endpoints File Type Record Fields

Field	Data Type	Description
AMP for Endpoints File Type ID	uint32	The AMP for Endpoints file type ID number.
AMP for Endpoints File Type Length	uint32	The number of bytes included in the AMP for Endpoints file type.
AMP for Endpoints File Type	string	The type of detected file.

Security Context Name

The eStreamer service transmits metadata containing Security Context Name information, the format of which is shown below. (Security Context Name information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `132`, indicating a Security Context Name record.





The following table describes the fields in the Security Context Name record.

Table 3-24 Security Context Name Record Fields

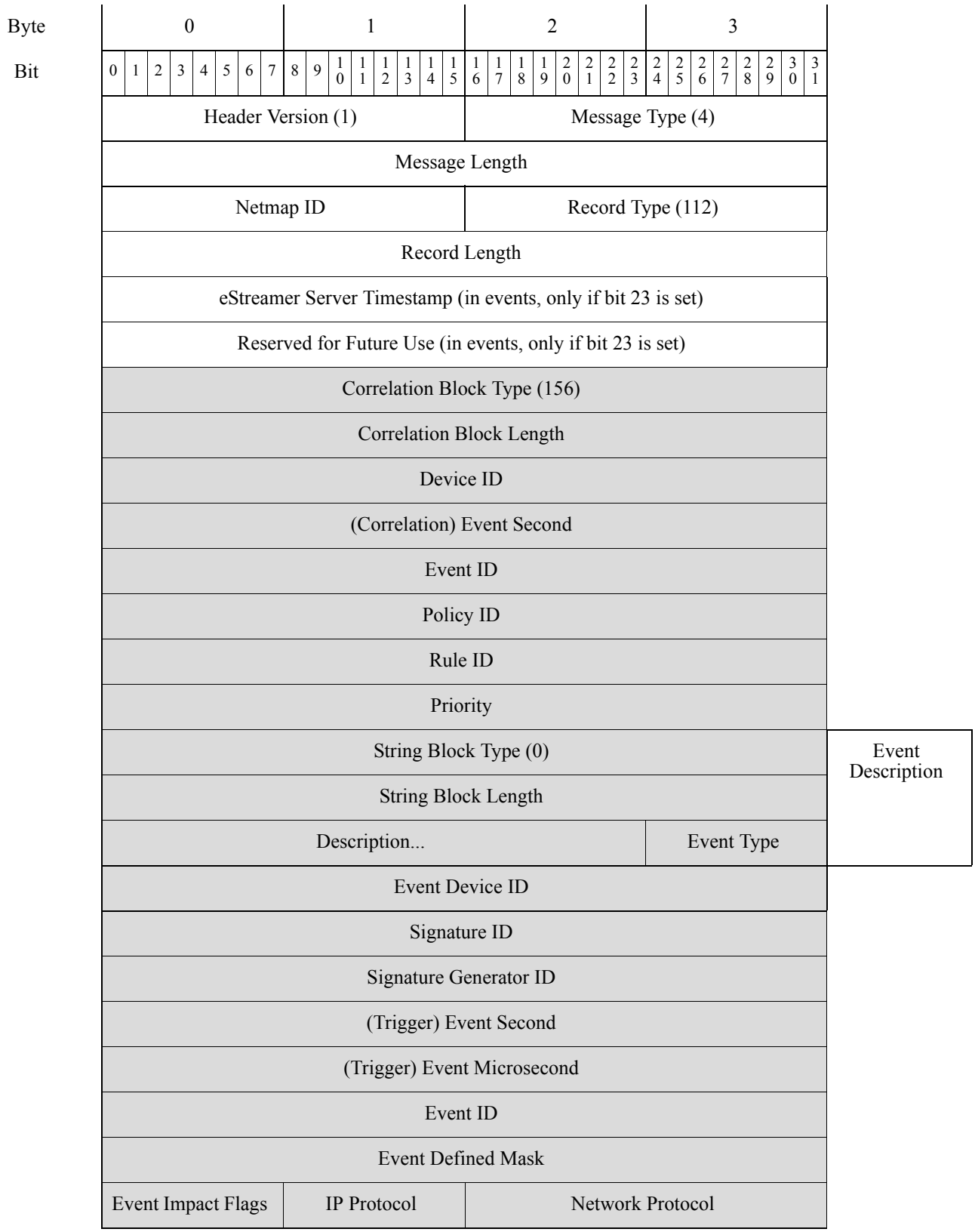
Field	Data Type	Description
Security Context UUID	uint8[16]	The UUID of the security context
String Block Type	uint32	Initiates a String data block containing the name of the security context. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Security Context Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Security Context name.
Security Context Name	string	The security context name.

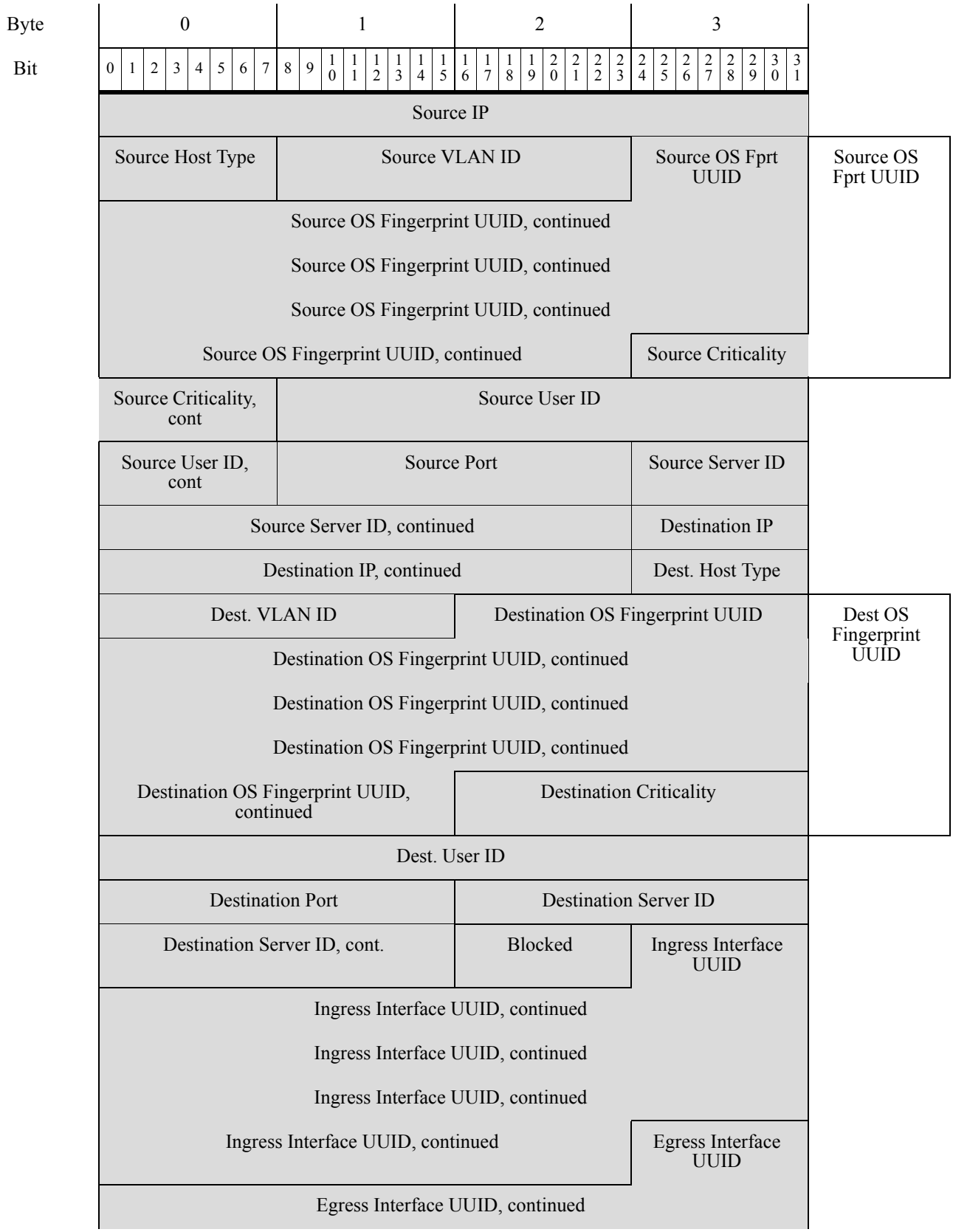
Correlation Event for 5.4+

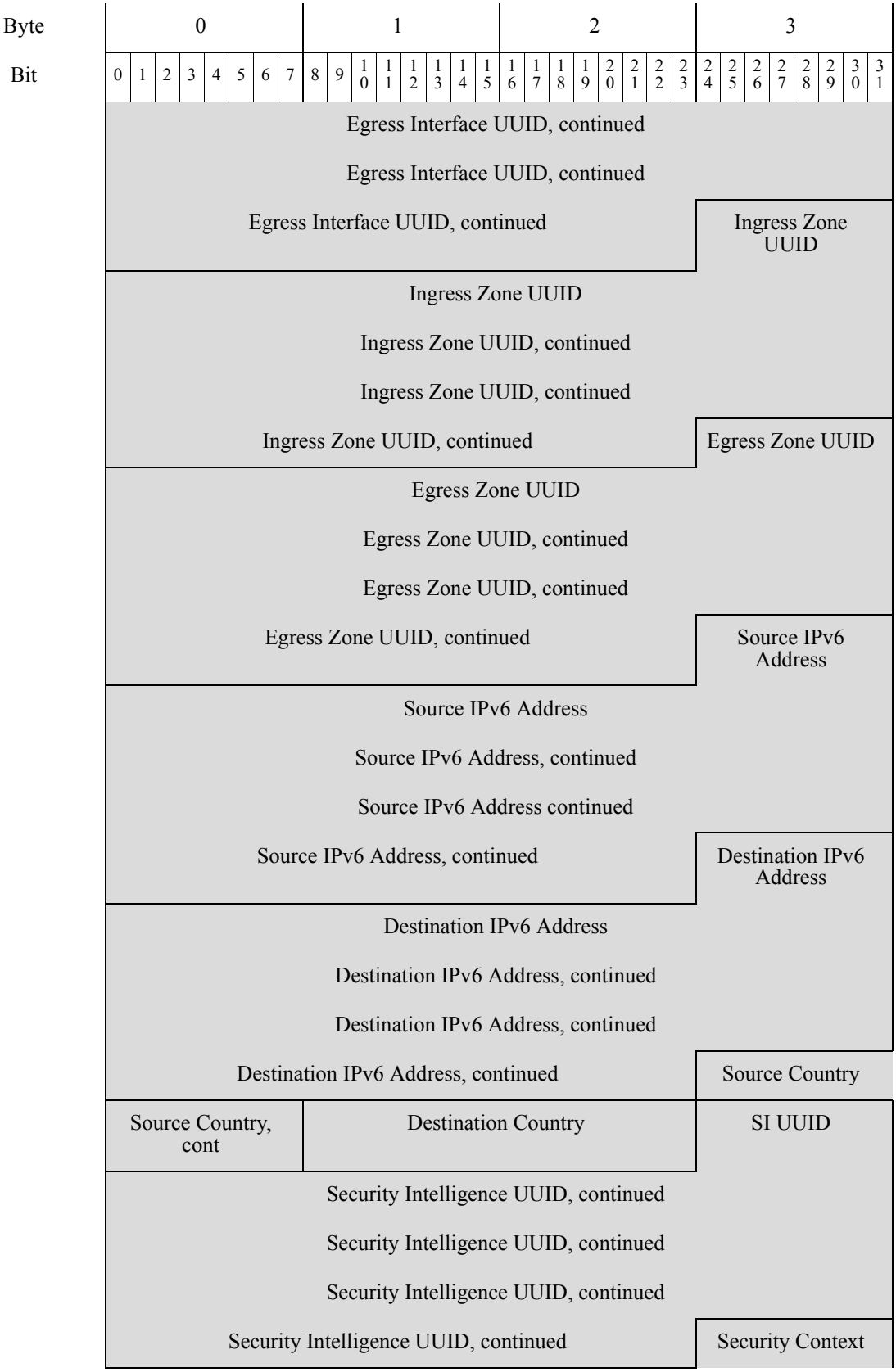
Correlation events (called compliance events in pre-5.0 versions) contain information about correlation policy violations. This message uses the standard eStreamer message header and specifies a record type of 112, followed by a correlation data block of type 156 in the series 1 set of data blocks. Data block type 156 differs from its predecessor (block type 128) in including IPv6 support.

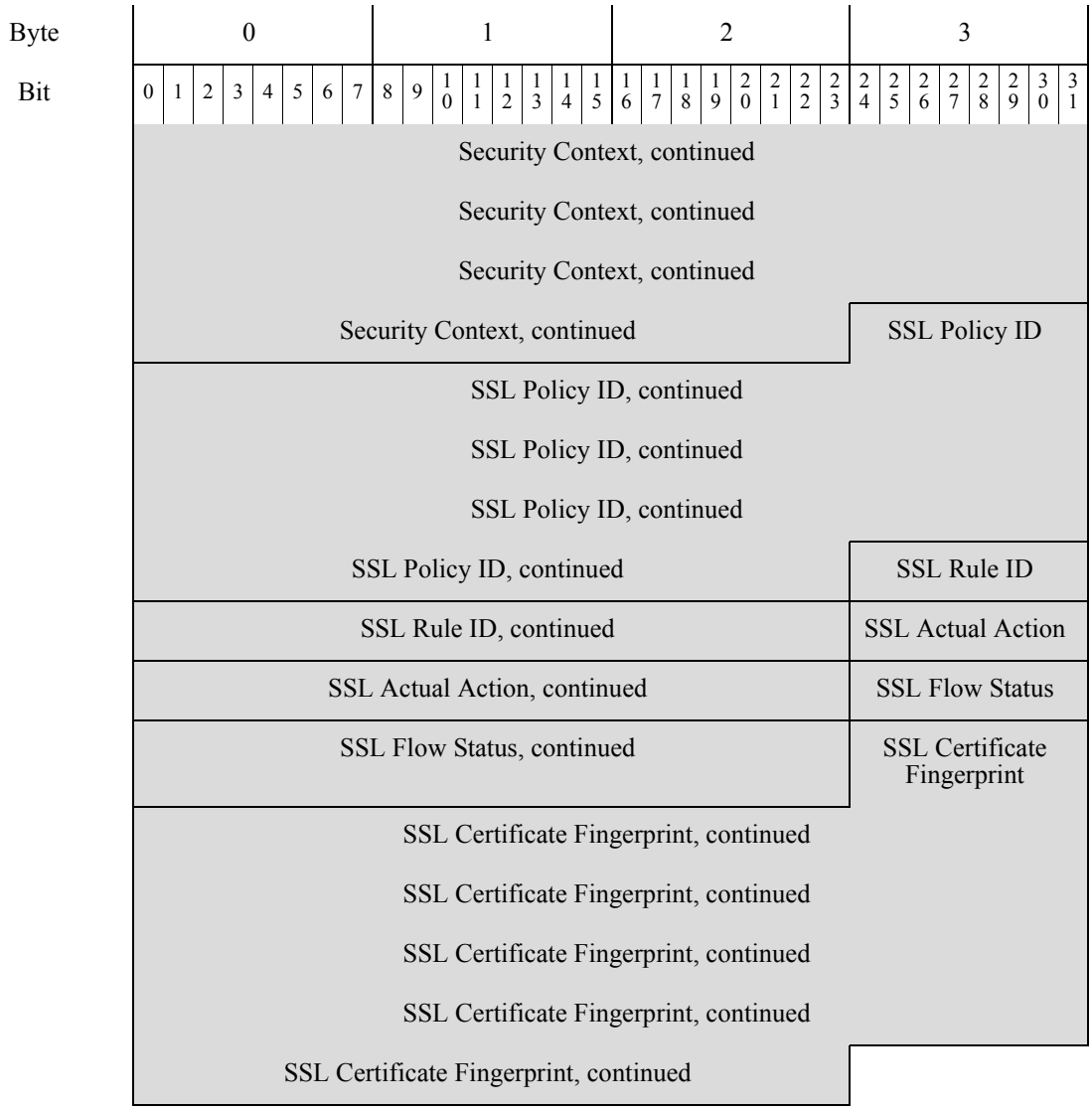
The 5.4+ version of correlation events has new fields for geolocation, Security Intelligence, and SSL support.

You can request 5.4+ correlation events from eStreamer only by extended request, for which you request event type code 31 and version code 9 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests). You can optionally enable bit 23 in the flags field of the initial event stream request message, to include the extended event header. You can also enable bit 20 in the flags field to include user metadata.


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Header Version (1)
Message Type (4)
Message Length
Netmap ID
Record Type (112)
Record Length
eStreamer Server Timestamp (in events, only if bit 23 is set)
Reserved for Future Use (in events, only if bit 23 is set)
Correlation Block Type (156)
Correlation Block Length
Device ID
(Correlation) Event Second
Event ID
Policy ID
Rule ID
Priority
String Block Type (0)
String Block Length
Description...
Event Type
Event Description
Event Device ID
Signature ID
Signature Generator ID
(Trigger) Event Second
(Trigger) Event Microsecond
Event ID
Event Defined Mask
Event Impact Flags
IP Protocol
Network Protocol


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Source IP
Source Host Type
Source VLAN ID
Source OS Fprt UUID
Source OS Fprt UUID
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source Criticality
Source Criticality, cont
Source User ID
Source User ID, cont
Source Port
Source Server ID
Source Server ID, continued
Destination IP
Destination IP, continued
Dest. Host Type
Dest. VLAN ID
Destination OS Fingerprint UUID
Dest OS Fingerprint UUID
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination Criticality
Dest. User ID
Destination Port
Destination Server ID
Destination Server ID, cont.
Blocked
Ingress Interface UUID
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Egress Interface UUID
Egress Interface UUID, continued


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Egress Interface UUID, continued
Egress Interface UUID, continued
Egress Interface UUID, continued
Ingress Zone UUID
Ingress Zone UUID
Ingress Zone UUID, continued
Ingress Zone UUID, continued
Ingress Zone UUID, continued
Egress Zone UUID
Egress Zone UUID
Egress Zone UUID, continued
Egress Zone UUID, continued
Egress Zone UUID, continued
Source IPv6 Address
Source IPv6 Address
Source IPv6 Address, continued
Source IPv6 Address continued
Source IPv6 Address, continued
Destination IPv6 Address
Destination IPv6 Address
Destination IPv6 Address, continued
Destination IPv6 Address, continued
Destination IPv6 Address, continued
Source Country
Source Country, cont
Destination Country
SI UUID
Security Intelligence UUID, continued
Security Intelligence UUID, continued
Security Intelligence UUID, continued
Security Intelligence UUID, continued
Security Context



Note that the record structure includes a String block type, which is a block in series 1. For information about series 1 blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.

Table 3-25 Correlation Event 5.4+ Data Fields

Field	Data Type	Description
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 156. See Understanding Discovery (Series 1) Blocks, page 4-56.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.

Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
Device ID	uint32	Internal identification number of the managed device or Management Center that generated the correlation event. A value of zero indicates the Management Center. You can obtain managed device names by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
(Correlation) Event Second	uint32	UNIX timestamp indicating the time that the correlation event was generated (in seconds from 01/01/1970).
Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to 0. For more information about string blocks, see String Data Block, page 4-64.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, host discovery, or user event: • 1 - intrusion • 2 - host discovery • 3 - user
Event Device ID	uint32	Identification number of the device that generated the event that triggered the correlation event. You can obtain device name by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Firepower System preprocessor or rules engine that generated the event.
(Trigger) Event Second	uint32	UNIX timestamp indicating the time of the event that triggered the correlation policy rule (in seconds from 01/01/1970).
(Trigger) Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the Cisco device.

Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See Table 3-23 on page 3-40 for a list of each bit value.
Event Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Management Center. An `x` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX`, `XXX1XXXX`, `X1XXXXXX`, `1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
IP Protocol	uint8	Identifier of the IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP Address	uint8[4]	This field is reserved but no longer populated. The Source IPv4 address is stored in the Source IPv6 Address field. See IP Addresses, page 1-6 for more information.

Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	This field is reserved but no longer populated. The Destination IPv4 address is stored in the Destination IPv6 Address field. See IP Addresses, page 1-6 for more information.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High

Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Service ID	uint32	Identification number for the server running on the source host.
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (drop when deployment is inline, switched, or routed) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in inline, switched, or routed deployment.
Ingress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the ingress interface associated with correlation event.
Egress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the egress interface associated with correlation event.
Ingress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the ingress security zone associated with correlation event.
Egress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the egress security zone associated with correlation event.
Source IPv6 Address	uint8[16]	IP address of the source host in the event, in IPv6 address octets.
Destination IPv6 Address	uint8[16]	IP address of the destination host in the event, in IPv6 address octets.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Security Intelligence UUID	uint8[16]	The UUID of the access control policy configured for Security Intelligence.
Security Context	uint8[16]	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Policy ID	uint8[16]	ID number of the SSL policy that handled the connection.
SSL Rule ID	uint32	ID number of the SSL rule or default action that handled the connection.

Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint32	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

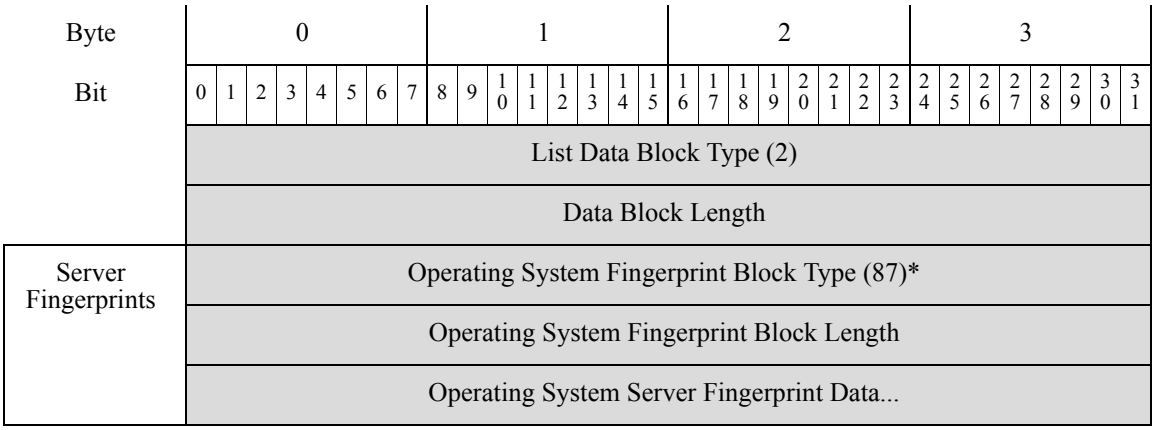
Table 3-25 Correlation Event 5.4+ Data Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint32	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.

Understanding Series 2 Data Blocks

Beginning in version 4.10.0, the eStreamer service uses a second series of data blocks to package certain records such as intrusion event extra data. See Table 3-26 on page 3-52 for a list of all block types in the series. Series 2 blocks, like series 1 blocks, support variable-length fields and hierarchies of nested blocks. The series 2 block types include primitive blocks that provide the same mechanism for encapsulating nested inner blocks as the series 1 primitive block types. However, series 2 blocks and series 1 blocks have separate numbering systems.

The following example shows the how primitive blocks are used. The list data block (series 2 block type 31) defines an array of operating system fingerprints (each of which is a type 87 block itself with variable length). The overall type 31 data block length is self-describing via the Data Block Length field, which contains the length of the data portion of the message, excluding the 8 bytes in the block type and block length fields.



In the following table, the Data Block Status field indicates whether the block is current (the latest version) or legacy (used in an older version and can still be requested through eStreamer).

Table 3-26 Series 2 Block Types

Type	Content	Data Block Status	Description
0	String	Current	Encapsulates variable string data. See String Data Block, page 3-55 for more information.
1	BLOB	Current	Encapsulates binary data and is used specifically for banners. See BLOB Data Block, page 3-56 for more information.
2	List	Current	Encapsulates a list of other data blocks. See List Data Block, page 3-57 for more information.
3	Generic List	Current	Encapsulates a list of other data blocks. For deserialization, it is the equivalent of the List data block. See Generic List Data Block, page 3-58 for more information.
4	Event Extra Data	Current	Contains intrusion event extra data. See Intrusion Event Extra Data Record, page 3-26 for more information.

Table 3-26 Series 2 Block Types (continued)

Type	Content	Data Block Status	Description
5	Extra Data Type	Current	Contains extra data metadata. See Intrusion Event Extra Data Metadata, page 3-28 for more information.
14	UUID String Mapping	Current	Block used by various metadata messages to map UUID values to descriptive strings. See UUID String Mapping Data Block, page 3-58.
15	Access Control Policy Rule ID Metadata	Current	Contains metadata for access control rules. See Access Control Policy Rule ID Metadata Block, page 3-61.
16	Malware Event	Legacy	Contains information on malware events, such as the malware detected or quarantined within a Cisco Advanced Malware Protection cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.1, page B-46. Deprecated by block 24, Malware Event Data Block 5.3.1, page B-70.
19	ICMP Type Data Block	Current	Contains metadata describing ICMP types. See ICMP Type Data Block, page 3-62.
20	ICMP Code Data Block	Current	Contains metadata describing ICMP codes. See ICMP Code Data Block, page 3-63.
21	Access Control Policy Rule Reason Data Block	Current	Contains information explaining access control policy rule reasons. See Access Control Policy Rule Reason Data Block for 6.0+, page 3-69.
22	IP Reputation Category Data Block	Current	Contains information on IP reputation categories explaining why an IP address was blocked. See Access Control Policy Name Data Block, page 3-70.
23	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.1.1.x, page B-178. It is superseded by block 32, Access Control Policy Rule ID Metadata Block, page 3-61.
24	Malware Event	Legacy	Contains information on malware events, such as the malware detected or quarantined within a Cisco Advanced Malware Protection cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.1.1.x, page B-50. Deprecates block 16, Malware Event Data Block 5.1, page B-46. Deprecated by block 33, Malware Event Data Block 5.3.1, page B-70.
25	Intrusion Event	Legacy	Contains information on intrusion events, including information to match intrusion events with connection and malware events. See Intrusion Event Record 5.1.1.x, page B-23. Deprecated by block 34, Intrusion Event Record 5.2.x, page B-12.
26	File Event SHA Hash	Legacy	Contains the SHA hash and name of files that have been identified as containing malware. See File Event SHA Hash for 5.1.1-5.2.x, page B-208. Deprecated by block 40, File Event SHA Hash for 5.3+, page 3-94.

Table 3-26 Series 2 Block Types (continued)

Type	Content	Data Block Status	Description
27	Rule Documentation Data Block	Current	Contains information about rules used to generate events. See Rule Documentation Data Block for 5.2+, page 3-97 for more information.
28	Geolocation Data Block	Current	Contains country codes and associated country name. See Geolocation Data Block for 5.2+, page 3-104.
32	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.2.x, page B-182. It deprecates File Event for 5.1.1.x, page B-178. Deprecated by block 38, File Event for 5.3, page B-186.
33	Malware Event	Current	Contains information on malware events, such as the malware detected or quarantined within a Cisco Advanced Malware Protection cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.2.x, page B-56. Deprecates block 24, Malware Event Data Block 5.1.1.x, page B-50. Deprecated by block 35, Malware Event Data Block 5.3, page B-63.
34	Intrusion Event	Legacy	Contains information on intrusion events, including information to match intrusion events with connection and malware events. See Intrusion Event Record 5.2.x, page B-12. Deprecates block 25. Deprecated by block 41, Intrusion Event Record 5.3, page B-17.
35	Malware Event	Legacy	Contains information on malware events, including IOC information. See Malware Event Data Block 5.3, page B-63. Deprecates block 33, Malware Event Data Block 5.2.x, page B-56. Deprecated by block 44, Malware Event Data Block 5.3, page B-63.
38	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.3, page B-186. It deprecates block 32. Deprecated by block 43, Malware Event Data Block 6.0+, page 3-83.
39	IOC Name Data Block	Current	Contains information about IOCs. See IOC Name Data Block for 5.3+, page 4-29
40	File Event SHA Hash	Current	Contains the SHA hash and name of files that have been identified as containing malware. See File Event SHA Hash for 5.3+, page 3-94. Deprecates block 26, File Event SHA Hash for 5.1.1-5.2.x, page B-208.
41	Intrusion Event	Legacy	Contains information on intrusion events, including information to match intrusion events with IOCs. See Intrusion Event Record 5.3, page B-17. Deprecates block 34. Deprecated by block 42, Intrusion Event Record 5.3.1, page B-29.

Table 3-26 Series 2 Block Types (continued)

Type	Content	Data Block Status	Description
42	Intrusion Event	Current	Contains information on intrusion events, including information to match intrusion events with IOCs. See Intrusion Event Record 5.3.1, page B-29. Deprecates block 41, Intrusion Event Record 5.3, page B-17.
43	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.3.1, page B-192. Deprecates block 38, File Event for 5.3, page B-186. Deprecated by block 46, File Event for 6.0+, page 3-73
44	Malware Event	Legacy	Contains information on malware events, including IOC information. See Malware Event Data Block 6.0+, page 3-83. Deprecates block 35, Malware Event Data Block 5.3, page B-63. Deprecated by block 47, Malware Event Data Block 6.0+, page 3-83
46	File Event	Current	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See Malware Event Data Block 6.0+, page 3-83. Deprecates block 43, File Event for 5.3.1, page B-192.
47	Malware Event	Current	Contains information on malware events, including IOC information. See Malware Event Data Block 6.0+, page 3-83. Deprecates block 44, Malware Event Data Block 5.3.1, page B-70.

Series 2 Primitive Data Blocks

Both series 2 and series 1 blocks include a set of primitives that are used to encapsulate lists of variable-length blocks as well as variable-length strings and BLOBs within messages. These primitive blocks have the standard eStreamer block header discussed above in Data Block Header, page 2-24, but they appear only within other data blocks. Any number can be included in a given block type. For details on the structure of these blocks, see the following:

- String Data Block, page 3-55
- BLOB Data Block, page 3-56
- List Data Block, page 3-57
- Generic List Data Block, page 3-58
- UUID String Mapping Data Block, page 3-58
- Name Description Mapping Data Block, page 3-59

String Data Block

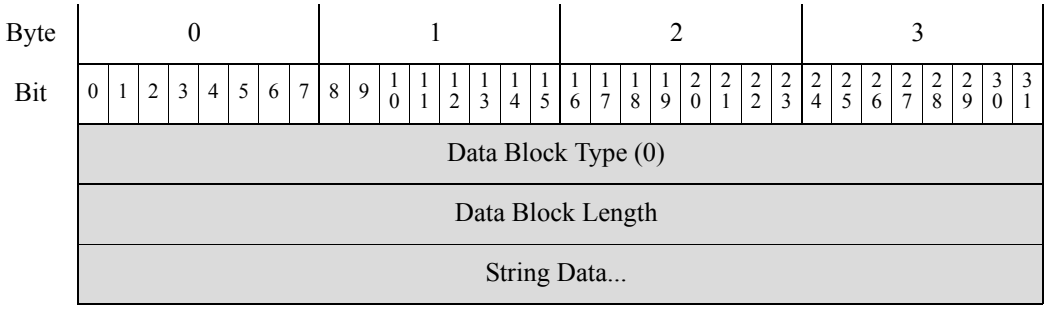
The eStreamer service uses the String data block to send string data in messages. These blocks commonly appear within other data blocks to identify, for example, operating system or server names.

Empty String data blocks (containing no data, only the header fields) have a block length of 8. eStreamer uses an empty String data block when it has no content for a string value, as might happen, for example, in the OS vendor string field in an Operating System data block when the vendor of the operating system is unknown.

The String data block has a block type of 0 in the series 2 group of blocks.

Note Strings returned in this data block are not always null-terminated (that is, the string characters are not always followed by a 0).

The following diagram shows the format of the String data block:



The following table describes the fields of the String data block.

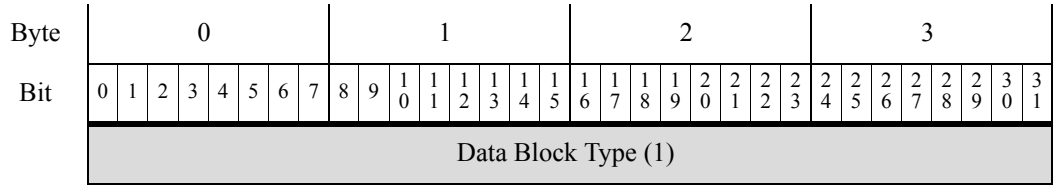
Table 3-27 String Block Fields

Field	Data Type	Description
Data Block Type	uint32	Initiates a String data block. This value is always `0`.
Data Block Length	uint32	Combined length in bytes of the string data block header and string data.
String Data	string	Contains the string data and may contain a terminating character (null byte) at the end of the string.

BLOB Data Block

The eStreamer service uses the BLOB data block to convey binary data. For example, host discovery records use the BLOB block to hold captured server banners. The BLOB data block has a block type of 1 in the series 2 group of blocks.

The following diagram shows the format of the BLOB data block:



Data Block Length
Binary Data...

The following table describes the fields of the BLOB data block.

Table 3-28 BLOB Data Block Fields

Field	Data Type	Description
Data Block Type	uint32	Initiates a BLOB data block. This value is always `1`.
Data Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Binary Data	variable	Contains binary data such as a server banner.

List Data Block

The eStreamer service uses the List data block to encapsulate a list of data blocks. For example, eStreamer can use the List data block to send a list of TCP servers, each of which is itself a data block. The List data block has a block type of 2 in the series 2 group of blocks.

The following diagram shows the basic format of a List data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Block Type (2)																																
Block Length																																
Encapsulated Data Blocks...																																

The following table describes the fields of the List data block.

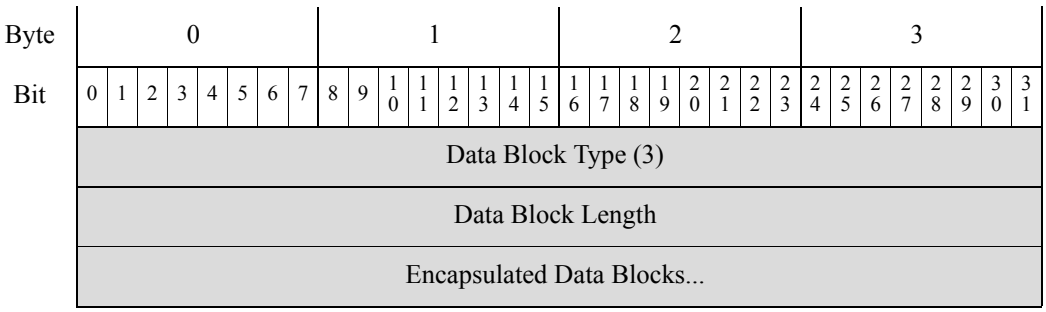
Table 3-29 List Data Fields

Field	Data Type	Description
Block Type	uint32	Initiates a List data block. This value is always `2`.
Block Length	uint32	Number of bytes in the List block and encapsulated data. For example, if there were three Sub-Server data blocks included in the list, the value here would include the total number of bytes in the Sub-Server blocks, plus eight bytes for the List block header.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Generic List Data Block

The eStreamer service uses the Generic List data block to encapsulate a list of data blocks. For example, the Host Profile data block contains information about multiple client applications and uses the Generic List block to embed a list of Client Application data blocks in the message. The Generic List data block has a block type of 3 in the series 2 group of blocks.

The following diagram shows the basic structure of a Generic List data block:



The following table describes the fields of the Generic List data block.

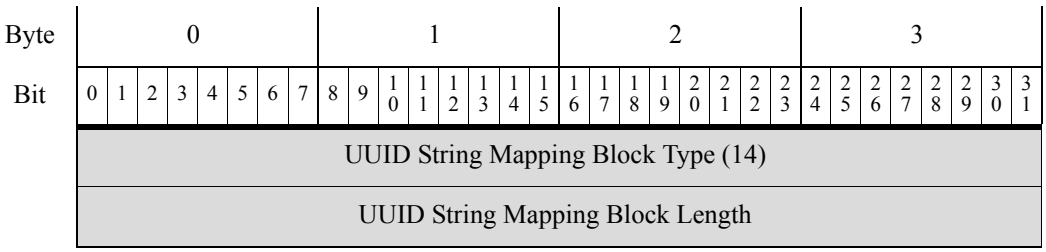
Table 3-30 Generic List Data Block Fields

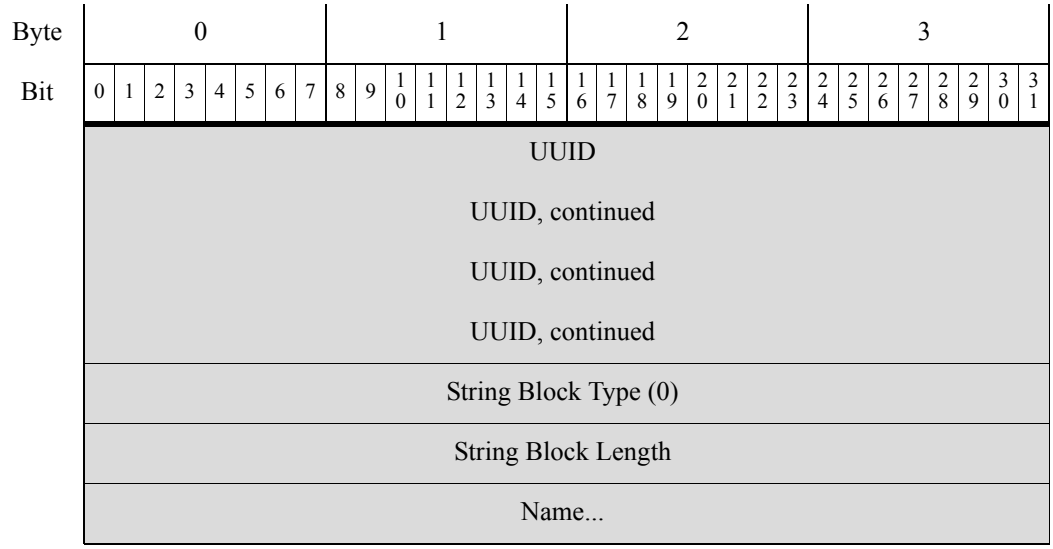
Field	Number of Bytes	Description
Data Block Type	uint32	Initiates a Generic List data block. This value is always 3.
Data Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the total number of bytes in all of the encapsulated data blocks.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the Generic List block length.

UUID String Mapping Data Block

The eStreamer service uses the UUID String Mapping data block in various metadata messages to map UUID values to descriptive strings. The UUID String Mapping data block has a block type of 14 in series 2.

The following diagram shows the structure of the UUID String Mapping data block.





The following table describes the fields in the UUID String Mapping data block.

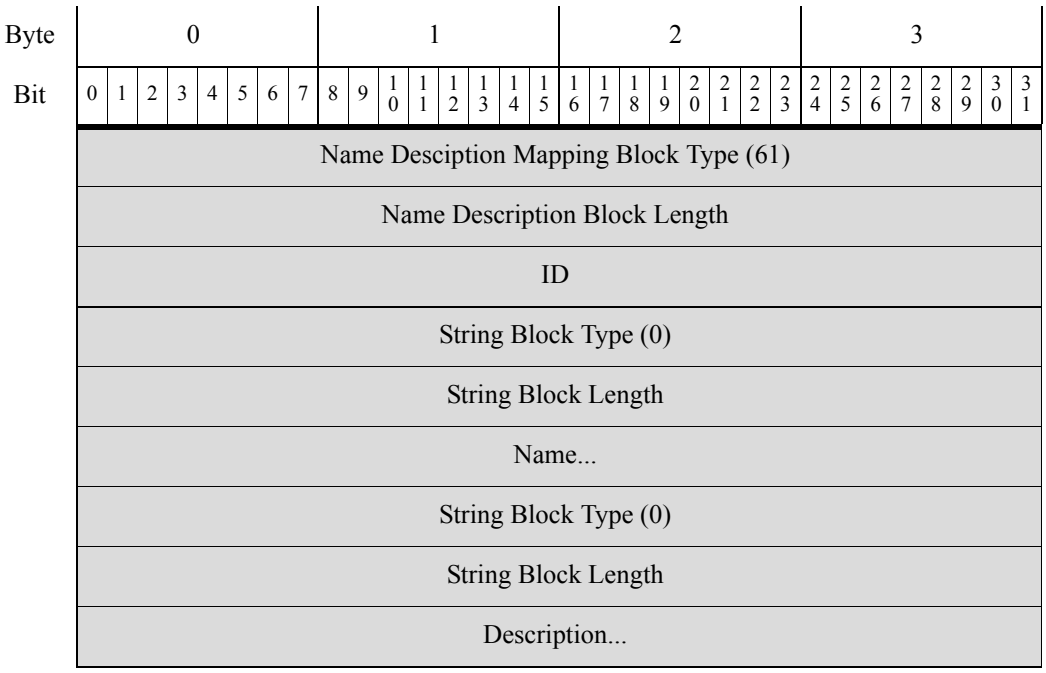
Table 3-31 UUID String Mapping Data Block Fields

Field	Data Type	Description
UUID String Mapping Block Type	uint32	Initiates a UUID String Mapping block. This value is always `14`.
UUID String Mapping Block Length	uint32	Total number of bytes in the UUID String Mapping block, including eight bytes for the UUID String Mapping block type and length fields, plus the number of bytes of data that follows.
UUID	uint8[16]	The unique identifier for the event or other object the UUID identifies.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the UUID. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name.

Name Description Mapping Data Block

The eStreamer service uses the Name Description Mapping data block in various metadata messages to map ID values to names and descriptive strings. The Name Description Mapping data block has a block type of 61 in series 2.

The following diagram shows the structure of the Name Description Mapping data block.



The following table describes the fields in the Name Description Mapping data block.

Table 3-32 Name Description Mapping Data Block Fields

Field	Data Type	Description
Name Description Mapping Block Type	uint32	Initiates a Name Description Mapping block. This value is always `61`.
Name Description Mapping Block Length	uint32	Total number of bytes in the Name Description Mapping block, including eight bytes for the Name Description Mapping block type and length fields, plus the number of bytes of data that follows.
ID	unit32	The unique identifier for the event or other object the ID identifies.
String Block Type	uint32	Initiates a String data block containing the name associated with the ID. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The name of the event or object.
String Block Type	uint32	Initiates a String data block containing the description associated with the ID. This value is always `0`.
String Block Length	uint32	The number of bytes included in the description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	A description of the object or event associated with the ID.

Access Control Policy Rule ID Metadata Block

The eStreamer service uses the Access Control Policy Rule ID metadata block to contain information about access control policy rule IDs. This data block has a block type of 15 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.

	Byte	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
		Access Control Policy Rule ID Metadata Block Type (15)																															
		Access Control Policy Rule ID Metadata Block Length																															
		Revision Revision, continued Revision, continued Revision, continued																															
		Rule ID																															
Name		String Block Type (0)																															
		String Block Length																															
		Name...																															

The following table describes the fields in the Access Control Policy Rule ID Metadata block.

Table 3-33 Access Control Policy Rule ID Metadata Block Fields

Field	Data Type	Description
Access Control Policy Rule ID Metadata Block Type	uint32	Initiates a Access Control Policy Rule ID Metadata block. This value is always `15`.
Access Control Policy Rule ID Metadata Block Length	uint32	Total number of bytes in the Access Control Policy Rule ID block, including eight bytes for the Access Control Policy Rule ID metadata block type and length fields, plus the number of bytes of data that follows.
Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event.
Rule ID	uint32	Internal identifier for the rule that triggered the event.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control policy rule. This value is always `0`.

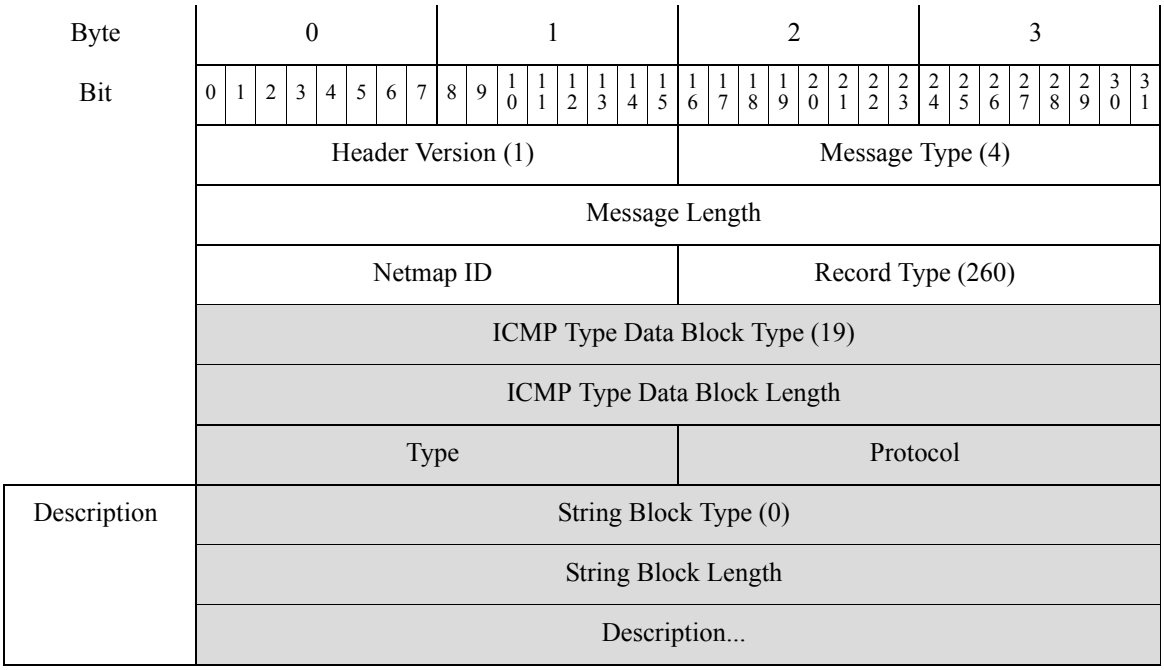
Table 3-33 Access Control Policy Rule ID Metadata Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name of the access control policy rule.

ICMP Type Data Block

The eStreamer service uses the ICMP Type data block to contain information about ICMP Types. This data block has a record type of 260, and a block type of 19 in series 2.

The following diagram shows the structure of the ICMP Type data block.



The following table describes the fields in the ICMP Type data block.

Table 3-34 ICMP Type Data Block Fields

Field	Data Type	Description
ICMP Type Data Block Type	uint32	Initiates an ICMP Type data block. This value is always `19`.
ICMP Type Data Block Length	uint32	Total number of bytes in the ICMP Type data block, including eight bytes for the ICMP Type data block type and length fields, plus the number of bytes of data that follows.
Type	uint16	The ICMP type of the event.

Table 3-34 ICMP Type Data Block Fields (continued)

Field	Data Type	Description
Protocol	uint16	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP
String Block Type	uint32	Initiates a String data block containing the description of the ICMP type. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the ICMP type for the event.

ICMP Code Data Block

The eStreamer service uses the ICMP Code data block to contain information about access control policy rule IDs. This data block has a record type of 270, and block type of 20 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (270)															
	ICMP Code Data Block Type (20)																															
	ICMP Code Data Block Length																															
	Code																Type															
Description	Protocol																String Block Type (0)															
	String Block Type (0), continued																String Block Length															
	String Block Length, continued																Description...															

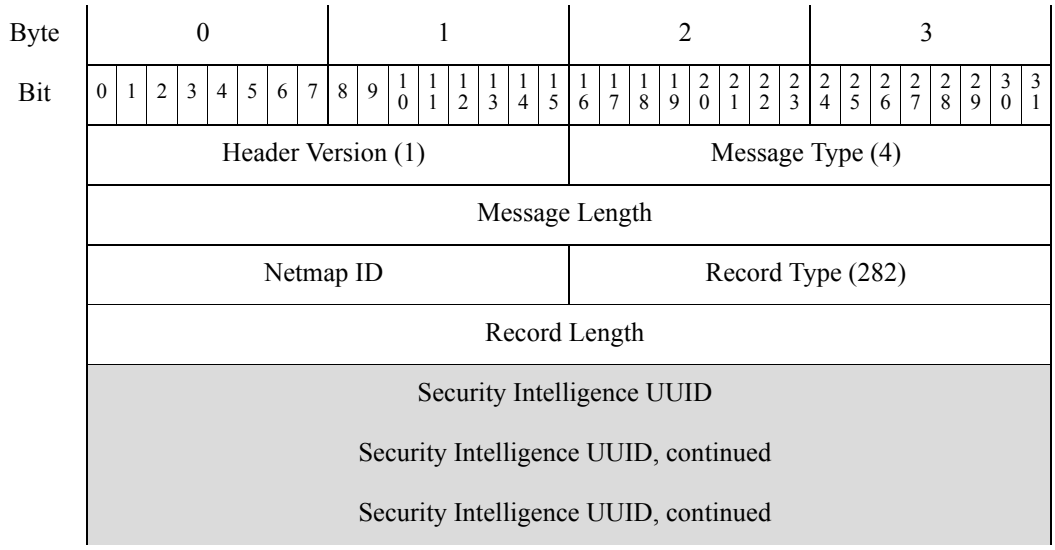
The following table describes the fields in the ICMP Code data block.

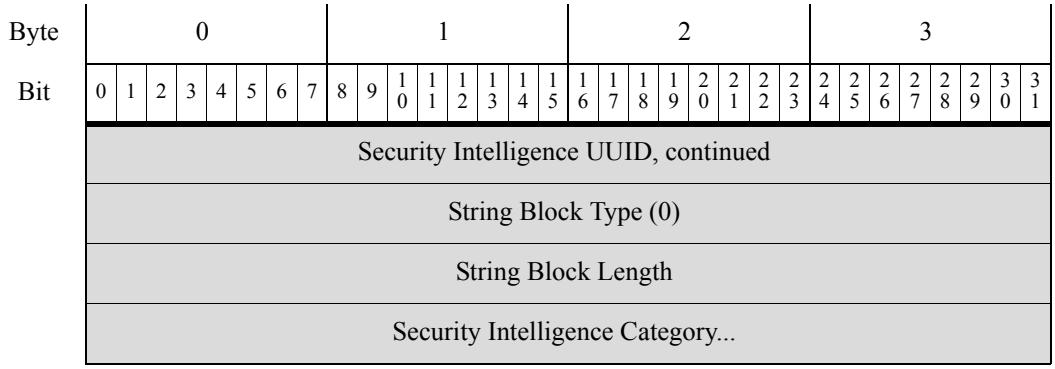
Table 3-35 ICMP Code Data Block Fields

Field	Data Type	Description
ICMP Code Data Block Type	uint32	Initiates a ICMP Code data block. This value is always 20.
ICMP Code Data Block Length	uint32	Total number of bytes in the ICMP Code data block, including eight bytes for the ICMP Code data block type and length fields, plus the number of bytes of data that follows.
Code	uint16	The ICMP code of the event.
Type	uint16	The ICMP type of the event.
Protocol	uint16	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP
String Block Type	uint32	Initiates a String data block containing the description of the ICMP code. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the ICMP code for the event.

Security Intelligence Category Metadata for 5.4.1+

The eStreamer service transmits metadata containing Security Intelligence Category information, the format of which is shown below. Note that the Record Type field, which appears after the Message Length field, has a value of 282, indicating a Security Intelligence Category record.





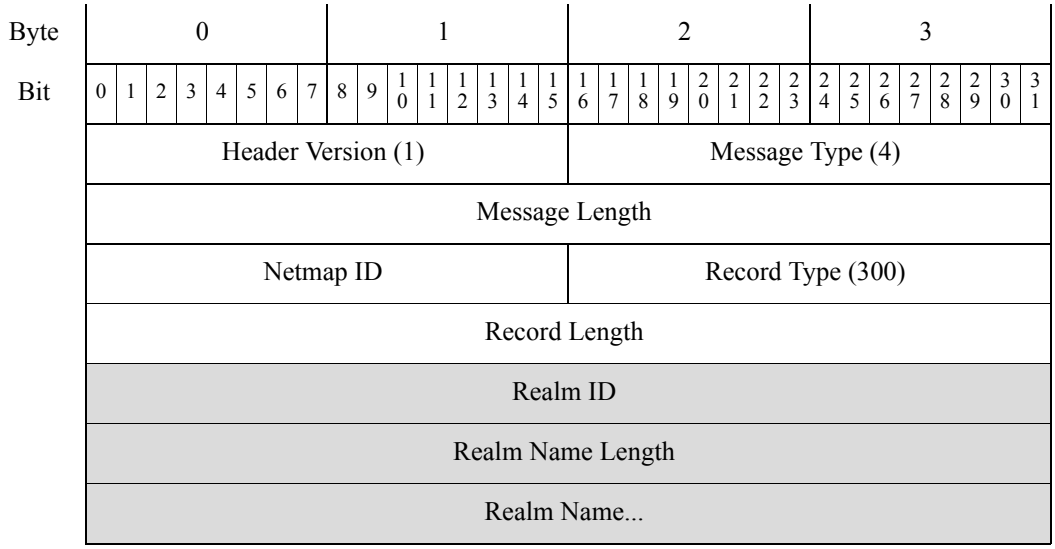
The following table describes the fields in the Security Context Name record.

Table 3-36 Security Context Name Record Fields

Field	Data Type	Description
Security Intelligence UUID	uint8[16]	The UUID of the Security Intelligence.
String Block Type	uint32	Initiates a String data block containing the Security Intelligence category. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Security Intelligence Category String data block, including eight bytes for the block type and header fields plus the number of bytes in the Profile Name field.
Security Intelligence Category	string	The Security Intelligence Category.

Realm Metadata for 6.0+

The eStreamer service transmits metadata containing Realm information, the format of which is shown below. Note that the Record Type field, which appears after the Message Length field, has a value of `300`, indicating a Realm Metadata record.



The following table describes the fields in the Realm Metadata record.

Table 3-37 Realm Metadata Record Fields

Field	Data Type	Description
Realm ID	uint32	The ID number of the realm.
Realm Name Length	uint32	The number of bytes included in the Realm Name.
Realm Name	string	The realm name

Endpoint Profile Data Block for 6.0+

The eStreamer service uses the Endpoint Profile data block to contain information about network endpoints. This data block has a record type of 301, and block type of 58 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (301)															
	Endpoint Profile Block Type (58)																															
	Endpoint Profile Data Block Length																															
	ID																															
Profile Name	String Block Type (0)																															
	String Block Length																															
	Profile Name...																															
Full Name	String Block Type (0)																															
	String Block Length																															
	Full Name...																															

The following table describes the fields in the Endpoint Profile data block.

Table 3-38 Endpoint Profile Data Block Fields

Field	Data Type	Description
Endpoint Profile Data Block Type	uint32	Initiates a ICMP Code data block. This value is always `58`.
Endpoint Profile Data Block Length	uint32	Total number of bytes in the Endpoint Profile data block, including eight bytes for the Endpoint Profile data block type and length fields, plus the number of bytes of data that follows.
ID	uint32	ID number of the endpoint.
String Block Type	uint32	Initiates a String data block containing the profile of the endpoint. This value is always `0`.
String Block Length	uint32	The number of bytes included in the profile name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Profile Name field.
Profile Name	string	Name of the endpoint profile.
String Block Type	uint32	Initiates a String data block containing the full name of the endpoint. This value is always `0`.
String Block Length	uint32	The number of bytes included in the full name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Full Name field.
Full Name	string	Fully qualified name of the profile, providing the relationship hierarchy of the type of endpoint.

Security Group Metadata for 6.0+

The eStreamer service transmits metadata containing Security Group information, the format of which is shown below. Note that the Record Type field, which appears after the Message Length field, has a value of `302`, indicating a Security Group Metadata record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (302)															
Record Length																																
Security Group ID																																
Security Group Name Length																																
Security Group Name...																																

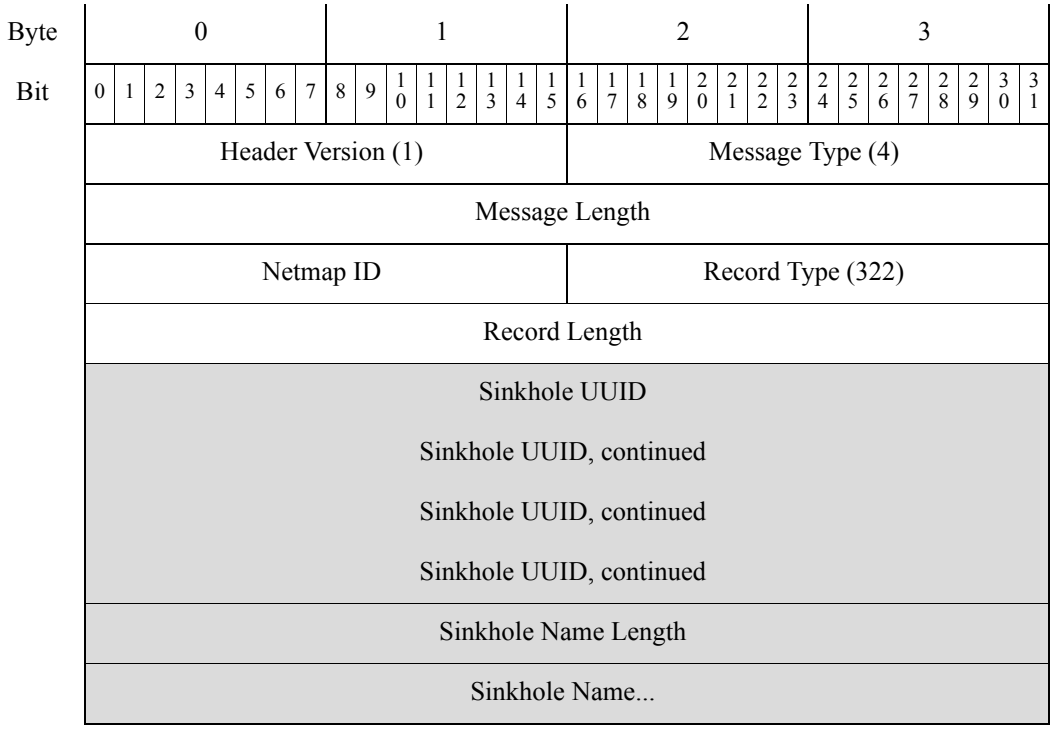
The following table describes the fields in the Security Group Metadata record.

Table 3-39 Security Group Metadata Record Fields

Field	Data Type	Description
Security Group ID	uint32	The ID number of the security group.
Security Group Name Length	uint32	The number of bytes included in the Security Group Name.
Security Group Name	string	The security group name

Sinkhole Metadata for 6.0+

The eStreamer service transmits metadata containing Sinkhole information, the format of which is shown below. Note that the Record Type field, which appears after the Message Length field, has a value of `322`, indicating a Sinkhole Metadata record.



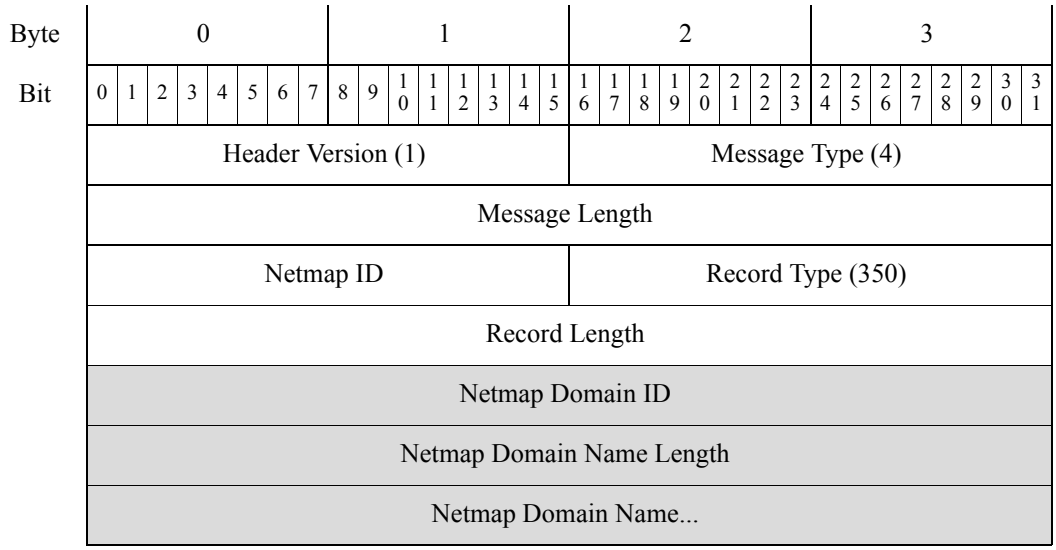
The following table describes the fields in the Sinkhole Metadata record.

Table 3-40 Sinkhole Metadata Record Fields

Field	Data Type	Description
Sinkhole UUID	uint8[16]	The UUID number of the sinkhole.
Sinkhole Name Length	uint32	The number of bytes included in the Sinkhole Name.
Sinkhole Name	string	The sinkhole name

Netmap Domain Metadata for 6.0+

The eStreamer service transmits metadata containing Netmap Domain information, the format of which is shown below. Note that the Record Type field, which appears after the Message Length field, has a value of `350`, indicating a Netmap Domain Metadata record.



The following table describes the fields in the Netmap Domain Metadata record.

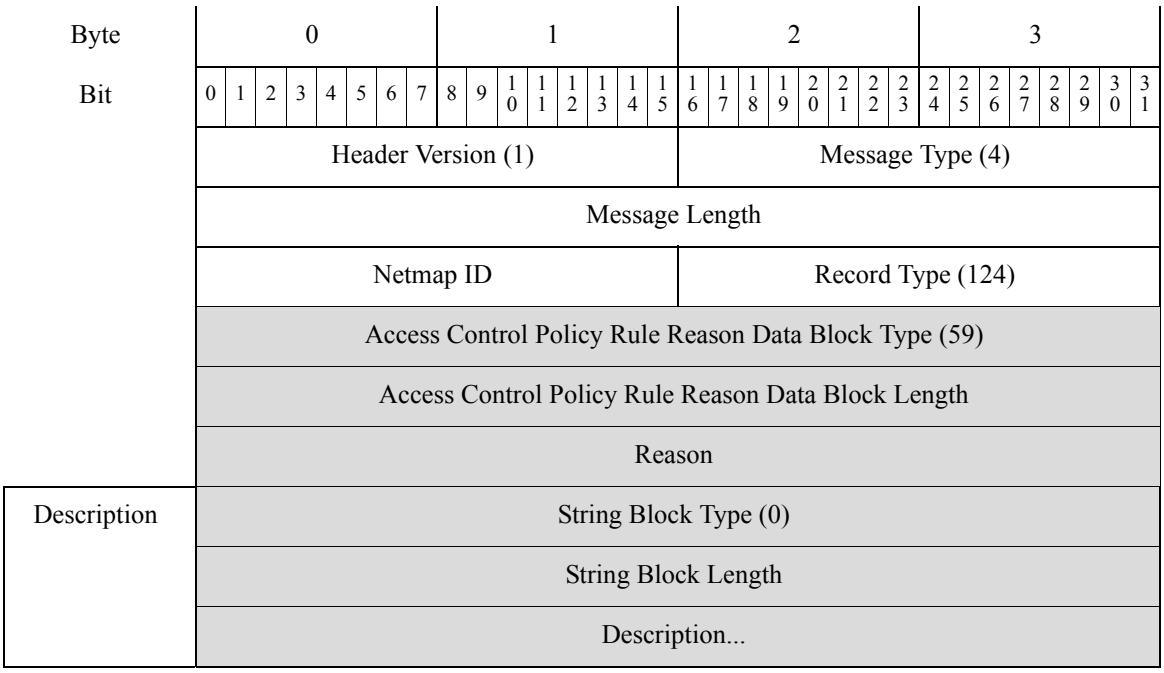
Table 3-41 Sinkhole Metadata Record Fields

Field	Data Type	Description
Netmap Domain ID	uint32	The ID number of the netmap domain.
Netmap Domain Name Length	uint32	The number of bytes included in the Netmap Domain Name.
Netmap Domain Name	string	The netmap domain name

Access Control Policy Rule Reason Data Block for 6.0+

The eStreamer service uses the Access Control Rule Policy Rule Reason Data block to contain information about access control policy rule IDs. This data block has a record type of 124, and a block type of 59 in series 2. It supersedes block type 21. The Reason field has been increased from 16 bits to 32 bits.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.



The following table describes the fields in the Access Control Policy Rule Reason data block.

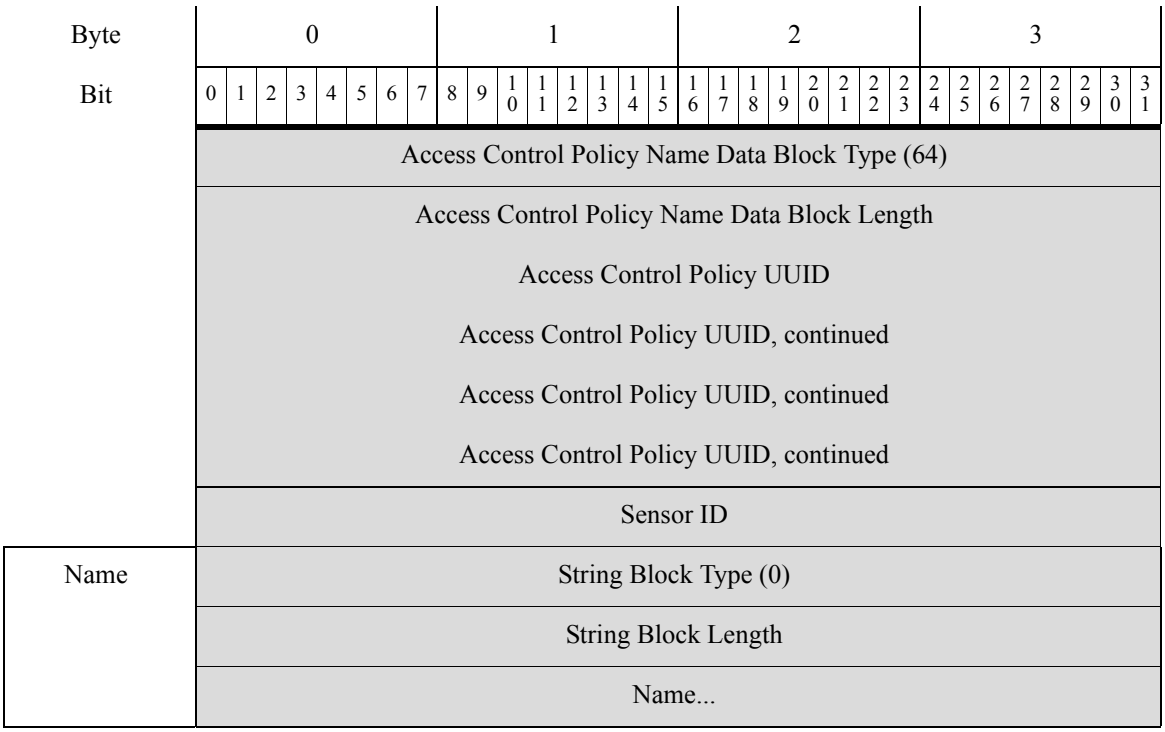
Table 3-42 Access Control Policy Rule Reason Data Block Fields

Field	Data Type	Description
Access Control Policy Rule Reason Data Block Type	uint32	Initiates an Access Control Policy Rule Reason data block. This value is always `59`.
Access Control Policy Rule Reason Data Block Length	uint32	Total number of bytes in the Access Control Policy Rule Reason data block, including eight bytes for the Access Control Policy Rule Reason data block type and length fields, plus the number of bytes of data that follows.
Reason	uint32	The number of the reason for the rule that triggered the event.
String Block Type	uint32	Initiates a String data block containing the description of the access control policy rule reason. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the reason for the rule.

Access Control Policy Name Data Block

The eStreamer service uses the Access Control Policy Name Data block to contain information about access control policy names. This data block has a a block type of 64 in series 2.

The following diagram shows the structure of the Access Control Policy Name metadata block.



The following table describes the fields in the Access Control Policy Name metadata block.

Table 3-43 Access Control Policy Policy Name Data Block Fields

Field	Data Type	Description
Access Control Policy Name Data Block Type	uint32	Initiates an Access Control Policy Name data block. This value is always `64`.
Access Control Policy Name Data Block Length	uint32	Total number of bytes in the Access Control Policy Name data block, including eight bytes for the Access Control Policy Name data block type and length fields, plus the number of bytes of data that follows.
Access Control Policy UUID	uint8[16]	UUID of the Access Control Policy
Sensor ID	uint32	ID Number of the sensor associated with the access control policy
String Block Type	uint32	Initiates a String data block containing the name of the access control policy. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	Name of the access control policy

IP Reputation Category Data Block

The eStreamer service uses the IP Reputation Category Data block to contain information about rule reputation categories. This data block has a block type of 22 in series 2.

The following diagram shows the structure of the IP Reputation Category data block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	IP Reputation Category Data Block Type (22)																															
	IP Reputation Category Data Block Length																															
	Rule ID																															
	Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																															
Description	String Block Type (0)																															
	String Block Length																															
	Category Name...																															

The following table describes the fields in the IP Reputation Category Data Block.

Table 3-44 IP Reputation Category Data Block Fields

Field	Data Type	Description
IP Reputation Category Data Block Type	uint32	Initiates a IP Reputation Category data block. This value is always `22`.
IP Reputation Category Data Block Length	uint32	Total number of bytes in the IP Reputation Category data block, including eight bytes for the IP Reputation Category data block type and length fields, plus the number of bytes of data that follows.
Rule ID	uint32	Internal identifier for the rule that triggered the event.
Policy UUID	uint8[16]	UUID of the policy that triggered the event.
String Block Type	uint32	Initiates a String data block containing the description of the IP Reputation Category. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Category Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category Name field.
Category Name	string	Name of the category for the rule.

File Event for 6.0+

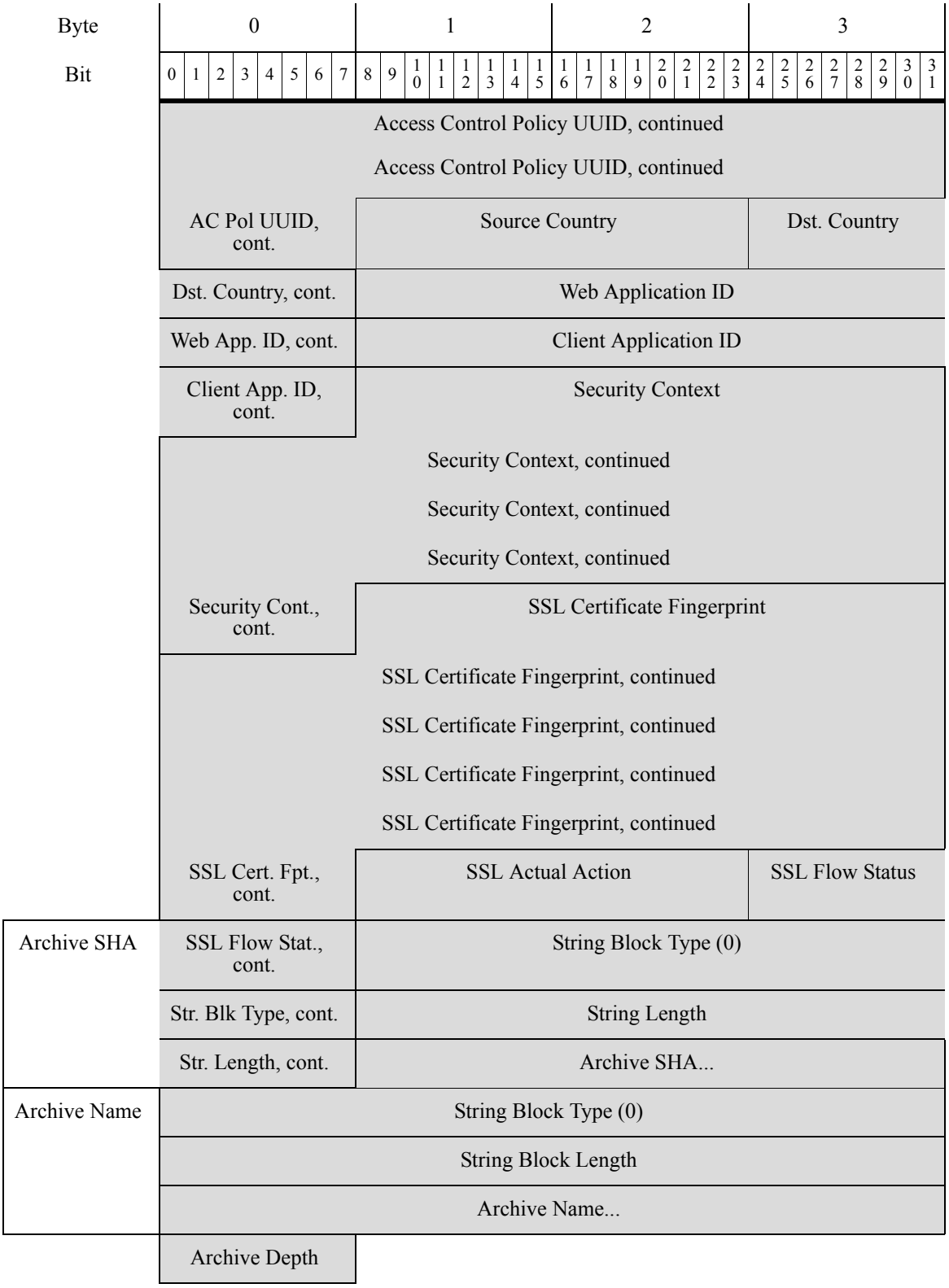
The File Event data block contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 56 in the series 2 group of blocks. It supersedes block type 46. Fields for ISE integration, file analysis, local malware analysis, and capacity handling statuses have been added.

You request file event records by setting the file event flag—bit 30 in the Request Flags field—in the request message with an event version of 5 and an event code of 111. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

The following graphic shows the structure of the File Event data block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
File Event Block Type (56)																																
File Event Block Length																																
Device ID																																
Connection Instance																	Connection Counter															
Connection Timestamp																																
File Event Timestamp																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Disposition									SPERO Disposition								File Storage Status								File Analysis Status							
Local Malware Analysis Stat.									Archive File Status								Threat Score								Action							
SHA Hash																																

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	File Type ID			
File Name	String Block Type (0)			
	String Block Length			
	File Name...			
	File Size			
	File Size, continued			
	Direction	Application ID		
	App ID, cont.	User ID		
URI	User ID, cont.	String Block Type (0)		
	String Block Type (0), cont.	String Block Length		
	String Block Length, cont.	URI...		
Signature	String Block Type (0)			
	String Block Length			
	Signature...			
	Source Port		Destination Port	
	Protocol	Access Control Policy UUID		
	Access Control Policy UUID, continued			



The following table describes the fields in the file event data block.

Table 3-45 File Event Data Block for 6.0+ Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always 56.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the AMP cloud for a disposition, or the AMP cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
SPERO Disposition	uint8	Indicates whether the SPERO signature was used in file analysis. If the value is 1, 2, or 3, SPERO analysis was used. If there is any other value SPERO analysis was not used.

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
File Storage Status	uint8	The storage status of the file. Possible values are: • 1 — File Stored • 2 — File Stored • 3 — Unable to Store File • 4 — Unable to Store File • 5 — Unable to Store File • 6 — Unable to Store File • 7 — Unable to Store File • 8 — File Size is Too Large • 9 — File Size is Too Small • 10 — Unable to Store File • 11 — File Not Stored, Disposition Unavailable

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
File Analysis Status	uint8	Indicates whether the file was sent for dynamic analysis. Possible values are: • 0 — File Not Sent for Analysis • 1 — Sent for Analysis • 2 — Sent for Analysis • 4 — Sent for Analysis • 5 — Failed to Send • 6 — Failed to Send • 7 — Failed to Send • 8 — Failed to Send • 9 — File Size is Too Small • 10 — File Size is Too Large • 11 — Sent for Analysis • 12 — Analysis Complete • 13 — Failure (Network Issue) • 14 — Failure (Rate Limit) • 15 — Failure (File Too Large) • 16 — Failure (File Read Error) • 17 — Failure (Internal Library Error) • 19 — File Not Sent, Disposition Unavailable • 20 — Failure (Cannot Run File) • 21 — Failure (Analysis Timeout) • 22 — Sent for Analysis • 23 —File Transmit File Capacity Handled — File capacity handled (stored on the sensor) because file could not be submitted to the sandbox for analysis • 25 — File Transmit Server Limited Exceeded Capacity Handled — File capacity handled due to rate limiting on server • 26 — Communication Failure — File capacity handled due to cloud connectivity failure • 27 — Not Sent — File not sent due to configuration • 28 — Preclass No Match —File not sent for dynamic analysis since pre-classification didn't find any embedded or suspicious object in the file • 29 — Transmit Sent Sandbox Private Cloud — File sent to the private cloud for dynamic analysis • 30 — Transmit Not Send Sendbox Private Cloud - File not send to the private cloud for analysis

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
Local Malware Analysis Status	uint8	The malware analysis status of the file. Possible values are: • 0 — File not Analyzed • 1 — Analysis Done • 2 — Analysis Failed • 3 — Manual Analysis Request
Archive File Status	uint8	The status of an archive being inspected. Can have the following values: • 0 — N/A — File is not being inspected as an archive • 1 — Pending — Archive is being inspected • 2 — Extracted — Successfully inspected without any problems • 3 — Failed — Failed to inspect, insufficient system resources • 4 — Depth Exceeded — Successful, but archive exceeded the nested inspection depth • 5 — Encrypted — Partially Successful, Archive was or contains an archive that is encrypted • 6 — Not Inspectable — Partially Successful, File is possibly Malformed or Corrupt
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist • 6 — Cloud Lookup Timeout • 7 — Custom Detection • 8 — Custom Detection Block • 9 — Archive Block (Depth Exceeded) • 10 — Archive Block (Encrypted) • 11 — Archive Block (Failed to Inspect)
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
File Type ID	uint32	ID number that maps to the file type. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
String Block Type	uint32	Initiates a String data block containing the Archive SHA. This value is always 0.

Table 3-45 File Event Data Block for 6.0+ Fields (continued)

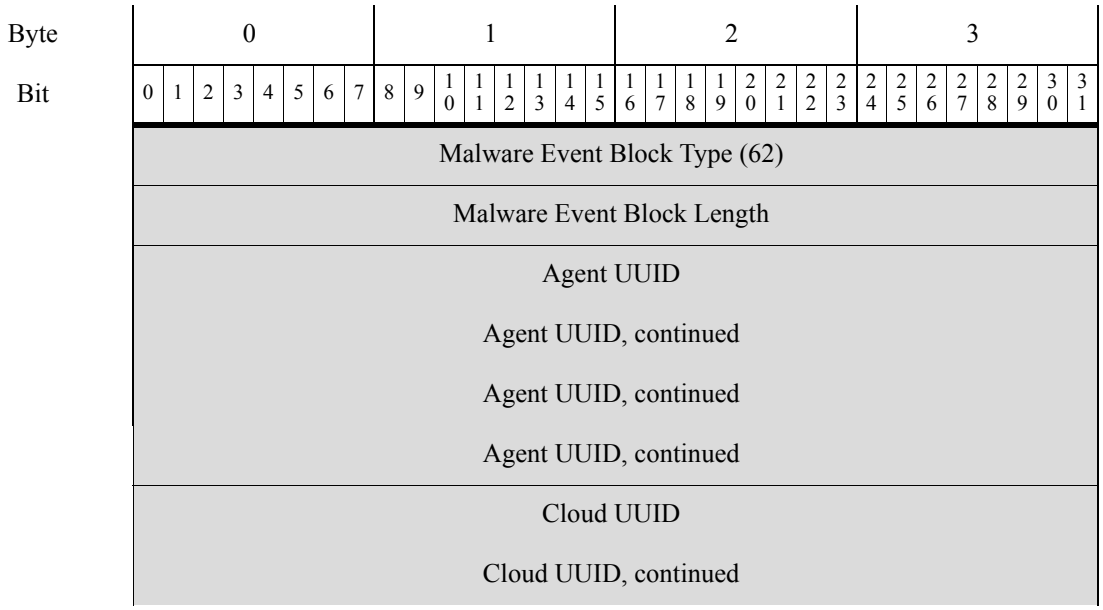
Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Archive SHA String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive SHA	string	SHA1 hash of the parent archive in which the file is contained.
String Block Type	uint32	Initiates a String data block containing the Archive Name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Archive Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive Name	string	Name of the parent archive.
Archive Depth	uint8	Number of layers in which the file is nested. For example, if a text file is in a zip archive, this has a value of `1`.

Malware Event Data Block 6.0+

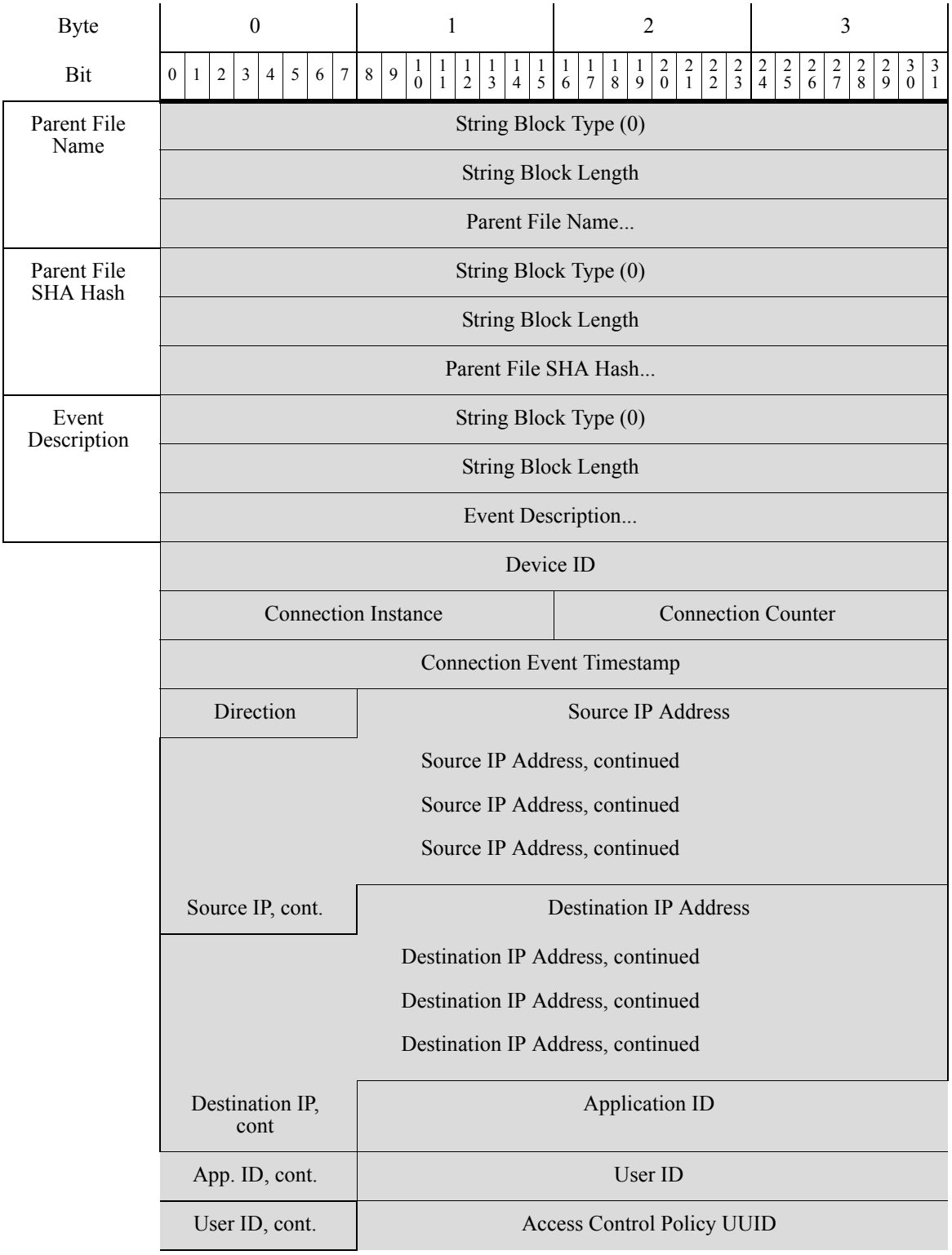
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 62 in the series 2 group of blocks. It supersedes block 47. A field for HTTP response has been added.

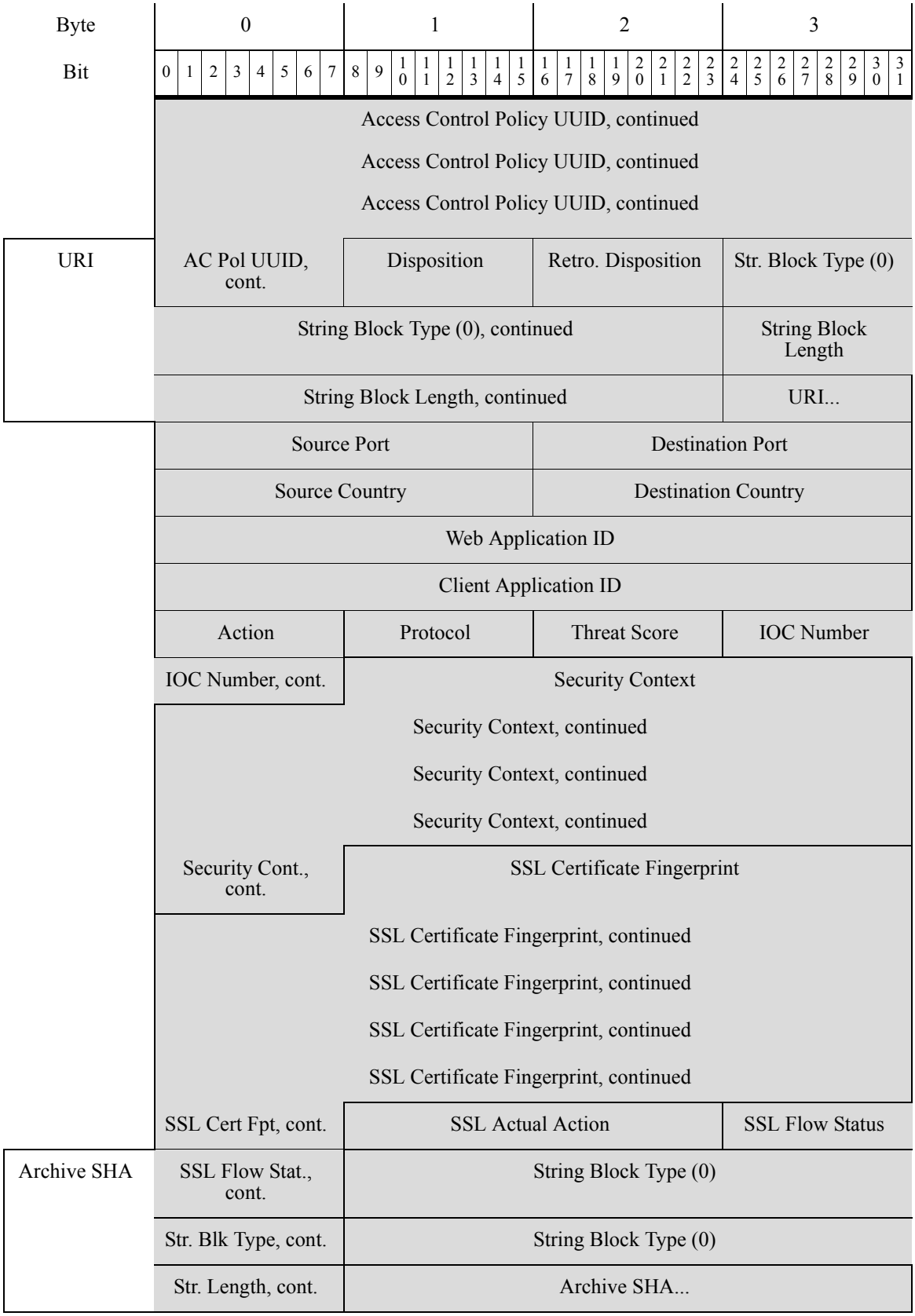
You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 7 and an event code of 101.

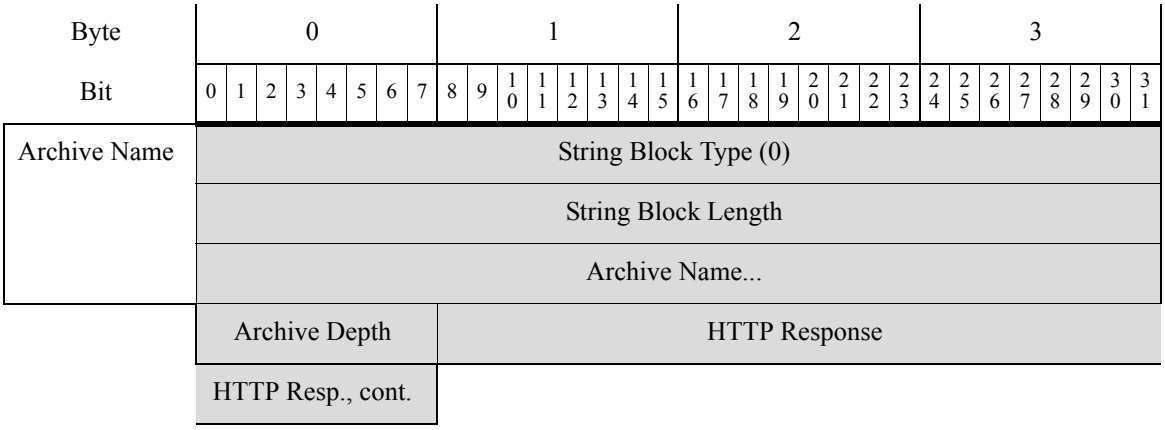
The following graphic shows the structure of the malware event data block.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
	Event Subtype ID																															
Detection Name	Detector ID								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Detection Name...																							
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															
File Path	String Block Type (0)																															
	String Block Length																															
	File Path...																															
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															
	File Type																															
	File Timestamp																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Parent File Name
String Block Type (0)
String Block Length
Parent File Name...
Parent File SHA Hash
String Block Type (0)
String Block Length
Parent File SHA Hash...
Event Description
String Block Type (0)
String Block Length
Event Description...
Device ID
Connection Instance
Connection Counter
Connection Event Timestamp
Direction
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Source IP, cont.
Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Destination IP, cont
Application ID
App. ID, cont.
User ID
User ID, cont.
Access Control Policy UUID


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Access Control Policy UUID, continued
Access Control Policy UUID, continued
Access Control Policy UUID, continued
URI
AC Pol UUID, cont.
Disposition
Retro. Disposition
Str. Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
URI...
Source Port
Destination Port
Source Country
Destination Country
Web Application ID
Client Application ID
Action
Protocol
Threat Score
IOC Number
IOC Number, cont.
Security Context
Security Context, continued
Security Context, continued
Security Context, continued
Security Cont., cont.
SSL Certificate Fingerprint
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Cert Fpt, cont.
SSL Actual Action
SSL Flow Status
Archive SHA
SSL Flow Stat., cont.
String Block Type (0)
Str. Blk Type, cont.
String Block Type (0)
Str. Length, cont.
Archive SHA...



The following table describes the fields in the malware event data block.

Table 3-46 Malware Event Data Block for 6.0+ Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 62.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the AMP cloud from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint32	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always `0`.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint32	The file type of the detected or quarantined file. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the AMP cloud for a disposition, or the AMP cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist • 6 — Cloud Lookup Timeout • 7 — Custom Detection • 8 — Custom Detection Block • 9 — Archive Block (Depth Exceeded) • 10 — Archive Block (Encrypted) • 11 — Archive Block (Failed to Inspect)
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
String Block Type	uint32	Initiates a String data block containing the Archive SHA. This value is always 0.

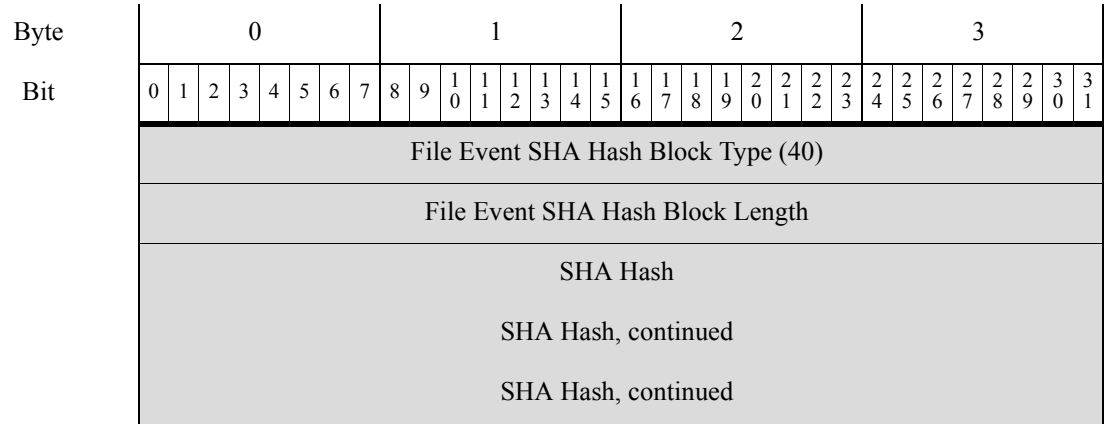
Table 3-46 Malware Event Data Block for 6.0+ Fields (continued)

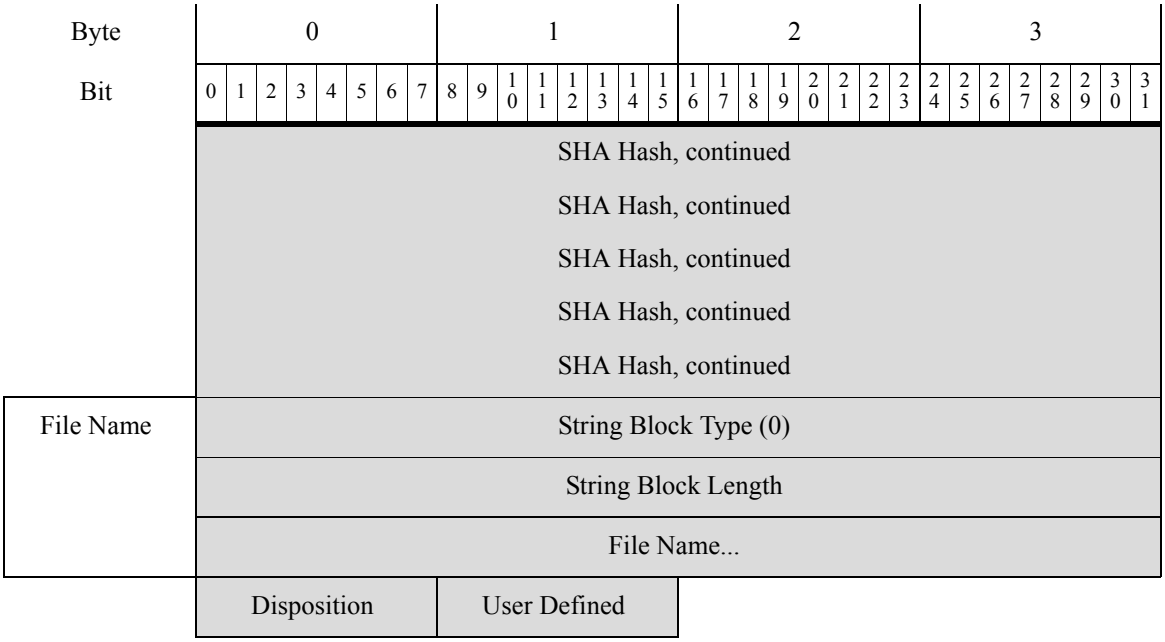
Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Archive SHA String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive SHA	string	SHA1 hash of the parent archive in which the file is contained.
String Block Type	uint32	Initiates a String data block containing the Archive Name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Archive Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive Name	string	Name of the parent archive.
Archive Depth	uint8	Number of layers in which the file is nested. For example, if a text file is in a zip archive, this has a value of `1`.
HTTP Response	uint32	Response code of the HTTP Request.

File Event SHA Hash for 5.3+

The eStreamer service uses the File Event SHA Hash data block to contain metadata of the mapping of the SHA hash of a file to its filename. The block type is 40 in the series 2 list of data blocks. It can be requested if file log events have been requested in the extended requests—event code `111`—and either bit 20 is set or metadata is requested with an event version of `5` and an event code of `21`.

The following diagram shows the structure of a file event hash data block:





The following table describes the fields in the file event SHA hash data block.

Table 3-47 File Event SHA Hash Data Block Fields

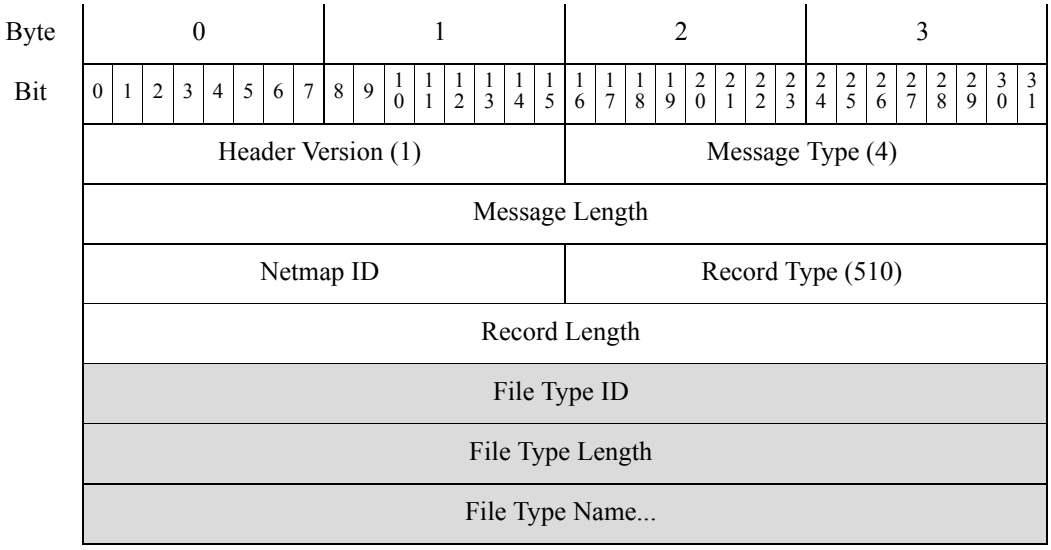
Field	Data Type	Description
File Event SHA Hash Block Type	uint32	Initiates a File Event SHA Hash block. This value is always `40`.
File Event SHA Hash Block Length	uint32	Total number of bytes in the File Event SHA Hash block, including eight bytes for the File Event SHA Hash block type and length fields, plus the number of bytes of data that follows.
SHA Hash	uint8[32]	The SHA-256 hash of the file in binary format.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the file. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
File Name or Disposition	string	The descriptive name or disposition of the file. If the file is clean, this value is `Clean`. If the file's disposition is unknown, the value is `Neutral`. If the file contains malware, the file name is given.

Table 3-47 File Event SHA Hash Data Block Fields (continued)

Field	Data Type	Description
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the AMP cloud for a disposition, or the AMP cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user
User Defined	uint8	Indicated how the file name was provided: • 0 — Defined by AMP • 1 — User defined

File Type ID Metadata for 5.3+

The eStreamer service transmits metadata containing file type information for an event with a file type id, the format of which is shown below. This record maps a file type id to a file type name. File type ID information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 510, indicating a file type id record.



The following table describes the fields in the File Type ID record.

Table 3-48 File Type ID Record Fields

Field	Data Type	Description
File Type ID	uint32	File Type ID number.
File Type Length	uint32	The number of bytes included in the file type name.
File Type Name	string	The descriptive name for the file type.

Rule Documentation Data Block for 5.2+

The eStreamer service uses the Rule Documentation data block to contain information about rules used to generate alerts. The block type is 27 in the series 2 set of data blocks. It can be requested with a host request message of type 10. See Host Request Message Format, page 2-25 for more information.

The following diagram shows the structure of a rule documentation data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Rule Documentation Block Type (27)																															
	Rule Documentation Block Length																															
	Signature ID																															
	Generator ID																															
	Revision																															
Summary	String Block Type (0)																															
	String Block Length																															
	Summary...																															
Impact	String Block Type (0)																															
	String Block Length																															
	Impact...																															
Detailed Info	String Block Type (0)																															
	String Block Length																															
	Detailed Information																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Affected Systems	String Block Type (0)																															
	String Block Length																															
	Affected Systems...																															
Attack Scenarios	String Block Type (0)																															
	String Block Length																															
	Attack Scenarios...																															
Ease of Attack	String Block Type (0)																															
	String Block Length																															
	Ease of Attack...																															
False Positives	String Block Type (0)																															
	String Block Length																															
	False Positives...																															
False Negatives	String Block Type (0)																															
	String Block Length																															
	False Negatives...																															
Corrective Action	String Block Type (0)																															
	String Block Length																															
	Corrective Action...																															
Contributors	String Block Type (0)																															
	String Block Length																															
	Contributors...																															
Additional References	String Block Type (0)																															
	String Block Length																															
	Additional References...																															

The following table describes the fields in the rule documentation data block.

Table 3-49 Rule Documentation Data Block Fields

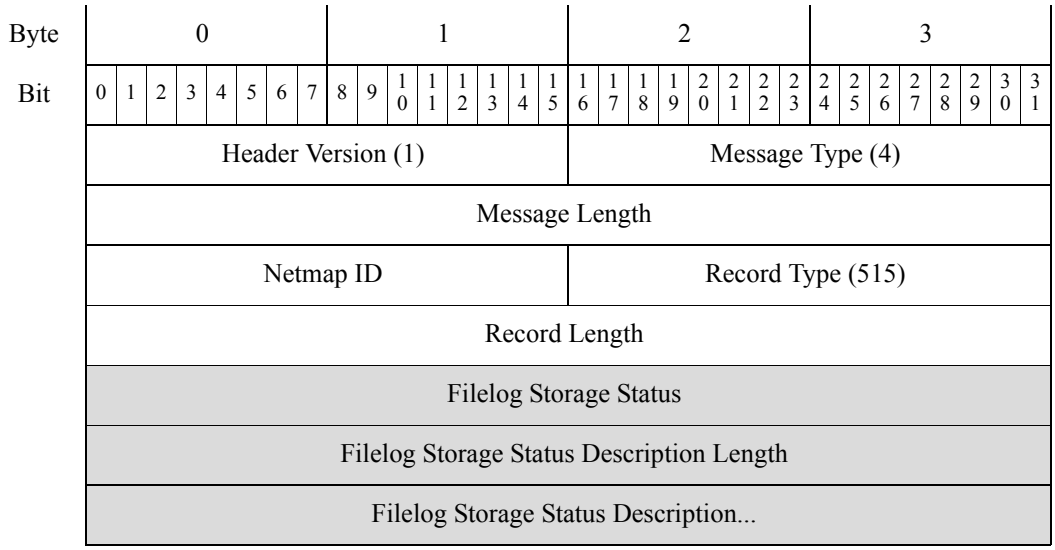
Field	Data Type	Description
Rule Documentation Data Block Type	uint32	Initiates a Rule Documentation data block. This value is always 27.
Rule Documentation Data Block Length	uint32	Total number of bytes in the Rule Documentation data block, including eight bytes for the Rule Documentation data block type and length fields, plus the number of bytes of data that follows.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
String Block Type	uint32	Initiates a String data block containing the summary associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Summary field.
Summary	string	Explanation of the threat or vulnerability.
String Block Type	uint32	Initiates a String data block containing the impact associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Impact field.
Impact	string	How a compromise that uses this vulnerability may impact various systems.
String Block Type	uint32	Initiates a String data block containing the detailed information associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detailed Information field.
Detailed Information	string	Information regarding the underlying vulnerability, what the rule actually looks for, and what systems are affected.
String Block Type	uint32	Initiates a String data block containing the list of affected systems associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Affected Systems field.
Affected Systems	string	Systems affected by the vulnerability.
String Block Type	uint32	Initiates a String data block containing the possible attack scenarios associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Attack Scenarios field.

Table 3-49 Rule Documentation Data Block Fields (continued)

Field	Data Type	Description
Attack Scenarios	string	Examples of possible attacks.
String Block Type	uint32	Initiates a String data block containing the ease of attack associated with the rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Ease of Attack field.
Ease of Attack	string	Whether the attack is considered simple, medium, hard, or difficult, and whether or not is can be performed using a script.
String Block Type	uint32	Initiates a String data block containing the possible false positives associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the False Positives field.
False Positives	string	Examples that may result in a false positive. The default value is `None Known`.
String Block Type	uint32	Initiates a String data block containing the possible false negatives associated with the rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the False Negatives field.
False Negatives	string	Examples that may result in a false negative. The default value is `None Known`.
String Block Type	uint32	Initiates a String data block containing the corrective action associated with the rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Corrective Action field.
Corrective Action	string	Information regarding patches, upgrades, or other means to remove or mitigate the vulnerability.
String Block Type	uint32	Initiates a String data block containing the contributors for the rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Contributors field.
Contributors	string	Contact information for the author of the rule and other relevant documentation.
String Block Type	uint32	Initiates a String data block containing the additional references associated with the rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Additional References field.
Additional References	string	Additional information and references.

Filelog Storage Metadata for 6.0+

The eStreamer service transmits metadata containing filelog storage information. Note that the Record Type field, which appears after the Message Length field, has a value of `515`, indicating a Filelog Storage Metadata record.



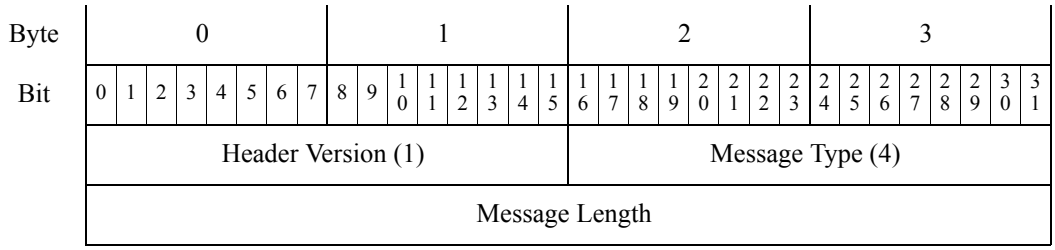
The following table describes the fields in the Filelog Storage Metadata record.

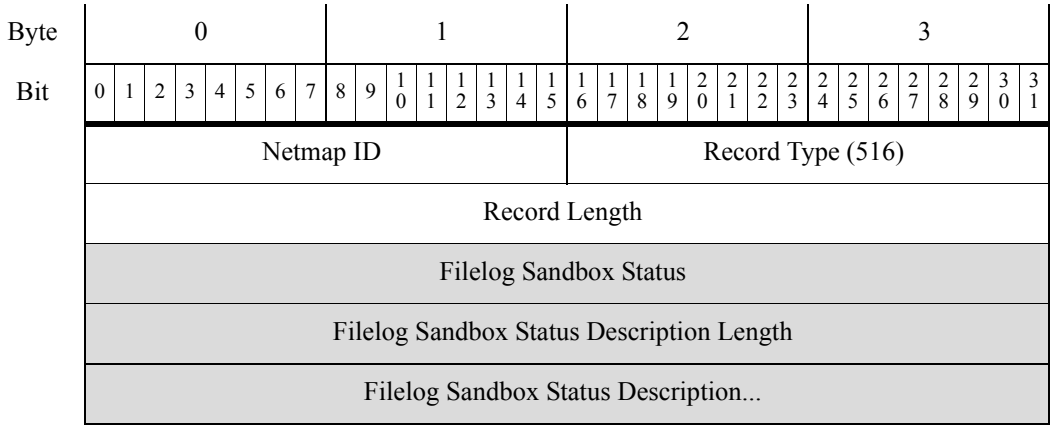
Table 3-50 Filelog Storage Metadata Record Fields

Field	Data Type	Description
Filelog Storage Status	uint32	Number denoting the filelog storage status
Filelog Storage Status Description Length	uint32	The number of bytes included in the Filelog Storage Status Description.
Filelog Storage Status Description	string	The descriptive name for the filelog storage status.

Filelog Sandbox Metadata for 6.0+

The eStreamer service transmits metadata containing filelog sandbox information. Note that the Record Type field, which appears after the Message Length field, has a value of `516`, indicating a Filelog Sandbox Metadata record.





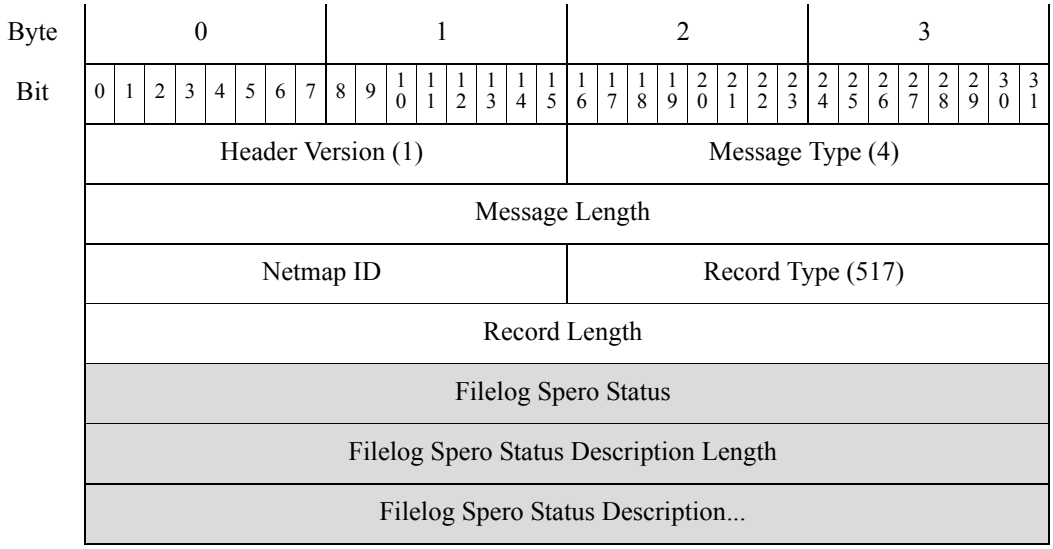
The following table describes the fields in the Filelog Sandbox Metadata record.

Table 3-51 Filelog Sandbox Metadata Record Fields

Field	Data Type	Description
Filelog Sandbox Status	uint32	Number denoting the filelog sandbox status
Filelog Sandbox Status Description Length	uint32	The number of bytes included in the Filelog Sandbox Status Description.
Filelog Sandbox Status Description	string	The descriptive name for the filelog sandbox status.

Filelog Spero Metadata for 6.0+

The eStreamer service transmits metadata containing filelog spero information. Note that the Record Type field, which appears after the Message Length field, has a value of `517`, indicating a filelog spero metadata record.



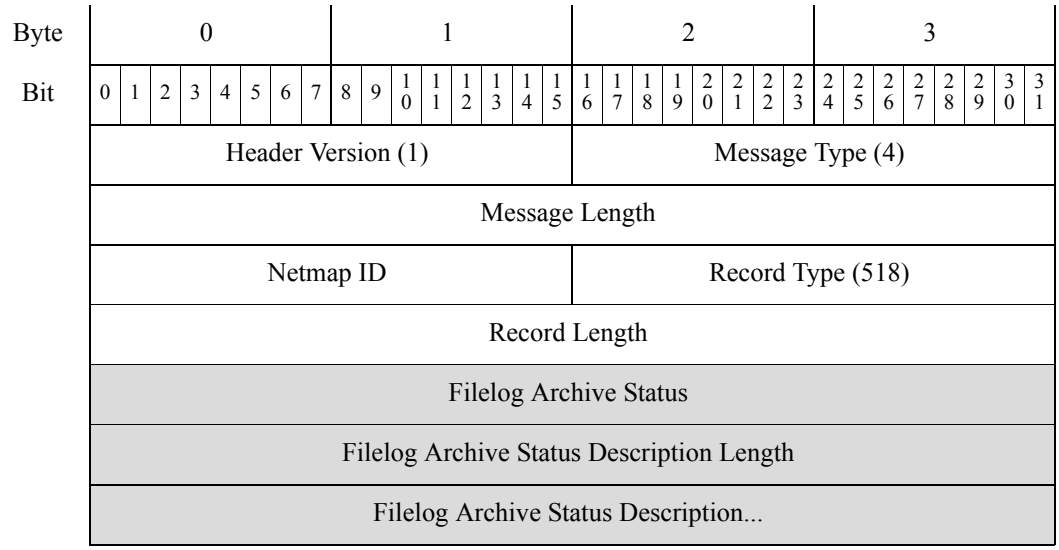
The following table describes the fields in the Filelog Spero Metadata record.

Table 3-52 Filelog Spero Metadata Record Fields

Field	Data Type	Description
Filelog Spero Status	uint32	Number denoting the filelog spero status
Filelog Spero Status Description Length	uint32	The number of bytes included in the Filelog Spero Status Description.
Filelog Spero Status Description	string	The descriptive name for the filelog spero status.

Filelog Archive Metadata for 6.0+

The eStreamer service transmits metadata containing filelog archive information. Note that the Record Type field, which appears after the Message Length field, has a value of `518`, indicating a Filelog Archive Metadata record.



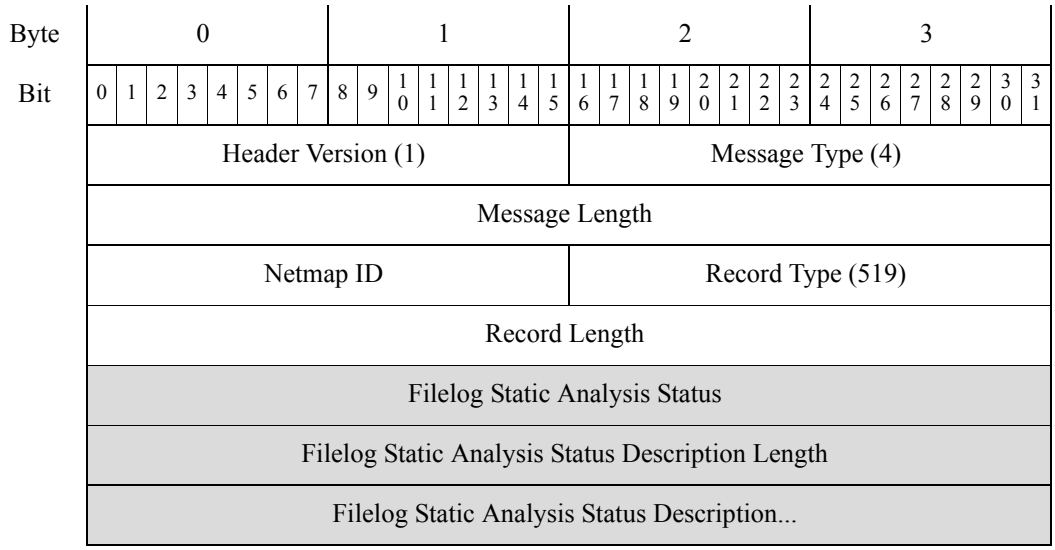
The following table describes the fields in the Filelog Archive Metadata record.

Table 3-53 Filelog Archive Metadata Record Fields

Field	Data Type	Description
Filelog Archive Status	uint32	Number denoting the filelog archive status
Filelog Archive Status Description Length	uint32	The number of bytes included in the Filelog Archive Status Description.
Filelog Archive Status Description	string	The descriptive name for the filelog archive status.

Filelog Static Analysis Metadata for 6.0+

The eStreamer service transmits metadata containing filelog static analysis information. Note that the Record Type field, which appears after the Message Length field, has a value of `519`, indicating a Filelog Static Analysis Metadata record.



The following table describes the fields in the Filelog Static Analysis Metadata record.

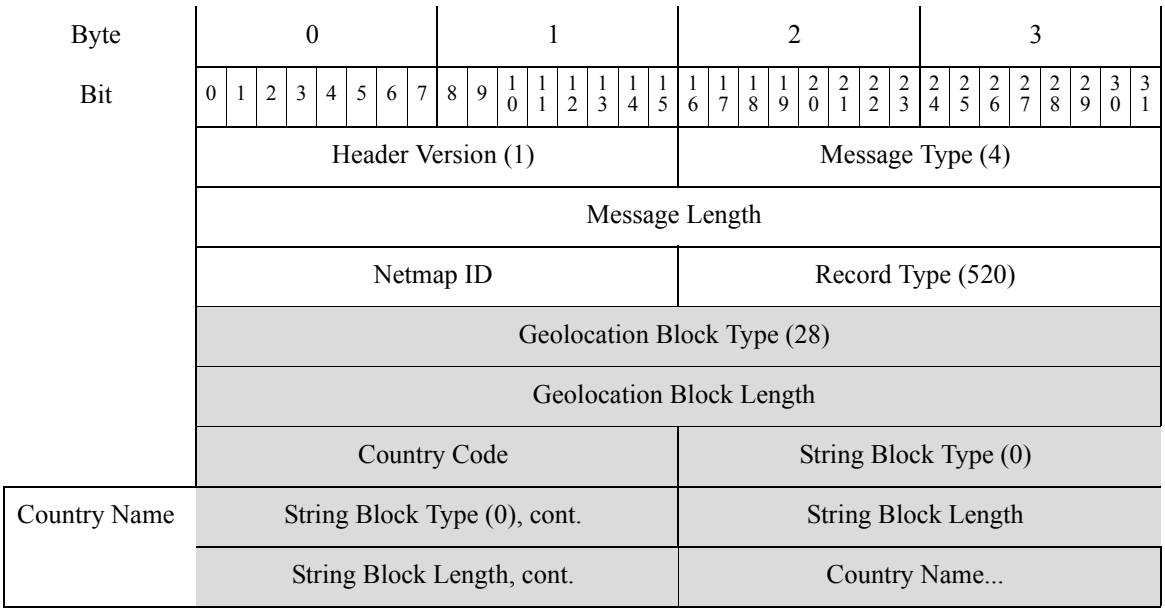
Table 3-54 Filelog Static Analysis Metadata Record Fields

Field	Data Type	Description
Filelog Static Analysis Status	uint32	Number denoting the filelog static analysis status
Filelog Static Analysis Status Description Length	uint32	The number of bytes included in the Filelog Static Analysis Status Description.
Filelog Static Analysis Status Description	string	The descriptive name for the filelog static analysis status.

Geolocation Data Block for 5.2+

This is a data block that contains the mapping of a country code to a country name. The record type is 520, and a block type of 28 in series 2. It is exposed as metadata for any event that has geolocation information. If metadata is requested and there is a value for the country code(s) in the event, then this block is returned along with other metadata.

The following diagram shows the structure of a geolocation data block:



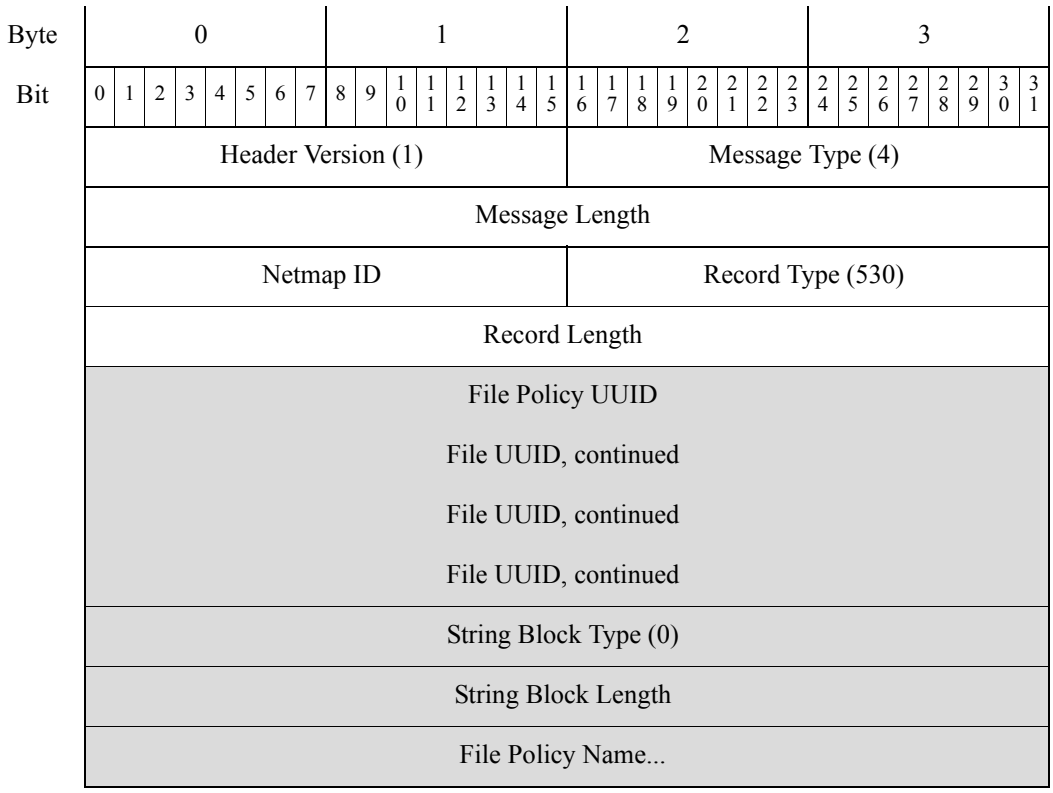
The following table describes the fields in the Geolocation data block.

Table 3-55 Geolocation Data Block Fields

Field	Data Type	Description
Geolocation Data Block Type	uint32	Initiates a Geolocation data block. This value is always `28`.
Geolocation Data Block Length	uint32	Total number of bytes in the Geolocation data block, including eight bytes for the Geolocation data block type and length fields, plus the number of bytes of data that follows.
Country Code	uint16	The country code.
String Block Type	uint32	Initiates a String data block containing the country name associated with the country code. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Country Name field.
Country Name	string	The name of the country associated with the country code.

File Policy Name for 6.0+

The eStreamer service transmits metadata containing File Policy Name information, the format of which is shown below. (File Policy Name information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `530`, indicating a File Policy Name record.



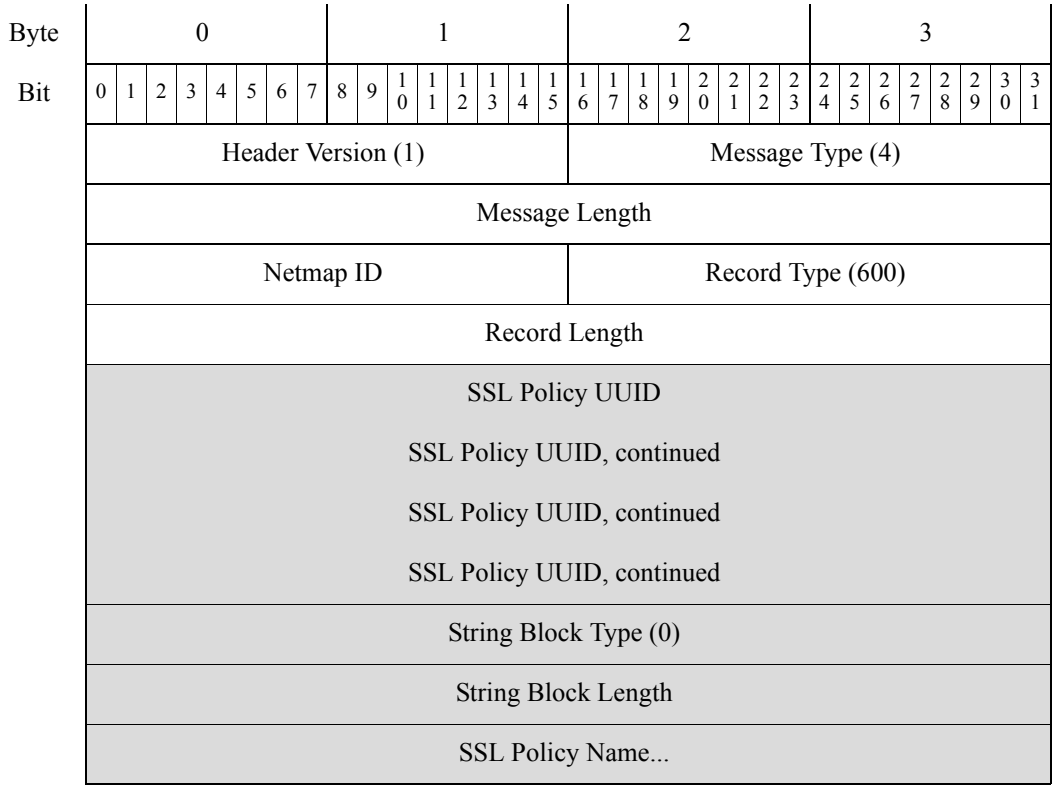
The following table describes the fields in the File Policy Name record.

Table 3-56 File Policy Name Fields

Field	Data Type	Description
File Policy UUID	uint8[16]	The UUID of the File Policy
String Block Type	uint32	Initiates a String data block containing the name of the File Policy. This value is always `0`.
String Block Length	uint32	The number of bytes included in the SSL Policy Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Policy name.
File Policy Name	string	The name of the File Policy.

SSL Policy Name

The eStreamer service transmits metadata containing SSL Policy Name information, the format of which is shown below. (SSL Policy Name information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `600`, indicating a SSL Policy Name record.



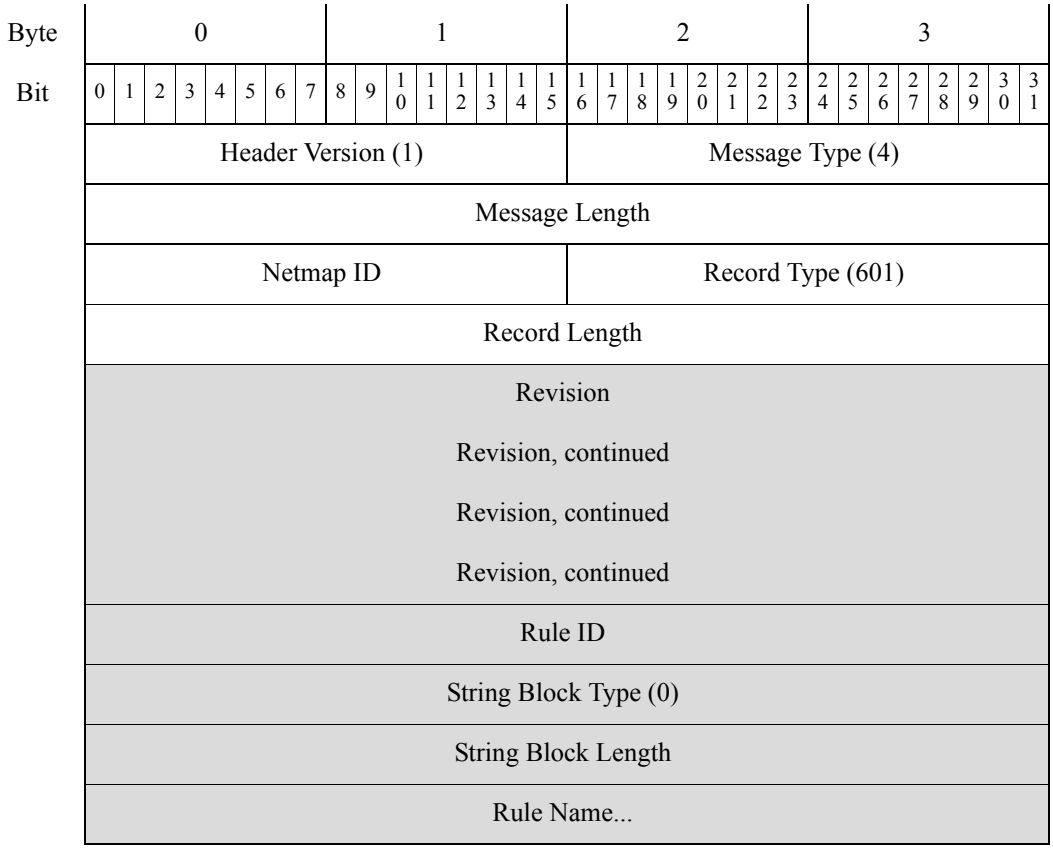
The following table describes the fields in the SSL Policy Name record.

Table 3-57 SSL Policy Name Record Fields

Field	Data Type	Description
SSL Policy UUID	uint8[16]	The UUID of the SSL Policy
String Block Type	uint32	Initiates a String data block containing the name of the SSL Policy. This value is always `0`.
String Block Length	uint32	The number of bytes included in the SSL Policy Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the SSL Policy name.
SSL Policy Name	string	The name of the SSL Policy.

SSL Rule ID

The eStreamer service transmits metadata containing SSL Rule ID information, the format of which is shown below. (SSL Rule ID information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `601`, indicating a SSL Rule ID record.



The following table describes the fields in the SSL Rule ID record.

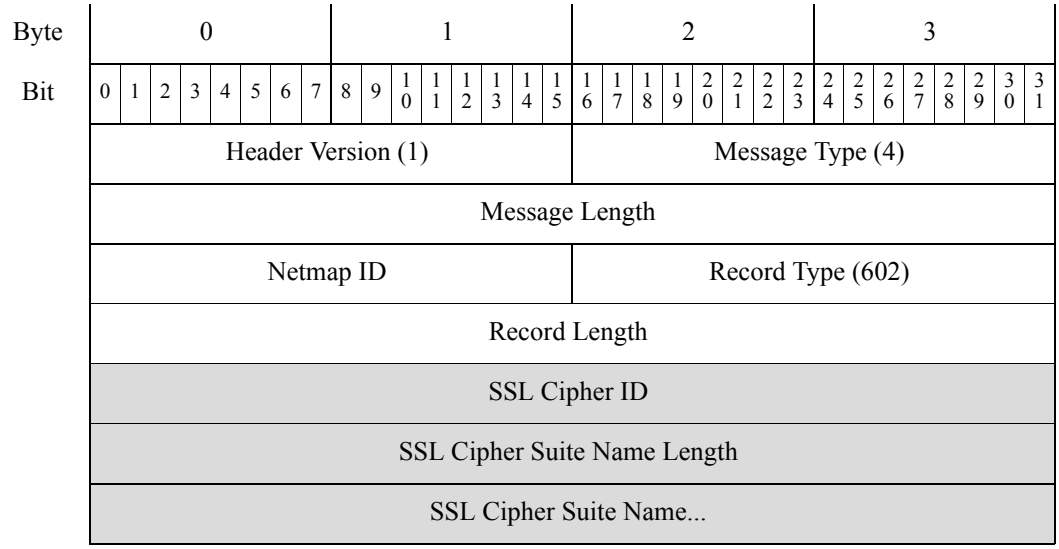
Table 3-58 SSL Policy Name Record Fields

Field	Data Type	Description
Revision	uint8[16]	The UUID of the SSL Rule Revision
Rule ID	uint32	ID number of the SSL Rule
String Block Type	uint32	Initiates a String data block containing the name of the SSL Rule. This value is always `0`.
String Block Length	uint32	The number of bytes included in the SSL Rule Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the SSL Rule Name.
SSL Rule Name	string	The name of the SSL Rule.

SSL Cipher Suite

The eStreamer service transmits metadata containing SSL Cipher Suite information for an event with a SSL Cipher id, the format of which is shown below. This record maps a SSL Cipher id to a SSL Cipher Suite name. SSL Cipher Suite information is sent when one of the metadata flags—bits 1, 14, 15, or 20

in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `602`, indicating a SSL Cipher Suite record.



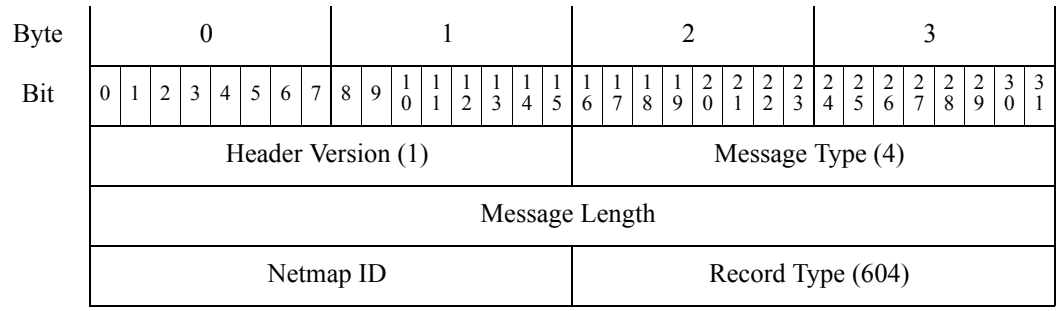
The following table describes the fields in the SSL Cipher Suite record.

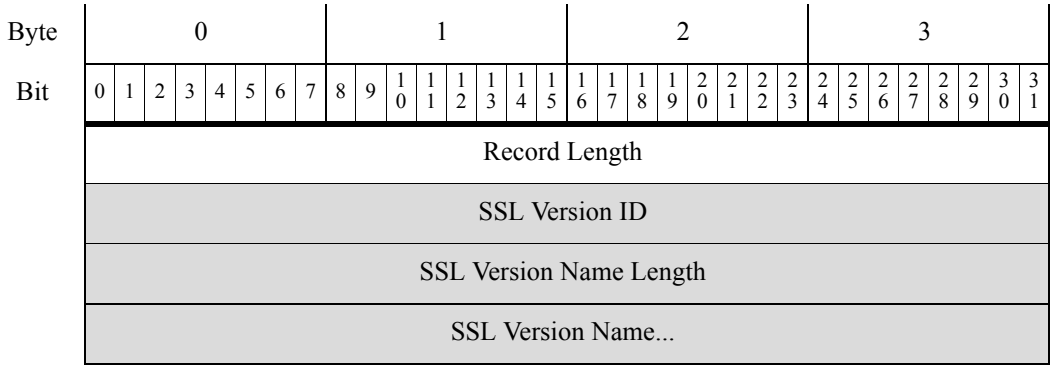
Table 3-59 SSL Cipher Suite Fields

Field	Data Type	Description
SSL Cipher ID	uint32	SSL Cipher ID number.
SSL Cipher Suite Name Length	uint32	The number of bytes included in the SSL cipher suite name.
SSL Cipher Suite Name	string	The descriptive name for the SSL Cipher Suite.

SSL Version

The eStreamer service transmits metadata containing SSL Version information for an event with a SSL Version, the format of which is shown below. This record maps a SSL Version ID to a SSL Version name. SSL Cipher Suite information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `604`, indicating a SSL Version record.





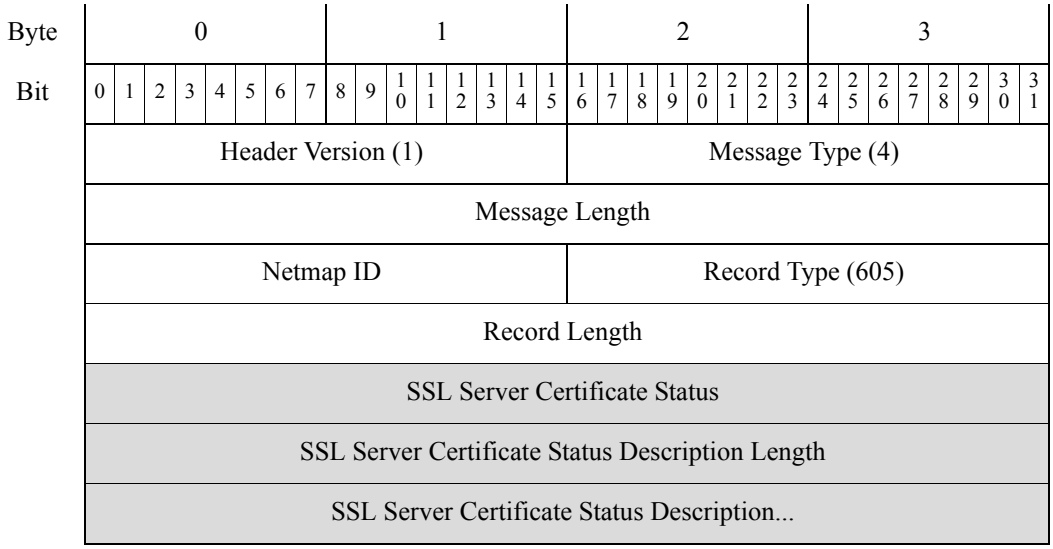
The following table describes the fields in the SSL Version record.

Table 3-60 SSL Version Fields

Field	Data Type	Description
SSL Version ID	uint32	SSL Version ID number.
SSL Version Name	uint32	The number of bytes included in the SSL Version Name.
SSL Cipher Suite Name	string	The descriptive name for the SSL Version.

SSL Server Certificate Status

The eStreamer service transmits metadata containing SSL Server Certificate Status information, the format of which is shown below. (SSL Server Certificate Status information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `605`, indicating a SSL Server Certificate Status record.



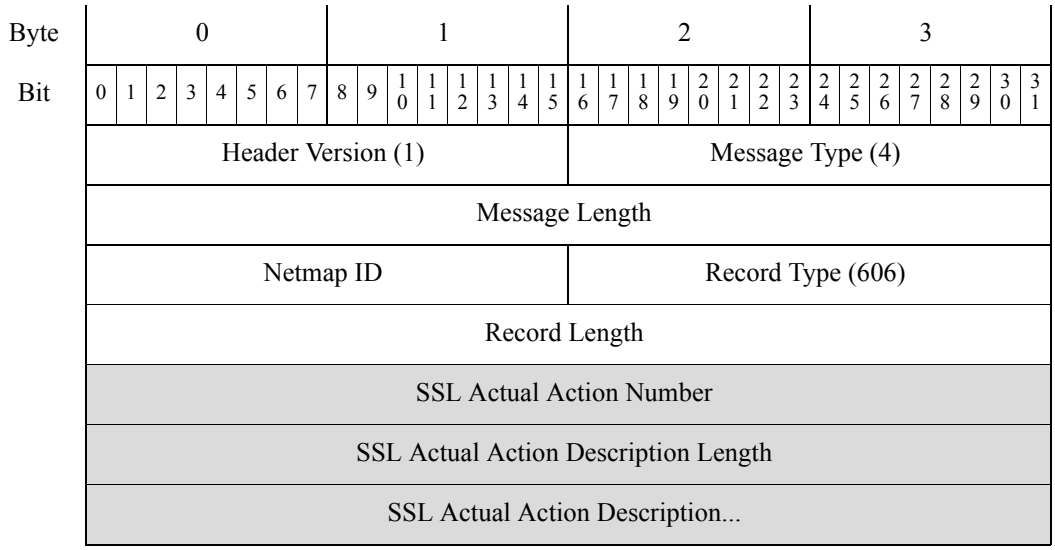
The following table describes the fields in the SSL Server Certificate Status record.

Table 3-61 SSL Server Certificate Status Record Fields

Field	Data Type	Description
SSL Server Certificate Status	uint32	The SSL Server Certificate Status Number
SSL Server Certificate Status Description Lenth	uint32	The number of bytes included in the SSL Server Certificate Status Description.
SSL Server Certificate Status Description	string	The description of the SSL Server Certificate Status.

SSL Actual Action

The eStreamer service transmits metadata containing SSL Actual Action information, the format of which is shown below. (SSL Actual Action information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `606`, indicating a SSL Actual Action record.



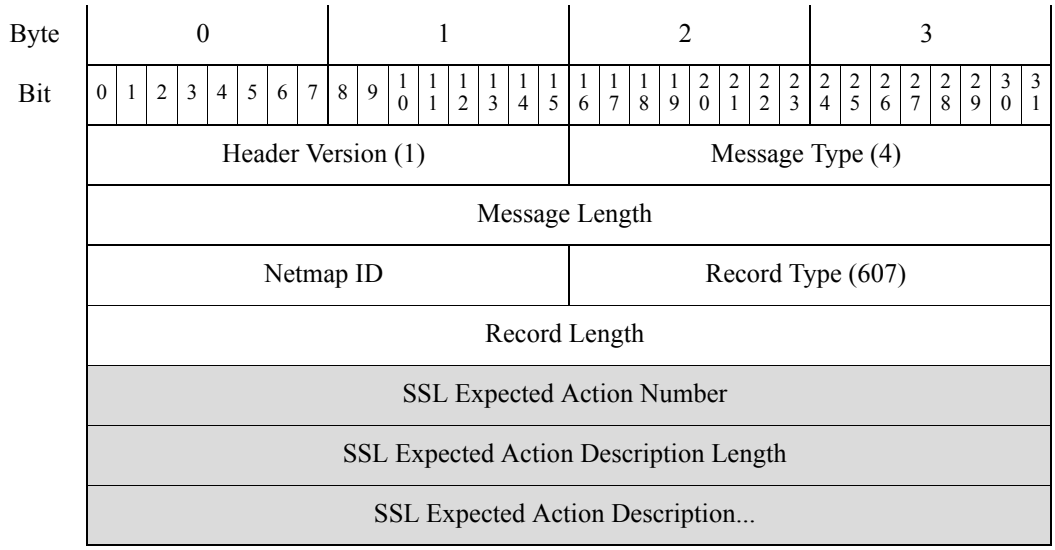
The following table describes the fields in the SSL Actual Action record.

Table 3-62 SSL Actual Action Fields

Field	Data Type	Description
SSL Actual Action Number	uint32	The number designating the SSL Actual Action
SSL Actual Action Description Length	uint32	The number of bytes included in the SSL Actual Action Description.
SSL Actual Action Description	string	The description of the SSL Actual Action.

SSL Expected Action

The eStreamer service transmits metadata containing SSL Expected Action information, the format of which is shown below. (SSL Expected Action information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `607`, indicating a SSL Expected Action record.



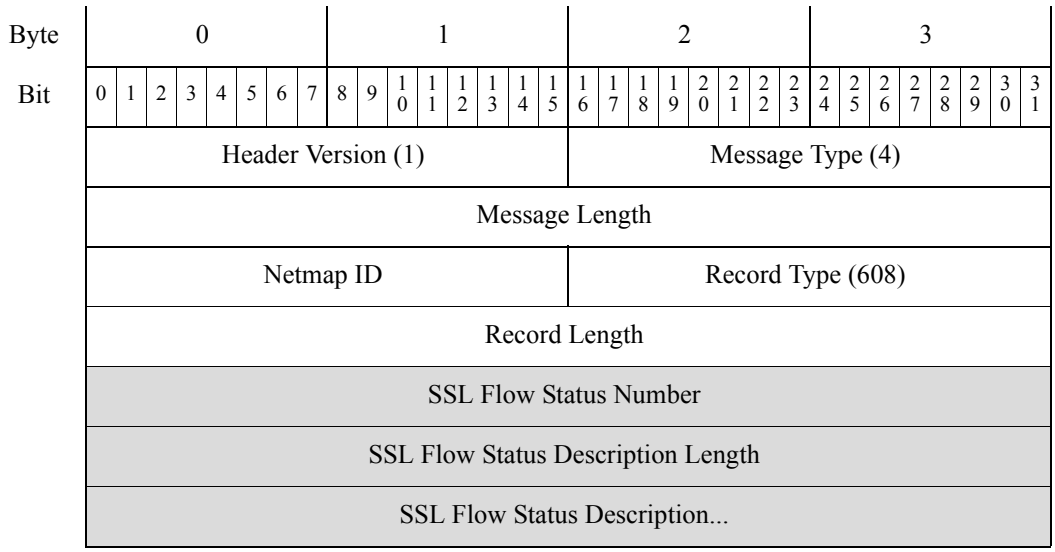
The following table describes the fields in the SSL Expected Action record.

Table 3-63 SSL Actual Action Fields

Field	Data Type	Description
SSL Expected Action Number	uint32	The number designating the SSL Expected Action
SSL Expected Action Description Length	uint32	The number of bytes included in the SSL Expected Action Description.
SSL Expected Action Description	string	The description of the SSL Expected Action.

SSL Flow Status

The eStreamer service transmits metadata containing SSL Flow Status information, the format of which is shown below. (SSL Flow Status information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `608`, indicating a SSL Flow Status record.



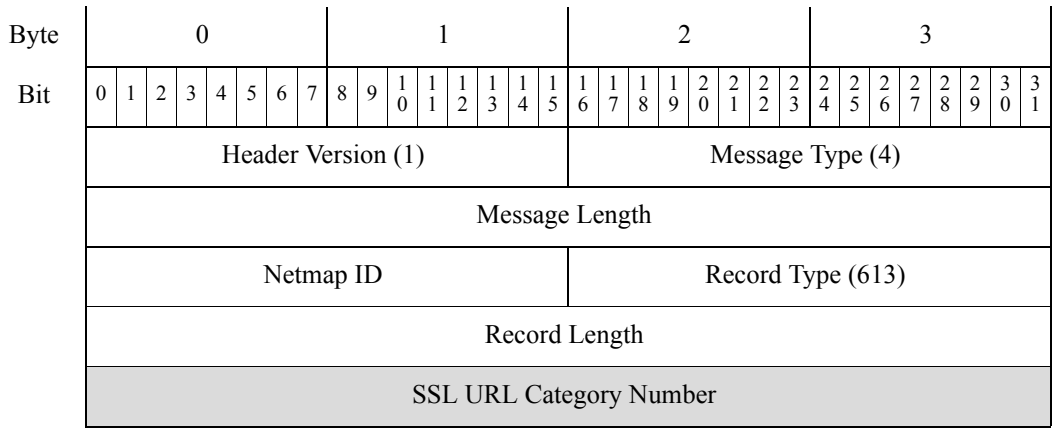
The following table describes the fields in the SSL Flow Status record.

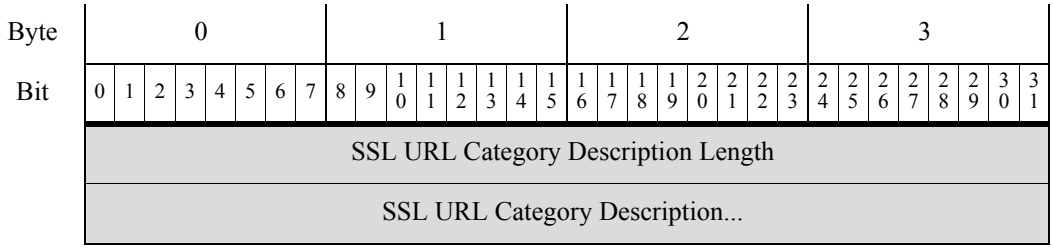
Table 3-64 SSL Flow Status Fields

Field	Data Type	Description
SSL Flow Status Number	uint32	The number designating the SSL Flow Status
SSL Flow Status Description Length	uint32	The number of bytes included in the SSL Flow Status Description.
SSL Flow Status Description	string	The description of the SSL Flow Status.

SSL URL Category

The eStreamer service transmits metadata containing SSL URL Category information, the format of which is shown below. (SSL URL Category information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `613`, indicating a SSL URL Category record.





The following table describes the fields in the SSL URL Category record.

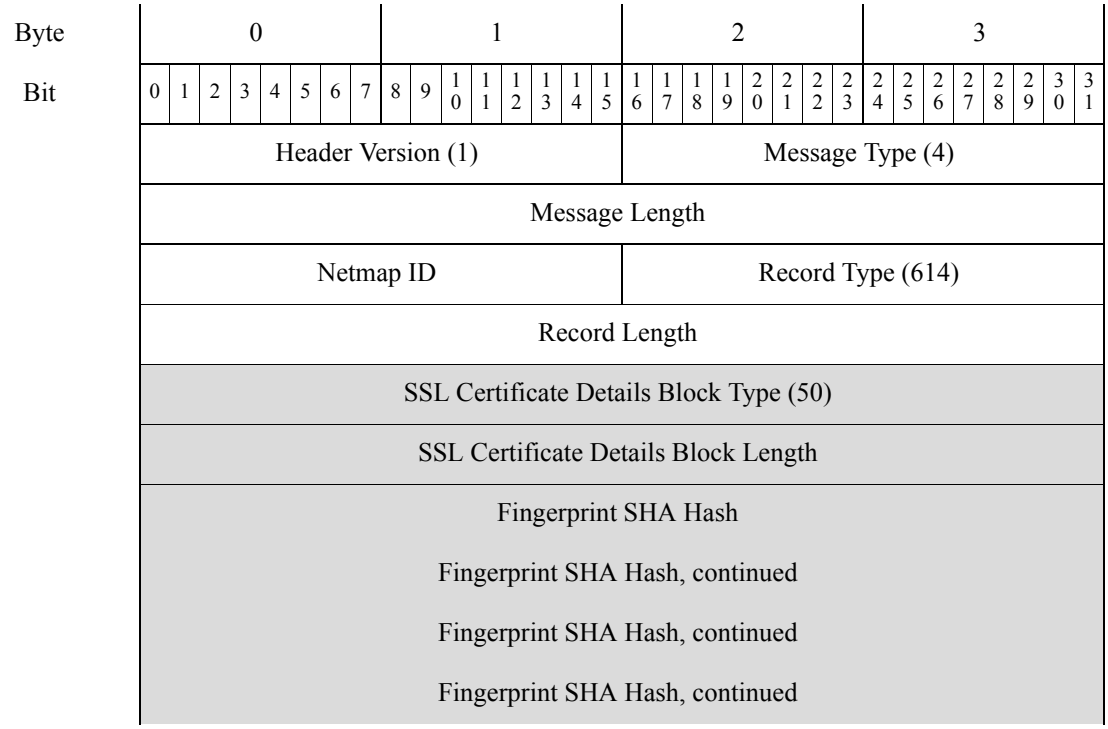
Table 3-65 SSL URL Category Fields

Field	Data Type	Description
SSL URL Category Number	uint32	The number designating the SSL URL Category
SSL URL Category Description Length	uint32	The number of bytes included in the SSL Server URL Category Description.
SSL URL Category Description	string	The description of the SSL URL Category.

SSL Certificate Details Data Block for 5.4+

This is a data block that provides detailed information regarding an SSL certificate. The record type is 614, with a block type of 50 in series 2. It is exposed as metadata for any event that has SSL information. These include malware events, file events, intrusion events, connection events, and correlation events.

The following diagram shows the structure of an SSL Certificate Details data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Fingerprint SHA Hash, continued																															
	Public Key SHA Hash Public Key SHA Hash, continued Public Key SHA Hash, continued Public Key SHA Hash, continued Public Key SHA Hash, continued																															
	Serial Number Serial Number, continued Serial Number, continued Serial Number, continued Serial Number, continued																															
	Serial Number Length																															
Subject Common Name	String Block Type (0)																															
	String Block Length																															
	Subject Common Name...																															
Subject Organization	String Block Type (0)																															
	String Block Length																															
	Subject Organization...																															
Subject Organizational Unit	String Block Type (0)																															
	String Block Length																															
	Subject Organizational Unit....																															
Subject Country	String Block Type (0)																															
	String Block Length																															
	Subject Country...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Issuer Common Name	String Block Type (0)																															
	String Block Length																															
	Issuer Common Name...																															
Issuer Organization	String Block Type (0)																															
	String Block Length																															
	Issuer Organization...																															
Issuer Organizational Unit	String Block Type (0)																															
	String Block Length																															
	Issuer Organizational Unit...																															
Issuer Country	String Block Type (0)																															
	String Block Length																															
	Issuer Country...																															
	Valid Start Date																															
	Valid End Date																															

The following table describes the fields in the SSL Certificate Details data block.

Table 3-66 SSL Certificate Details Data Block Fields

Field	Data Type	Description
SSL Certificate Details Data Block Type	uint32	Initiates an SSL Certificate Details data block. This value is always `50`.
SSL Certificate Details Data Block Length	uint32	Total number of bytes in the SSL Certificate Details data block, including eight bytes for the SSL Certificate Details data block type and length fields, plus the number of bytes of data that follows.
Fingerprint SHA Hash	uint8[20]	SHA1 hash of the SSL Server certificate.
Public Key SHA Hash	uint8[20]	The SHA hash value used to authenticate the public key contained within the certificate.
Serial Number	uint8[20]	The serial number assigned by the issuing CA. While this number cannot exceed 20 bytes in length, it can be less than 20 bytes as specified in the Serial Number Length field.

Table 3-66 SSL Certificate Details Data Block Fields (continued)

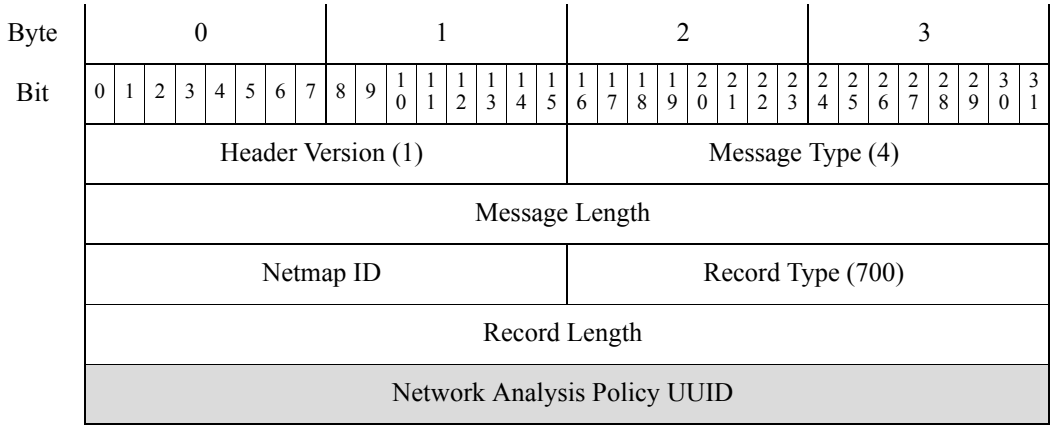
Field	Data Type	Description
Serial Number Length	uint32	The length of the serial number in bytes.
String Block Type	uint32	Initiates a String data block containing the category associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category field.
Subject Common Name	string	Subject Common name from the SSL Certificate This is typically the host and domain name of the certificate subject, but may contain other information.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Subject Organization	string	The organization of the certificate subject.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Subject Organizational Unit	string	The organizational unit of the certificate subject.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Subject Country	string	The country of the certificate subject.
String Block Type	uint32	Initiates a String data block containing the category associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category field.
Issuer Common Name	string	Issuer Common name from the SSL Certificate This is typically the host and domain name of the certificate issuer, but may contain other information.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.

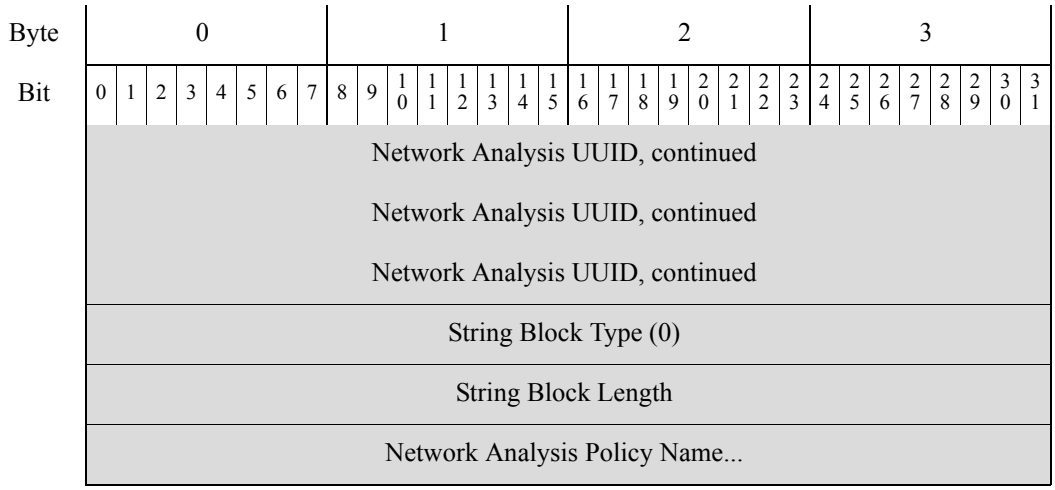
Table 3-66 SSL Certificate Details Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Issuer Organization	string	The organization of the certificate issuer.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Issuer Organizational Unit	string	The organizational unit of the certificate issuer.
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Issuer Country	string	The country of the certificate issuer.
Valid Start Date	uint32	The Unix timestamp when the certificate was issued.
Valid End Date	uint32	The Unix timestamp on which the certificate ceases to be valid.

Network Analysis Policy Name Record

The eStreamer service transmits metadata containing Network Analysis Policy Name information, the format of which is shown below. (Network Analysis Policy Name information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 700, indicating a Network Analysis Policy Name record.





The following table describes the fields in the Network Analysis Policy Name record.

Table 3-67 Network Analysis Policy Name Record Fields

Field	Data Type	Description
Network Analysis Policy UUID	uint8[16]	The UUID of the Network Analysis Policy
String Block Type	uint32	Initiates a String data block containing the name of the Network Analysis Policy. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Network Analysis Policy Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Network Analysis Policy name.
Network Analysis Policy Name	string	The name of the Network Analysis Policy.

Understanding Discovery & Connection Data Structures

This chapter provides details about the data structures used in eStreamer messages for discovery and connection events, as well as the metadata for those events. Discovery and connection event messages use the same general message format and series of data blocks; the differences are in the contents of data blocks themselves.

Discovery events include two sub-categories of events:

- Host discovery events, which identify new and changed hosts on your managed network, including the applications running on the hosts detected from the contents of the packets, and the host vulnerabilities.
- User events, which report the detection of new users and user activity, such as logins.

Connection events report information about the session traffic between your monitored hosts and all other hosts. Connection information includes the first and last packet of the transaction, source and destination IP address, source and destination port, and the number of packets and bytes sent and received. If applicable, connection events also report the client application and URL involved in the session.

For information about requesting discovery or connection events from the eStreamer server, see Request Flags, page 2-11.

For information about the general structure of eStreamer event data messages, see Understanding the Organization of Event Data Messages, page 2-17.

See the following sections in this chapter for more information about discovery and connection event data structures:

- Discovery and Connection Event Data Messages, page 4-2 provides a high-level view of the structure that eStreamer uses for host discovery, user, and connection messages.
- Discovery and Connection Event Record Types, page 4-2 describes the record types for discovery and connection events.
- Metadata for Discovery Events, page 4-6 describes the metadata records that you can request for context information to convert numeric and coded data to text; for example, convert the user ID in an event to a user name.
- Discovery Event Header 5.2+, page 4-34 describes the structure of the standard event header used in all discovery and connection messages, and the values that can occur in the event type and event subtype fields. The event type and subtype fields further define the structure of the data record carried in the message.

- Host Discovery Structures by Event Type, page 4-38 describes the structure of the data record that eStreamer uses for the various host discovery event types.
- User Data Structures by Event Type, page 4-54 describes the structure of the data record that eStreamer uses for the various user event types.
- Understanding Discovery (Series 1) Blocks, page 4-56 describes the series of data block structures that are used to convey complex records in discovery and connection event messages. Series 1 data blocks also appear in correlation events.
- User Vulnerability Data Block 5.0+, page 4-147 describes other series 1 block structures that are used to convey complex user event records.

Tip See "Data Structure Examples" section on page A-1 for examples that illustrate sample discovery events.

Discovery and Connection Event Data Messages

eStreamer packages the data for discovery and connection events in the same message structure, which contains:

- An option netmap ID
- a record header that defines the record type
- a discovery event header that identifies and characterizes the event, and specifically identifies the event type and subtype. For information, see Discovery Event Header 5.2+, page 4-34.
- a data record consisting of a block header and a data block. Discovery and connection event data messages use series 1 data blocks. For information, see Host Discovery and Connection Data Blocks, page 4-56 or User Vulnerability Data Block 5.0+, page 4-147.

Discovery and Connection Event Record Types

The following table lists the event record types for host discovery and connection events, and provides links to the event message structure for each record type. The list includes metadata record types as well. Some records contain a single data block which stores a specific piece of data. These data blocks are broken up into series 1 blocks that contain most types of data, and series 2 blocks that specifically contain discovery data. The table also indicates the status of each version (current or legacy). A current record is the latest version. A legacy record has been superseded by a later version but can still be requested from eStreamer.

Table 4-1 Discovery and Connection Event Record Types

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
10	139	1	New Host Detected	Current	New Host and Host Last Seen Messages, page 4-39
11	103	1	New TCP Server	Current	Server Messages, page 4-40
12	103	1	New UDP Server	Current	Server Messages, page 4-40
13	4	1	New Network Protocol	Current	New Network Protocol Message, page 4-40
14	4	1	New Transport Protocol	Current	New Transport Protocol Message, page 4-41

Table 4-1 Discovery and Connection Event Record Types (continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
15	122	1	New Client Application	Current	Client Application Messages, page 4-41
16	103	1	TCP Server Information Update	Current	Server Messages, page 4-40
17	103	1	UDP Server Information Update	Current	Server Messages, page 4-40
18	53	1	OS Information Update	Current	Operating System Update Messages, page 4-43
19	N/A	N/A	Host Timeout	Current	IP Address Reused and Host Timeout/Deleted Messages, page 4-43
20	N/A	N/A	Host IP Address Reused	Current	IP Address Reused and Host Timeout/Deleted Messages, page 4-43
21	N/A	N/A	Host Deleted: Host Limit Reached	Current	IP Address Reused and Host Timeout/Deleted Messages, page 4-43
22	N/A	N/A	Hops Change	Current	Hops Change Message, page 4-44
23	N/A	N/A	TCP Port Closed	Current	TCP and UDP Port Closed/Timeout Messages, page 4-44
24	N/A	N/A	UDP Port Closed	Current	TCP and UDP Port Closed/Timeout Messages, page 4-44
25	N/A	N/A	TCP Port Timeout	Current	TCP and UDP Port Closed/Timeout Messages, page 4-44
26	N/A	N/A	UDP Port Timeout	Current	TCP and UDP Port Closed/Timeout Messages, page 4-44
27	N/A	N/A	MAC Information Change	Current	MAC Address Messages, page 4-45
28	N/A	N/A	Additional MAC Detected for Host	Current	MAC Address Messages, page 4-45
29	N/A	N/A	Host IP Address Changed	Current	IP Address Change Message, page 4-42
31	N/A	N/A	Host Identified as Router/Bridge	Current	Host Identified as a Bridge/Router Message, page 4-45
34	14	1	VLAN Tag Information Update	Current	VLAN Tag Information Update Messages, page 4-46
35	122	1	Client Application Timeout	Current	Client Application Messages, page 4-41
42	35	1	NetBIOS Name Change	Current	Change NetBIOS Name Message, page 4-46
44	N/A	N/A	Host Dropped: Host Limit Reached	Current	IP Address Reused and Host Timeout/Deleted Messages, page 4-43
45	37	1	Update Banner	Current	Update Banner Message, page 4-47
46	55	1	Add Host Attribute	Current	Attribute Messages, page 4-50
47	55	1	Update Host Attribute	Current	Attribute Messages, page 4-50
48	55	1	Delete Host Attribute	Current	Attribute Messages, page 4-50

Table 4-1 Discovery and Connection Event Record Types (continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
51	103	1	TCP Server Confidence Update	Legacy	Server Messages, page 4-40
52	103	1	UDP Server Confidence Update	Legacy	Server Messages, page 4-40
53	53	1	OS Confidence Update	Legacy	Operating System Update Messages, page 4-43
54	N/A	N/A	Fingerprint Metadata	Current	Fingerprint Record, page 4-7
55	N/A	N/A	Client Application Metadata	Current	Client Application Record, page 4-8
57	N/A	N/A	Vulnerability Metadata	Current	Vulnerability Record, page 4-9
58	N/A	N/A	Criticality Metadata	Current	Criticality Record, page 4-11
	N/A	N/A	Network Protocol Metadata	Current	Network Protocol Record, page 4-12
60	N/A	N/A	Attribute Metadata	Current	Attribute Record, page 4-13
61	N/A	N/A	Scan Type Metadata	Current	Scan Type Record, page 4-13
63	N/A	N/A	Server Metadata	Current	Server Record, page 4-14
71	144	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.2.x, page B-125
71	152	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.3, page B-138
71	154	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.3.1, page B-145
71	155	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.4, page B-152
71	157	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.4.1, page B-165
71	160	1	Connection Statistics	Current	Connection Statistics Data Block 6.0+, page 4-110
73	136	1	Connection Chunks	Current	Connection Chunk Message, page 4-48
74	N/A	N/A	User Set OS	Current	User Server and Operating System Messages, page 4-51
75	N/A	N/A	User Set Server	Current	User Server and Operating System Messages, page 4-51
76	83	1	User Delete Protocol	Current	User Protocol Messages, page 4-52
77	60	1	User Delete Client Application	Current	User Client Application Messages, page 4-52
78	78	1	User Delete Address	Current	User Add and Delete Host Messages, page 4-49
79	77	1	User Delete Server	Current	User Delete Server Message, page 4-49
80	80	1	User Set Valid Vulnerabilities	Current	User Set Vulnerabilities Messages for Version 4.6.1+, page 4-48
81	80	1	User Set Invalid Vulnerabilities	Current	User Set Vulnerabilities Messages for Version 4.6.1+, page 4-48
82	81	1	User Set Host Criticality	Current	User Set Host Criticality Messages, page 4-50

Table 4-1 Discovery and Connection Event Record Types (continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
83	55	1	User Set Attribute Value	Current	Attribute Value Messages, page 4-51
84	82	1	User Delete Attribute Value	Current	Attribute Value Messages, page 4-51
85	78	1	User Add Host	Current	User Add and Delete Host Messages, page 4-49
86	N/A	N/A	User Add Server	Current	User Server and Operating System Messages, page 4-51
87	60	1	User Add Client Application	Current	User Client Application Messages, page 4-52
88	83	1	User Add Protocol	Current	User Protocol Messages, page 4-52
89	142	1	User Add Scan Result	Current	Add Scan Result Messages, page 4-53
90	N/A	N/A	Source Type Record	Current	Source Type Record, page 4-15
91	N/A	N/A	Source Application Record	Current	Source Application Record, page 4-16
92	120	1	User Dropped Change Event	Current	User Modification Messages, page 4-54
93	120	1	User Removed Change Event	Current	User Modification Messages, page 4-54
94	120	1	New User Identification Event	Current	User Modification Messages, page 4-54
95	121	1	User Login Change Event	Current	User Information Update Message Block, page 4-55
96	N/A	N/A	Source Detector Record	Current	Source Detector Record, page 4-17
98	N/A	N/A	User Record	Current	User Record, page 4-19
101	N/A	N/A	New OS Event	Current	New Operating System Messages, page 4-53
102	94	1	Identity Conflict Event	Current	Identity Conflict and Identity Timeout System Messages, page 4-54
103	94	1	Identity Timeout Event	Current	Identity Conflict and Identity Timeout System Messages, page 4-54
106	N/A	N/A	Third Party Scanner Vulnerability Record	Current	Third Party Scanner Vulnerability Record, page 4-17
107	122	1	Client Application Update	Current	Client Application Messages, page 4-41
109	N/A	N/A	Web Application Record	Current	Web Application Record, page 4-20
115	N/A	N/A	Security Zone Name Record	Current	Security Zone Name Record, page 3-29
116	14	2	Interface Name Record	Current	Interface Name Record, page 3-31
117	14	2	Access Control Policy Name Metadata	Current	Access Control Policy Name Record, page 3-32

Table 4-1 Discovery and Connection Event Record Types (continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
118	14	2	Intrusion Policy Name Record	Current	Intrusion Policy Name Record, page 4-21
119	14	2	Access Control Rule ID Record	Current	Access Control Rule ID Record Metadata, page 3-33
120	N/A	N/A	Access Control Rule Action Record	Current	Access Control Rule Action Record Metadata, page 4-22
121	N/A	N/A	URL Category Record	Current	URL Category Record Metadata, page 4-23
122	N/A	N/A	URL Reputation Metadata	Current	URL Reputation Record Metadata, page 4-24
124	21	2	Access Control Rule Reason Metadata	Current	Access Control Rule Reason Metadata, page 4-24
160	150	1	IOC State Data Block for 5.3+	Current	IOC State Data Block for 5.3+, page 4-28
161	39	2	IOC Name Data Block for 5.3+	Current	IOC Name Data Block for 5.3+, page 4-30
280	22	2	Security Intelligence Category Metadata	Current	Security Intelligence Category Metadata, page 4-26
281	N/A	N/A	Security Intelligence Source/Destination Record	Current	Security Intelligence Source/Destination Record, page 4-27

Metadata for Discovery Events

You request metadata by metadata version number. For the metadata version that corresponds to your version of the Firepower System, see Understanding Metadata, page 2-37. For important information on how eStreamer streams metadata records, see Metadata Transmission, page 2-37.

For information on the structures of the various metadata records types for host discovery and user event records, see:

- Fingerprint Record, page 4-7
- Client Application Record, page 4-8
- Vulnerability Record, page 4-9
- Criticality Record, page 4-11
- Network Protocol Record, page 4-12
- Attribute Record, page 4-13
- Scan Type Record, page 4-13
- Server Record, page 4-14
- Source Type Record, page 4-15
- Source Application Record, page 4-16
- Source Detector Record, page 4-17

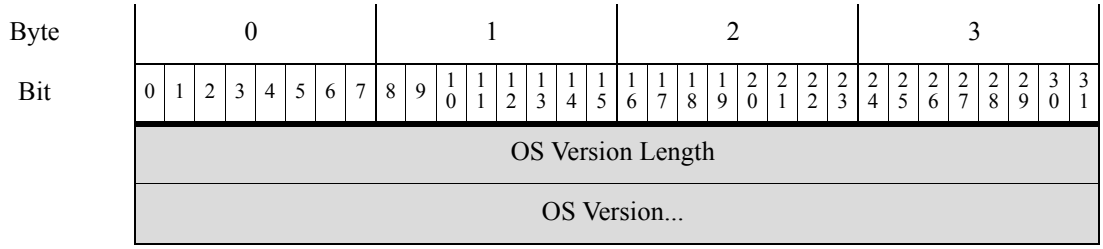
- Third Party Scanner Vulnerability Record, page 4-17
- User Record, page 4-19
- Web Application Record, page 4-20
- Intrusion Policy Name Record, page 4-21
- Access Control Rule Action Record Metadata, page 4-22
- URL Category Record Metadata, page 4-23
- URL Reputation Record Metadata, page 4-24
- Access Control Rule Reason Metadata, page 4-24
- Security Intelligence Category Metadata, page 4-26
- Security Intelligence Source/Destination Record, page 4-27

For metadata records for intrusion and correlation events, see Intrusion Event and Metadata Record Types, page 3-1.

Fingerprint Record

The eStreamer service transmits the fingerprint metadata for an event within a Fingerprint record, the format of which is shown below. (Fingerprint metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `54`, indicating a Fingerprint record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (54)															
	Record Length																															
Fingerprint UUID	Fingerprint UUID																															
	Fingerprint UUID cont.																															
	Fingerprint UUID cont.																															
	Fingerprint UUID cont.																															
	OS Name Length																															
	OS Name...																															
	OS Vendor Length																															
	OS Vendor...																															



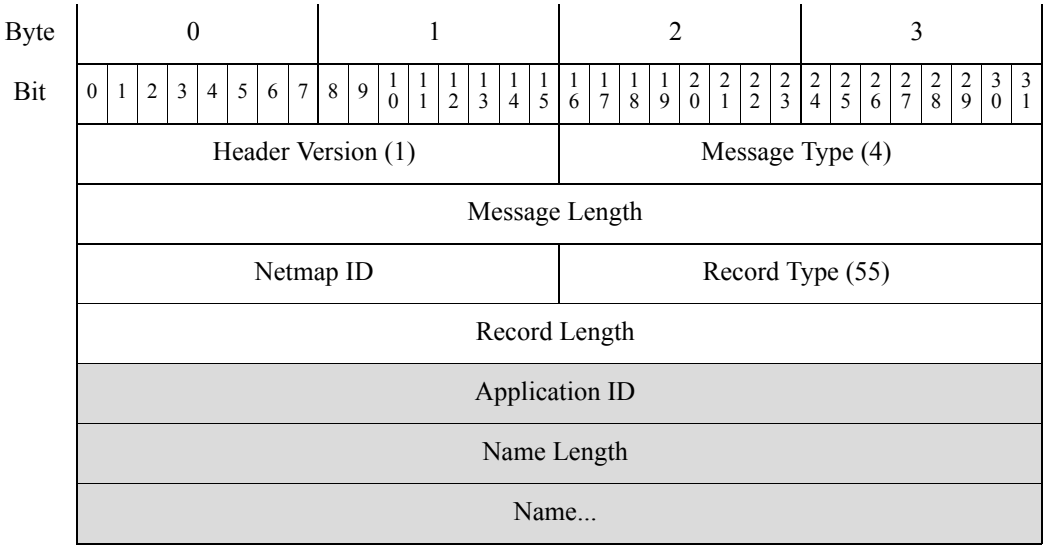
The following table describes the fields in the Fingerprint record.

Table 4-2 Fingerprint Record Fields

Field	Data Type	Description
Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the operating system.
OS Name Length	uint32	The number of bytes included in the operating system name.
OS Name	string	The name of the operating system for the fingerprint.
OS Vendor Length	uint32	The number of bytes included in the operating system vendor name.
OS Vendor	string	The name of the operating system vendor for the fingerprint.
OS Version Length	uint32	The number of bytes included in the operating system version.
OS Version	string	The version of the operating system for the fingerprint.

Client Application Record

The eStreamer service transmits the client application metadata for an event within a Client Application record, the format of which is shown below. (Client application metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `55`, indicating a Client Application record.



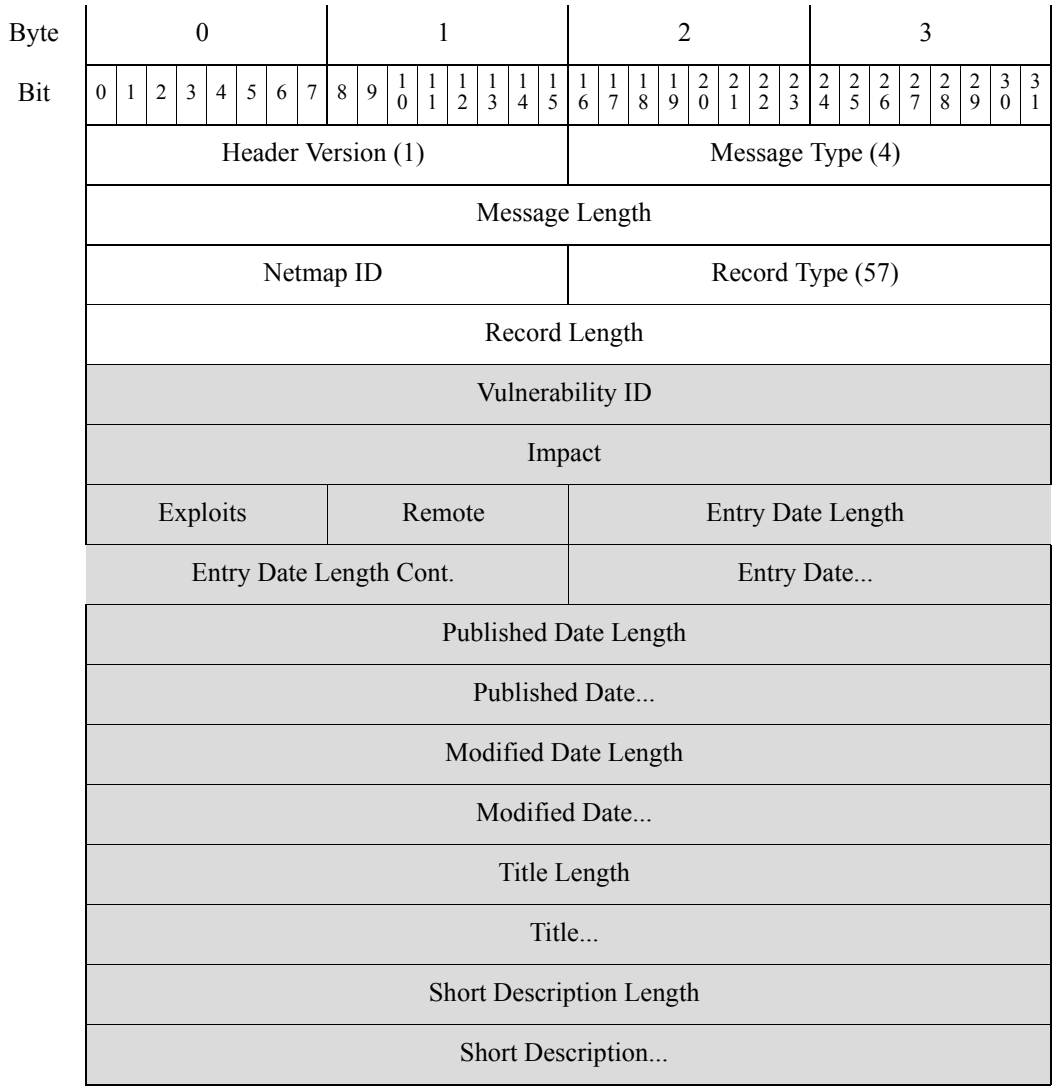
The following table describes the fields in the Client Application record.

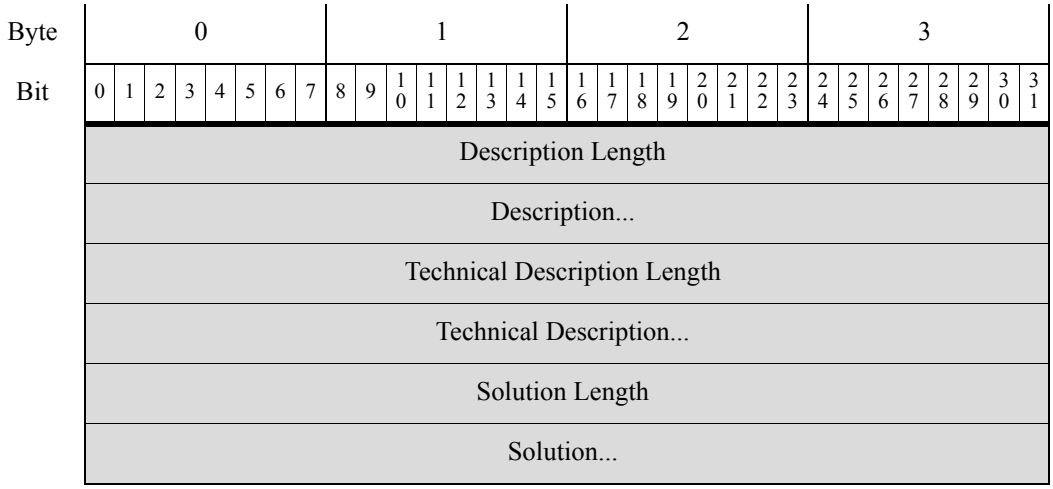
Table 4-3 Client Application Record Fields

Field	Data Type	Description
Application ID	uint32	The application ID number for the client application.
Name Length	uint32	The number of bytes included in the name.
Name	string	The client application name.

Vulnerability Record

The eStreamer service transmits metadata containing vulnerability information for an event within a Vulnerability record, the format of which is shown below. (Vulnerability information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 57, indicating a Vulnerability record.





The following table describes the fields in the Vulnerability record.

Table 4-4 Vulnerability Record Fields

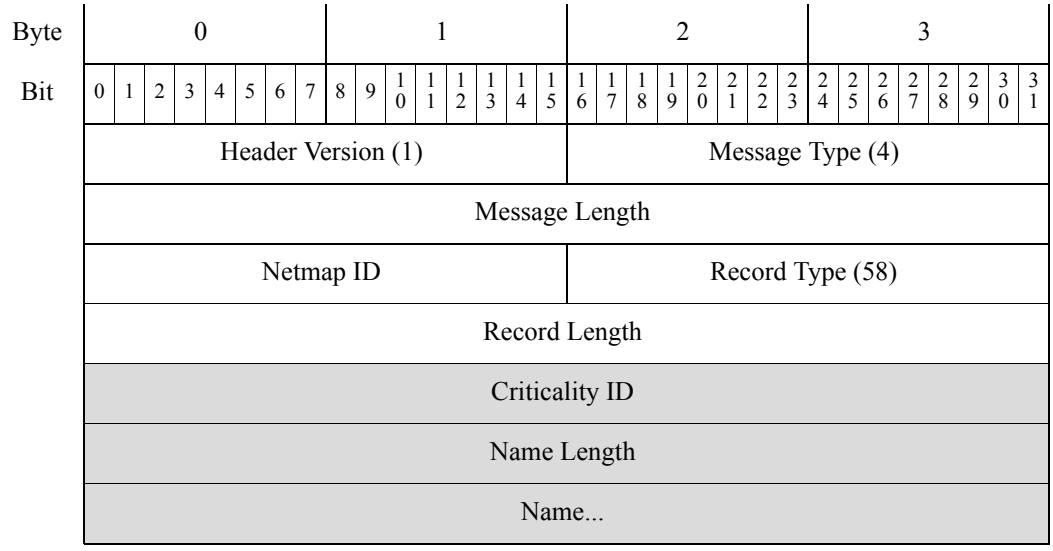
Field	Data Type	Description
Vulnerability ID	uint32	The vulnerability ID number.
Impact	uint32	The vulnerability impact, corresponding to the impact level determined through correlation of intrusion data, host discovery events, and vulnerability assessments. The value can be from `1` to `10`, with `10` being the most severe. The impact value of a vulnerability is determined by the writer of the Bugtraq entry.
Exploits	uint8	Indicates whether known exploits exist for the vulnerability. Possible values include: • `0` — Yes • `1` — No
Remote	uint8	Indicates whether the vulnerability can be exploited across a network. Possible values include: • `0` — Yes • `1` — No • Blank — Vulnerability to remote exploits unknown
Entry Date Length	uint32	The length of the entry date field.
Entry Date	string	The date the vulnerability was entered in the database.
Published Date Length	uint32	The length of the published date field.
Published Date	string	The date the vulnerability was published.
Modified Date Length	uint32	The length of the modified date field.
Modified Date	string	The date of the most recent modification to the vulnerability, if applicable.
Title Length	uint32	The length of the title field.

Table 4-4 Vulnerability Record Fields (continued)

Field	Data Type	Description
Title	string	The title of the vulnerability.
Short Description Length	uint32	The length of the short description field.
Short Description	string	A summary description of the vulnerability.
Description Length	uint32	The length of the description field.
Description	string	A general description of the vulnerability.
Technical Description Length	uint32	The length of the technical description field.
Technical Description	string	The technical description of the vulnerability.
Solution Length	uint32	The length of the solution field.
Solution	string	The solution to the vulnerability.

Criticality Record

The eStreamer service transmits metadata containing host criticality information for an event within a Criticality record, the format of which is shown below. (Criticality information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `58`, indicating a Criticality record.



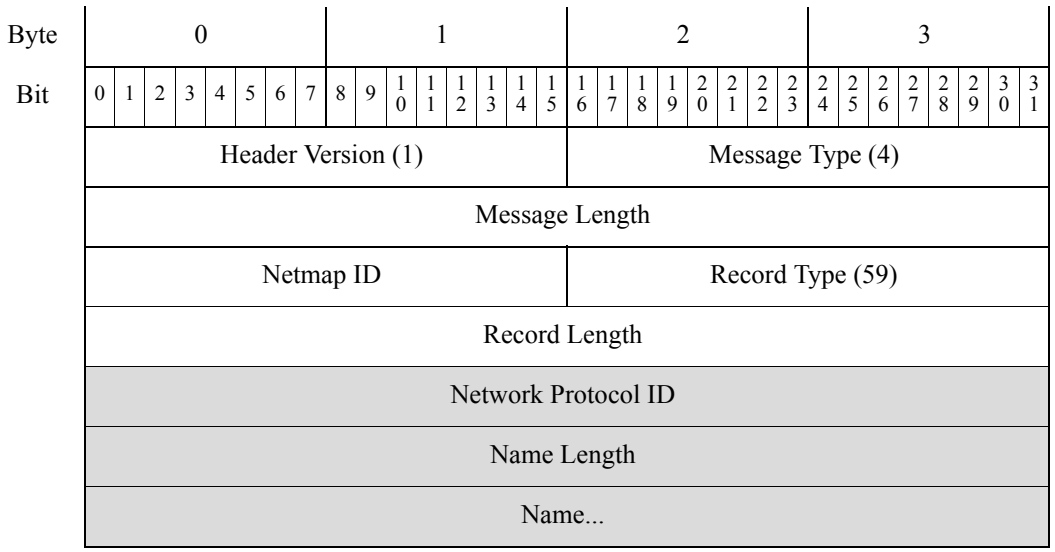
The following table describes the fields in the Criticality record.

Table 4-5 Criticality Record Fields

Field	Data Type	Description
Criticality ID	uint32	The criticality ID number.
Name Length	uint32	The number of bytes included in the criticality level.
Name	string	The criticality level.

Network Protocol Record

The eStreamer service transmits metadata containing network protocol information for an event within a Network Protocol record, the format of which is shown below. (Network protocol information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 59, indicating a Network Protocol record.



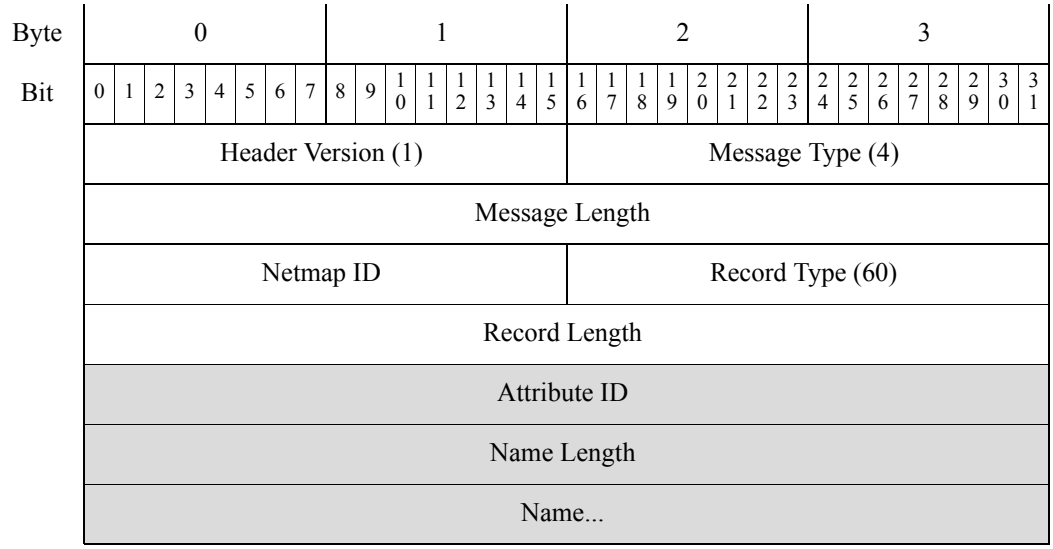
The following table describes the fields in the Network Protocol record.

Table 4-6 Network Protocol Record Fields

Field	Data Type	Description
Network Protocol ID	uint32	The network protocol ID number.
Name Length	uint32	The number of bytes included in the network protocol name.
Name	string	The name of the network protocol.

Attribute Record

The eStreamer service transmits metadata containing attribute information for an event within an Attribute record, the format of which is shown below. (Attribute information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `60`, indicating an Attribute record.



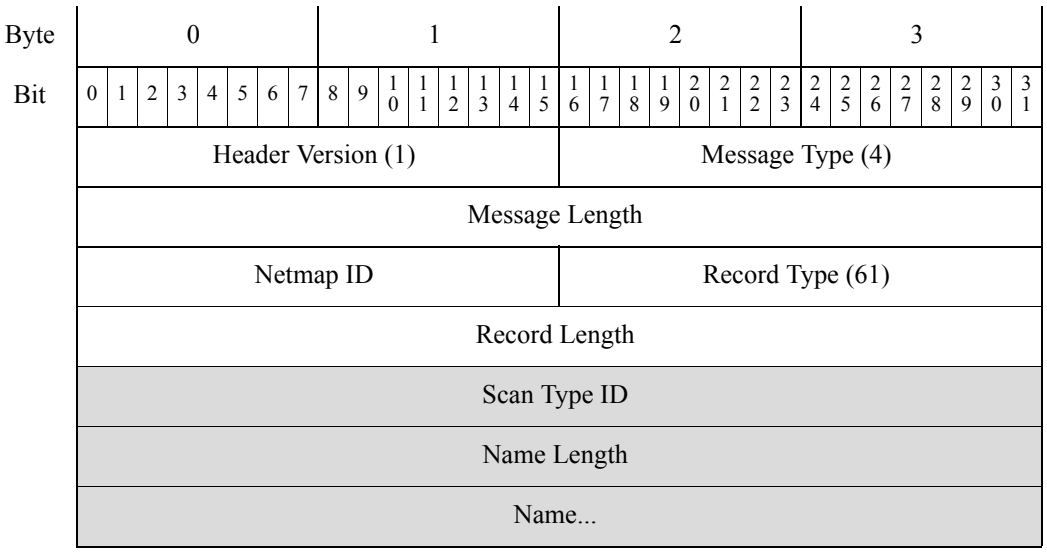
The following table describes the fields in the Attribute record.

Table 4-7 Attribute Record Fields

Field	Data Type	Description
Attribute ID	uint32	The attribute ID number.
Name Length	uint32	The number of bytes included in the attribute name.
Name	string	The name of the attribute.

Scan Type Record

The eStreamer service transmits metadata containing scan type information for an event within a Scan Type record, the format of which is shown below. (Scan type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `61`, indicating a Scan Type record.



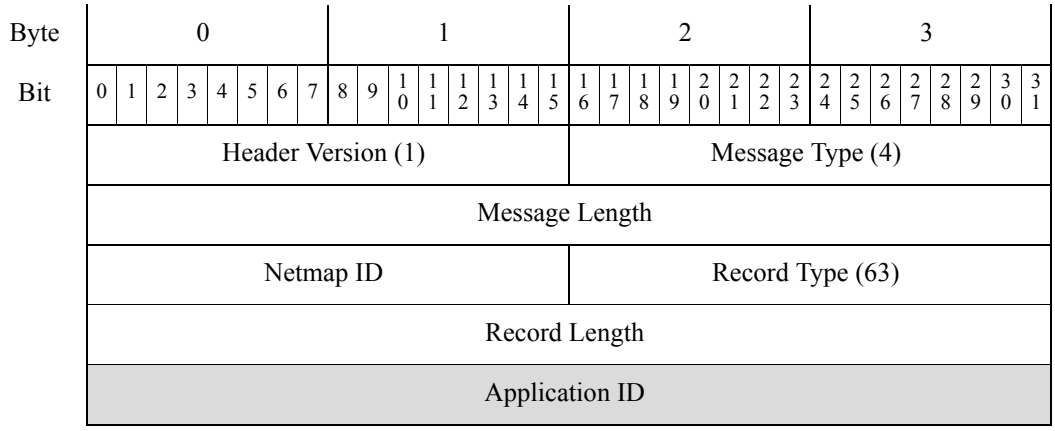
The following table describes the fields in the Scan Type record.

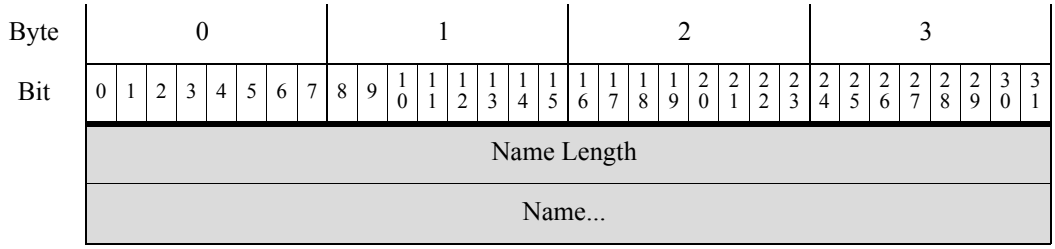
Table 4-8 Scan Type Record Fields

Field	Data Type	Description
Scan Type ID	uint32	The scan type ID number.
Name Length	uint32	The number of bytes included in the scan type name.
Name	string	The name of the scan type.

Server Record

The eStreamer service transmits metadata containing server information for an event within a Server record, the format of which is shown below. The application ID of the server's application protocol provides the cross-reference to the metadata. (Server information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `63`, indicating a Server record.





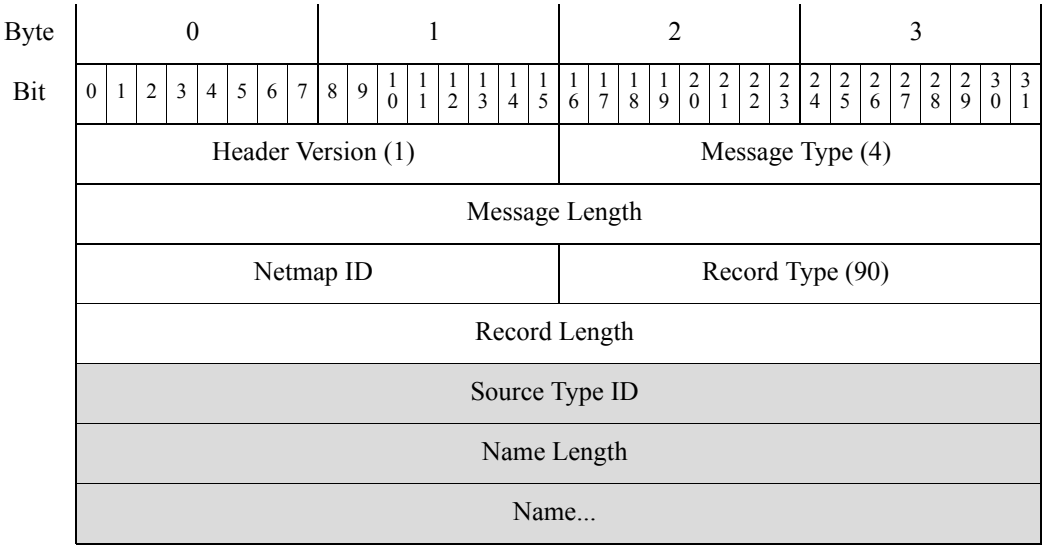
The following table describes the fields in the Server record.

Table 4-9 Server Record Fields

Field	Data Type	Description
Application ID	uint32	The application ID number of the application protocol.
Name Length	uint32	The number of bytes included in the server name.
Name	string	The name of the application protocol. For application ID 65535, the name is `unknown`.

Source Type Record

The eStreamer service transmits metadata containing information about the source application for an event within a Source Type record, the format of which is shown below. (Source type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `90`, indicating a Source Type record.



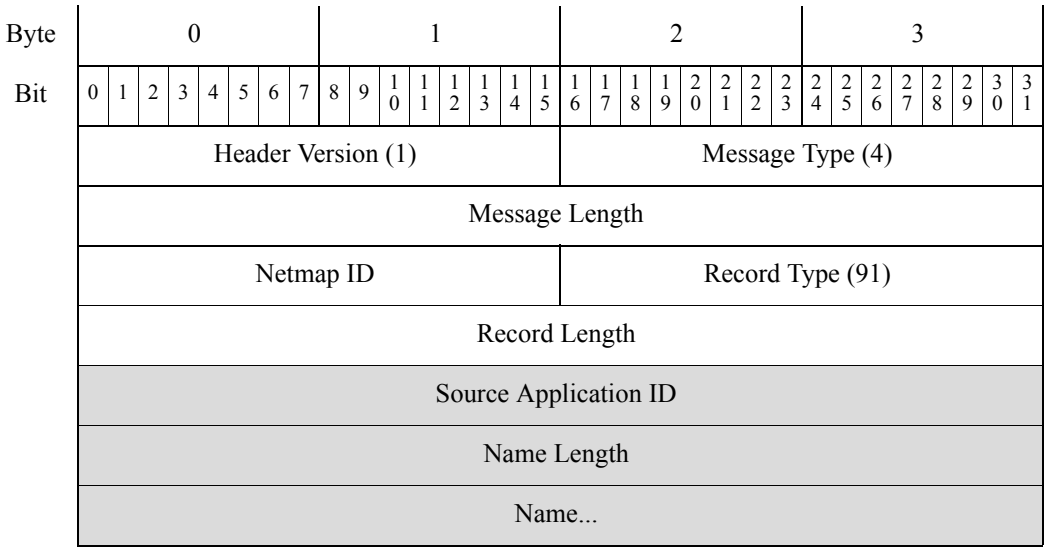
The following table describes the fields in the Source Type record.

Table 4-10 Source Type Record Fields

Field	Data Type	Description
Source Type ID	uint32	The identification number for the source type.
Name Length	uint32	The number of bytes included in the source type name.
Name	string	The name of the source type.

Source Application Record

The eStreamer service transmits metadata containing information about the source application for a host discovery event within a Source Application record, the format of which is shown below. (Source application information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `91`, indicating a Source Application record.



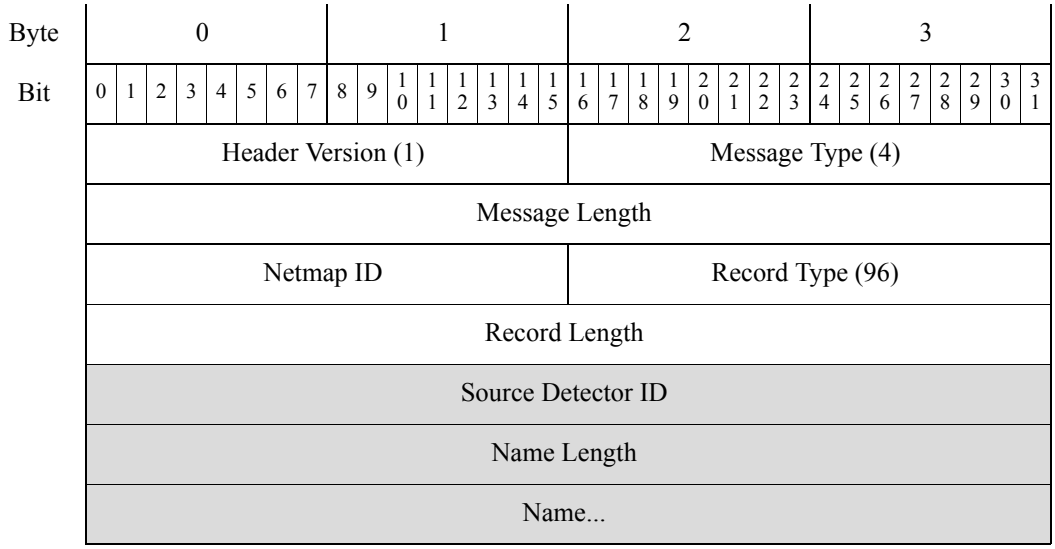
The following table describes the fields in the Source Application record.

Table 4-11 Source Application Record Fields

Field	Data Type	Description
Source Application ID	uint32	The ID number for the source application.
Name Length	uint32	The number of bytes included in the source application name.
Name	string	The name of the source application.

Source Detector Record

The eStreamer service transmits metadata containing information about the source application for a host discovery event within a Source Type record, the format of which is shown below. (Source type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `96`, indicating a Source Detector record.



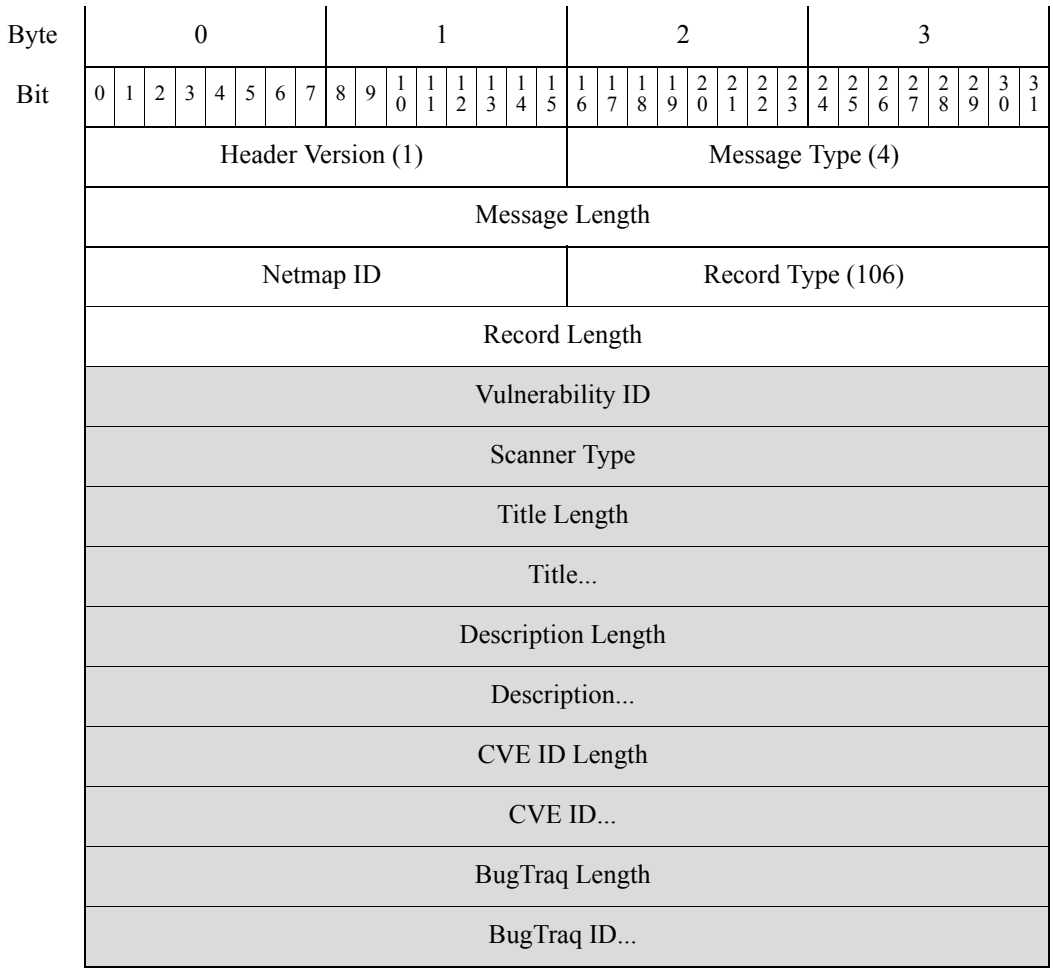
The following table describes the fields in the Source Detector record.

Table 4-12 Source Detector Record Fields

Field	Data Type	Description
Source Detector ID	uint32	The ID string for the source detector.
Name Length	uint32	The number of bytes included in the source type name.
Name	string	The name of the source detector.

Third Party Scanner Vulnerability Record

The eStreamer service transmits metadata containing third-party vulnerability information for an event within a Third Party Scanner Vulnerability record, the format of which is shown below. (Vulnerability information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `106`, indicating a Third Party Scanner Vulnerability record.



The following table describes the fields in the Vulnerability record.

Table 4-13 Third Party Scanner Vulnerability Record Fields

Field	Data Type	Description
Vulnerability ID	uint32	The third-party vulnerability ID number.
Scanner Type	uint32	The third-party scanner type.
Title Length	uint32	The length of the title field.
Title	string	The title of the vulnerability.
Description Length	uint32	The length of the description field.
Description	string	A general description of the vulnerability.
CVE ID Length	uint32	The length of the CVE ID field.
CVE ID	string	The Common Vulnerabilities and Exposures (CVE) ID number for the vulnerability.
BugTraq ID Length	uint32	The length of the BugTraq ID field.
BugTraq ID	string	The BugTraq ID number for the vulnerability.

User Record

The eStreamer service transmits metadata containing information about users detected by the system within a User record, the format of which is shown below. (User information is sent when the Version 4 metadata and the policy event request flag—bits 20 and 22, respectively, in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of `98`, indicating a User record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (98)															
Record Length																																
User Data Block Type (57)																																
User Data Block Length																																
User ID																																
Protocol																																
String Block Type (0)																																
String Block Length																																
Username...																																

The following table describes the fields in the User record.

Table 4-14 User Record Fields

Field	Data Type	Description
User Data Block Type	uint32	Initiates an User Data block. This value is always `57`. The block type is a series 2 block.
User Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
User ID	uint32	The unique identifier for the user.

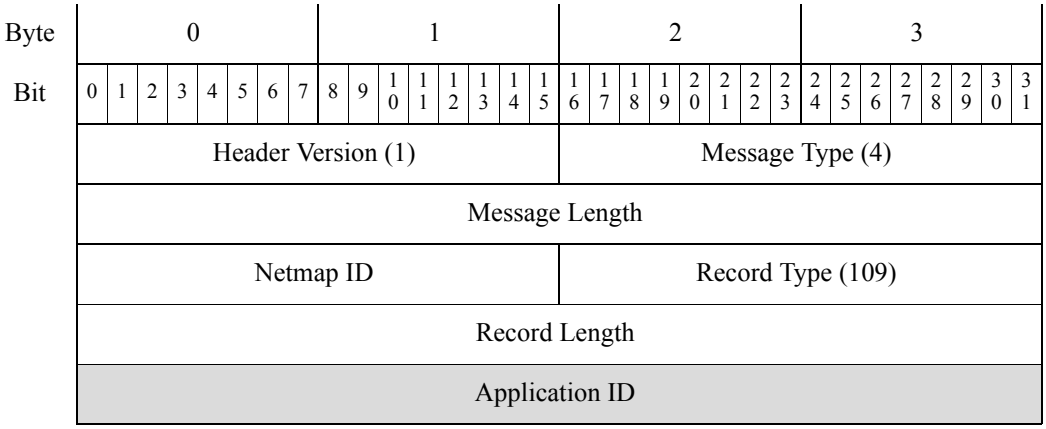
Table 4-14 User Record Fields (continued)

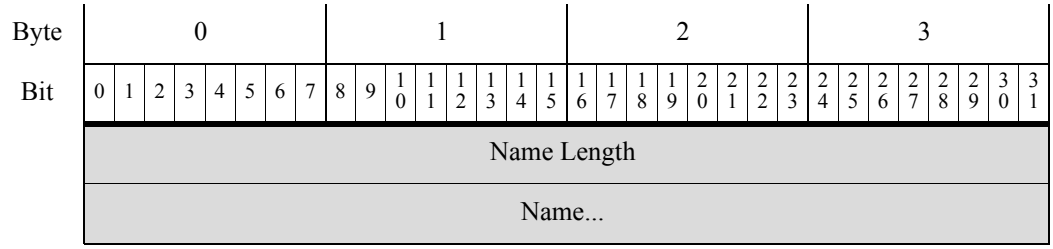
Field	Data Type	Description
Protocol	uint32	Protocol used to detect or report the user. Possible values are: • 165 - FTP • 426 - SIP • 547 - AOL Instant Messenger • 683 - IMAP • 710 - LDAP • 767 - NTP • 773 - Oracle Database • 788 - POP3 • 1755 - MDNS
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the username String data block, including eight bytes for the block type and header fields plus the number of bytes in the Username field.
Username	string	The name of the user

Web Application Record

The system detects the content of HTTP traffic from websites, if available. Web application metadata for a host discovery event may include the specific type of content (for example, WMV or QuickTime).

The eStreamer service transmits the web application metadata for an event within a Web Application record, the format of which is shown below. (Web application metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 109, indicating a Web Application record.





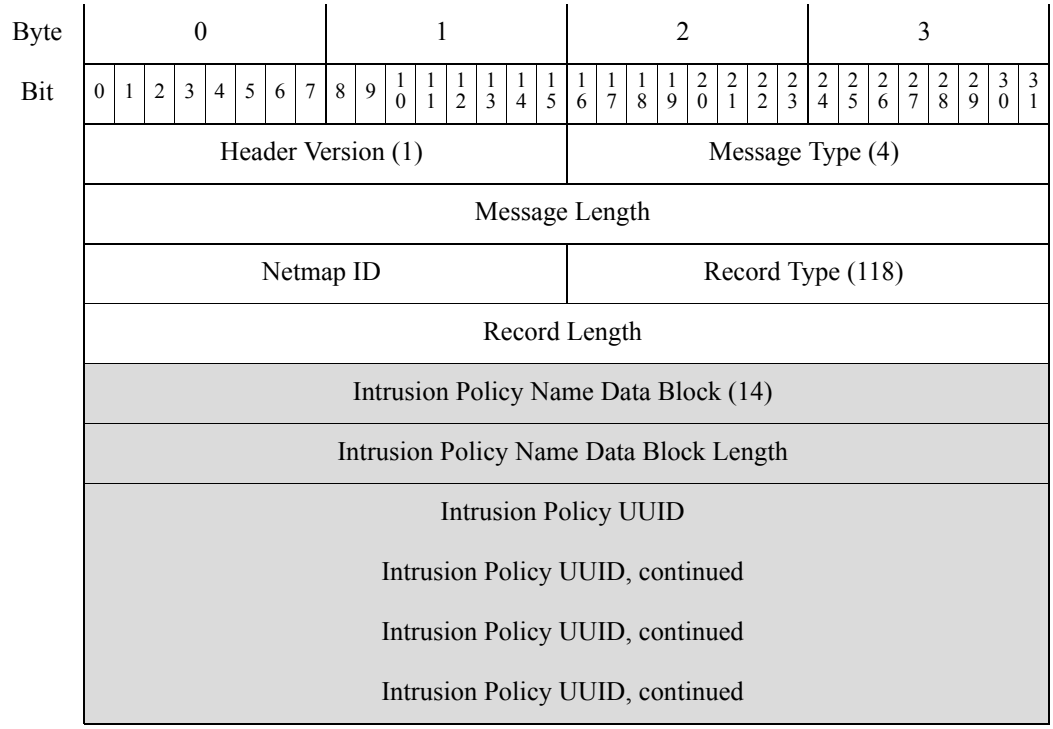
The following table describes the fields in the Web Application record.

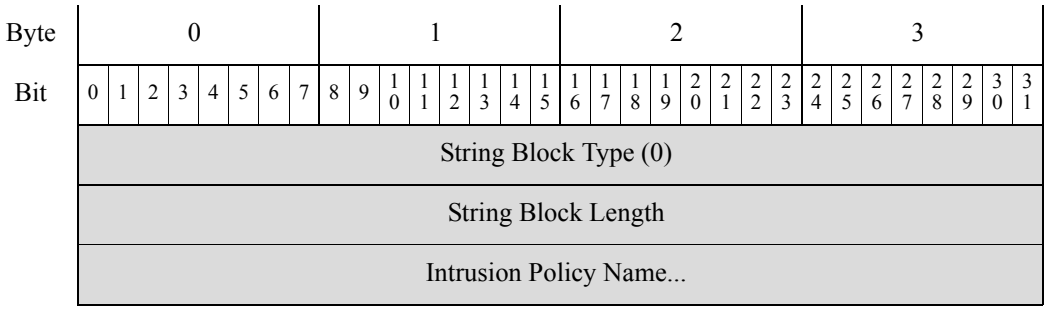
Table 4-15 Web Application Record Fields

Field	Data Type	Description
Application ID	uint32	Application ID number of the web application.
Name Length	uint32	The number of bytes included in the name.
Name	string	The web application content name.

Intrusion Policy Name Record

The eStreamer service transmits metadata containing intrusion policy name information for a connection event within an Intrusion Policy Name record, the format of which is shown below. (Intrusion policy name information is sent when one of the metadata flags—version 4 metadata bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Intrusion Policy Name record field, which appears after the Message Length field, has a value of `118`, indicating an Intrusion Policy Name record. It contains a UUID String data block, block type 14 in the series 2 set of data blocks.





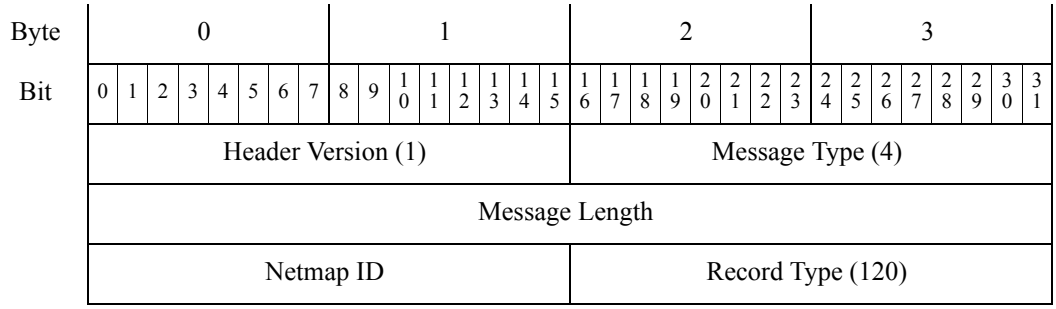
The following table describes the fields in the Intrusion Policy Name data block.

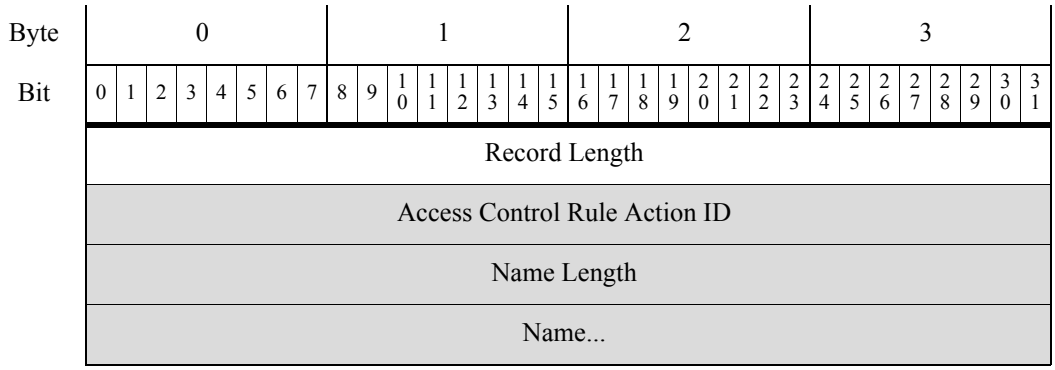
Table 4-16 Intrusion Policy Name Data Block Fields

Field	Data Type	Description
Intrusion Policy Name Data Block Type	uint32	Initiates an Intrusion Policy Name data block. This value is always `14`. The block type is a series 2 block.
Intrusion Policy Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Intrusion Policy UUID	uint8[16]	The unique identifier for the intrusion policy associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the intrusion policy. This value is always `0`.
String Block Length	uint32	The number of bytes included in the intrusion policy name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Intrusion Policy Name	string	The intrusion policy name.

Access Control Rule Action Record Metadata

The eStreamer service transmits metadata containing the action associated with a triggered access control rule within an Access Control Rule Action record, the format of which is shown below. (Access Control Rule Action information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Access Control Rule Action record field, which appears after the Message Length field, has a value of `120`, indicating an Access Control Rule Action record.





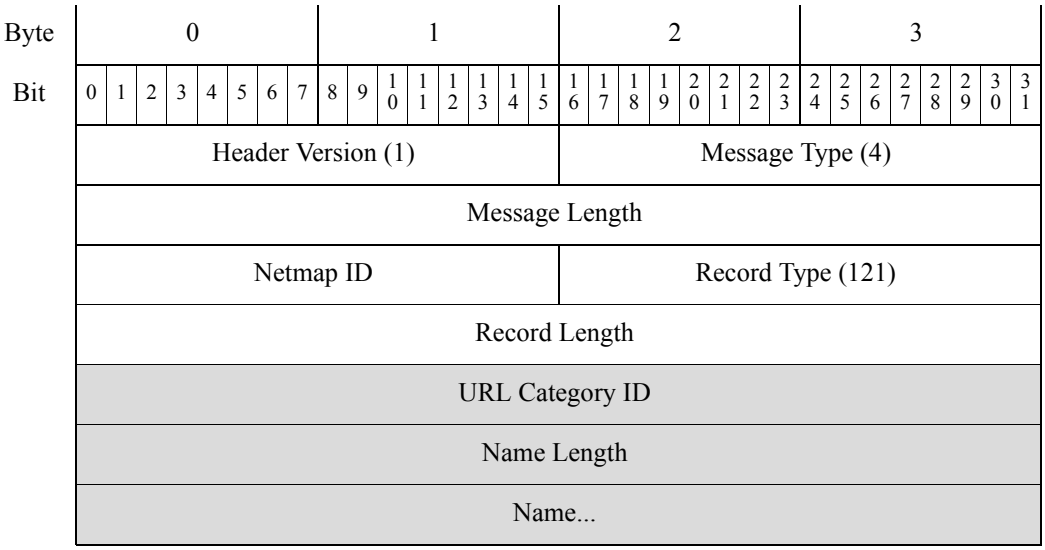
The following table describes the fields in the Access Control Rule Action record.

Table 4-17 Access Control Rule Action Record Fields

Field	Data Type	Description
Access Control Rule Action ID	uint32	ID number of the access control rule action.
Name Length	uint32	The number of bytes included in the name.
Name	string	The firewall rule action name.

URL Category Record Metadata

The eStreamer service transmits metadata containing the category name associated with a URL in a connection log within a URL Category record, the format of which is shown below. (URL category information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the record field, which appears after the Message Length field, has a value of `121`, indicating a URL Category record.



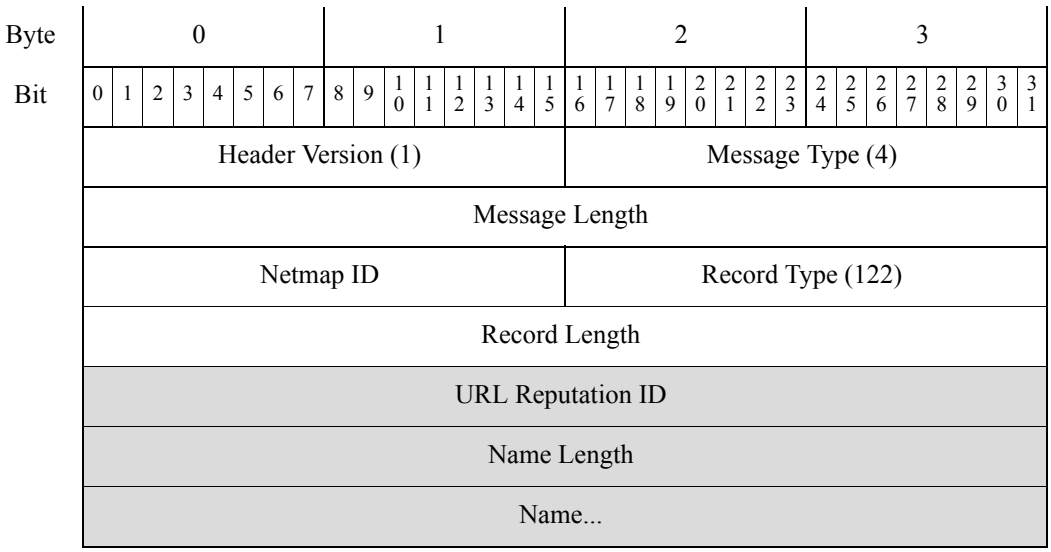
The following table describes the fields in the URL Category record.

Table 4-18 URL Category Record Fields

Field	Data Type	Description
URL Category ID	uint32	ID number of the URL category.
Name Length	uint32	The number of bytes included in the name.
Name	string	The URL category name.

URL Reputation Record Metadata

The eStreamer service transmits metadata containing the reputation (that is, risk level) associated with a URL in a connection log within a URL Reputation record, the format of which is shown below. (URL reputation information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the URL Reputation metadata record field, which appears after the Message Length field, has a value of `122`, indicating a URL Reputation metadata record.



The following table describes the fields in the URL Reputation record.

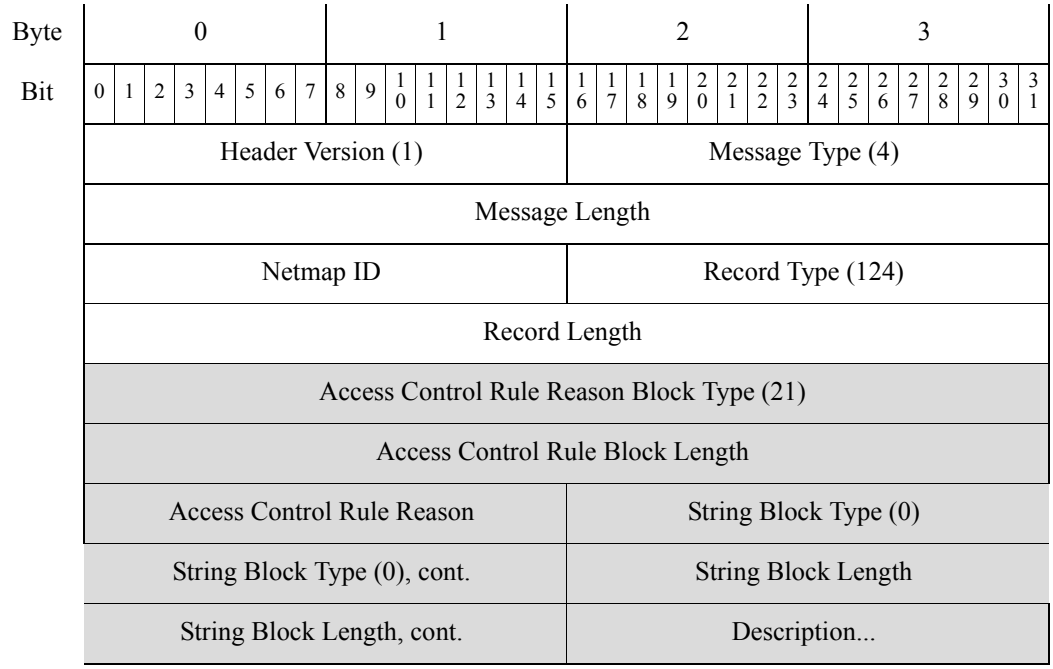
Table 4-19 URL Reputation Record Fields

Field	Data Type	Description
URL Reputation ID	uint32	ID number of the URL reputation.
Name Length	uint32	The number of bytes included in the name.
Name	string	The URL reputation name.

Access Control Rule Reason Metadata

The eStreamer service transmits metadata containing information about the reason an access control rule triggered an intrusion event or connection event within an Access Control Rule Reason record, the format of which is shown below. Access control rule reason metadata is sent when the Version 4

metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11. Note that the Record Type field, which appears after the Message Length field, has a value of `124`, indicating an Access Control Rule Reason record. It contains an Access Control Rule Reason Block (as documented in Access Control Rule Reason Data Block 5.1+, page 4-184). The Access Control Rule Reason data block is block type 21 in series 2.



The following table describes the fields in the Access Control Rule ID data block.

Table 4-20 Access Control Rule Reason Metadata Fields

Field	Data Type	Description
Access Control Rule Reason Block Type	uint32	Initiates an Access Control Rule Reason block. This value is always `21`. This is a series 2 data block.
Access Control Rule Reason Block Length	uint32	Total number of bytes in the Access Control Rule Reason block, including eight bytes for the Access Control Rule Reason block type and length fields, plus the number of bytes of data that follows.
Access Control Rule Reason	uint16	The reason the Access Control rule logged the connection.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the Access Control rule reason.

Security Intelligence Category Metadata

The eStreamer service transmits metadata containing information about the Security Intelligence category within a Security Intelligence Category record, the format of which is shown below. Access control rule reason metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11. Note that the Record Type field, which appears after the Message Length field, has a value of `280`, indicating a Security Intelligence Category record. It contains a Security Intelligence Category data block (as documented in Security Intelligence Category Data Block 5.1+, page 4-185). The Security Intelligence data block is block type 22 in series 2.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (280)															
Record Length																																
Security Intelligence Category Block Type (22)																																
Security Intelligence Category Block Length																																
Security Intelligence List ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
String Block Type (0)																																
String Block Length																																
Security Intelligence List Name...																																

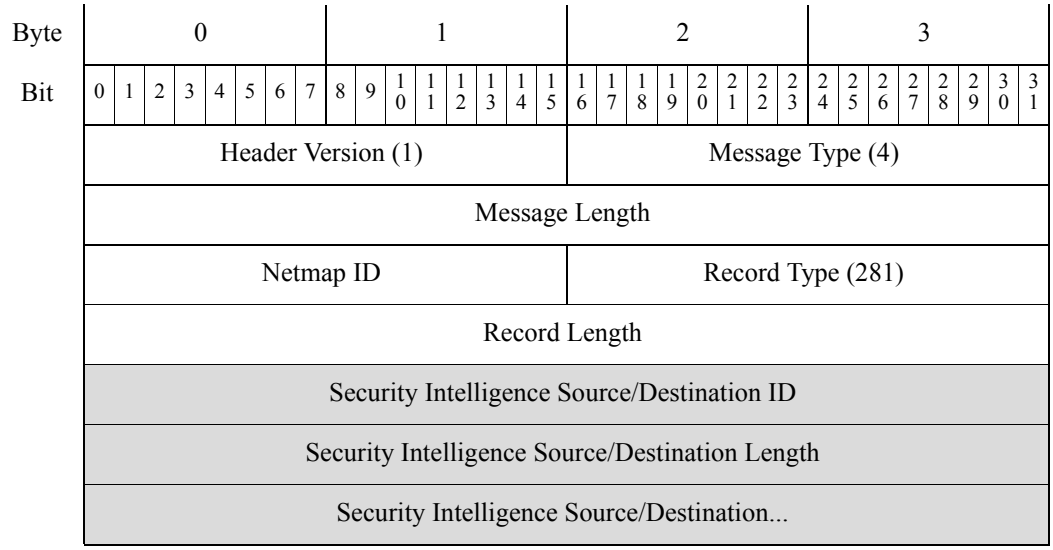
The following table describes the fields in the Security Intelligence Category record.

Table 4-21 Security Intelligence Category Metadata Fields

Field	Data Type	Description
Security Intelligence Category Block Type	uint32	Initiates an Security Intelligence Category data block. This value is always 22. This is a series 2 data block.
Security Intelligence Category Block Length	uint32	Total number of bytes in the Security Intelligence Category block, including eight bytes for the Security Intelligence Category block type and length fields, plus the number of bytes of data that follows.
Security Intelligence List ID	uint32	The ID of the IP blacklist or whitelist triggered by the connection.
Access Control Policy UUID	uint8[16]	The UUID of the access control policy configured for Security Intelligence.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Security Intelligence List Name field.
Security Intelligence List Name	string	The name of the IP category blacklist or whitelist triggered by the connection.

Security Intelligence Source/Destination Record

The eStreamer service transmits metadata containing whether a Security Intelligence-detected IP address is a source IP address or destination IP address within a Security Intelligence Source/Destination record, the format of which is shown below. (The source/destination IP information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags, page 2-11.) Note that the Record Type field, which appears after the Message Length field, has a value of 281, indicating a Security Intelligence Source/Destination record.



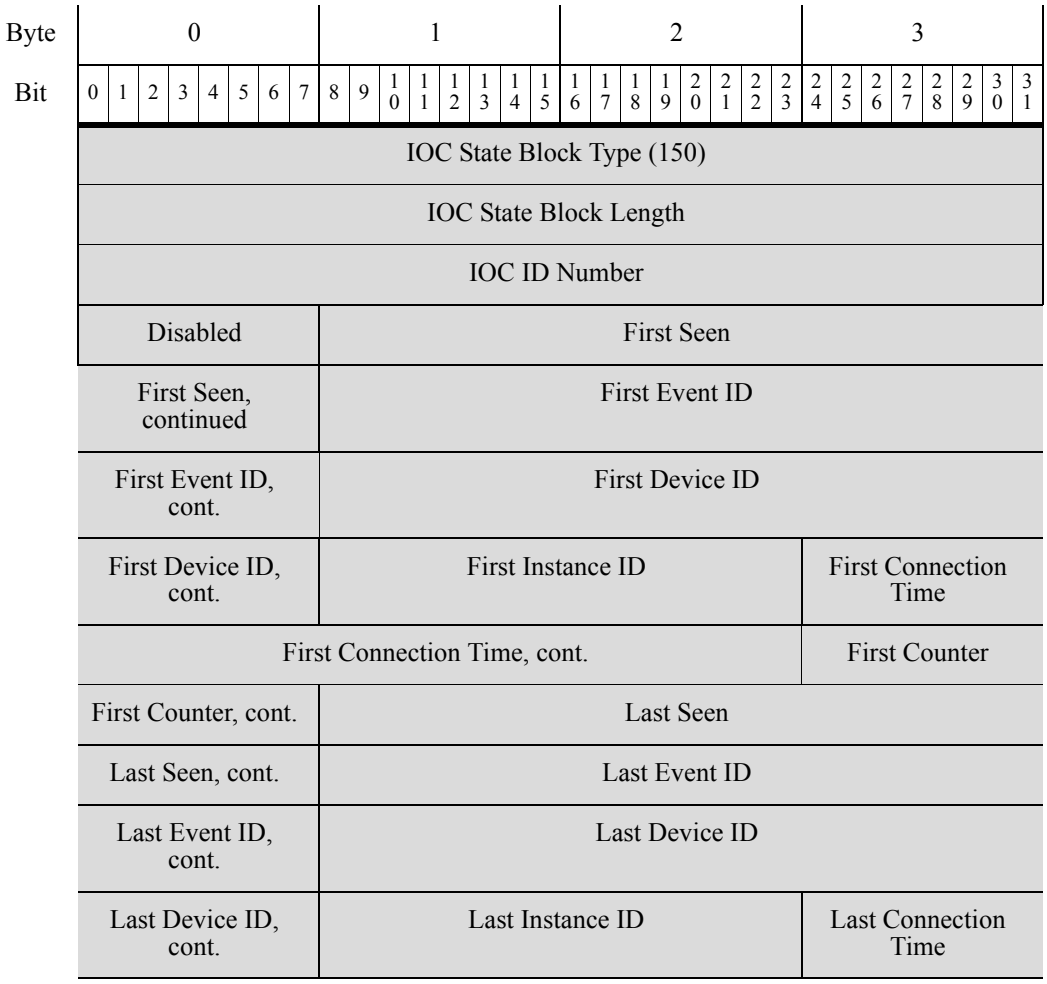
The following table describes the fields in the Security Intelligence Source/Destination record.

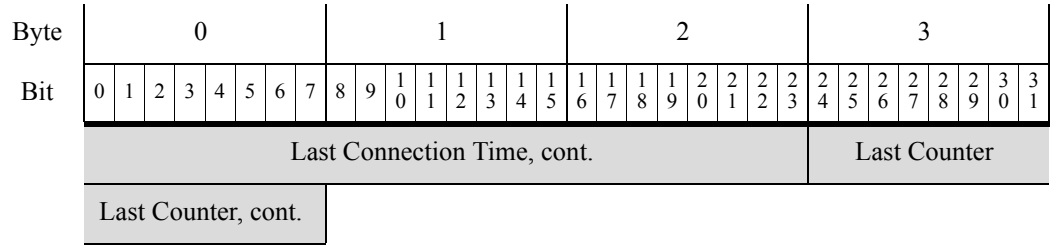
Table 4-22 Security Intelligence Source/Destination Record Fields

Field	Data Type	Description
Security Intelligence Source/ Destination ID	uint32	The Security Intelligence source/destination ID number.
Security Intelligence Source/ Destination Length	uint32	The number of bytes included in the Security Intelligence source/destination.
Security Intelligence Source/ Destination	string	Whether the detected IP address is a source or destination IP address.

IOC State Data Block for 5.3+

The IOC State data block provides information about an Indication of Compromise (IOC). It is block type of 150 in series 1. It is used by the host tracker to store information about a compromise on a host. The following diagram shows the structure of an IOC State data block:





The following table describes the components of the IOC State data block.

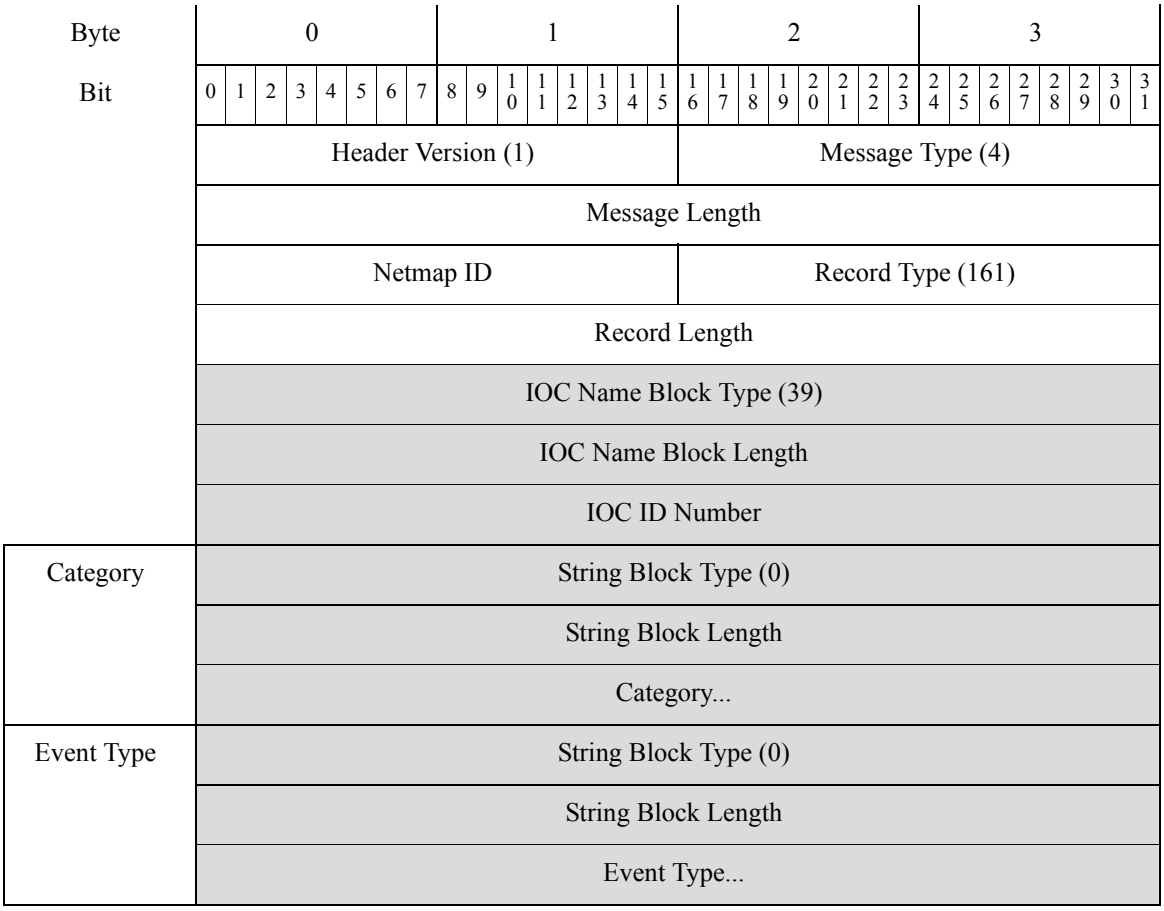
Table 4-23 IOC State Data Block Fields

Field	Data Type	Description
IOC State Data Block Type	uint32	Initiates an IOC State data block. This value is always `150`.
IOC State Data Block Length	uint32	Total number of bytes in the IOC State data block, including eight bytes for the IOC State data block type and length fields, plus the number of bytes of data that follows.
IOC ID Number	uint32	Unique ID number for the compromise.
Disabled	uint8	Indicates whether the compromise has been disabled on the host: • `0` — The compromise is not disabled. • `1` — The compromise is disabled.
First Seen	uint32	Unix timestamp of when this compromise was first seen.
First Event ID	uint32	ID number of the event on which this compromise was first seen.
First Device ID	uint32	ID of the sensor which first detected the IOC.
First Instance ID	uint16	Numerical ID of the Snort instance on the managed device that first detected the compromise.
First Connection Time	uint32	Unix timestamp of the connection where this compromise was first seen.
First Counter	uint16	Counter for the connection on which this compromise was last seen. Used to differentiate between multiple connections occurring at the same time.
Last Seen	uint32	Unix timestamp of when this compromise was last seen
Last Event ID	uint32	ID number of the event on which this compromise was last seen.
Last Device ID	uint32	ID of the sensor which most recently detected the IOC.
Last Instance ID	uint16	Numerical ID of the Snort instance on the managed device that last detected the compromise.
Last Connection Time	uint32	Unix timestamp of the connection on which this compromise was last seen.
Last Counter	uint16	Counter for the connection on which this compromise was last seen. Used to differentiate between multiple connections occurring at the same time.

IOC Name Data Block for 5.3+

This is a data block that provides the category and event type for an Indication of Compromise (IOC). The record type is 161, with a block type of 39 in series 2. It is exposed as metadata for any event that has IOC information. These include malware events, file events, and intrusion events.

The following diagram shows the structure of an IOC Name data block:



The following table describes the fields in the IOC Name data block.

Table 4-24 IOC Name Data Block Fields

Field	Data Type	Description
IOC Name Data Block Type	uint32	Initiates an IOC Name data block. This value is always `39`.
IOC Name Data Block Length	uint32	Total number of bytes in the IOC Name data block, including eight bytes for the IOC Name data block type and length fields, plus the number of bytes of data that follows.
IOC ID Number	uint32	Unique ID number for the compromise.
String Block Type	uint32	Initiates a String data block containing the category associated with the compromise. This value is always `0`.

Table 4-24 IOC Name Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category field.
Category	string	The category for the compromise. Possible values include: • `CnC Connected` • `Exploit Kit` • `High Impact Attack` • `Low Impact Attack` • `Malware Detected` • `Malware Executed` • `Dropper Infection` • `Java Compromise` • `Word Compromise` • `Adobe Reader Compromise` • `Excel Compromise` • `PowerPoint Compromise` • `QuickTime Compromise`
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always `0`.

Table 4-24 IOC Name Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.

Table 4-24 IOC Name Data Block Fields (continued)

Field	Data Type	Description
Event Type	string	The event type for the compromise. Possible values include: • Adobe Reader launched shell • Dropper Infection Detected by AMP for Endpoints • Excel Compromise Detected by AMP for Endpoints • Excel launched shell • Impact 1 Intrusion Event - attempted-admin • Impact 1 Intrusion Event - attempted-user • Impact 1 Intrusion Event - successful-admin • Impact 1 Intrusion Event - successful-user • Impact 1 Intrusion Event - web-application-attack • Impact 2 Intrusion Event - attempted-admin • Impact 2 Intrusion Event - attempted-user • Impact 2 Intrusion Event - successful-admin • Impact 2 Intrusion Event - successful-user • Impact 2 Intrusion Event - web-application-attack • Intrusion Event - exploit-kit • Intrusion Event - malware-backdoor • Intrusion Event - malware-cnc • Java Compromise Detected by AMP for Endpoints • Java launched shell • PDF Compromise Detected by AMP for Endpoints • PowerPoint Compromise Detected by AMP for Endpoints • PowerPoint launched shell • QuickTime Compromise Detected by AMP for Endpoints • QuickTime launched shell • Security Intelligence Event - CnC • Security Intelligence Event - DNS CnC • Security Intelligence Event - DNS Malware • Security Intelligence Event - DNS Phishing • Security Intelligence Event - Sinkhole CnC • Security Intelligence Event - Sinkhole Malware • Security Intelligence Event - Sinkhole Phishing • Security Intelligence Event - URL CnC • Security Intelligence Event - URL Malware • Security Intelligence Event - URL Phishing • Suspected Botnet Detected by AMP for Endpoints • Threat Detected by AMP for Endpoints - Executed • Threat Detected by AMP for Endpoints - Not Executed • Threat Detected in File Transfer • Word Compromise Detected by AMP for Endpoints • Word launched shell

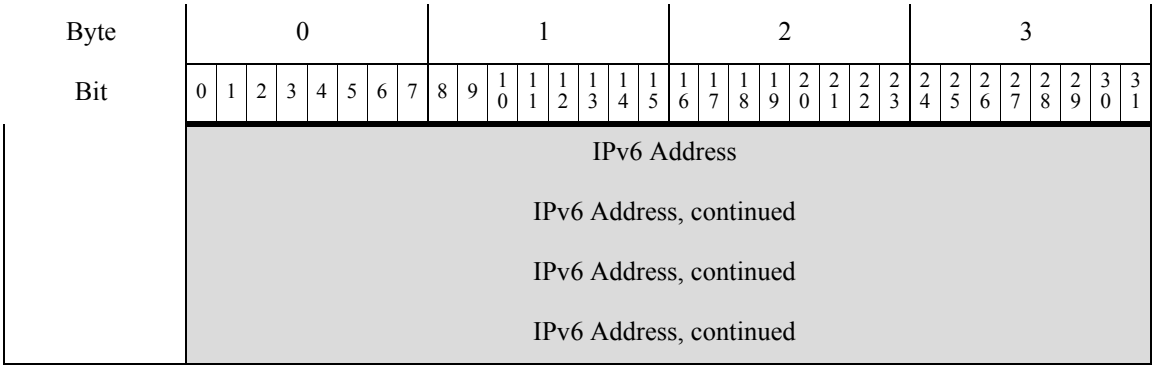
Discovery Event Header 5.2+

Discovery and connection event messages contain a discovery event header. It conveys the type and subtype of the event, the time the event occurred, the device on which the event occurred, and the structure of the event data in the message. This header is followed by the actual host discovery, user, or connection event data. The structures associated with the different event type/subtype values are described in Host Discovery Structures by Event Type, page 4-38. This header has IPv6 support, and deprecates Discovery Event Header 5.0 - 5.1.1.x, page B-87.

The event type and event subtype fields of the discovery event header identify the structure of the transmitted event message. Once the structure of the event data block is determined, your program can parse the message appropriately.

The shaded rows in the following diagram illustrate the format of the discovery event header.

	Byte	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																	Message Type (4)															
	Message Length																																
	Netmap ID																	Record Type															
	Record Length																																
	eStreamer Server Timestamp (in events, only if bit 23 is set)																																
	Reserved for Future Use (in events, only if bit 23 is set)																																
Discovery Event Header	Device ID																																
	Legacy IP Address																																
	MAC Address																																
	MAC Address, continued																	Has IPv6								Reserved for future use							
	Event Second																																
	Event Microsecond																																
	Event Type																																
	Event Subtype																																
	File Number (Internal Use Only)																																
	File Position (Internal Use Only)																																



The following table describes the discovery event header.

Table 4-25 Discovery Event Header Fields

Field	Data Types	Description
Device ID	uint32	ID number of the device that generated the discovery event. You can obtain the metadata for the device by requesting Version 3 and 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Legacy IP Address	uint32	This field is reserved but no longer populated. The IPv4 address is stored in the IPv6 Address field. See IP Addresses, page 1-6 for more information.
MAC Address	uint8[6]	MAC address of the host involved in the event.
Has IPv6	uint8	Flag indicating that the host has an IPv6 address.
Reserved for future use	uint8	Reserved for future use
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) that the system generated the event.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the system generated the event.
Event Type	uint32	Event type (`1000` for new events, `1001` for change events, `1002` for user input events, `1050` for full host profile). See Host Discovery Structures by Event Type, page 4-38 for a list of available event types.
Event Subtype	uint32	Event subtype. See Host Discovery Structures by Event Type, page 4-38 for a list of available event subtypes.
File Number	byte[4]	Serial file number. This field is for Cisco internal use and can be disregarded.
File Position	byte[4]	Event's position in the serial file. This field is for Cisco internal use and can be disregarded.
IPv6 Address	uin8[16]	IPv6 address. This field is present and used if the Has IPv6 flag is set.

Discovery and Connection Event Types and Subtypes

The values in the Event Type and Event Subtype fields identify and classify the event contained in a host discovery or user data message. They also identify the structure of the data in the message.

The following table lists the event types and event subtypes for discovery and connection events.

Table 4-26 Discovery and Connection Events by Type and Subtype

Event Name	Event Type	Event Subtype
New Host	1000	1
New TCP Server	1000	2
New Network Protocol	1000	3
New Transport Protocol	1000	4
New IP to IP Traffic	1000	5
New UDP Server	1000	6
New Client Application	1000	7
New OS	1000	8
New IPv6 to IPv6 Traffic	1000	9
Host IP Address Changed	1001	1
OS Information Update	1001	2
Host IP Address Reused	1001	3
Vulnerability Change	1001	4
Hops Change	1001	5
TCP Server Information Update	1001	6
Host Timeout	1001	7
TCP Port Closed	1001	8
UDP Port Closed	1001	9
UDP Server Information Update	1001	10
TCP Port Timeout	1001	11
UDP Port Timeout	1001	12
MAC Information Change	1001	13
Additional MAC Detected for Host	1001	14
Host Last Seen	1001	15
Host Identified as Router/Bridge	1001	16
Connection Statistics	1001	17
VLAN Tag Information Update	1001	18
Host Deleted: Host Limit Reached	1001	19
Client Application Timeout	1001	20
NetBIOS Name Change	1001	21
NetBIOS Domain Change	1001	22

Table 4-26 Discovery and Connection Events by Type and Subtype (continued)

Event Name	Event Type	Event Subtype
Host Dropped: Host Limit Reached	1001	23
Banner Update	1001	24
TCP Server Confidence Update	1001	25
UDP Server Confidence Update	1001	26
Identity Conflict	1001	29
Identity Timeout	1001	30
Secondary Host Update	1001	31
Client Application Update	1001	32
User Set Valid Vulnerabilities (Legacy)	1002	1
User Set Invalid Vulnerabilities (Legacy)	1002	2
User Delete Address (Legacy)	1002	3
User Delete Server (Legacy)	1002	4
User Set Host Criticality	1002	5
Host Attribute Add	1002	6
Host Attribute Update	1002	7
Host Attribute Delete	1002	8
Host Attribute Set Value (Legacy)	1002	9
Host Attribute Delete Value (Legacy)	1002	10
Add Scan Result	1002	11
User Set Vulnerability Qualification	1002	12
User Policy Control	1002	13
Delete Protocol	1002	14
Delete Client Application	1002	15
User Set Operating System	1002	16
User Account Seen	1002	17
User Account Update	1002	18
User Set Server	1002	19
User Delete Address (Current)	1002	20
User Delete Server (Current)	1002	21
User Set Valid Vulnerabilities (Current)	1002	22
User Set Invalid Vulnerabilities (Current)	1002	23
User Host Criticality	1002	24
Host Attribute Set Value (Current)	1002	25
Host Attribute Delete Value (Current)	1002	26
User Add Host	1002	27
User Add Server	1002	28

Table 4-26 Discovery and Connection Events by Type and Subtype (continued)

Event Name	Event Type	Event Subtype
User Add Client Application	1002	29
User Add Protocol	1002	30
Reload App	1002	31
Account Delete	1002	32
Connection Statistics	1003	1
Connection Chunks	1003	2
New User Identity	1004	1
User Login	1004	2
Delete User Identity	1004	3
User Identity Dropped: User Limit Reached	1004	4
Full Host Profile	1050	N/A

Tip For information about the data structure used for each event type/subtype, see Host Discovery Structures by Event Type, page 4-38.

Host Discovery Structures by Event Type

eStreamer builds host discovery event messages based on the event type indicated in the discovery event header. The following sub-sections describe the high-level structure for each event type:

The data block diagrams in the following sections depict the different record data blocks returned in host discovery event messages.

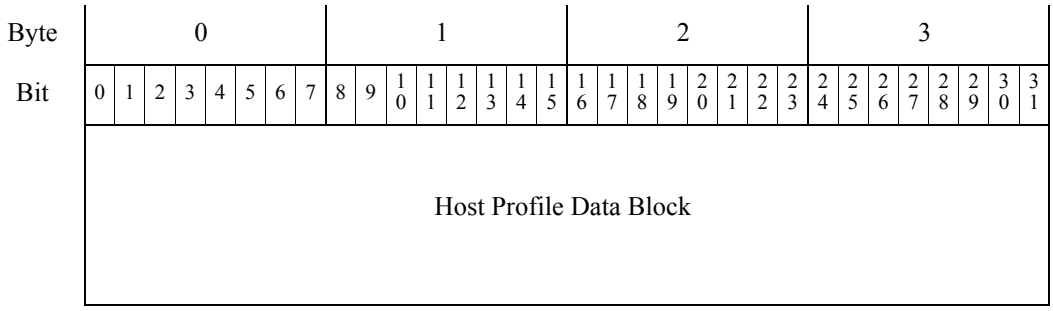
New Host and Host Last Seen Messages

New Host and Host Last Seen event messages have a standard discovery event header and a Host Profile data block (as documented in Host Profile Data Block for 5.2+, page 4-152). The Host Profile data block is block type 139 in series 1.

Note that the Host Last Seen message includes server information only for servers on the host that have changed within the Update Interval set in the discovery detection policy. In other words, only servers that have changed since the system last reported information will be included in the Host Last Seen message.

Note The Host Profile data block differs depending on which system version created the message. For information on legacy versions of the Host Profile data block, see Legacy Host Data Structures, page B-224.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Discovery Event Header																																



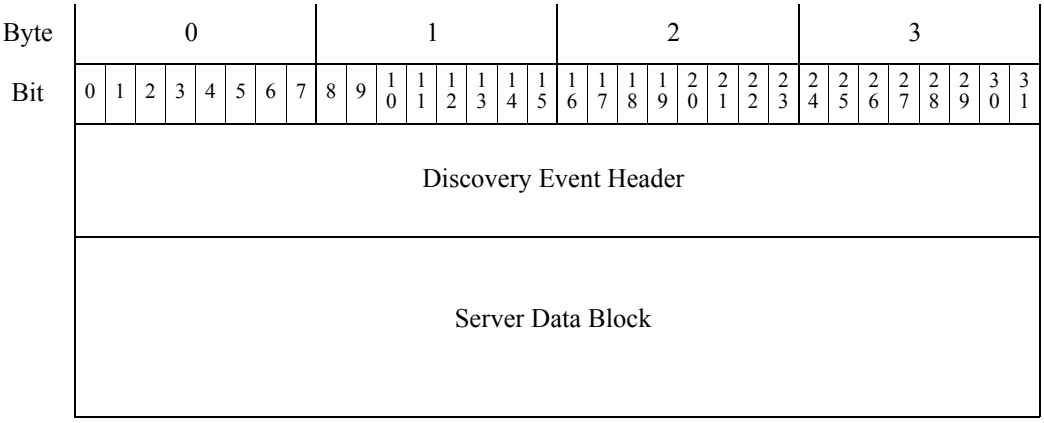
Server Messages

The following TCP and UDP server event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Server data block (as documented in Host Server Data Block 4.10.0+, page 4-128, block type 103 in series 1):

- New TCP Server
- New UDP Server
- TCP Server Information Update
- UDP Server Information Update
- TCP Server Confidence Update
- UDP Server Confidence Update

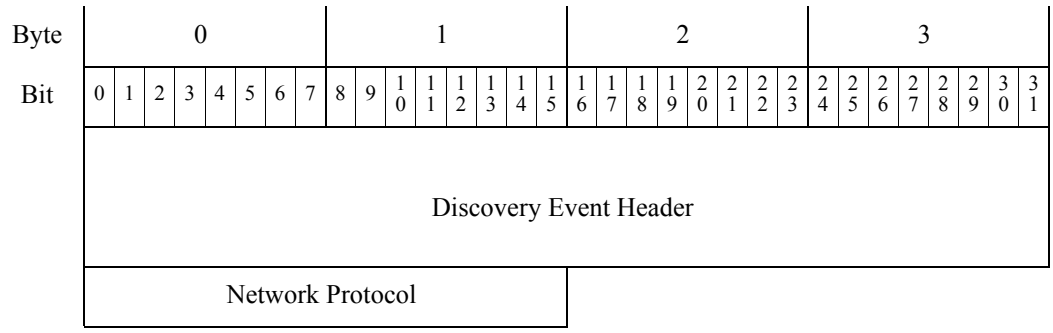
Note The Server data block differs depending on which system version created the message. For information on the legacy versions of the Server data block, see Understanding Legacy Data Structures, page B-1.

Each of these events use the following format:



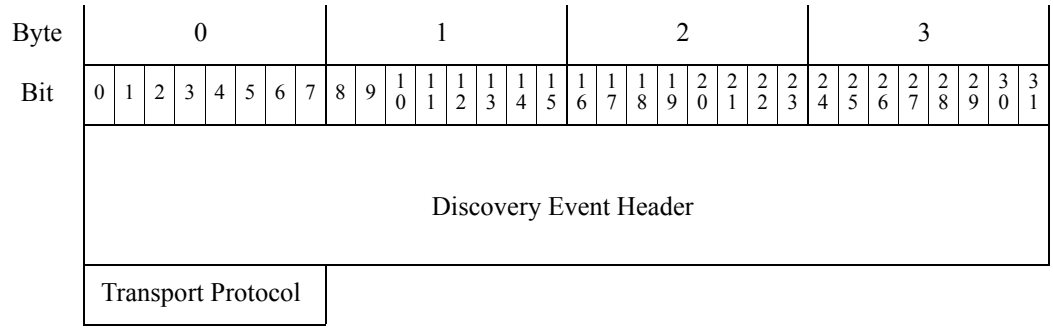
New Network Protocol Message

A New Network Protocol event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a two-byte field for the network protocol (using protocol values described in following table).



New Transport Protocol Message

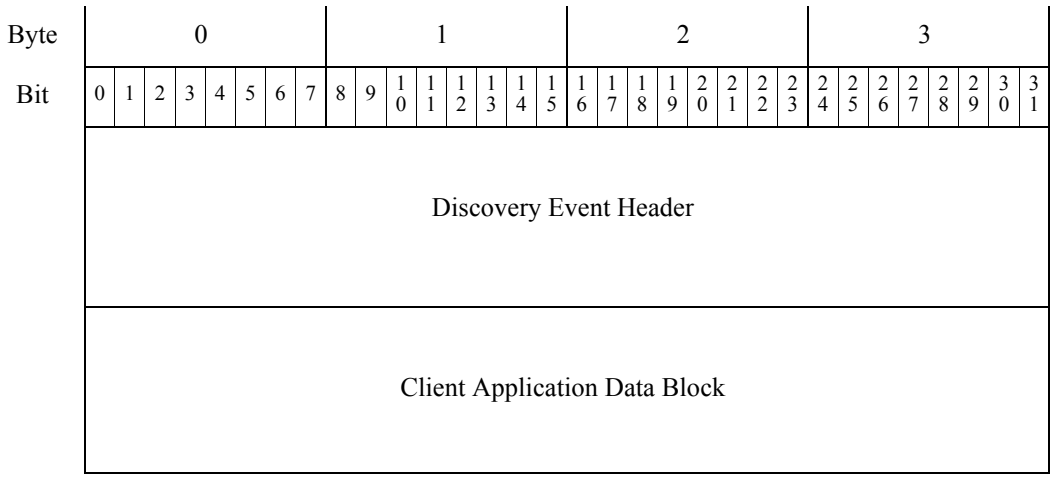
A New Transport Protocol event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34, block type 4 in series 1) and a one-byte field for the transport protocol number (using values described in following table).



Client Application Messages

New Client Application, Client Application Update, and Client Application Timeout events have the same format and contain a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Client Application data block (see Host Client Application Data Block for 5.0+, page 4-145, block type 122 in series 1). The discovery event header has a different record type, event type, and event subtype, depending on the event transmitted.

Note The Client Application data block differs depending on the system version that created the message. For information on the legacy version of the Client Application data block, see Understanding Legacy Data Structures, page B-1.

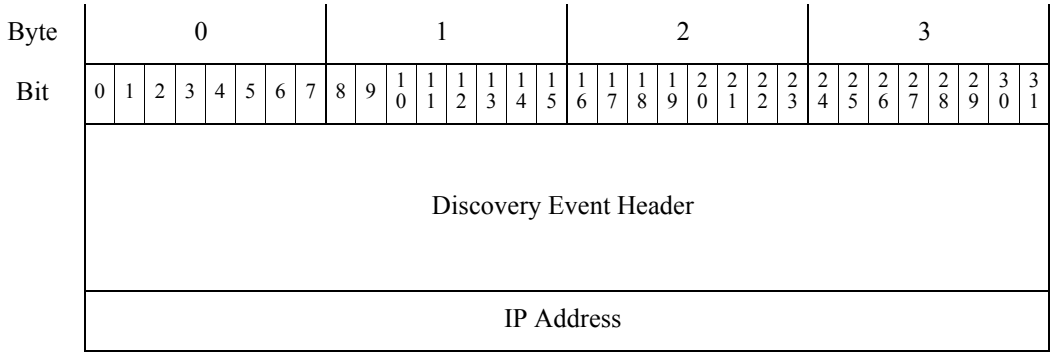


IP Address Change Message

The following host discovery messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) and two different forms, structures, one with four bytes for the IP address and one with 16 bytes for the IP address.

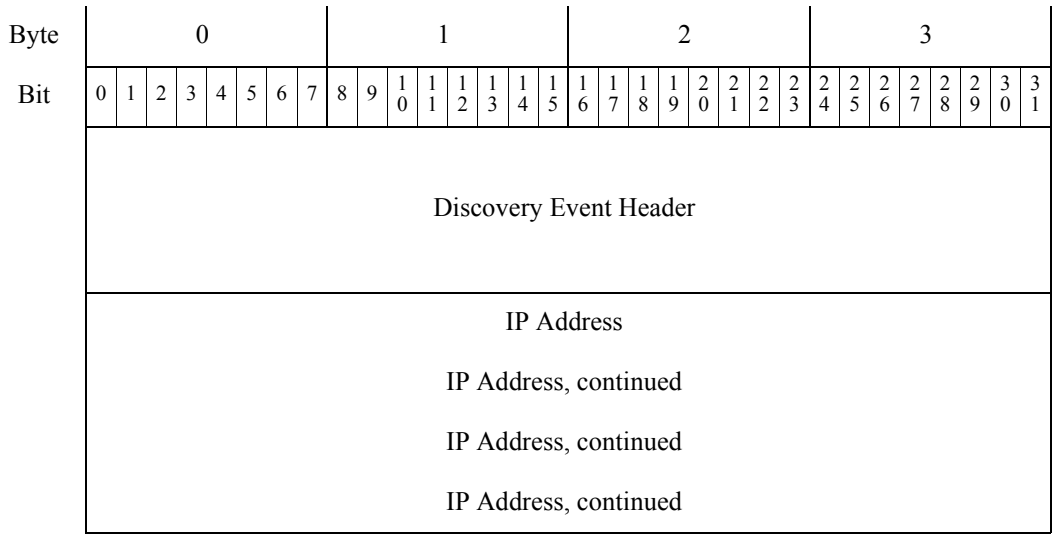
Four bytes are used for the IP address (in IP address octets) in the following case:

- New IPv4 to IPv4 Traffic
- Host IP Address Changed, when the RNA event version is less than 10



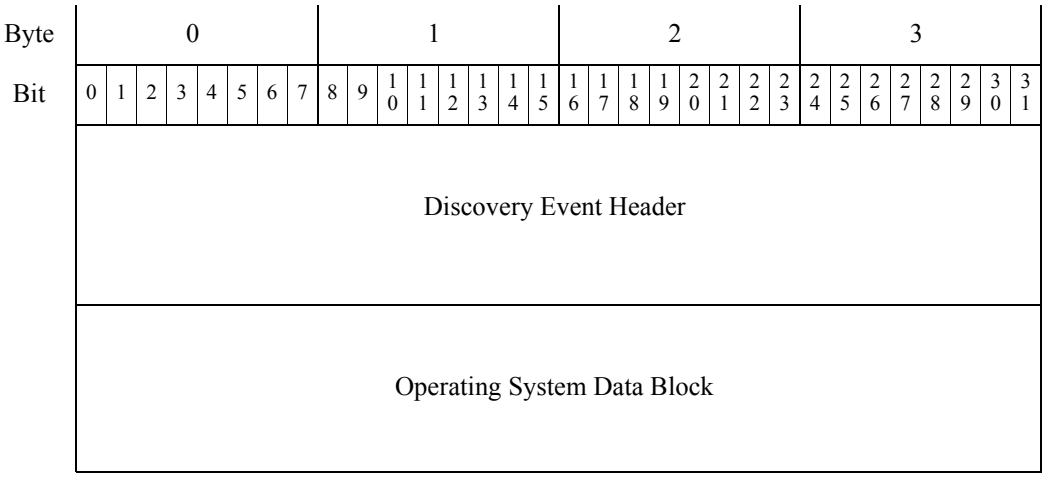
16 bytes are used for the IP address in the following cases:

- New IPv6 to IPv6 Traffic
- Host IP Address Changed, when the RNA event version is 10



Operating System Update Messages

The OS Information Update event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by an Operating System data block (as documented in Operating System Data Block 3.5+, page 4-78, block type 53 in series 1).

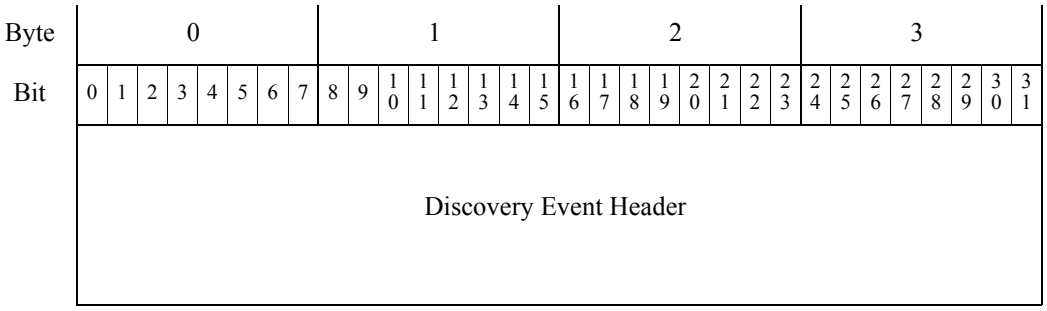


IP Address Reused and Host Timeout/Deleted Messages

The following host event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) with no other data:

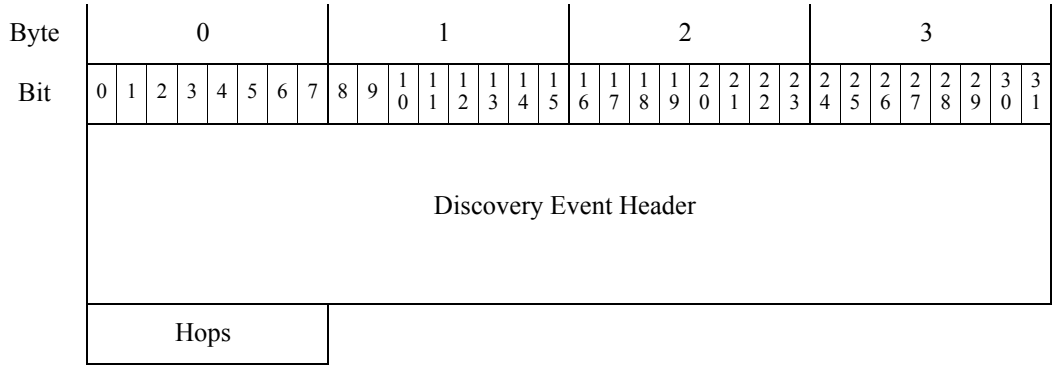
- Host IP Address Reused
- Host Timeout
- Host Deleted: Host Limit Reached

- Host Dropped: Host Limit Reached



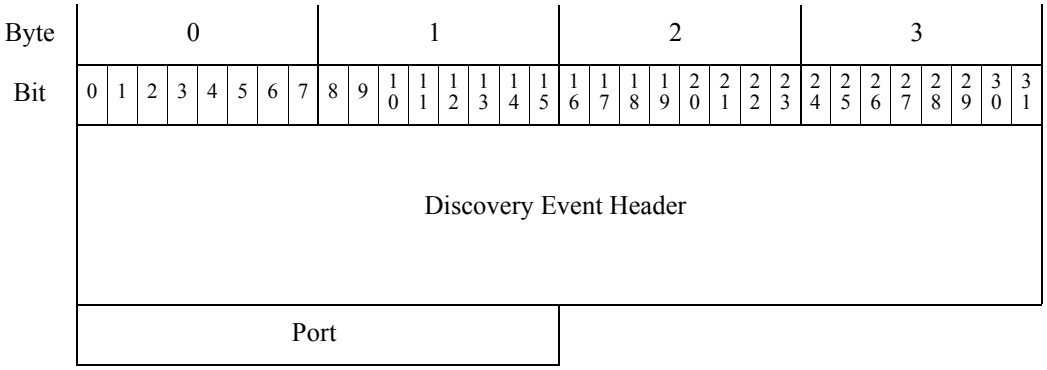
Hops Change Message

A Hops Change event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a one-byte field for the hops count.



TCP and UDP Port Closed/Timeout Messages

TCP and UDP Port Closed and Port Timeout event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a two-byte field for the port number.

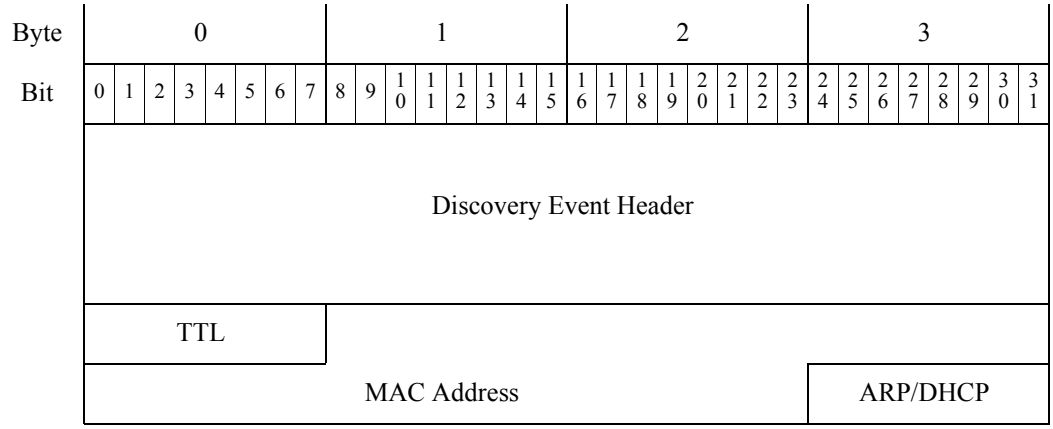


MAC Address Messages

MAC Information Change and Additional MAC Detected for Host messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34), 1 byte for the TTL value, 6 bytes for the MAC address, and 1 byte to indicate whether the MAC address was detected via ARP/DHCP traffic as the actual MAC address.

Note If you receive MAC address messages from a system running version 4.9.x, you must check for the length of the MAC address data block and decode accordingly. If the data block is 8 bytes in length (16 bytes with the header), see MAC Address Messages, page 4-45. If the data block is 12 bytes in length (20 bytes with the header), see Host MAC Address 4.9+, page 4-107.

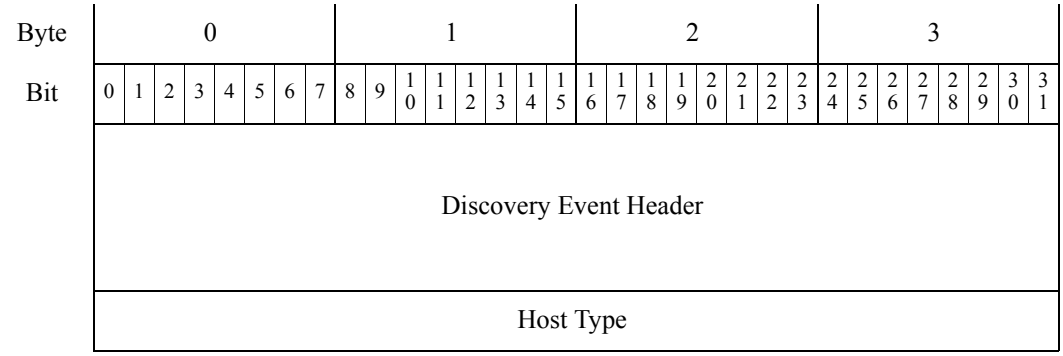
Note that the MAC address data block header is **not** used within MAC Information Change and Additional MAC Detected for Host messages.



Host Identified as a Bridge/Router Message

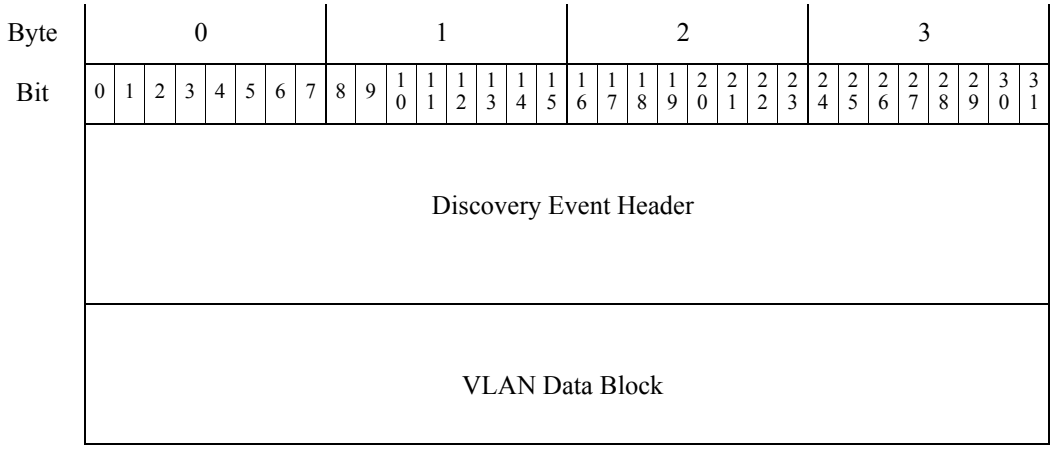
A Host Identified as a Bridge/Router event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a four-byte field for the value that matches the host type:

- `0` — Host
- `1` — Router
- `2` — Bridge



VLAN Tag Information Update Messages

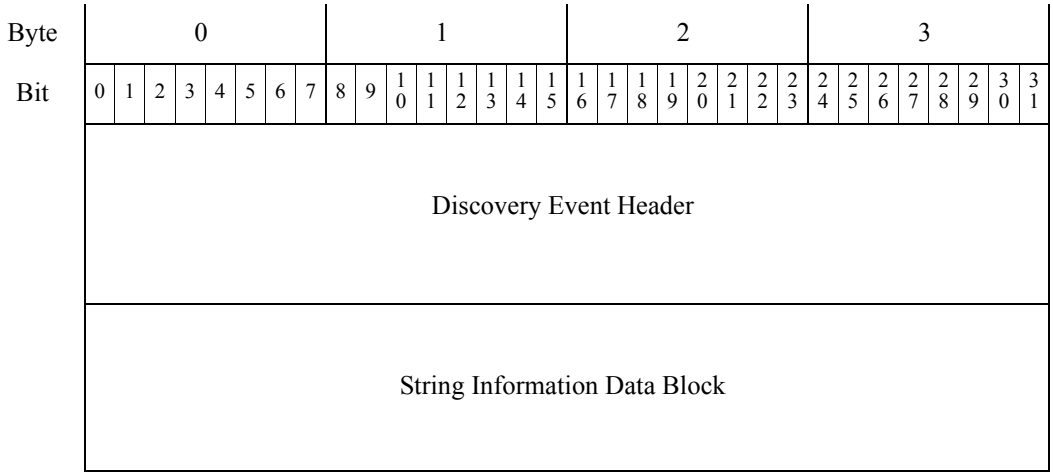
The VLAN Tag Information Update event has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by VLAN data block (as documented in VLAN Data Block, page 4-70). The VLAN Data block is block type 14 in the series 1 group of blocks.



Change NetBIOS Name Message

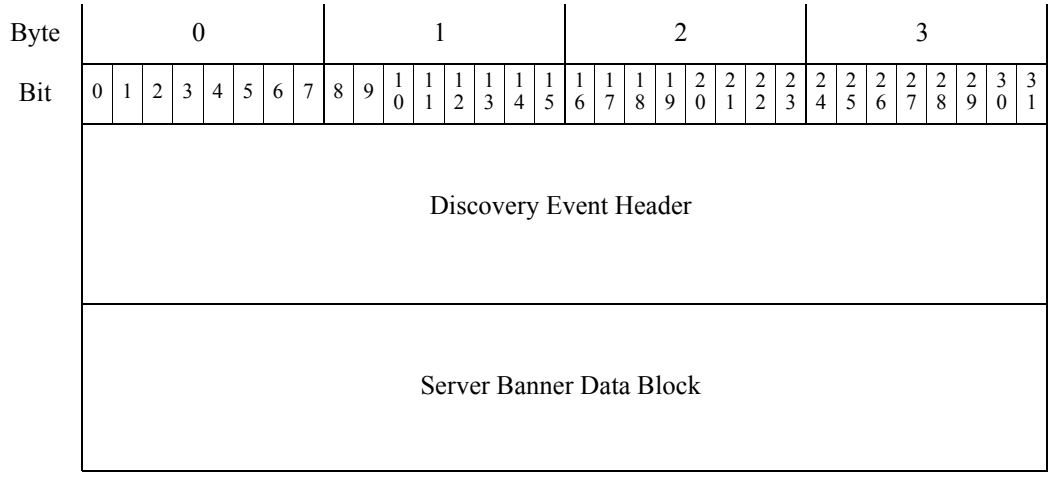
A Change NetBIOS Name event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a String Information data block (as documented in String Information Data Block, page 4-71). The String Information data block is block type 35 in series 1.

Note The Change NetBIOS Domain event is not currently generated by the Firepower System.



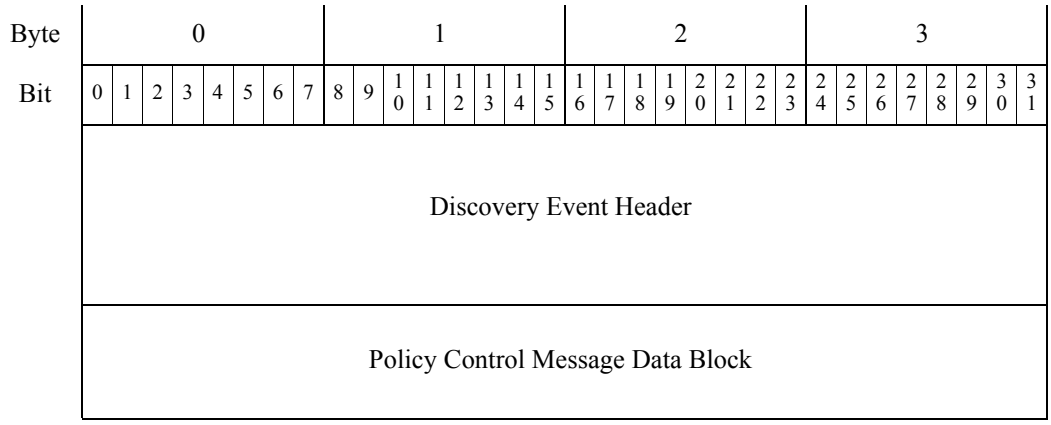
Update Banner Message

An Update Banner event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Server Banner data block (as documented in Server Banner Data Block, page 4-70). The server banner data block is block type 37 in series 1.



Policy Control Message

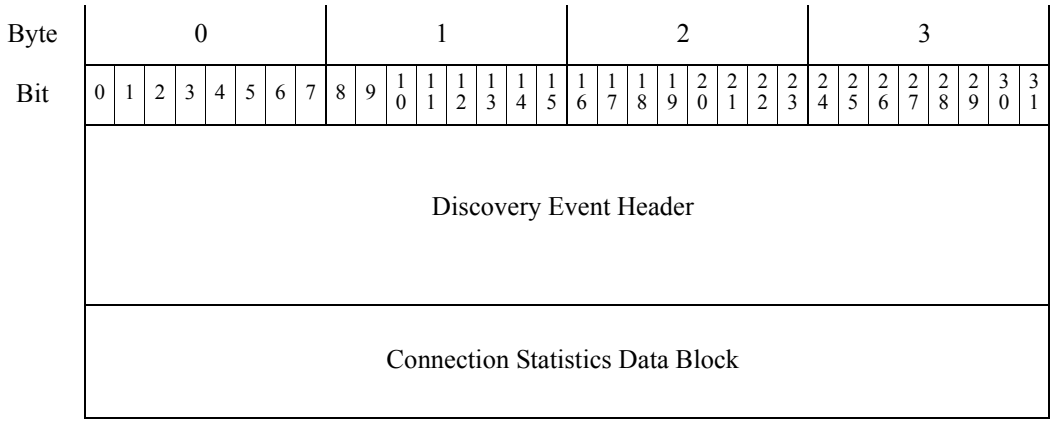
The Policy Control Message event has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Policy Control Message data block. The format of the Policy Control Message data block differs depending on the system version. For information on policy control message data block format for the current version, see Policy Engine Control Message Data Block, page 4-78.



Connection Statistics Data Message

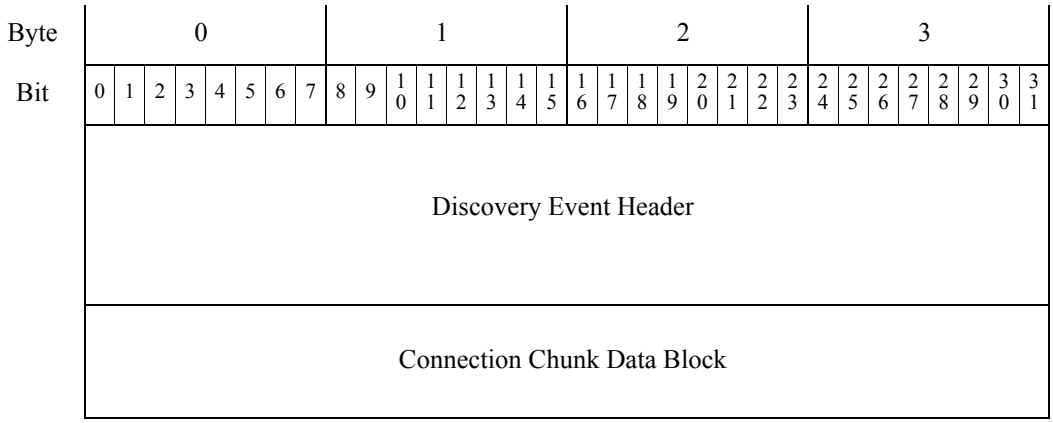
The Connection Statistics event has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Connection Statistics data block. The documentation of each version of the Connection Statistics data block includes the system versions that use it. For information on the connection statistics data block format for version 5.3.1+, see The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116.

Note The Connection Statistics data block differs depending on which system version created the message. For information on legacy versions, see the Connection Statistics data block in Understanding Legacy Data Structures, page B-1.



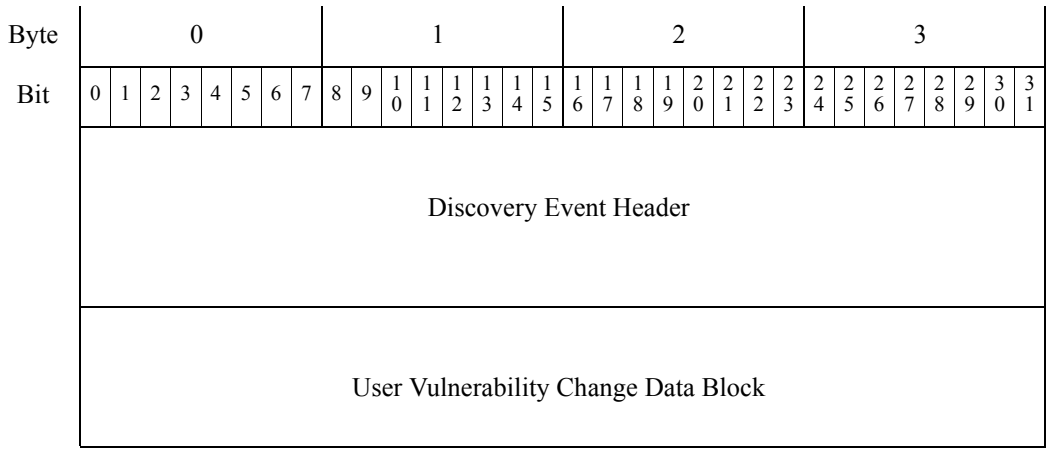
Connection Chunk Message

The Connection Chunk event has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Connection Chunk data block. The format differs depending on the system version. For information on connection chunk data block format for the current version, see Connection Chunk Data Block for 5.1.1+, page 4-92. The Connection Chunk data block is block type 136 in series 1.



User Set Vulnerabilities Messages for Version 4.6.1+

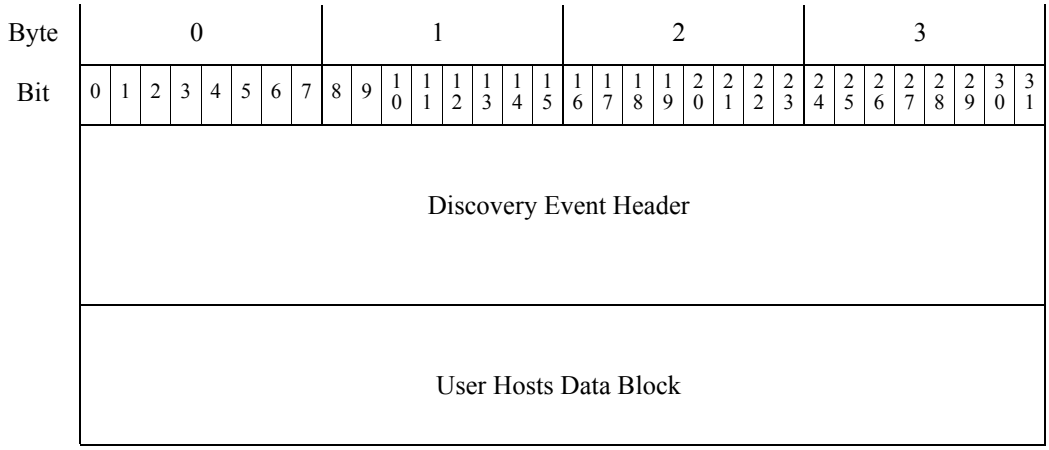
User Set Valid Vulnerabilities, User Set Invalid Vulnerabilities, and User Vulnerability Qualification messages use the same data format: the standard discovery event header (see Discovery Event Header 5.2+, page 4-34) followed by a User Vulnerability change data block (see User Vulnerability Change Data Block 4.7+, page 4-98, block type 80 in series 1). They are differentiated by record type, event type, and event subtype.



User Add and Delete Host Messages

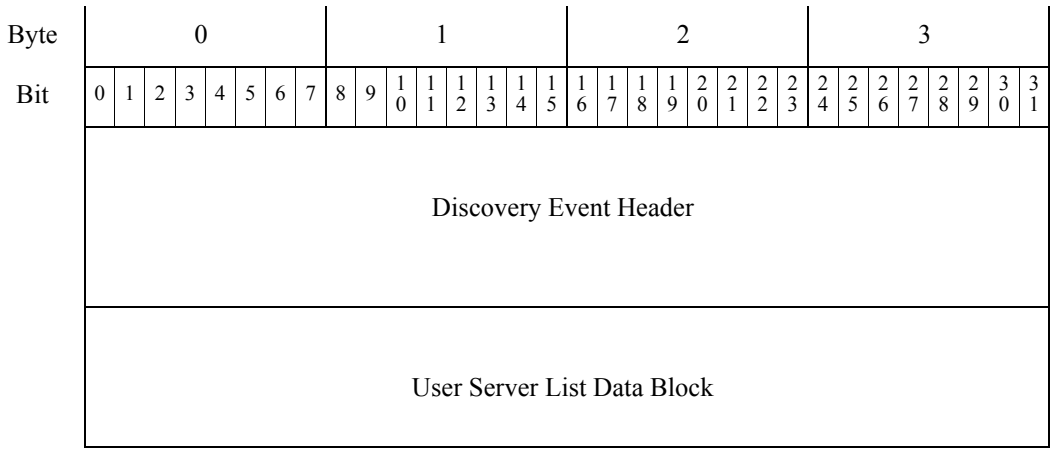
The following host input event messages have the standard discovery event header (see Discovery Event Header 5.2+, page 4-34) followed by a User Hosts data block (see User Hosts Data Block 4.7+, page 4-97, block type 78 in series 1):

- User Delete Address
- User Add Hosts



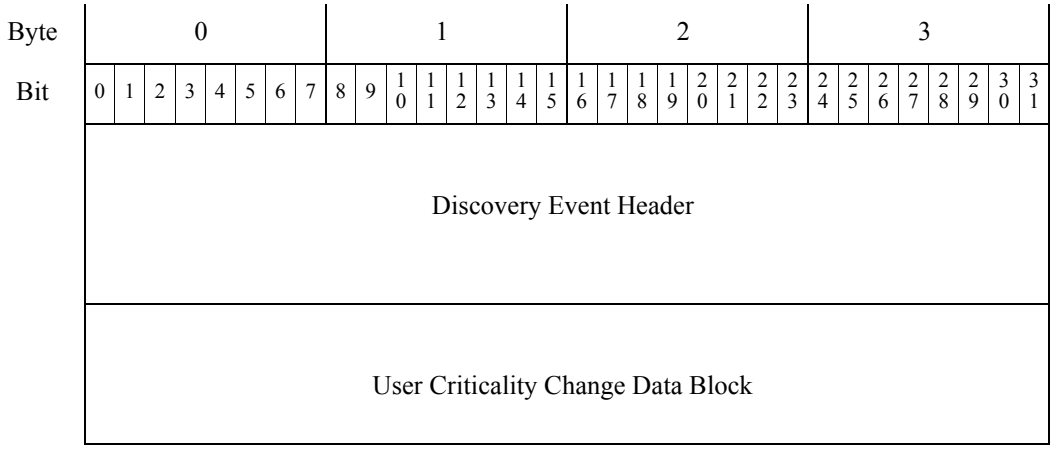
User Delete Server Message

User Delete Server messages have the standard discovery event header (see Discovery Event Header 5.2+, page 4-34) followed by a User Server List data block (see User Server List Data Block, page 4-96). The User Server List data block is block type 77 in series 1.



User Set Host Criticality Messages

User Set Host Criticality messages have the standard discovery event header (see Discovery Event Header 5.2+, page 4-34) followed by a User Criticality Change data block (see User Criticality Change Data Block 4.7+, page 4-100). The User Criticality Change data block is block type 81 in series 1.

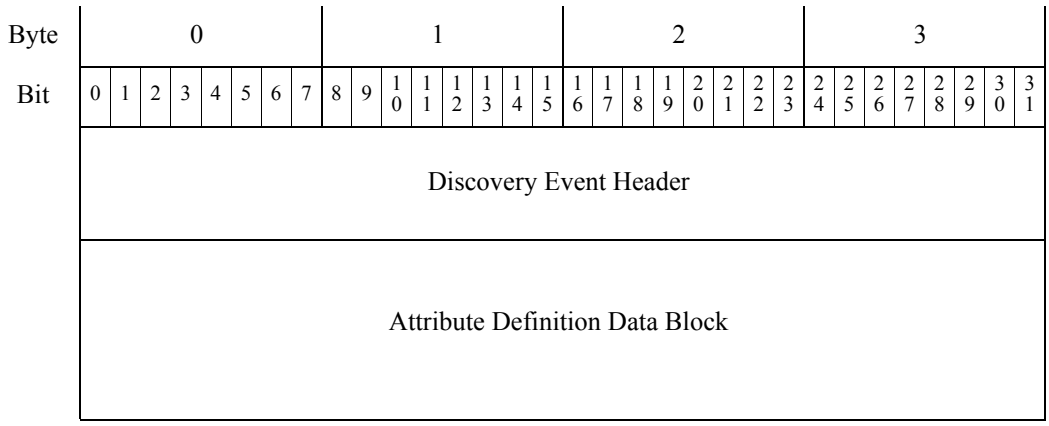


Attribute Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by an Attribute Definition data block (as documented in Attribute Definition Data Block for 4.7+, page 4-79, block type 55 in series 1):

- Add Host Attribute
- Update Host Attribute
- Delete Host Attribute

Each of these events use the following format:

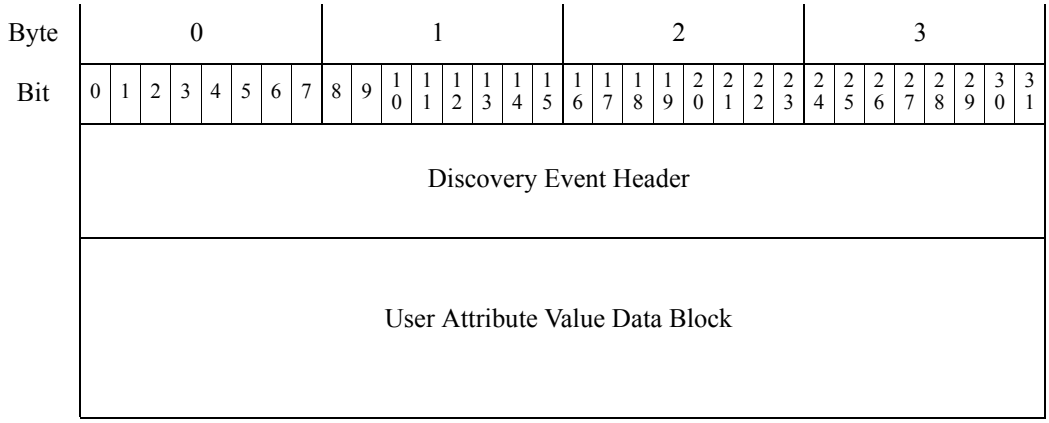


Attribute Value Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a User Attribute Value data block (as documented in User Attribute Value Data Block 4.7+, page 4-101, block type 82 in series 1):

- Set Host Attribute Value
- Delete Host Attribute Value

Each of these events use the following format:

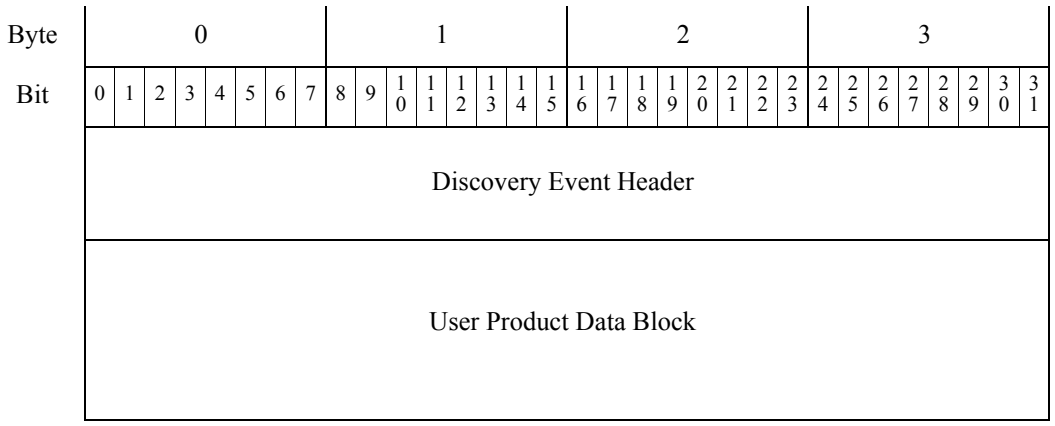


User Server and Operating System Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a User Product data block (as documented in User Product Data Block 5.1+, page 4-160, block type 60 in series 1):

- Set Operating System Definition
- Set Server Definition
- Add Server

Each of these events use the following format:

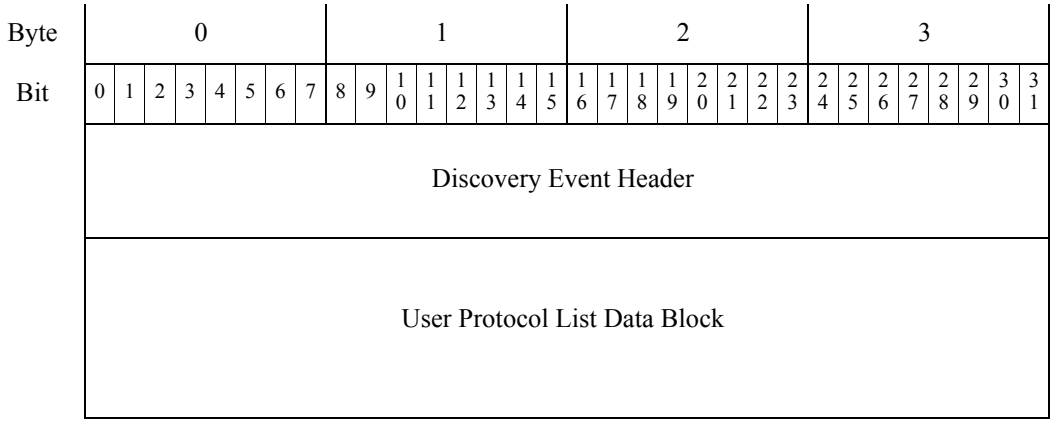


User Protocol Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a User Protocol List data block (as documented in User Protocol List Data Block 4.7+, page 4-103, block type 83 in series 1):

- Delete Protocol
- Add Protocol

Each of these events use the following format:

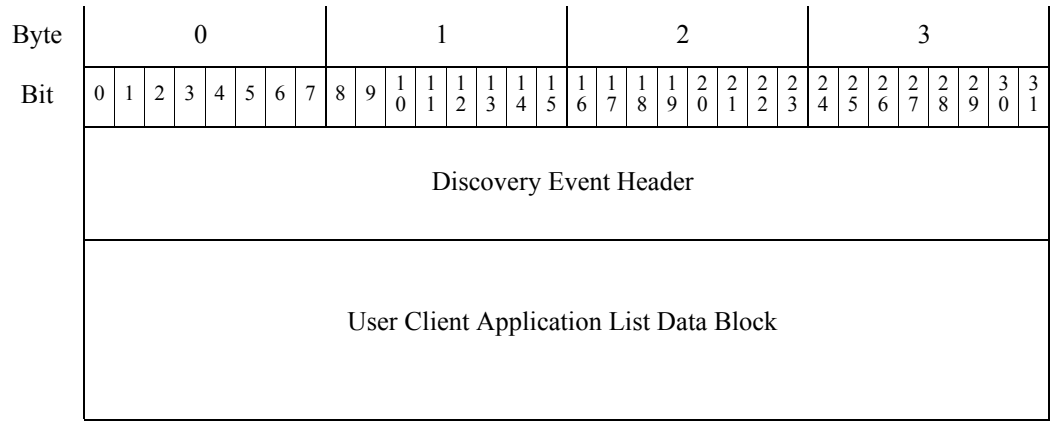


User Client Application Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a User Client Application List data block (as documented in User Client Application List Data Block, page 4-85, block type 60 in series 1):

- Delete Client Application
- Add Client Application

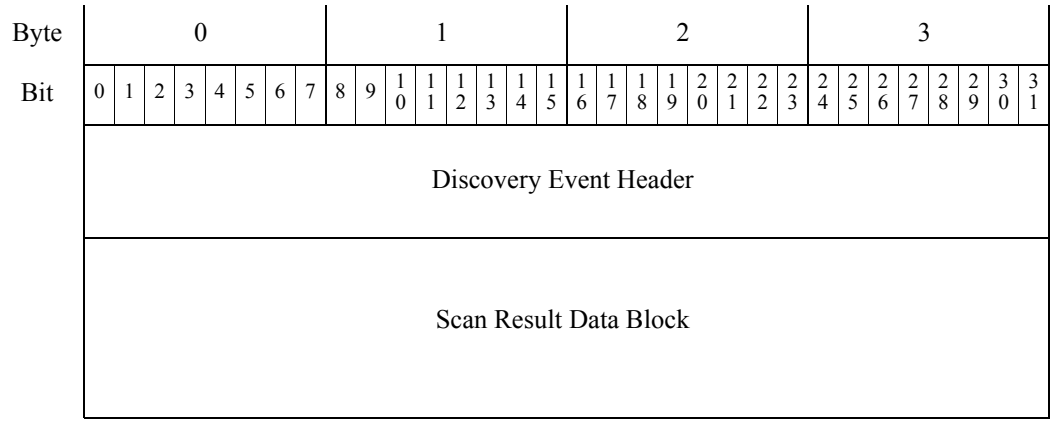
Each of these events use the following format:



Add Scan Result Messages

The Add Scan Result event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by a Scan Results data block (as documented in The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116). The Scan Result data block is block type 142 in series 1.

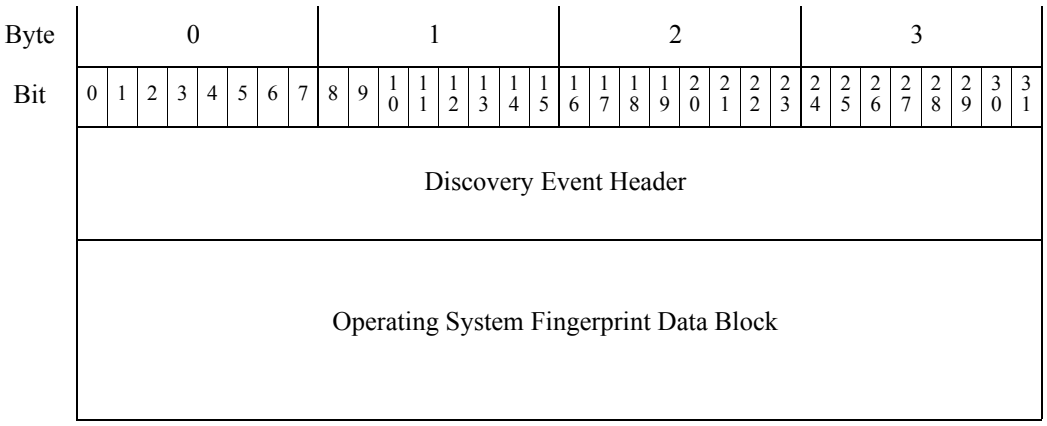
This event uses the following format:



New Operating System Messages

The New OS event message has a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by an Operating System Fingerprint data block (as documented in Operating System Fingerprint Data Block 5.1+, page 4-149).

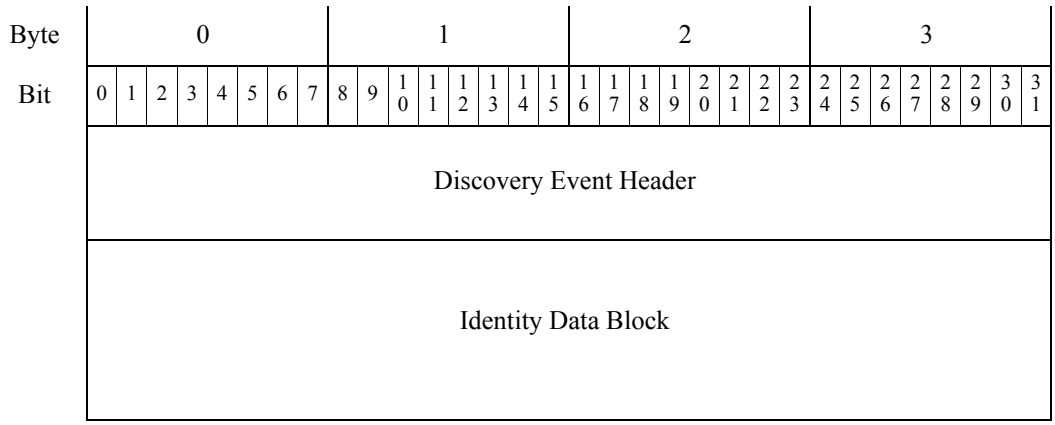
This event uses the following format:



Identity Conflict and Identity Timeout System Messages

The Identity Conflict and Identity Timeout event messages each have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) followed by an Identity data block (as documented in Identity Data Block, page 4-105). The Identity data block is block type 94 in series 1. These messages are generated when there are conflicts or timeouts in a fingerprint source identity.

This event uses the following format:



User Data Structures by Event Type

eStreamer builds user event messages based on the event type indicated in the discovery event header. The following sub-sections describe the high-level structure for each event type:

- User Modification Messages, page 4-54
- User Information Update Message Block, page 4-55

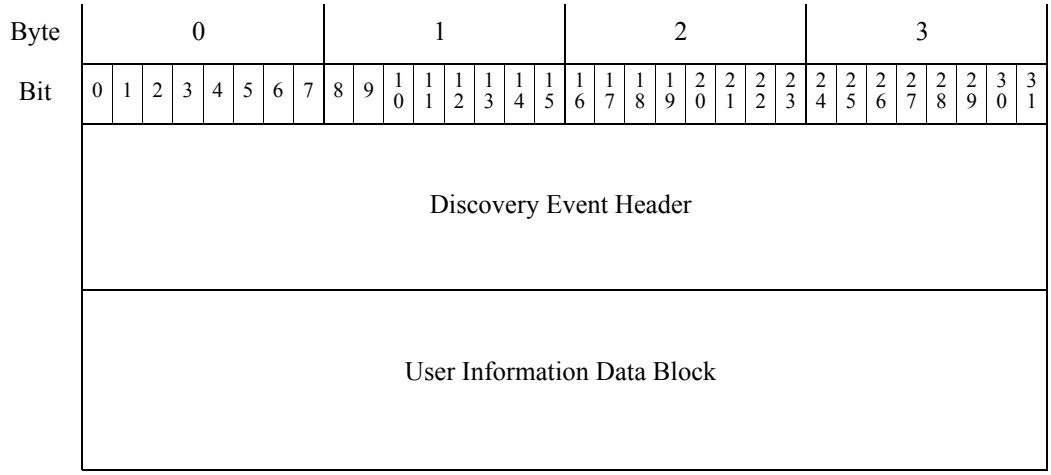
User Modification Messages

When any of the following events occurs through system detection, a user modification message is sent:

- a new user is detected (a New User Identity event—event type 1004, subtype 1)

- a user is removed (a Delete User Identity event—event type 1004, subtype 3)
- a user is dropped (a User Identity Dropped: User Limit Reached event—event type 1004, subtype 4)

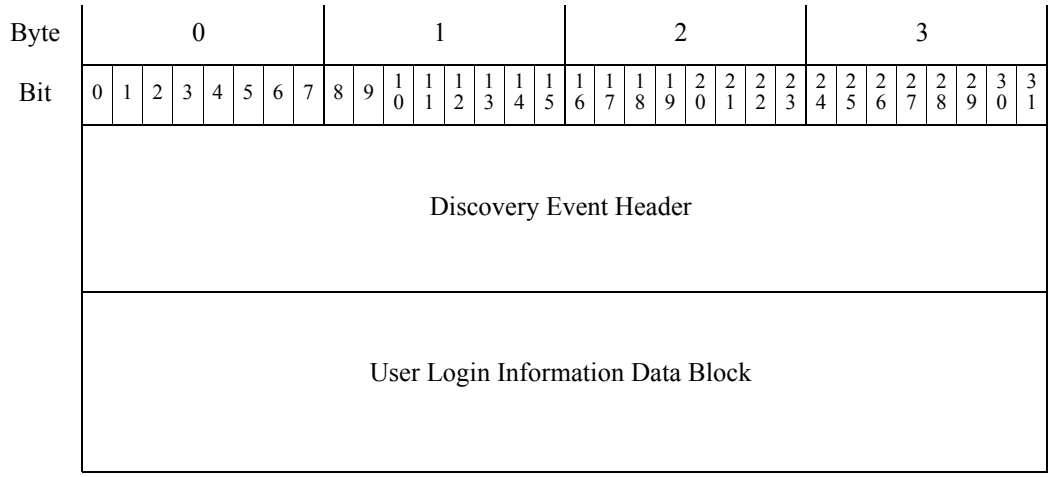
User Modification event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) and a User Information data block (as documented in User Information Data Block for 6.0+, page 4-177). The User Information data block is block type 120 in series 1.



User Information Update Message Block

When the login changes for a user (a User Login event—event type 1004, subtype 2) detected by the system, a user information update message is sent.

User Information Update event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+, page 4-34) and a User Login Information data block (as documented in User Login Information Data Block 6.0+, page 4-179). The User Login Information data block is block type 121 in series 1.

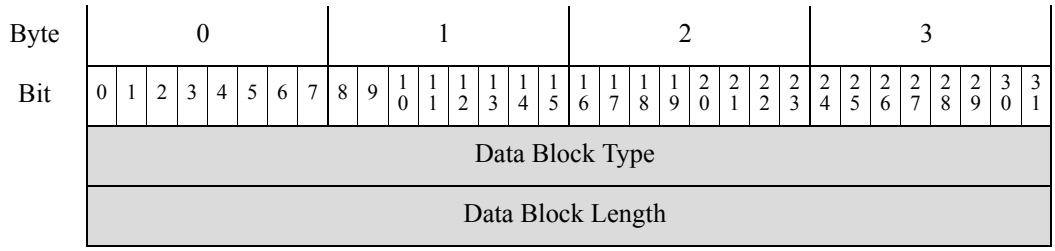


Understanding Discovery (Series 1) Blocks

Most discovery and connection events incorporate one or more data blocks from the series 1 group of data structures. Each series 1 data block type conveys a particular type of information. The block type number appears in the data block header which precedes the data in the block. For information on block header format, see Data Block Header, page 2-24.

Series 1 Data Block Header

The series 1 data block header, like the series 2 block header, has two 32-bit integer fields that contain the block's type number and the block length.



Note The data block length field contains the number of bytes in the entire data block, including the eight bytes of the two data block header fields.

For some block series 1 types, the block header is followed immediately by raw data. In more complex block types, the header may be followed by standard fixed length fields or by the header of a series 1 primitive block that encapsulates another series 1 data block or list of blocks.

Series 1 Primitive Data Blocks

Both series 1 and series 2 blocks include a set of primitives that encapsulate lists of variable-length blocks as well as variable-length strings and BLOBs within messages. These primitive blocks have the standard series 1 block header discussed above. These primitives appear only within other series 1 data blocks. Any number can be included in a given block type. For details on the structure of the primitive blocks, see the following:

- String Data Block, page 4-64
- BLOB Data Block, page 4-65
- List Data Block, page 4-66
- Generic List Block, page 4-66

Host Discovery and Connection Data Blocks

For the list of block types in host discovery and connection events, see Table 4-27 on page 4-57. The block types in user events are described in Table 4-82 on page 4-167. These are all Series 1 data blocks.

Each entry in the table below contains a link to the subsection where the data block is defined. For each block type, the status (current or legacy) is indicated. A current data block is the latest version. A legacy data block is one that is used for an older version of the product, and the message format can still be requested from eStreamer.

Table 4-27 Host Discovery and Connection Data Block Types

Type	Content	Data Block Status	Description
0	String	Current	Contains string data. See String Data Block, page 4-64 for more information.
1	Sub-Server	Current	Contains information about a sub-server detected on a server. See Sub-Server Data Block, page 4-67 for more information.
4	Protocol	Current	Contains protocol data. See Protocol Data Block, page 4-68 for more information.
7	Integer Data	Current	Contains integer (numeric) data. See Integer (INT32) Data Block, page 4-69 for more information.
10	BLOB	Current	Contains a raw block of binary data and is used specifically for banners. See BLOB Data Block, page 4-65 for more information.
11	List	Current	Contains a list of other data blocks. See List Data Block, page 4-66 for more information.
14	VLAN	Current	Contains VLAN information. See VLAN Data Block, page 4-70 for more information.
20	Intrusion Impact Alert	Current	Contains intrusion impact alert information. Intrusion impact alert events have slightly different headers than other data blocks. See Intrusion Impact Alert Data 5.3+, page 3-16 for more information.
31	Generic List	Current	Contains generic list information, for example, to encapsulate lists of blocks, such as Client Application blocks, in the Host Profile block. See Generic List Block, page 4-66 for more information.
35	String Information	Current	Contains string information. For example, when used in the Scan Vulnerability data block, the String Information data block contains the CVE identification number data. See String Information Data Block, page 4-71.
37	Server Banner	Current	Contains server banner data. See Server Banner Data Block, page 4-70 for more information.
38	Attribute Address	Legacy	Contains the host attribute address (as documented in earlier versions of the product). The successor block is 146.
39	Attribute List Item	Current	Contains a host attribute list item value. See Attribute List Item Data Block, page 4-73 for more information.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
42	Host Client Application	Legacy	Contains client application information for New Client Application events (as documented for earlier versions of the product).
47	Full Host Profile	Legacy	Contains complete host profile information (as documented in earlier versions of the product).
48	Attribute Value	Current	Contains attribute identification numbers and values for host attributes. See Attribute Value Data Block, page 4-74 for more information.
51	Full Sub-Server	Current	Contains information about a sub-server detected on a server. Referenced in Full Server information blocks and in full host profiles. Includes vulnerability information for each sub-server. See Full Sub-Server Data Block, page 4-75 for more information.
53	Operating System	Current	Contains operating system information for Version 3.5+. See Operating System Data Block 3.5+, page 4-78 for more information.
54	Policy Engine Control Message	Current	Contains information on user policy control changes. See Policy Engine Control Message Data Block, page 4-78 for more information.
55	Attribute Definition	Current	Contains information on attribute definitions. See Attribute Definition Data Block for 4.7+, page 4-79 for more information.
56	Connection Statistics	Legacy	Contains information for connection statistics events in 4.7 - 4.9.0 (as documented in earlier versions of the product).
57	User Protocol	Current	Contains protocol information from user input. See User Protocol Data Block, page 4-82 for more information.
	User Client Application	Legacy	Contains client application data from user input. See User Client Application Data Block for 5.0 - 5.1, page B-90 for more information. Superseded by block 138.
60	User Client Application List	Current	Contains lists of user client application data blocks. See User Client Application List Data Block, page 4-85 for more information.
61	IP Range Specification	Legacy	Contains IP address range specifications. See IP Range Specification Data Block for 5.0 - 5.1.1.x, page B-260 for more information. Superseded by block 141.
	Attribute Specification	Current	Contains an attribute name and value. See Attribute Specification Data Block, page 4-88 for more information.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
63	MAC Address Specification	Current	Contains MAC address range specifications. See MAC Address Specification Data Block, page 4-90 for more information.
64	IP Address Specification	Current	Contains lists of IP and MAC address specification blocks. See Address Specification Data Block, page 4-91 for more information.
65	User Product	Legacy	Contains host input data imported from a third-party application, including third-party application string mappings. See User Product Data Block for 5.0.x, page B-94 for more information. The successor block type 118 introduced for 5.0 has an identical structure as block type 65.
66	Connection Chunk	Legacy	Contains connection chunk information. See Connection Chunk Data Block for 5.0 - 5.1, page B-131 for more information. The successor block type 119 introduced for 5.0 has an identical structure as block type 66.
67	Fix List	Current	Contains a fix that applies to a host. See Fix List Data Block, page 4-94 for more information.
71	Generic Scan Results	Legacy	Contains results from an Nmap scan (as documented in earlier versions of the product).
72	Scan Result	Legacy	Contains results from a third-party scan (as documented in earlier versions of the product).
76	User Server	Current	Contains server information from a user input event. See User Server Data Block, page 4-94 for more information.
77	User Server List	Current	Contains lists of user server blocks. See User Server List Data Block, page 4-96 for more information.
78	User Hosts	Current	Contains information about host ranges from a user host input event. See User Hosts Data Block 4.7+, page 4-97 for more information.
79	User Vulnerability	Legacy	Contains information about a vulnerability for a host or hosts (as documented in earlier versions of the product). The successor block introduced for version 5.0 has block type 124.
80	User Host Vulnerability Change	Current	Contains lists of deactivated or activated vulnerabilities. See User Vulnerability Change Data Block 4.7+, page 4-98 for more information.
81	User Criticality	Current	Contains information on criticality changes for a host or host. See User Criticality Change Data Block 4.7+, page 4-100 for more information.
82	User Attribute Value	Current	Contains attribute value changes for a host or hosts. See User Attribute Value Data Block 4.7+, page 4-101 for more information.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
83	User Protocol List	Current	Contains lists of protocols for a host or hosts. See User Protocol List Data Block 4.7+, page 4-103 for more information.
85	Vulnerability List	Current	Contains vulnerabilities that apply to a host. See Host Vulnerability Data Block 4.9.0+, page 4-104 for more information.
86	Scan Vulnerability	Legacy	Contains information on vulnerabilities detected by a scan (as documented in earlier versions of the product).
87	Operating System Fingerprint	Legacy	Contains lists of operating system fingerprints. See Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-113 for more information. The successor block introduced for version 5.1 has block type 130.
88	Server Information	Legacy	Contains server information used in server fingerprints (as documented in earlier versions of the product).
89	Host Server	Legacy	Contains server information for a host (as documented in earlier versions of the product).
90	Full Host Server	Legacy	Contains server information for a host (as documented in earlier versions of the product).
91	Host Profile	Legacy	Contains profile information for a host. See Host Profile Data Block for 5.2+, page 4-152 for more information. The successor block introduced for version 5.1 has block type 132.
92	Full Host Profile	Legacy	Contains complete host profile information (as documented in earlier versions of the product). Supersedes data block 47.
94	Identity Data	Current	Contains identity data for a host. See Identity Data Block, page 4-105 for more information.
95	Host MAC Address	Current	Contains MAC address information for a host. See Host MAC Address 4.9+, page 4-107 for more information.
96	Secondary Host Update	Current	Contains lists of MAC address information reported by a secondary Secondary Host Update, page 4-108.
97	Web Application	Legacy	Contains lists of web application data (as documented in earlier versions of the product). The successor block introduced for version 5.0 has block type 123.
98	Host Server	Legacy	Contains server information for a host (as documented in earlier versions of the product).
99	Full Host Server	Legacy	Contains server information for a host (as documented in earlier versions of the product).

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
100	Host Client Application	Legacy	Contains client application information for New Client Application events (as documented in earlier versions of the product). The successor block type 122 introduced for version 5.0 has the same structure as block type 100.
101	Connection Statistics	Legacy	Contains information for connection statistics events in 4.9.1+ (as documented in earlier versions of the product).
102	Scan Results	Legacy	Contains information about a vulnerability and is used within Add Scan Result events. See Scan Result Data Block 5.0 - 5.1.1.x, page B-92.
103	Host Server	Current	Contains server information for a host. See Host Server Data Block 4.10.0+, page 4-128 for more information.
104	Full Host Server	Current	Contains server information for a host. See Full Host Server Data Block 4.10.0+, page 4-130 for more information.
105	Server Information	Legacy	Contains server information used in server fingerprints. See Server Information Data Block for 4.10.x, 5.0 - 5.0.2, page 4-134 for more information. The successor block type 117 introduced for 5.0 has an identical structure as block type 105.
106	Full Server Information	Current	Contains information about a server detected on a host. See Full Server Information Data Block, page 4-136 for more information.
108	Generic Scan Results	Current	Contains results from an Nmap scan. See Generic Scan Results Data Block for 4.10.0+, page 4-138 for more information.
109	Scan Vulnerability	Current	Contains information on vulnerabilities detected by a third-party scan. See Scan Vulnerability Data Block for 4.10.0+, page 4-141.
111	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.0 - 5.0.2, page B-224 for more information. Supersedes data block 92.
112	Full Host Client Application	Current	Contains client application information for New Client Application events and includes a list of vulnerabilities. See Full Host Client Application Data Block 5.0+, page 4-143 for more information.
115	Connection Statistics	Legacy	Contains information for connection statistics events in 5.0 - 5.0.2. See Connection Statistics Data Block 5.0 - 5.0.2, page B-114 for more information. The successor block introduced for version 5.1 has block type 126.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
117	Server Information	Current	Contains server information used in server fingerprints. See Server Information Data Block for 4.10.x, 5.0 - 5.0.2, page 4-134 for more information.
118	User Product	Legacy	Contains host input data imported from a third-party application, including third-party application string mappings. See User Product Data Block for 5.0.x, page B-94 for more information. The predecessor block type 65, superseded in 5.0, has the same structure as this block type. The successor block introduced for version 5.1 has block type 132.
119	Connection Chunk	Legacy	Contains connection chunk information for versions 4.10.1 - 5.1. See Connection Chunk Data Block for 5.0 - 5.1, page B-131 for more information. The successor block is 136.
122	Host Client Application	Current	Contains client application information for New Client Application events for version 5.0+. See Host Client Application Data Block for 5.0+, page 4-145 for more information. It supersedes block type 100.
123	Web Application	Current	Contains web application data for version 5.0+. See Web Application Data Block for 5.0+, page 4-109 for more information. It supersedes block type 97.
124	User Vulnerability	Current	Contains information about a vulnerability for a host or hosts. See User Vulnerability Data Block 5.0+, page 4-147. It supersedes block type 79.
125	Connection Statistics	Legacy	Contains information for connection statistics events in 4.10.2 (as documented in earlier versions of the product). The successor block introduced for version 5.1 has block type 115.
126	Connection Statistics	Legacy	Contains information for connection statistics events in 5.1. See Connection Statistics Data Block 5.1, page B-119 for more information. It supersedes block type 115. This block type is superseded by block type 137.
130	Operating System Fingerprint	Current	Contains lists of operating system fingerprints. See Operating System Fingerprint Data Block 5.1+, page 4-149 for more information. It supersedes block type 87.
131	Mobile Device Information	Current	Contains information about a detected mobile device's hardware. See Mobile Device Information Data Block for 5.1+, page 4-151 for more information.
132	Host Profile	Legacy	Contains profile information for a host. See Full Host Profile Data Block 5.2.x, page B-242 for more information. It supersedes block type 91. Superseded by block 139.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
134	User Product	Current	Contains host input data imported from a third-party application, including third-party application string mappings. See User Product Data Block 5.1+, page 4-160 for more information. This supersedes the predecessor block type 118.
135	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.1.1, page B-233 for more information. Supersedes data block 111.
136	Connection Chunk	Current	Contains connection chunk information. See Connection Chunk Data Block for 5.1.1+, page 4-92 for more information. Supersedes block 119.
137	Connection Statistics	Legacy	Contains information for connection events in 5.1.1. See Connection Chunk Data Block for 5.0 - 5.1, page B-131 for more information. It supersedes block type 126. It is superseded by block type 144.
138	User Client Application	Current	Contains client application data from user input. See User Client Application Data Block for 5.1.1+, page 4-84 for more information. It supersedes block type .
139	Host Profile	Current	Contains profile information for a host. See Host Profile Data Block for 5.2+, page 4-152 for more information. It supersedes block type 132.
140	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.3+, page 5-1 for more information. Supersedes data block 135.
141	IP Range Specification	Current	Contains IP address range specifications. See IP Address Range Data Block for 5.2+, page 4-87 for more information. It supersedes block 61.
142	Scan Results	Current	Contains information about a vulnerability and is used within Add Scan Result events. See The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116. It supersedes block 102.
143	Host IP	Current	Contains a host's IP address and last seen information. See Host IP Address Data Block, page 4-89 for more information.
144	Connection Statistics	Legacy	Contains information for connection events in 5.2.x. See Connection Statistics Data Block 5.2.x, page B-125 for more information. It supersedes block type 137.
146	Attribute Address	Current	Contains the host attribute address for 5.2+. See Attribute Address Data Block 5.2+, page 4-72 for more information. It supersedes block type 38.

Table 4-27 Host Discovery and Connection Data Block Types (continued)

Type	Content	Data Block Status	Description
140	Full Host Profile	Current	Contains complete host profile information. See Full Host Profile Data Block 5.3+, page 5-1 for more information. Supersedes data block 135.
152	Connection Statistics	Legacy	Contains information for connection events in 5.3+. See Connection Statistics Data Block 5.3, page B-138 for more information. It supersedes block type 144.
154	Connection Statistics	Legacy	Contains information for connection events in 5.3. See Connection Statistics Data Block 5.3.1, page B-145 for more information. It supersedes block type 152.
155	Connection Statistics	Legacy	Contains information for connection events in 5.4. See Connection Statistics Data Block 5.4, page B-152 for more information. It supersedes block type 154.
157	Connection Statistics	Legacy	Contains information for connection events in 5.4.1. See Connection Statistics Data Block 5.4.1, page B-165 for more information. It supersedes block type 155.
160	Connection Statistics	Current	Contains information for connection events in 6.0+. See Connection Statistics Data Block 6.0+, page 4-110 for more information. It supersedes block type 157.

String Data Block

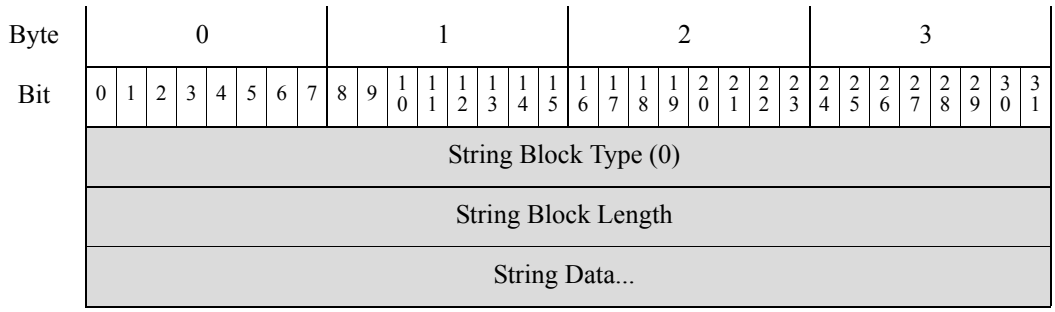
The String data block is used for sending string data in series 1 blocks. It commonly appears within other series 1 data blocks to describe, for example, operating system or server names.

Empty string data blocks (string data blocks containing no string data) have a block length value of 8 and are followed by zero bytes of string data. An empty string data block is returned when there is no content for the string value, as might happen, for example, in the OS vendor string field in an Operating System data block when the vendor of the operating system is unknown.

The String data block has a block type of 0 in the series 1 group of blocks.

Note Strings returned in this data block are not always null-terminated (that is, they are not always terminated with a 0).

The following diagram shows the format of the String data block:



The following table describes the fields of the String data block.

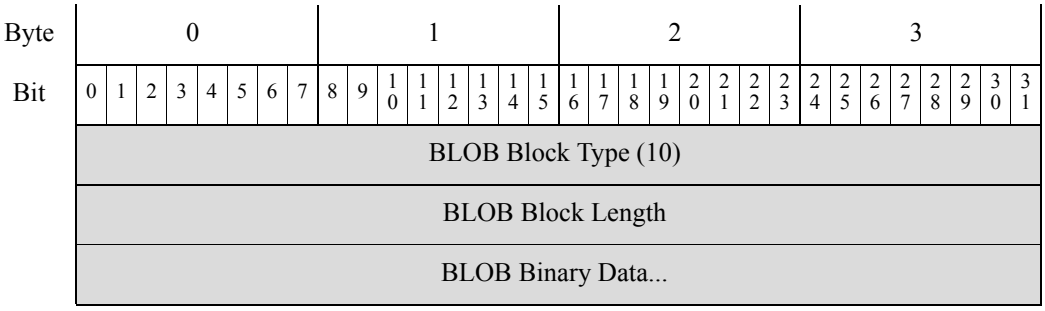
Table 4-28 String Data Block Fields

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block. This value is always `0`.
String Block Length	uint32	Combined length of the string data block header and string data.
String Data	string	Contains the string data and may contain a terminating character (null byte) at the end of the string.

BLOB Data Block

The BLOB data block can be used to convey binary data. For example, it is used to hold the server banner captured by the system. The BLOB data block has a block type of 10 in the series 1 group of blocks.

The following diagram shows the format of the BLOB data block:



The following table describes the fields of the BLOB data block.

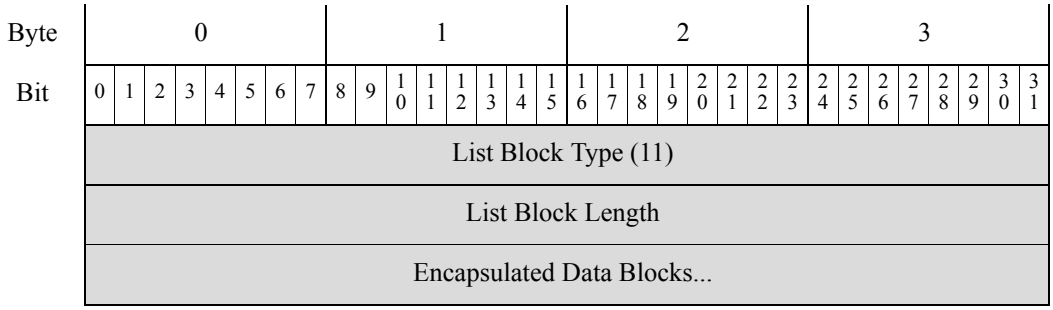
Table 4-29 BLOB Data Block Fields

Field	Data Type	Description
BLOB Block Type	uint32	Initiates a BLOB data block. This value is always `10`.
BLOB Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Binary Data	variable	Contains binary data, typically a server banner.

List Data Block

The List data block is used to encapsulate a list of series 1 data blocks. For example, if a list of TCP servers is being transmitted, the Server data blocks containing the data are encapsulated in a List data block. The List data block has a block type of 11 in the series 1 group of blocks.

The following diagram shows the basic format of a List data block:



The following table describes the fields of the List data block.

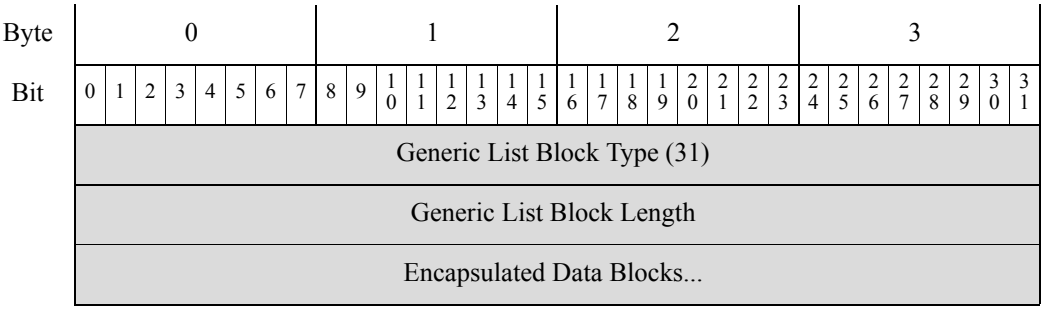
Table 4-30 List Data Block Fields

Field	Data Type	Description
List Block Type	uint32	Initiates a List data block. This value is always `11`.
List Block Length	uint32	Number of bytes in the list block and encapsulated data. For example, if there were three sub-server data blocks included in the list, the value here would include the number of bytes in the sub-server blocks, plus eight bytes for the list block header.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Generic List Block

The Generic List data block is used to encapsulate a list of series 1 data blocks. For example, when client application information is transmitted within a Host Profile data block, a list of Client Application data blocks are encapsulated by the Generic List data block. The Generic List data block has a block type of 31 in the series 1 group of blocks.

The following diagram shows the basic structure of a Generic List data block:



The following table describes the fields of the Generic List data block.

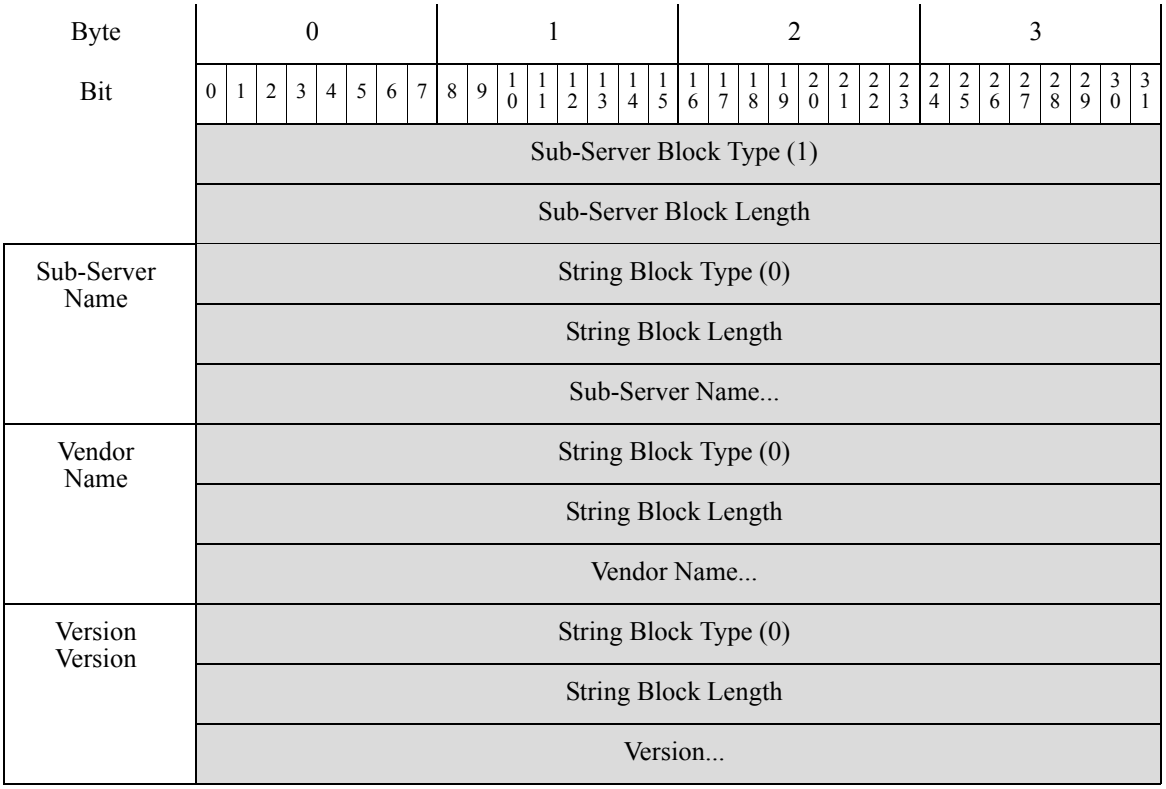
Table 4-31 Generic List Data Block Fields

Field	Number of Bytes	Description
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Sub-Server Data Block

The Sub-Server data block conveys information about an individual sub-server, which is a server called by another server on the same host and has associated vulnerabilities. The Sub-Server data block has a block type of 1 in the series 1 group of blocks.

The following diagram shows the format of the Sub-Server data block:



The following table describes the fields of the Sub-Server data block.

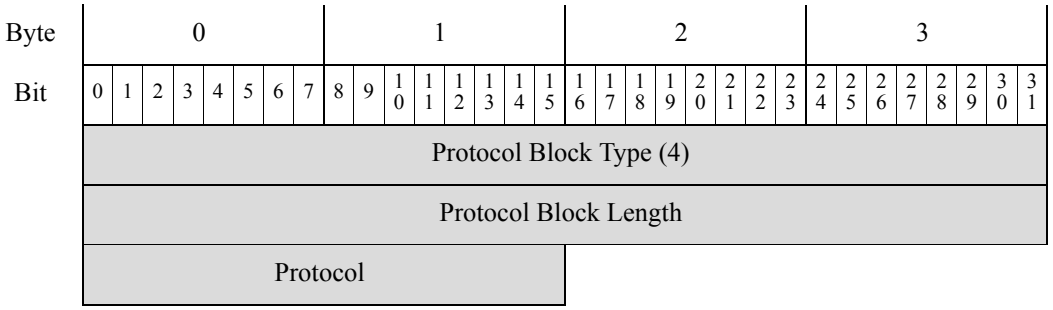
Table 4-32 Sub-Server Data Block Fields

Field	Data Type	Description
Sub-Server Block Type	uint32	Initiates a Sub-Server data block. This value is always `1`.
Sub-Server Block Length	uint32	Total number of bytes in the Sub-Server data block, including eight bytes for the Sub-Server block type and length fields, plus the number of bytes of data that follows.
String Block Type	uint32	Initiates a String data block containing the sub-server name. This value is always `0`.
String Block Length	uint32	Number of bytes in the sub-server name String data block, including the string block type and length fields, plus the number of bytes in the sub-server name.
Sub-Server Name	string	Name of the sub-server.
String Block Type	uint32	Initiates a String data block that contains the sub-server vendor. This value is always `0`.
String Block Length	uint32	Number of bytes in the vendor name String data block, including the string block type and length fields, plus the number of bytes in the vendor name.
Vendor Name	string	Sub-server vendor name.
String Block Type	uint32	Initiates a String data block that contains the sub-server version. This value is always `0`.
String Block Length	uint32	Number of bytes in the Sub-Server version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Sub-server version.

Protocol Data Block

The Protocol data block defines protocols. It is a very simple data block, with only the block type, block length, and the IANA protocol number identifying the protocol. The Protocol data block has a block type of 4 in the series 1 group of blocks.

The following graphic shows the format of the Protocol data block:



The following table describes the fields of the Protocol data block.

Table 4-33 Protocol Data Block Fields

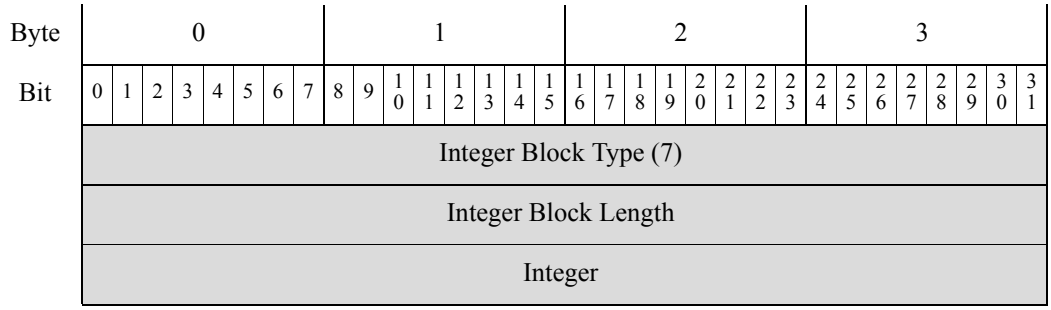
Field	Data Type	Description
Protocol Block Type	uint32	Initiates a Protocol data block. This value is always `4`.
Protocol Block Length	uint32	Number of bytes in the Protocol data block. This value is always `10`.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • `6` — TCP • `17` — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • `2048` — IP

Integer (INT32) Data Block

The Integer (INT32) data block is used in List data blocks to convey 32-bit integer data.

The Integer data block has a block type of 7 in the series 1 group of blocks.

The following diagram shows the format of the integer data block:



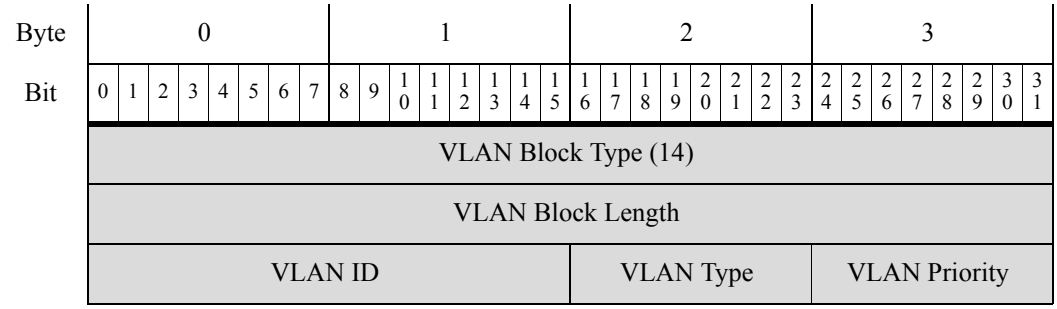
The following table describes the fields of the Integer data block:

Table 4-34 Integer Data Block Fields

Field	Data Type	Description
Integer Block Type	uint32	Initiates an Integer data block. The value is always `7`.
Integer Block Length	uint32	Number of bytes in the Integer data block. This value is always `12`.
Integer	uint32	Contains the integer value.

VLAN Data Block

The VLAN data block contains VLAN tag information for a host. The VLAN data block has a block type of 14 in the series 1 group of blocks.The following diagram shows the format of the VLAN data block:



The following table describes the fields of the VLAN data block.

Table 4-35 VLAN Data Block Fields

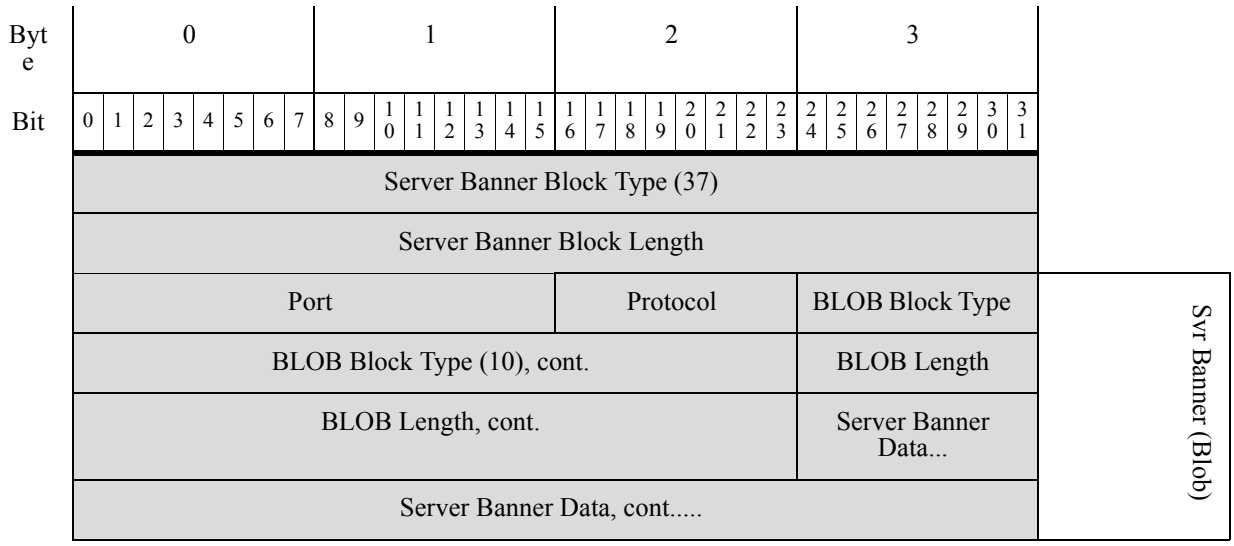
Field	Data Type	Description
VLAN Block Type	uint32	Initiates a VLAN data block. This value is always `14`.
VLAN Block Length	uint32	Number of bytes in the VLAN data block. This value is always `12`.
VLAN ID	uint16	Contains the VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag. • `0` — Ethernet • `1` — Token Ring
VLAN Priority	uint8	Priority value included in the VLAN tag.

Server Banner Data Block

The Server Banner data block provides information about the banner for a server running on a host. It contains the server port, protocol, and the banner data. The Server Banner data block has a block type of 37 in the series 1 group of blocks.

The following diagram shows the format of the Server Banner data block.

Note An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.



The following table describes the fields of the Server Banner data block.

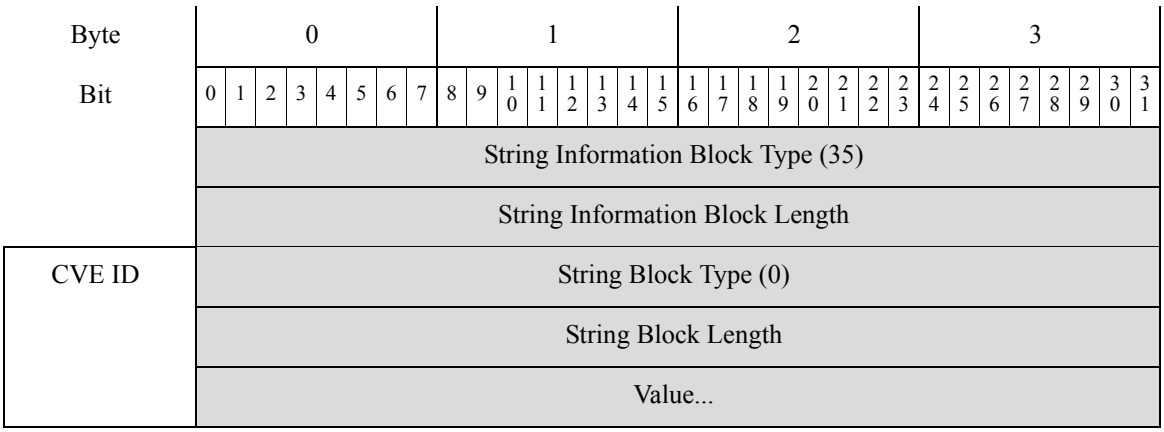
Table 4-36 Server Banner Data Block Fields

Field	Data Type	Description
Server Banner Block Type	uint32	Initiates a Server Banner data block. This value is always `37`.
Server Banner Block Length	uint32	Total number of bytes in the Server Banner data block, including the eight bytes in the server banner block type and length fields, plus the number of bytes of data that follows.
Port	uint16	Port number on which the server runs.
Protocol	uint8	Protocol number for the server.
BLOB Block Type	uint32	Initiates a BLOB data block containing server banner data. This value is always `10`.
Length	uint32	Total number of bytes in the BLOB data block (typically 264 bytes).
Banner	byte[*n*]	First *n* bytes of the packet involved in the server event, where *n* is equal to or less than 256.

String Information Data Block

The String Information data block contains string data. For example, the String Information data block is used to convey the Common Vulnerabilities and Exposures (CVE) identification string within a Scan Vulnerability data block. The String Information data block has a block type of 35 in the series 1 group of blocks.

The following diagram shows the format of the String Information data block:



The following table describes the fields of the String Information data block.

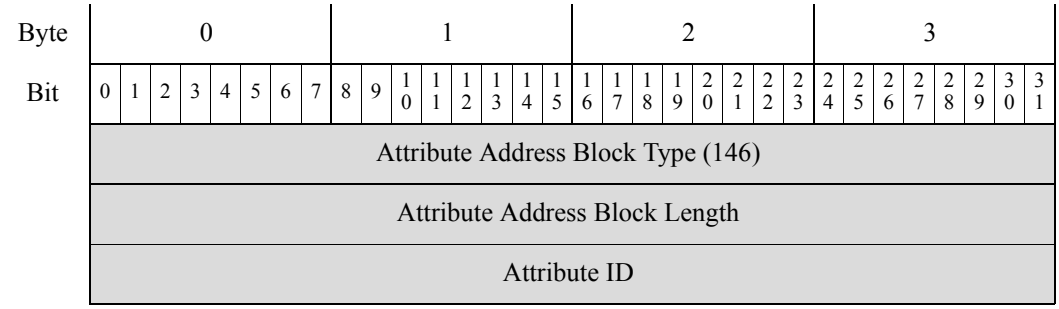
Table 4-37 String Information Data Block Fields

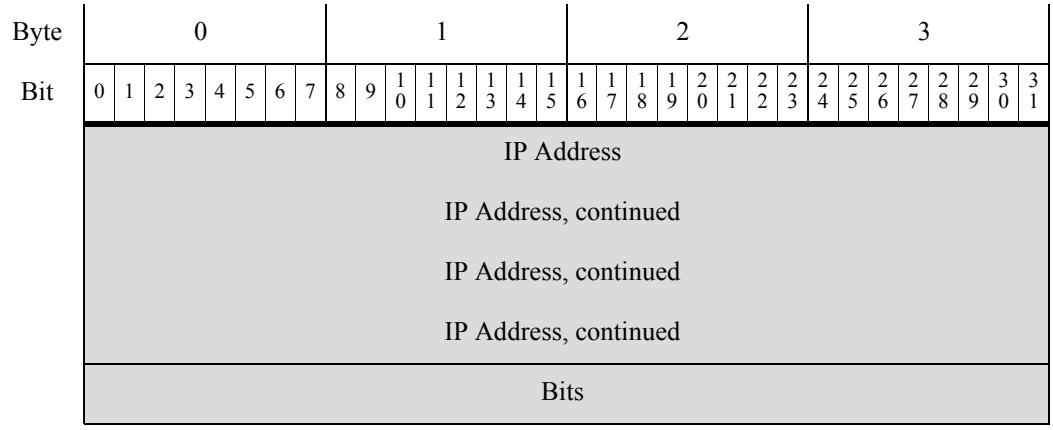
Field	Data Type	Description
String Information Block Type	uint32	Initiates a String Information data block. This value is always `35`.
String Information Block Length	uint32	Combined length of the String Information data block header and String Information data.
String Block Type	uint32	Initiates a string data block for the value.
String Block Length	uint32	Number of bytes in the string data block for the value, including eight bytes for the string block type and length, plus the number of bytes in the value.
Value	string	The value of the Common Vulnerabilities and Exposures (CVE) identification number for the vulnerability data block where the String Information data block is used.

Attribute Address Data Block 5.2+

The Attribute Address data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 146 in the series 1 group of blocks.

The following diagram shows the basic structure of an Attribute Address data block:





The following table describes the fields of the Attribute Address data block.

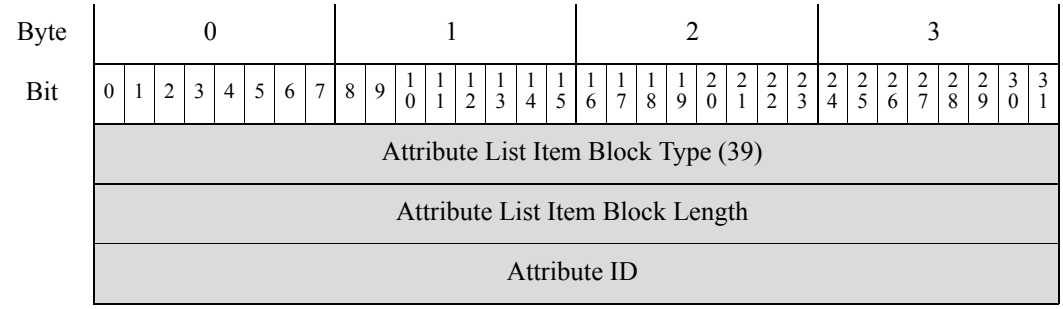
Table 4-38 Attribute Address Data Block 5.2+ Fields

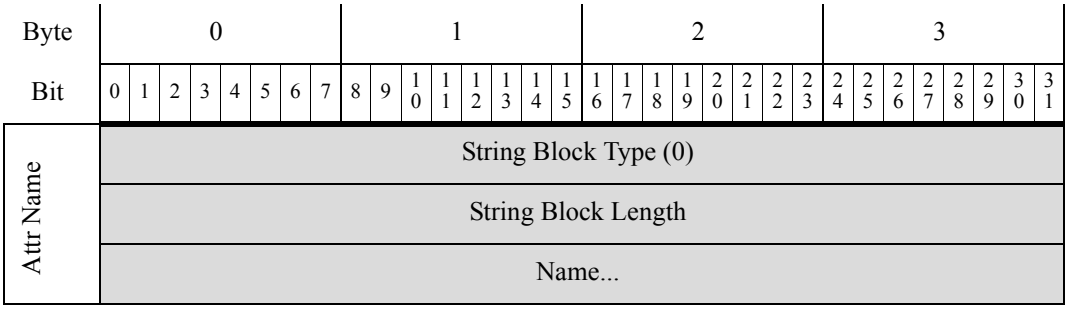
Field	Data Type	Description
Attribute Address Block Type	uint32	Initiates an Attribute Address data block. This value is always `146`.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the attribute address block type and length, plus the number of bytes in the attribute address data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
IP Address	uint8[16]	IP address of the host, if the address was automatically assigned. The address can be IPv4 or IPv6.
Bits	uint32	Contains the significant bits used to calculate the netmask if an IP address was automatically assigned.

Attribute List Item Data Block

The Attribute List Item data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 39 in the series 1 group of blocks.

The following diagram shows the basic structure of an Attribute List Item data block:





The following table describes the fields of the Attribute List Item data block.

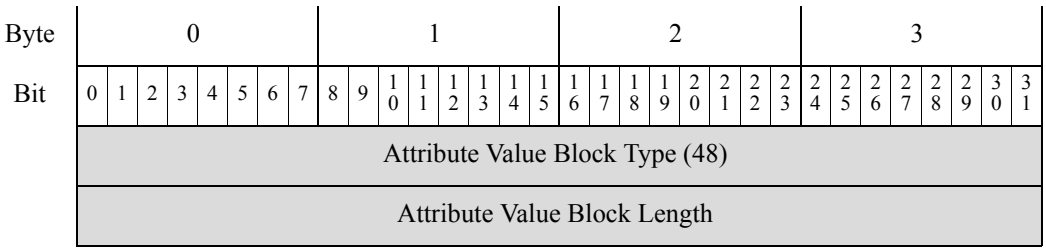
Table 4-39 Attribute List Item Data Block Fields

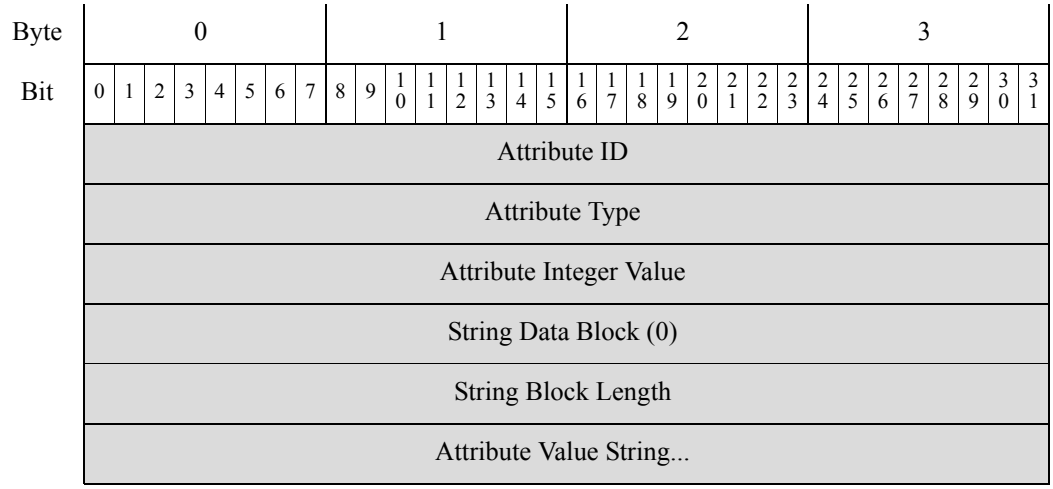
Field	Data Type	Description
Attribute List Item Block Type	uint32	Initiates an Attribute List Item data block. This value is always `39`.
Attribute List Item Block Length	uint32	Number of bytes in the Attribute List Item data block, including eight bytes for the attribute list item block type and length, plus the number of bytes in the attribute list item data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
String Block Type	uint32	Initiates a String data block for the attribute list item name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the attribute list item name, including eight bytes for the string block type and length, plus the number of bytes in the attribute list item name.
Name	string	Attribute list item name.

Attribute Value Data Block

The Attribute Value data block conveys attribute identification numbers and values for host attributes. An Attribute Value data block for each attribute applied to the host in the event is included in a list in the Full Host Profile data block. The Attribute Value data block has a block type of 48 in the series 1 group of blocks.

The following diagram shows the format of the Attribute Value data block:





The following table describes the components of the Attribute Value data block.

Table 4-40 Attribute Value Data Block Fields

Field	Data Type	Description
Attribute Value Block Type	uint32	Initiates an Attribute Value data block. This value is always `48`.
Attribute Value Block Length	uint32	Total number of bytes in the Attribute Value data block, including eight bytes for the attribute value block type and length fields, plus the number of bytes of attribute block data that follows.
Attribute ID	uint32	The identification number for the attribute.
Attribute Type	uint32	Type of affected attribute. Possible values are: • `0` — Attribute with text as value; this uses string data • `1` — Attribute with value in range; this uses integer data • `2` — Attribute with a list of possible values, this uses integer data • `3` — Attribute with a URL as value; this uses string data • `4` — Attribute with binary BLOB as value; this uses string data
Attribute Integer Value	uint32	Integer value for the attribute, if applicable.
String Block Type	uint32	Initiates a String data block containing the attribute name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including the string block type and length fields, plus the number of bytes in the attribute name.
Attribute Value	string	Value of the attribute.

Full Sub-Server Data Block

The Full Sub-Server data block conveys information about a sub-server associated with a server detected on a host, and includes information about the sub-server such as its vendor and version and any related VDB and third-party vulnerabilities for the sub-server on the host. A sub-server is a loadable module of

a server that has its own associated vulnerabilities. A Full Host Server data block includes a Full Sub-Server data block for each sub-server detected on the host. The Full Sub-Server data block has a block type of 51 in the series 1 group of blocks.

Note An asterisk (*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Sub-Server data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Full Sub-Server Block Type (51)																																
Full Sub-Server Block Length																																
String Block Type (0)																																
String Block Length																																
Sub-Server Name String...																																
String Block Type (0)																																
String Block Length																																
Sub-Server Vendor Name String...																																
String Block Type (0)																																
String Block Length																																
Sub-Server Version String...																																
Generic List Block Type (31)																																
Generic List Block Length																																
(VDB) Host Vulnerability Data Blocks*																																
Generic List Block Type (31)																																
Generic List Block Length																																
(Third-Party Scan) Host Vulnerability Data Blocks*																																

The following table describes the components of the Full Sub-Server data block.

Table 4-41 Full Sub-Server Data Block Fields

Field	Data Type	Description
Full Sub-Server Block Type	uint32	Initiates a Full Sub-Server data block. This value is always 51.
Full Sub-Server Block Length	uint32	Total number of bytes in the Full Sub-Server data block, including eight bytes for the Full Sub-Server block type and length fields, plus the number of bytes in the full sub-server data that follows.
String Block Type	uint32	Initiates a String data block containing the sub-server name. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server name.
Sub-Server Name	string	Sub-server name.
String Block Type	uint32	Initiates a String data block containing the sub-server vendor's name. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server vendor name.
Sub-Server Vendor Name	string	Name of the sub-server vendor.
String Block Type	uint32	Initiates a String data block that contains the sub-server version. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server version.
Sub-Server Version	string	Sub-server version.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB Vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
VDB Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by Cisco. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Third-Party Scan Vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third-Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by a third-party vulnerability scanner. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.

Operating System Data Block 3.5+

The operating system data block for Version 3.5+ has a block type of 53 in the series 1 group of blocks. The block includes a fingerprint Universally Unique Identifier (UUID). The following diagram shows the format of an operating system data block in 3.5+.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Operating System Block Type (53)																															
	Operating System Block Length																															
	Confidence																															
OS Fingerprint UUID	Fingerprint UUID																															
	Fingerprint UUID, continued																															
	Fingerprint UUID, continued																															
	Fingerprint UUID, continued																															

The following table describes the fields of the v3.5 operating system data block.

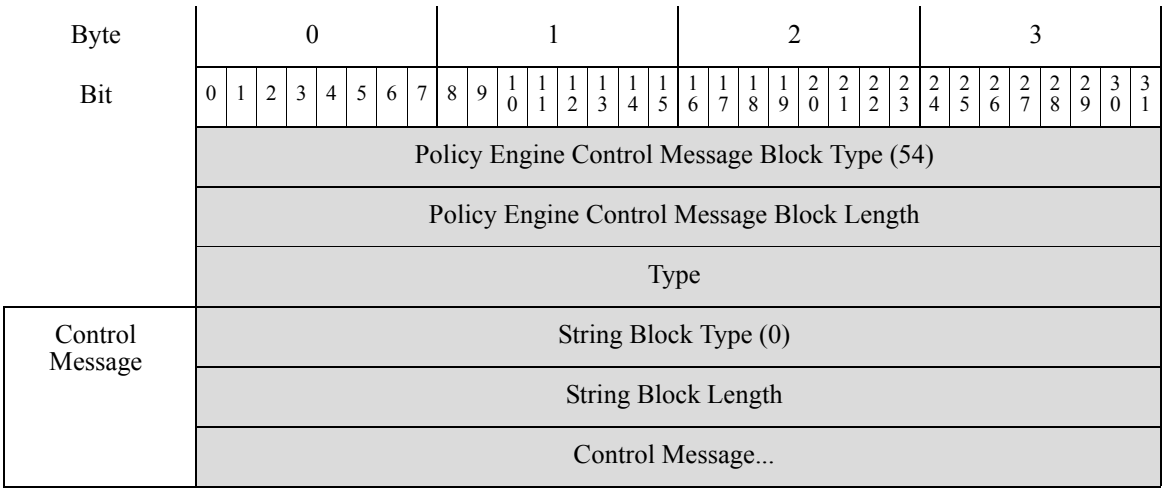
Table 4-42 Operating System Data Block 3.5+ Fields

Field	Data Type	Description
Operating System Data Block Type	uint32	Initiates the operating system data block. This value is always `53`.
Operating System Data Block Length	uint32	Number of bytes in the Operating System data block. This value should always be `28`: eight bytes for the data block type and length fields, plus four bytes for the confidence value and sixteen bytes for the fingerprint UUID value.
Confidence	uint32	Confidence percentage value.
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the Cisco database.

Policy Engine Control Message Data Block

The Policy Engine Control Message data block conveys the control message content for policy types. The Policy Engine Control Message data block has a block type of 54 in the series 1 group of blocks.

The following diagram shows the format of the Policy Engine Control Message data block:



The following table describes the components of the Policy Engine Control Message data block.

Table 4-43 Policy Engine Control Message Data Block Fields

Field	Data Type	Description
Policy Engine Control Message Block Type	uint32	Initiates a Policy Engine Control Message data block. This value is always `54`.
Policy Engine Control Message Length	uint32	Total number of bytes in the Policy Engine Control Message data block, including eight bytes for the policy engine control block type and length fields, plus the number of bytes of policy engine control data that follows.
Type	uint32	Indicates the type of policy for the event.
String Block Type	uint32	Initiates a String data block that contains the control message. This value is always `0`.
String Block Length	uint32	Number of bytes in the control message String data block, including eight bytes for the block type and length fields, plus the number of bytes in the control message.
Control Message	uint32	The control message from the policy engine.

Attribute Definition Data Block for 4.7+

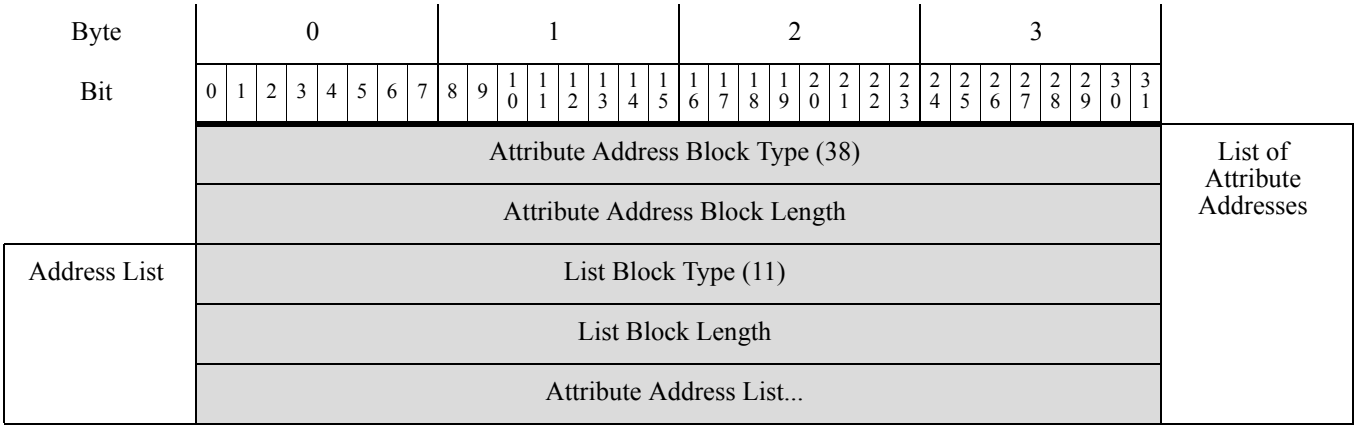
The Attribute Definition data block contains the attribute definition in an attribute creation, change, or deletion event and is used within Host Attribute Add events (event type 1002, subtype 6), Host Attribute Update events (event type 1002, subtype 7), and Host Attribute Delete events (event type 1002, subtype 8). It has a block type of 55 in the series 1 group of blocks.

For more information on those events, see Attribute Messages, page 4-50.

The following diagram shows the basic structure of an Attribute Definition data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Attribute Definition Block Type (55)																															
	Attribute Definition Block Length																															
	Source ID																															
	UUID UUID, continued UUID, continued UUID, continued																															
	ID																															
Name	String Block Type (0)																															
	String Block Length																															
	Name...																															
	Attribute Type																															
	Attribute Category																															
	Starting Value for Integer Range																															
	Ending Value for Integer Range																															
	Auto-Assigned IP Address Flag																															
	Attribute List Item Block Type (39)																															
	Attribute List Item Block Length																															
List Item	List Block Type (11)																															
	List Block Length																															
	Attribute List Items...																															

List of Attribute List Items



The following table describes the fields of the Attribute Definition data block.

Table 4-44 Attribute Definition Data Block Fields

Field	Data Type	Description
Attribute Definition Block Type	uint32	Initiates an Attribute Definition data block. This value is always `55`.
Attribute Definition Block Length	uint32	Number of bytes in the Attribute Definition data block, including eight bytes for the attribute definition block type and length, plus the number of bytes in the attribute definition data that follows.
Source ID	uint32	Identification number that maps to the source of the attribute data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
UUID	uint8[16]	An ID number that acts as a unique identifier for the affected attribute.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
String Block Type	uint32	Initiates a String data block for the attribute definition name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the attribute definition name, including eight bytes for the string block type and length, plus the number of bytes in the attribute definition name.
Name	string	Attribute definition name.
Attribute Type	uint32	Type of attribute. Possible values are: • `0` — Attribute with text as value; this uses string data • `1` — Attribute with value in range; this uses integer data • `2` — Attribute with a list of possible values; this uses integer data • `3` — Attribute with a URL as value; this uses string data • `4` — Attribute with binary BLOB as value; this uses string data
Attribute Category	uint32	Attribute category.
Starting Value for Range	uint32	First integer in the integer range for the defined attribute.

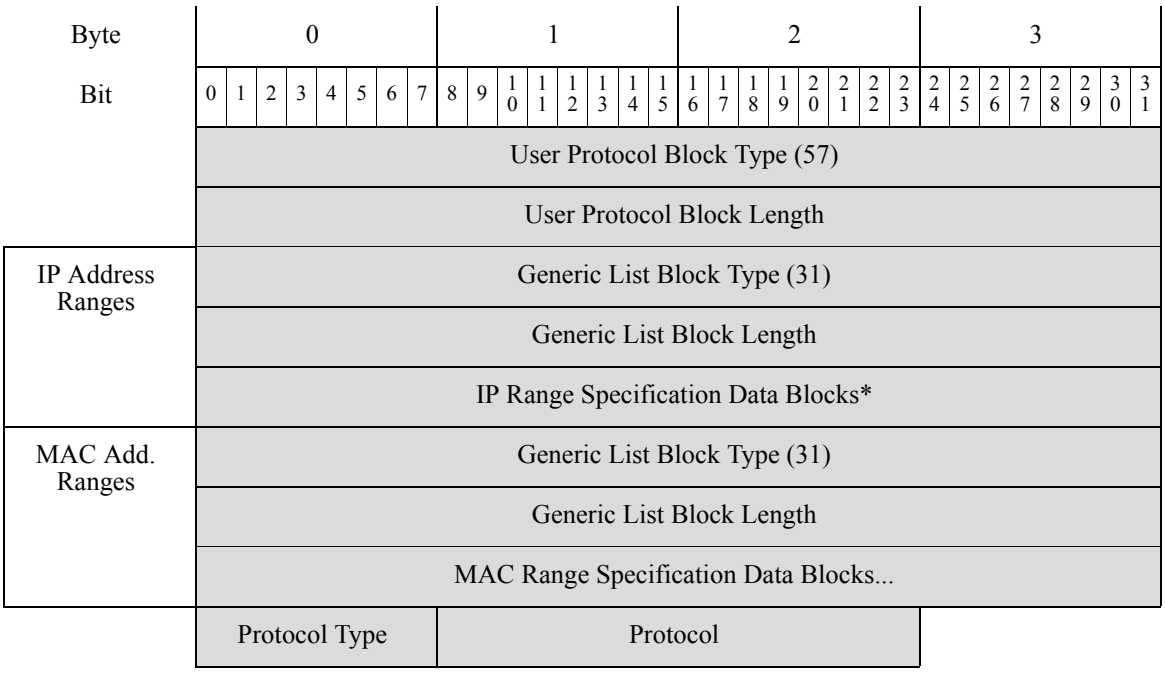
Table 4-44 Attribute Definition Data Block Fields (continued)

Field	Data Type	Description
Ending Value for Range	uint32	Last integer in the integer range for the defined attribute.
Auto-Assigned IP Address Flag	uint32	Flag indicating if an IP address is auto-assigned based on the attribute.
List Block Type	uint32	Initiates a List data block comprising Attribute List Item data blocks conveying attribute list items. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Attribute List Item data blocks. This field is followed by zero or more Attribute List Item data blocks.
Attribute List Item Block Type	uint32	Initiates the first Attribute List Item data block. This data block can be followed by other Attribute List Item data blocks up to the limit defined in the list block length field.
Attribute List Item Block Length	uint32	Number of bytes in the Attribute List Item String data block, including eight bytes for the block type and header fields, plus the number of bytes in the attribute list item.
Attribute List Item	variable	Attribute List Item data as documented in Attribute List Item Data Block, page 4-73.
List Block Type	uint32	Initiates a List data block comprising Attribute Address data blocks conveying IP addresses for hosts with the attribute. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Attribute Address data blocks. This field is followed by zero or more Attribute Address data blocks.
Attribute Address Block Type	uint32	Initiates the first Attribute Address data block. This data block can be followed by other Attribute Address data blocks up to the limit defined in the list block length field.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the block type and header fields, plus the number of bytes in the attribute address.
Attribute Address	variable	Attribute Address data as documented in Attribute Address Data Block 5.2+, page 4-72.

User Protocol Data Block

The User Protocol data block is used to contain information about added protocols, the type of the protocol, and lists of IP address and MAC address ranges for the hosts with the protocol. The User Protocol data block has a block type of 57 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Protocol data block:



The following table describes the fields of the User Protocol data block.

Table 4-45 User Protocol Data Block Fields

Field	Number of Bytes	Description
User Protocol Block Type	uint32	Initiates a User Protocol data block. This value is always `57`.
User Protocol Block Length	uint32	Total number of bytes in the User Protocol data block, including eight bytes for the user protocol block type and length fields, plus the number of bytes of user protocol data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising MAC Range Specification data blocks conveying MAC address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated MAC Range Specification data blocks.
MAC Range Specification Data Blocks *	variable	MAC Range Specification data blocks containing information about the MAC address ranges for the user input. See MAC Address Specification Data Block, page 4-90 for a description of this data block.

Table 4-45 User Protocol Data Block Fields (continued)

Field	Number of Bytes	Description
Protocol Type	uint8	Indicates the type of the protocol. The protocol can be either `0`, for a network layer protocol such as IP, or `1` for a transport layer protocol such as TCP or UDP.
Protocol	uint16	Indicates the protocol for the data contained in the data block.

User Client Application Data Block for 5.1.1+

The User Client Application data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of IP address range data blocks. The payload ID, which was added in Version 6.0, specifies the application instance associated with the record. The User Client Application data block has a block type of 138 in the series 1 group of blocks. It replaces block type .

The following diagram shows the basic structure of a User Client Application data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Client Application Block Type (138)																															
	User Client Application Block Length																															
IP Range Specification	Generic List Block Type (31)																															
	Generic List Block Length																															
	IP Range Specification Data Blocks*																															
	Application Protocol ID																															
	Client Application ID																															
Version	String Block Type (0)																															
	String Block Length																															
	Version...																															
	Payload Type																															
	Web Application ID																															

The following table describes the fields of the User Client Application data block.

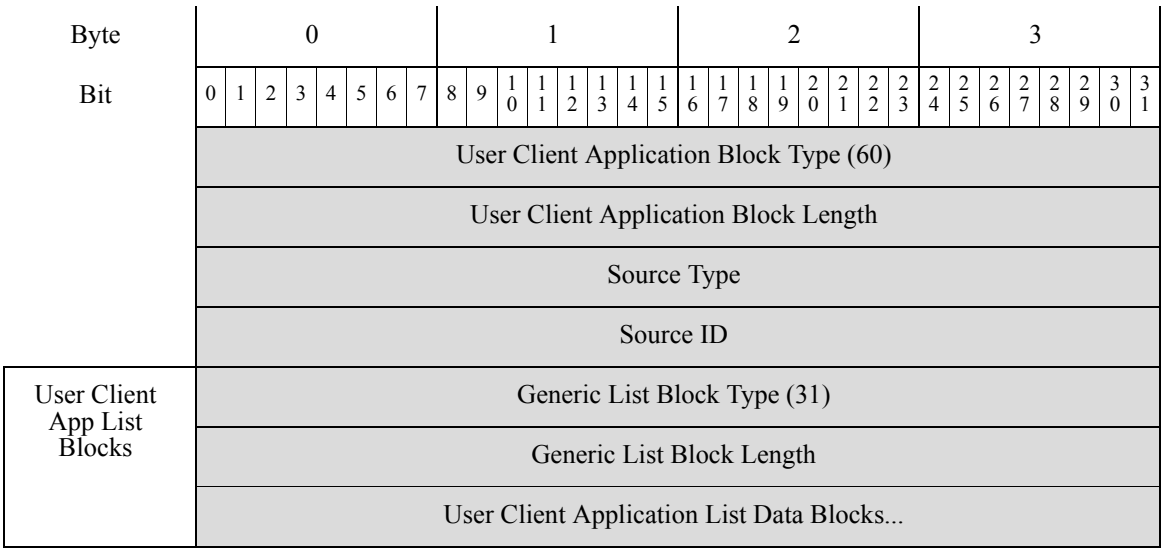
Table 4-46 User Client Application Data Block Fields

Field	Number of Bytes	Description
User Client Application Block Type	uint32	Initiates a User Client Application data block. This value is always `138`.
User Client Application Block Length	uint32	Total number of bytes in the User Client Application data block, including eight bytes for the user client application block type and length fields, plus the number of bytes of user client application data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block that contains the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Client application version.
Payload Type	uint32	This field is included for backwards compatibility. It is always `0`.
Web Application ID	uint32	The internal identification number for the web application, if applicable.

User Client Application List Data Block

The User Client Application List data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of client application blocks. The User Client Application List data block has a block type of 60 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Client Application List data block:



The following table describes the fields of the User Client Application List data block.

Table 4-47 User Client Application List Data Block Fields

Field	Number of Bytes	Description
User Client Application List Block Type	uint32	Initiates a User Client Application List data block. This value is always `60`.
User Client Application List Block Length	uint32	Total number of bytes in the User Client Application List data block, including eight bytes for the user client application list block type and length fields, plus the number of bytes of user client application list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the client data was detected by RNA • `1` if the client data was provided by a user • `2` if the client data was detected by a third-party scanner • `3` if the client data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Source ID	uint32	Identification number that maps to the source that added the affected client application. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.

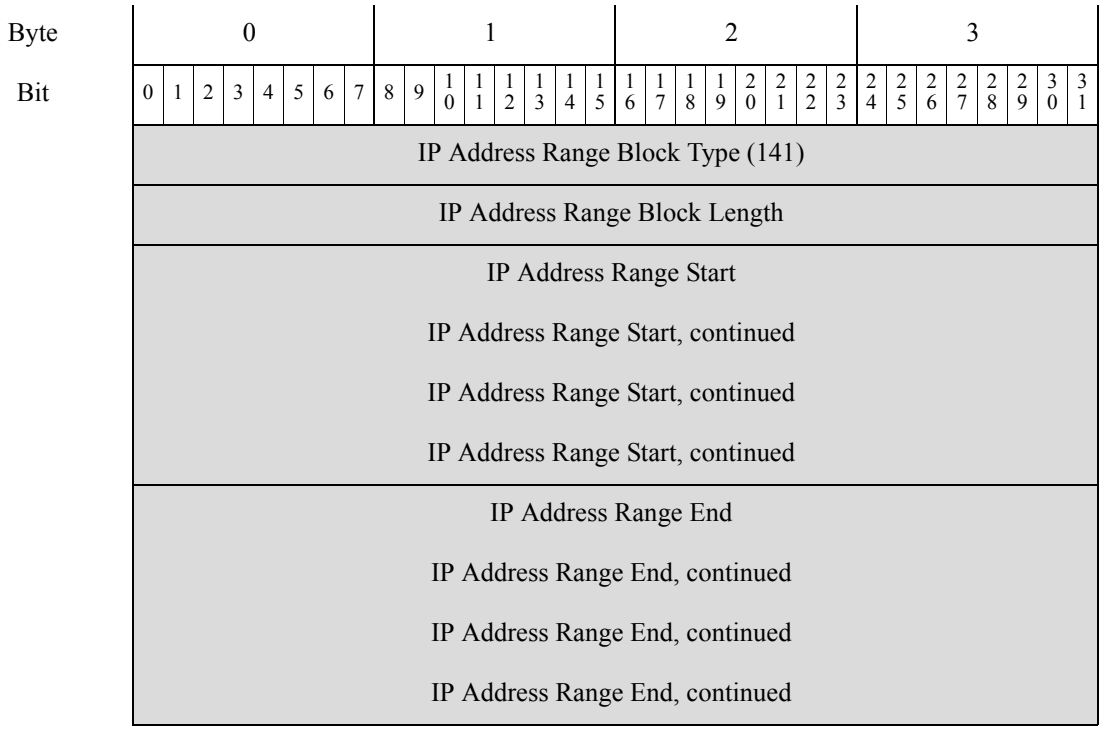
Table 4-47 User Client Application List Data Block Fields (continued)

Field	Number of Bytes	Description
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Client Application Blocks	variable	Encapsulated User Client Application data blocks up to the maximum number of bytes in the list block length. For more information on the User Client Application data block, see User Client Application Data Block for 5.1.1+, page 4-84.

IP Address Range Data Block for 5.2+

The IP Address Range data block for 5.2+ conveys a range of IP addresses. IP Address Range data blocks are used in User Protocol, User Client Application, Address Specification, User Product, User Server, User Hosts, User Vulnerability, User Criticality, and User Attribute Value data blocks. The IP Address Range data block has a block type of 141 in the series 1 group of blocks.

The following diagram shows the format of the IP Address Range data block:



The following table describes the components of the IP Address Range Specification data block.

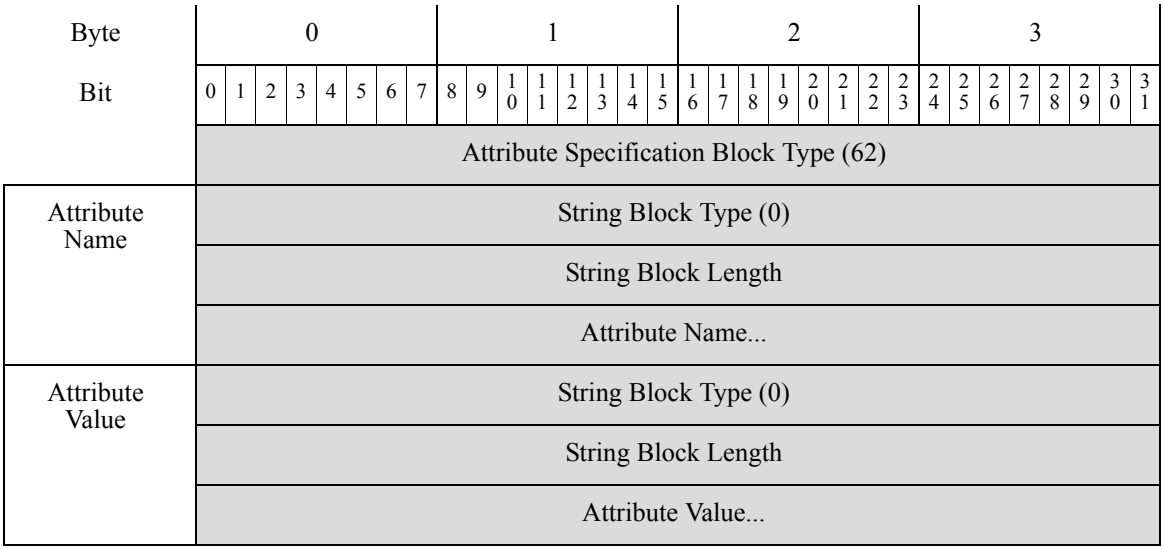
Table 4-48 IP Address Range Data Block Fields

Field	Data Type	Description
IP Address Range Block Type	uint32	Initiates a IP Address Range data block. This value is always `61`.
IP Address Range Block Length	uint32	Total number of bytes in the IP Address Range data block, including eight bytes for the IP Address Range block type and length fields, plus the number of bytes of IP Address Range data that follows.
IP Address Range Start	uint8[16]	The starting IP address for the IP address range.
IP Address Range End	uint8[16]	The ending IP address for the IP address range.

Attribute Specification Data Block

The Attribute Specification data block conveys the attribute name and value. The Attribute Specification data block has a block type of 62 in the series 1 group of blocks.

The following diagram shows the format of the Attribute Specification data block:



The following table describes the components of the Attribute Specification data block.

Table 4-49 Attribute Specification Data Block Fields

Field	Data Type	Description
Attribute Specification Block Type	uint32	Initiates an Attribute Specification data block. This value is always 62.
String Block Type	uint32	Initiates a String data block that contains the attribute name. This value is always `0`.

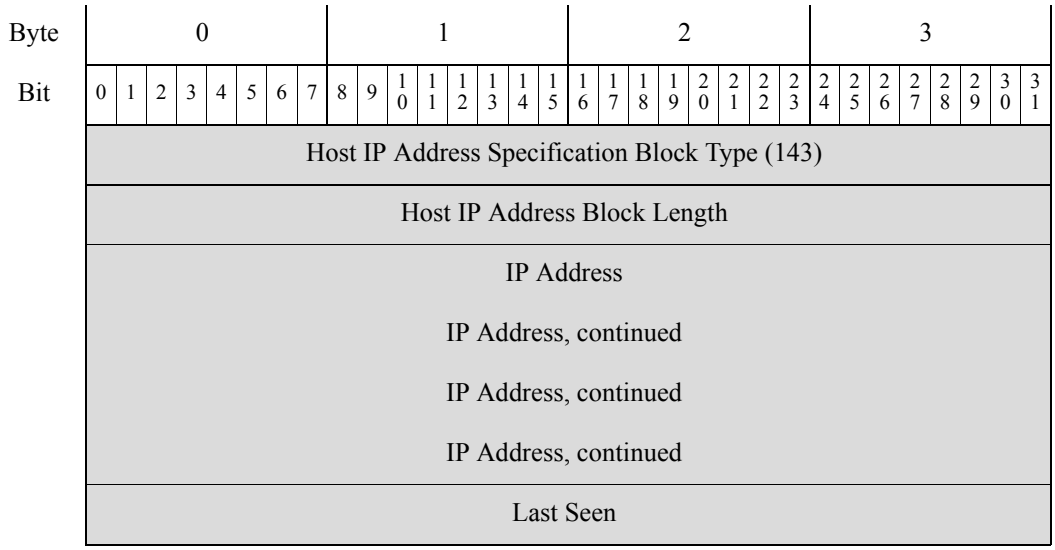
Table 4-49 Attribute Specification Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the attribute name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the attribute name.
Attribute Value	uint32	The value of the attribute.
String Block Type	uint32	Initiates a String data block that contains the attribute name. This value is always `0`.
String Block Length	uint32	Number of bytes in the attribute name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the attribute name.
Attribute Name	uint32	The name of the attribute.

Host IP Address Data Block

The Host IP Address data block conveys an individual IP address. The IP address may be either an IPv4 or IPv6 address. Host IP Address data blocks are used in User Protocol, Address Specification, and User Host data blocks. The Host IP data block has a block type of 143 in the series 1 group of blocks.

The following diagram shows the format of the Host IP Address data block:



The following table describes the components of the Host IP Address data block.

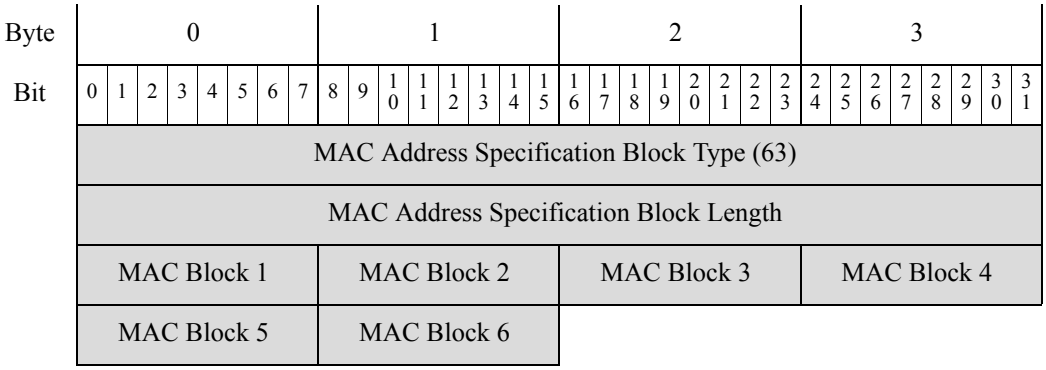
Table 4-50 Host IP Address Data Block Fields

Field	Data Type	Description
Host IP Address Block Type	uint32	Initiates a Host IP Address data block. This value is always `143`.
Host IP Block Length	uint32	Total number of bytes in the Host IP Address data block, including eight bytes for the Host IP block type and length fields, plus the number of bytes of Host IP Address data that follows.
IP Address	uint8[16]	The IP address. This can be IPv4 or IPv6.
Last Seen	uint32	UNIX timestamp that represents the last time the IP address was detected.

MAC Address Specification Data Block

The MAC Address Specification data block conveys an individual MAC address. MAC Address Specification data blocks are used in User Protocol, Address Specification, and User Hosts data blocks. The MAC Address Specification data block has a block type of 63 in the series 1 group of blocks.

The following diagram shows the format of the MAC Address Specification data block:



The following table describes the components of the MAC Address Specification data block.

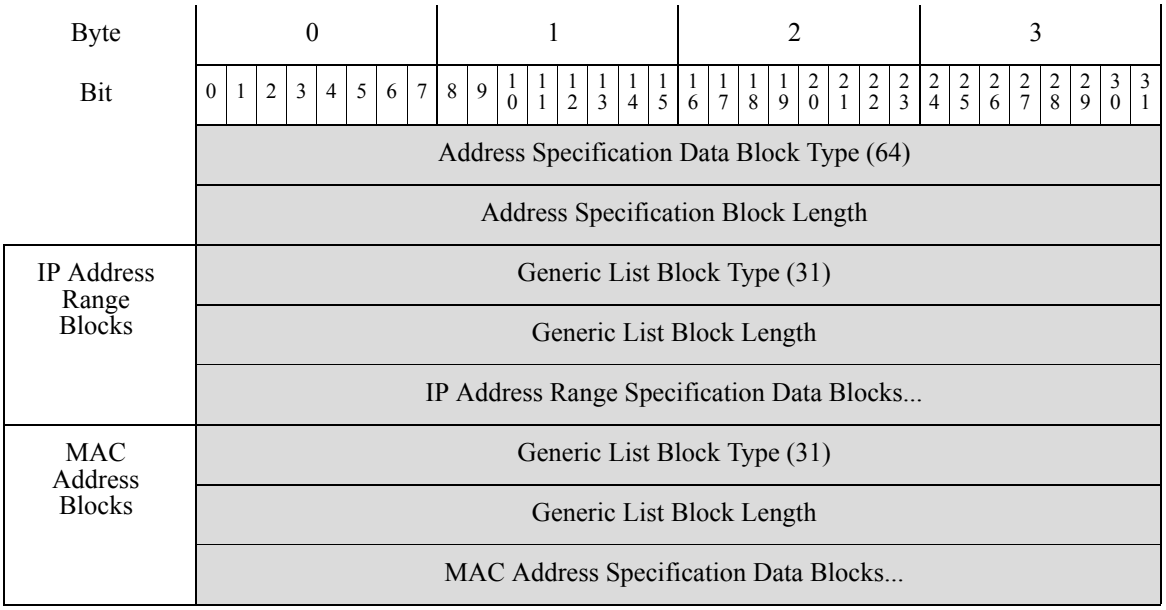
Table 4-51 MAC Address Specification Data Block Fields

Field	Data Type	Description
MAC Address Specification Block Type	uint32	Initiates a MAC Address Specification data block. This value is always `63`.
MAC Address Specification Block Length	uint32	Total number of bytes in the MAC Address Specification data block, including eight bytes for the MAC Address Specification block type and length fields, plus the number of bytes of MAC address specification data that follows.
MAC Address Blocks 1 - 6	uint8	The blocks of the MAC address in sequential order.

Address Specification Data Block

The Address Specification data block is used to contain lists of IP address range specifications and MAC address specifications. The Address Specification data block has a block type of 64 in the series 1 group of blocks.

The following diagram shows the basic structure of an Address Specification data block:



The following table describes the fields of the Address Specification data block.

Table 4-52 Address Specification Data Block Fields

Field	Number of Bytes	Description
Address Specification Data Block Type	uint32	Initiates an Address Specification data block. This value is always `64`.
Address Specification Block Length	uint32	Total number of bytes in the Address Specification data block, including eight bytes for the address specification block type and length fields, plus the number of bytes of address specification data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.

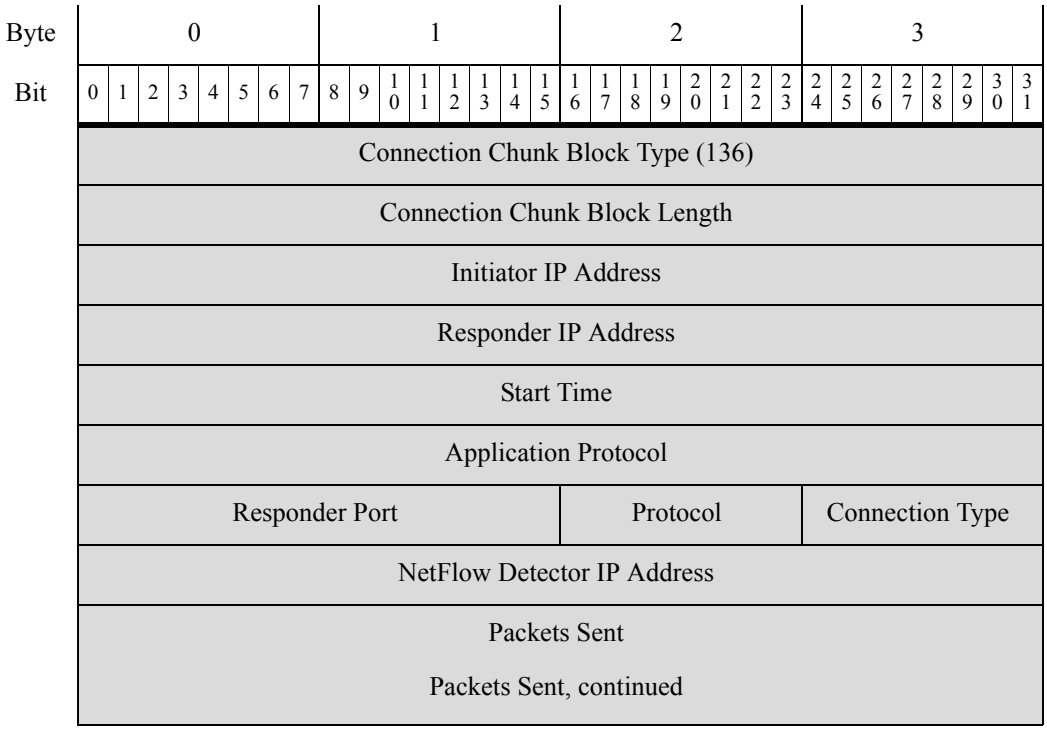
Table 4-52 Address Specification Data Block Fields (continued)

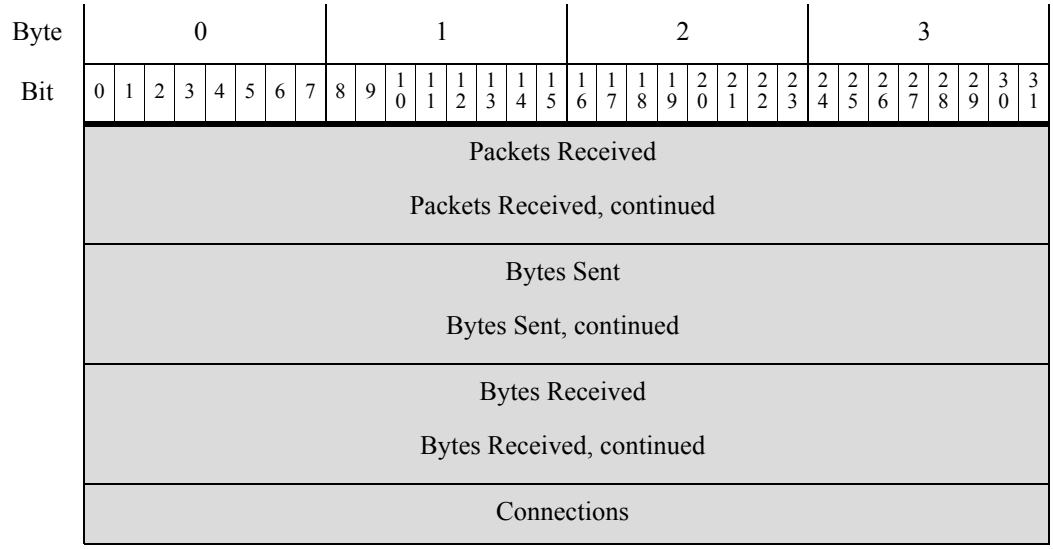
Field	Number of Bytes	Description
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length. For more information, see IP Address Range Data Block for 5.2+, page 4-87.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
MAC Address Specification Data Blocks	variable	Encapsulated MAC Address Specification data blocks up to the maximum number of bytes in the list block length. For more information, see MAC Address Specification Data Block, page 4-90.

Connection Chunk Data Block for 5.1.1+

The Connection Chunk data block conveys connection data. It stores connection log data that aggregates over a five-minute period. The Connection Chunk data block has a block type of 136 in the series 1 group of blocks. It supersedes block type 119.

The following diagram shows the format of the Connection Chunk data block:





The following table describes the components of the Connection Chunk data block.

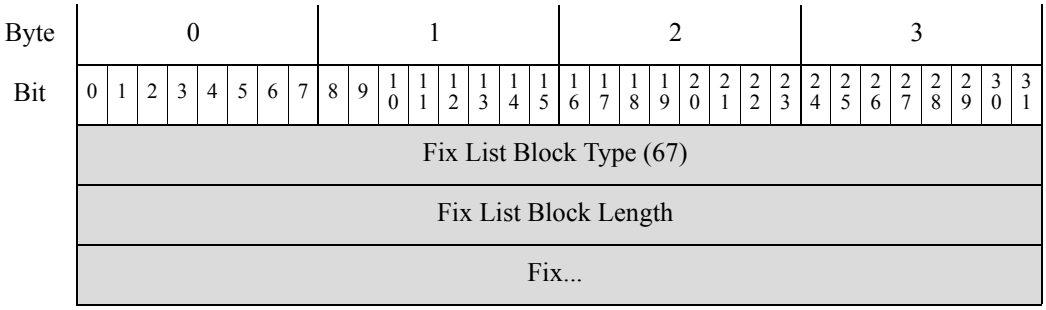
Table 4-53 Connection Chunk Data Block Fields

Field	Data Type	Description
Connection Chunk Block Type	uint32	Initiates a Connection Chunk data block. This value is always `136`.
Connection Chunk Block Length	uint32	Total number of bytes in the Connection Chunk data block, including eight bytes for the connection chunk block type and length fields, plus the number of bytes in the connection chunk data that follows.
Initiator IP Address	uint8(4)	IP address of the initiator of this type of connection. This is used with the responder IP address to identify identical connections.
Responder IP Address	uint8(4)	IP address of the responder to this type of connection. This is used with the initiator IP address to identify identical connections.
Start Time	uint32	The starting time for the connection chunk.
Application Protocol	uint32	Identification number for the protocol used in the connection.
Responder Port	uint16	The port used by the responder in the connection chunk.
Protocol	uint8	The protocol for the packet containing the user information.
Connection Type	uint8	The type of connection.
NetFlow Detector IP Address	uint8[4]	IP address of the NetFlow device that detected the connection, in IP address octets.
Packets Sent	uint64	The number of packets sent in the connection chunk.
Packets Received	uint64	The number of packets received in the connection chunk.
Bytes Sent	uint64	The number of bytes sent in the connection chunk.
Bytes Received	uint64	The number of bytes received in the connection chunk.
Connections	uint32	The number of connections over a five-minute period.

Fix List Data Block

The Fix List data block conveys a fix that applies to a host. A Fix List data block for each fix applied to the affected host is included in a User Product data block. The Fix List data block has a block type of 67 in the series 1 group of blocks.

The following diagram shows the format of the Fix List data block:



The following table describes the components of the Fix List data block.

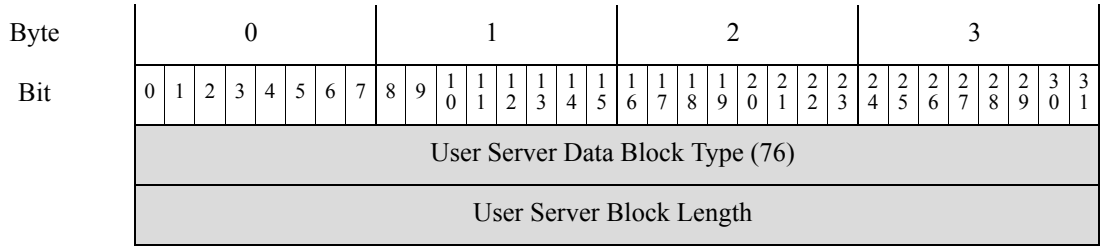
Table 4-54 Fix List Data Block Fields

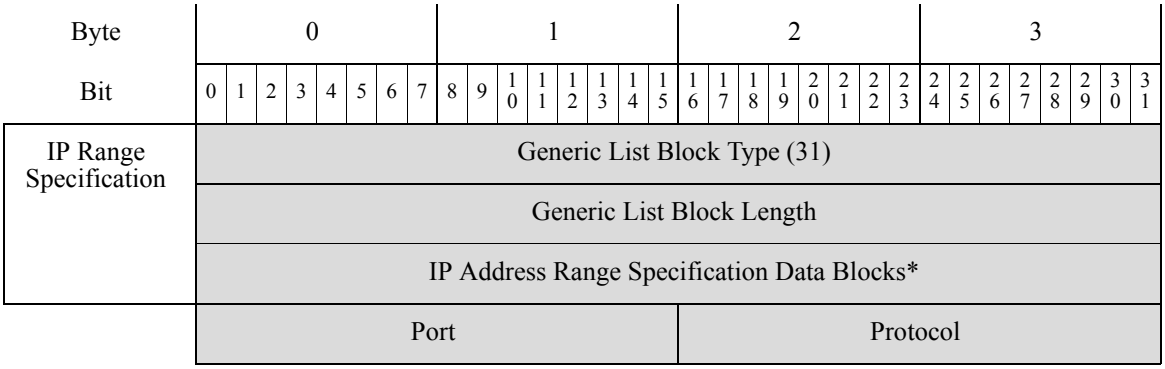
Field	Data Type	Description
Fix List Block Type	uint32	Initiates a Fix List data block. This value is always `67`.
Fix List Block Length	uint32	Total number of bytes in the Fix List data block, including eight bytes for the Fix List block type and length fields, plus the number of bytes of fix identification data that follows.
Fix ID	uint32	The identification number for the fix.

User Server Data Block

The User Server data block contains server details from a user input event. The User Server data block has a block type of 76 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Server data block:





The following table describes the fields of the User Server data block.

Table 4-55 User Server Data Block Fields

Field	Number of Bytes	Description
User Server Data Block Type	uint32	Initiates a User Server data block. This value is always `76`.
User Server Block Length	uint32	Total number of bytes in the User Server data block, including eight bytes for the user server block type and length fields, plus the number of bytes of user server data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length.
Port	uint16	Port used by the server.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • `6` — TCP • `17` — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • `2048` — IP

User Server List Data Block

The User Server List data block contains a list of server data blocks from a user input event. The User Server List data block has a block type of 77 in the series 1 group of blocks. The following diagram shows the basic structure of a User Server List data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Server List Data Block Type (77)																															
	User Server List Block Length																															
	Source Type																															
	Source ID																															
User Server Blocks	Generic List Block Type (31)																															
	Generic List Block Length																															
	User Server Data Block*																															

The following table describes the fields of the User Server List data block.

Table 4-56 User Server List Data Block Fields

Field	Number of Bytes	Description
User Server List Data Block Type	uint32	Initiates a User Server List data block. This value is always `77`.
User Server List Block Length	uint32	Total number of bytes in the User Server List data block, including eight bytes for the user server list block type and length fields, plus the number of bytes of user server list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was detected by RNA • `1` if the server data was provided by a user • `2` if the server data was detected by a third-party scanner • `3` if the server data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.

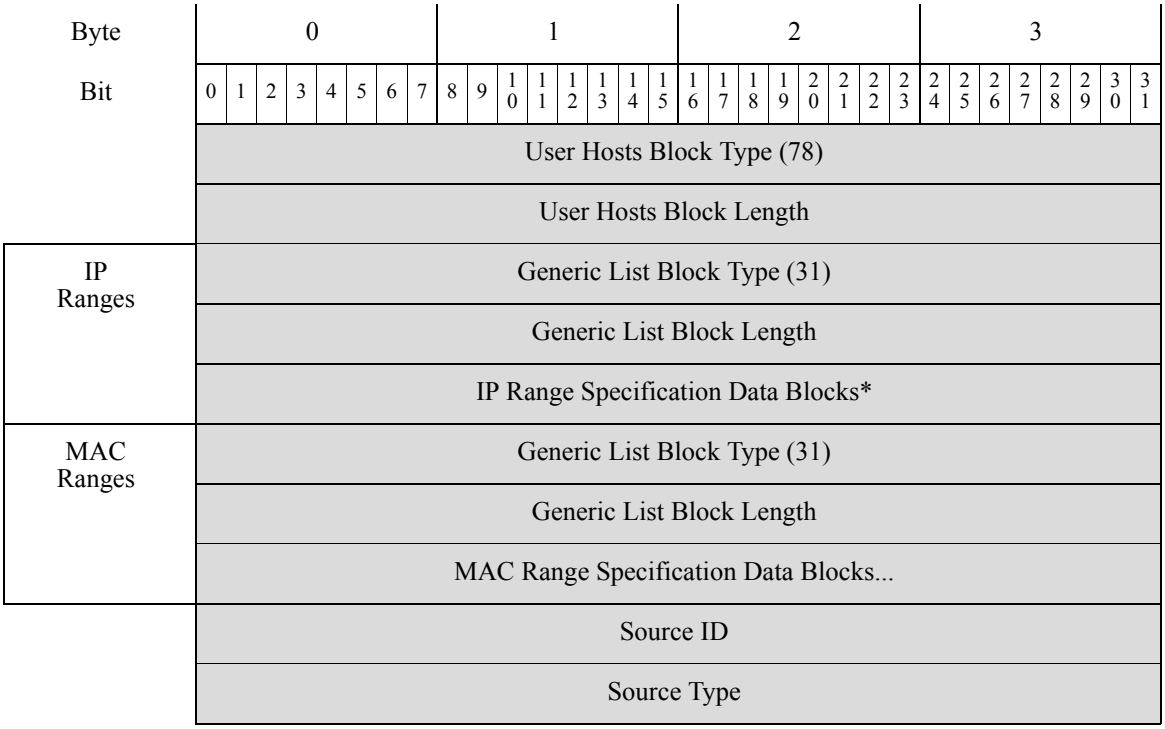
Table 4-56 User Server List Data Block Fields (continued)

Field	Number of Bytes	Description
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Server Data Blocks	variable	Encapsulated User Server data blocks up to the maximum number of bytes in the list block length.

User Hosts Data Block 4.7+

The User Hosts data block is used in User Add and Delete Host Messages, page 4-49 to contain information about host ranges and user and source identity from a user host input event. The User Hosts data block has a block type of 78 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Hosts data block:



The following table describes the fields of the User Hosts data block:

Table 4-57 User Hosts Data Block Fields

Field	Number of Bytes	Description
User Hosts Block Type	uint32	Initiates a User Hosts data block. This value is always `78`.
User Hosts Block Length	uint32	Total number of bytes in the User Hosts data block, including eight bytes for the user hosts block type and length fields, plus the number of bytes of user hosts data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising MAC Range Specification data blocks conveying MAC address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated MAC Range Specification data blocks.
MAC Range Specification Data Blocks *	variable	MAC Range Specification data blocks containing information about the MAC address ranges for the user input. See MAC Address Specification Data Block, page 4-90 for a description of this data block.
Source ID	uint32	Identification number that maps to the source that added or updated the hostdata. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the host data was detected by RNA • `1` if the host data was provided by a user • `2` if the host data was detected by a third-party scanner • `3` if the host data was provided by a command line tool such as `nmimport.pl` or the Host Input API client

User Vulnerability Change Data Block 4.7+

The User Vulnerability Change data block contains a list of deactivated vulnerabilities for the host, the identification number for the user who deactivated the vulnerabilities, information about the source that supplied the vulnerability changes, and the criticality value. The User Vulnerability Change data block has a block type of 80 in the series 1 group of blocks. Changes from the previous User Vulnerability Change data block include a new source type field and the use of the Generic list data block instead of the List data block to store vulnerability deactivations. This data block is used in user vulnerability change messages as documented in User Set Vulnerabilities Messages for Version 4.6.1+, page 4-48.

The following diagram shows the basic structure of a User Vulnerability Change data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Vulnerability Change Data Block Type (80)																															
	User Vulnerability Change Block Length																															
	Source ID																															
	Source Type																															
Vuln Ack Blocks	Generic List Block Type (31)																															
	Generic List Block Length																															
	User Vulnerability Data Blocks...*																															

The following table describes the fields of the Generic List data block.

Table 4-58 User Vulnerability Change Data Block Fields

Field	Number of Bytes	Description
User Vulnerability Change Data Block Type	uint32	Initiates a User Vulnerability Change data block. This value is always `80`.
User Vulnerability Change Block Length	uint32	Total number of bytes in the User Vulnerability Change data block, including eight bytes for the host vulnerability block type and length fields, plus the number of bytes of host vulnerability data that follows.
Source ID	uint32	Identification number that maps to the source that updated or added the host vulnerability change value. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the host vulnerability data was detected by RNA • `1` if the host vulnerability data was provided by a user • `2` if the host vulnerability data was detected by a third-party scanner • `3` if the host vulnerability data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Type	uint32	Type of vulnerability.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.

Table 4-58 User Vulnerability Change Data Block Fields (continued)

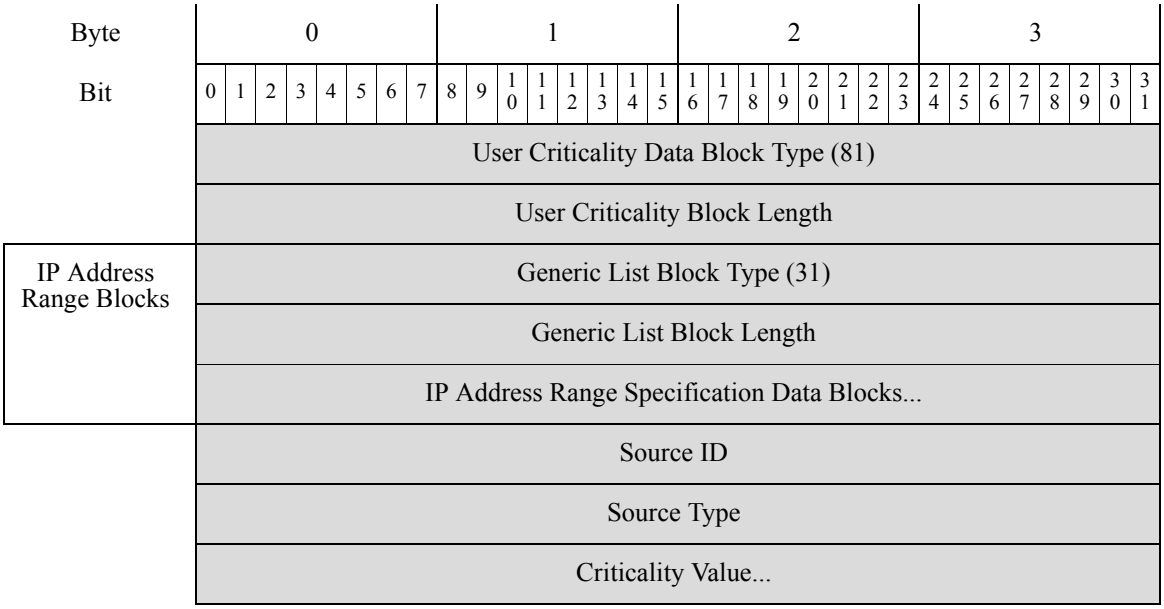
Field	Number of Bytes	Description
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Vulnerability Data Blocks	variable	Encapsulated User Vulnerability data blocks up to the maximum number of bytes in the list block length. For more information, see User Vulnerability Data Block 5.0+, page 4-147.

User Criticality Change Data Block 4.7+

The User Criticality data block is used to contain a list of IP address range specifications for hosts where the host criticality changed, the identification number for the user who updated the criticality value, information about the source that supplied the criticality value, and the criticality value. The User Criticality data block has a block type of 81 in the series 1 group of blocks. Changes from the previous User Criticality data block include a new source type field and the use of the Generic list data block instead of the List data block to store IP addresses.

The User Criticality data block is used in user set host criticality messages as documented in User Set Host Criticality Messages, page 4-50.

The following diagram shows the basic structure of a User Criticality data block:



The following table describes the fields of the User Criticality data block.

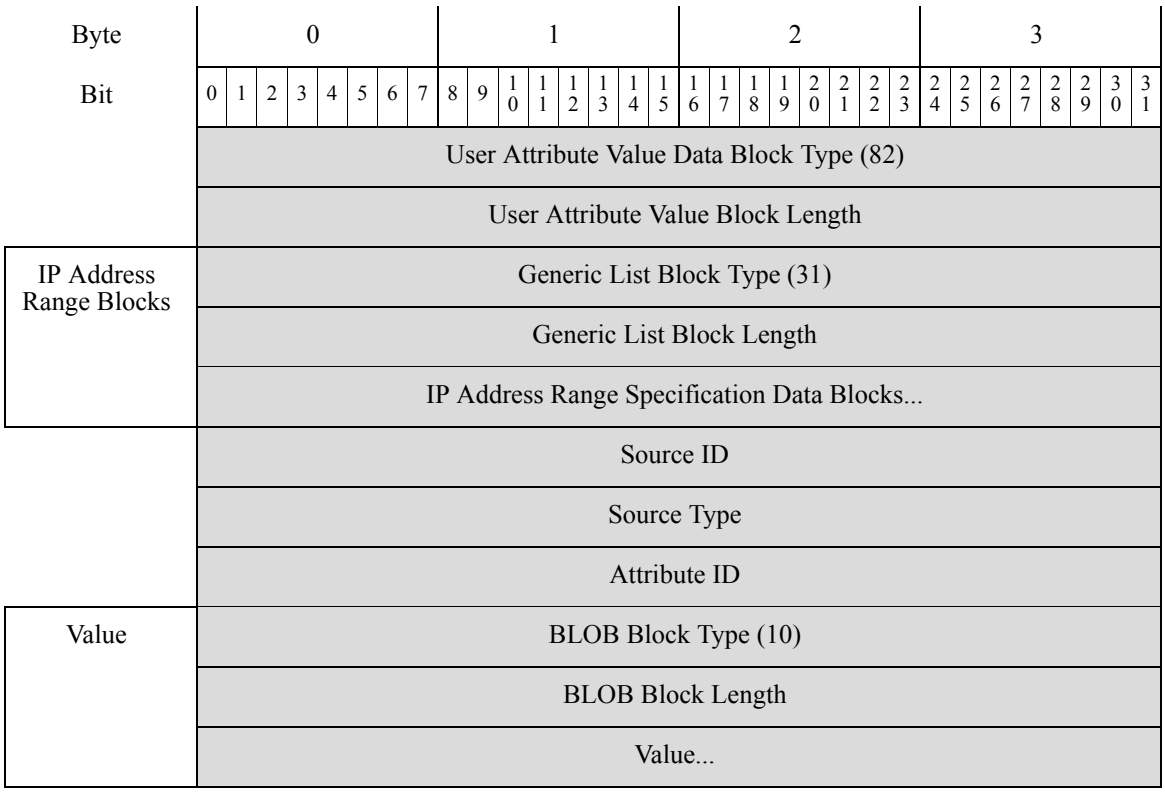
Table 4-59 User Criticality Data Block Fields

Field	Number of Bytes	Description
User Criticality Data Block Type	uint32	Initiates a User Criticality data block. This value is always `81`.
User Criticality Block Length	uint32	Total number of bytes in the User Criticality data block, including eight bytes for the user criticality block type and length fields, plus the number of bytes of user criticality data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length.
Source ID	uint32	Identification number that maps to the source that updated or added the user criticality value. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the user criticality value was provided by RNA • `1` if the user criticality value was provided by a user • `2` if the user criticality value was provided by a third-party scanner • `3` if the user criticality value was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Criticality Value	uint32	User criticality value.

User Attribute Value Data Block 4.7+

The User Attribute Value data block contains a list of IP address ranges that indicate the hosts where the attribute value has changed, together with the identification number for the user who added the attribute value, information about the source that supplied the attribute value, and the BLOB data block containing the attribute value. The User Attribute Value data block has a block type of 82 in the series 1 group of blocks. Changes from the previous User Attribute Value data block include a new source type field and the use of the Generic list data block instead of the List data block to store IP addresses.

The following diagram shows the structure of a User Attribute Value data block:



The following table describes the fields of the User Attribute Value data block.

Table 4-60 User Attribute Value Data Block Fields

Field	Number of Bytes	Description
User Attribute Value Data Block Type	uint32	Initiates a User Attribute Value data block. This value is always `82`.
User Attribute Value Block Length	uint32	Total number of bytes in the Attribute Value data block, including eight bytes for the user attribute value block type and length fields, plus the number of bytes of user attribute value data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	IP Address Range Specification data blocks (each with a start IP address and end IP address) up to the maximum number of bytes in the list block length.
Source ID	uint32	Identification number that maps to the source that added or updated the attribute data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.

Table 4-60 User Attribute Value Data Block Fields (continued)

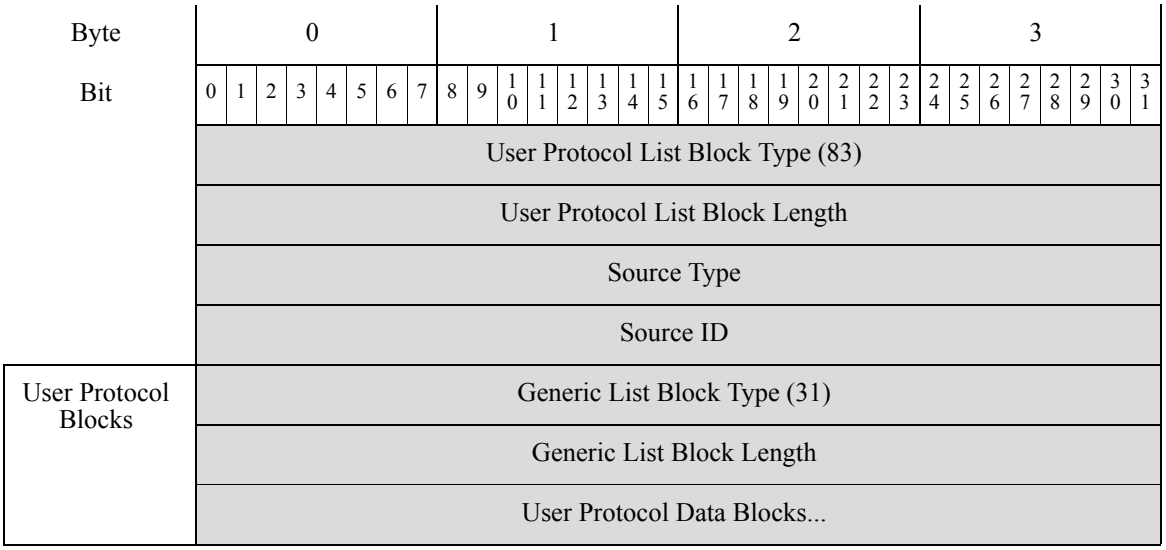
Field	Number of Bytes	Description
Source Type	uint32	Number that maps to the type of data source: • `0` if the user attribute value was provided by RNA • `1` if the user attribute value was provided by a user • `2` if the user attribute value was provided by a third-party scanner • `3` if the user attribute value was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Attribute ID	uint32	Identification number of the updated attribute.
BLOB Block Type	uint32	Initiates a BLOB data block. This value is always `10`.
BLOB Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Value	variable	Contains the user attribute value, in binary format.

User Protocol List Data Block 4.7+

The User Protocol List data block is used to contain information about the source of the protocol data, the identification number for the user who added the data, and the lists of user protocol data blocks. The User Protocol List data block has a block type of 83 in the series 1 group of blocks. For more information on User Protocol data blocks, see User Protocol Data Block, page 4-82.

The User Protocol List data block is used in user protocol messages, as documented in User Protocol Messages, page 4-52.

The following diagram shows the basic structure of a User Protocol List data block:



The following table describes the fields of the Generic List data block.

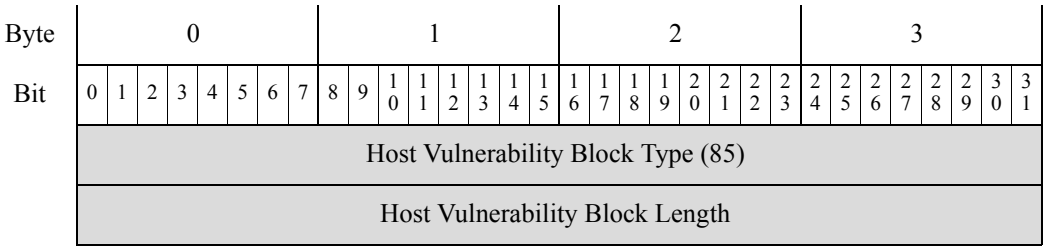
Table 4-61 User Protocol List Data Block Fields

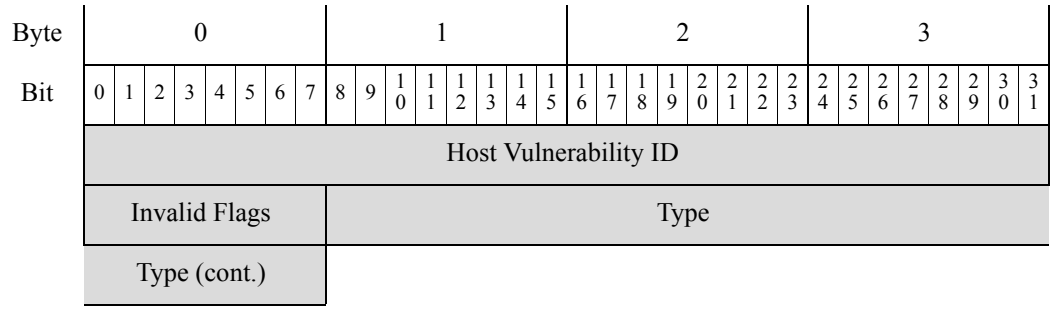
Field	Number of Bytes	Description
User Protocol List Block Type	uint32	Initiates a User Protocol List data block. This value is always `83`.
User Protocol List Block Length	uint32	Total number of bytes in the User Protocol List data block, including eight bytes for the user protocol list block type and length fields, plus the number of bytes of user protocol list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the protocol data was provided by RNA • `1` if the protocol data was provided by a user • `2` if the protocol data was provided by a third-party scanner • `3` if the protocol data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the affected protocols. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Protocol Data Blocks	variable	Encapsulated User Protocol data blocks up to the maximum number of bytes in the list block length.

Host Vulnerability Data Block 4.9.0+

The Host Vulnerability data block conveys vulnerabilities that apply to a host. Each Host Vulnerability data block describes one vulnerability for a host in an event. Host Vulnerability data blocks appear in the Full Host Profile, Full Host Server, and Full Sub-Server data blocks. The Host Vulnerability data block has a block type of 85 in the series 1 group of blocks.

The following diagram shows the format of the Host Vulnerability data block:





The following table describes the components of the Host Vulnerability data block.

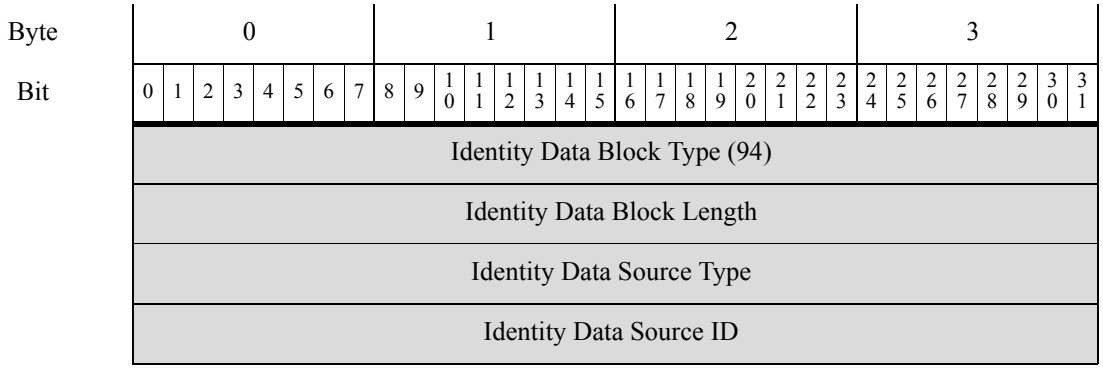
Table 4-62 Host Vulnerability Data Block Fields

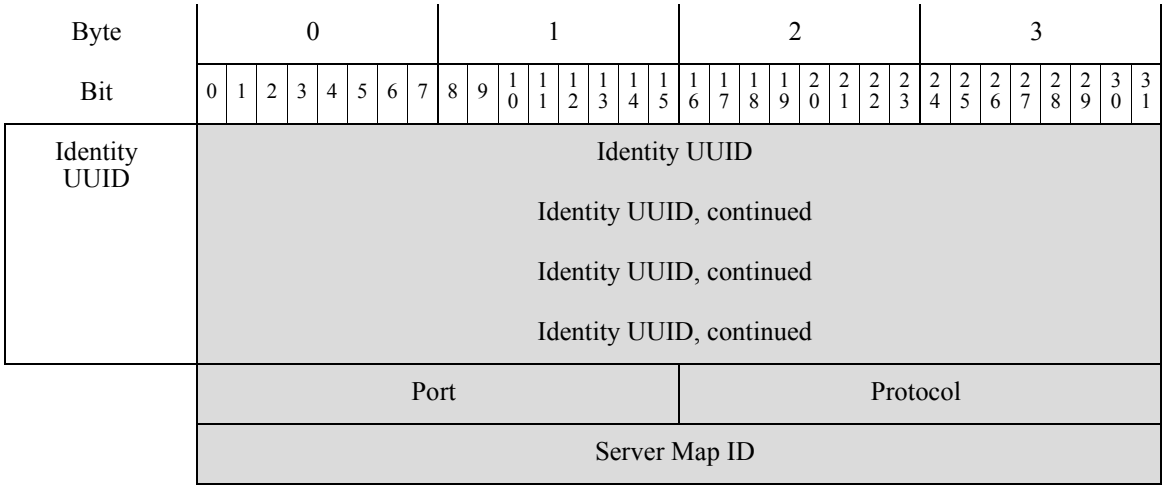
Field	Data Type	Description
Host Vulnerability Block Type	uint32	Initiates an Host Vulnerability data block. This value is always 85.
Host Vulnerability Block Length	uint32	Total number of bytes in the Host Vulnerability data block, including eight bytes for the host vulnerability block type and length fields, plus the number of bytes of host vulnerability data that follows.
Host Vulnerability ID	uint32	The identification number for the vulnerability.
Invalid Flags	uint8	A value indicating whether the vulnerability is valid for the host.
Type	uint32	The type of vulnerability.

Identity Data Block

The identity data block has a block type of 94 in the series 1 group of blocks. Identity data blocks are used in identity conflict and identity timeout messages, which indicate when the identities of an operating system or server fingerprint source conflict or time out. The data block describes reported identities that have been identified as being in conflict with active source identities (user, scanner, or application). For more information, see Identity Conflict and Identity Timeout System Messages, page 4-54.

The following diagram shows the format of an identity data block for 4.9+.





The following table describes the fields of the Cisco identity data block.

Table 4-63 Identity Data Block Fields

Field	Data Type	Description
Identity Data Block Type	uint32	Initiates the Identity data block. This value is always `94`.
Identity Data Block Length	uint32	Number of bytes in the Identity data block. This value should always be 40: sixteen bytes for the data block type and length fields and the source type and ID fields, sixteen bytes for the fingerprint UUID value, two bytes for the port, two bytes for the protocol, and four bytes for the SM ID.
Identity Data Source Type	uint32	Number that maps to the type of data source: • `0` if the fingerprint data was provided by RNA • `1` if the fingerprint data was provided by a user • `2` if the fingerprint data was provided by a third-party scanner • `3` if the fingerprint data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Identity Data Source ID	uint32	Identification number that maps to the source of the fingerprint data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
UUID	uint8[16]	If the identity is an operating system identity, the identification number, in octets, that acts as a unique identifier for the fingerprint.
Port	uint16	If the identity is a server identity, indicates the port used by the packet containing the server data.

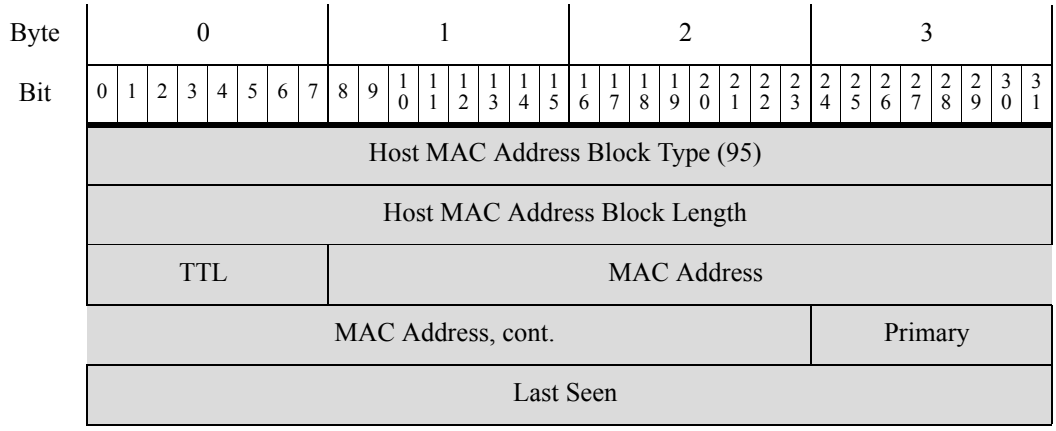
Table 4-63 Identity Data Block Fields (continued)

Field	Data Type	Description
Protocol	uint16	If the identity is a server identity, indicates the IANA number of the network protocol or Ethertype used by the packet containing the server data. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 7 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
Server Map ID	uint32	If the identity is a server identity, indicates the server map ID, representing the combination of ID, vendor, and version for the server.

Host MAC Address 4.9+

The host MAC address data block has a block type of 95 in the series 1 group of blocks. The block includes the time-to-live value for the host data, as well as the MAC address, the primary subnet of the host, and the last seen value for the host.

The following diagram shows the format of a host MAC address data block in 4.9+:



The following table describes the fields of the Host MAC Address data block.

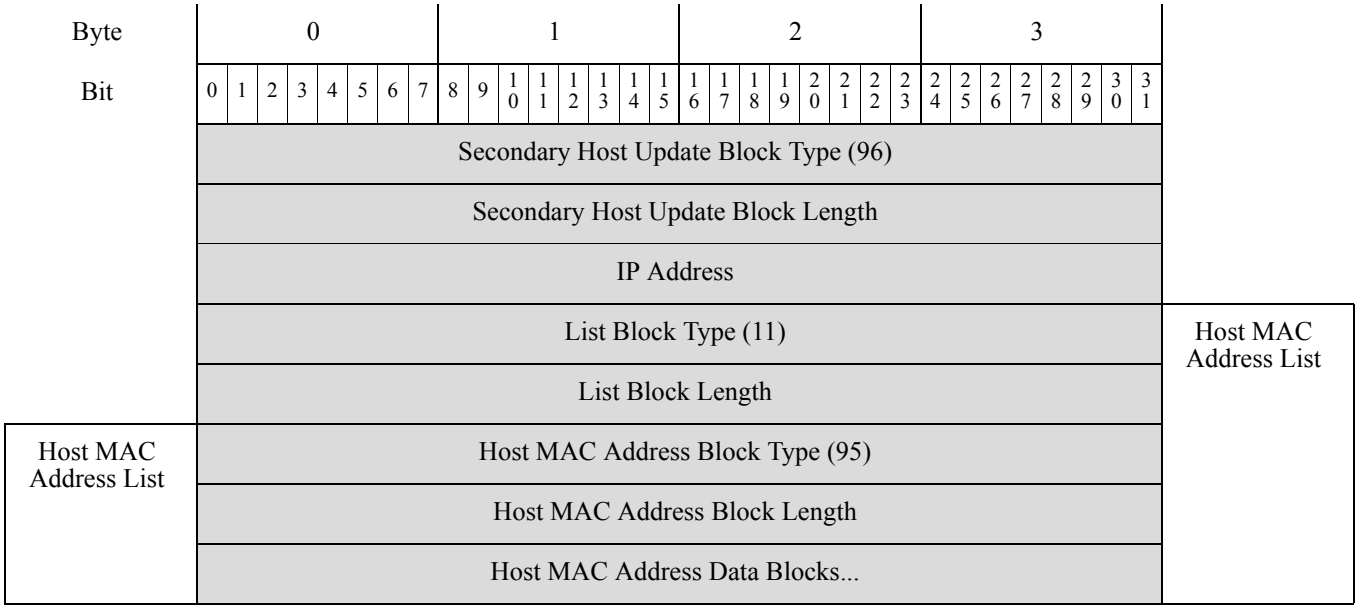
Table 4-64 Host MAC Address Data Block Fields

Field	Data Type	Description
Host MAC Address Data Block Type	uint32	Initiates the Host MAC Address data block. This value is always 95.
Host MAC Address Data Block Length	uint32	Number of bytes in the Host MAC Address data block. This value should always be 20: eight bytes for the data block type and length fields, one byte for the TTL value, 6 bytes for the MAC address, one byte for the primary subnet, and four bytes for the last seen value.
TTL	uint8	Indicates the difference between the TTL value in the packet used to fingerprint the host.
MAC Address	uint8 [6]	Indicates the MAC address of the host.
Primary	uint8	Indicates the primary subnet of the host.
Last Seen	uint32	Indicates when the host was last seen in traffic.

Secondary Host Update

The Secondary Host Update data block contains information for a host sent as a secondary host update from a device monitoring a subnet other than that where the host resides. It is used within Change Secondary Update events (event type 1001, subtype 31). The Secondary Host Update data block has a block type of 96 in the series 1 group of blocks.

The following diagram shows the format of a Secondary Host Update data block:



The following table describes the fields of the Secondary Host Update data block.

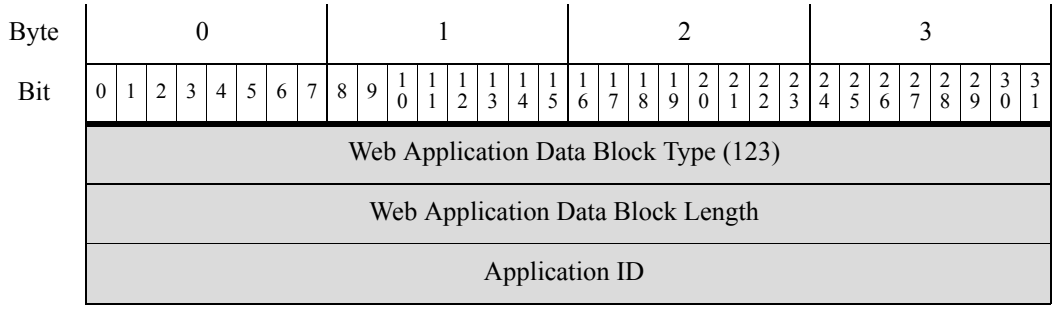
Table 4-65 Secondary Host Update Data Block Fields

Field	Data Type	Description
Secondary Host Update Block Type	uint32	Initiates a Secondary Host Update data block. This value is always 96.
Secondary Host Update Block Length	uint32	Number of bytes in the Secondary Host Update data block, including eight bytes for the secondary host update block type and length fields, plus the number of bytes of secondary host update data that follows.
IP Address	uint8[4]	IP address of the host described in the update, in IP address octets.
List Block Type	uint32	Initiates a List data block comprising Host MAC Address data blocks conveying host MAC address data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Host MAC Address data blocks. This field is followed by zero or more Host MAC Address data blocks.
Host MAC Address Block Type	uint32	Initiates a Host MAC Address data block describing the secondary host. This value is always 95.
Host MAC Address Data Block Length	uint32	Number of bytes in the Host MAC Address data block. This value should always be 20: eight bytes for the data block type and length fields, one byte for the TTL value, six bytes for the MAC address, one byte for the primary subnet, and four bytes for the last seen value.
Host MAC Address Data Blocks	string	Information related to MAC addresses of hosts in the update.

Web Application Data Block for 5.0+

The Web Application data block for 5.0+ has a block type of 123 in the series 1 group of blocks. The data block describes the web application from detected HTTP client requests.

The following diagram shows the format of a Web Application data block in 5.0+.



The following table describes the fields of the Web Application data block.

Table 4-66 Web Application Data Block Fields

Field	Data Type	Description
Web Application Data Block Type	uint32	Initiates the Web Application data block. This value is always `123`.
Web Application Data Block Length	uint32	Number of bytes in the Web Application data block, including eight bytes for the Web Application data block type and length, plus the number of bytes in the application ID field that follows.
Application ID	uint32	Application ID of the web application.

Connection Statistics Data Block 6.0+

The connection statistics data block is used in connection data messages. Several new fields have been added to the Connection Statistics Data Block for 6.0. Fields have been added to support ISE Integration and Multiple Network Maps. The connection statistics data block for version 6.0+ has a block type of 160 in the series 1 group of blocks. It supersedes block type 157, Connection Statistics Data Block 5.4.1, page B-165. New fields have been added to support DNS lookup and Security Intelligence.

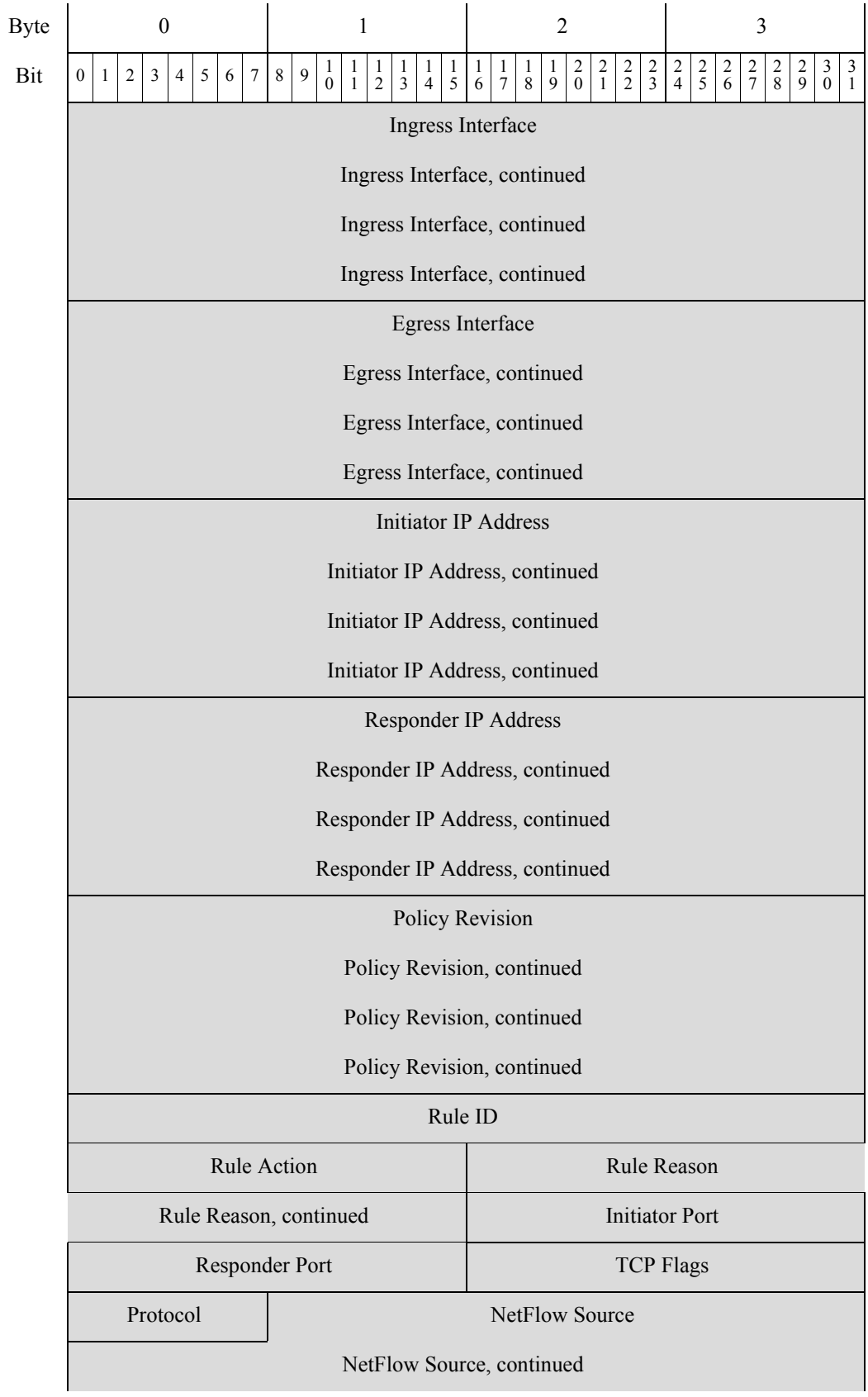
You request connection event records by setting the extended event flag—bit 30 in the Request Flags field—in the request message with an event version of 13 and an event code of 71. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

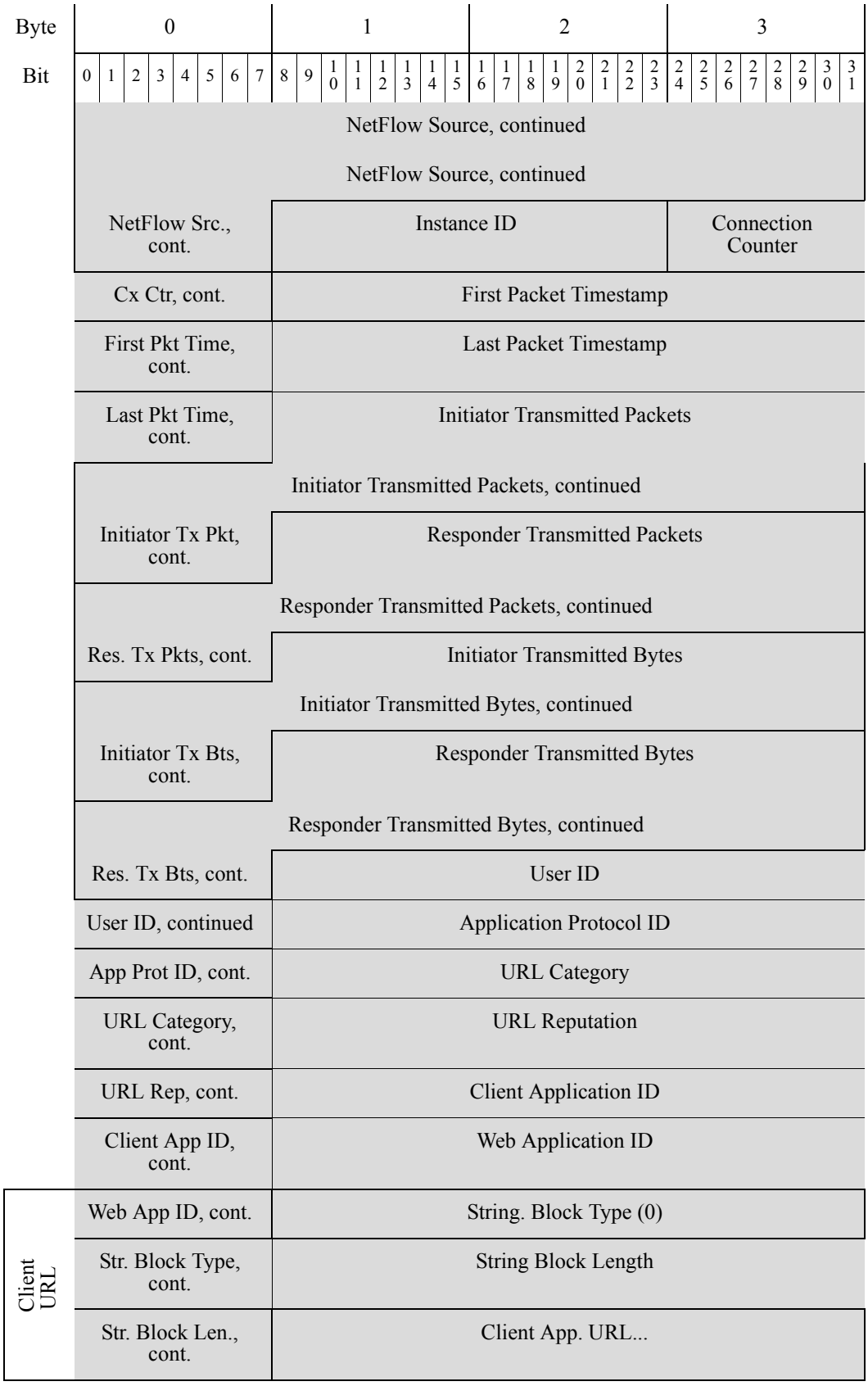
For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

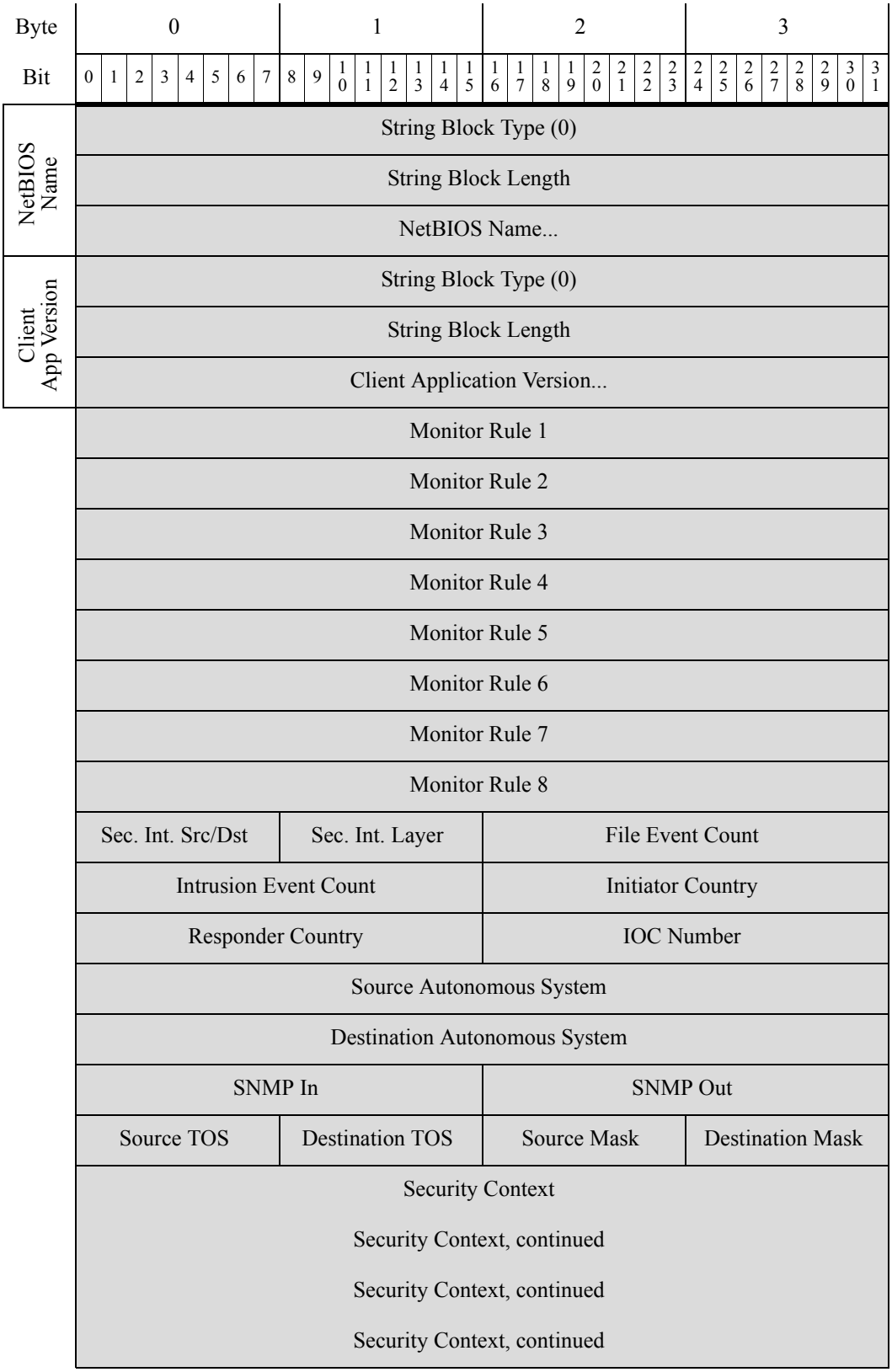
The following diagram shows the format of a Connection Statistics data block for 6.0+:

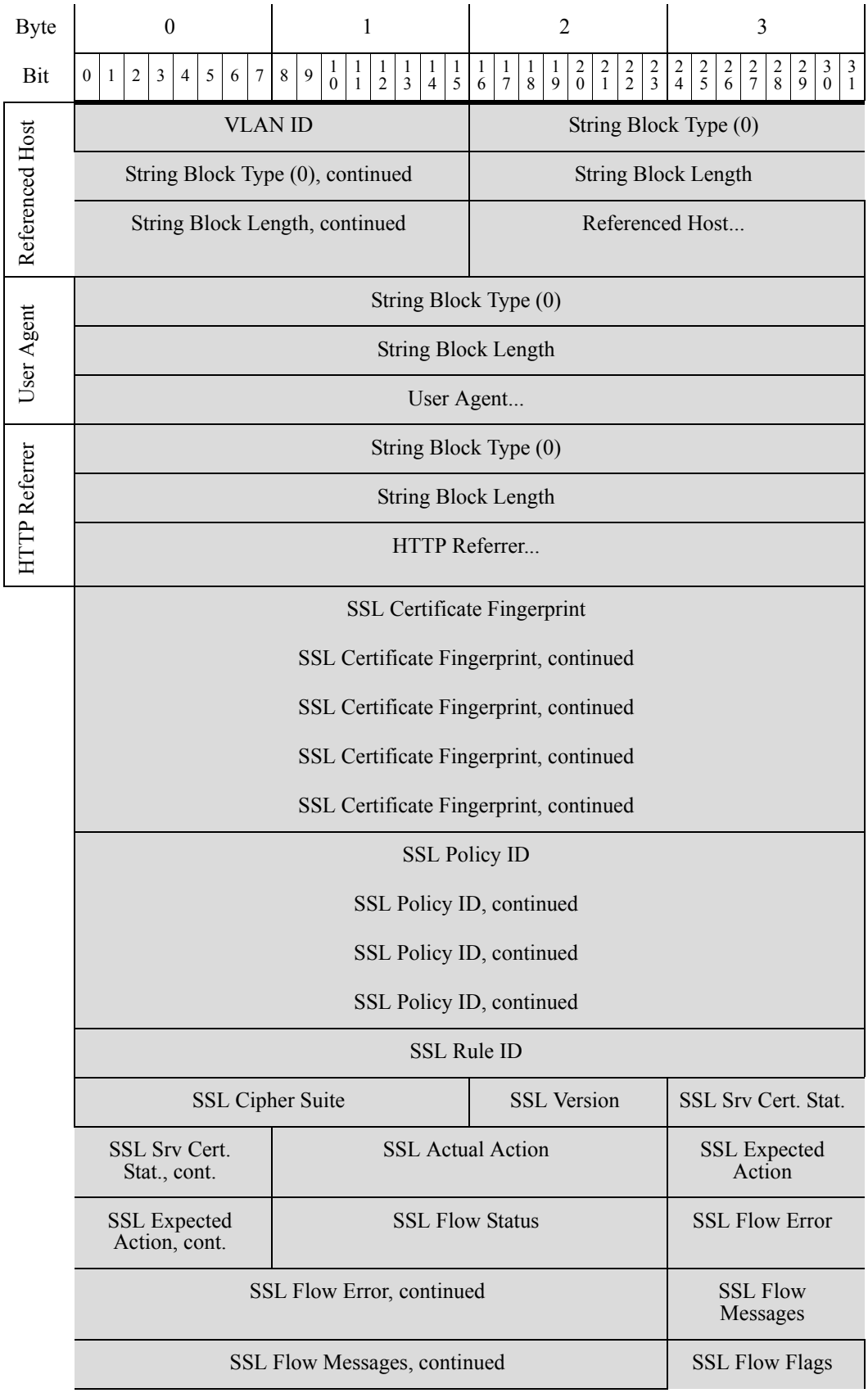
7

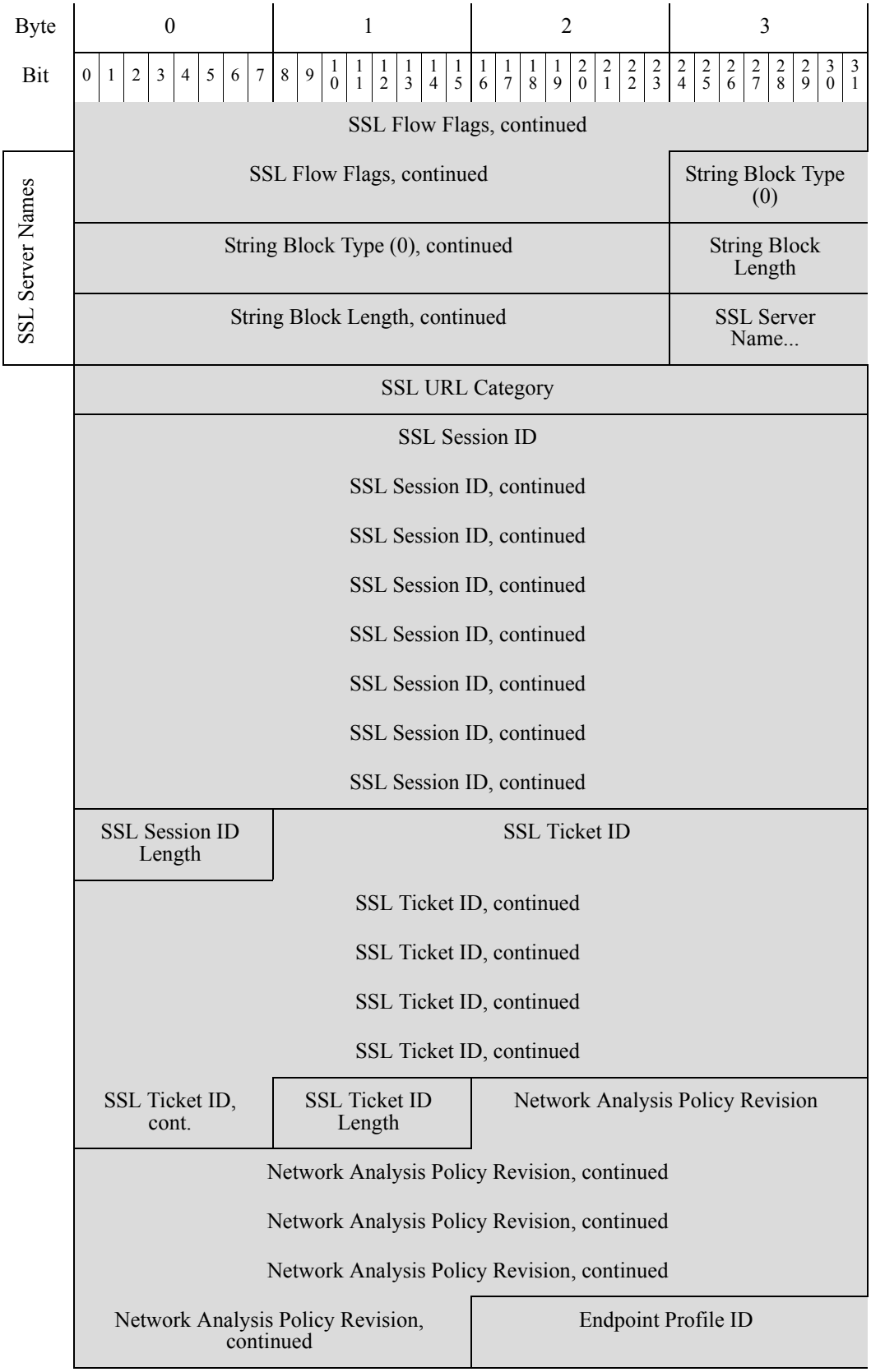
Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Connection Statistics Data Block Type (160)																																
Connection Statistics Data Block Length																																
Device ID																																
Ingress Zone Ingress Zone, continued Ingress Zone, continued Ingress Zone, continued																																
Egress Zone Egress Zone, continued Egress Zone, continued Egress Zone, continued																																

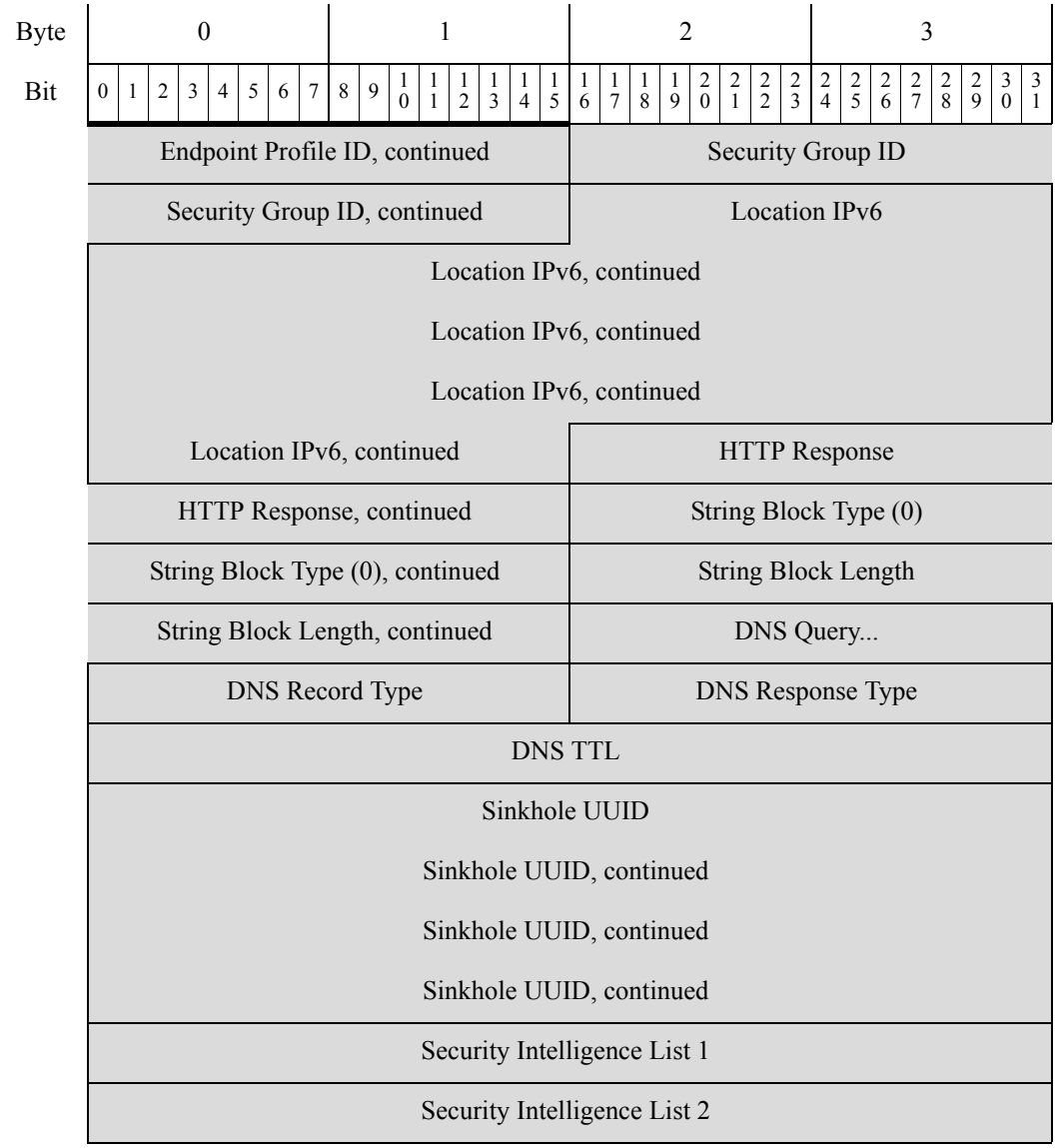

Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Ingress Interface
Ingress Interface, continued
Ingress Interface, continued
Ingress Interface, continued
Egress Interface
Egress Interface, continued
Egress Interface, continued
Egress Interface, continued
Initiator IP Address
Initiator IP Address, continued
Initiator IP Address, continued
Initiator IP Address, continued
Responder IP Address
Responder IP Address, continued
Responder IP Address, continued
Responder IP Address, continued
Policy Revision
Policy Revision, continued
Policy Revision, continued
Policy Revision, continued
Rule ID
Rule Action
Rule Reason
Rule Reason, continued
Initiator Port
Responder Port
TCP Flags
Protocol
NetFlow Source
NetFlow Source, continued


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
NetFlow Source, continued
NetFlow Source, continued
NetFlow Src., cont.
Instance ID
Connection Counter
Cx Ctr, cont.
First Packet Timestamp
First Pkt Time, cont.
Last Packet Timestamp
Last Pkt Time, cont.
Initiator Transmitted Packets
Initiator Transmitted Packets, continued
Initiator Tx Pkt, cont.
Responder Transmitted Packets
Responder Transmitted Packets, continued
Res. Tx Pkts, cont.
Initiator Transmitted Bytes
Initiator Transmitted Bytes, continued
Initiator Tx Bts, cont.
Responder Transmitted Bytes
Responder Transmitted Bytes, continued
Res. Tx Bts, cont.
User ID
User ID, continued
Application Protocol ID
App Prot ID, cont.
URL Category
URL Category, cont.
URL Reputation
URL Rep, cont.
Client Application ID
Client App ID, cont.
Web Application ID
Client URL
Web App ID, cont.
String. Block Type (0)
Str. Block Type, cont.
String Block Length
Str. Block Len., cont.
Client App. URL...


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
NetBIOS Name
String Block Type (0)
String Block Length
NetBIOS Name...
Client App Version
String Block Type (0)
String Block Length
Client Application Version...
Monitor Rule 1
Monitor Rule 2
Monitor Rule 3
Monitor Rule 4
Monitor Rule 5
Monitor Rule 6
Monitor Rule 7
Monitor Rule 8
Sec. Int. Src/Dst
Sec. Int. Layer
File Event Count
Intrusion Event Count
Initiator Country
Responder Country
IOC Number
Source Autonomous System
Destination Autonomous System
SNMP In
SNMP Out
Source TOS
Destination TOS
Source Mask
Destination Mask
Security Context
Security Context, continued
Security Context, continued
Security Context, continued


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Referenced Host
VLAN ID
String Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
Referenced Host...
User Agent
String Block Type (0)
String Block Length
User Agent...
HTTP Referrer
String Block Type (0)
String Block Length
HTTP Referrer...
SSL Certificate Fingerprint
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Policy ID
SSL Policy ID, continued
SSL Policy ID, continued
SSL Policy ID, continued
SSL Rule ID
SSL Cipher Suite
SSL Version
SSL Srv Cert. Stat.
SSL Srv Cert. Stat., cont.
SSL Actual Action
SSL Expected Action
SSL Expected Action, cont.
SSL Flow Status
SSL Flow Error
SSL Flow Error, continued
SSL Flow Messages
SSL Flow Messages, continued
SSL Flow Flags


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SSL Flow Flags, continued
SSL Server Names
SSL Flow Flags, continued
String Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
SSL Server Name...
SSL URL Category
SSL Session ID
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID Length
SSL Ticket ID
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, cont.
SSL Ticket ID Length
Network Analysis Policy Revision
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued
Endpoint Profile ID



The following table describes the fields of the Connection Statistics data block for 6.0+.

Table 4-67 Connection Statistics Data Block 6.0+ Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 6.0+. The value is always `160`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.
Destination Mask	uint8	Destination address prefix mask.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
String Block Type	uint32	Initiates a String data block containing the Referenced Host. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Referenced Host String data block, including eight bytes for the block type and header fields plus the number of bytes in the Referenced Host field.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
Referenced Host	string	Host name information provided in HTTP or DNS.
String Block Type	uint32	Initiates a String data block containing the User Agent. This value is always `0`.
String Block Length	uint32	The number of bytes included in the User Agent String data block, including eight bytes for the block type and header fields plus the number of bytes in the User Agent field.
User Agent	string	Information from the UserAgent header field in the session.
String Block Type	uint32	Initiates a String data block containing the HTTP Referrer. This value is always `0`.
String Block Length	uint32	The number of bytes included in the HTTP Referrer String data block, including eight bytes for the block type and header fields plus the number of bytes in the HTTP Referrer field.
HTTP Referrer	string	The site from which a page originated. This is found int he Referred header information in HTTP traffic.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.
SSL Policy ID	uint8[16]	ID number of the SSL policy that handled the connection.
SSL Rule ID	uint32	ID number of the SSL rule or default action that handled the connection.
SSL Cipher Suite	uint16	Encryption suite used by the SSL connection. The value is stored in decimal format. See `www.iana.org/assignments/tls-parameters/tls-parameters.xhtml` for the cipher suite designated by the value.
SSL Version	uint8	The SSL or TLS protocol version used to encrypt the connection.
SSL Server Certificate Status	uint16	The status of the SSL certificate. Possible values include: • `0` — Not checked — The server certificate status was not evaluated. • `1` — Unknown — The server certificate status could not be determined. • `2` — Valid — The server certificate is valid. • `4` — Self-signed — The server certificate is self-signed. • `16` — Invalid Issuer — The server certificate has an invalid issuer. • `32` — Invalid Signature — The server certificate has an invalid signature. • `64` — Expired — The server certificate is expired. • `128` — Not valid yet — The server certificate is not yet valid. • `256` — Revoked — The server certificate has been revoked.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'
SSL Expected Action	uint16	The action which should be performed on the connection based on the SSL Rule. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
SSL Flow Error	uint32	Detailed SSL error code. These values may be needed for support purposes.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
SSL Flow Messages	uint32	The messages exchanged between client and server during the SSL handshake. See http://tools.ietf.org/html/rfc5246 for more information. • 0x00000001 — NSE_MT__HELLO_REQUEST • 0x00000002 — NSE_MT__CLIENT_ALERT • 0x00000004 — NSE_MT__SERVER_ALERT • 0x00000008 — NSE_MT__CLIENT_HELLO • 0x00000010 — NSE_MT__SERVER_HELLO • 0x00000020 — NSE_MT__SERVER_CERTIFICATE • 0x00000040 — NSE_MT__SERVER_KEY_EXCHANGE • 0x00000080 — NSE_MT__CERTIFICATE_REQUEST • 0x00000100 — NSE_MT__SERVER_HELLO_DONE • 0x00000200 — NSE_MT__CLIENT_CERTIFICATE • 0x00000400 — NSE_MT__CLIENT_KEY_EXCHANGE • 0x00000800 — NSE_MT__CERTIFICATE_VERIFY • 0x00001000 — NSE_MT__CLIENT_CHANGE_CIPHER_SPEC • 0x00002000 — NSE_MT__CLIENT_FINISHED • 0x00004000 — NSE_MT__SERVER_CHANGE_CIPHER_SPEC • 0x00008000 — NSE_MT__SERVER_FINISHED • 0x00010000 — NSE_MT__NEW_SESSION_TICKET • 0x00020000 — NSE_MT__HANDSHAKE_OTHER • 0x00040000 — NSE_MT__APP_DATA_FROM_CLIENT • 0x00080000 — NSE_MT__APP_DATA_FROM_SERVER
SSL Flow Flags	uint64	The debugging level flags for an encrypted connection. Possible values include: • 0x00000001 — NSE_FLOW__VALID - must be set for other fields to be valid • 0x00000002 — NSE_FLOW__INITIALIZED - internal structures ready for processing • 0x00000004 — NSE_FLOW__INTERCEPT - SSL session has been intercepted
String Block Type	uint32	Initiates a String data block containing the SSL Server Name. This value is always 0.

Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the SSL Server Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the SSL Server Name field.
SSL Server Name	string	Name provided in the server name indication in the SSL Client Hello.
SSL URL Category	uint32	Category of the flow as identified from the server name and certificate common name.
SSL Session ID	uint8[32]	Value of the session ID used during the SSL handshake when the client and server agree to do session reuse
SSL Session ID Length	uint8	Length of the SSL Session ID. While the session ID cannot exceed 32 bytes, it may be less than 32 bytes.
SSL Ticket ID	uint8[20]	Hash of the session ticket used when the client and server agree to use a session ticket.
SSL Ticket ID Length	uint8	Length of the SSL Ticket ID. While the ticket ID cannot exceed 20 bytes, it may be less than 20 bytes.
Network Analysis Policy revision	uint8[16]	Revision of the Network Analysis Policy associated with the connection event.
Endpoint Profile ID	uint32	ID number of the type of device used by the connection endpoint as identified by ISE. This is unique for each DC and resolved in metadata.
Security Group ID	uint32	ID number assigned to the user by ISE based on policy.
Location IPv6	uint8[16]	IP address of the interface communicating with ISE. Can be IPv4 or IPv6.
HTTP Response	uint32	Response code of the HTTP Request.
String Block Type	uint32	Initiates a String data block for the DNS query. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the DNS query string.
DNS Query	string	The content of the query sent to the DNS server.
DNS Record Type	uint16	The numerical value for the type of DNS record.

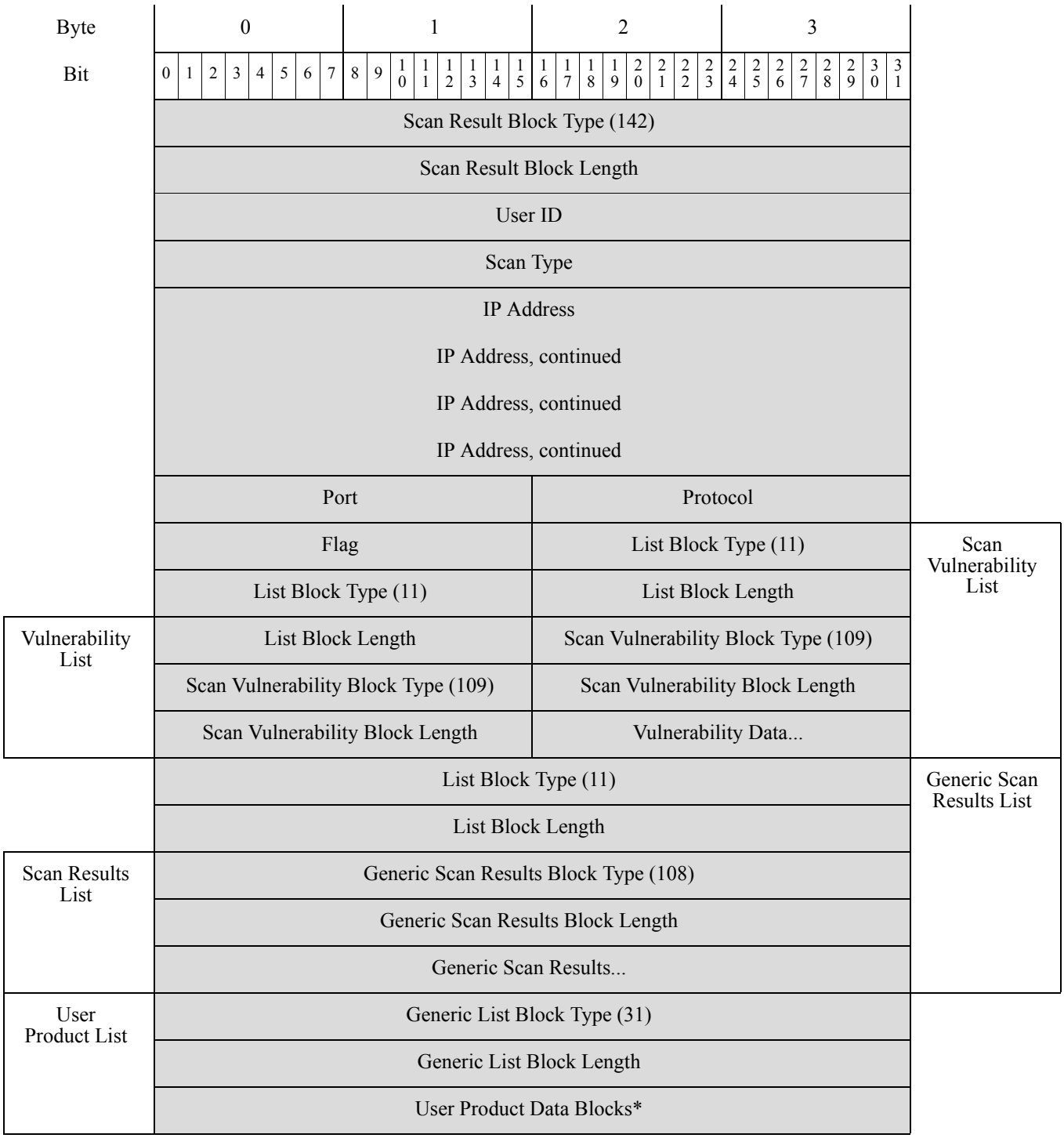
Table 4-67 Connection Statistics Data Block 6.0+ Fields (continued)

Field	Data Type	Description
DNS Response Type	uint16	0 — NoError — No Error 1 — FormErr — Format Error 2 — ServFail — Server Failure 3 — NXDomain — Non-Existent Domain 4 — NotImp — Not Implemented 5 — Refused — Query Refused 6 — YXDomain — Name Exists when it should not 7 — YXRRSet — RR Set Exists when it should not 8 — NXRRSet — RR Set that should exist does not 9 — NotAuth — Not Authorized 10 — NotZone — Name not contained in zone 16 — BADSIG — TSIG Signature Failure 17 — BADKEY — Key not recognized 18 — BADTIME — Signature out of time window 19 — BADMODE — Bad TKEY Mode 20 — BADNAME — Duplicate key name 21 — BADALG — Algorithm not supported 22 — BADTRUNC — Bad Truncation 3841 — NXDOMAIN — NXDOMAIN response from firewall 3842 — SINKHOLE — Sinkhole response from firewall
DNS TTL	uint32	The time to live for the DNS response, in seconds.
Sinkhole UUID	uin8[16]	Revision UUID associated with this sinkhole object.
Security Intelligence List 1	uint32	Security Intelligence List associated with the event. This maps to a Security Intelligence list in associated metadata. There may be two Security Intelligence lists associated with the connection.
Security Intelligence List 2	uint32	Security Intelligence List associated with the event. This maps to a Security Intelligence list in associated metadata. There may be two Security Intelligence lists associated with the connection.

Scan Result Data Block 5.2+

The Scan Result data block describes a vulnerability and is used within Add Scan Result events (event type 1002, subtype 11). The Scan Result data block has a block type of 142 in the series 1 group of blocks. It supersedes block type 102. The IP address field was increased to 16 bytes for version 5.2.

The following diagram shows the format of a Scan Result data block:



The following table describes the fields of the Scan Result data block.

Table 4-68 Scan Result Data Block Fields

Field	Data Type	Description
Scan Result Block Type	uint32	Initiates a Scan Result data block. This value is always 142.
Scan Result Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
User ID	uint32	Contains the user identification number for the user who imported the scan result or ran the scan that produced the scan result.
Scan Type	uint32	Indicates how the results were added to the system.
IP Address	uint8[16]	IP address of the host affected by the vulnerabilities in the result, in IP address octets.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
Flag	uint16	Reserved
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block describing a vulnerability detected during a scan. This value is always 109.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes in the scan vulnerability data that follows.
Vulnerability Data	string	Information relating to each vulnerability.
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.

Table 4-68 Scan Result Data Block Fields (continued)

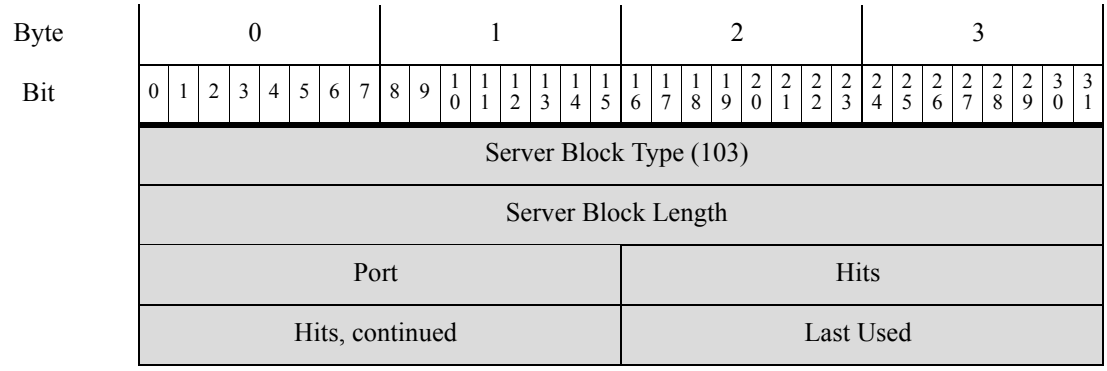
Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Generic Scan Results Block Type	uint32	Initiates a Generic Scan Results data block describing server and operating system data detected during a scan. This value is always `108`.
Generic Scan Results Block Length	uint32	Number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes in the scan result data that follows.
Generic Scan Results Data	string	Information relating to each scan result.
Generic List Block Type	uint32	Initiates a Generic List data block comprising User Product data blocks conveying host input data from a third-party application. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated User Product data blocks.
User Product Data Blocks *	variable	User Product data blocks containing host input data. See User Product Data Block 5.1+, page 4-160 for a description of this data block.

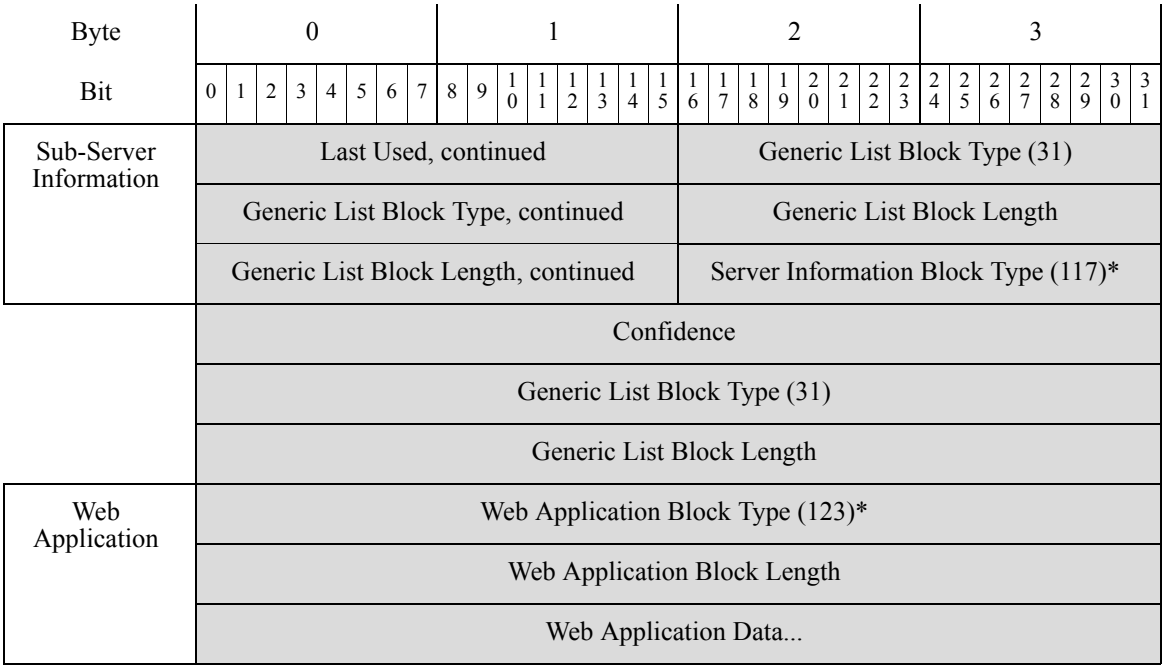
Host Server Data Block 4.10.0+

The Host Server data block conveys information about the detected servers on a host. It contains a block for each detected server, and also includes a list of web application data blocks for the web applications the server is running. Host Server data blocks are contained in messages for new and changed TCP and UDP servers. For more information, see Server Messages, page 4-40. The Host Server data block has a block type of 103 in the series 1 group of blocks.

Note An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Host Server data block:





The following table describes the fields of the Host Server data block.

Table 4-69 Host Server Data Block Fields

Field	Data Type	Description
Host Server Block Type	uint32	Initiates a Host Server data block. This value is always `103`.
Host Server Block Length	uint32	Total number of bytes in the Host Server data block, including the eight bytes in the Host Server block type and length fields, plus the number of bytes of data that follows.
Port	uint16	Port number where the server runs.
Hits	uint32	Number of hits the server has received.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated sub-server information data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Server Information Data Blocks*	variable	Server information data blocks up to the maximum number of bytes in the list block length. For details, see Server Information Data Block for 4.10.x, 5.0 - 5.0.2, page 4-134.
Confidence	uint32	Confidence percentage.
Generic List Block Type	uint32	Initiates a Generic data block. This value is always `31`.

Table 4-69 Host Server Data Block Fields (continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic block and encapsulated web application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated web application data blocks.
Web Application Data Blocks*	variable	Encapsulated web application data blocks up to the maximum number of bytes in the list block length. For details, see Web Application Data Block for 5.0+, page 4-109.

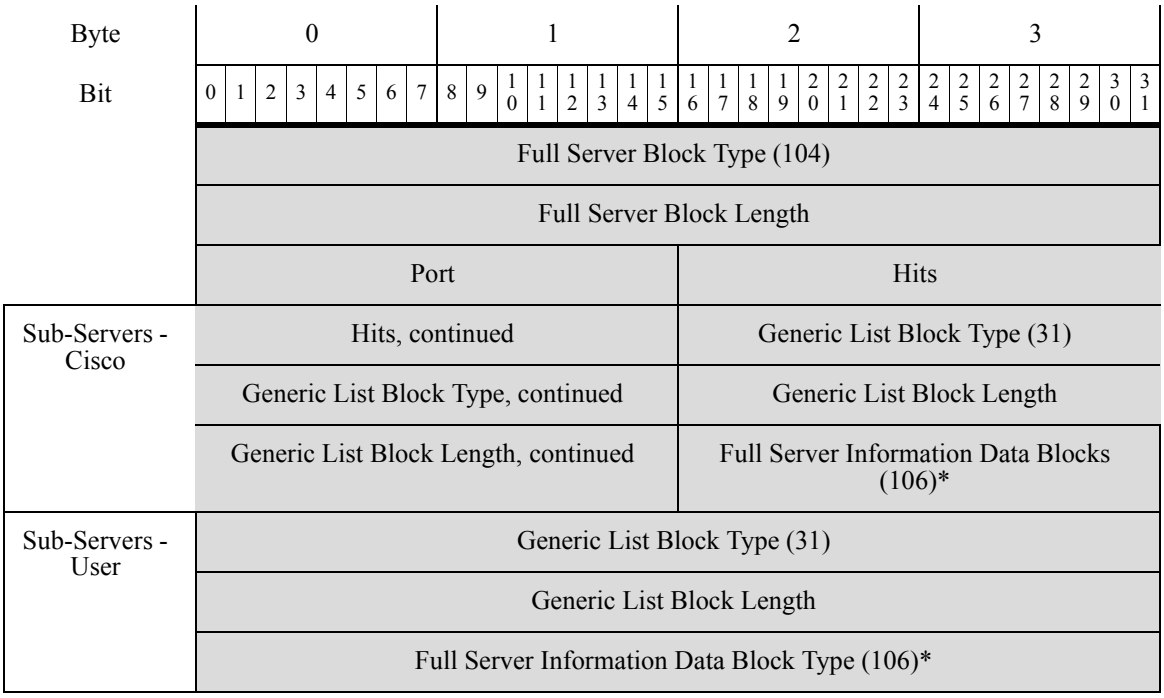
Full Host Server Data Block 4.10.0+

The Full Host Server data block conveys information about a server, including the server port, the frequency of use and most recent update, confidence of data accuracy, and Cisco and third-party vulnerabilities related to that server for the host. The Full Host Server data block contains a Full Sub-Server Information data block for each sub-server on the server. Each Full Host Profile data block contains a Full Host Server data block for each TCP and UDP server on the host. The Full Host Server data block has a block type of 104 in the series 1 group of blocks.

Note An asterisk(*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server data block:

::



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Sub-Servers - Scanner	Generic List Block Type (31)																															
	Generic List Block Length																															
	Full Server Information Data Blocks (106)*																															
Sub-Servers - Application	Generic List Block Type (31)																															
	Generic List Block Length																															
	Full Server Information Data Blocks (106)*																															
	Confidence																															
Server Banner	BLOB Block Type (10)																															
	BLOB Block Length																															
	Server Banner Data...																															
VDB Vulnerability	Generic List Block Type (31)																															
	Generic List Block Length																															
	(VDB) Host Vulnerability Data Blocks (85)*																															
Third Pty/VDB Vulnerability	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*																															
Third Pty Host Vulnerability	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party) Host Vulnerability Data Blocks (85)*																															
Web Application	Generic List Block Type (31)																															
	Generic List Block Length																															
	Web Application Data (123)*																															

The following table describes the components of the Full Server data block.

Table 4-70 Full Server Data Block 4.10.0+ Fields

Field	Data Type	Description
Full Server Block Type	uint32	Initiates a Full Server data block. This value is always 104.
Full Server Block Length	uint32	Total number of bytes in the Full Server data block, including eight bytes for the full server block type and length fields, plus the number of bytes of full server data that follows.
Port	uint16	Server port number.
Hits	uint32	Number of hits the server has received.
Generic List Block Type	uint32	Initiates a Generic List data block comprising data blocks of detected sub-server data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.
Sub-Server Information - Cisco Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers for a host server detected by Cisco. See Full Server Information Data Block, page 4-136 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Sub-Server Information- User Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by a user. See Full Server Information Data Block, page 4-136 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by a scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.
Sub-Server Information- Scan Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by a scanner. See Full Server Information Data Block, page 4-136 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.

Table 4-70 Full Server Data Block 4.10.0+ Fields (continued)

Field	Data Type	Description
Sub-Server Information - Application Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by an application. See Full Server Information Data Block, page 4-136 for a description of this data block.
Confidence	uint32	Percentage of confidence of Cisco in correct identification of the full server data.
BLOB Block Type	uint32	Initiates a BLOB data block, which contains banner data. This value is always 10.
BLOB Block Length	uint32	Total number of bytes in the BLOB data block, including eight bytes for the block type and length fields, plus the number of bytes in the banner.
Server Banner Data	byte[*n*]	First *n* bytes of the packet involved in the server event, where *n* is equal to or less than 256.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Cisco vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities in the vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party host vulnerability data sourced from a third-party scanner and containing vulnerability information already cataloged in the VDB. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third-party scanner and containing information about host vulnerabilities cataloged in the vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party host vulnerability data generated by a third-party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing third-party vulnerability data for vulnerabilities identified by a third-party scanner but not cataloged in the VDB. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.

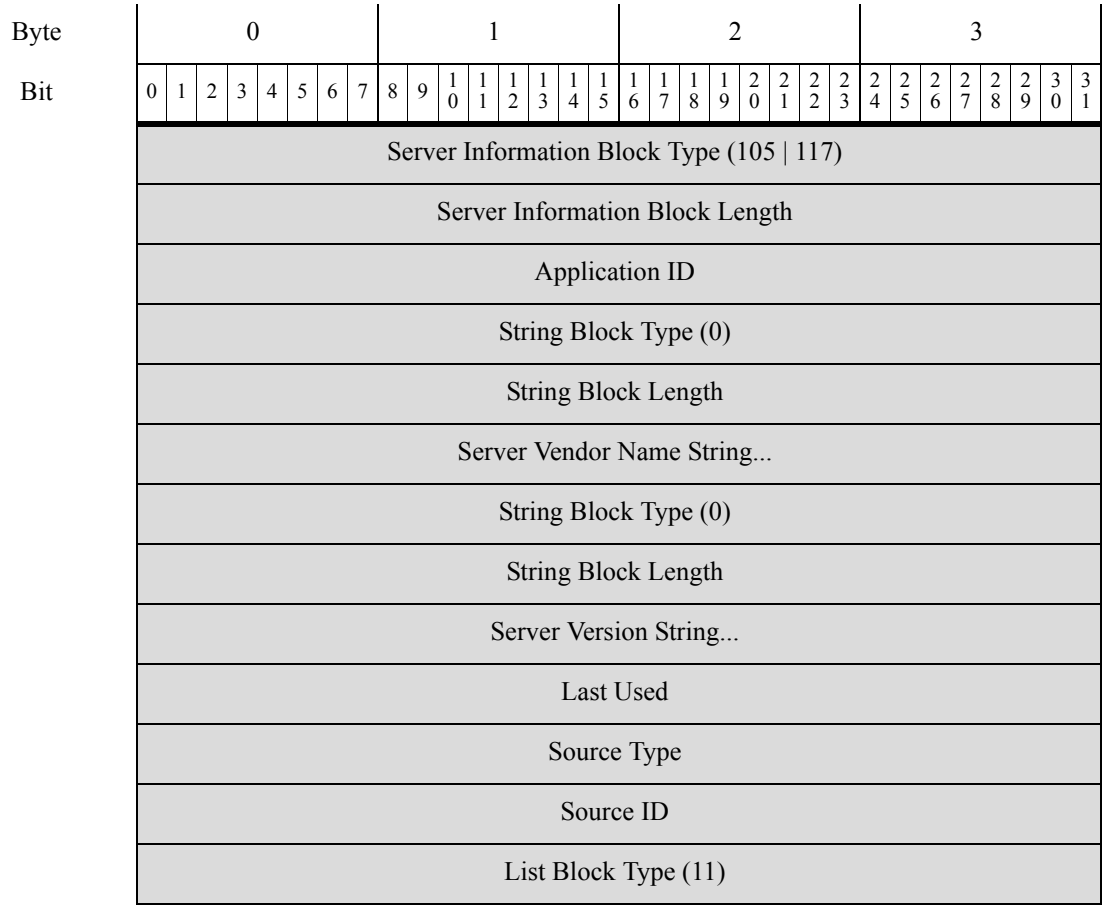
Table 4-70 Full Server Data Block 4.10.0+ Fields (continued)

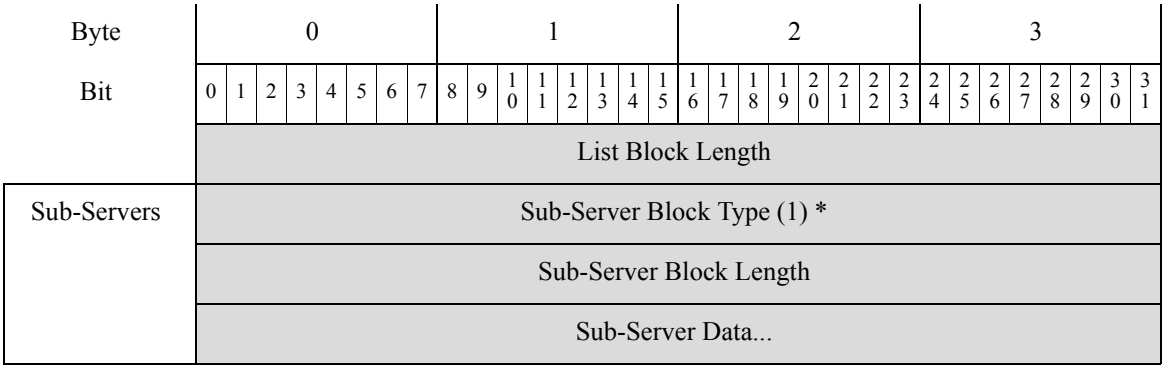
Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Web Application Data Blocks*	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length.

Server Information Data Block for 4.10.x, 5.0 - 5.0.2

The Server Information data block conveys information about a server, including the server ID, server vendor and version, and source information. The Server Information data block has a block type of 105 in the series 1 group of blocks for 4.10.x and a block type of 117 in the series 1 group of blocks for 5.0 - 5.0.2. Server information data blocks are conveyed in lists within Host Server blocks and Full Host server data blocks. For more information see Host Server Data Block 4.10.0+, page 4-128 and Full Host Server Data Block 4.10.0+, page 4-130.

The following diagram shows the format of the Server Information data block:





The following table describes the components of the Server Information data block.

Table 4-71 Server Information Data Block Fields

Field	Data Type	Description
Server Information Block Type	uint32	Initiates a Server Information data block. The block type is `105` for 4.10.x and 117 for 5.0+.
Server Information Block Length	uint32	Total number of bytes in the Server Information data block, including eight bytes for the Server Information block type and length fields, four bytes for the server ID, eight bytes for the vendor name block type and length, another four for the vendor name, eight bytes for the version string block type and length, another four for the version string, and four bytes each for the last used, source type, and source ID fields.
Application ID	uint32	The application ID for the application protocol running on the detected server.
String Block Type	uint32	Initiates a String data block containing the server vendor's name. This value is always `0`.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server vendor name.
Server Vendor Name	string	Name of the server vendor.
String Block Type	uint32	Initiates a String data block that contains the server version. This value is always `0`.
String Block Length	uint32	Number of bytes in the server version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server version.
Server Version	string	Server version.
Last Time Used	uint32	Indicates when the server information was last used in traffic.

Table 4-71 Server Information Data Block Fields (continued)

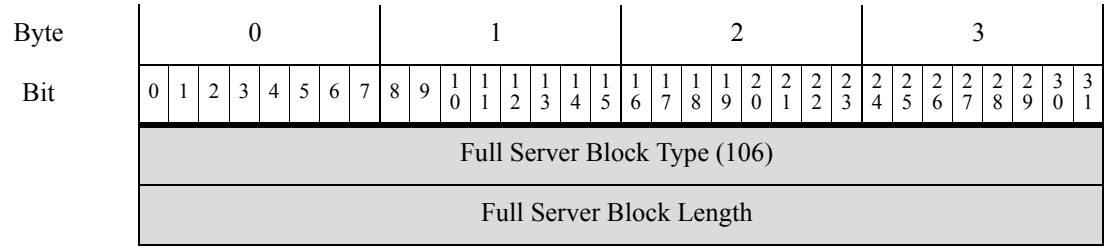
Field	Data Type	Description
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was provided by RNA • `1` if the server data was provided by a user • `2` if the server data was provided by a third-party scanner • `3` if the server data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
List Block Type	uint32	Initiates a list of Sub-Server data blocks. This value is always `11`.
List Block Length	uint32	Number of bytes in the List data block, including eight bytes for the list block type and length fields, plus the number of bytes in the encapsulated Sub-Server data blocks that follow.
Sub-Server Block Type	uint32	Initiates the first Sub-Server data block. This data block can be followed by other Sub-Server data blocks up to the limit defined in the list block length field.
Sub-Server Block Length	uint32	Total number of bytes in each Sub-Server data block, including the eight bytes in the Sub-Server block type and length fields, plus the number of bytes of data that follows.
Sub-Server Data	variable	Sub-server data as documented in Sub-Server Data Block, page 4-67.

Full Server Information Data Block

The Full Server Information data block conveys information about a server detected on a host, including the server's application protocol, vendor, and version, and the list of its associated sub-servers. For each sub-server, information is included by a Full Sub-Server data block (see Full Sub-Server Data Block, page 4-75). The Full Server Information data block has a block type of 106 in the series 1 group of blocks.

Note An asterisk(*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server Information data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Application Protocol ID																															
Vendor	String Block Type (0)																															
	String Block Length																															
	Vendor Name String...																															
Version	String Block Type (0)																															
	String Block Length																															
	Version String...																															
	Last Used																															
	Source Type																															
	Source ID																															
	List Block Type (11)																															
	List Block Length																															
Sub-Servers	Full Sub-Server Block Type (51) *																															
	Full Sub-Server Block Length																															
	Full Sub-Server Data...																															

The following table describes the components of the Full Server Information data block.

Table 4-72 Full Server Information Data Block Fields

Field	Data Type	Description
Full Server Information Block Type	uint32	Initiates a Full Server Information data block. This value is always `106`.
Full Server Information Block Length	uint32	Total number of bytes in the Full Server Information data block, including eight bytes for the full server block type and length fields, plus the number of bytes in the full server data that follows.
Application Protocol ID	uint32	The application ID of the application protocol running on the server.
String Block Type	uint32	Initiates a String data block containing the application protocol vendor's name. This value is always `0`.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.

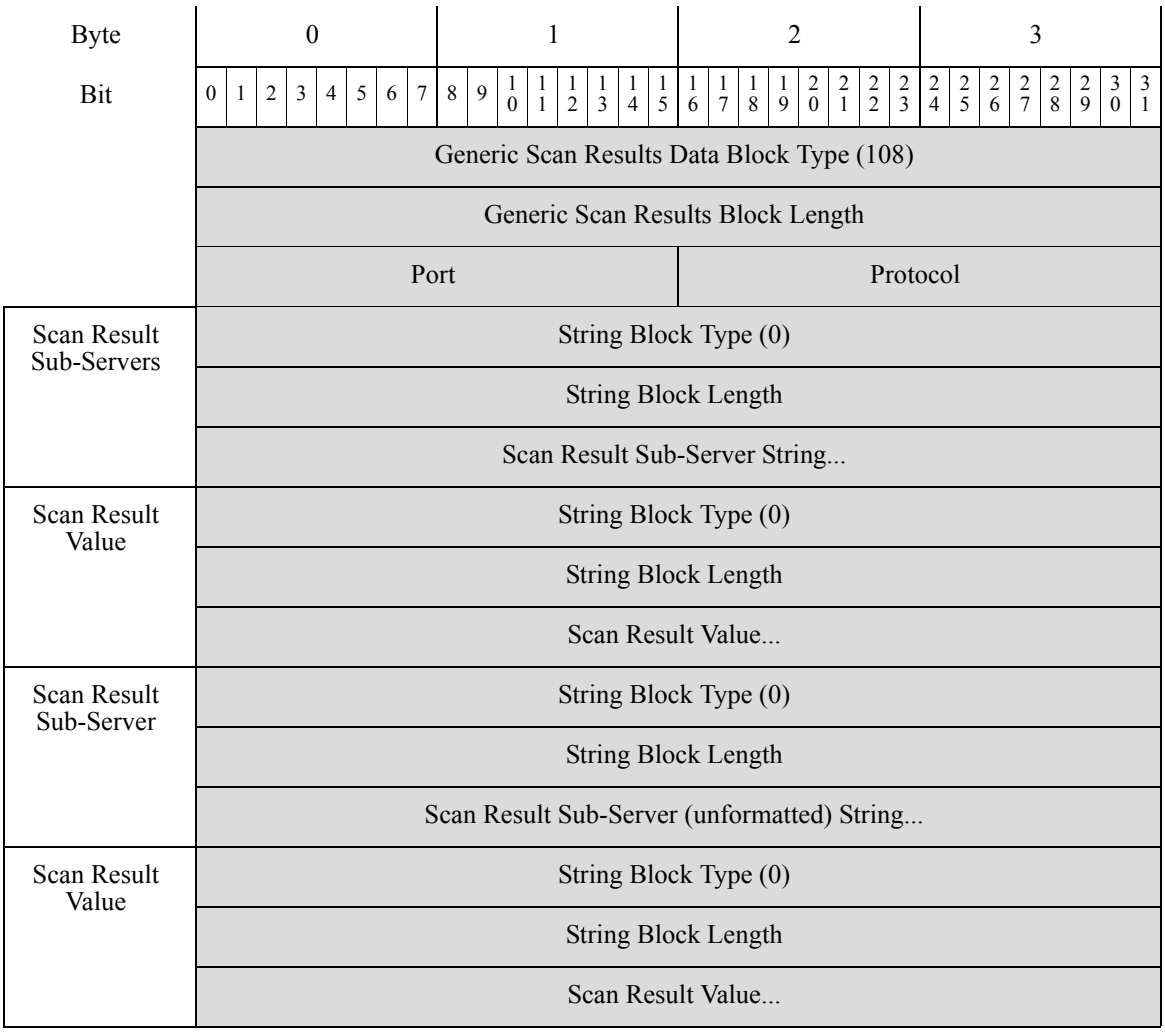
Table 4-72 Full Server Information Data Block Fields (continued)

Field	Data Type	Description
Vendor Name	string	Name of the server vendor.
String Block Type	uint32	Initiates a String data block that contains the application protocol version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Version	string	The version of the server.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was provided by RNA • `1` if the server data was provided by a user • `2` if the client data was provided by a third-party scanner • `3` if the server data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
List Block Type	uint32	Initiates a List data block comprising Full Server Information data blocks conveying sub-server data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Full Sub-Server data blocks. This field is followed by zero or more Full Sub-Server data blocks.
Full Sub-Server Block Type	uint32	Initiates the first Full Sub-Server data block. This data block can be followed by other Full Sub-Server data blocks up to the limit defined in the list block length field.
Full Sub-Server Block Length	uint32	Total number of bytes in each Full Sub-Server data block, including the eight bytes in the Full Sub-Server block type and length fields, plus the number of bytes of data that follows.
Full Sub-Server Data Blocks *	uint32	Full Sub-Server data blocks containing sub-servers for the server. See Full Sub-Server Data Block, page 4-75 for a description of this data block.

Generic Scan Results Data Block for 4.10.0+

The Generic Scan Results data block contains scan results and is used in the The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116. The Generic Scan Results data block has a block type of 108 in the series 1 group of blocks.

The following diagram shows the basic structure of a Generic Scan Results data block:



The following table describes the fields of the Generic Scan Results data block.

Table 4-73 Generic Scan Result Data Block Fields

Field	Number of Bytes	Description
Generic Scan Results Data Block Type	uint32	Initiates a Generic Scan Results data block. This value is always `108`.
Generic Scan Results Block Length	uint32	Total number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes of scan results data that follows.
Port	uint16	Port used by the server affected by the vulnerabilities in the results.

Table 4-73 Generic Scan Result Data Block Fields (continued)

Field	Number of Bytes	Description
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • `6` — TCP • `17` — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • `2048` — IP
String Block Type	uint32	Initiates a String data block that contains the sub-server. This value is always `0`.
String Block Length	uint32	Number of bytes in the sub-server String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server.
Scan Result Sub-Server	string	Sub-server.
String Block Type	uint32	Initiates a String data block that contains the value. This value is always `0`.
String Block Length	uint32	Number of bytes in the value String data block, including eight bytes for the block type and length fields, plus the number of bytes in the value.
Scan result value	string	Scan result value.
String Block Type	uint32	Initiates a String data block that contains the sub-server. This value is always `0`.
String Block Length	uint32	Number of bytes in the sub-server String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server.
Scan Result Sub-Server	string	Sub-server (unformatted).
String Block Type	uint32	Initiates a String data block that contains the value. This value is always `0`.
String Block Length	uint32	Number of bytes in the value String data block, including eight bytes for the block type and length fields, plus the number of bytes in the value.
Scan Result Value	string	Scan result value (unformatted).

Scan Vulnerability Data Block for 4.10.0+

The Scan Vulnerability data block describes a vulnerability and is used within Scan Result data blocks, which in turn are used in Add Scan Result events (event type 1002, subtype 11). For more information, see The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116 and Add Scan Result Messages, page 4-53. The Scan Vulnerability data block has a block type of 109 in the series 1 group of blocks.

The following diagram shows the format of a Scan Vulnerability data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Scan Vulnerability Block Type (109)																															
	Scan Vulnerability Block Length																															
	Port																Protocol															
ID	String Block Type (0)																															
	String Block Length																															
	ID																															
Name	String Block Type (0)																															
	String Block Length																															
	Vulnerability Name...																															
Description	String Block Type (0)																															
	String Block Length																															
	Description...																															
Name Clean	String Block Type (0)																															
	String Block Length																															
	Vulnerability Name Clean...																															
Description Clean	String Block Type (0)																															
	String Block Length																															
	Description Clean...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bugtraq ID	List Block Type (11)																															
	List Block Length																															
	Integer Data Blocks (Bugtraq IDs)...																															
CVE ID	List Block Type (11)																															
	List Block Length																															
	CVE ID...																															

The following table describes the fields of the Scan Vulnerability data block.

Table 4-74 Scan Vulnerability Data Block Fields

Field	Data Type	Description
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block. This value is always `109`.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
Port	uint16	Port used by the sub-server affected by the vulnerability.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • `6` — TCP • `17` — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • `2048` — IP
String Block Type	uint32	Initiates a String data block for the ID.
String Block Length	uint32	Number of bytes in the String data block for the ID, including eight bytes for the string block type and length, plus the number of bytes in the ID.
ID	string	The ID for the reported vulnerability as specified by the scan utility that detected it. For a vulnerability detected by a Qualys scan, for example, this field indicates the Qualys ID.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.

Table 4-74 Scan Vulnerability Data Block Fields (continued)

Field	Data Type	Description
Name	string	Name of the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability description.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability description, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability description.
Description	string	Description of the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Name Clean	string	Name of the vulnerability (unformatted).
String Block Type	uint32	Initiates a String data block for the vulnerability description.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability description, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability description.
Description Clean	string	Description of the vulnerability (unformatted).
List Block Type	uint32	Initiates a List data block for the list of Bugtraq identification numbers.
List Block Length	uint32	Number of bytes in the List data block for the list of Bugtraq identification numbers, including eight bytes for the string block type and length, plus the number of bytes in the Integer data blocks containing the Bugtraq IDs.
Bugtraq ID	string	Contains zero or more Integer (INT32) data blocks that form a list of Bugtraq identification numbers. For more information on these data blocks, see Integer (INT32) Data Block, page 4-69.
List Block Type	uint32	Initiates a List data block for the list of Common Vulnerability Exposure (CVE) identification numbers.
List Block Length	uint32	Number of bytes in the List data block for the CVE identification number, including eight bytes for the string block type and length, plus the number of bytes in the CVE identification number.
CVE ID	string	Contains zero or more String Information data blocks that form a list of CVE identification numbers. For more information on these data blocks, see String Information Data Block, page 4-71.

Full Host Client Application Data Block 5.0+

The Full Host Client Application data block for version 5.0+ describes a client application, plus an appended list of associated web applications and vulnerabilities. The Full Host Client Application data block is used within the Full Host Profile data block (type 111). It has a block type of 112 in the series 1 group of blocks.

The following diagram shows the basic structure of a Full Host Client Application data block for 5.0+:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Full Host Client Application Block Type (112)																															
	Full Host Client Application Block Length																															
	Hits																															
	Last Used																															
	Application ID																															
Version	String Block Type (0)																															
	String Block Length																															
	Version...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Web Application	Web Application Block Type (123)*																															
	Web Application Block Length																															
	Web Application Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Vulnerability	Vulnerability Block Type (85)*																															
	Vulnerability Block Length																															
	Vulnerability Data...																															

The following table describes the fields of the Full Host Client Application data block.

Table 4-75 Full Host Client Application Data Block 5.0+ Fields

Field	Data Type	Description
Full Host Client Application Block Type	uint32	Initiates a Full Host Client Application data block. This value is always `112`.
Full Host Client Application Block Length	uint32	Number of bytes in the Full Host Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.

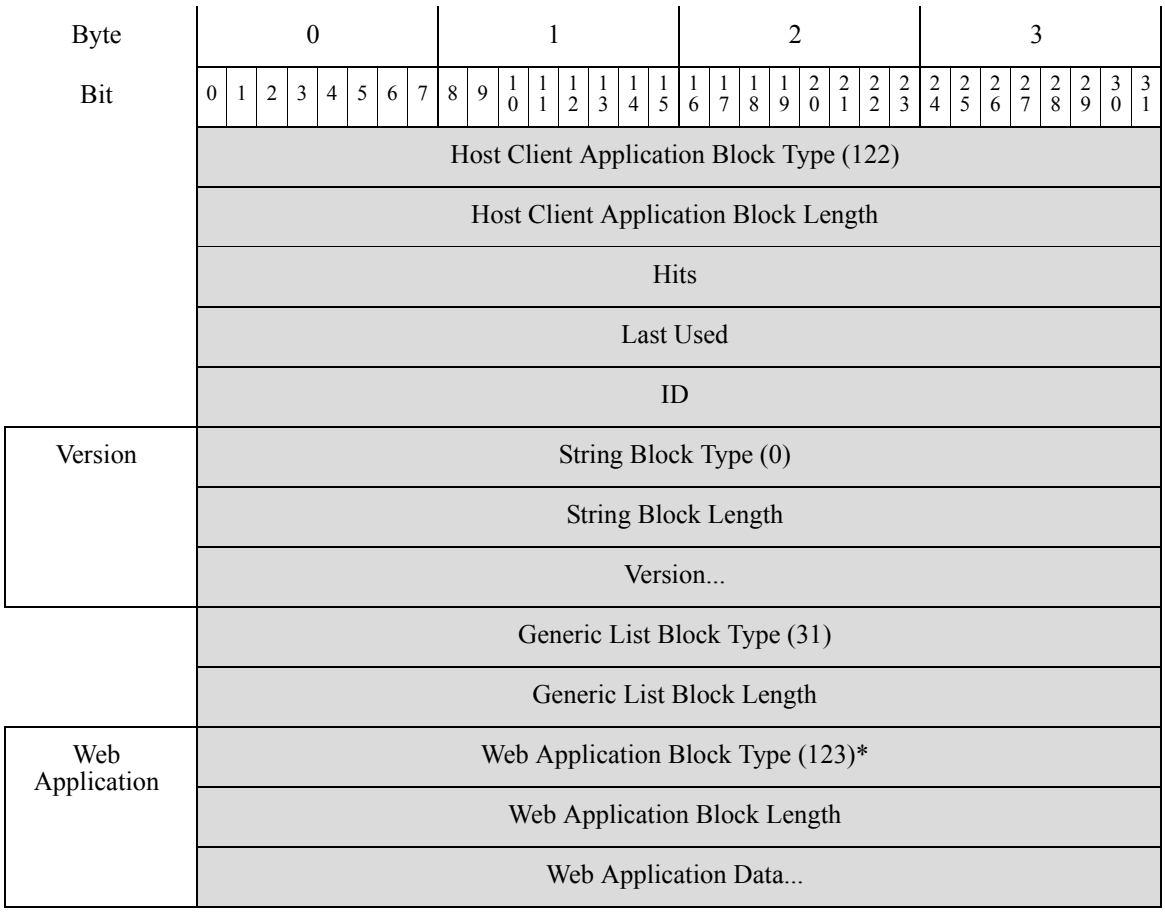
Table 4-75 Full Host Client Application Data Block 5.0+ Fields (continued)

Field	Data Type	Description
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
Application ID	uint32	Application ID of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application name, including eight bytes for the string block type and length, plus the number of bytes in the client application version.
Version	string	Client application version.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and the encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the generic list block length.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Vulnerability data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated Vulnerability data blocks.
Vulnerability Data Blocks	variable	Encapsulated Vulnerability data blocks up to the maximum number of bytes in the generic list block length.

Host Client Application Data Block for 5.0+

The Host Client Application data block for 5.0+ describes a client application and is used within New Client Application events (event type 1000, subtype 7), Client Application Timeout events (event type 1001, subtype 20), and Client Application Update events (event type 1001, subtype 32). The Host Client Application data block for 4.10.2+ has a block type of 122 in the series 1 group of blocks.

The following diagram shows the basic structure of a Host Client Application data block for 5.0+:



The following table describes the fields of the Host Client Application data block.

Table 4-76 Host Client Application Data Block Fields

Field	Data Type	Description
Client Application Block Type	uint32	Initiates a Host Client Application data block. This value is always `122`.
Client Application Block Length	uint32	Number of bytes in the Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.

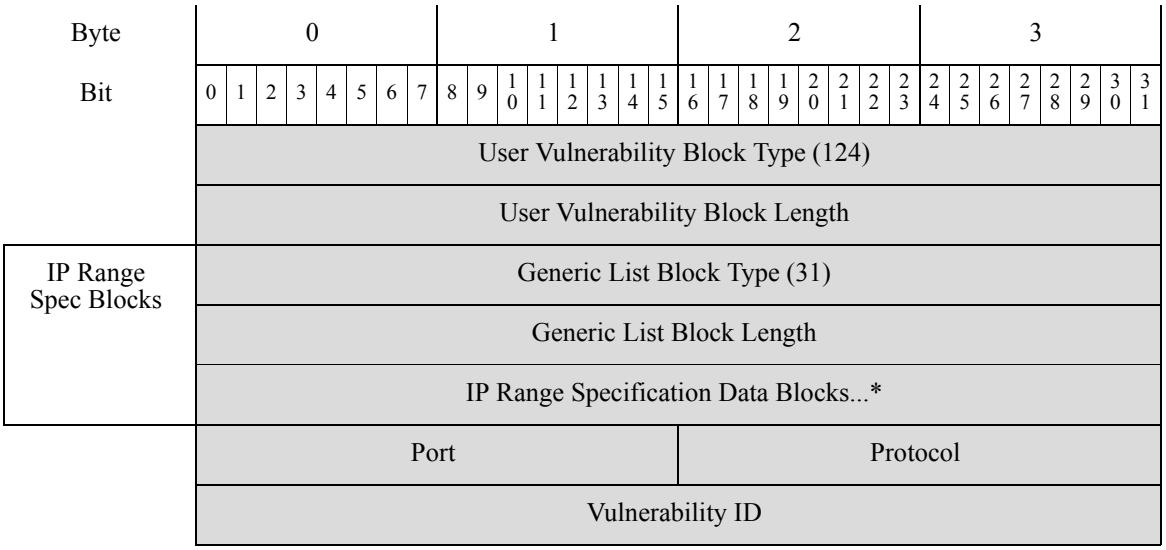
Table 4-76 Host Client Application Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the client application version.
Version	string	Client application version.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length. See Web Application Data Block for 5.0+, page 4-109 for information on the encapsulated data blocks (block type 123).

User Vulnerability Data Block 5.0+

The User Vulnerability data block describes a vulnerability and is used within User Vulnerability Change data blocks. These in turn are used in User Set Valid Vulnerabilities events and User Set Invalid Vulnerabilities events. The User Vulnerability data block for 5.0+ has a block type of 124 in the series 1 group of blocks. It supersedes block type 79. For more information on User Vulnerability Change data blocks, see User Vulnerability Change Data Block 4.7+, page 4-98.

The following diagram shows the format of a User Vulnerability data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
3rd Party Vuln UUID	Third-Party Vulnerability UUID																															
	UUID continued																															
	UUID continued																															
	UUID continued																															
	String Block Type (0)																															
	String Block Length																															
	Vulnerability String...																															
	Client Application ID																															
	Application Protocol ID																															
	String Block Type (0)																															
	String Block Length																															
	Version String...																															

The following table describes the fields of the User Vulnerability data block.

Table 4-77 User Vulnerability Data Block Fields

Field	Data Type	Description
User Vulnerability Block Type	uint32	Initiates a User Vulnerability data block. This value is always `124`.
User Vulnerability Block Length	uint32	Number of bytes in the User Vulnerability data block, including eight bytes for the user vulnerability block type and length fields, plus the number of bytes of user vulnerability data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP address ranges from user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Port	uint16	Port used by the server affected by the vulnerability. For client application vulnerabilities, the value is `0`.

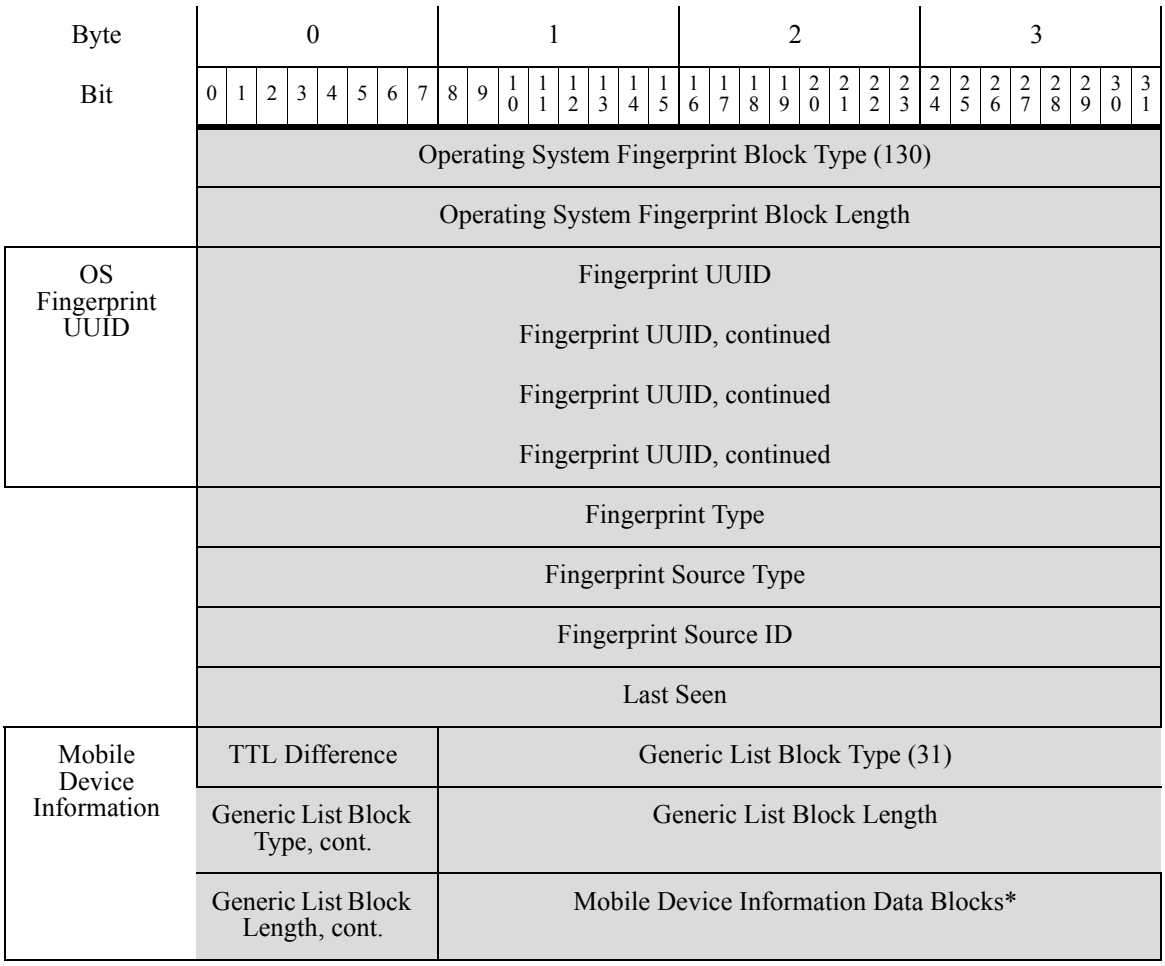
Table 4-77 User Vulnerability Data Block Fields (continued)

Field	Data Type	Description
Protocol	uint16	IANA protocol number or Ethertype for the protocol used by the server affected by the vulnerability. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP For client application vulnerabilities, the value is 0.
Vulnerability ID	uint32	The Cisco vulnerability ID.
Third-Party Vulnerability UUID	uint8 [16]	A unique ID number for the third-party vulnerability, if one exists. Otherwise, the value is 0.
String Block Type	uint32	Initiates a String data block for the vulnerability name. The value is always 0.
String Block Length	uint32	The number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Vulnerability Name	string	The vulnerability name.
Client Application ID	uint32	The application ID of the client application. For server vulnerabilities, the value is 0.
Application Protocol ID	uint32	The application ID of the application protocol used by client application. For server vulnerabilities, the value is 0.
String Block Type	uint32	Initiates a String data block for the version string. The value is always 0.
String Block Length	uint32	The number of bytes in the String data block for the version, including eight bytes for the string block type and length, plus the number of bytes in the client application version string.
Version	string	The client application version. For server vulnerabilities, the value is 0.

Operating System Fingerprint Data Block 5.1+

The Operating System Fingerprint data block has a block type of 130 in the series 1 group of blocks. The block includes a fingerprint Universally Unique Identifier (UUID), as well as the fingerprint type, the fingerprint source type, and the fingerprint source ID.

The following diagram shows the format of an Operating System Fingerprint data block in 5.1+.



Byte	0				1			
Bit	0 1 2 3 4 5 6 7				8 9 10 11 12 13 14 15			
	Operating System Fingerprint Block Type (130)							
	Operating System Fingerprint Block Length							
OS Fingerprint UUID	Fingerprint UUID							
	Fingerprint UUID, continued							
	Fingerprint UUID, continued							
	Fingerprint UUID, continued							
	Fingerprint Type							
	Fingerprint Source Type							
	Fingerprint Source ID							
	Last Seen							
Mobile Device Information	TTL Difference				Generic List Block Type (31)			
	Generic List Block Type, cont.				Generic List Block Length			
	Generic List Block Length, cont.				Mobile Device Information Data Blocks*			

Byte	2				3			
Bit	16 17 18 19 20 21 22 23				24 25 26 27 28 29 30 31			

The following table describes the fields of the operating system fingerprint data block.

Table 4-78 Operating System Fingerprint Data Block Fields

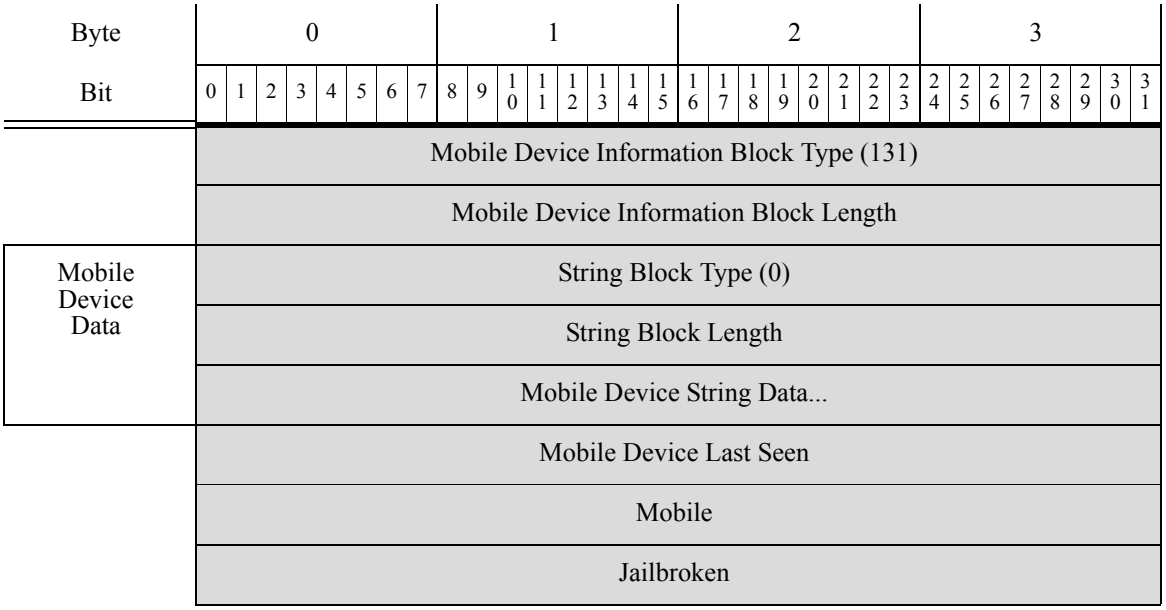
Field	Data Type	Description
Operating System Fingerprint Data Block Type	uint32	Initiates the operating system data block. This value is always `130`.
Operating System Data Block Length	uint32	Number of bytes in the Operating System Fingerprint data block, including eight bytes for the Operating System Fingerprint Data Block block type and length, plus the number of bytes in the Operating System Fingerprint data that follows.
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the vulnerability database (VDB).
Fingerprint Type	uint32	Indicates the type of fingerprint.
Fingerprint Source Type	uint32	Indicates the type (i.e., user or scanner) of the source that supplied the operating system fingerprint.

Table 4-78 Operating System Fingerprint Data Block Fields (continued)

Field	Data Type	Description
Fingerprint Source ID	uint32	Identification number that maps to the login name of the user that supplied the operating system fingerprint.
Last Seen	uint32	Indicates when the fingerprint was last seen in traffic.
TTL Difference	uint8	Indicates the difference between the TTL value in the fingerprint and the TTL value seen in the packet used to fingerprint the host.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Mobile Device Information Data Blocks	variable	Encapsulated Mobile Device Information data blocks up to the maximum number of bytes in the list block length. See Mobile Device Information Data Block for 5.1+, page 4-151 for a description of this data block.

Mobile Device Information Data Block for 5.1+

The following diagram shows the format of a Mobile Device Information data block. The data block contains the last time the host was detected, mobile device information, and whether the mobile device is jailbroken. The Mobile Device Information data block has a block type of 131 in the series 1 group of blocks.



The describes the fields of the Mobile Device Information data block returned by 5.1+.

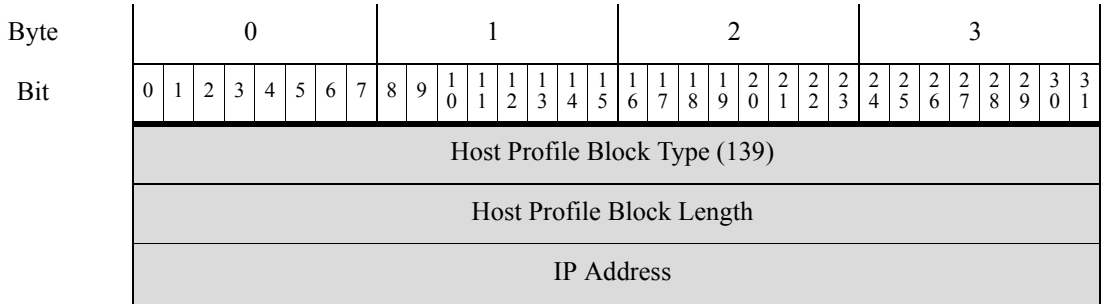
Table 4-79 Mobile Device Information Data Block 5.1+ Fields

Field	Data Type	Description
Mobile Device Information Block Type (131)	uint32	Initiates the operating system data block. This value is always `131`.
Mobile Device Information Block Length	uint32	Number of bytes in the Mobile Device Information data block, including eight bytes for the Mobile Device Information Data Block block type and length, plus the number of bytes in the Mobile Device Information data that follows.
String Block Type	uint32	Initiates a string data block for the mobile device string. This value is set to `0` to indicate string data.
String Block Length	uint32	Indicates the number of bytes in the mobile device string data block, including eight bytes for the string block type and length fields, plus the number of bytes in the mobile device string data that follows.
Mobile Device String Data	Variable	Contains the mobile device hardware information of the host detected.
Mobile Device Last Seen	uint32	Contains the time stamp the mobile device was last seen.
Mobile	uint32	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint32	True-false flag indicating whether the host is a mobile device that is jailbroken.

Host Profile Data Block for 5.2+

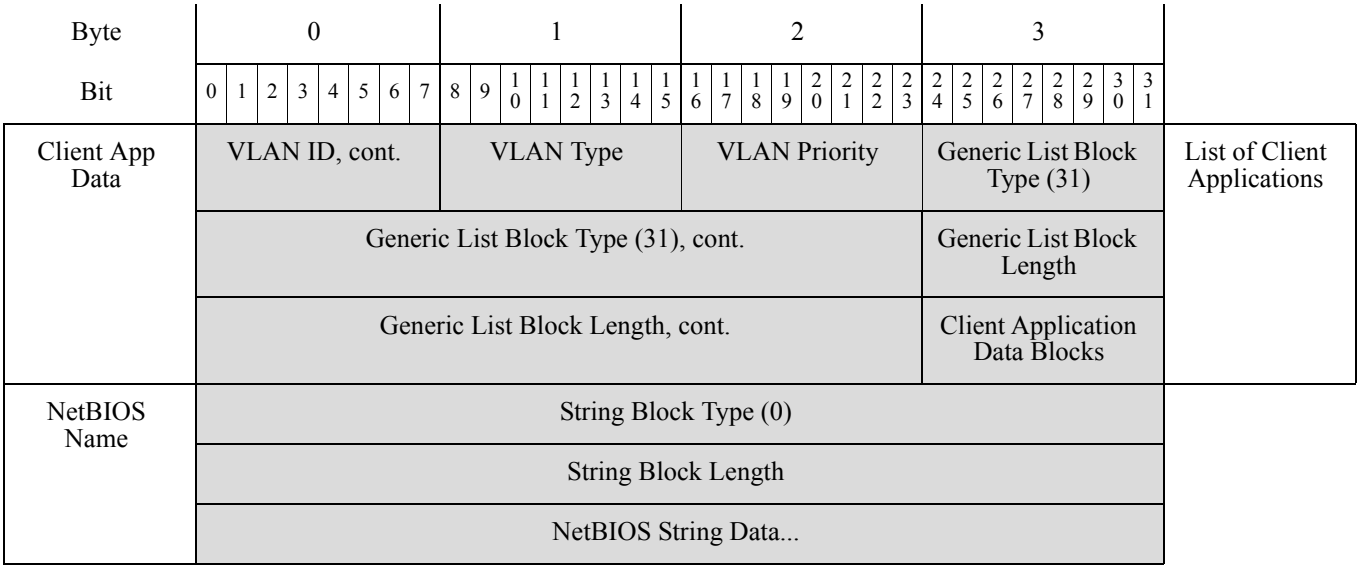
The following diagram shows the format of a Host Profile data block. The data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a data block can convey a NetBIOS name for the host. The Host Profile data block has a block type of 139 in the series 1 group of blocks. The data block now supports IPv6 addresses, and client application data blocks have been added.

Note An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	IP Address, continued																															
	IP Address, continued																															
	IP Address, continued																															
Server Fingerprints	Hops								Primary/Secondary								Generic List Block Type (31)															
	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Server Fingerprint Data Blocks*															
Client Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	Client Fingerprint Data Blocks*																															
SMB Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	SMB Fingerprint Data Blocks*																															
DHCP Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	DHCP Fingerprint Data Blocks*																															
Mobile Device Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	Mobile Device Fingerprint Data Blocks*																															
IPv6 Sever Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	Ipv6 Server Fingerprint Data Blocks*																															
IPv6 Client Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	IPv6 Client Fingerprint Data Blocks*																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IPv6 DHCP Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	IPv6 DHCP Fingerprint Data Blocks*																															
User Agent Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	User Agent Fingerprint Data Blocks*																															
TCP Server Block*	List Block Type (11)																															List of TCP Servers
	List Block Length																															
	TCP Server Data Blocks																															
UDP Server Block*	List Block Type (11)																															List of UDP Servers
	List Block Length																															
	UDP Server Data Blocks																															
Network Protocol Block*	List Block Type (11)																															List of Network Protocols
	List Block Length																															
	Network Protocol Data Blocks																															
Transport Protocol Block*	List Block Type (11)																															List of Transport Protocols
	List Block Length																															
	Transport Protocol Data Blocks																															
MAC Address Block*	List Block Type (11)																															List of MAC Addresses
	List Block Length																															
	Host MAC Address Data Blocks																															
	Host Last Seen																															
	Host Type																															
	Mobile								Jailbroken								VLAN Presence								VLAN ID							



The following table describes the fields of the host profile data block returned by 5.2+.

Table 4-80 Host Profile Data Block 5.2+ Fields

Field	Data Type	Description
Host Profile Block Type	uint32	Initiates the Host Profile data block for 5.2+. This value is always `139`.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8(16)	IP Address of the host. This can be IPv4 or IPv6.
Hops	uint8	Number of hops from the host to the device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • `0` — Host is in the primary network. • `1` — Host is in the secondary network.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.

Table 4-80 Host Profile Data Block 5.2+ Fields (continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a mobile device fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table 4-80 Host Profile Data Block 5.2+ Fields (continued)

Field	Data Type	Description
Operating System Fingerprint Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a mobile device fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Client) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table 4-80 Host Profile Data Block 5.2+ Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (User Agent Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
TCP Server Data Blocks	variable	Host server data blocks describing a TCP server. See Host Server Data Block 4.10.0+, page 4-128 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
UDP Server Data Blocks	uint32	Host server data blocks describing a UDP server. See Host Server Data Block 4.10.0+, page 4-128 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Network Protocol Data Blocks	uint32	Protocol data blocks describing a network protocol. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Transport Protocol Data Blocks	uint32	Protocol data blocks describing a transport protocol. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.

Table 4-80 Host Profile Data Block 5.2+ Fields (continued)

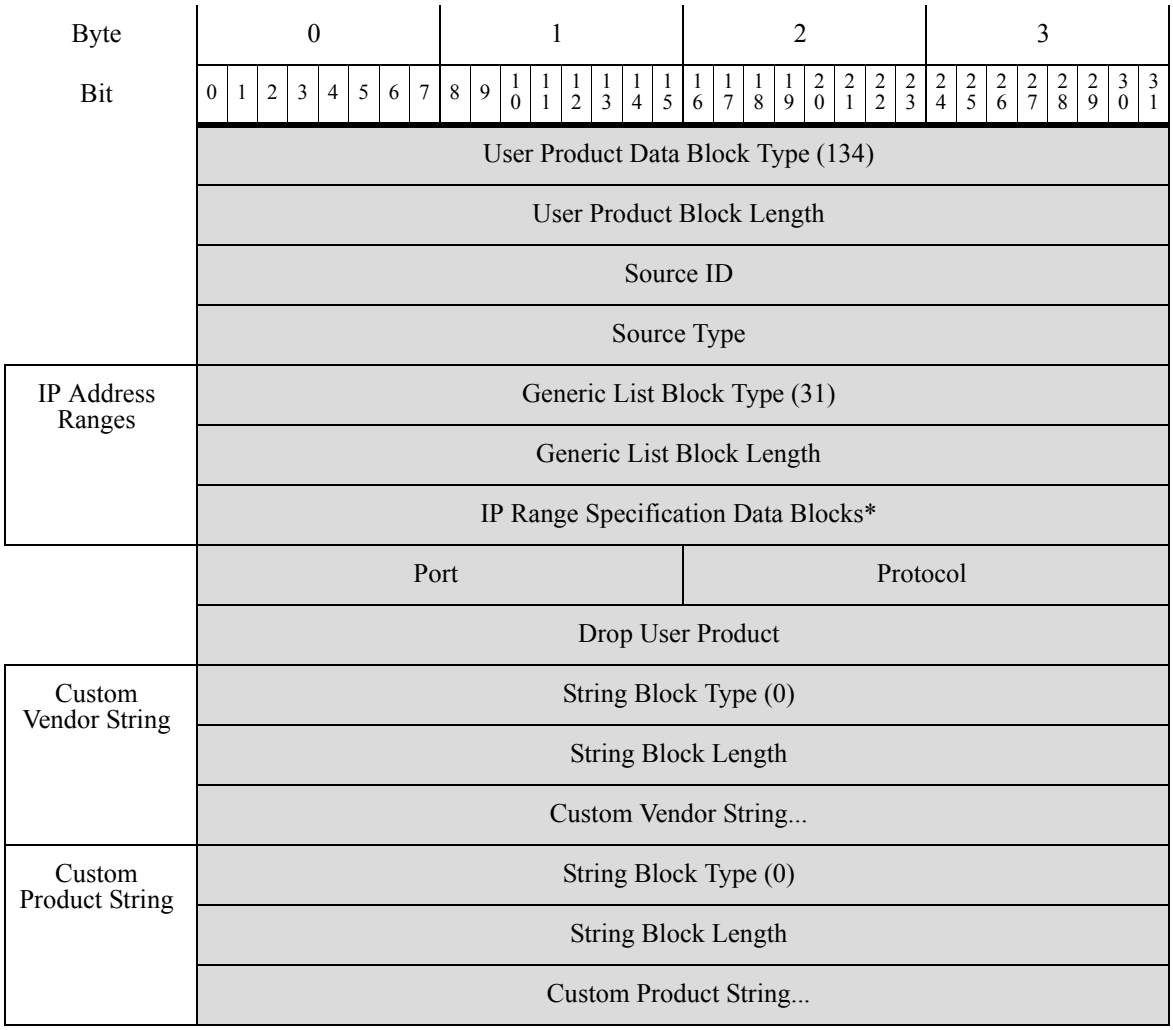
Field	Data Type	Description
Host MAC Address Data Blocks	uint32	Host MAC Address data blocks describing a host MAC address. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — Host • 1 — Router • 2 — Bridge • 3 — NAT device • 4 — LB (load balancer)
Mobile	uint8	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint8	True-false flag indicating whether the host is a mobile device that is also jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
String Block Type	uint32	Initiates a String data block for the host client application data. This value is always 112.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the host client application data.
Host Client Application Data Blocks	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.

User Product Data Block 5.1+

The User Product data block conveys host input data imported from a third-party application, including third-party application string mappings. This data block is used in The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116 and User Server and Operating System Messages, page 4-51. The User Product data block has a block type of 65 in the series 1 group of blocks for versions up to 4.7-4.10.1, a block type of 118 for 4.10.2-5.0.x, and a block type of 134 in the series 1 group of blocks for 5.1+. Block types 65 and 118 have the same structure.

Note An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the User Product data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Custom Version String	String Block Type (0)																															
	String Block Length																															
	Custom Version String...																															
	Software ID																															
	Server ID																															
	Vendor ID																															
	Product ID																															
Major Version String	String Block Type (0)																															
	String Block Length																															
	Major Version String...																															
Minor Version String	String Block Type (0)																															
	String Block Length																															
	Minor Version String...																															
Revision String	String Block Type (0)																															
	String Block Length																															
	Revision String...																															
To Major String	String Block Type (0)																															
	String Block Length																															
	To Major Version String...																															
To Minor String	String Block Type (0)																															
	String Block Length																															
	To Minor Version String...																															
To Revision String	String Block Type (0)																															
	String Block Length																															
	To Revision String...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Build String	String Block Type (0)																															
	String Block Length																															
	Build String...																															
Patch String	String Block Type (0)																															
	String Block Length																															
	Patch String...																															
Extension String	String Block Type (0)																															
	String Block Length																															
	Extension String...																															
OS UUID	Operating System UUID																															
	Operating System UUID cont.																															
	Operating System UUID cont.																															
	Operating System UUID cont.																															
Device String	String Block Type (0)																															
	String Block Length																															
	Device String...																															
List of Fixes	Mobile								Jailbroken								Generic List Block Type (31)															
	Generic List Block Type (31) cont.																Generic List Block Length															
	Generic List Block Length cont.																Fix List Data Blocks*															
	Fix List Data Blocks* cont.																															

The following table describes the components of the User Product data block.

Table 4-81 User Product Data Block Fields

Field	Data Type	Description
User Product Data Block Type	uint32	Initiates a User Product data block. This value is `134` for 5.1+.
User Product Block Length	uint32	Total number of bytes in the User Product data block, including eight bytes for the user product block type and length fields, plus the number of bytes in the user product data that follows.
Source ID	uint32	Identification number that maps to the source that imported the data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the data was provided by RNA • `1` if the data was provided by a user • `2` if the data was provided by a third-party scanner • `3` if the data was provided by a command line tool such as `nmimport.pl` or the Host Input API client
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Port	uint16	Port specified by the user.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • `6` — TCP • `17` — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • `2048` — IP
Drop User Product	uint32	Indicates whether the user OS definition was deleted from the host: • `0` — No • `1` — Yes
String Block Type	uint32	Initiates a String data block containing the custom vendor name specified in the user input. This value is always `0`.
String Block Length	uint32	Number of bytes in the custom vendor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.

Table 4-81 User Product Data Block Fields (continued)

Field	Data Type	Description
Custom Vendor Name	string	The custom vendor name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom product name specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom product String data block, including eight bytes for the block type and length fields, plus the number of bytes in the product name.
Custom Product Name	string	The custom product name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom version specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Custom Version	string	The custom version specified in the user input.
Software ID	uint32	The identifier for a specific revision of a server or operating system in the database.
Server ID	uint32	The Firepower System application identifier for the application protocol on the host server specified in user input.
Vendor ID	uint32	The identifier for the vendor of a third-party operating system specified when the third-party operating system is mapped to a Firepower System OS definition.
Product ID	uint32	The product identification string of a third-party operating system string specified when the third-party operating system string is mapped to a Firepower System OS definition.
String Block Type	uint32	Initiates a String data block containing the major version number of the Firepower System operating system definition that a third-party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Major Version	string	Major version of the Firepower System operating system definition that a third-party OS string is mapped to.
String Block Type	uint32	Initiates a String data block containing the minor version number of the Firepower System operating system definition that a third-party OS string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Minor Version	string	Minor version number of the Firepower System operating system definition that a third-party OS string in the user input is mapped to.

Table 4-81 User Product Data Block Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the revision number of the Firepower System operating system definition that a third-party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
Revision	string	Revision number of the Firepower System operating system definition that a third-party OS string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last major version of the Firepower System operating system definition that a third-party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Major	string	Last version number in a range of major version numbers of the Firepower System operating system definition that a third-party OS string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last minor version of the Firepower System operating system definition that a third-party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Minor	string	Last version number in a range of minor version numbers of the Firepower System operating system definition that a third-party OS string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the Last revision number of the Firepower System operating system definition that a third-party OS string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
To Revision	string	Last revision number in a range of revision numbers of the Firepower System operating system definitions that a third-party OS string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the build number of the Firepower System operating system that the third-party OS string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Build	string	Build number of the Firepower System operating system that the third-party OS string in the user input is mapped to.

Table 4-81 User Product Data Block Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the patch number of the Firepower System operating system that the third-party OS string is mapped to. This value is always `0`.
String Block Length	uint32	Number of bytes in the patch String data block, including eight bytes for the block type and length fields, plus the number of bytes in the patch number.
Patch	string	Patch number of the Firepower System operating system that the third-party OS string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the extension number of the Firepower System OS that the third-party operating system string is mapped. This value is always `0`.
String Block Length	uint32	Number of bytes in the extension String data block, including eight bytes for the block type and length fields, plus the number of bytes in the extension number.
Extension	string	Extension number of the Firepower System operating system that the third-party OS string in the user input is mapped to.
UUID	uint8 [x16]	Contains the unique identification number for the operating system.
String Block Type	uint32	Initiates a String data block containing the device hardware information in the user input. This value is always `0`.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Device String	string	Mobile device hardware information.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Fix List data blocks conveying user input data regarding what fixes have been applied to hosts in the specified IP address ranges. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Fix List data blocks.
Fix List Data Blocks *	variable	Fix List data blocks containing information about fixes applied to the hosts. See Fix List Data Block, page 4-94 for a description of this data block.

User Data Blocks

User data blocks appear in user event messages. They are a subset of the series 1 data blocks. For information on the general format of series 1 data blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.

Note The data block length field of the user data block header contains the number of bytes in the data block, including the eight bytes of the two data block header fields.

The following table lists the user data blocks that can appear in user event messages. Data blocks are listed by data block type. Current data blocks are the latest versions. Legacy blocks are supported but not produced by the current version of the Firepower System.

Table 4-82 User Data Block Type

Type	Content	Data Block Category	Description
73	User Login Information	Legacy	Contains changes in login information for users detected by the system. See User Login Information Data Block 6.0+, page 4-179 for more information. The successor block type introduced for version 5.0 has the same structure as block type 73 but with different data in the fields.
74	User Account Update Message	Current	Contains changes in user account information. See User Account Update Message Data Block, page 4-168 for more information.
75	User Information for 4.7 - 4.10.x	Legacy	Contains changes in information for users detected by the system. See User Information Data Block for 6.0+, page 4-177 for more information. The successor block introduced for version 6.0 has block type 158.
120	User Information for 5.x	Current	Contains changes in information for users detected by the system. See User Information Data Block for 6.0+, page 4-177 for more information. Supersedes block type 75. It is superseded by block type 158.
121	User Login Information	Legacy	Contains changes in login information for users detected by the system. See User Login Information Data Block for 5.0 - 5.0.2, page B-100 for more information. Differs from block 73 in the content of the Protocol field, which stores the Version 5.0+ application ID for the application protocol ID detected in the event. The successor block introduced for version 5.1 has block type 127.
127	User Login Information	Legacy	Contains changes in login information for users detected by the system. See User Login Information Data Block 5.1-5.4.x, page B-102 for more information. It supersedes block type 121. The successor block introduced for 6.0 has block type 159.
150	IOC State	Current	Contains information about compromises. See IOC State Data Block for 5.3+, page 4-28 for more information.

Table 4-82 User Data Block Type (continued)

Type	Content	Data Block Category	Description
158	User Information for 6.0+	Current	Contains changes in information for users detected by the system. See User Information Data Block for 6.0+, page 4-177 for more information. Supersedes block type 120.
159	User Login Information	Current	Contains changes in login information for users detected by the system. See User Login Information Data Block 6.0+, page 4-179 for more information. It supersedes block type 127.

User Account Update Message Data Block

The User Account Update Message data block conveys information about updates to a user's account information.

The User Account Update Message data block has a block type of 74 in the series 1 group of blocks.

The following diagram shows the format of the User Account Update Message data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Account Update Message Block Type (74)																															
	User Account Update Message Block Length																															
User Name	String Block Type (0)																															
	String Block Length																															
	User Name...																															
First Name	String Block Type (0)																															
	String Block Length																															
	First Name...																															
Middle Initials	String Block Type (0)																															
	String Block Length																															
	Middle Initials...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Last Name	String Block Type (0)																															
	String Block Length																															
	Last Name...																															
Full Name	String Block Type (0)																															
	String Block Length																															
	Full Name...																															
Title	String Block Type (0)																															
	String Block Length																															
	Title...																															
Staff Identity	String Block Type (0)																															
	String Block Length																															
	Staff Identity...																															
Address	String Block Type (0)																															
	String Block Length																															
	Address...																															
City	String Block Type (0)																															
	String Block Length																															
	City...																															
State	String Block Type (0)																															
	String Block Length																															
	State...																															
Country/ Region	String Block Type (0)																															
	String Block Length																															
	Country/Region...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Postal Code	String Block Type (0)																															
	String Block Length																															
	Postal Code...																															
Building	String Block Type (0)																															
	String Block Length																															
	Building...																															
Location	String Block Type (0)																															
	String Block Length																															
	Location...																															
Room	String Block Type (0)																															
	String Block Length																															
	Room...																															
Company	String Block Type (0)																															
	String Block Length																															
	Company...																															
Division	String Block Type (0)																															
	String Block Length																															
	Division...																															
Dept	String Block Type (0)																															
	String Block Length																															
	Department...																															
Office	String Block Type (0)																															
	String Block Length																															
	Office...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Mailstop	String Block Type (0)																															
	String Block Length																															
	Mailstop...																															
Email	String Block Type (0)																															
	String Block Length																															
	Email...																															
Phone	String Block Type (0)																															
	String Block Length																															
	Phone...																															
IP Phone	String Block Type (0)																															
	String Block Length																															
	IP Phone...																															
User 1	String Block Type (0)																															
	String Block Length																															
	User 1...																															
User 2	String Block Type (0)																															
	String Block Length																															
	User 2...																															
User 3	String Block Type (0)																															
	String Block Length																															
	User 3...																															
User 4	String Block Type (0)																															
	String Block Length																															
	User 4...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Email Alias 1	String Block Type (0)																															
	String Block Length																															
	Email Alias 1...																															
Email Alias 2	String Block Type (0)																															
	String Block Length																															
	Email Alias 2...																															
Email Alias 3	String Block Type (0)																															
	String Block Length																															
	Email Alias 3...																															

The following table describes the components of the User Account Update Message data block.

Table 4-83 User Account Update Message Data Block Fields

Field	Data Type	Description
User Account Update Message Block Type	uint32	Initiates a User Account Update Message data block. This value is always 74.
User Account Update Message Block Length	uint32	Total number of bytes in the User Account Update Message data block, including eight bytes for the user account update message block type and length fields, plus the number of bytes in the user account update message data that follows.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The username for the user.
String Block Type	uint32	Initiates a String data block containing the first name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the first name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the first name.
First Name	string	The first name for the user.
String Block Type	uint32	Initiates a String data block containing the middle initials for the user. This value is always 0.

Table 4-83 User Account Update Message Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the middle initials String data block, including eight bytes for the block type and length fields, plus the number of bytes in the middle initials.
Middle Initials	string	The middle initials for the user.
String Block Type	uint32	Initiates a String data block containing the last name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the last name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the last name.
Last Name	string	The last name for the user.
String Block Type	uint32	Initiates a String data block containing the full name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the full name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the full name.
Full Name	string	The full name for the user.
String Block Type	uint32	Initiates a String data block containing the title for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the title String data block, including eight bytes for the block type and length fields, plus the number of bytes in the title.
Title	string	The title for the user.
String Block Type	uint32	Initiates a String data block containing the staff identification for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the staff identity String data block, including eight bytes for the block type and length fields, plus the number of bytes in the staff identity.
Staff Identity	string	The staff identity for the user.
String Block Type	uint32	Initiates a String data block containing the address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the address.
Address	string	The address for the user.
String Block Type	uint32	Initiates a String data block containing the city from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the city String data block, including eight bytes for the block type and length fields, plus the number of bytes in the city.
City	string	The city from the user's address.
String Block Type	uint32	Initiates a String data block containing the state from the user's address. This value is always 0.

Table 4-83 User Account Update Message Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the state String data block, including eight bytes for the block type and length fields, plus the number of bytes in the state.
State	string	The state for the user.
String Block Type	uint32	Initiates a String data block containing the country or region from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the country or region String data block, including eight bytes for the block type and length fields, plus the number of bytes in the country or region.
Country or Region	string	The country or region from the user's address.
String Block Type	uint32	Initiates a String data block containing the postal code from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the postal code String data block, including eight bytes for the block type and length fields, plus the number of bytes in the postal code.
Postal Code	string	The postal code from the user's address.
String Block Type	uint32	Initiates a String data block containing the building from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the building String data block, including eight bytes for the block type and length fields, plus the number of bytes in the building name.
Building	string	The building from the user's address.
String Block Type	uint32	Initiates a String data block containing the location from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the location String data block, including eight bytes for the block type and length fields, plus the number of bytes in the location name.
Location	string	The location from the user's address.
String Block Type	uint32	Initiates a String data block containing the room from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the room String data block, including eight bytes for the block type and length fields, plus the number of bytes in the room.
Room	string	The room from the user's address.
String Block Type	uint32	Initiates a String data block containing the company from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the company String data block, including eight bytes for the block type and length fields, plus the number of bytes in the company name.
Company	string	The company from the user's address.

Table 4-83 User Account Update Message Data Block Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the division from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the division String data block, including eight bytes for the block type and length fields, plus the number of bytes in the division name.
Division	string	The division from the user's address.
String Block Type	uint32	Initiates a String data block containing the department from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the department String data block, including eight bytes for the block type and length fields, plus the number of bytes in the department.
Department	string	The department from the user's address.
String Block Type	uint32	Initiates a String data block containing the office from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the office String data block, including eight bytes for the block type and length fields, plus the number of bytes in the office.
Office	string	The office from the user's address.
String Block Type	uint32	Initiates a String data block containing the mailstop from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the mailstop String data block, including eight bytes for the block type and length fields, plus the number of bytes in the mailstop.
Mailstop	string	The mailstop from the user's address.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
String Block Type	uint32	Initiates a String data block containing the phone number for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the phone number.
Phone	string	The phone number for the user.
String Block Type	uint32	Initiates a String data block containing the Internet phone number for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the Internet phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the Internet phone number.
Internet Phone	string	The Internet phone number for the user.

Table 4-83 User Account Update Message Data Block Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 1	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 2	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 3	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 4	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 1	string	An email alias for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 2	string	An email alias for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 3	string	An email alias for the user.

User Information Data Block for 6.0+

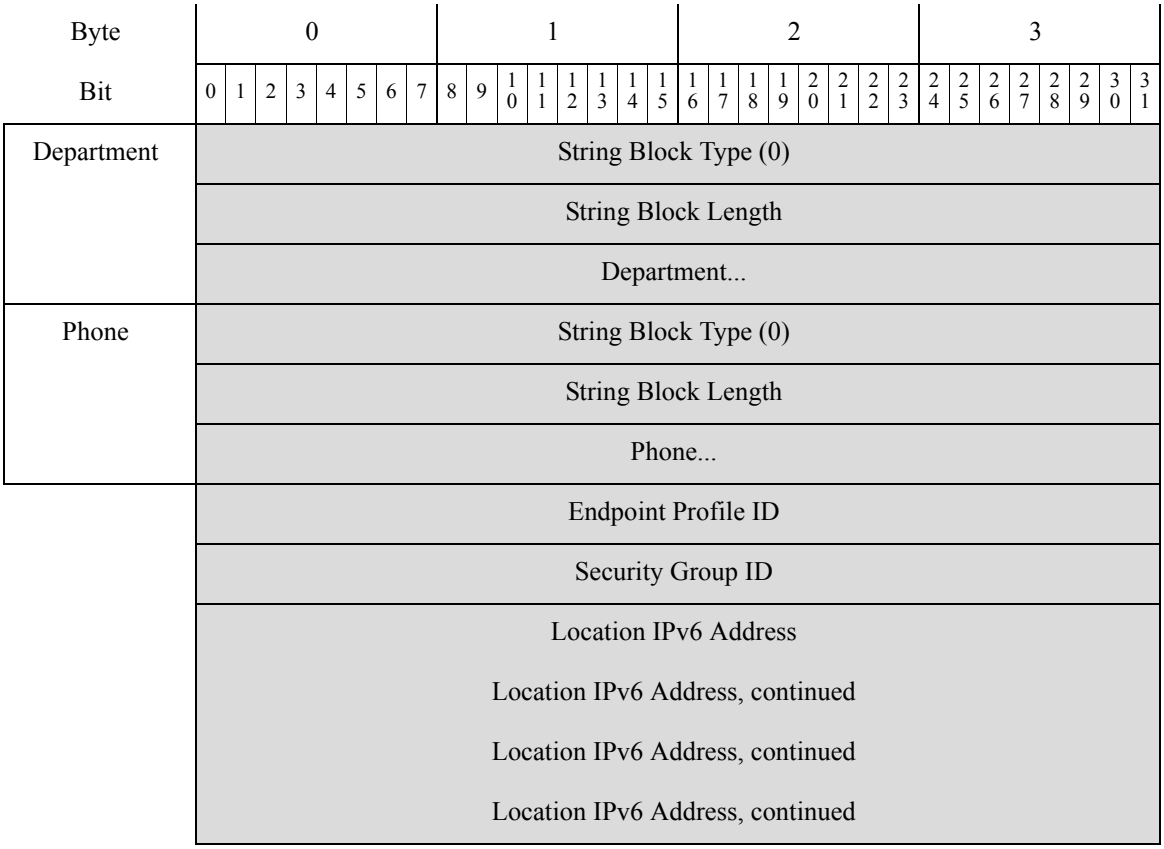
The User Information data block is used in User Modification messages and conveys information for a user detected, removed, or dropped. For more information, see User Modification Messages, page 4-54

The User Information data block has a block type of 158 in the series 1 group of blocks for version 6.0+. It has new endpoint profile, Security Intelligence, and IPv6 fields.

The User Information data block has a block type of 75 in the series 1 group of blocks for version 4.7 - 4.10.x and a block type of 120 in the series 1 group of blocks for 5.x. See User Information Data Block for 5.x, page B-104 for more information.

The following diagram shows the format of the User Information data block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Information Block Type (75 | 120)																															
	User Information Block Length																															
	User ID																															
User Name	String Block Type (0)																															
	String Block Length																															
	User Name...																															
	Realm ID																															
	Protocol																															
First Name	String Block Type (0)																															
	String Block Length																															
	First Name...																															
Last Name	String Block Type (0)																															
	String Block Length																															
	Last Name...																															
Email	String Block Type (0)																															
	String Block Length																															
	Email...																															



The following table describes the components of the User Information data block.

Table 4-84 User Information Data Block Fields

Field	Data Type	Description
User Information Block Type	uint32	Initiates a User Information data block. This value is `158`.
User Information Block Length	uint32	Total number of bytes in the User Information data block, including eight bytes for the user information block type and length fields plus the number of bytes in the user information data that follows.
User ID	uint32	Identification number of the user.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields plus the number of bytes in the username.
Username	string	The username for the user.
Realm ID	uint32	Integer ID which corresponds to an identity realm.
Protocol	uint32	The protocol for the packet containing the user information.
String Block Type	uint32	Initiates a String data block containing the first name of the user. This value is always `0`.

Table 4-84 User Information Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the first name String data block, including eight bytes for the block type and length fields plus the number of bytes in the first name.
First Name	string	The first name for the user.
String Block Type	uint32	Initiates a String data block containing the last name for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the user last name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the last name.
Last Name	string	The last name for the user.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
String Block Type	uint32	Initiates a String data block containing the department for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the department String data block, including eight bytes for the block type and length fields, plus the number of bytes in the department.
Department	string	The department for the user.
String Block Type	uint32	Initiates a String data block containing the phone number for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the phone number.
Phone	string	The phone number for the user.
Endpoint Profile ID	uint32	ID number of the type of device used by the connection endpoint. This is unique for each defense center and is resolved in metadata.
Security Group ID	uint32	ID number of the network traffic group.
Location IPv6 Address	uint16[8]	IP address of the interface communicating with ISE. Can be IPv4 or IPv6.

User Login Information Data Block 6.0+

The User Login Information data block is used in User Information Update messages and conveys changes in login information for a detected user. For more information, see User Information Update Message Block, page 4-55.

he User Login Information data block has a block type of 159 for version 6.0+. It has new ISE integration endpoint profile, Security Intelligence fields.

The User Login Information data block has a block type of 73 for version 4.7 - 4.10.x, a block type of 121 in the series 1 group of blocks for version 5.0 - 5.0.2, and a block type of 127 in the series 1 group of blocks for version 5.1+. See User Login Information Data Block 5.1-5.4.x, page B-102 for more information.

The graphic below shows the format of the User Login Information data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Login Information Block Type (159)																															
	User Login Information Block Length																															
	Timestamp																															
	IPv4 Address																															
User Name	String Block Type (0)																															
	String Block Length																															
	User Name...																															
Domain	String Block Type (0)																															
	String Block Length																															
	Domain...																															
	User ID																															
	Realm ID																															
	Endpoint Profile ID																															
	Security Group ID																															
	Application ID																															
	Protocol																															
Email	String Block Type (0)																															
	String Block Length																															
	Email...																															
	IPv6 Address																															
	IPv6 Address, continued																															
	IPv6 Address, continued																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	IPv6 Address, continued																															
	Location IPv6 Address Location IPv6 Address, continued Location IPv6 Address, continued Location IPv6 Address, continued																															
Reported By	Login Type								Auth. Type								String Block Type (0)															
	String Block Type (0), cont.																String Block Length															
	String Block Length, cont.																Reported By...															

The following table describes the components of the User Login Information data block.

Table 4-85 User Login Information Data Block Fields

Field	Data Type	Description
User Login Information Block Type	uint32	Initiates a User Login Information data block. This value is `159` for version 6.0+.
User Login Information Block Length	uint32	Total number of bytes in the User Login Information data block, including eight bytes for the user login information block type and length fields, plus the number of bytes in the user login information data that follows.
Timestamp	uint32	Timestamp of the event.
IPv4 Address	uint32	This field is reserved but no longer populated. The IPv4 address is stored in the IPv6 Address field. See IP Addresses, page 1-6 for more information.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The user name for the user.
String Block Type	uint32	Initiates a String data block containing the domain. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the domain.
Domain	string	Domain in which the user logged in.

Table 4-85 User Login Information Data Block Fields (continued)

Field	Data Type	Description
User ID	uint32	Identification number of the user.
Realm ID	uint32	Integer ID which corresponds to an identity realm.
Endpoint Profile ID	uint32	ID number of the type of device used by the connection endpoint. This is unique for each DC and resolved in metadata.
Security Group ID	uint32	ID number of the network traffic group.
Application ID	uint32	The application ID for the application protocol used in the connection that the login information was derived from.
Protocol	uint32	Protocol used to detect or report the user. Possible values are: • 165 - FTP • 426 - SIP • 547 - AOL Instant Messenger • 683 - IMAP • 710 - LDAP • 767 - NTP • 773 - Oracle Database • 788 - POP3 • 1755 - MDNS
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
IPv6 Address	uint8[16]	IPv6 address from the host where the user was detected logging in, in IP address octets.
Location IPv6 Address	uint8[16]	Most recent IP address on which the user logged in. Can be either an IPv4 or IPv6 address.
Login Type	uint8	The type of user login detected.
Authentication Type	uint8	Type of authentication used by the user. Values may be: • 0 - no authorization required • 1 - passive authentication, AD agent, or ISE session • 2 - captive portal successful authentication • 3 - captive portal guest authentication • 4 - captive portal failed authentication
String Block Type	uint32	Initiates a String data block containing the Reported By value. This value is always 0.

Table 4-85 User Login Information Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the Reported By String data block, including eight bytes for the block type and length fields, plus the number of bytes in the Reported By field.
Reported By	string	The name of the Active Directory server reporting a login.

Discovery and Connection Event Series 2 Data Blocks

In the following table, the Data Block Status field indicates whether the block is current (the latest version) or legacy (used in an older version and can still be requested through eStreamer).

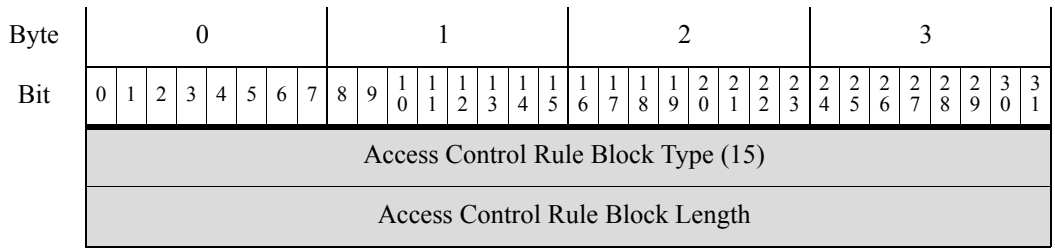
Table 4-86 Discovery and Connection Event Series 2 Block Types

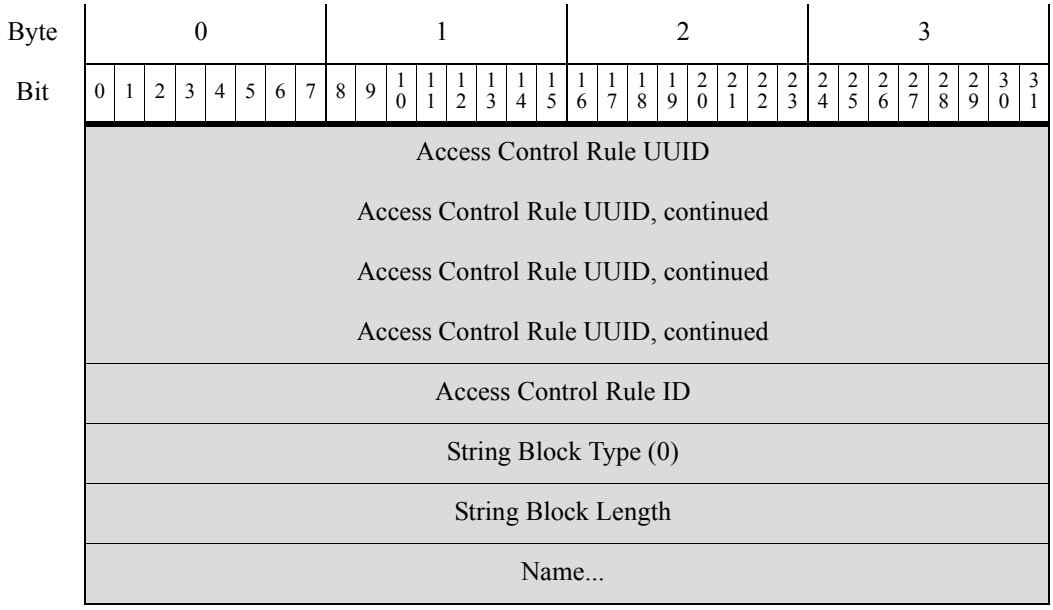
Type	Content	Data Block Status	Description
15	Access Control Rule	Current	Used by access control rule metadata messages to map policy UUID and rule ID values to a descriptive string. See Access Control Rule Data Block, page 4-183.
21	Access Control Rule Reason	Current	Used by access control rule metadata messages to map access control rule reasons to a descriptive string. See Access Control Rule Reason Data Block 5.1+, page 4-184.
22	Security Intelligence Category	Current	Used to store Security Intelligence information. See Security Intelligence Category Data Block 5.1+, page 4-185.
57	User Data	Current	Used by the User Record metadata messages to provide the user ID number, protocol on which the user was detected, and the user name. See User Data Block, page 4-187.

Access Control Rule Data Block

The eStreamer service uses the Access Control Rule data block in access control rule metadata messages to map policy UUID and rule ID combinations to a descriptive string. The Access Control Rule data block has a block type of 15 in the series 2 group of blocks.

The following graphic shows the structure of the Access Control Rule data block:





The following table describes the fields in the Access Control Rule data block.

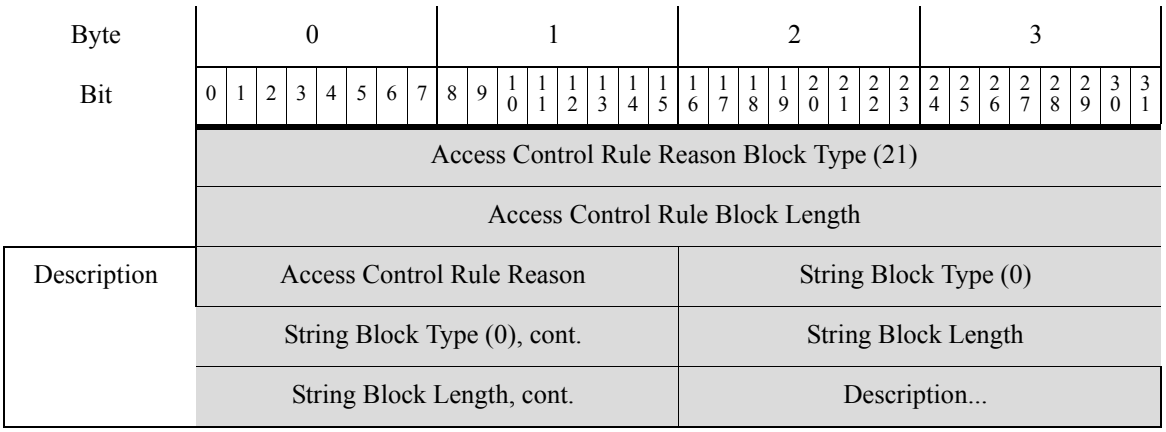
Table 4-87 Access Control Rule Data Block Fields

Field	Data Type	Description
Access Control Rule Block Type	uint32	Initiates an Access Control Rule block. This value is always `15`.
Access Control Rule Block Length	uint32	Total number of bytes in the Access Control Rule block, including eight bytes for the Access Control Rule block type and length fields, plus the number of bytes of data that follows.
Access Control Rule UUID	uint8[16]	The unique identifier for the access control rule.
Access Control Rule ID	uint32	The internal Cisco identifier for the access control rule.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule UUID and access control rule ID. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name.

Access Control Rule Reason Data Block 5.1+

The eStreamer service uses the Access Control Rule Reason data block in Access Control Rule Reason metadata messages to map Access Control reasons to a descriptive string. The Access Control Rule Reason data block has a block type of 21 in the series 2 group of blocks.

The following graphic shows the structure of the Access Control Rule Reason data block:



The following table describes the fields in the Access Control Rule Reason data block.

Table 4-88 Access Control Rule Reason Data Block Fields

Field	Data Type	Description
Access Control Rule Reason Block Type	uint32	Initiates an Access Control Rule Reason block. This value is always `21`.
Access Control Rule Reason Block Length	uint32	Total number of bytes in the Access Control Rule Reason block, including eight bytes for the Access Control Rule Reason block type and length fields, plus the number of bytes of data that follows.
Access Control Rule Reason	uint16	The reason the Access Control rule logged the connection.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the Access Control rule reason.

Security Intelligence Category Data Block 5.1+

The eStreamer service uses the Security Intelligence Category data block in access control rule metadata messages to stream Security Intelligence information. The Security Intelligence Category data block has a block type of 22 in the series 2 group of blocks.

The following graphic shows the structure of the Security Intelligence Category data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Security Intelligence Category Block Type (22)																															
	Security Intelligence Category Block Length																															
	Security Intelligence List ID																															
AC Policy UUID	Access Control Policy UUID																															
	Access Control Policy UUID, continued																															
	Access Control Policy UUID, continued																															
	Access Control Policy UUID, continued																															
Rule Name	String Block Type (0)																															
	String Block Length																															
	Security Intelligence List Name...																															

The following table describes the fields in the Security Intelligence Category data block:

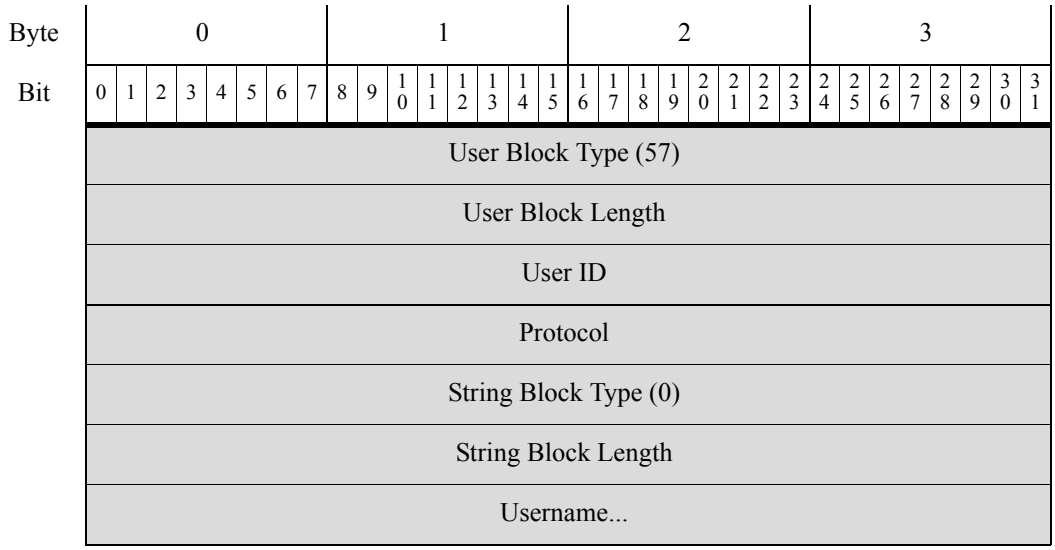
Table 4-89 Security Intelligence Category Data Block fields

Field	Data Type	Description
Security Intelligence Category Block Type	uint32	Initiates an Security Intelligence Category data block. This value is always `22`.
Security Intelligence Category Block Length	uint32	Total number of bytes in the Security Intelligence Category block, including eight bytes for the Security Intelligence Category block type and length fields, plus the number of bytes of data that follows.
Security Intelligence List ID	uint32	The ID of the IP blacklist or whitelist triggered by the connection.
Access Control Policy UUID	uint8[16]	The UUID of the access control policy configured for Security Intelligence.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Security Intelligence List Name field.
Security Intelligence List Name	string	The name of the Security Intelligence category IP blacklist or whitelist triggered by the connection.

User Data Block

The eStreamer service uses the User data block in User Record metadata messages to provide the user ID number, protocol on which the user was detected, and the user name. The User data block has a block type of 57 in the series 2 group of blocks.

The following graphic shows the structure of the User data block:



The following table describes the fields in the User data block.

Table 4-90 User Data Block Fields

Field	Data Type	Description
User Block Type	uint32	Initiates a User block. This value is always 57.
User Block Length	uint32	Total number of bytes in the User block, including eight bytes for the User block type and length fields, plus the number of bytes of data that follows.
User ID	uint32	The unique identifier for the user.

Table 4-90 User Data Block Fields (continued)

Field	Data Type	Description
Protocol	uint32	Protocol used to detect or report the user. Possible values are: • 165 - FTP • 426 - SIP • 547 - AOL Instant Messenger • 683 - IMAP • 710 - LDAP • 767 - NTP • 773 - Oracle Database • 788 - POP3 • 1755 - MDNS
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the username String data block, including eight bytes for the block type and header fields plus the number of bytes in the Username field.
Username	string	The name of the user

Understanding Host Data Structures

This chapter describes the format of the Full Host Profile data block that conveys a set of data describing a single host. The eStreamer server generates and sends these blocks on request for host data. For information about the client request procedure, the message structure, and the delivery method, see Host Data and Multiple Host Data Message Format, page 2-28.

eStreamer uses the series 1 data block structure to package these Full Host profile blocks. For the general structure of series 1 blocks, see Series 1 Data Block Header, page 4-56. The Full Host Profile data block contains a number of encapsulated blocks which are individually described in the subsections where they are defined in Understanding Discovery & Connection Data Structures, page 4-1.

See the following sections for more information about current and legacy Full Host Profile data blocks:

- Full Host Profile Data Block 5.3+, page 5-1 describes the current Full Host Profile data block structure.
- Full Host Profile Data Block 5.0 - 5.0.2, page B-224 describes the legacy Full Host Profile data block structure for versions 5.0 - 5.0.2.

Full Host Profile Data Block 5.3+

The Full Host Profile data block for version 5.3+ contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures, page 4-1. The Full Host Profile data block a block type value of 149. It supersedes the prior version, which has a block type of 140.

Note An asterisk (*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Host Profile data block for 5.3+:

	Byte	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Full Host Profile Data Block (149)																																
	Data Block Length																																
	Host ID Host ID, continued Host ID, continued Host ID, continued																																
IP Addresses	List Block Type (11)																																
IP Addresses	List Block Length																																
IP Addresses	IP Address Data Blocks (143)*																																
	Hops									Generic List Block Type (31)																							
	Generic List Block Type, continued									Generic List Block Length																							
OS Derived Fingerprints	Generic List Block Length, continued									Operating System Fingerprint Block Type (130)*																							
OS Derived Fingerprints	OS Fingerprint Block Type (130)*, con't									Operating System Fingerprint Block Length																							
OS Derived Fingerprints	OS Fingerprint Block Length, con't									Operating System Derived Fingerprint Data...																							
	Generic List Block Type (31)																																
	Generic List Block Length																																
Server Fingerprints	Operating System Fingerprint Block Type (130)*																																
Server Fingerprints	Operating System Fingerprint Block Length																																
Server Fingerprints	Operating System Server Fingerprint Data...																																
	Generic List Block Type (31)																																
	Generic List Block Length																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Client Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Client Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
User Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System User Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Scan Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Scan Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Application Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Application Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Conflict Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Mobile Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Mobile Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
IPv6 Server Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System IPv6 Server Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Ipv6 Client Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Ipv6 Client Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ipv6 DHCP Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System IPv6 DHCP Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
User Agent Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System User Agent Fingerprint Data...																															
(TCP) Full Server Data	List Block Type (11)...																															
	List Block Length...																															
	(TCP) Full Server Data Blocks (104)*																															
(UDP) Full Server Data	List Block Type (11)																															
	List Block Length																															
	(UDP) Full Server Data Blocks (104)*																															
Network Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Network) Protocol Data Blocks (4)*																															
Transport Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Transport) Protocol Data Blocks (4)*																															
MAC Address Data	List Block Type (11)																															
	List Block Length																															
	Host MAC Address Data Blocks (95)*																															
	Last Seen																															
	Host Type																															
	Business Criticality																VLAN ID															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	VLAN Type								VLAN Priority								Generic List Block Type (31)															
Host Client Data	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Full Host Client Application Data Blocks (112)*															
NetBios Name	String Block Type (0)																															
Name	String Block Length																															
	NetBIOS Name String...																															
Notes Data	String Block Type (0)																															
	String Block Length																															
	Notes String....																															
(VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty Scan Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*																															
Attribute Value Data	List Block Type (11)																															
	List Block Length																															
	Attribute Value Data Blocks *																															
	Mobile								Jailbroken								Generic List Block Type (31)															
IOC State	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																IOC State Data Blocks (150)*															

The following table describes the components of the Full Host Profile for 5.3+ record.

Table 5-1 Full Host Profile Record 5.3+ Fields

Field	Data Type	Description
Host ID	uint8[16]	Unique ID number of the host. This is a UUID.
List Block Type	uint32	Initiates a List data block comprising IP address data blocks conveying TCP service data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated IP address data blocks.
IP Address	variable	IP addresses of the host and when each IP address was last seen. See Host IP Address Data Block, page 4-89 for a description of this data block.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always `31`.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying mobile device fingerprint data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a mobile device host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (IPv6 Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 DHCP) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Agent) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — Host • 1 — Router • 2 — Bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Cisco-detected IDs. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.

Table 5-1 Full Host Profile Record 5.3+ Fields (continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block, page 4-74 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IOC State data blocks.This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IOC State data blocks.
IOC State Data Blocks *	variable	IOC State data blocks containing information about compromises on a host. See IOC State Data Block for 5.3+, page 4-28 for a description of this data block.

Configuring eStreamer

After you create a client application, you can connect it to the eStreamer server, start the eStreamer service, and begin exchanging data.

Note An *eStreamer server* is a Management Center or managed device (version 4.9 or higher) where the eStreamer service is running.

Perform the following tasks to manage eStreamer and client interaction:

1. Enable eStreamer on the eStreamer server.

   See Configuring eStreamer on the eStreamer Server, page 6-1 for information about allowing access to the eStreamer server, adding clients, and generating authentication credentials to establish an authenticated connection.

2. If required, manually run the eStreamer service (eStreamer). You can stop, start, and view the status of the service, and use command line options to debug client-server communication.

   See Managing the eStreamer Service, page 6-4 for more information.

3. Optionally, to use the eStreamer reference client to troubleshoot a connection or data stream, set up the reference client on the computer where you plan to run your client.

   See Configuring the eStreamer Reference Client, page 6-6.

Configuring eStreamer on the eStreamer Server

License: Any

Before the Management Center or managed device you want to use as an eStreamer server can begin streaming events to a client application, you must configure the eStreamer server to send events to clients, provide information about the client, and generate a set of authentication credentials to use when establishing communication. You can perform all of these tasks from the Management Center or managed device user interface.

See the following sections for more information:

- Configuring eStreamer Event Types, page 6-2
- Adding Authentication for eStreamer Clients, page 6-3

Configuring eStreamer Event Types

License: Any

You can control which types of events the eStreamer server is able to transmit to client applications that request them.

Available event types on a managed device or a Management Center include:

- Intrusion events
- Intrusion event packet data
- Intrusion event extra data

Available event types on a Management Center include:

- Discovery events (this also enables connection events)
- Correlation and white list events
- Impact flag alerts
- User activity events
- Malware events
- File events

Note that the primary and secondary in a stacked 3D9900 pair report intrusion events to the Management Center as if they were separate managed devices. If you configure communication with an eStreamer client on the primary in a 3D9900 stack, you also need to configure the client on the secondary; the client configuration is not replicated. Similarly, when you delete the client, delete it in both places. If you configure an eStreamer client for a Management Center managing 3D9900s in a stack configuration, note that the Management Center reports all events received from both managed devices, even if the same event is reported by both.

If you configure an eStreamer client on a Management Center in a high availability configuration, the client configuration is not replicated from the primary Management Center to the secondary Management Center.

To configure the types of events captured by eStreamer:

Access: Admin

Step 1 Select **System > Local > Registration**.

Step 2 Click **eStreamer**.

The eStreamer page appears with the **eStreamer Event Configuration** menu.

Step 3 Select the check boxes next to the types of events you want eStreamer to capture and forward to requesting clients. Note that if a check box is currently unchecked, that data is not being captured. Unchecking a check box does not delete data that has already been captured.

You can select any or all of the following on a Management Center or managed device:

- **Intrusion Events** to transmit intrusion events generated by managed devices.
- **Intrusion Event Packet Data** to transmit packets associated with intrusion events.
- **Intrusion Event Extra Data** to transmit additional data associated with intrusion events, such as the URI associated with the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

You can also select any or all of the following on a Management Center:

- **Discovery Events** to transmit host discovery events
- **Correlation Events** to transmit correlation and white list events.
- **Impact Flag Alerts** to transmit impact alerts generated by the Management Center.
- **User Activity Events** to transmit user events.
- **Intrusion Event Extra Data** to transmit additional data for intrusion events, such as the URI associated with the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

Note Note that this controls which events the eStreamer server can transmit. Your client application must still specifically request the types of events you want it to receive. For more information, see Request Flags, page 2-11.

Step 4 Click **Save**.

Your settings are saved and the events you selected will be forwarded to eStreamer clients when requested.

Adding Authentication for eStreamer Clients

License: Any

Before eStreamer can send events to a client, you must add the client to the eStreamer server's peers database. You must also copy the authentication certificate generated by the eStreamer server to the client.

To add an eStreamer client:

Access: Admin

Step 1 Select **Local** > **Registration** > **eStreamer**.

The **eStreamer** page appears.

Step 2 Click **Create Client**.

The Create Client page appears.

Step 3 In the **Hostname** field, enter the host name or IP address of the host running the eStreamer client.

Note If you use a host name, the host input server **must** be able to resolve the host to an IP address. If you have not configured DNS resolution, you should configure it first or use an IP address.

Step 4 If you want to encrypt the certificate file, enter a password in the **Password** field.

Step 5 Click **Save**.

The eStreamer server allows the client computer to access port 8302 on the Management Center and creates an authentication certificate to use during client-server authentication. The eStreamer Client page re-appears, with the new client listed under **eStreamer Clients**.

Step 6 Click the download icon ( ) next to the certificate file.

Step 7 Save the certificate file to the directory used by your client computer for SSL authentication.

The client can now connect to the Management Center.

Tip To revoke access for a client, click the delete icon ( ) next to the host you want to remove. Note that you do not need to restart the host input service on the Management Center; access is revoked immediately.

Managing the eStreamer Service

License: Any

You can manage the eStreamer service from the user interface. However, you can also use the command line to start and stop the service. The following sections describe eStreamer command line options:

- Starting and Stopping the eStreamer Service, page 6-4 describes how to start and stop the eStreamer service.
- eStreamer Service Options, page 6-4 describes the command line options available for the eStreamer service and how to use them.

Starting and Stopping the eStreamer Service

License: Any

You can manage the eStreamer service using the `manage_estreamer.pl` script, which allows you to start, stop, reload, and restart the service.

Tip You can also add command line options to the eStreamer initialization script. See eStreamer Service Options, page 6-4 for more information.

The following table describes the options in the `manage_estreamer.pl` script you can use on the Management Center or managed device.

Table 6-1 eStreamer Management Options

Option	Description	Select option Number....
enable	Starts the service.	3
disable	Stops the service.	2
restart	Restarts the service.	4
status	Indicates whether the service is running.	1

eStreamer Service Options

License: Any

eStreamer provides many service options that allow you to troubleshoot the service. You can use the options described in the following table with the eStreamer service.

Table 6-2 eStreamer Service Options

Option	Description
`--debug`	Runs eStreamer with debug-level logging. Errors are saved in the syslog and (when used in conjunction with `--nodaemon`) appear on screen.
`--nodaemon`	Runs eStreamer as a foreground process. Errors appear on-screen.
`--nohostcheck`	Runs eStreamer with host name checking disabled. That is, if the client host name does not match the host name contained in the subjectAltName:dNSName entry in the client certificate, access is still allowed. The `nohostcheck` option is useful in cases where the network DNS and/or NAT configuration prevent the host name check from succeeding. Note that all other security checks are performed. **Caution** Enabling this option can negatively affect the security of your system.

Use the above options by first stopping the eStreamer service, then running it with the options you want, and finally restarting the service. For example, you can follow the instructions provided in Running the eStreamer Service in Debug Mode, page 6-5 to debug eStreamer functionality.

Running the eStreamer Service in Debug Mode

License: Any

You can run the eStreamer service in debug mode to view each status message the service generates on your terminal screen. Use the following procedure to do debugging.

To run the eStreamer service in debug mode:

Access: Admin

Step 1 Log into the Management Center or managed device using SSH.

Step 2 Use `manage_estreamer.pl` and select option `2` to stop the eStreamer service.

Step 3 Use `./usr/local/sf/bin/sfestreamer --nodaemon --debug` to restart the eStreamer service in debug mode.

Status messages for the service appear on the terminal screen.

Step 4 When you are finished debugging, restart the service in normal mode using `manage_estreamer.pl` and selecting option `4`.

Configuring the eStreamer Reference Client

The *reference client* provided with the eStreamer SDK is a set of sample client scripts and Perl modules included to illustrate how the eStreamer API can be used. You can run them to familiarize yourself with eStreamer output, or you can use them to debug problems with installations of your custom-built client.

For more information on setting up the reference client, see the following sections:

- Setting Up the eStreamer Perl Reference Client, page 6-6
- Running the eStreamer Perl Reference Client, page 6-11

Setting Up the eStreamer Perl Reference Client

To use the eStreamer Perl reference client, you must first configure the sample scripts to fit your environment and requirements.

For more information, see the following sections:

- Understanding the eStreamer Perl Reference Client, page 6-6
- Configuring Communications for the eStreamer Reference Client, page 6-7
- Loading General Prerequisites for the Perl Reference Client, page 6-7
- Loading Prerequisites for the Perl SNMP Reference Client, page 6-7
- Understanding the Data Requested by a Test Script, page 6-8
- Modifying the Type of Data Requested by a Test Script, page 6-9
- Creating a Certificate for the Perl Reference Client, page 6-10

Understanding the eStreamer Perl Reference Client

You can download the `eStreamerSDK.zip` package, which contains the eStreamer Perl reference client, from the Cisco support site. The following files are included in the `eStreamerSDK.zip` package:

- `SF_CUSTOM_ALERT.MIB`

  This MIB file is used by the `snmp.pm` file to set up traps for SNMP.

- `SFRecords.pm`

  This Perl module contains definitions of discovery message record blocks.

- `SFStreamer.pm`

  This Perl module contains the functions called by the Perl clients.

- `SFPkcs12.pm`

  This Perl module parses the client certificate and allows the client to connect to the eStreamer server.

- `SFRNABlocks.pm`

  This Perl module contains definitions of discovery data blocks.

- `ssl_test.pl`

  You can use this Perl script to test an intrusion event request over an SSL connection.

- `OutputPlugins/csv.pm`

  This Perl module prints intrusion events to a comma-separated value (CSV) format.

- `OutputPlugins/print.pm`

  This Perl module prints events to a human-readable format.

- `OutputPlugins/snmp.pm`

  This Perl module sends events to the specified SNMP server.

- `OutputPlugins/pcap.pm`

  This Perl module stores packet captures as a pcap file.

- `OutputPlugins/syslog.pm`

  This Perl module sends events to the local syslog server.

Configuring Communications for the eStreamer Reference Client

The reference client uses the Secure Sockets Layer (SSL) for data communication. You must install OpenSSL on the computer you plan to use as a client and configure it appropriately for your environment.

Note For initial installations on Linux operating systems, you must install the `libssl-dev` component as part of this download.

To set up SSL on your client:

Step 1 Download OpenSSL from http://openssl.org/source/.

Step 2 Unpack the source to `/usr/local/src`.

Step 3 Configure the source by running the Configure script.

Step 4 Make and install the compiled source.

Loading General Prerequisites for the Perl Reference Client

Before you can run the eStreamer Perl reference client, you must install the `IO::Socket::SSL` Perl module on the client computer. You can install the module manually or use `cpan` to do so.

Note If the `Net::SSLeay` module is not installed on the client computer, install that module as well. `Net::SSLeay` is required for communication with OpenSSL.

You also need to install and configure OpenSSL to support an SSL connection to the eStreamer server. For more information, see Configuring Communications for the eStreamer Reference Client, page 6-7.

Loading Prerequisites for the Perl SNMP Reference Client

Before you can run the eStreamer SNMP module of the Perl reference client, you must install the latest `net-snmp` Perl modules available for the client operating system on the client computer.

Downloading and Unpacking the Perl Reference Client

You can download the `EventStreamerSDK.zip` file that contains the eStreamer Perl reference client the Cisco support site.

Unpack the zip file to a computer running the Linux operating system, where you plan to run the client.

Understanding the Data Requested by a Test Script

By default, when you use the `ssl_test -o` setting in the reference client, you request data as indicated in the following table.

Table 6-3 Default Requests Made by Output Plugins

This syntax...	Calls plugin...	And sends...	To request the following data...
`./ssl_test.pl` *`eStreamerServerName`* `-h HostIPAddresses`	N/A	Host request, message type 5, with bit 11 set to `1`	Host data (see Host Data and Multiple Host Data Message Format, page 2-28)
`./ssl_test.pl` *`eStreamerServerName`* `-o print -f` *`TextFile`*	OutputPlugins/print.pm	Event stream request, message type 2, with bits 2 and 20-24 set to `1`	Event data (see Event Stream Request Message Format, page 2-10, Correlation Policy Record, page 3-23, Correlation Rule Record, page 3-25, Metadata for Discovery Events, page 4-6, Host Discovery Structures by Event Type, page 4-38, and User Data Structures by Event Type, page 4-54) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o pcap -f` *`TargetPCAPFile`*	OutputPlugins/pcap.pm	Event stream request, message type 2, with bits 0 and 23 set to `1`	Packet data (see Event Data Message Format, page 2-17 and Packet Record 4.8.0.2+, page 3-5) eStreamer transmits only packet data because bit 0 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o csv -f` *`CSVFile`*	OutputPlugins/csv.pm	Event stream request, message type 2, with bits 2 and 23 set to `1`	Intrusion event data (see Event Data Message Format, page 2-17 and Intrusion Event Record 6.0+, page 3-7) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o snmp -f` *`SNMPServer`*	OutputPlugins/snmp.pm	Event stream request, message type 2, with bits 2, 20, and 23 set to `1`	Intrusion event data (see Event Data Message Format, page 2-17 and Intrusion Event Record 6.0+, page 3-7) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o syslog`	OutputPlugins/syslog.pm	Event stream request, message type 2, with bits 2, 20, and 23 set to `1`	Intrusion event data (see Event Data Message Format, page 2-17 and Intrusion Event Record 6.0+, page 3-7) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.

Modifying the Type of Data Requested by a Test Script

The `SFStreamer.pm` Perl module defines several request flag variables that you can use in the sample scripts to request data. The following table indicates what request flag variable to call to set each request flag in an event stream request message. If you want to request different data using one of the output modules, you can edit the `$FLAG` settings in the module.

For more information on the request flags, the data they request, and the product versions corresponding to each flag, see Request Flags, page 2-11.

Table 6-4 Request Flag Variables Used in Sample Scripts

Variable	Sets Request Flag...	To request the following data...
`$FLAG_PKTS`	`0`	Packet data
`$FLAG_METADATA`	`1`	Version 1 metadata
`$FLAG_IDS`	`2`	Type 1 intrusion events
`$FLAG_RNA`	`3`	Version 1 discovery events
`$FLAG_POLICY_EVENTS`	`4`	Version 1 correlation events
`$FLAG_IMPACT_ALERTS`	`5`	Intrusion impact alerts
`$FLAG_IDS_IMPACT_FLAG`	`6`	Type 7 intrusion events
`$FLAG_RNA_EVENTS_2`	`7`	Version 2 discovery events
`$FLAG_RNA_FLOW`	`8`	Version 1 connection data
`$FLAG_POLICY_EVENTS_2`	`9`	Version 2 correlation events
`$FLAG_RNA_EVENTS_3`	`10`	Version 3 discovery events
`$FLAG_HOST_ONLY`	`11`	When sent in conjunction with `$FLAG_HOST_SINGLE` (for one host) or `$FLAG_HOST_MULTI` (for multiple hosts), only host data with no event data
`$FLAG_RNA_FLOW_3`	`12`	Version 3 connection data
`$FLAG_POLICY_EVENTS_3`	`13`	Version 3 correlation events
`$FLAG_METADATA_2`	`14`	Version 2 metadata
`$FLAG_METADATA_3`	`15`	Version 3 metadata
`$FLAG_RNA_EVENTS_4`	`17`	Version 4 discovery events
`$FLAG_RNA_FLOW_4`	`18`	Version 4 connection data
`$FLAG_POLICY_EVENTS_4`	`19`	Version 4 correlation events
`$FLAG_METADATA_4`	`20`	Version 4 metadata
`$FLAG_RUA`	`21`	User activity events
`$FLAG_POLICY_EVENTS_5`	`22`	Version 5 correlation events
`$FLAGS_SEND_ARCHIVE_TIMESTAMP`	`23`	Extended event headers that include the timestamp applied when the event was archived for eStreamer server to process
`$FLAG_RNA_EVENTS_5`	`24`	Version 5 discovery events
`$FLAG_RNA_EVENTS_6`	`25`	Version 6 discovery events
`$FLAG_RNA_FLOW_5`	`26`	Version 5 connection data

Table 6-4 Request Flag Variables Used in Sample Scripts (continued)

Variable	Sets Request Flag...	To request the following data...
`$FLAG_EXTRA_DATA`	`27`	Intrusion event extra data record
`$FLAG_RNA_EVENTS_7`	`28`	Version 7 discovery events
`$FLAG_POLICY_EVENTS_6`	`29`	Version 6 correlation events
`$FLAG_DETAIL_REQUEST`	`30`	Extended request to eStreamer

Caution In all event types, prior to version 5.x, the reference client labels `detection engine ID` fields as `sensor ID`.

Creating a Certificate for the Perl Reference Client

License: Any

Before you can use the Perl reference client, you need to create a certificate on the Management Center or managed device for the computer where you want to run the client. You then download the certificate file to the client computer and use it to create a certificate (`server.crt`) and RSA key file (`server.key`).

To create a certificate for the Perl Reference Client:

Access: Admin

Step 1 Select **Operations** > **Configuration** > **eStreamer**.

The eStreamer page appears.

Step 2 Click **Create Client**.

The Create Client page appears.

Step 3 In the **Hostname** field, enter the host name or IP address of the host running the eStreamer client.

Note If you use a host name, the host input server **must** be able to resolve the host to an IP address. If you have not configured DNS resolution, you should configure it first or use an IP address.

Step 4 If you want to encrypt the certificate file, enter a password in the **Password** field.

Step 5 Click **Save**.

The eStreamer server allows the client computer to access port 8302 on the Management Center and creates an authentication certificate to use during client-server authentication. The eStreamer Client page re-appears, with the new client listed under **eStreamer Clients**.

Step 6 Click the download icon ( ) next to the certificate file.

Step 7 Save the certificate file to the directory used by your client computer for SSL authentication.

The client can now connect to the Management Center.

Tip To revoke access for a client, click the delete icon ( ) next to the host you want to remove. Note that you do not need to restart the host input service on the Management Center; access is revoked immediately.

Running the eStreamer Perl Reference Client

The eStreamer Perl reference client scripts are designed for use on a 64-bit operating system with the Linux kernel but should work on any POSIX-based 64-bit operating system, as long as the client machine meets the prerequisites defined in Setting Up the eStreamer Perl Reference Client, page 6-6.

For more information, see the following sections:

Testing a Client Connection over SSL Using a Host Request

You can use the `ssl_test.pl` script to test the connection between the eStreamer server and the eStreamer client. The `ssl_test.pl` script handles any record type and prints it to STDOUT or to an output plugin you specify. When you use the `-h` option without an output option, it streams host data for the specified hosts to your terminal.

Note You cannot use this script to stream packet data without directing it to an output plugin because printing raw packet data to STDOUT interferes with your terminal.

Use the following syntax to use the `ssl_test.pl` script to send host data to the standard output:

```
./ssl_test.pl eStreamerServerIPAddress -h HostIPAddresses
```

For example, to test receipt of host data for the hosts in the 10.0.0.0/8 subnet over a connection to an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -h 10.0.0.0/8
```

Capturing a PCAP Using the Reference Client

You can use the reference client to capture streamed packet data in a PCAP file to see the structure of the data the client receives. Note that you must use `-f` to specify a target file when you use the `-o pcap` output option.

Use the following syntax to capture streamed packet data in a PCAP file using the `ssl_test.pl` script:

```
./ssl_test.pl eStreamerServerIPAddress -o pcap -f ResultingPCAPFile
```

For example, to create a PCAP file named `test.pcap` using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o pcap -f test.pcap
```

Capturing CSV Records Using the Reference Client

You can also use the reference client to capture streamed intrusion event data in a CSV file to see the structure of the data the client receives.

Use the following syntax to run the `streamer_csv.pl` script:

`./ssl_test.pl` *`eStreamerServerIPAddress`* `-o csv -f` *`ResultingCSVFile`*

For example, to create a CSV file named `test.csv` using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o csv -f test.csv
```

Sending Records to an SNMP Server Using the Reference Client

You can also use the reference client to stream intrusion event data to an SNMP server. Use the `-f` option to indicate the name of the SNMP trap server that should receive events. Note that this output method requires a binary named `snmptrapd` in the path and therefore only works on UNIX-like systems.

Use the following syntax to send intrusion events to an SNMP server:

```
./ssl_test.pl eStreamerServerIPAddress -o snmp
   -f SNMPServerName
```

For example, to send events to an SNMP server at 10.10.0.3 using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o snmp -f 10.10.0.3
```

Logging Events to the Syslog Using the Reference Client

You can also use the reference client to stream intrusion events to the local syslog server on the client.

Use the following syntax to send events to the syslog:

```
./ssl_test.pl eStreamerServerIPAddress -o syslog
```

For example, to log events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o syslog
```

Connecting to an IPv6 Address

You can use the reference client to connect to a Management Center with an IPv6 address through the primary management interface. You must have the Socket6 and IO::Socket::INET6 Perl modules installed on the client machine and use the`-ipv6` option or the shortened form `-i`.

Use the following syntax to specify an IPv6 address using the ssl_test.pl script:

```
./ssl_test.pl -ipv6 eStreamerServerIPAddress
```

or

```
./ssl_test.pl -i eStreamerServerIPAddress
```

For example, to connect to a Management Center with the IPv6 address `2001:470:e09c:20:7c1e:5248:1bf7:2ea0` use the following:

```
./ssl_test.pl -ipv6 2001:470:e09c:20:7c1e:5248:1bf7:2ea0
```

Data Structure Examples

This appendix contains data structure examples for selected intrusion, correlation, and discovery events. Each example is displayed in binary format to clearly display how each bit is set.

See the following sections for more information:

- Intrusion Event Data Structure Examples
- Discovery Data Structure Examples, page A-17

Intrusion Event Data Structure Examples

This section contains examples of data structures that may be transmitted by eStreamer for intrusion events. The following examples are provided:

- Example of an Intrusion Event for the Management Center 5.4+, page A-1
- Example of an Intrusion Impact Alert, page A-6
- Example of a Packet Record, page A-8
- Example of a Classification Record, page A-9
- Example of a Priority Record, page A-11
- Example of a Rule Message Record, page A-12
- Example of a Version 5.1+ User Event, page A-14

Example of an Intrusion Event for the Management Center 5.4+

The following diagram shows an example event record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
5	0	1	0	1	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
11	0	1	0	1	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1	1	0	1	1	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	1	0	1
20	1	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	1	1	1	1	1	1	0	0	1	0	0	0	0
21	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
24	1	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	1	1	1	1	1	1	0
	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0	0
	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1
	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1
27	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0	1	0	0	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
30	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1	1	0
	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0	0
	1	0	1	0	0	1	0	1	1	1	1	1	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1
	0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	0	0	1	0	1	0	0
31	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	0	0	1	0	0	0	1	1	0	1	0
	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1
	1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	0	1
	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1	0	0	1	1	1	0	0
32	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	0	0	1	0	0	0	1	1	0	1	0
	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1
	1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	0	1
	1	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1	0	0	1	1	1	0	0
33	0	0	1	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0
	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1
	1	0	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1
	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1
34	0	0	1	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1
	1	0	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1
	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1
35	0	1	0	1	0	0	1	1	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

In the preceding example, the following event information appears:

Number	Description
1	The first two bytes of this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (that is, message type four).
2	This line indicates that the message that follows is `294` bytes long.

Number	Description
3	The first bit of this is a flag indicating that the header is an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `400`, which represents an intrusion event record.
4	This line indicates that the event record that follows is `278` bytes long.
5	This line is the timestamp when the event was saved. In this case, it was saved on Wednesday, July 2, 2014 at 16:11:27.
6	This line is reserved for future use and is populated with zeros.
7	This line indicates that the block type is `45`, which is the block type for Intrusion Event records for version 5.4+.
8	This line indicates that the data block is `278` bytes long.
9	This line indicates that the event is collected from sensor number `5`.
10	This line indicates that the event identification number is `65580`.
11	This line indicates that the event occurred at second `1404317489`.
12	This line indicates that the event occurred at microsecond `46542`.
13	This line indicates that the rule ID number is `4`.
14	This line indicates that the event was detected by generator ID number `119`, the rules engine.
15	This line indicates that the rule revision number is `1`.
16	This line indicates that the classification identification number is `1`.
17	This line indicates that the priority identification number is `3`.
18	This line indicates that the source IP address is `10.5.61.220`. Note that this field can contain either IPv4 or IPv6 addresses.
19	This line indicates that the destination IP address is `10.5.56.133`. Note that this field can contain either IPv4 or IPv6 addresses.
20	The first two bytes in this line indicate that the source port number is `33018`, and the second two bytes indicate that the destination port number is `8080`.
21	This first byte in this line indicates that TCP (`6`) is the protocol used in the event. The second byte is the impact flag, which indicates that the event is red (vulnerable) since the second bit is `1`; that the source or destination host is in a network monitored by the system, the source or destination host exists in the network map, and that the source or destination host is running a server on the port in the event; because the second and third flags are one, this is an orange event which is potentially vulnerable. The third byte in this line is the impact, which is 2 indicating that the event is orange and potentially vulnerable. The last byte indicates that the event was not blocked.
22	This line contains the MPLS label, if present.
23	The first two bytes in this line indicate that the VLAN ID is `0`. The last two bytes are reserved and set to `0`.
24	This line contains the unique ID number for the intrusion policy.
25	This line contains the internal identification number for the user. Since there is no applicable user, it is all zeros.
26	This line contains the internal identification number for the web application, which is `847`.

Number	Description
27	This line contains the internal identification number for the client application, which is `2000000676`.
28	This line contains the internal identification number for the application protocol, which is `676`.
29	This line contains the unique identifier for the access control rule, which is `1`.
30	This line contains the unique identifier for the access control policy.
31	This line contains the unique identifier for the ingress interface.
32	This line contains unique identifier for the egress interface. Since this event was blocked.
33	This line contains the unique identifier for the ingress security zone.
34	This line contains the unique identifier for the egress security zone.
35	This line contains the Unix timestamp of the connection event associated with the intrusion event.
36	The first two bytes in this line indicate the numerical ID of the Snort instance on the managed device that generated the connection event. The remaining two bytes indicate the value used to distinguish between connection events that happen during the same second.
37	The first two bytes in this line indicate the code for the country of the source host. The remaining two bytes indicate the code for the country of the destination host.
38	The first two bytes of this line contain the ID number of the compromise associated with this event. The remaining two bytes contain the beginning of the ID number for the security context (virtual firewall) that the traffic passed through.
39	This line contains the rest of the ID number for the security context (virtual firewall) that the traffic passed through.
40	The first two bytes of this line contain the last two bytes of the security context (virtual firewall) that the traffic passed through. The second two bytes contain the beginning of the SHA1 Hash of the SSL Server certificate if SSL was used.
41	This line contains the rest of the SHA1 Hash of the SSL Server certificate if SSL was used.
42	The first two bytes of this line contain the last two bytes of the SHA1 Hash of the SSL Server certificate. The second two bytes contain the SSL Action which was actually taken. Since SSL was not used in this connection, this is `0`.
43	The first two bytes of this line contain the SSL Flow Status. Since SSL was not used in this connection, this is `0`. The second two bytes contain the first two bytes of the UUID of the Network Analysis Policy associated with this event.
44	This line contains the rest of the UUID of the Network Analysis Policy associated with this event.

Example of an Intrusion Impact Alert

The following diagram shows an example intrusion impact alert record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
9	0	1	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
15	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1	0
	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0
	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	1																

In the preceding example, the following information appears:

Number	Description
1	The first two bytes of this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (message type four).
2	This line indicates that the message that follows is `58` bytes long.
3	The first bit of this is a flag indicating that the header is not an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `9`, which represents an intrusion impact alert record.
4	This line indicates that the data that follows is `50` bytes long.
5	This line contains a value of `20`, indicating that an intrusion impact alert data block follows.

Number	Description
6	This line indicates that the length of the impact alert block, including the impact alert block header, is `50` bytes.
7	This line indicates that the event identification number is `201256`.
8	This line indicates that the event is collected from device number `2`.
9	This line indicates that the event occurred at second `1087223700`.
10	This line indicates that `1` (red, vulnerable) is the impact level associated with the event.
11	This line indicates that the IP address associated with the violation event is `172.16.1.22`.
12	This line indicates that there is no destination IP address associated with the violation (values are set to `0`).
13	This line indicates that a string block follows, containing a string block length and a text string which, in this case, contains the impact name. For more information about string blocks, see String Data Block, page 3-55.
14	This line indicates that the total length of the string block, including the string block indicator and length is `18` bytes. This includes 10 bytes for the impact description and 8 bytes for the string header.
15	This line indicates that the description of the impact is “Vulnerable.”

Example of a Packet Record

The following diagram shows an example packet record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	0	1	1	0
7	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0
8	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	1	1	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
12	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0

In the preceding example, the following packet information appears:

Number	Description
1	The first two bytes of this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (message type four).
2	This line indicates that the message that follows is `989` bytes long.
3	The first bit of this is a flag indicating that the header is not an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `2`, which represents a packet record.
4	This line indicates that the packet record that follows is `981` bytes long.
5	This line indicates that the event is collected from device number `3`.
6	This line indicates that the event identification number is `195430`.
7	This line indicates that the event occurred at second `10572378`.
8	This line indicates that the packet was collected at second `10572380`.
9	This line indicates that the packet was collected at microsecond `254365`.
10	This line indicates that the link type is `1` (Ethernet layer).
11	This line indicates that the packet data that follows is `953` bytes long.
12	This line and the following line show the actual payload data. Note that the actual data is `953` bytes and has been truncated for the sake of this example.

Example of a Classification Record

The following diagram shows an example classification record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	0	1	0
	0	1	1	0	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
	0	0	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0	0
	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	0
7	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0
	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	0	1	1
	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1
	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0
	0	1	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1
	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0	0	1	0	0
8	1	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	0	0
	1	1	0	0	1	0	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1
	1	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In the preceding example, the following event information appears:

Number	Description
1	The first two bytes of the line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (message type four).
2	This line indicates that the message that follows is `92` bytes long.

Number	Description
3	The first bit of this is a flag indicating that the header is not an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `67`, which represents a classification record.
4	This line indicates that the classification record that follows is `84` bytes long.
5	This line indicates that the Classification ID is `35`.
6	The first two bytes of this line indicate that the classification name that follows it is `15` bytes long. The second two bytes begin the classification name itself, which, in this case, is “trojan-activity”.
7	The first byte in this line is a continuation of the classification name described in line 6. The next two bytes in this line indicate that the classification description that follows it is `29` bytes long. The remaining byte begins the classification description, which, in this case, is `"A Network Trojan was Detected."`
8	This line indicates the classification ID number that acts as a unique identifier for the classification.
9	This line indicates the classification revision ID number that acts as a unique identifier for the classification revision, which is null because there are no revisions to the classification.

Example of a Priority Record

The following example shows a sample priority record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0	1
	0	1	1	0	0	1	1	1	0	1	1	0	1	0	0	0																

In the preceding example, the following event information appears:

Number	Description
1	The first two bytes in this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (message type four).
2	This line indicates that the message that follows is `16` bytes.
3	This line indicates a record type value of `4`, which represents a priority record.
4	This line indicates that the priority record that follows is `8` bytes long.
5	This line indicates that the priority ID is one.
6	The first two bytes of this line indicate that there are four bytes included in the priority name. The second two bytes plus the two bytes on the following line show the priority name itself (“high”).

Example of a Rule Message Record

The following example shows a sample rule record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
9	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1
	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1
	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	1
10	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1
	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	1	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	1
11	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0
	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1
	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	1	1	0	1	0	1	0	1	0	0
	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1
	0	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1
	0	1	1	1	0	1	0	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1	0	1	0	0
	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	0	0	1	0
	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	0	1	1	1	0	1	0	0
	0	1	1	0	0	1	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	1
	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	1	0	1
	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	1	1	0	0	0	0	1
	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1
	0	1	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	0	0	1	1	1
	0	1	1	1	0	1	0	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	1	0	0
	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	0	0	0	0	0
	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1
	0	1	1	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1
	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	0	0	0	1	1
	0	1	1	0	1	1	1	0																								

In the preceding example, the following event information appears:

Number	Description
1	The first two bytes of this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (that is, message type four).
2	This line indicates that the message that follows is `129` bytes.

Number	Description
3	The first bit of this is a flag indicating that the header is not an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `66`, which represents a rule message record.
4	This line indicates that the rule message record that follows is `121` bytes long.
5	This line indicates that the generator identification number is `1`, the rules engine.
6	This line indicates that the rule identification number is `28069`.
7	This line indicates that the rule revision number is `1`.
8	This line indicates that the rule identification number rendered to the Firepower System is `28069`.
9	The first two bytes of this line indicate that there are `71` bytes included in the rule text name. The second two bytes begin the unique identifier number for the rule.
10	The first two bytes of this line finish the unique identifier number of the rule. The next two bytes begin the unique identifier number for the revision of the rule.
11	The first two bytes of this line finish the unique identifier number for the revision of the rule. The second two bytes begin the text of the rule message itself. The full text of the transmitted rule message is: `APP-DETECT DNS request for potential malware SafeGuard to domain 360.cn`.

Example of a Version 5.1+ User Event

The following diagram shows an example user event record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
5	0	1	0	1	0	0	1	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
11	0	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	0	1	0
16	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
20	0	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	0	1	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
24	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1
	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0
	0	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1
	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0	1	0	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0																								

In the preceding example, the following information appears:

Number	Description
1	The first two bytes of this line indicate the standard header value of `1`. The second two bytes indicate that the message is a data message (that is, message type four).
2	This line indicates that the message that follows is `153` bytes long.
3	The first bit of this is a flag indicating that the header is an extended header containing an archive timestamp. The next 15 bits are an optional field containing the Netmap ID for the domain on which the event was detected. The remainder of the line indicates a record type value of `95`, which represents a user information update message block.
4	This line indicates that the data that follows is `137` bytes long.
5	This line contains the archive timestamp. It is included since bit 23 was set. The timestamp is a Unix timestamp, stored as seconds since 1/1/1970. This time stamp is `1,391,789,354`, which is Mon Feb 3 19:43:49 2014.
6	This line contains zeros and is reserved for future use.
7	This line indicates that the detection engine ID is `3`.
8	This line is for the legacy (IPv4) IP address. It contains all zeros as it is not populated and the IPv4 address is stored in the IPv6 field.
9	This line contains the MAC address associated with the event. As there is no MAC address, it contains zeros.
10	The first half of this line is the remainder of the MAC address, which is zeros. The next byte indicates the presence of an IPv6 address. The last byte in this line is reserved for future use and contains zeros.
11	This line contains the UNIX timestamp (seconds since 01/01/1970) that the system generated the event.
12	This line contains the microsecond (one millionth of a second) increment that the system generated the event.
13	This line contains the event type. This has a value of `1004`, which indicates a user modification message.
14	This line contains the event subtype. This has a value of `2`, which indicates a user login event.

Number	Description
15	This line contains the serial file number. This field is for internal use and can be disregarded.
16	This line contains the event's position in the serial file. This field is for internal use and can be disregarded.
17	This line contains the IPv6 address. This field is present and used if the Has IPv6 flag is set. In this case, however, it contains the IPv4 address `10.4.15.120`.
18	This line initiates a User Login Information data block, indicated by block type `127`.
19	This line indicates that the block that follows is `81` bytes long.
20	This line indicates that the user login timestamp is `1,391,456,7`, which means it was generated at Mon, 03 Oct 2014 19:43:47 GMT.
21	This line is for the legacy (IPv4) IP address. It contains all zeros as it is not populated and the IPv4 address is stored in the IPv6 field.
22	This line indicates that a string block follows, containing a string block length and a text string which, in this case, contains the user name. For more information about string blocks, see String Data Block, page 3-55.
23	This line indicates that the length of the data in the string block is `16` bytes.
24	This line indicates that the name of the user is "`301@10.4.11.175`."
25	The line indicates the ID number of the user.
26	This line indicates the application ID for the application protocol used in the connection that the login information was derived from.
27	This line indicates that a string block follows, containing a string block length and a text string which, in this case, contains the email address. For more information about string blocks, see String Data Block, page 3-55.
28	This line indicates that the length of the data in the string block is `0` bytes. This is because there is no email address associated with this user.
29	This line contains IP address from the host where the user was detected logging in.
30	The first bye contains the login type. The remainder of this line indicates that a string block follows, containing a string block length and a text string which, in this case, contains the name of the Active Directory server reporting a login. For more information about string blocks, see String Data Block, page 3-55.
31	The first byte of this line completes the initiation of the string data block.This remainder of this line indicates that the length of the data in the string block is `0` bytes. This is because there is no Active Directory server associated with this login.

Discovery Data Structure Examples

This section contains examples of data structures that may be transmitted by eStreamer for discovery events. The following examples are provided:

- Example of a New Network Protocol Message, page A-18
- Example of a New TCP Server Message, page A-19

Example of a New Network Protocol Message

The following diagram illustrates a sample new network protocol message for 3.0+:

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
Header Version 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0	Start Standard Message Header with Event Msg (4)
Message Length (49B)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 1 1 0 0 0 1	
New NW Protocol Msg (13)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 1 1 0 1	
Msg Length 41B)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 1 0 1 0 0 0	
Detection Engine ID (2)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	
IP (192.168.1.10)	1 1 0 0 0 0 0 0	1 0 1 0 1 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 1 0 1 0	
MAC Address (none)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Reserved Bytes (0)
Unix Sec (1047242787)	0 0 1 1 1 1 1 0	0 1 1 0 1 0 1 1	1 0 1 0 1 0 0 0	0 0 1 0 0 0 1 1	
Unix MSec (973208)	0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 0	1 1 0 1 1 0 0 1	1 0 0 1 1 0 0 0	
Reserved Bytes (0)	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1	1 1 1 0 1 0 0 0	Event Type 1000—New
EventSub 4-New Trans Prot	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0	
File Number	0 1 0 0 0 0 0 0	0 1 0 0 0 1 1 1	1 0 0 0 1 0 0 1	1 1 0 1 0 0 0 1	
File Position	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0	End Standard Message Header
Protocol (6—TCP)	0 0 0 0 0 1 1 0				

Example of a New TCP Server Message

The following diagram illustrates a sample new TCP server message for 3.0:

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Header Version 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Start Standard Message Header with Event Msg (4)
Message Length (256B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
New TCP Svc Msg (11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	
Msg Length (248B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	
Detection Engine ID (2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
IP (192.168.1.10)	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	
MAC Address (none)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reserved Bytes (0)
Unix Sec (1047242787)	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	
Unix MSec (973208)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	0	
Reserved Bytes (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	Event Type 1000—New
Event Subtype 2 -New Host	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
File Number	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	1	
File Position	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	End Standard Message Header
Server Block Header (12)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	Start Server Data Block

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Server Length (208B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	
Server Port (80)	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Hits
Hits (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Length (13B)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	1	0	0	
Server Name (`https`)	0	1	1	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Length (15B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	1	
Server Vendor (`Apache` + null byte)	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0	
	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Length (8-no product)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Length (22B)	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	
Version - 1.3.26 (Unix)	0	0	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	
	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	
	0	1	1	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	
List Block Header (11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	Start Sub-server List
List Block Size (94B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Sub-server Hdr (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Start Sub-server Block
Sub-server Len (46B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Length (16B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
Sub-server Name - mod_ssl	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1	1	
	0	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Block Len (8B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	(No subtype vendor)
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Block Length (14B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
Sub-server Version - 2.8.9 + null character	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1	1	0	End Sub-server Block
	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Start Sub-server Block
Sub-server Hdr (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sub-server Length
Sub-server Length (48B)	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Size
String Block Size (16B)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	
Sub-server Name - OpenSSL	0	1	1	0	0	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	
	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Data Length
String Length (8-no vendor)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Hdr (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Len (16B)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	
Sub-server Version - 0.9.6.d + null byte	0	0	1	1	1	0	0	1	0	0	1	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	End Sub-server Block
	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Confidence %
Confidence % (100)	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	Last used
Last Used (1047242787)	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Blob Data Block
Blob Data Block (10)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Blob Data Length
Blob Data Length (22B)	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	1	0	0	
Server Banner (HTTP/1.1 414 Reque) -Server banner shortened for example, typically 256B.	0	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0	1	
	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0
	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0
	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	0	0	1	0	1	End Server Data Block

Understanding Legacy Data Structures

This appendix contains information about data structures supported by eStreamer at previous versions of Firepower System products.

If your client uses event stream requests with bits set to request data in older version formats, you can use the information in this appendix to identify the data structures of the data messages you receive.

Note that prior to version 5.0, separate detection engines were assigned IDs. For version 5.0, devices are assigned IDs. Based on the version, data structures reflect this.

Note This appendix describes only data structures from version 4.9 or later of the Firepower System. If you require documentation for structures from earlier data structure versions, contact Cisco Customer Support.

See the following sections for more information:

Legacy Intrusion Data Structures

Intrusion Event (IPv4) Record 5.0.x - 5.1

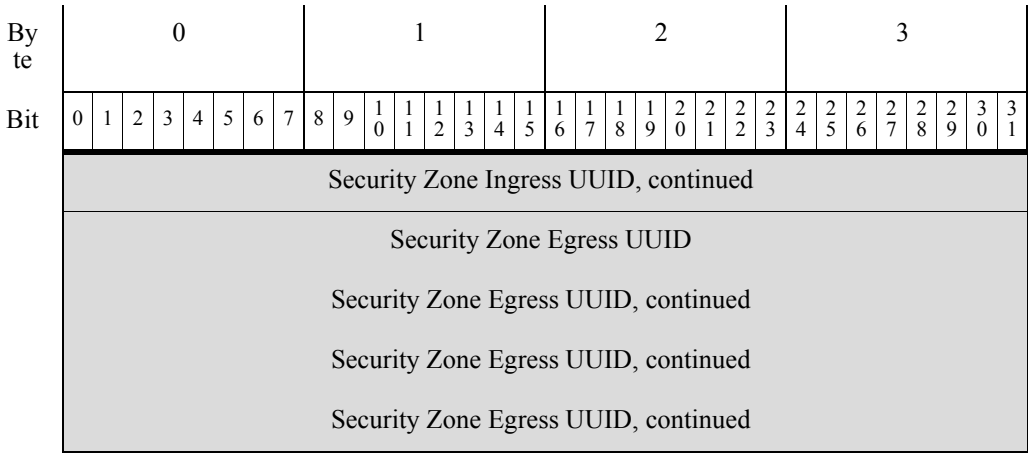
The fields in the intrusion event (IPv4) record are shaded in the following graphic. The record type is 207.

You request intrusion event records by setting the intrusion event flag or the extended requests flag in the request message. See Request Flags, page 2-11 and Submitting Extended Requests, page 2-4.

For version 5.0.x - 5.1 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (207)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IPv4 Address																																
Destination IPv4 Address																																
Source Port																	Destination Port															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
VLAN ID																	Pad															
Policy UUID																																
Policy UUID, continued																																
Policy UUID, continued																																
Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID																																
Access Control Policy UUID, continued																																
Access Control Policy UUID, continued																																
Access Control Policy UUID, continued																																
Interface Ingress UUID																																
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Egress UUID																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Security Zone Ingress UUID																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																



The following table describes each intrusion event record data field.

Table B-1 Intrusion Event (IPv4) Record Fields

Field	Data Type	Description
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv4 Address	uint8[4]	Source IPv4 address used in the event, in address octets.
Destination IPv4 Address	uint8[4]	Destination IPv4 address used in the event, in address octets.
Source Port	uint16	The source port number if the event protocol type is TCP or UDP.
Destination Port	uint16	The destination port number if the event protocol type is TCP or UDP.

Table B-1 Intrusion Event (IPv4) Record Fields (continued)

Field	Data Type	Description
IP Protocol Number	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be `0` or `1`: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`

Table B-1 Intrusion Event (IPv4) Record Fields (continued)

Field	Data Type	Description
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • 0 — Not blocked • 1 — Blocked • 2 — Would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.

Intrusion Event (IPv6) Record 5.0.x - 5.1

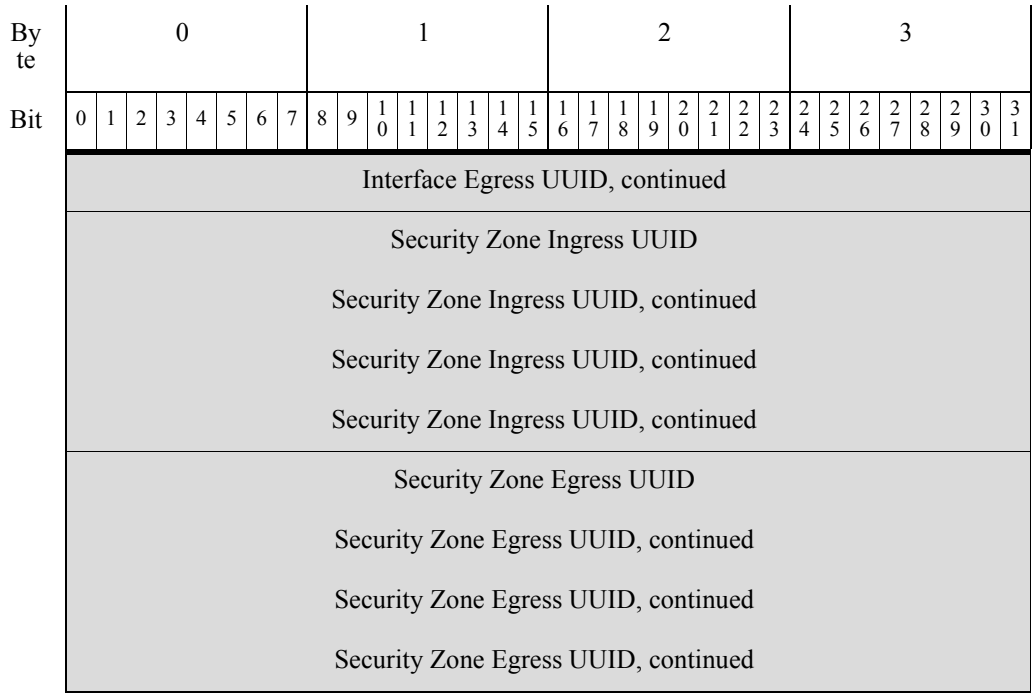
The fields in the intrusion event (IPv6) record are shaded in the following graphic. The record type is 208.

You request intrusion event records by setting the intrusion event flag or the extended requests flag in the request message. See Request Flags, page 2-11 and Submitting Extended Requests, page 2-4.

For version 5.0.x - 5.1 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (208)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IPv6 Address Source IPv6 Address, continued Source IPv6 Address, continued Source IPv6 Address, continued																																
Destination IPv6 Address Destination IPv6 Address, continued Destination IPv6 Address, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Destination IPv6 Address, continued																																
Source Port/ICMP Type																	Destination Port/ICMP Code															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued																																
Interface Egress UUID Interface Egress UUID, continued Interface Egress UUID, continued																																



The following table describes each intrusion event record data field.

Table B-2 Intrusion Event (IPv6) Record Fields

Field	Data Type	Description
Device ID	unit32	Contains the identification number of the detecting device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv6 Address	uint8[16]	Source IPv6 address used in the event, in address octets.
Destination IPv6 Address	uint8[16]	Destination IPv6 address used in the event, in address octets.

Table B-2 Intrusion Event (IPv6) Record Fields (continued)

Field	Data Type	Description
Source Port/ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP. If the protocol type is ICMP, this indicates the ICMP type.
Destination Port/ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP. If the protocol type is ICMP, this indicates the ICMP code.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be `0` or `1`: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`

Table B-2 Intrusion Event (IPv6) Record Fields (continued)

Field	Data Type	Description
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • 0 — Not blocked • 1 — Blocked • 2 — Would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label. (Applies to 4.9+ events only.)
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated. (Applies to 4.9+ events only.)
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.

Intrusion Event Record 5.2.x

The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 34 in the series 2 set of data blocks.

You can request 5.2.x intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 5 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).

For version 5.2.x intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (400)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Block Type (34)																																
Block Length																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Source Port or ICMP Type																	Destination Port or ICMP Code															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Egress UUID																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Security Zone Ingress UUID																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Egress UUID																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Connection Timestamp																																
Connection Instance ID																	Connection Counter															
Source Country																	Destination Country															

The following table describes each intrusion event record data field.

Table B-3 Intrusion Event Record 5.2.x Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always `34`.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.

Table B-3 Intrusion Event Record 5.2.x Fields (continued)

Field	Data Type	Description
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP

Table B-3 Intrusion Event Record 5.2.x Fields (continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be 0 or 1: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • `0` — Not blocked • `1` — Blocked • `2` — Would be blocked (but not permitted by configuration)

Table B-3 Intrusion Event Record 5.2.x Fields (continued)

Field	Data Type	Description
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.

Intrusion Event Record 5.3

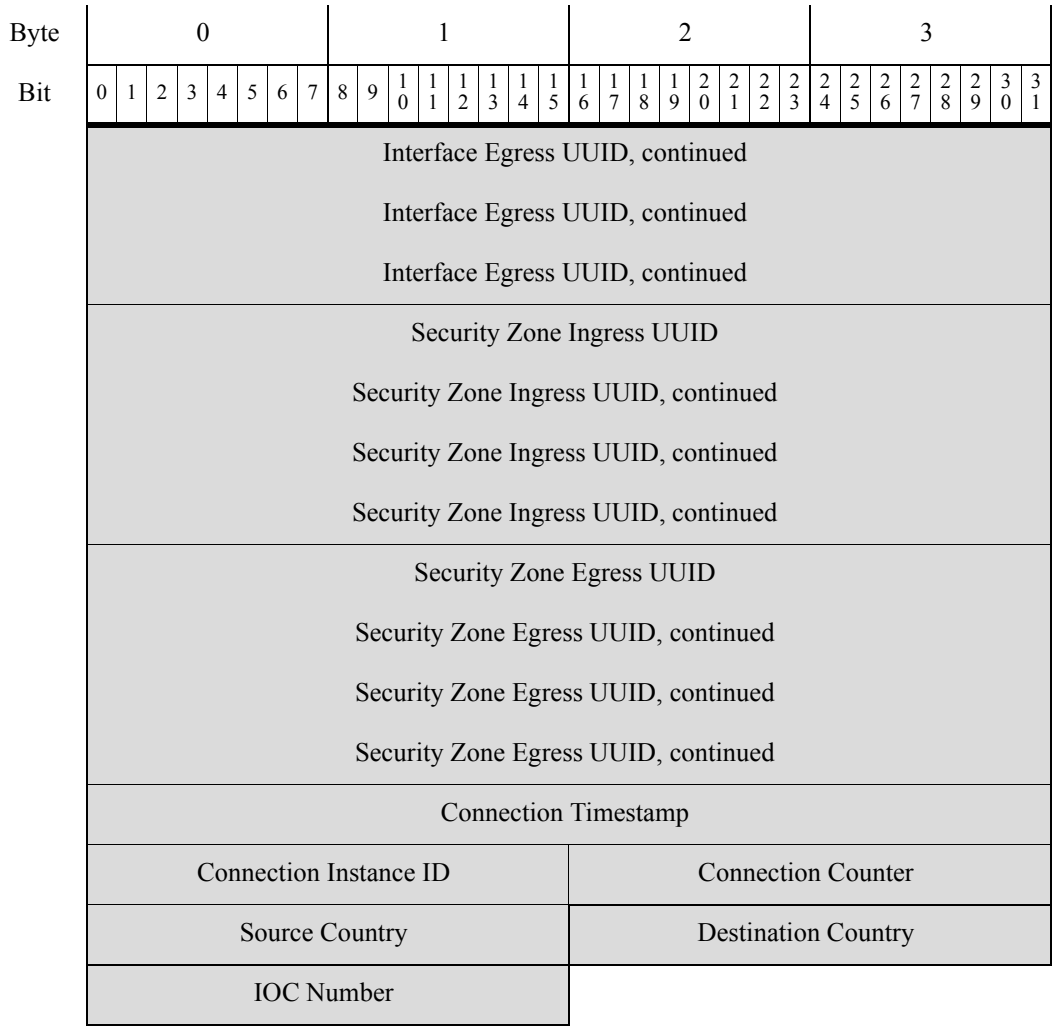
The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 41 in the series 2 set of data blocks.

You can request 5.3 intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 6 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).

For version 5.3 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (400)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Block Type (41)																																
Block Length																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Source Port or ICMP Type																	Destination Port or ICMP Code														
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued																																
Interface Egress UUID																																



The following table describes each intrusion event record data field.

Table B-4 Intrusion Event Record 5.3 Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always `34`.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.

Table B-4 Intrusion Event Record 5.3 Fields (continued)

Field	Data Type	Description
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP

Table B-4 Intrusion Event Record 5.3 Fields (continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be `0` or `1`: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • `0` — Not blocked • `1` — Blocked • `2` — Would be blocked (but not permitted by configuration)

Table B-4 Intrusion Event Record 5.3 Fields (continued)

Field	Data Type	Description
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
IOC Number	uint16	ID Number of the compromise associated with this event.

Intrusion Event Record 5.1.1.x

The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 25.

You can request 5.1.1.x intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 4 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).

For version 5.1.1.x intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (400)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Block Type (25)																																
Block Length																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Source Port/ICMP Type																	Destination Port/ICMP Code															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Interface Egress UUID																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Security Zone Ingress UUID																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Egress UUID																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Connection Timestamp																																
Connection Instance ID																	Connection Counter															

The following table describes each intrusion event record data field.

Table B-5 Intrusion Event Record 5.1.1 Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always `25`.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.

Table B-5 Intrusion Event Record 5.1.1 Fields (continued)

Field	Data Type	Description
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port/ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port/ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • `0` — IP • `1` — ICMP • `6` — TCP • `17` — UDP

Table B-5 Intrusion Event Record 5.1.1 Fields (continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be `0` or `1`: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • `0` — Not blocked • `1` — Blocked • `2` — Would be blocked (but not permitted by configuration)

Table B-5 Intrusion Event Record 5.1.1 Fields (continued)

Field	Data Type	Description
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.

Intrusion Event Record 5.3.1

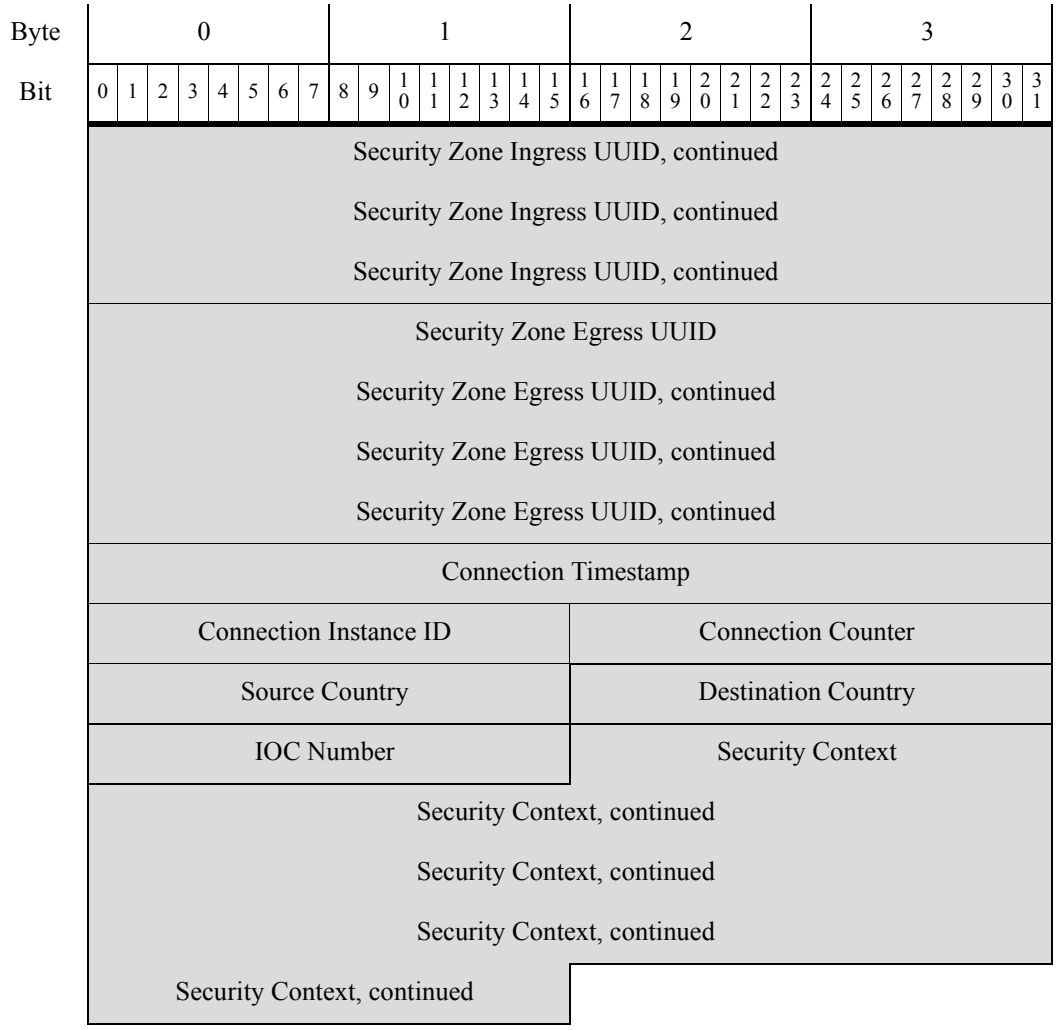
The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 42 in the series 2 set of data blocks.

You can request 5.3.1 intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 7 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).

For version 5.3.1 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (400)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Block Type (42)																																
Block Length																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source Port or ICMP Type																	Destination Port or ICMP Code															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID Policy UUID, continued Policy UUID, continued Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued																																
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued																																
Interface Egress UUID Interface Egress UUID, continued Interface Egress UUID, continued Interface Egress UUID, continued																																
Security Zone Ingress UUID																																



The following table describes each intrusion event record data field.

Table B-6 Intrusion Event Record 5.3.1 Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 42.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.

Table B-6 Intrusion Event Record 5.3.1 Fields (continued)

Field	Data Type	Description
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP

Table B-6 Intrusion Event Record 5.3.1 Fields (continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be 0 or 1: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • `0` — Not blocked • `1` — Blocked • `2` — Would be blocked (but not permitted by configuration)

Table B-6 Intrusion Event Record 5.3.1 Fields (continued)

Field	Data Type	Description
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.

Intrusion Event Record 5.4.x

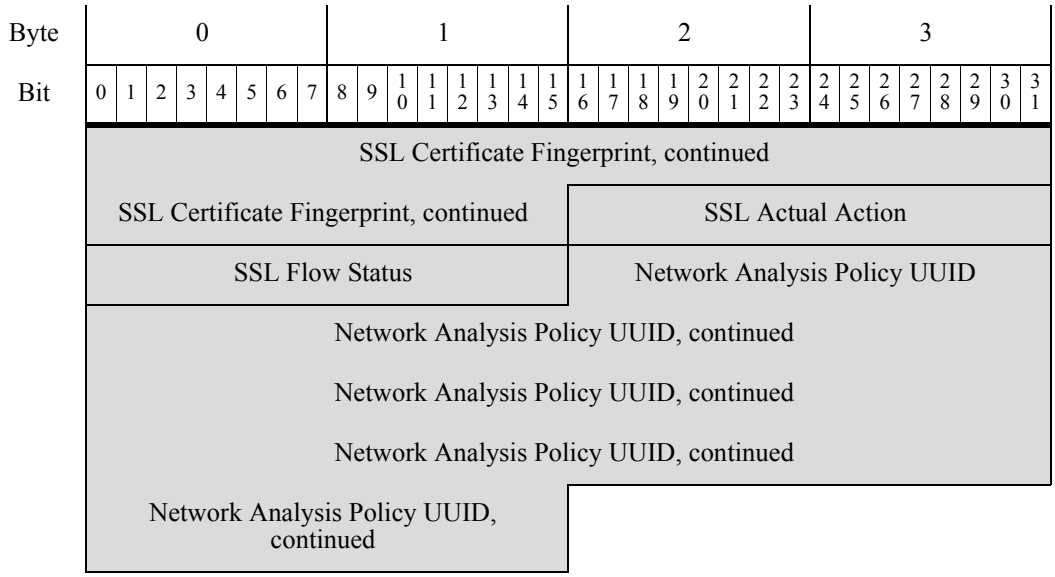
The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 45 in the series 2 set of data blocks. It supersedes block type 42, and is superseded by block type 60. Fields for SSL support and Network Analysis Policy have been added.

You can request 5.4.x intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 8 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests).

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (400)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Block Type (45)																																
Block Length																																
Device ID																																
Event ID																																
Event Second																																
Event Microsecond																																
Rule ID (Signature ID)																																
Generator ID																																
Rule Revision																																
Classification ID																																
Priority ID																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source IP Address																																
Source IP Address, continued																																
Source IP Address, continued																																
Source IP Address, continued																																
Destination IP Address																																
Destination IP Address, continued																																
Destination IP Address, continued																																
Destination IP Address, continued																																
Source Port or ICMP Type																	Destination Port or ICMP Code															
IP Protocol ID									Impact Flags								Impact								Blocked							
MPLS Label																																
VLAN ID																	Pad															
Policy UUID																																
Policy UUID, continued																																
Policy UUID, continued																																
Policy UUID, continued																																
User ID																																
Web Application ID																																
Client Application ID																																
Application Protocol ID																																
Access Control Rule ID																																
Access Control Policy UUID																																
Access Control Policy UUID, continued																																
Access Control Policy UUID, continued																																
Access Control Policy UUID, continued																																
Interface Ingress UUID																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Ingress UUID, continued																																
Interface Egress UUID																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Interface Egress UUID, continued																																
Security Zone Ingress UUID																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Ingress UUID, continued																																
Security Zone Egress UUID																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Security Zone Egress UUID, continued																																
Connection Timestamp																																
Connection Instance ID																	Connection Counter															
Source Country																	Destination Country															
IOC Number																	Security Context															
Security Context, continued																																
Security Context, continued																																
Security Context, continued																																
Security Context, continued																	SSL Certificate Fingerprint															
SSL Certificate Fingerprint, continued																																
SSL Certificate Fingerprint, continued																																
SSL Certificate Fingerprint, continued																																



The following table describes each intrusion event record data field.

Table B-7 Intrusion Event Record 5.4.x Fields

Field	Data Type	Description
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 45.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Firepower System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.

Table B-7 Intrusion Event Record 5.4.x Fields (continued)

Field	Data Type	Description
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`

Table B-7 Intrusion Event Record 5.4.x Fields (continued)

Field	Data Type	Description
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact)
Blocked	uint8	Value indicating whether the event was blocked. • 0 — Not blocked • 1 — Blocked • 2 — Would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.

Table B-7 Intrusion Event Record 5.4.x Fields (continued)

Field	Data Type	Description
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8[16]	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table B-7 Intrusion Event Record 5.4.x Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
Network Analysis Policy UUID	uint8[16]	The UUID of the Network Analysis Policy that created the intrusion event.

Intrusion Impact Alert Data

The Intrusion Impact Alert event contains information about impact events. It is transmitted when an intrusion event is compared to the system network map data and the impact is determined. It uses the standard record header with a record type of 9, followed by an Intrusion Impact Alert data block with a data block type of 20 in the series 1 group of blocks. (The Impact Alert data block is a type of series 1 data block. For more information about series 1 data blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.)

You can request that eStreamer only transmit intrusion impact events by setting bit 5 in the Flags field of the request message. See Event Stream Request Message Format, page 2-10 for more information about request messages. Version 1 of these alerts only handles IPv4. Version 2, introduced in 5.3, handles IPv6 events in addition to IPv4.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Netmap ID																Record Type (9)															
	Record Length																															
	Intrusion Impact Alert Block Type (20)																															
	Intrusion Impact Alert Block Length																															
	Event ID																															
	Device ID																															
	Event Second																															
	Impact																															
	Source IP Address																															
	Destination IP Address																															
Impact Description	String Block Type (0)																															
	String Block Length																															
	Description...																															

The following table describes each data field in an impact event.

Table B-8 Impact Event Data Fields

Field	Data Type	Description
Intrusion Impact Alert Block Type	uint32	Indicates that an intrusion impact alert data block follows. This field will always have a value of `20`. See Intrusion Event and Metadata Record Types, page 3-1.
Intrusion Impact Alert Block Length	uint32	Indicates the length of the intrusion impact alert data block, including all data that follows and 8 bytes for the intrusion impact alert block type and length.
Event ID	uint32	Indicates the event identification number.
Device ID	uint32	Indicates the managed device identification number.
Event Second	uint32	Indicates the second (from 01/01/1970) that the event was detected.
Impact	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be `0` or `1`: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX`, `XXX1XXXX`, `X1XXXXXX`, `1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`

Table B-8 Impact Event Data Fields (continued)

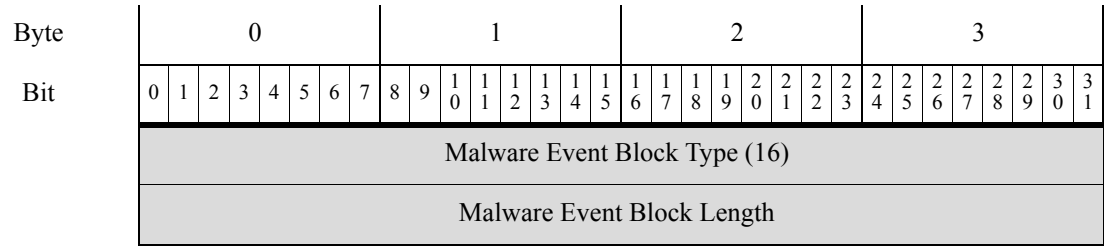
Field	Data Type	Description
Source IP Address	uint8[4]	IP address of the host associated with the impact event, in IP address octets.
Destination IP Address	uint8[4]	IP address of the destination IP address associated with the impact event (if applicable), in IP address octets. This value is `0` if there is no destination IP address.
String Block Type	uint32	Initiates a string data block that contains the impact name. This value is always set to `0`. For more information about string blocks, see String Data Block, page 4-64.
String Block Length	uint32	Number of bytes in the event description string block. This includes the four bytes for the string block type, the four bytes for the string block length, and the number of bytes in the description.
Description	string	Description of the impact event.

Legacy Malware Event Data Structures

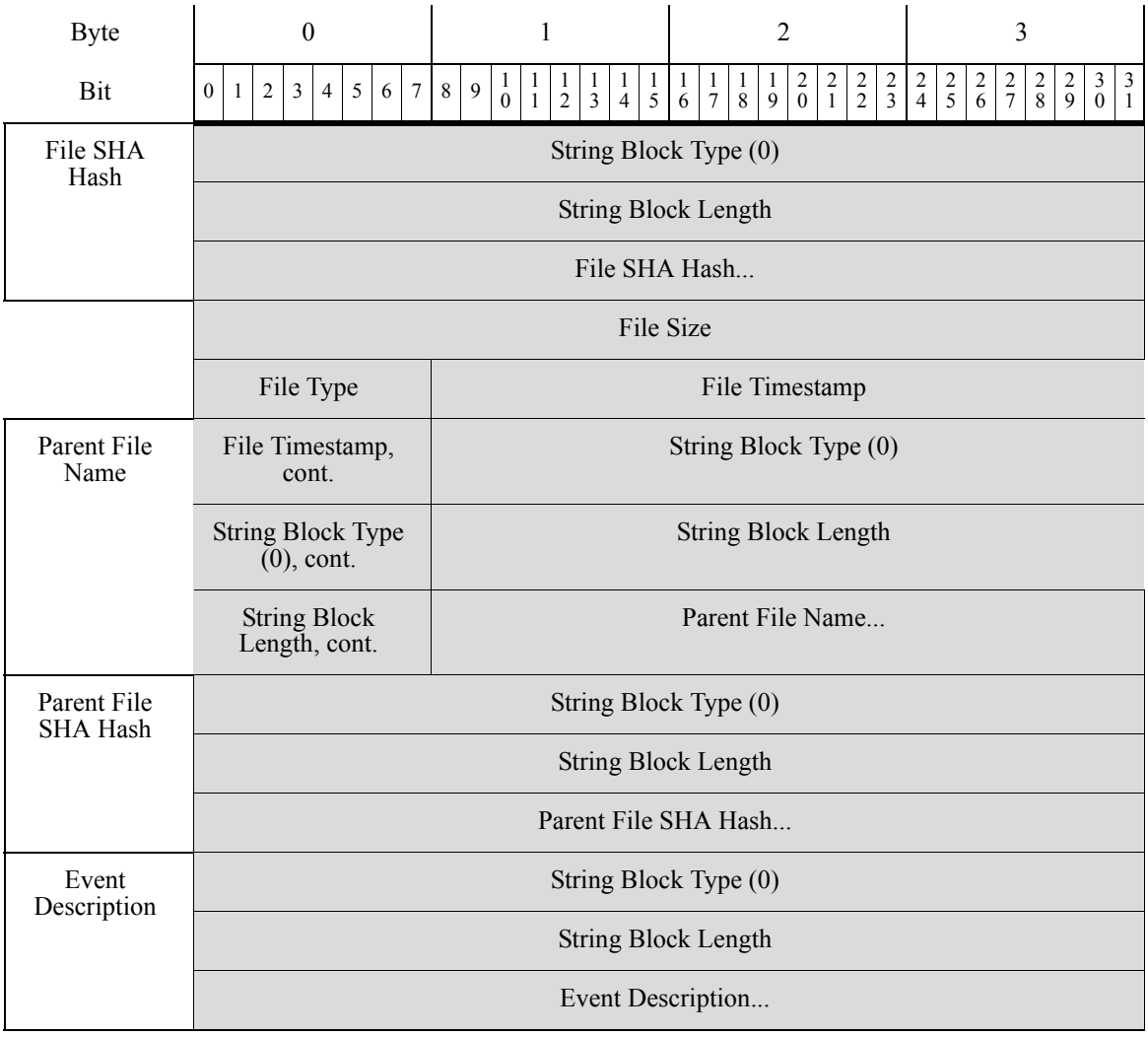
Malware Event Data Block 5.1

The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 16 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 1 and an event code of 101.

The following graphic shows the structure of the malware event data block:



	Byte	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Agent UUID																																
	Agent UUID, continued																																
	Agent UUID, continued																																
	Agent UUID, continued																																
	Cloud UUID																																
	Cloud UUID, continued																																
	Cloud UUID, continued																																
	Cloud UUID, continued																																
	Timestamp																																
	Event Type ID																																
	Event Subtype ID									Host IP Address																							
Detection Name	Host IP Address, cont.									Detector ID								String Block Type (0)															
	String Block Type (0), cont.																	String Block Length															
	String Block Length, cont.																	Detection Name...															
User	String Block Type (0)																																
	String Block Length																																
	User...																																
File Name	String Block Type (0)																																
	String Block Length																																
	File Name...																																
File Path	String Block Type (0)																																
	String Block Length																																
	File Path...																																



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															
	File Type								File Timestamp																							
Parent File Name	File Timestamp, cont.								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Parent File Name...																							
Parent File SHA Hash	String Block Type (0)																															
	String Block Length																															
	Parent File SHA Hash...																															
Event Description	String Block Type (0)																															
	String Block Length																															
	Event Description...																															

The following table describes the fields in the malware event data block.

Table B-9 Malware Event Data Block Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always `16`.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.

Table B-9 Malware Event Data Block Fields (continued)

Field	Data Type	Description
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.
Host IP Address	uint32	The host IP address associated with the malware event.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file.
File Timestamp	uint32	The creation timestamp of the detected or quarantined file.

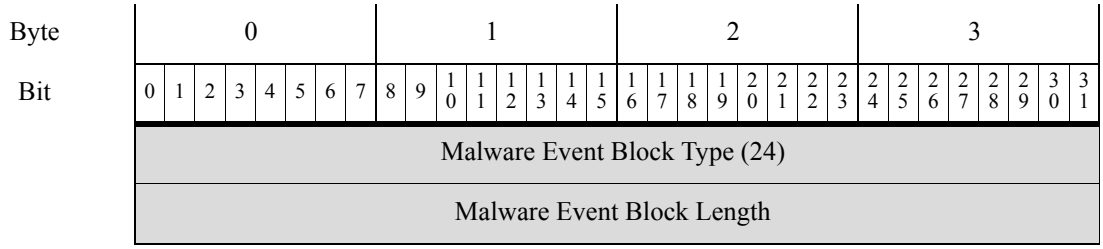
Table B-9 Malware Event Data Block Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.

Malware Event Data Block 5.1.1.x

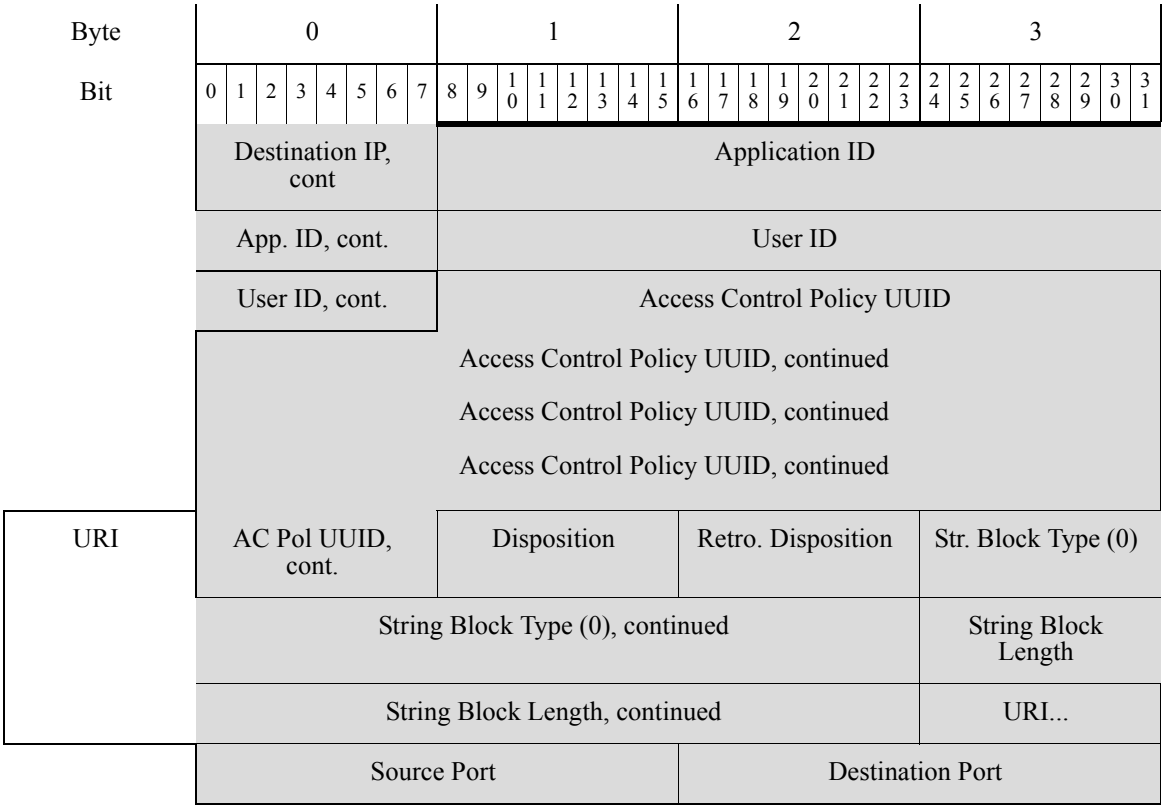
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 24 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 2 and an event code of 101.

The following graphic shows the structure of the malware event data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Agent UUID																															
	Agent UUID, continued																															
	Agent UUID, continued																															
	Agent UUID, continued																															
	Cloud UUID																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
	Event Subtype ID								Host IP Address																							
Detection Name	Host IP Address, cont.								Detector ID								String Block Type (0)															
	String Block Type (0), cont.																String Block Length															
	String Block Length, cont.																Detection Name...															
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															
File Path	String Block Type (0)																															
	String Block Length																															
	File Path...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															
	File Type								File Timestamp																							
Parent File Name	File Timestamp, cont.								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Parent File Name...																							
Parent File SHA Hash	String Block Type (0)																															
	String Block Length																															
	Parent File SHA Hash...																															
Event Description	String Block Type (0)																															
	String Block Length																															
	Event Description...																															
	Device ID																															
	Connection Instance																Connection Counter															
	Connection Event Timestamp																															
	Direction								Source IP Address																							
	Source IP Address, continued																															
	Source IP Address, continued																															
	Source IP Address, continued																															
	Source IP, cont.								Destination IP Address																							
	Destination IP Address, continued																															
	Destination IP Address, continued																															
	Destination IP Address, continued																															



The following table describes the fields in the malware event data block.

Table B-10 Malware Event Data Block for 5.1.1.x Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always `24`.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.
Host IP Address	uint32	The host IP address associated with the malware event.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always `0`.

Table B-10 Malware Event Data Block for 5.1.1.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.

Table B-10 Malware Event Data Block for 5.1.1.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.

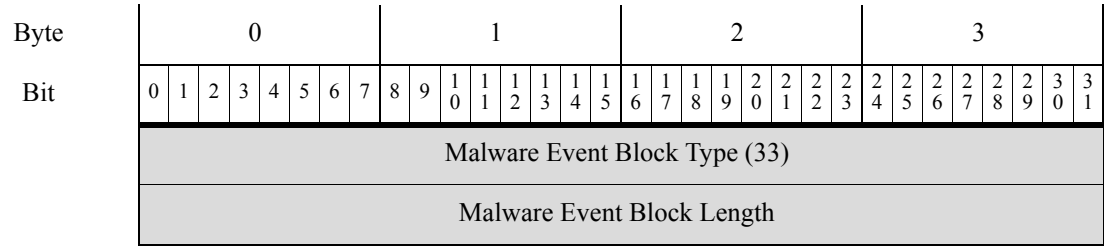
Table B-10 Malware Event Data Block for 5.1.1.x Fields (continued)

Field	Data Type	Description
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Cisco cloud for a disposition. • 5 — NO_CLOUD_RESP — The Cisco cloud services did not respond to the request.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.

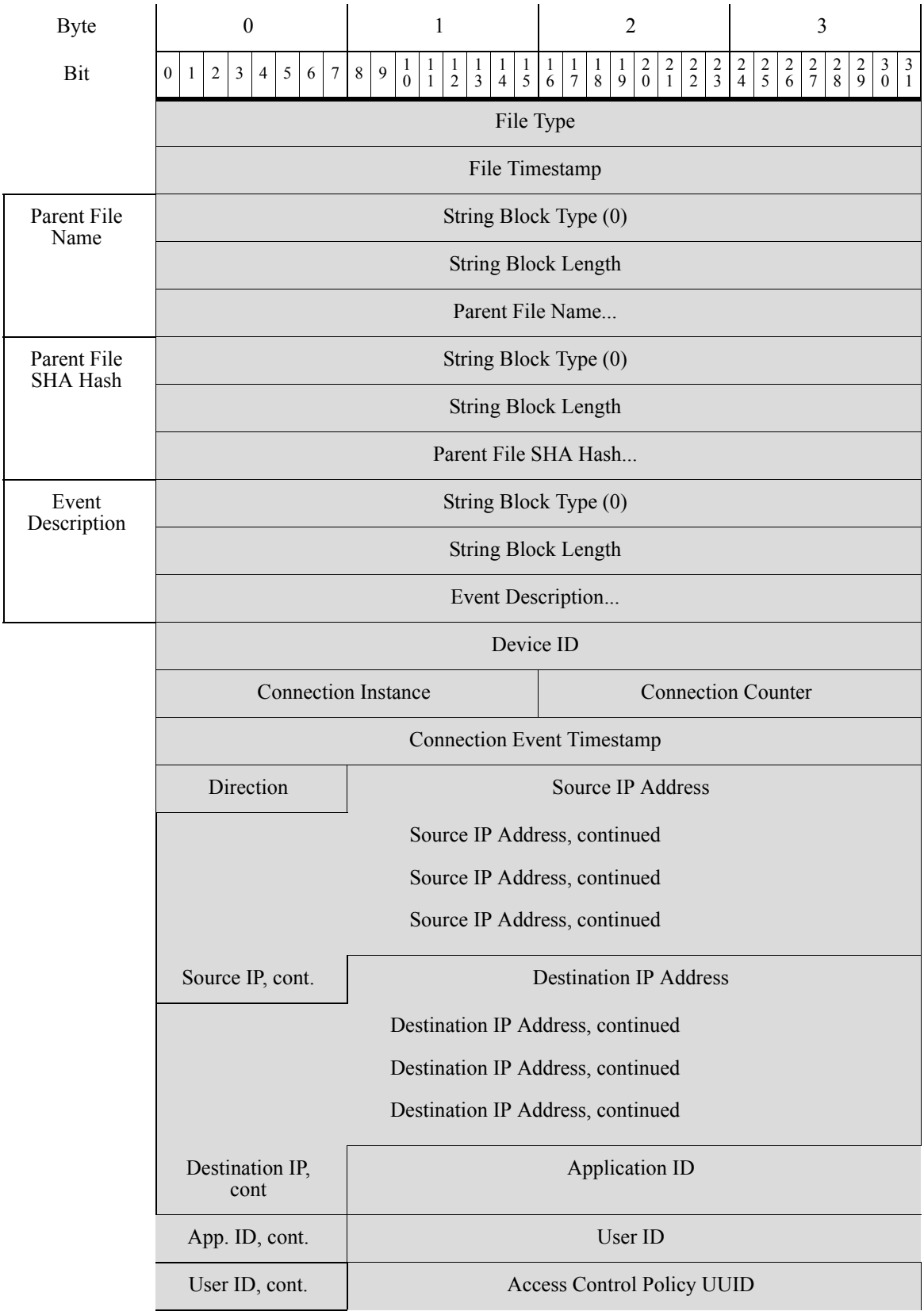
Malware Event Data Block 5.2.x

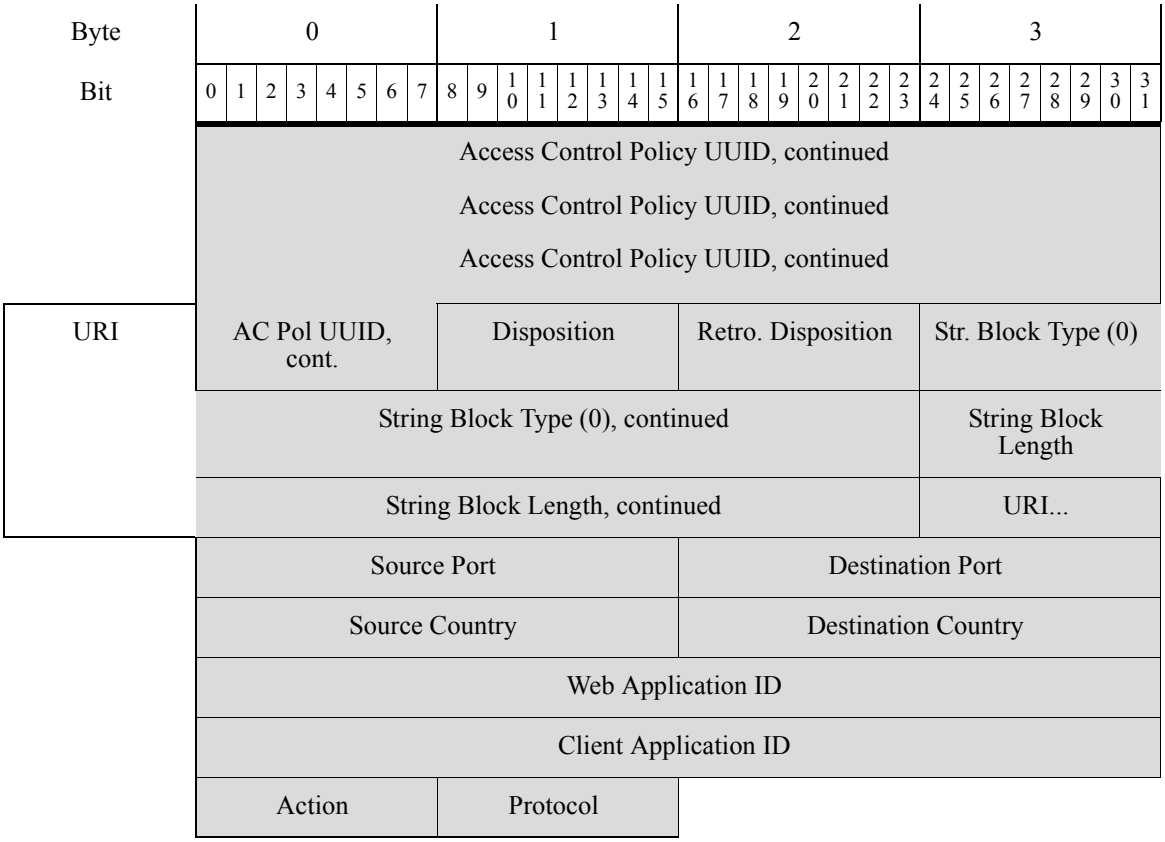
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 33 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 3 and an event code of 101.

The following graphic shows the structure of the malware event data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Agent UUID																															
	Agent UUID, continued																															
	Agent UUID, continued																															
	Agent UUID, continued																															
	Cloud UUID																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
Detection Name	Event Subtype ID								Detector ID								String Block Type (0)															
	String Block Type (0), cont.																String Block Length															
	String Block Length, cont.																Detection Name...															
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															
File Path	String Block Type (0)																															
	String Block Length																															
	File Path...																															
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
File Type
File Timestamp
Parent File Name
String Block Type (0)
String Block Length
Parent File Name...
Parent File SHA Hash
String Block Type (0)
String Block Length
Parent File SHA Hash...
Event Description
String Block Type (0)
String Block Length
Event Description...
Device ID
Connection Instance
Connection Counter
Connection Event Timestamp
Direction
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Source IP, cont.
Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Destination IP, cont
Application ID
App. ID, cont.
User ID
User ID, cont.
Access Control Policy UUID



The following table describes the fields in the malware event data block.

Table B-11 Malware Event Data Block for 5.2.x Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always `33`.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always `0`.

Table B-11 Malware Event Data Block for 5.2.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint32	The file type of the detected or quarantined file.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.

Table B-11 Malware Event Data Block for 5.2.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.

Table B-11 Malware Event Data Block for 5.2.x Fields (continued)

Field	Data Type	Description
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — NEUTRAL — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.

Table B-11 Malware Event Data Block for 5.2.x Fields (continued)

Field	Data Type	Description
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.

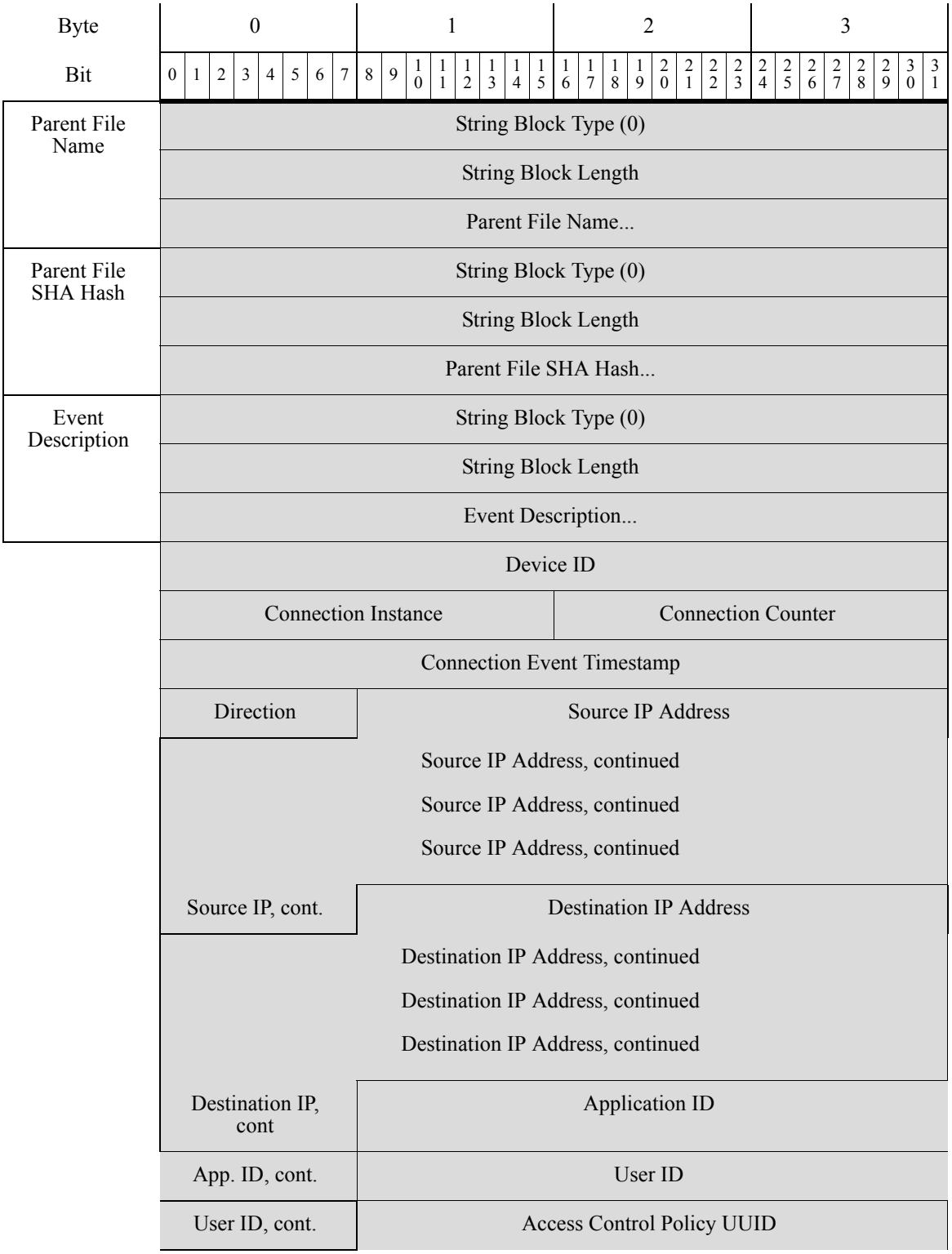
Malware Event Data Block 5.3

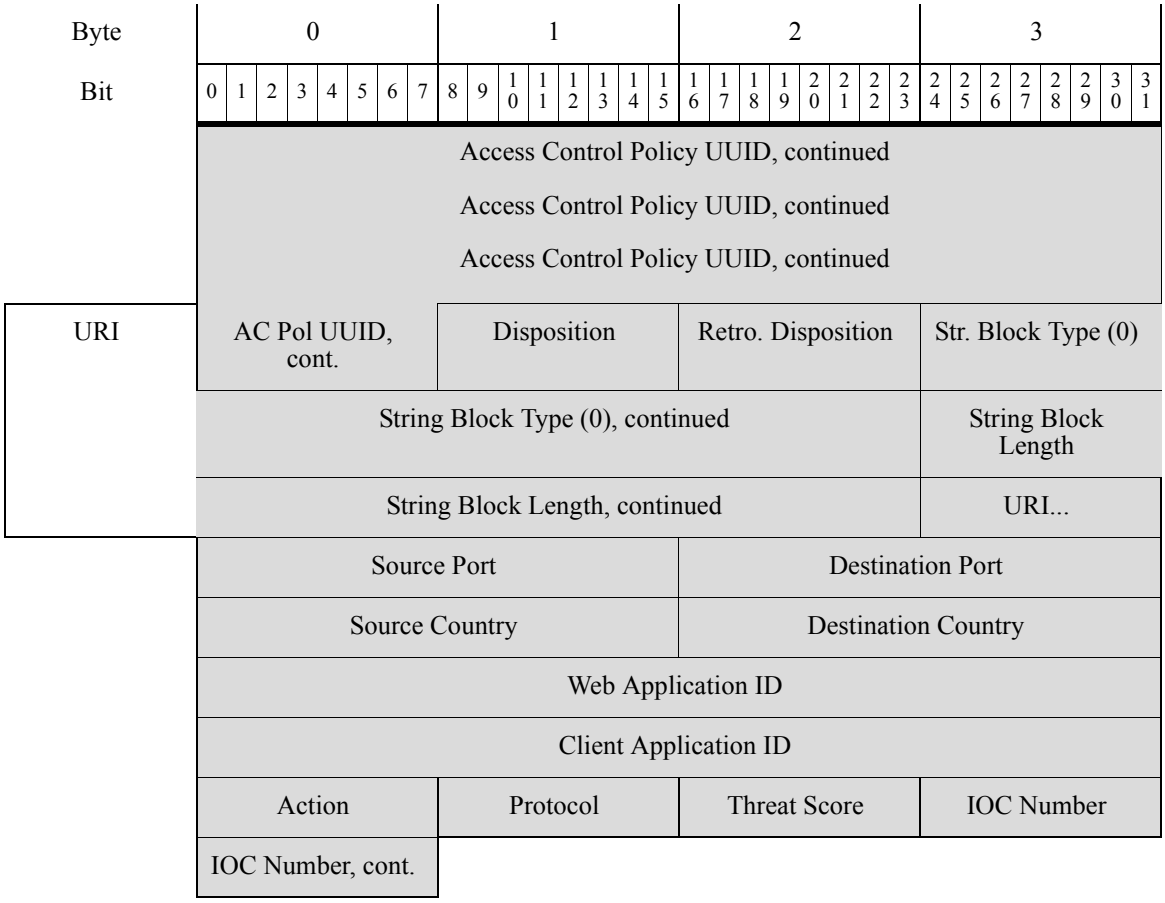
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 35 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 4 and an event code of 101.

The following graphic shows the structure of the malware event data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Malware Event Block Type (35)																																
Malware Event Block Length																																
Agent UUID																																
Agent UUID, continued																																
Agent UUID, continued																																
Agent UUID, continued																																
Cloud UUID																																
Cloud UUID, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
	Event Subtype ID																															
Detection Name	Detector ID								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Detection Name...																							
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															
File Path	String Block Type (0)																															
	String Block Length																															
	File Path...																															
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															
	File Type																															
	File Timestamp																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Parent File Name
String Block Type (0)
String Block Length
Parent File Name...
Parent File SHA Hash
String Block Type (0)
String Block Length
Parent File SHA Hash...
Event Description
String Block Type (0)
String Block Length
Event Description...
Device ID
Connection Instance
Connection Counter
Connection Event Timestamp
Direction
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Source IP, cont.
Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Destination IP, cont
Application ID
App. ID, cont.
User ID
User ID, cont.
Access Control Policy UUID



The following table describes the fields in the malware event data block.

Table B-12 Malware Event Data Block for 5.3 Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always `35`.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint32	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.

Table B-12 Malware Event Data Block for 5.3 Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint32	The file type of the detected or quarantined file. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.

Table B-12 Malware Event Data Block for 5.3 Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.

Table B-12 Malware Event Data Block for 5.3 Fields (continued)

Field	Data Type	Description
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.

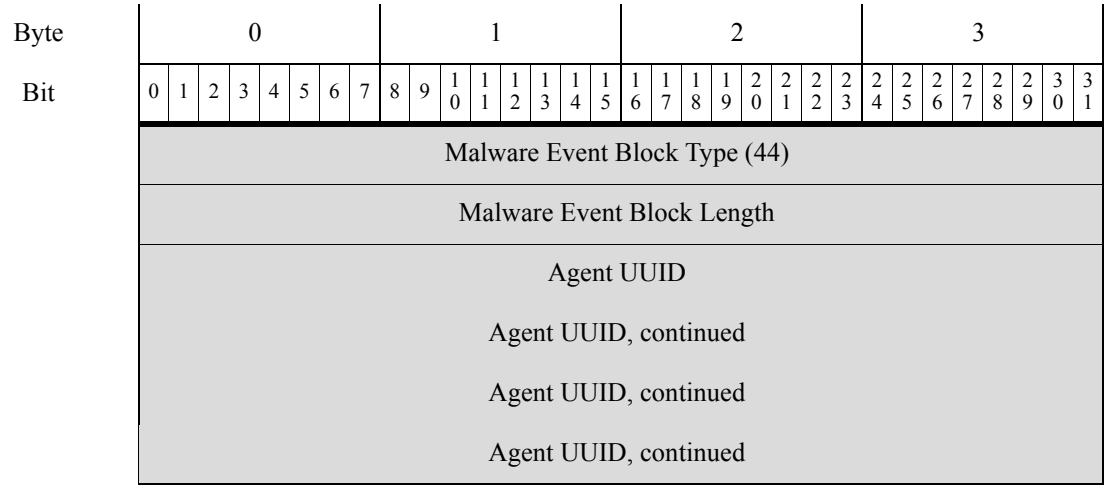
Table B-12 Malware Event Data Block for 5.3 Fields (continued)

Field	Data Type	Description
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
IOC Number	uint16	ID Number of the compromise associated with this event.

Malware Event Data Block 5.3.1

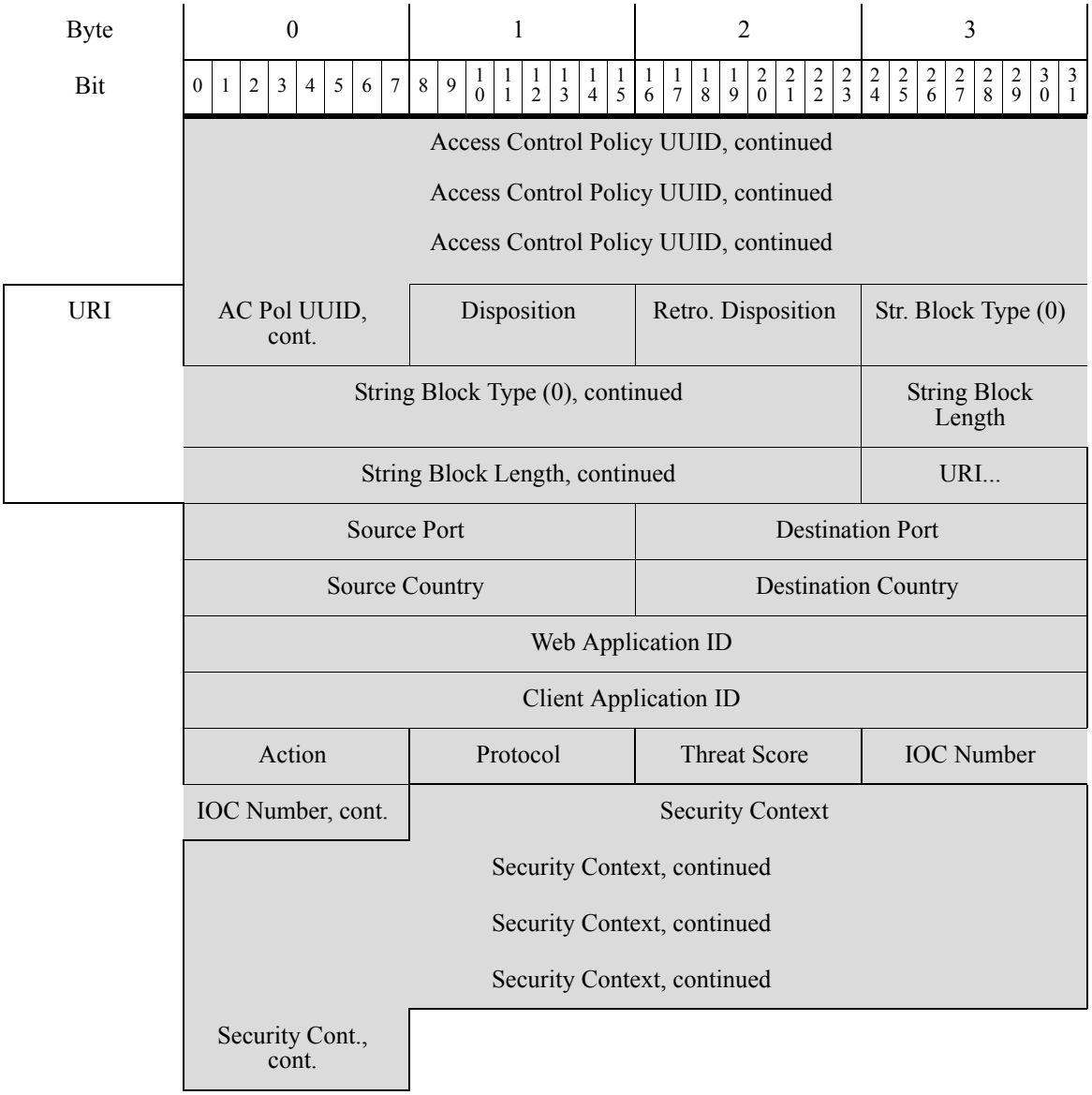
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 44 in the series 2 group of blocks. It supersedes block 35. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 5 and an event code of 101.

The following graphic shows the structure of the malware event data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Cloud UUID																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
	Event Subtype ID																															
Detection Name	Detector ID								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Detection Name...																							
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															
File Path	String Block Type (0)																															
	String Block Length																															
	File Path...																															
File SHA Hash	String Block Type (0)																															
	String Block Length																															
	File SHA Hash...																															
	File Size																															
	File Type																															
	File Timestamp																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Parent File Name	String Block Type (0)																															
	String Block Length																															
	Parent File Name...																															
Parent File SHA Hash	String Block Type (0)																															
	String Block Length																															
	Parent File SHA Hash...																															
Event Description	String Block Type (0)																															
	String Block Length																															
	Event Description...																															
	Device ID																															
	Connection Instance																Connection Counter															
	Connection Event Timestamp																															
	Direction								Source IP Address																							
	Source IP Address, continued																															
	Source IP Address, continued																															
	Source IP Address, continued																															
	Source IP, cont.								Destination IP Address																							
	Destination IP Address, continued																															
	Destination IP Address, continued																															
	Destination IP Address, continued																															
	Destination IP, cont								Application ID																							
	App. ID, cont.								User ID																							
	User ID, cont.								Access Control Policy UUID																							



The following table describes the fields in the malware event data block.

Table B-13 Malware Event Data Block for 5.3.1 Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always `44`.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the Cisco Advanced Malware Protection cloud from which the malware event originated.

Table B-13 Malware Event Data Block for 5.3.1 Fields (continued)

Field	Data Type	Description
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint32	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.

Table B-13 Malware Event Data Block for 5.3.1 Fields (continued)

Field	Data Type	Description
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint32	The file type of the detected or quarantined file. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.

Table B-13 Malware Event Data Block for 5.3.1 Fields (continued)

Field	Data Type	Description
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.

Table B-13 Malware Event Data Block for 5.3.1 Fields (continued)

Field	Data Type	Description
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.

Malware Event Data Block 5.4.x

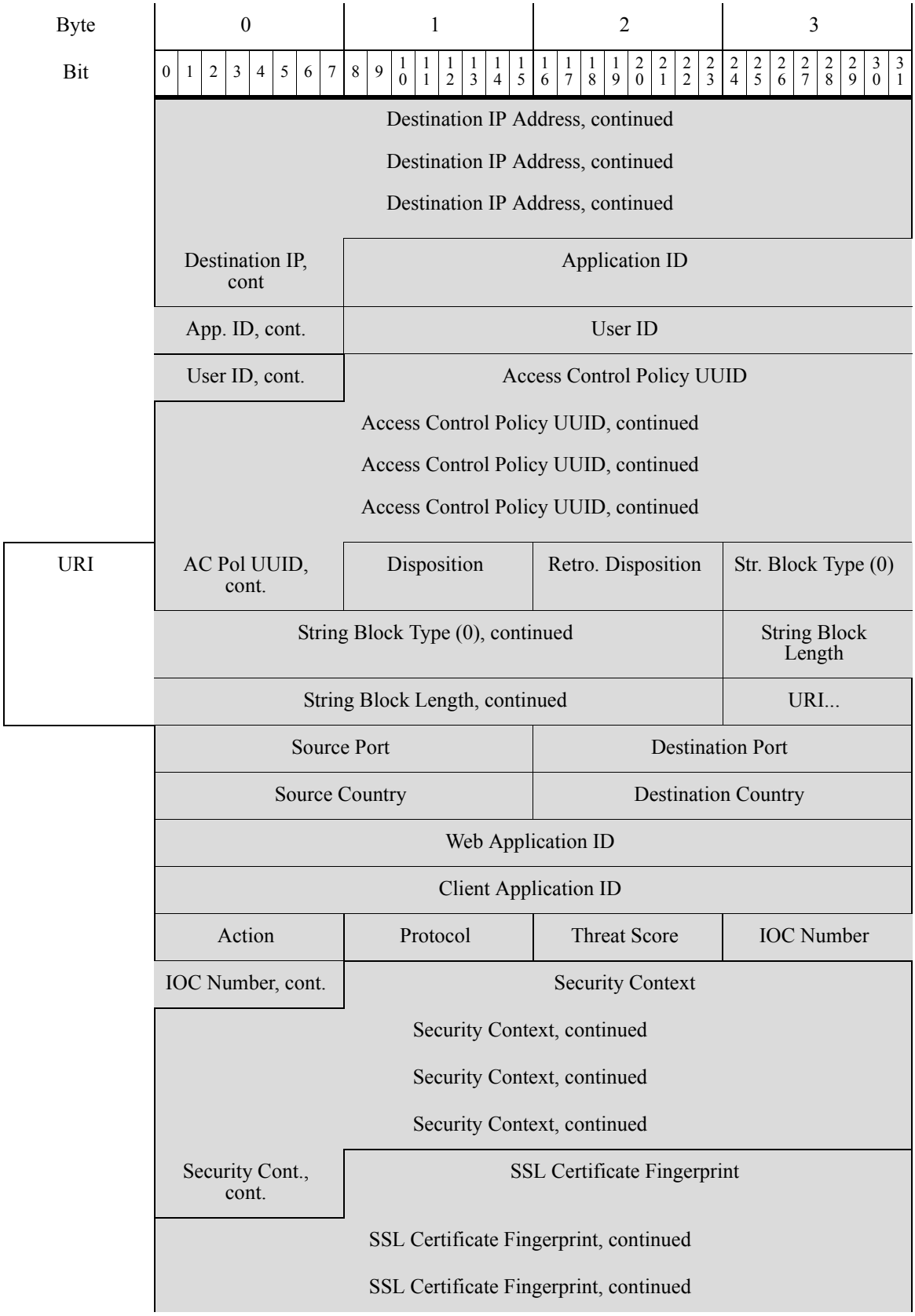
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 47 in the series 2 group of blocks. It supersedes block 44 and is superseded by block . Fields for SSL and file archive support have been added.

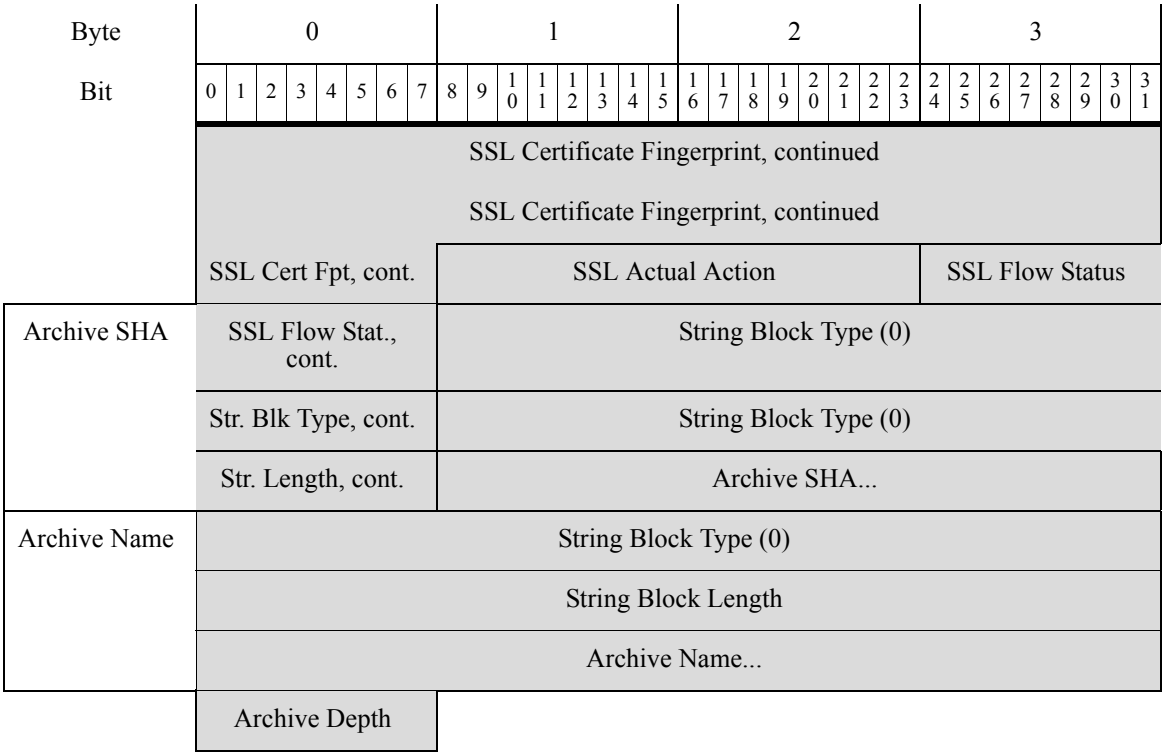
You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 6 and an event code of 101.

The following graphic shows the structure of the malware event data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Malware Event Block Type (47)																															
	Malware Event Block Length																															
	Agent UUID Agent UUID, continued Agent UUID, continued Agent UUID, continued																															
	Cloud UUID Cloud UUID, continued Cloud UUID, continued Cloud UUID, continued																															
	Malware Event Timestamp																															
	Event Type ID																															
	Event Subtype ID																															
Detection Name	Detector ID								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								Detection Name...																							
User	String Block Type (0)																															
	String Block Length																															
	User...																															
File Name	String Block Type (0)																															
	String Block Length																															
	File Name...																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
File Path
String Block Type (0)
String Block Length
File Path...
File SHA Hash
String Block Type (0)
String Block Length
File SHA Hash...
File Size
File Type
File Timestamp
Parent File Name
String Block Type (0)
String Block Length
Parent File Name...
Parent File SHA Hash
String Block Type (0)
String Block Length
Parent File SHA Hash...
Event Description
String Block Type (0)
String Block Length
Event Description...
Device ID
Connection Instance
Connection Counter
Connection Event Timestamp
Direction
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Source IP, cont.
Destination IP Address


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Destination IP, cont
Application ID
App. ID, cont.
User ID
User ID, cont.
Access Control Policy UUID
Access Control Policy UUID, continued
Access Control Policy UUID, continued
Access Control Policy UUID, continued
URI
AC Pol UUID, cont.
Disposition
Retro. Disposition
Str. Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
URI...
Source Port
Destination Port
Source Country
Destination Country
Web Application ID
Client Application ID
Action
Protocol
Threat Score
IOC Number
IOC Number, cont.
Security Context
Security Context, continued
Security Context, continued
Security Context, continued
Security Cont., cont.
SSL Certificate Fingerprint
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued



The following table describes the fields in the malware event data block.

Table B-14 Malware Event Data Block for 5.4.x Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 47.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the AMP for Endpoints agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the Cisco Advanced Malware Protection cloud from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint32	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always `0`.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Cisco Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint32	The file type of the detected or quarantined file. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist • 6 — Cloud Lookup Timeout • 7 — Custom Detection • 8 — Custom Detection Block • 9 — Archive Block (Depth Exceeded) • 10 — Archive Block (Encrypted) • 11 — Archive Block (Failed to Inspect)
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
IOC Number	uint16	ID number of the compromise associated with this event.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
String Block Type	uint32	Initiates a String data block containing the Archive SHA. This value is always 0.

Table B-14 Malware Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Archive SHA String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive SHA	string	SHA1 hash of the parent archive in which the file is contained.
String Block Type	uint32	Initiates a String data block containing the Archive Name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Archive Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive Name	string	Name of the parent archive.
Archive Depth	uint8	Number of layers in which the file is nested. For example, if a text file is in a zip archive, this has a value of `1`.

Legacy Discovery Data Structures

Legacy Discovery Event Header

Discovery Event Header 5.0 - 5.1.1.x

Discovery and connection event messages contain a discovery event header. It conveys the type and subtype of the event, the time the event occurred, the device on which the event occurred, and the structure of the event data in the message. This header is followed by the actual host discovery, user, or connection event data. The structures associated with the different event type/subtype values are described in Host Discovery Structures by Event Type, page 4-38.

The event type and event subtype fields of the discovery event header identify the structure of the transmitted event message. Once the structure of the event data block is determined, your program can parse the message appropriately.

The shaded rows in the following diagram illustrate the format of the discovery event header.

	Byte	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																	Message Type (4)															
	Message Length																																
	Netmap ID																	Record Type															
	Record Length																																
	eStreamer Server Timestamp (in events, only if bit 23 is set)																																
	Reserved for Future Use (in events, only if bit 23 is set)																																
Discovery Event Header	Device ID																																
	IP Address																																
	MAC Address																																
	MAC Address, continued																	Reserved for future use															
	Event Second																																
	Event Microsecond																																
	Reserved (Internal)									Event Type																							
	Event Subtype																																
	File Number (Internal Use Only)																																
	File Position (Internal Use Only)																																

The following table describes the discovery event header.

Table B-15 Discovery Event Header Fields

Field	Data Types	Description
Device ID	uint32	ID number of the device that generated the discovery event. You can obtain the metadata for the device by requesting Version 3 and 4 metadata. See Managed Device Record Metadata, page 3-34 for more information.
IP Address	uint32	IP address of the host involved in the event.
MAC Address	uint8[6]	MAC address of the host involved in the event.
Reserved for future use	byte[2]	Two bytes of padding with values set to `0`.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) that the system generated the event.

Table B-15 Discovery Event Header Fields (continued)

Field	Data Types	Description
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the system generated the event.
Reserved (Internal)	byte	Internal data from Cisco and can be disregarded.
Event Type	uint32	Event type (`1000` for new events, `1001` for change events, `1002` for user input events, 1050 for full host profile). See Host Discovery Structures by Event Type, page 4-38 for a list of available event types.
Event Subtype	uint32	Event subtype. See Host Discovery Structures by Event Type, page 4-38 for a list of available event subtypes.
File Number	byte[4]	Serial file number. This field is for Cisco internal use and can be disregarded.
File Position	byte[4]	Event's position in the serial file. This field is for Cisco internal use and can be disregarded.

Legacy Server Data Blocks

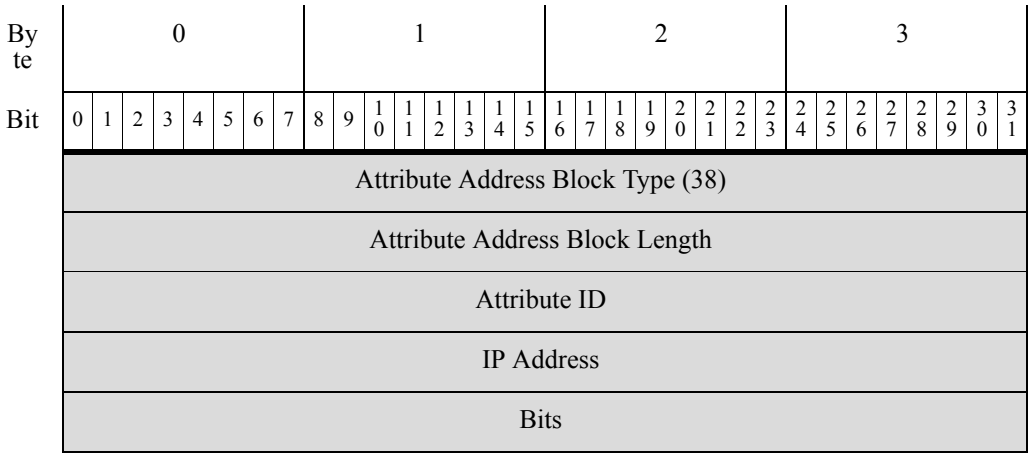
For more information, see the following sections:

- Attribute Address Data Block for 5.0 - 5.1.1.x, page B-90

Attribute Address Data Block for 5.0 - 5.1.1.x

The Attribute Address data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 38.

The following diagram shows the basic structure of an Attribute Address data block:



The following table describes the fields of the Attribute Address data block.

Table B-16 Attribute Address Data Block Fields

Field	Data Type	Description
Attribute Address Block Type	uint32	Initiates an Attribute Address data block. This value is always 38.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the attribute address block type and length, plus the number of bytes in the attribute address data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
IP Address	uint8[4]	IP address of the host, if the address was automatically assigned, in IP address octets.
Bits	uint32	Contains the significant bits used to calculate the netmask if an IP address was automatically assigned.

Legacy Client Application Data Blocks

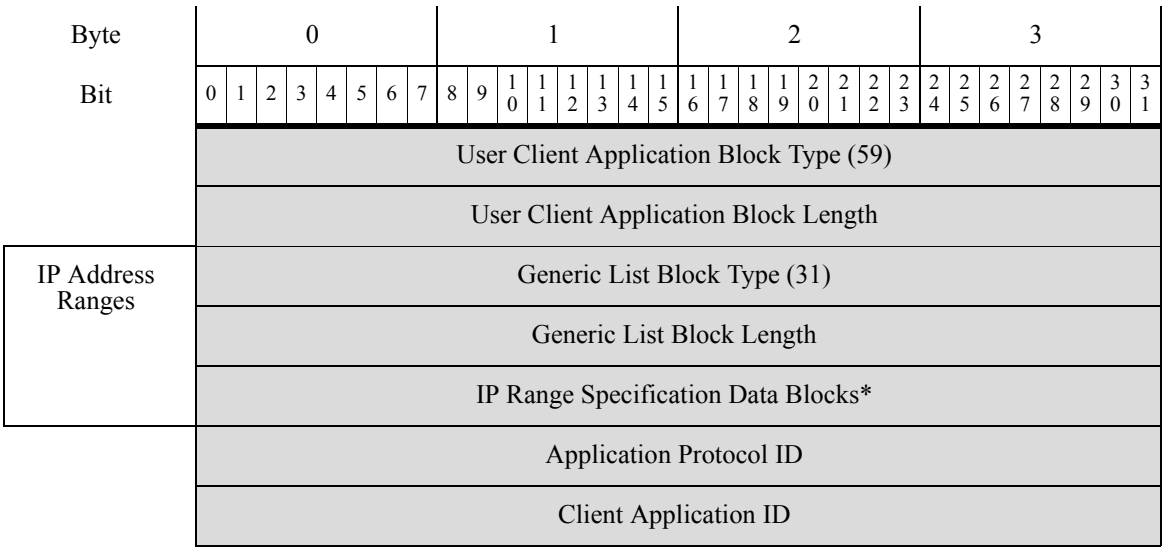
For more information, see the following sections:

- User Client Application Data Block for 5.0 - 5.1, page B-91

User Client Application Data Block for 5.0 - 5.1

The User Client Application data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of IP address range data blocks. The User Client Application data block has a block type of 59.

The following diagram shows the basic structure of a User Client Application data block:



Version	String Block Type (0)
	String Block Length
	Version...

The following table describes the fields of the User Client Application data block.

Table B-17 User Client Application Data Block Fields

Field	Number of Bytes	Description
User Client Application Block Type	uint32	Initiates a User Client Application data block. This value is always .
User Client Application Block Length	uint32	Total number of bytes in the User Client Application data block, including eight bytes for the user client application block type and length fields, plus the number of bytes of user client application data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See Table 4-55User Server Data Block Fields, page 4-95 for a description of this data block.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block that contains the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Client application version.

Legacy Scan Result Data Blocks

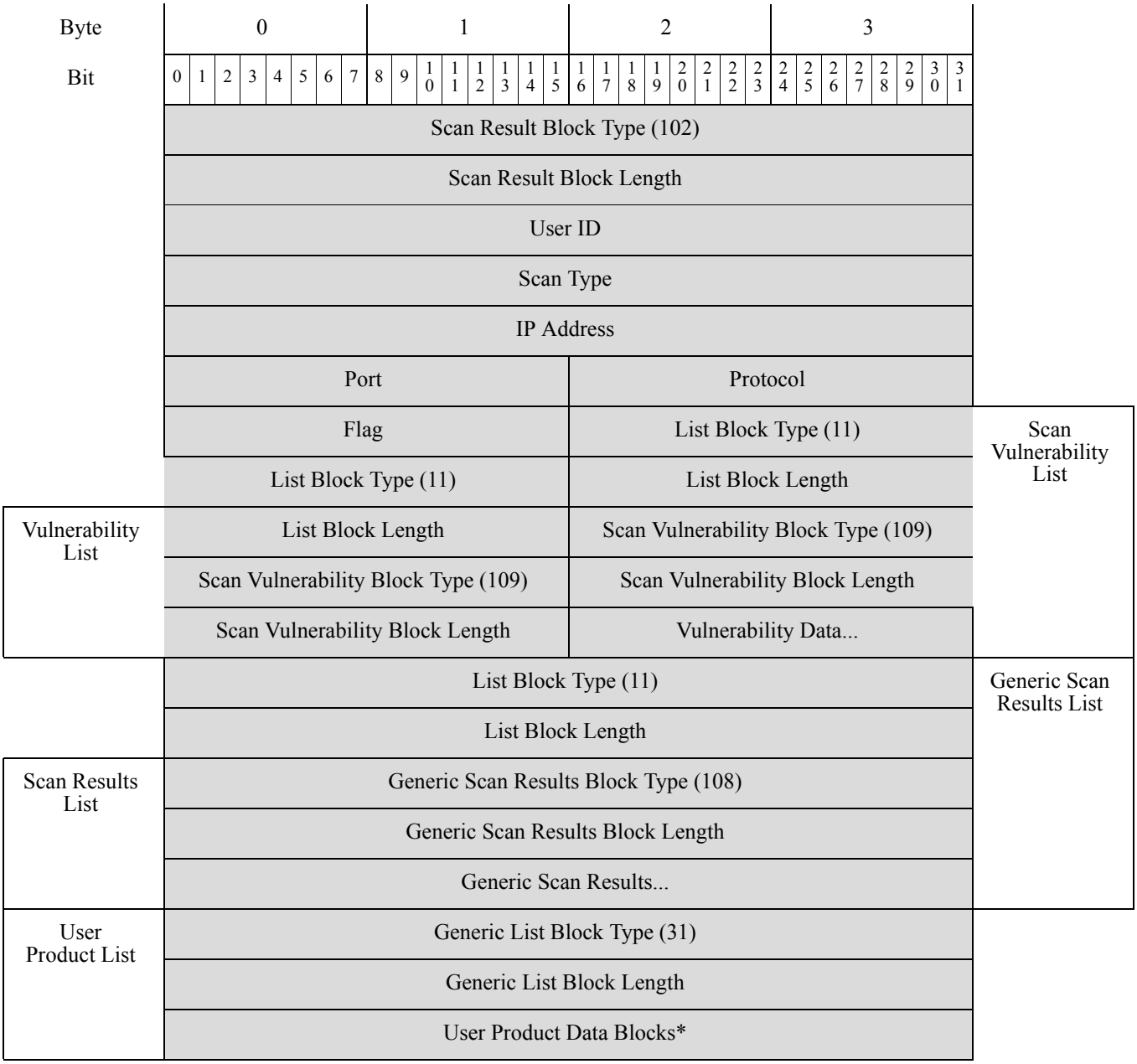
For more information, see the following sections:

- Scan Result Data Block 5.0 - 5.1.1.x, page B-93
- User Product Data Block for 5.0.x, page B-95
- User Information Data Block for 5.x, page B-105

Scan Result Data Block 5.0 - 5.1.1.x

The Scan Result data block describes a vulnerability and is used within Add Scan Result events (event type 1002, subtype 11). The Scan Result data block has a block type of 102.

The following diagram shows the format of a Scan Result data block:



The following table describes the fields of the Scan Result data block.

Table B-18 Scan Result Data Block Fields

Field	Data Type	Description
Scan Result Block Type	uint32	Initiates a Scan Result data block. This value is always 102.
Scan Result Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
User ID	uint32	Contains the user identification number for the user who imported the scan result or ran the scan that produced the scan result.
Scan Type	uint32	Indicates how the results were added to the system.
IP Address	uint32	IP address of the host affected by the vulnerabilities in the result, in IP address octets.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.
Protocol	uint16	IANA protocol number. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP
Flag	uint16	Reserved
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block describing a vulnerability detected during a scan. This value is always 109.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes in the scan vulnerability data that follows.
Vulnerability Data	string	Information relating to each vulnerability.
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Generic Scan Results Block Type	uint32	Initiates a Generic Scan Results data block describing server and operating system data detected during a scan. This value is always 108.

Table B-18 Scan Result Data Block Fields (continued)

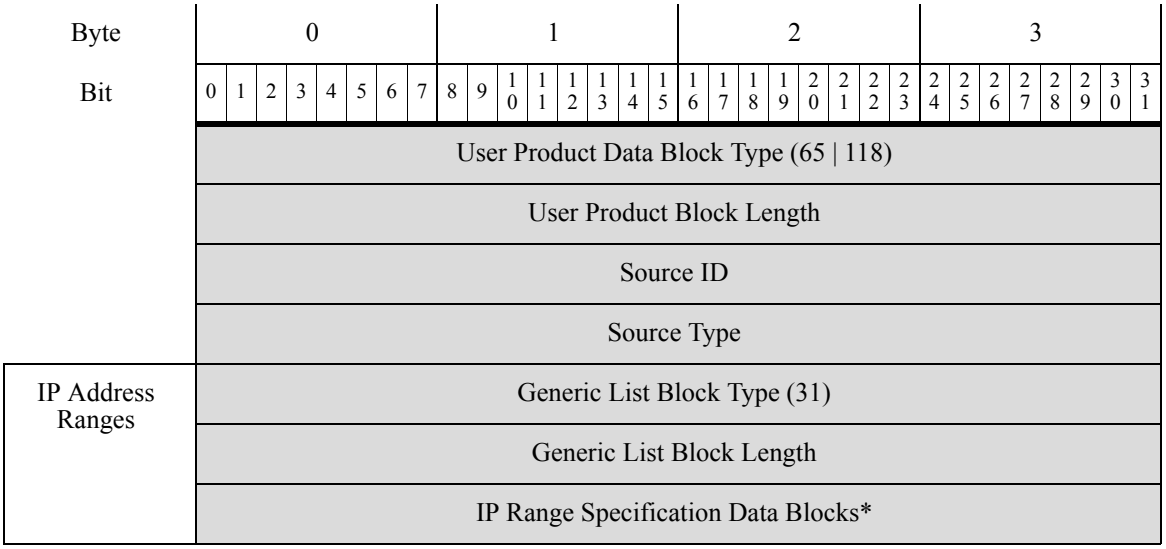
Field	Data Type	Description
Generic Scan Results Block Length	uint32	Number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes in the scan result data that follows.
Generic Scan Results Data	string	Information relating to each scan result.
Generic List Block Type	uint32	Initiates a Generic List data block comprising User Product data blocks conveying host input data from a third party application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated User Product data blocks.
User Product Data Blocks *	variable	User Product data blocks containing host input data. See User Product Data Block 5.1+, page 4-160 for a description of this data block.

User Product Data Block for 5.0.x

The User Product data block conveys host input data imported from a third party application, including third party application string mappings. This data block is used in The following table describes the fields of the Connection Statistics data block for 6.0+., page 4-116. The User Product data block has a block type of 65 for 4.10.x, and a block type of 118 for 5.0 - 5.0.x. The block types have the same structure.

Note An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the User Product data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Port																Protocol															
	Drop User Product																															
Custom Vendor String	String Block Type (0)																															
	String Block Length																															
	Custom Vendor String...																															
Custom Product String	String Block Type (0)																															
	String Block Length																															
	Custom Product String...																															
Custom Version String	String Block Type (0)																															
	String Block Length																															
	Custom Version String...																															
	Software ID																															
	Server ID																															
	Vendor ID																															
	Product ID																															
Major Version String	String Block Type (0)																															
	String Block Length																															
	Major Version String...																															
Minor Version String	String Block Type (0)																															
	String Block Length																															
	Minor Version String...																															
Revision String	String Block Type (0)																															
	String Block Length																															
	Revision String...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
To Major String	String Block Type (0)																															
	String Block Length																															
	To Major Version String...																															
To Minor String	String Block Type (0)																															
	String Block Length																															
	To Minor Version String...																															
To Revision String	String Block Type (0)																															
	String Block Length																															
	To Revision String...																															
Build String	String Block Type (0)																															
	String Block Length																															
	Build String...																															
Patch String	String Block Type (0)																															
	String Block Length																															
	Patch String...																															
Extension String	String Block Type (0)																															
	String Block Length																															
	Extension String...																															
OS UUID	Operating System UUID																															
	Operating System UUID cont.																															
	Operating System UUID cont.																															
	Operating System UUID cont.																															
List of Fixes	Generic List Block Type (31)																															
	Generic List Block Length																															
	Fix List Data Blocks*																															

The following table describes the components of the User Product data block.

Table B-19 User Product Data Block Fields for 4.10.x, 5.0-5.0.x

Field	Data Type	Description
User Product Data Block Type	uint32	Initiates a User Product data block. This value is `65` for version 4.10.x and `118` for version 5.0 - 5.0.x.
User Product Block Length	uint32	Total number of bytes in the User Product data block, including eight bytes for the user product block type and length fields, plus the number of bytes in the user product data that follows.
Source ID	uint32	Identification number of the source that imported the data.
Source Type	uint32	The source type of the source that supplied the data.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+, page 4-87 for a description of this data block.
Port	uint16	Port specified by the user.
Protocol	uint16	IANA protocol number specified by the user. For example: • `1` — ICMP • `4` — IP • `6` — TCP • `17` — UDP
Drop User Product	uint32	Indicates whether the user OS definition was deleted from the host: • `0` — No • `1` — Yes
String Block Type	uint32	Initiates a String data block containing the custom vendor name specified in the user input. This value is always `0`.
String Block Length	uint32	Number of bytes in the custom vendor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.
Custom Vendor Name	string	The custom vendor name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom product name specified in the user input. This value is always `0`.
String Block Length	uint32	Number of bytes in the custom product String data block, including eight bytes for the block type and length fields, plus the number of bytes in the product name.
Custom Product Name	string	The custom product name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom version specified in the user input. This value is always `0`.

Table B-19 User Product Data Block Fields for 4.10.x, 5.0-5.0.x (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the custom version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Custom Version	string	The custom version specified in the user input.
Software ID	uint32	The identifier for a specific revision of a server or operating system in the Cisco database.
Server ID	uint32	The Cisco application identifier for the application protocol on the host server specified in user input.
Vendor ID	uint32	The identifier for the vendor of a third party operating system specified when the third party operating system is mapped to a Cisco 3D operating system definition.
Product ID	uint32	The product identification string of a third party operating system string specified when the third party operating system string is mapped to a Cisco 3D operating system definition.
String Block Type	uint32	Initiates a String data block containing the major version number of the Cisco 3D operating system definition that a third party operating system string in the user input is mapped to. This value is always `0`.
String Block Length	uint32	Number of bytes in the major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Major Version	string	Major version of the Cisco 3D operating system definition that a third party operating system string is mapped to.
String Block Type	uint32	Initiates a String data block containing the minor version number of the Cisco 3D operating system definition that a third party operating system string is mapped to. This value is always `0`.
String Block Length	uint32	Number of bytes in the minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Minor Version	string	Minor version number of the Cisco 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the revision number of the Cisco operating system definition that a third party operating system string in the user input is mapped to. This value is always `0`.
String Block Length	uint32	Number of bytes in the revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
Revision	string	Revision number of the Cisco 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last major version of the Cisco 3D operating system definition that a third party operating system string is mapped to. This value is always `0`.

Table B-19 User Product Data Block Fields for 4.10.x, 5.0-5.0.x (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the To Major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Major	string	Last version number in a range of major version numbers of the Cisco 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last minor version of the Cisco 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Minor	string	Last version number in a range of minor version numbers of the Cisco 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the Last revision number of the Cisco 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
To Revision	string	Last revision number in a range of revision numbers of the Cisco 3D operating system definitions that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the build number of the Cisco 3D operating system that the third party operating system string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Build	string	Build number of the Cisco 3D operating system that the third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the patch number of the Cisco 3D operating system that the third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the patch String data block, including eight bytes for the block type and length fields, plus the number of bytes in the patch number.
Patch	string	Patch number of the Cisco 3D operating system that the third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the extension number of the Cisco 3D operating system that the third party operating system string is mapped. This value is always 0.

Table B-19 User Product Data Block Fields for 4.10.x, 5.0-5.0.x (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the extension String data block, including eight bytes for the block type and length fields, plus the number of bytes in the extension number.
Extension	string	Extension number of the Cisco 3D operating system that the third party operating system string in the user input is mapped to.
UUID	uint8 [x16]	Contains the unique identification number for the operating system.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Fix List data blocks conveying user input data regarding what fixes have been applied to hosts in the specified IP address ranges. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Fix List data blocks.
Fix List Data Blocks *	variable	Fix List data blocks containing information about fixes applied to the hosts. See Fix List Data Block, page 4-94 for a description of this data block.

Legacy User Login Data Blocks

See the following sections for more information:

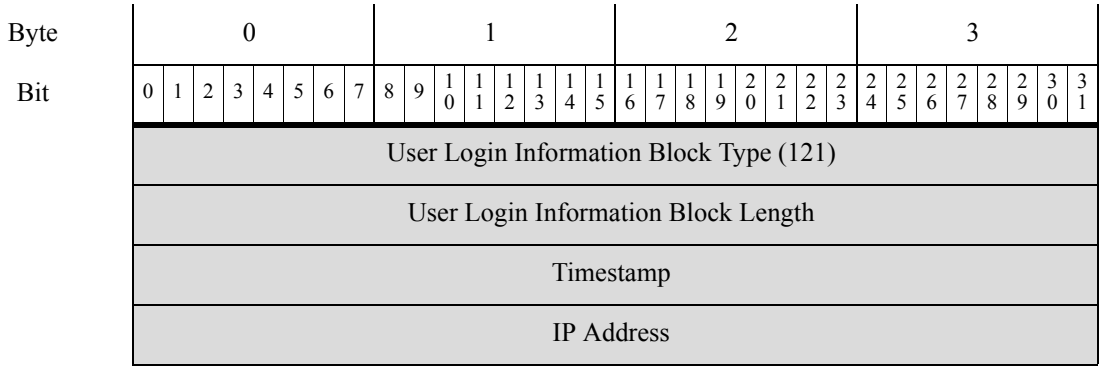
- User Login Information Data Block for 5.0 - 5.0.2, page B-101
- User Login Information Data Block 5.1-5.4.x, page B-103
- User Information Data Block for 5.x, page B-105

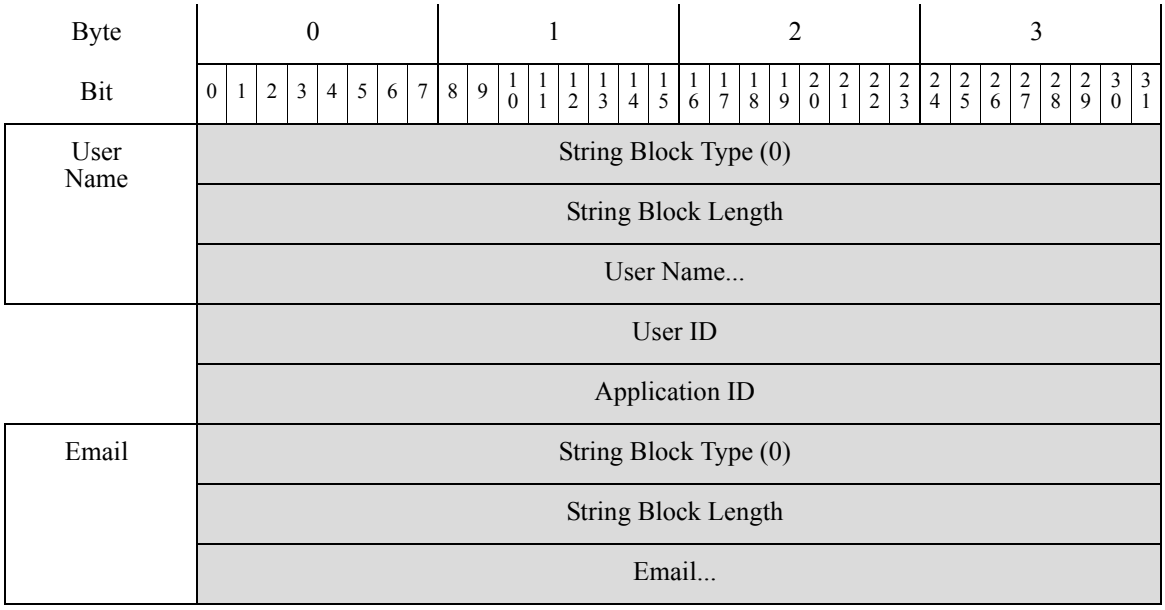
User Login Information Data Block for 5.0 - 5.0.2

The User Login Information data block is used in User Information Update messages and conveys changes in login information for a detected user. For more information, see User Information Update Message Block, page 4-55.

The User Login Information data block has a block type of 121 for version 5.0 - 5.0.2.

The graphic below shows the format of the User Login Information data block:





The following table describes the components of the User Login Information data block.

Table B-20 User Login Information Data Block Fields 5.0 - 5.0.2

Field	Data Type	Description
User Login Information Block Type	uint32	Initiates a User Login Information data block. This value is `121` for version 5.0 - 5.0.2.
User Login Information Block Length	uint32	Total number of bytes in the User Login Information data block, including eight bytes for the user login information block type and length fields, plus the number of bytes in the user login information data that follows.
Timestamp	uint32	Timestamp of the event.
IP Address	uint8[4]	IP address from the host where the user was detected logging in, in IP address octets.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The user name for the user.
User ID	uint32	Identification number of the user.
Application ID	uint32	The application ID for the application protocol used in the connection that the login information was derived from.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always `0`.

Table B-20 User Login Information Data Block Fields 5.0 - 5.0.2 (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.

User Login Information Data Block 5.1-5.4.x

The User Login Information data block is used in User Information Update messages and conveys changes in login information for a detected user. For more information, see User Account Update Message Data Block, page 4-168.

The User Login Information data block has a block type of 73 for version 4.7 - 4.10.x, a block type of 121 in the series 1 group of blocks for version 5.0 - 5.0.2, and a block type of 127 in the series 1 group of blocks for version 5.1-5.4.x.

The graphic below shows the format of the User Login Information data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Login Information Block Type (127)																															
	User Login Information Block Length																															
	Timestamp																															
	IPv4 Address																															
User Name	String Block Type (0)																															
	String Block Length																															
	User Name...																															
	User ID																															
	Application ID																															
Email	String Block Type (0)																															
	String Block Length																															
	Email...																															
	IPv6 Address IPv6 Address, continued IPv6 Address, continued																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IPv6 Address, continued																																
Reported By	Login Type								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length								Reported By...																							

The following table describes the components of the User Login Information data block.

Table B-21 User Login Information Data Block Fields

Field	Data Type	Description
User Login Information Block Type	uint32	Initiates a User Login Information data block. This value is `127` for version 5.1+.
User Login Information Block Length	uint32	Total number of bytes in the User Login Information data block, including eight bytes for the user login information block type and length fields, plus the number of bytes in the user login information data that follows.
Timestamp	uint32	Timestamp of the event.
IPv4 Address	uint32	This field is reserved but no longer populated. The IPv4 address is stored in the IPv6 Address field. See IP Addresses, page 1-6 for more information.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The user name for the user.
User ID	uint32	Identification number of the user.
Application ID	uint32	The application ID for the application protocol used in the connection that the login information was derived from.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
IPv6 Address	uint8[16]	IPv6 address from the host where the user was detected logging in, in IP address octets.

Table B-21 User Login Information Data Block Fields (continued)

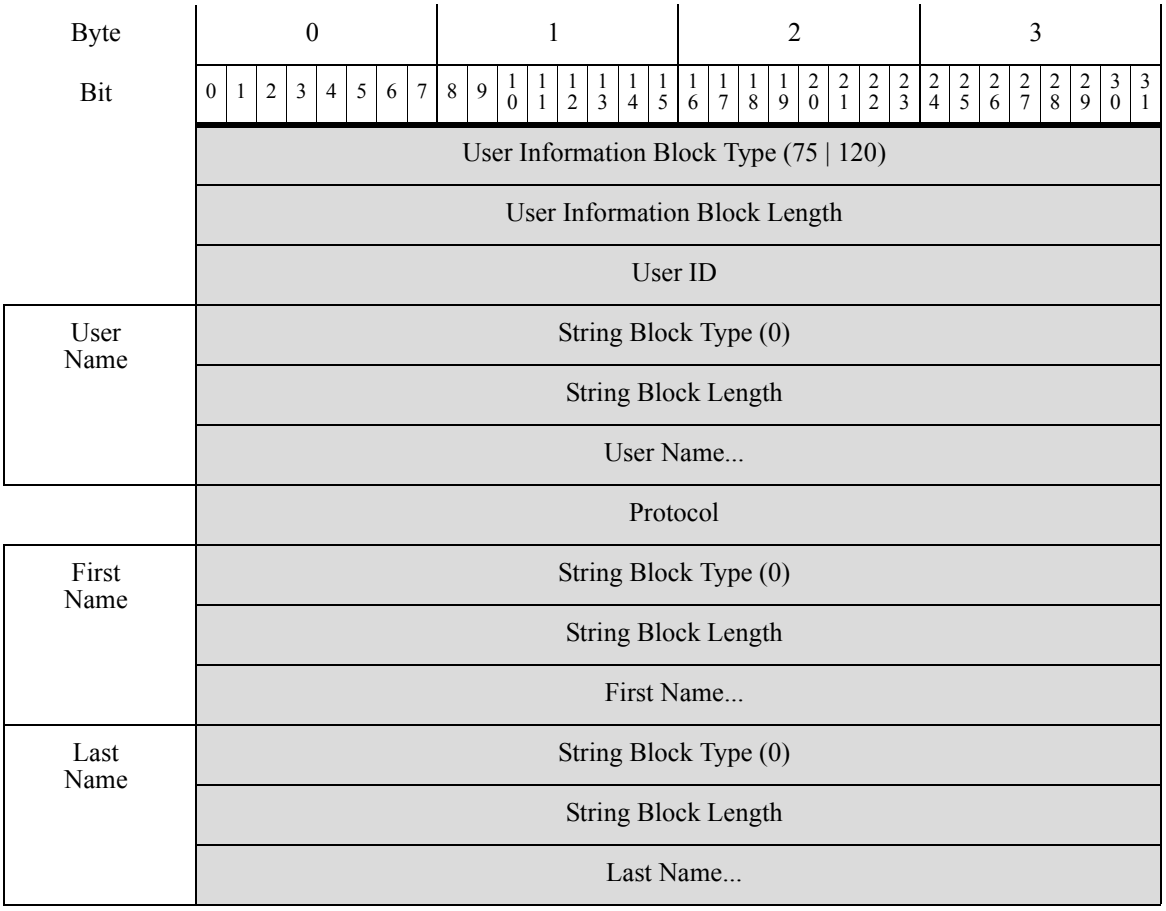
Field	Data Type	Description
Login Type	uint8	The type of user login detected.
String Block Type	uint32	Initiates a String data block containing the Reported By value. This value is always 0.
String Block Length	uint32	Number of bytes in the Reported By String data block, including eight bytes for the block type and length fields, plus the number of bytes in the Reported By field.
Reported By	string	The name of the Active Directory server reporting a login.

User Information Data Block for 5.x

The User Information data block is used in User Modification messages and conveys information for a user detected, removed, or dropped. For more information, see User Modification Messages, page 4-54

The User Information data block has a block type of 75 in the series 1 group of blocks for version 4.7 - 4.10.x and a block type of 120 in the series 1 group of blocks for 5.x. The structures are the same for block types 75 and 120.

The following diagram shows the format of the User Information data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Email	String Block Type (0)																															
	String Block Length																															
	Email...																															
Department	String Block Type (0)																															
	String Block Length																															
	Department...																															
Phone	String Block Type (0)																															
	String Block Length																															
	Phone...																															

The following table describes the components of the User Information data block.

Table B-22 User Information Data Block Fields

Field	Data Type	Description
User Information Block Type	uint32	Initiates a User Information data block. This value is `75` for version 4.7 - 4.10.x and a value of `120` for 5.0+.
User Information Block Length	uint32	Total number of bytes in the User Information data block, including eight bytes for the user information block type and length fields plus the number of bytes in the user information data that follows.
User ID	uint32	Identification number of the user.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields plus the number of bytes in the username.
Username	string	The username for the user.
Protocol	uint32	The protocol for the packet containing the user information.
String Block Type	uint32	Initiates a String data block containing the first name of the user. This value is always `0`.
String Block Length	uint32	Number of bytes in the first name String data block, including eight bytes for the block type and length fields plus the number of bytes in the first name.
First Name	string	The first name for the user.
String Block Type	uint32	Initiates a String data block containing the last name for the user. This value is always `0`.

Table B-22 User Information Data Block Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the user last name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the last name.
Last Name	string	The last name for the user.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
String Block Type	uint32	Initiates a String data block containing the department for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the department String data block, including eight bytes for the block type and length fields, plus the number of bytes in the department.
Department	string	The department for the user.
String Block Type	uint32	Initiates a String data block containing the phone number for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the phone number.
Phone	string	The phone number for the user.

Legacy Host Profile Data Blocks

See the following sections for more information:

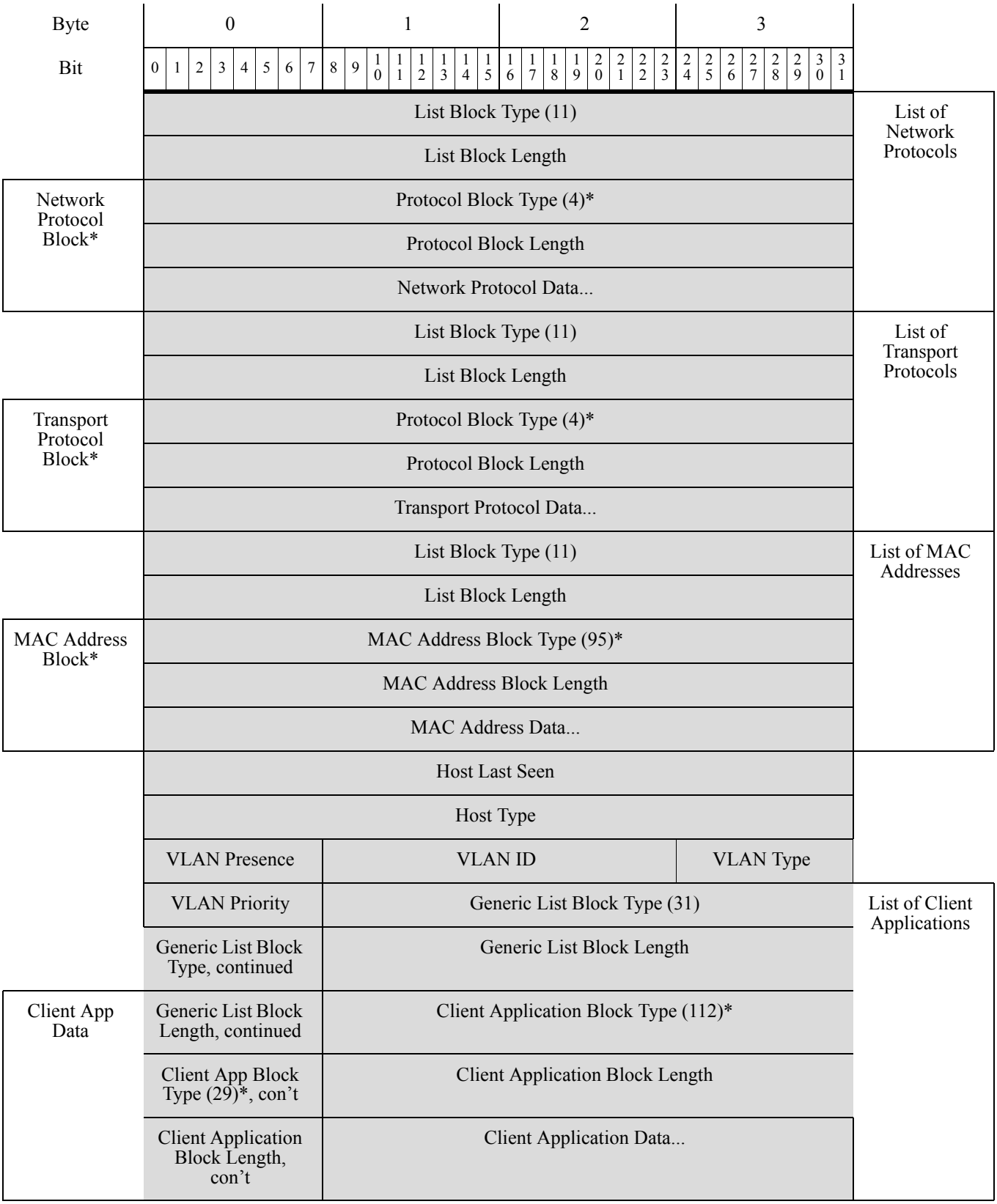
- Host Profile Data Block for 5.0 - 5.0.2, page B-107

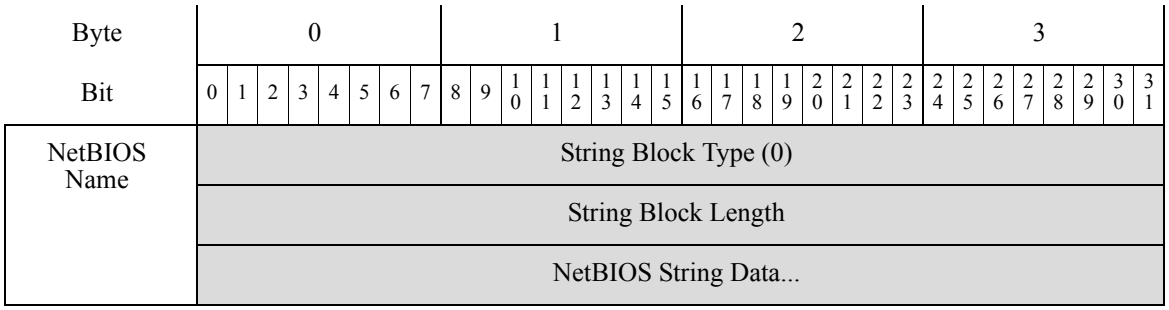
Host Profile Data Block for 5.0 - 5.0.2

The following diagram shows the format of a Host Profile data block in versions 5.0 to 5.0.2. The Host Profile data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a Host Profile data block can convey a NetBIOS name for the host. This Host Profile data block has a block type of 91.

Note An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Host Profile Block Type (91)																															
	Host Profile Block Length																															
	IP Address																															
Server Fingerprints	Hops								Primary/Secondary								Generic List Block Type (31)															
	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Server Fingerprint Data Blocks*															
Client Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	Client Fingerprint Data Blocks*																															
SMB Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	SMB Fingerprint Data Blocks*																															
DHCP Fingerprints	Generic List Block Type (31)																															
	Generic List Block Length																															
	DHCP Fingerprint Data Blocks*																															
	List Block Type (11)																															List of TCP Servers
	List Block Length																															
TCP Server Block*	Server Block Type (36)																															
	Server Block Length																															
	TCP Server Data...																															
	List Block Type (11)																															List of UDP Servers
	List Block Length																															
UDP Server Block*	Server Block Type (36)*																															
	Server Block Length																															
	UDP Server Data...																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
List Block Type (11)
List Block Length
List of Network Protocols
Network Protocol Block*
Protocol Block Type (4)*
Protocol Block Length
Network Protocol Data...
List Block Type (11)
List Block Length
List of Transport Protocols
Transport Protocol Block*
Protocol Block Type (4)*
Protocol Block Length
Transport Protocol Data...
List Block Type (11)
List Block Length
List of MAC Addresses
MAC Address Block*
MAC Address Block Type (95)*
MAC Address Block Length
MAC Address Data...
Host Last Seen
Host Type
VLAN Presence
VLAN ID
VLAN Type
VLAN Priority
Generic List Block Type (31)
List of Client Applications
Generic List Block Type, continued
Generic List Block Length
Client App Data
Generic List Block Length, continued
Client Application Block Type (112)*
Client App Block Type (29)*, con't
Client Application Block Length
Client Application Block Length, con't
Client Application Data...



The following table describes the fields of the host profile data block returned by version 4.9 to version 5.0.2.

Table B-23 Host Profile Data Block for 5.0 - 5.0.2 Fields

Field	Data Type	Description
Host Profile Block Type	uint32	Initiates the Host Profile data block for 4.9 to 5.0.2. This data block has a block type of `91`.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8[4]	IP address of the host described in the profile, in IP address octets.
Hops	uint8	Number of hops from the host to the device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • `0` — Host is in the primary network. • `1` — Host is in the secondary network.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-114 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table B-23 Host Profile Data Block for 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-114 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-114 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-114 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
Server Block Type	uint32	Initiates a Server data block. This value is always `89`.
Server Block Length	uint32	Number of bytes in the Server data block, including eight bytes for the server block type and length fields, plus the number of bytes of TCP server data that follows.
TCP Server Data	variable	Data fields describing a TCP server (as documented for earlier versions of the product).
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always `11`.

Table B-23 Host Profile Data Block for 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
Server Block Type	uint32	Initiates a Server data block describing a UDP server. This value is always `89`.
Server Block Length	uint32	Number of bytes in the Server data block, including eight bytes for the server block type and length fields, plus the number of bytes of UDP server data that follows.
UDP Server Data	variable	Data fields describing a UDP server (as documented for earlier versions of the product).
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Protocol Block Type	uint32	Initiates a Protocol data block describing a network protocol. This value is always `4`.
Protocol Block Length	uint32	Number of bytes in the Protocol data block, including eight bytes for the protocol block type and length fields, plus the number of bytes in the protocol data that follows.
Network Protocol Data	uint16	Data field containing a network protocol number, as documented in Protocol Data Block, page 4-68.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Protocol Block Type	uint32	Initiates a Protocol data block describing a transport protocol. This value is always `4`.
Protocol Block Length	uint32	Number of bytes in the protocol data block, including eight bytes for the protocol block type and length, plus the number of bytes in the protocol data that follows.
Transport Protocol Data	variable	Data field containing a transport protocol number, as documented in Protocol Data Block, page 4-68.
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always `11`.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.

Table B-23 Host Profile Data Block for 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Host MAC Address Block Type	uint32	Initiates a Host MAC Address data block. This value is always 95.
Host MAC Address Block Length	uint32	Number of bytes in the Host MAC Address data block, including eight bytes for the Host MAC address block type and length fields, plus the number of bytes in the Host MAC address data that follows.
Host MAC Address Data	variable	Host MAC address data fields described in Host MAC Address 4.9+, page 4-107.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — Host • 1 — Router • 2 — Bridge • 3 — NAT device • 4 — LB (load balancer)
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Client Application data blocks conveying client application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated client application data blocks.
Client Application Block Type	uint32	Initiates a client application block. This value is always 5.
Client Application Block Length	uint32	Number of bytes in the client application block, including eight bytes for the client application block type and length fields, plus the number of bytes in the client application data that follows.
Client Application Data	variable	Client application data fields describing a client application, as documented in Host Client Application Data Block for 5.0+, page 4-145.
String Block Type	uint32	Initiates a string data block for the NetBIOS name. This value is set to 0 to indicate string data.

Table B-23 Host Profile Data Block for 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Indicates the number of bytes in the NetBIOS name data block, including eight bytes for the string block type and length, plus the number of bytes in the NetBIOS name.
NetBIOS String Data	Variable	Contains the NetBIOS name of the host described in the host profile.

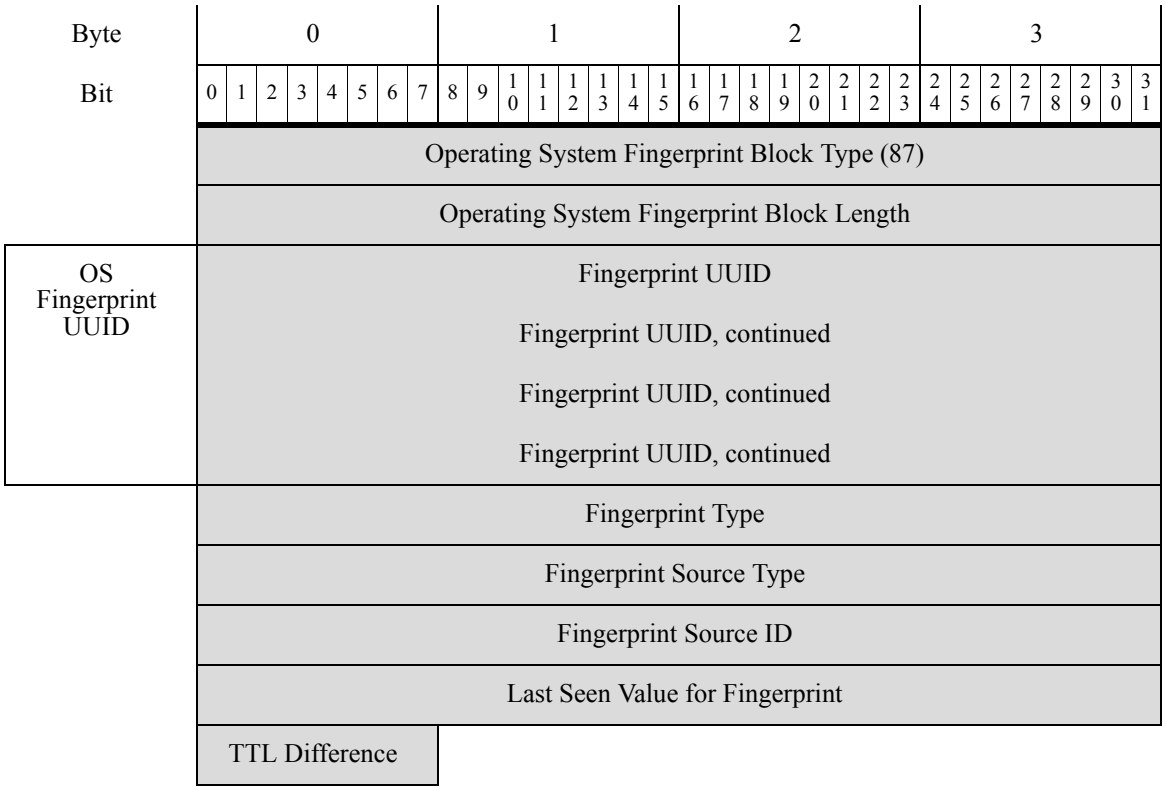
Legacy OS Fingerprint Data Blocks

See the following sections for more information:

- Operating System Fingerprint Data Block for 5.0 - 5.0.2, page B-114

Operating System Fingerprint Data Block for 5.0 - 5.0.2

The Operating System Fingerprint data block has a block type of 87. The block includes a fingerprint Universally Unique Identifier (UUID), as well as the fingerprint type, the fingerprint source type, and the fingerprint source ID. The following diagram shows the format of an Operating System Fingerprint data block for version 5.0 to version 5.0.2.



The following table describes the fields of the operating system fingerprint data block.

Table B-24 Operating System Fingerprint Data Block Fields

Field	Data Type	Description
Operating System Fingerprint Data Block Type	uint32	Initiates the operating system data block. This value is always 87.
Operating System Data Block Length	uint32	Number of bytes in the Operating System Fingerprint data block. This value should always be 41: eight bytes for the data block type and length fields, sixteen bytes for the fingerprint UUID value, four bytes for the fingerprint type, four bytes for the fingerprint source type, four bytes for the fingerprint source ID, four bytes for the last seen value, and one byte for the TTL difference.
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the vulnerability database (VDB).
Fingerprint Type	uint32	Indicates the type of fingerprint.
Fingerprint Source Type	uint32	Indicates the type (i.e., user or scanner) of the source that supplied the operating system fingerprint.
Fingerprint Source ID	uint32	Indicates the ID of the source that supplied the operating system fingerprint.
Last Seen	uint32	Indicates when the fingerprint was last seen in traffic.
TTL Difference	uint8	Indicates the difference between the TTL value in the fingerprint and the TTL value seen in the packet used to fingerprint the host.

Legacy Connection Data Structures

For more information, see the following sections:

Connection Statistics Data Block 5.0 - 5.0.2

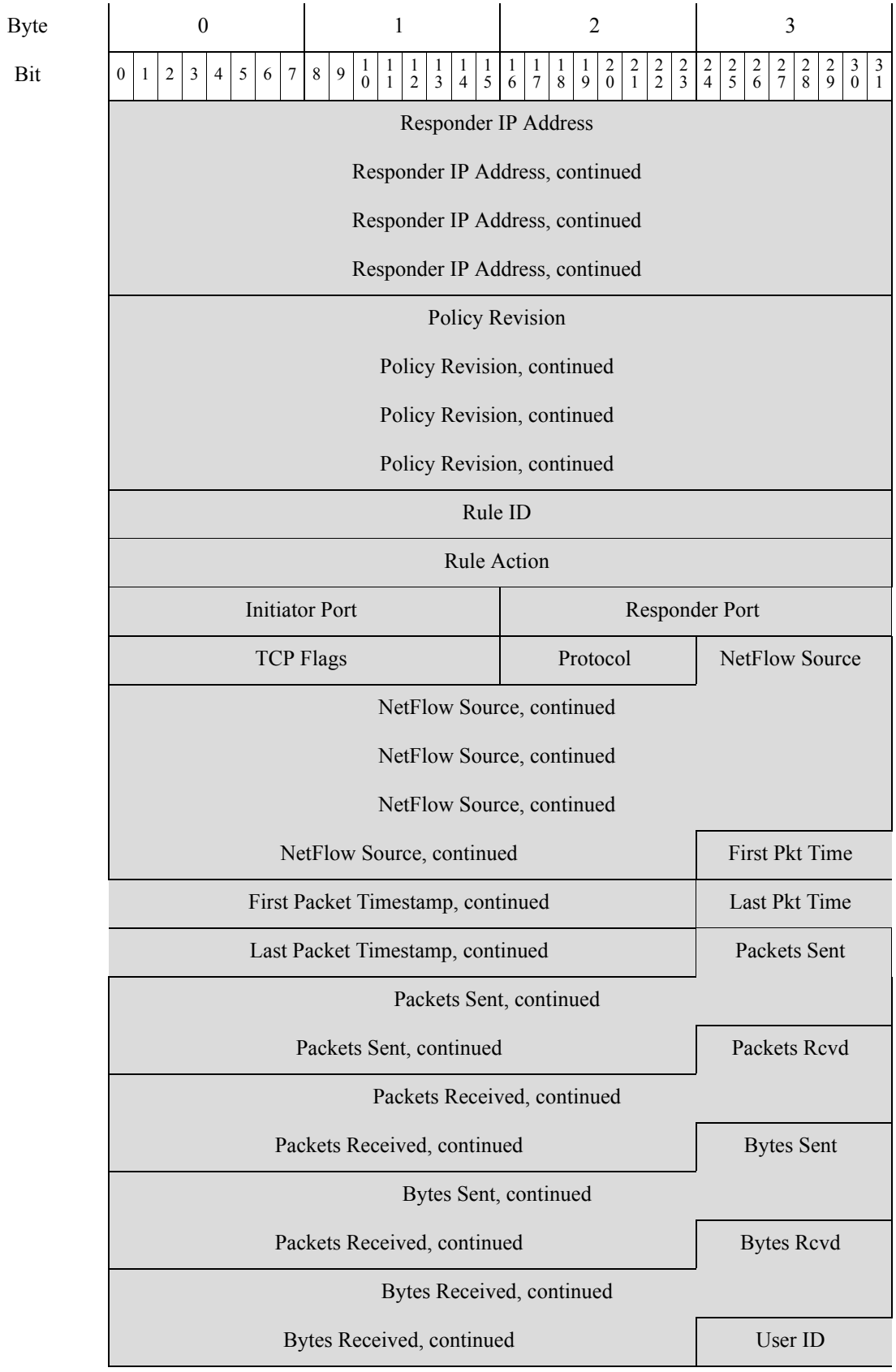
The Connection Statistics data block is used in Connection Data messages. The Connection Statistics data block for version 5.0 - 5.0.2 has a block type of 115.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

The following diagram shows the format of a Connection Statistics data block for 5.0 - 5.0.2:

::

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Connection Data Block Type (115)																																
Connection Data Block Length																																
Device ID																																
Ingress Zone																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Egress Zone																																
Egress Zone, continued																																
Egress Zone, continued																																
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Responder IP Address
Responder IP Address, continued
Responder IP Address, continued
Responder IP Address, continued
Policy Revision
Policy Revision, continued
Policy Revision, continued
Policy Revision, continued
Rule ID
Rule Action
Initiator Port
Responder Port
TCP Flags
Protocol
NetFlow Source
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
First Pkt Time
First Packet Timestamp, continued
Last Pkt Time
Last Packet Timestamp, continued
Packets Sent
Packets Sent, continued
Packets Sent, continued
Packets Rcvd
Packets Received, continued
Packets Received, continued
Bytes Sent
Bytes Sent, continued
Packets Received, continued
Bytes Rcvd
Bytes Received, continued
Bytes Received, continued
User ID

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User ID, continued																								Application Protocol ID							
	Application Protocol ID, continued																								URL Category							
	URL Category, continued																								URL Reputation							
	URL Reputation, continued																								Client App ID							
	Client Application ID, continued																								Web App ID							
	Web Application ID, continued																								String Block Type (0)							
Client App URL	String Block Type, continued																								String Block Length							
	String Block Length, continued																								Client Application URL...							
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name....																															
Client App Version	String Block Type (0)																															
	String Block Length																															
	Client Application Version...																															

The following table describes the fields of the Connection Statistics data block for 5.0 - 5.0.2.

Table B-25 Connection Statistics Data Block 5.0 - 5.0.2 Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.0 to 5.0.2. The value is always `115`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.

Table B-25 Connection Statistics Data Block 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint32	The action selected in the user interface for that rule (allow, block, and so forth).
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Packets Sent	uint64	Number of packets transmitted by the initiating host.
Packets Received	uint64	Number of packets transmitted by the responding host.
Bytes Sent	uint64	Number of bytes transmitted by the initiating host.
Bytes Received	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.

Table B-25 Connection Statistics Data Block 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

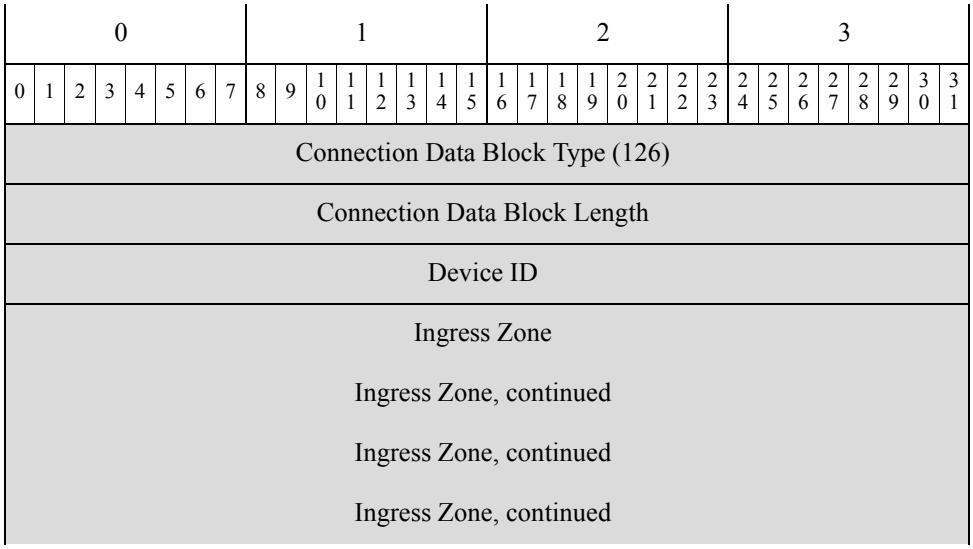
Connection Statistics Data Block 5.1

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection data block between 5.0.2 and 5.1 include the addition of new fields with configuration parameters introduced in 5.1 (rule action reason, monitor rules, Security Intelligence source/destination, Security Intelligence layer). The Connection Statistics data block for version 5.1 has a block type of 126.

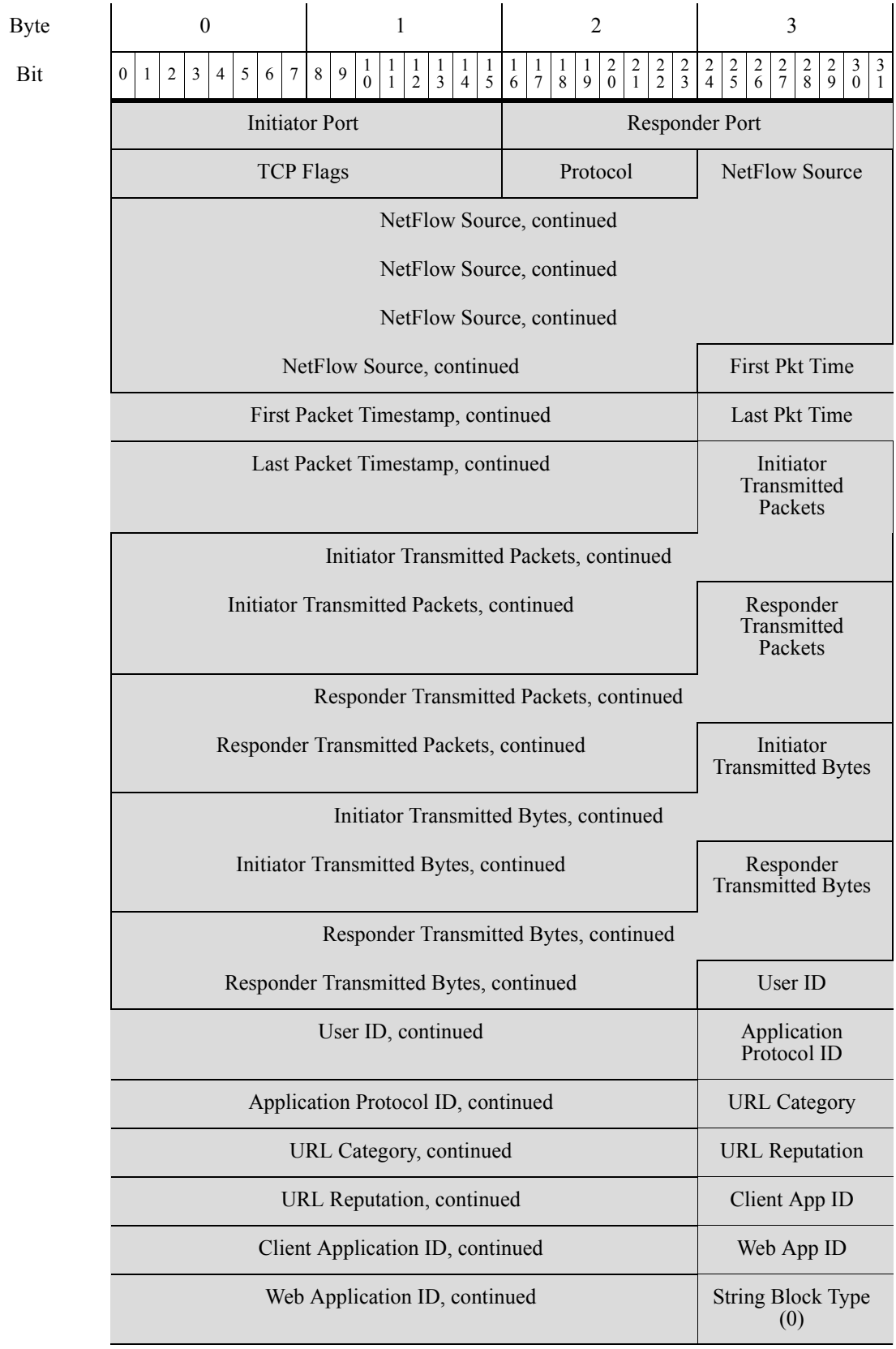
For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

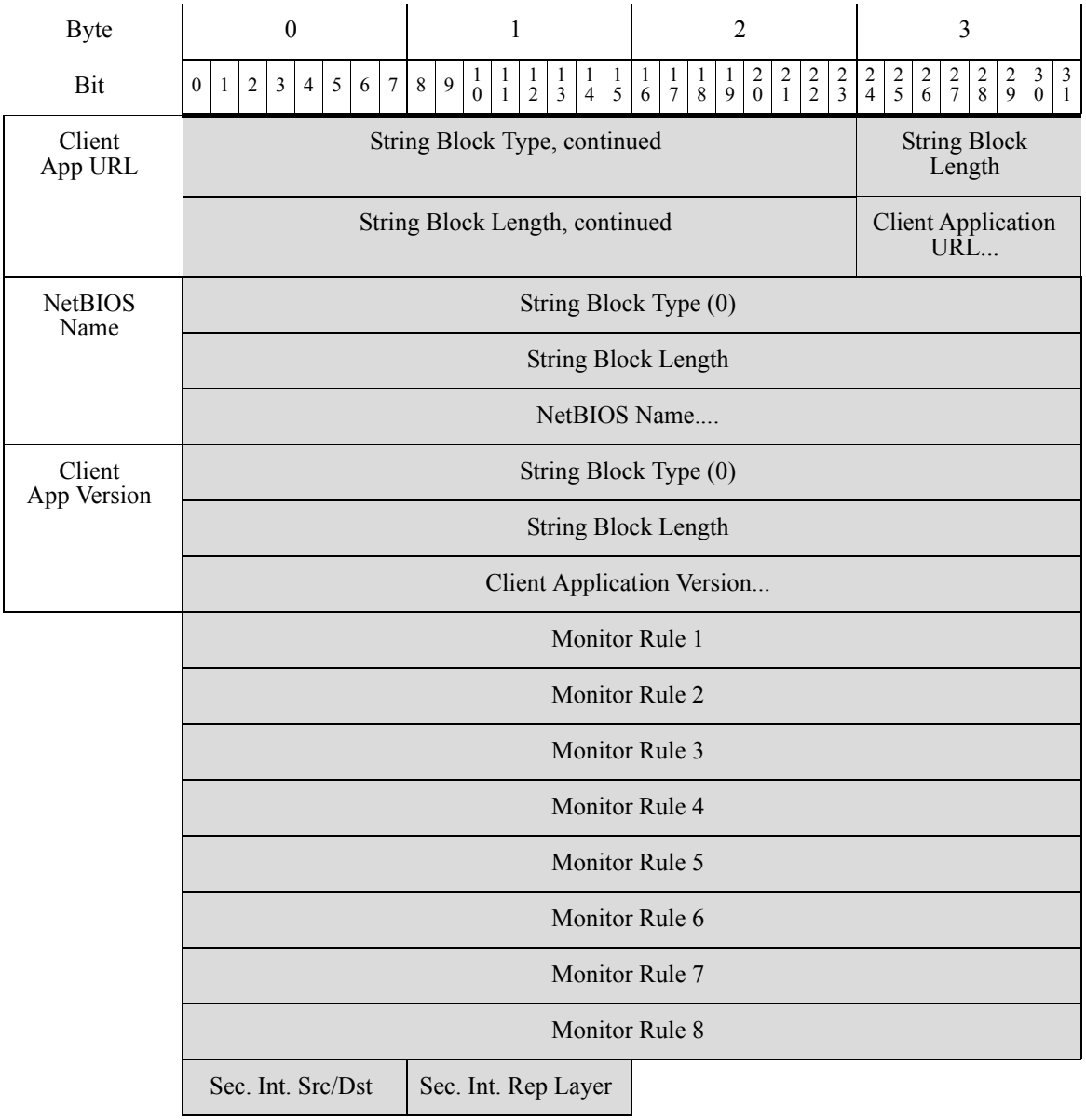
The following diagram shows the format of a Connection Statistics data block for 5.1:

::



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Egress Zone																																
Egress Zone, continued																																
Egress Zone, continued																																
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																
Policy Revision, continued																																
Policy Revision, continued																																
Policy Revision, continued																																
Rule ID																																
Rule Action																	Rule Reason															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Initiator Port
Responder Port
TCP Flags
Protocol
NetFlow Source
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
First Pkt Time
First Packet Timestamp, continued
Last Pkt Time
Last Packet Timestamp, continued
Initiator Transmitted Packets
Initiator Transmitted Packets, continued
Initiator Transmitted Packets, continued
Responder Transmitted Packets
Responder Transmitted Packets, continued
Responder Transmitted Packets, continued
Initiator Transmitted Bytes
Initiator Transmitted Bytes, continued
Initiator Transmitted Bytes, continued
Responder Transmitted Bytes
Responder Transmitted Bytes, continued
Responder Transmitted Bytes, continued
User ID
User ID, continued
Application Protocol ID
Application Protocol ID, continued
URL Category
URL Category, continued
URL Reputation
URL Reputation, continued
Client App ID
Client Application ID, continued
Web App ID
Web Application ID, continued
String Block Type (0)



The following table describes the fields of the Connection Statistics data block for 5.1.

Table B-26 Connection Statistics Data Block 5.1 Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.1. The value is always `126`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.

Table B-26 Connection Statistics Data Block 5.1 Fields (continued)

Field	Data Type	Description
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.

Table B-26 Connection Statistics Data Block 5.1 Fields (continued)

Field	Data Type	Description
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.

Table B-26 Connection Statistics Data Block 5.1 Fields (continued)

Field	Data Type	Description
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.

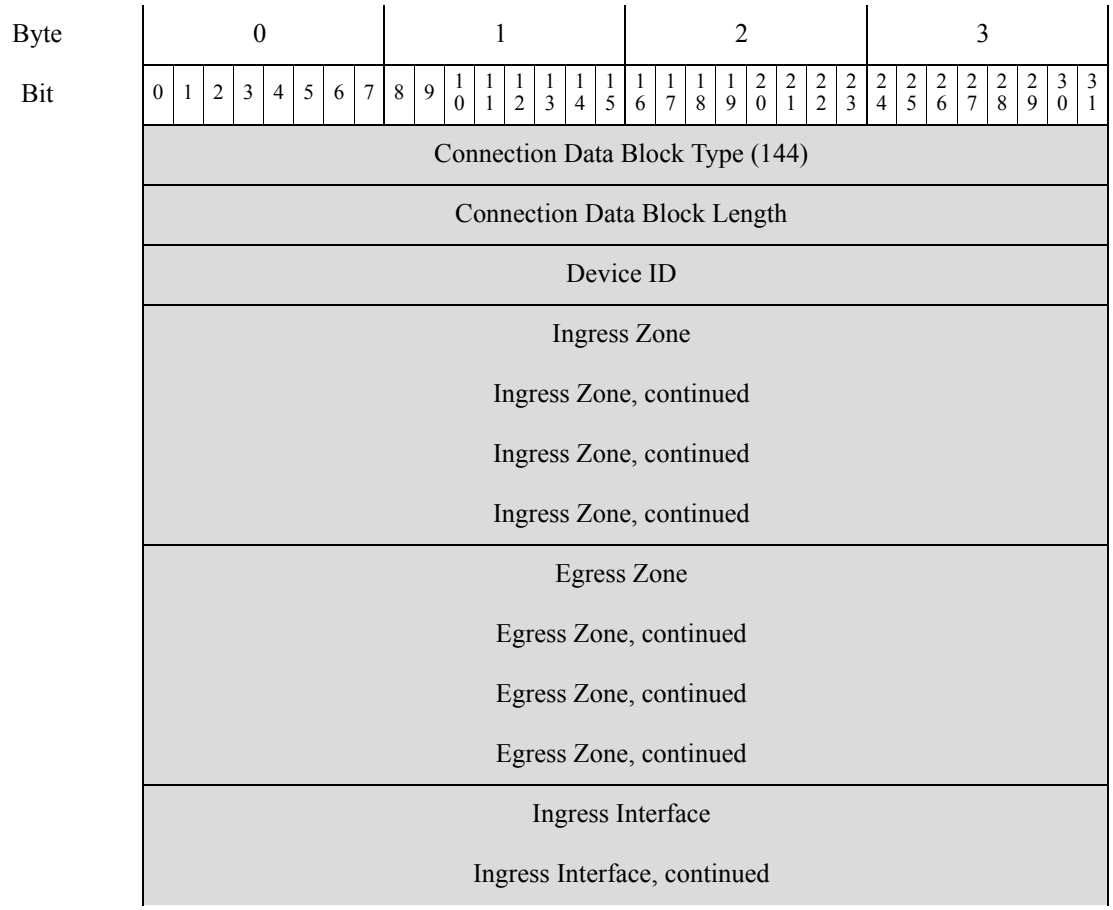
Connection Statistics Data Block 5.2.x

The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.1.1 and 5.2 include the addition of new fields to support geolocation. The connection statistics data block for version 5.2.x has a block type of 144 in the series 1 group of blocks. It deprecates block type 137, Connection Statistics Data Block 5.1.1.x, page B-133.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

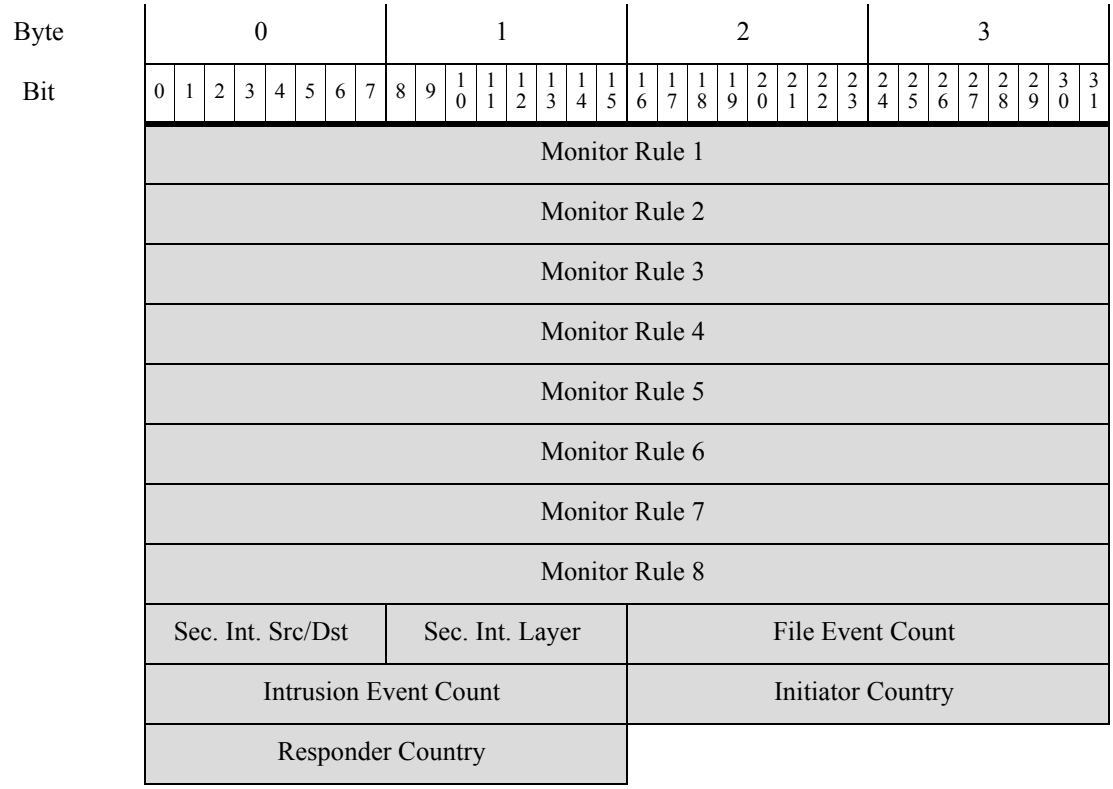
The following diagram shows the format of a Connection Statistics data block for 5.2.x:

::



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																
Policy Revision, continued																																
Policy Revision, continued																																
Policy Revision, continued																																
Rule ID																																
Rule Action																	Rule Reason															
Initiator Port																	Responder Port															
TCP Flags																	Protocol								NetFlow Source							
NetFlow Source, continued																																
NetFlow Source, continued																																
NetFlow Source, continued																																
NetFlow Source, continued																									Instance ID							

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Instance ID, cont.								Connection Counter																First Pkt Time							
	First Packet Timestamp, continued																								Last Pkt Time							
	Last Packet Timestamp, continued																								Initiator Tx Packets							
	Initiator Transmitted Packets, continued																															
	Initiator Transmitted Packets, continued																								Resp. Tx Packets							
	Responder Transmitted Packets, continued																															
	Responder Transmitted Packets, continued																								Initiator Tx Bytes							
	Initiator Transmitted Bytes, continued																															
	Initiator Transmitted Bytes, continued																								Resp. Tx Bytes							
	Responder Transmitted Bytes, continued																															
	Responder Transmitted Bytes, continued																								User ID							
	User ID, continued																								Application Prot. ID							
	Application Protocol ID, continued																								URL Category							
	URL Category, continued																								URL Reputation							
	URL Reputation, continued																								Client App ID							
	Client Application ID, continued																								Web App ID							
Client URL	Web Application ID, continued																								Str. Block Type (0)							
	String Block Type, continued																								String Block Length							
	String Block Length, continued																								Client App. URL...							
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name...																															
Client App Version	String Block Type (0)																															
	String Block Length																															
	Client Application Version...																															



The following table describes the fields of the Connection Statistics data block for 5.2.x:

Table B-27 Connection Statistics Data Block 5.2.x Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.2.x. The value is always `144`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.

Table B-27 Connection Statistics Data Block 5.2.x Fields (continued)

Field	Data Type	Description
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.

Table B-27 Connection Statistics Data Block 5.2.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.

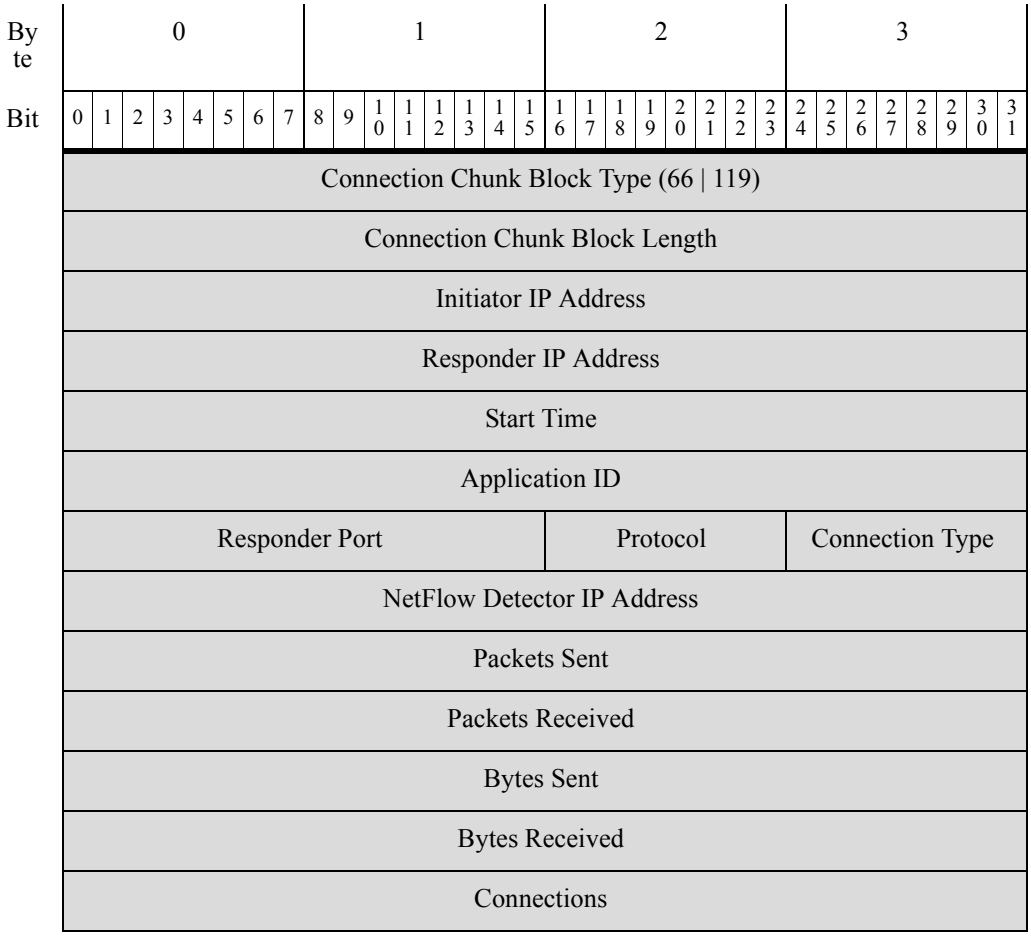
Table B-27 Connection Statistics Data Block 5.2.x Fields (continued)

Field	Data Type	Description
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint16	Code for the country of the responding host.

Connection Chunk Data Block for 5.0 - 5.1

The Connection Chunk data block conveys connection data detected by a NetFlow device. The Connection Chunk data block has a block type of 66 for pre-4.10.1 versions. For versions 5.0 - 5.1, it has a block type of 119.

The following diagram shows the format of the Connection Chunk data block:



The following table describes the components of the Connection Chunk data block:

Table B-28 Connection Chunk Data Block Fields

Field	Data Type	Description
Connection Chunk Block Type	uint32	Initiates a Connection Chunk data block. This value is `66` for versions before 4.10.1 and a value of `119` for version 5.0.
Connection Chunk Block Length	uint32	Total number of bytes in the Connection Chunk data block, including eight bytes for the connection chunk block type and length fields, plus the number of bytes in the connection chunk data that follows.
Initiator IP Address	uint8[4]	IP address of the host that initiated the connection, in IP address octets.
Responder IP Address	uint8[4]	IP address of the host responding in the connection, in IP address octets.
Start Time	uint32	The starting time for the connection chunk.
Application ID	uint32	Application identification number for the application protocol used in the connection.
Responder Port	uint16	The port used by the responder in the connection chunk.
Protocol	uint8	The protocol for the packet containing the user information.
Connection Type	uint8	The type of connection.
Source Device IP Address	uint8[4]	IP address of the NetFlow device that detected the connection, in IP address octets.
Packets Sent	uint32	The number of packets sent in the connection chunk.
Packets Received	uint32	The number of packets received in the connection chunk.
Bytes Sent	uint32	The number of bytes sent in the connection chunk.
Bytes Received	uint32	The number of bytes received in the connection chunk.
Connections	uint32	The number of sessions made in the connection chunk.

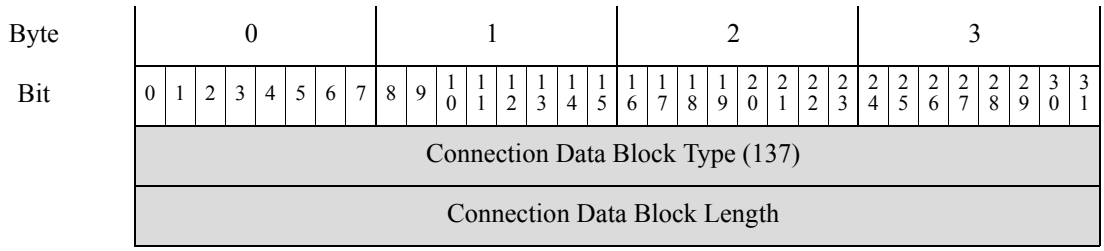
Connection Statistics Data Block 5.1.1.x

The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.1 and 5.1.1 include the addition of new fields to identify associated intrusion events. The connection statistics data block for version 5.1.1.x has a block type of 137. It deprecates block type 126, Connection Statistics Data Block 5.1, page B-120.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

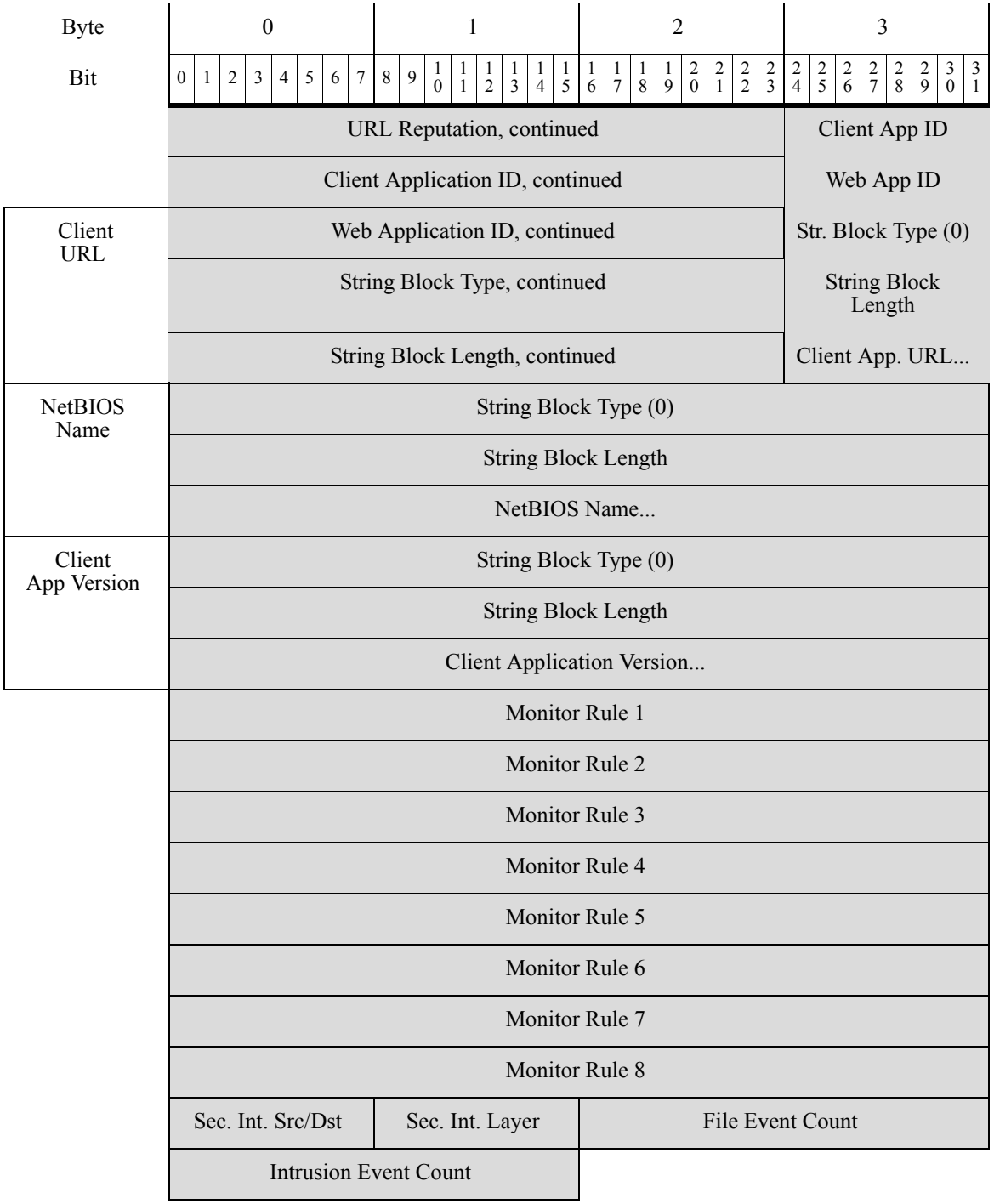
The following diagram shows the format of a Connection Statistics data block for 5.1.1:

::



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Device ID																																
Ingress Zone																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Egress Zone																																
Egress Zone, continued																																
Egress Zone, continued																																
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Policy Revision, continued																																
Policy Revision, continued																																
Policy Revision, continued																																
Rule ID																																
Rule Action																	Rule Reason															
Initiator Port																	Responder Port															
TCP Flags																	Protocol								NetFlow Source							
NetFlow Source, continued																																
NetFlow Source, continued																																
NetFlow Source, continued																																
NetFlow Source, continued																									Instance ID							
Instance ID, cont.									Connection Counter																First Pkt Time							
First Packet Timestamp, continued																									Last Pkt Time							
Last Packet Timestamp, continued																									Initiator Tx Packets							
Initiator Transmitted Packets, continued																																
Initiator Transmitted Packets, continued																									Resp. Tx Packets							
Responder Transmitted Packets, continued																																
Responder Transmitted Packets, continued																									Initiator Tx Bytes							
Initiator Transmitted Bytes, continued																																
Initiator Transmitted Bytes, continued																									Resp. Tx Bytes							
Responder Transmitted Bytes, continued																																
Responder Transmitted Bytes, continued																									User ID							
User ID, continued																									Application Prot. ID							
Application Protocol ID, continued																									URL Category							
URL Category, continued																									URL Reputation							



The following table describes the fields of the Connection Statistics data block for 5.1.1.x.

Table B-29 Connection Statistics Data Block 5.1.1.x Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.1.1.x. The value is always `137`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.

Table B-29 Connection Statistics Data Block 5.1.1.x Fields (continued)

Field	Data Type	Description
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

Table B-29 Connection Statistics Data Block 5.1.1.x Fields (continued)

Field	Data Type	Description
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.

Connection Statistics Data Block 5.3

The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.2.x and 5.3 include the addition of new fields for NetFlow information. The connection statistics data block for version 5.3 has a block type of 152 in the series 1 group of blocks. It deprecates block type 144, Connection Statistics Data Block 5.2.x, page B-126.

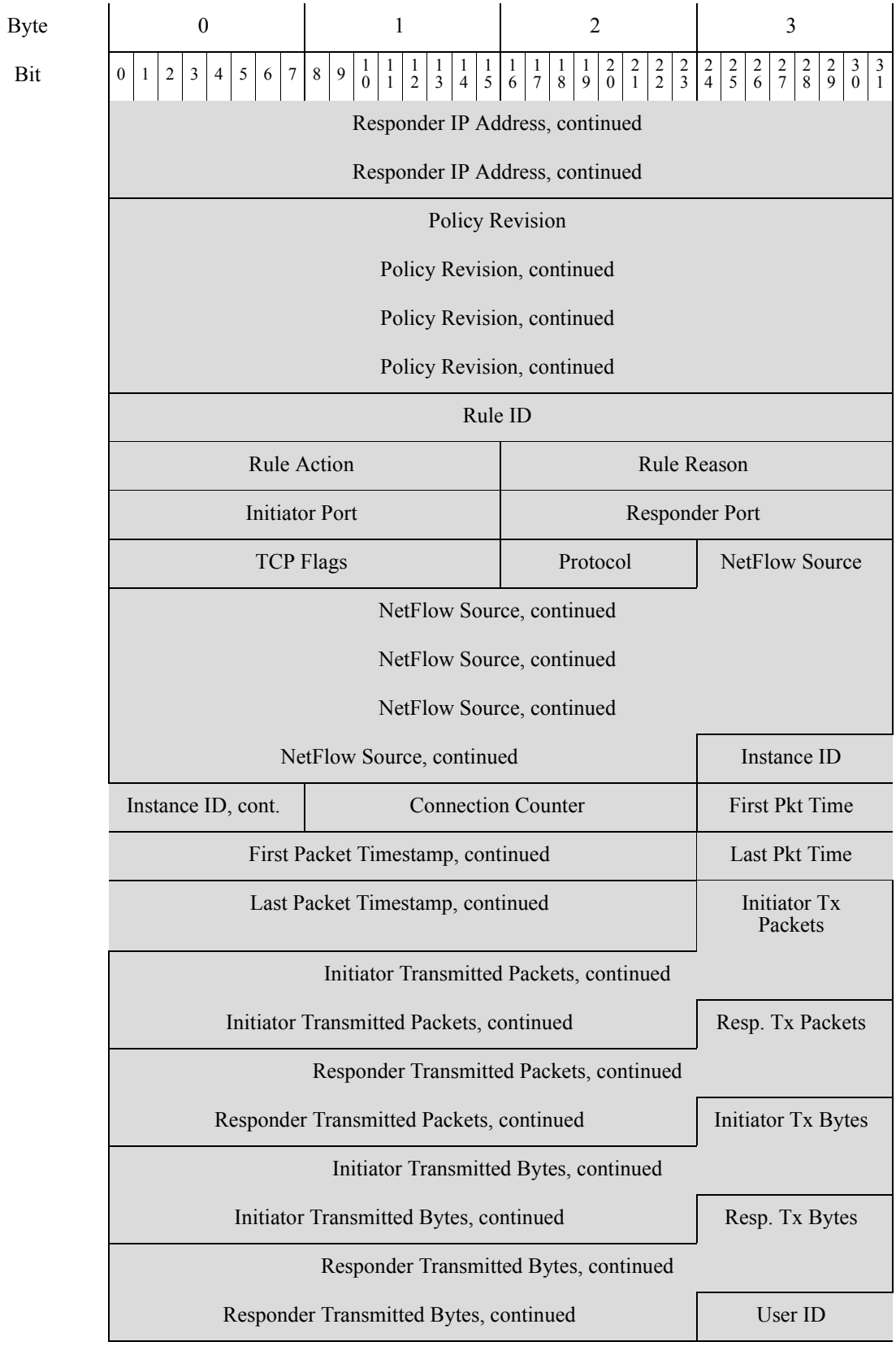
You request connection event records by setting the extended event flag—bit 30 in the Request Flags field—in the request message with an event version of 10 and an event code of 71. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

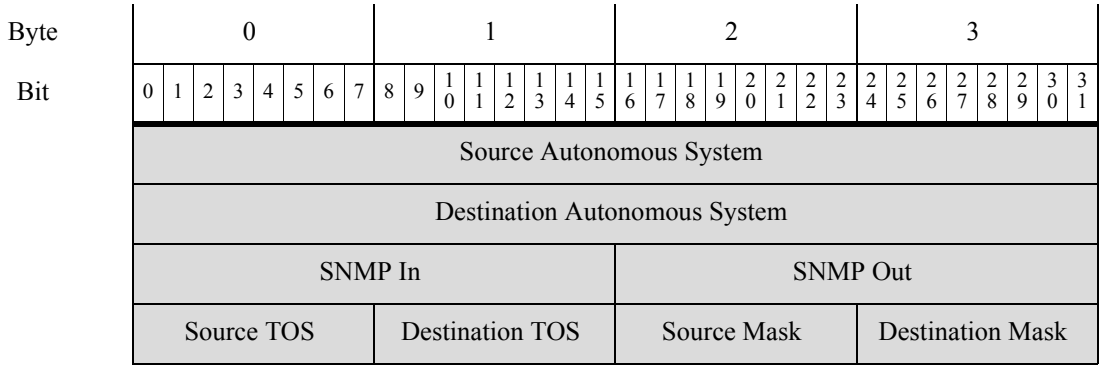
The following diagram shows the format of a Connection Statistics data block for 5.3+:

::

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Connection Data Block Type (152)																																
Connection Data Block Length																																
Device ID																																
Ingress Zone																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Egress Zone																																
Egress Zone, continued																																
Egress Zone, continued																																
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Responder IP Address, continued
Responder IP Address, continued
Policy Revision
Policy Revision, continued
Policy Revision, continued
Policy Revision, continued
Rule ID
Rule Action
Rule Reason
Initiator Port
Responder Port
TCP Flags
Protocol
NetFlow Source
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
Instance ID
Instance ID, cont.
Connection Counter
First Pkt Time
First Packet Timestamp, continued
Last Pkt Time
Last Packet Timestamp, continued
Initiator Tx Packets
Initiator Transmitted Packets, continued
Initiator Transmitted Packets, continued
Resp. Tx Packets
Responder Transmitted Packets, continued
Responder Transmitted Packets, continued
Initiator Tx Bytes
Initiator Transmitted Bytes, continued
Initiator Transmitted Bytes, continued
Resp. Tx Bytes
Responder Transmitted Bytes, continued
Responder Transmitted Bytes, continued
User ID

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User ID, continued																								Application Prot. ID							
	Application Protocol ID, continued																								URL Category							
	URL Category, continued																								URL Reputation							
	URL Reputation, continued																								Client App ID							
	Client Application ID, continued																								Web App ID							
Client URL	Web Application ID, continued																								Str. Block Type (0)							
	String Block Type, continued																								String Block Length							
	String Block Length, continued																								Client App. URL...							
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name...																															
Client App Version	String Block Type (0)																															
	String Block Length																															
	Client Application Version...																															
	Monitor Rule 1																															
	Monitor Rule 2																															
	Monitor Rule 3																															
	Monitor Rule 4																															
	Monitor Rule 5																															
	Monitor Rule 6																															
	Monitor Rule 7																															
	Monitor Rule 8																															
	Sec. Int. Src/Dst								Sec. Int. Layer								File Event Count															
	Intrusion Event Count																Initiator Country															
	Responder Country																IOC Number															



The following table describes the fields of the Connection Statistics data block for 5.3.

Table B-30 Connection Statistics Data Block 5.3+ Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.3. The value is always `152`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.

Table B-30 Connection Statistics Data Block 5.3+ Fields (continued)

Field	Data Type	Description
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.

Table B-30 Connection Statistics Data Block 5.3+ Fields (continued)

Field	Data Type	Description
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.

Table B-30 Connection Statistics Data Block 5.3+ Fields (continued)

Field	Data Type	Description
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.
Destination Mask	uint8	Destination address prefix mask.

Connection Statistics Data Block 5.3.1

The connection statistics data block is used in connection data messages. The only changes to the connection data block between versions 5.3 and 5.3.1 is the addition of a security context field. The connection statistics data block for version 5.3.1 has a block type of 154 in the series 1 group of blocks. It deprecates block type 152, Connection Statistics Data Block 5.3, page B-139.

You request connection event records by setting the extended event flag—bit 30 in the Request Flags field—in the request message with an event version of 11 and an event code of 71. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

The following diagram shows the format of a Connection Statistics data block for 5.3.1:

::

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Connection Data Block Type (154)																																
Connection Data Block Length																																
Device ID																																
Ingress Zone Ingress Zone, continued Ingress Zone, continued Ingress Zone, continued																																
Egress Zone Egress Zone, continued Egress Zone, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																
Policy Revision, continued																																
Policy Revision, continued																																
Policy Revision, continued																																
Rule ID																																
Rule Action																	Rule Reason															
Initiator Port																	Responder Port															
TCP Flags																	Protocol								NetFlow Source							
NetFlow Source, continued																																

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	NetFlow Source, continued																															
	NetFlow Source, continued																															
	NetFlow Source, continued																								Instance ID							
	Instance ID, cont.								Connection Counter																First Pkt Time							
	First Packet Timestamp, continued																								Last Pkt Time							
	Last Packet Timestamp, continued																								Initiator Tx Packets							
	Initiator Transmitted Packets, continued																															
	Initiator Transmitted Packets, continued																								Resp. Tx Packets							
	Responder Transmitted Packets, continued																															
	Responder Transmitted Packets, continued																								Initiator Tx Bytes							
	Initiator Transmitted Bytes, continued																															
	Initiator Transmitted Bytes, continued																								Resp. Tx Bytes							
	Responder Transmitted Bytes, continued																															
	Responder Transmitted Bytes, continued																								User ID							
	User ID, continued																								Application Prot. ID							
	Application Protocol ID, continued																								URL Category							
	URL Category, continued																								URL Reputation							
	URL Reputation, continued																								Client App ID							
	Client Application ID, continued																								Web App ID							
Client URL	Web Application ID, continued																								Str. Block Type (0)							
	String Block Type, continued																								String Block Length							
	String Block Length, continued																								Client App. URL...							
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name...																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Client App Version	String Block Type (0)																															
	String Block Length																															
	Client Application Version...																															
	Monitor Rule 1																															
	Monitor Rule 2																															
	Monitor Rule 3																															
	Monitor Rule 4																															
	Monitor Rule 5																															
	Monitor Rule 6																															
	Monitor Rule 7																															
	Monitor Rule 8																															
	Sec. Int. Src/Dst								Sec. Int. Layer								File Event Count															
	Intrusion Event Count																Initiator Country															
	Responder Country																IOC Number															
	Source Autonomous System																															
	Destination Autonomous System																															
	SNMP In																SNMP Out															
	Source TOS								Destination TOS								Source Mask								Destination Mask							
	Security Context Security Context, continued Security Context, continued Security Context, continued																															

The following table describes the fields of the Connection Statistics data block for 5.3.1.

Table B-31 Connection Statistics Data Block 5.3.1 Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.3.1+. The value is always 154.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.

Table B-31 Connection Statistics Data Block 5.3.1 Fields (continued)

Field	Data Type	Description
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

Table B-31 Connection Statistics Data Block 5.3.1 Fields (continued)

Field	Data Type	Description
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.

Table B-31 Connection Statistics Data Block 5.3.1 Fields (continued)

Field	Data Type	Description
Destination Mask	uint8	Destination address prefix mask.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.

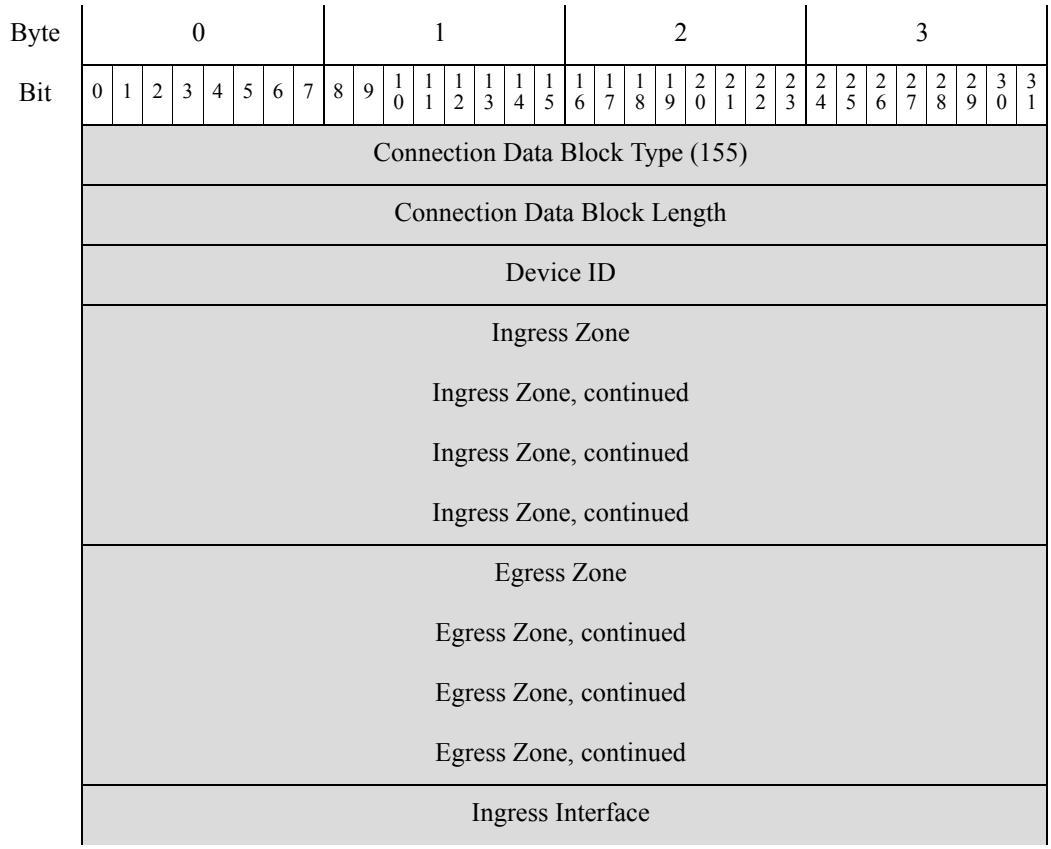
Connection Statistics Data Block 5.4

The connection statistics data block is used in connection data messages. Several new fields have been added to the Connection Statistics Data Block for 5.4. Fields have been added to support SSL connections, HTTP redirection, and network analysis policies. The connection statistics data block for version 5.4 has a block type of 155 in the series 1 group of blocks. It deprecates block type 154, Connection Statistics Data Block 5.3.1, page B-146.

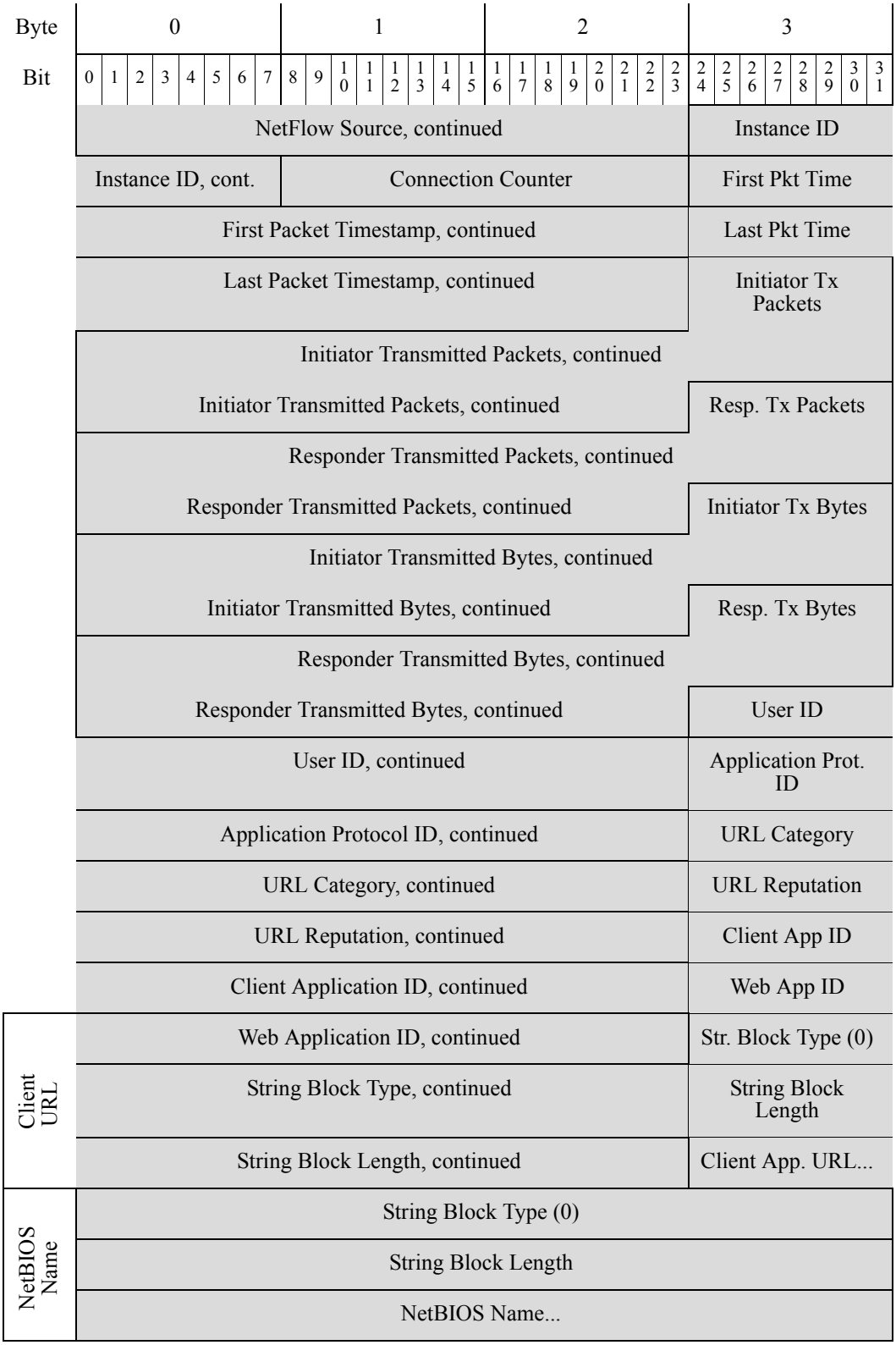
You request connection event records by setting the extended event flag—bit 30 in the Request Flags field—in the request message with an event version of 12 and an event code of 71. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

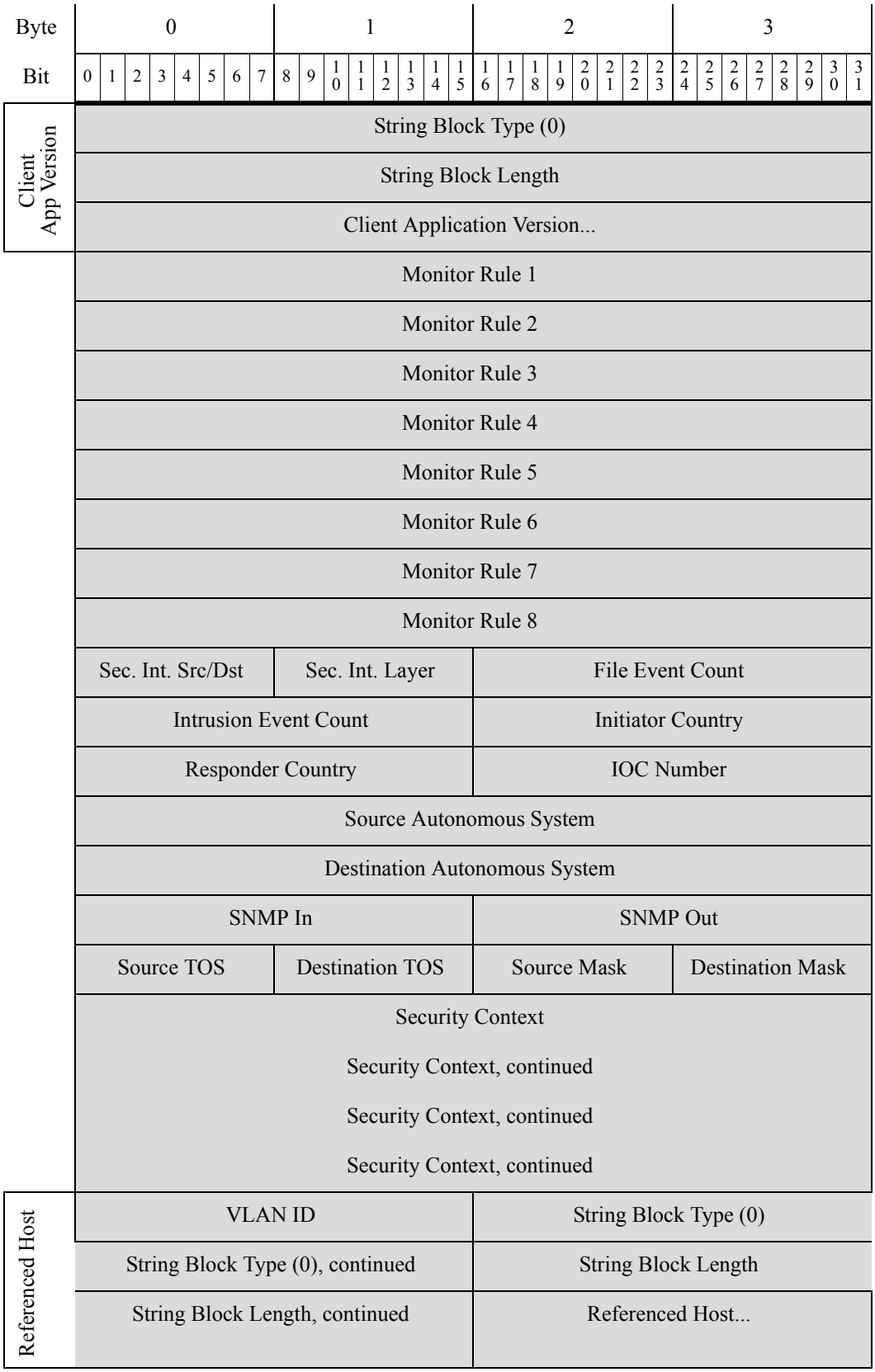
For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

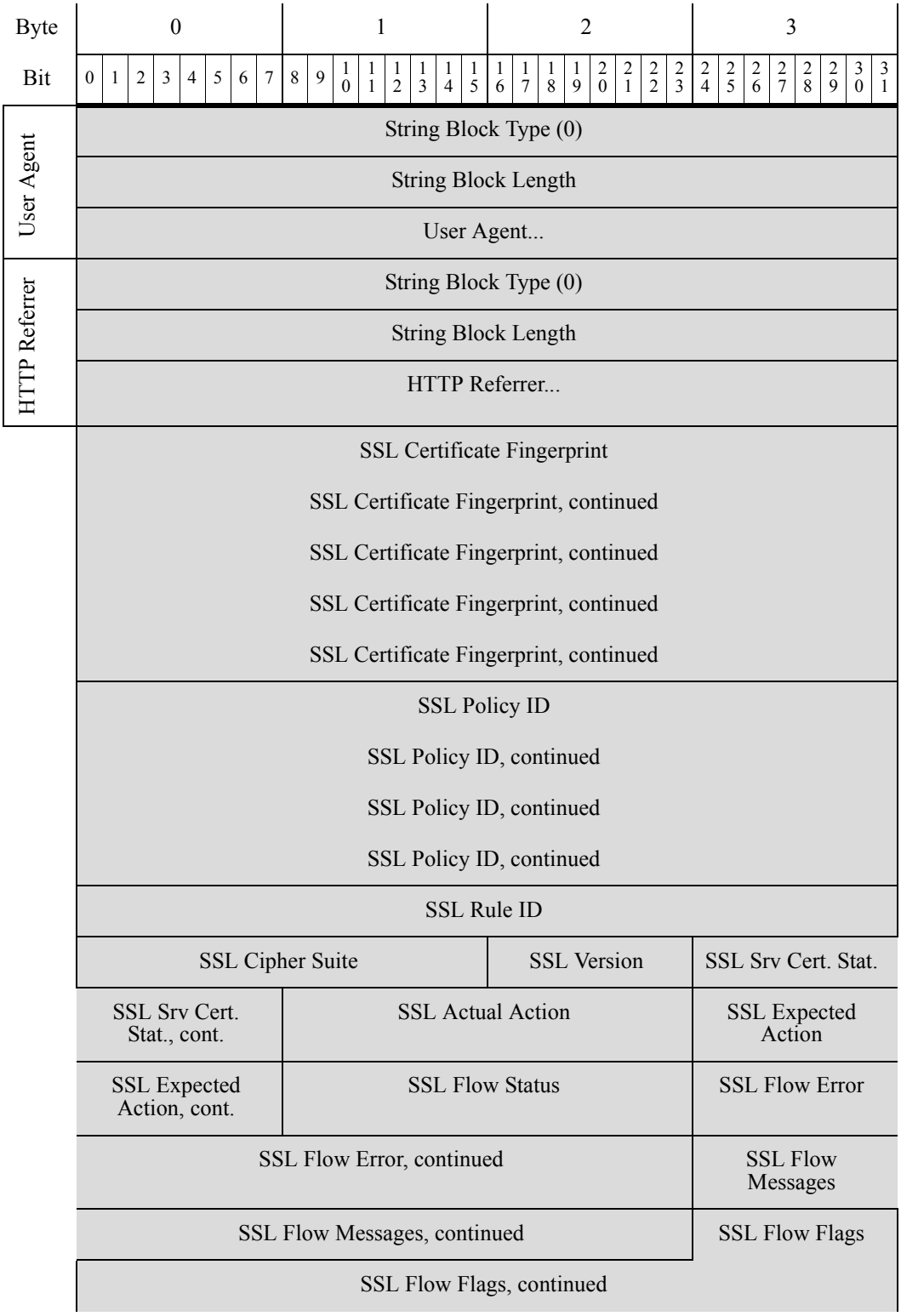
The following diagram shows the format of a Connection Statistics data block for 5.4:

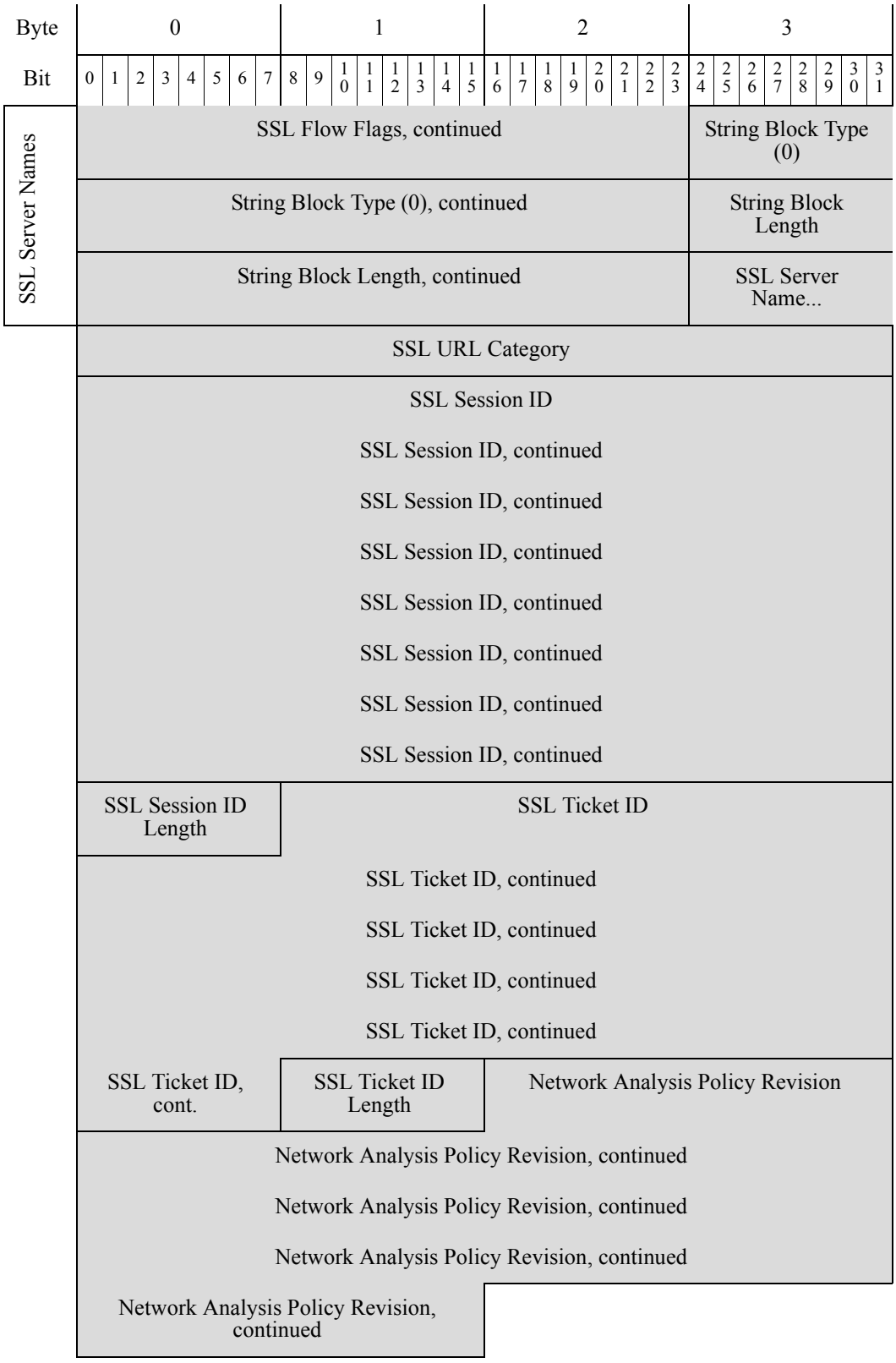


Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																
Policy Revision, continued																																
Policy Revision, continued																																
Policy Revision, continued																																
Rule ID																																
Rule Action																	Rule Reason															
Initiator Port																	Responder Port															
TCP Flags																	Protocol								NetFlow Source							
NetFlow Source, continued																																
NetFlow Source, continued																																
NetFlow Source, continued																																


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
NetFlow Source, continued
Instance ID
Instance ID, cont.
Connection Counter
First Pkt Time
First Packet Timestamp, continued
Last Pkt Time
Last Packet Timestamp, continued
Initiator Tx Packets
Initiator Transmitted Packets, continued
Initiator Transmitted Packets, continued
Resp. Tx Packets
Responder Transmitted Packets, continued
Responder Transmitted Packets, continued
Initiator Tx Bytes
Initiator Transmitted Bytes, continued
Initiator Transmitted Bytes, continued
Resp. Tx Bytes
Responder Transmitted Bytes, continued
Responder Transmitted Bytes, continued
User ID
User ID, continued
Application Prot. ID
Application Protocol ID, continued
URL Category
URL Category, continued
URL Reputation
URL Reputation, continued
Client App ID
Client Application ID, continued
Web App ID
Client URL
Web Application ID, continued
Str. Block Type (0)
String Block Type, continued
String Block Length
String Block Length, continued
Client App. URL...
NetBIOS Name
String Block Type (0)
String Block Length
NetBIOS Name...


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Client App Version
String Block Type (0)
String Block Length
Client Application Version...
Monitor Rule 1
Monitor Rule 2
Monitor Rule 3
Monitor Rule 4
Monitor Rule 5
Monitor Rule 6
Monitor Rule 7
Monitor Rule 8
Sec. Int. Src/Dst
Sec. Int. Layer
File Event Count
Intrusion Event Count
Initiator Country
Responder Country
IOC Number
Source Autonomous System
Destination Autonomous System
SNMP In
SNMP Out
Source TOS
Destination TOS
Source Mask
Destination Mask
Security Context
Security Context, continued
Security Context, continued
Security Context, continued
Referenced Host
VLAN ID
String Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
Referenced Host...


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
User Agent
String Block Type (0)
String Block Length
User Agent...
HTTP Referrer
String Block Type (0)
String Block Length
HTTP Referrer...
SSL Certificate Fingerprint
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Policy ID
SSL Policy ID, continued
SSL Policy ID, continued
SSL Policy ID, continued
SSL Rule ID
SSL Cipher Suite
SSL Version
SSL Srv Cert. Stat.
SSL Srv Cert. Stat., cont.
SSL Actual Action
SSL Expected Action
SSL Expected Action, cont.
SSL Flow Status
SSL Flow Error
SSL Flow Error, continued
SSL Flow Messages
SSL Flow Messages, continued
SSL Flow Flags
SSL Flow Flags, continued


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SSL Server Names
SSL Flow Flags, continued
String Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
SSL Server Name...
SSL URL Category
SSL Session ID
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID Length
SSL Ticket ID
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, cont.
SSL Ticket ID Length
Network Analysis Policy Revision
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued
Network Analysis Policy Revision, continued

The following table describes the fields of the Connection Statistics data block for 5.4+.

Table B-32 Connection Statistics Data Block 5.4+ Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.4+. The value is always `155`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
Destination Mask	uint8	Destination address prefix mask.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
String Block Type	uint32	Initiates a String data block containing the Referenced Host. This value is always 0.
String Block Length	uint32	The number of bytes included in the Referenced Host String data block, including eight bytes for the block type and header fields plus the number of bytes in the Referenced Host field.
Referenced Host	string	Host name information provided in HTTP or DNS.
String Block Type	uint32	Initiates a String data block containing the User Agent. This value is always 0.
String Block Length	uint32	The number of bytes included in the User Agent String data block, including eight bytes for the block type and header fields plus the number of bytes in the User Agent field.
User Agent	string	Information from the UserAgent header field in the session.
String Block Type	uint32	Initiates a String data block containing the HTTP Referrer. This value is always 0.
String Block Length	uint32	The number of bytes included in the HTTP Referrer String data block, including eight bytes for the block type and header fields plus the number of bytes in the HTTP Referrer field.
HTTP Referrer	string	The site from which a page originated. This is found int he Referred header information in HTTP traffic.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.
SSL Policy ID	uint8[16]	ID number of the SSL policy that handled the connection.
SSL Rule ID	uint32	ID number of the SSL rule or default action that handled the connection.
SSL Cipher Suite	uint16	Encryption suite used by the SSL connection. The value is stored in decimal format. See `www.iana.org/assignments/tls-parameters/tls-parameters.xhtml` for the cipher suite designated by the value.
SSL Version	uint8	The SSL or TLS protocol version used to encrypt the connection.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Server Certificate Status	uint16	The status of the SSL certificate. Possible values include: • 0 — Not checked — The server certificate status was not evaluated. • 1 — Unknown — The server certificate status could not be determined. • 2 — Valid — The server certificate is valid. • 4 — Self-signed — The server certificate is self-signed. • 16 — Invalid Issuer — The server certificate has an invalid issuer. • 32 — Invalid Signature — The server certificate has an invalid signature. • 64 — Expired — The server certificate is expired. • 128 — Not valid yet — The server certificate is not yet valid. • 256 — Revoked — The server certificate has been revoked.
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'
SSL Expected Action	uint16	The action which should be performed on the connection based on the SSL Rule. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
SSL Flow Error	uint32	Detailed SSL error code. These values may be needed for support purposes.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Flow Messages	uint32	The messages exchanged between client and server during the SSL handshake. See http://tools.ietf.org/html/rfc5246 for more information. • 0x00000001 — NSE_MT__HELLO_REQUEST • 0x00000002 — NSE_MT__CLIENT_ALERT • 0x00000004 — NSE_MT__SERVER_ALERT • 0x00000008 — NSE_MT__CLIENT_HELLO • 0x00000010 — NSE_MT__SERVER_HELLO • 0x00000020 — NSE_MT__SERVER_CERTIFICATE • 0x00000040 — NSE_MT__SERVER_KEY_EXCHANGE • 0x00000080 — NSE_MT__CERTIFICATE_REQUEST • 0x00000100 — NSE_MT__SERVER_HELLO_DONE • 0x00000200 — NSE_MT__CLIENT_CERTIFICATE • 0x00000400 — NSE_MT__CLIENT_KEY_EXCHANGE • 0x00000800 — NSE_MT__CERTIFICATE_VERIFY • 0x00001000 — NSE_MT__CLIENT_CHANGE_CIPHER_SPEC • 0x00002000 — NSE_MT__CLIENT_FINISHED • 0x00004000 — NSE_MT__SERVER_CHANGE_CIPHER_SPEC • 0x00008000 — NSE_MT__SERVER_FINISHED • 0x00010000 — NSE_MT__NEW_SESSION_TICKET • 0x00020000 — NSE_MT__HANDSHAKE_OTHER • 0x00040000 — NSE_MT__APP_DATA_FROM_CLIENT • 0x00080000 — NSE_MT__APP_DATA_FROM_SERVER
SSL Flow Flags	uint64	The debugging level flags for an encrypted connection. Possible values include: • 0x00000001 — NSE_FLOW__VALID - must be set for other fields to be valid • 0x00000002 — NSE_FLOW__INITIALIZED - internal structures ready for processing • 0x00000004 — NSE_FLOW__INTERCEPT - SSL session has been intercepted
String Block Type	uint32	Initiates a String data block containing the SSL Server Name. This value is always 0.

Table B-32 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the SSL Server Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the SSL Server Name field.
SSL Server Name	string	Name provided in the server name indication in the SSL Client Hello.
SSL URL Category	uint32	Category of the flow as identified from the server name and certificate common name.
SSL Session ID	uint8[32]	Value of the session ID used during the SSL handshake when the client and server agree to do session reuse
SSL Session ID Length	uint8	Length of the SSL Session ID. While the session ID cannot exceed 32 bytes, it may be less than 32 bytes.
SSL Ticket ID	uint8[20]	Hash of the session ticket used when the client and server agree to use a session ticket.
SSL Ticket ID Length	uint8	Length of the SSL Ticket ID. While the ticket ID cannot exceed 20 bytes, it may be less than 20 bytes.
Network Analysis Policy revision	uint8[16]	Revision of the Network Analysis Policy associated with the connection event.

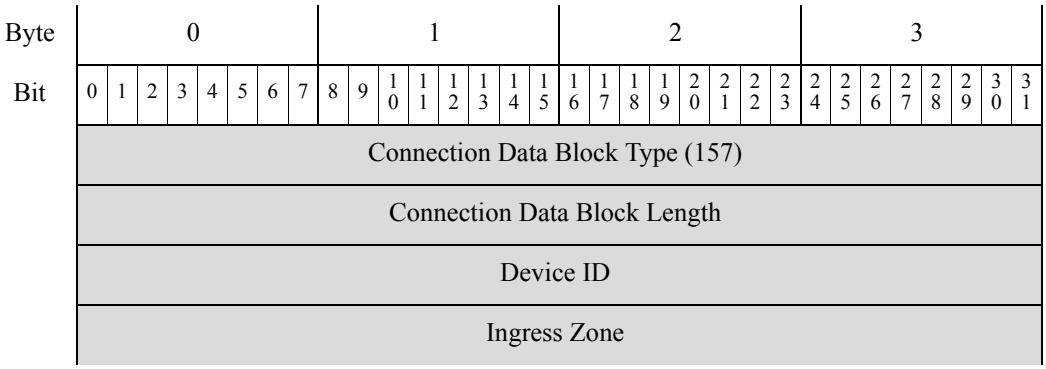
Connection Statistics Data Block 5.4.1

The connection statistics data block is used in connection data messages. Several new fields have been added to the Connection Statistics Data Block for 5.4. Fields have been added to support SSL connections, HTTP redirection, and network analysis policies. The connection statistics data block for version 5.4+ has a block type of 157 in the series 1 group of blocks. It deprecates block type 155, Connection Statistics Data Block 5.3.1, page B-146.

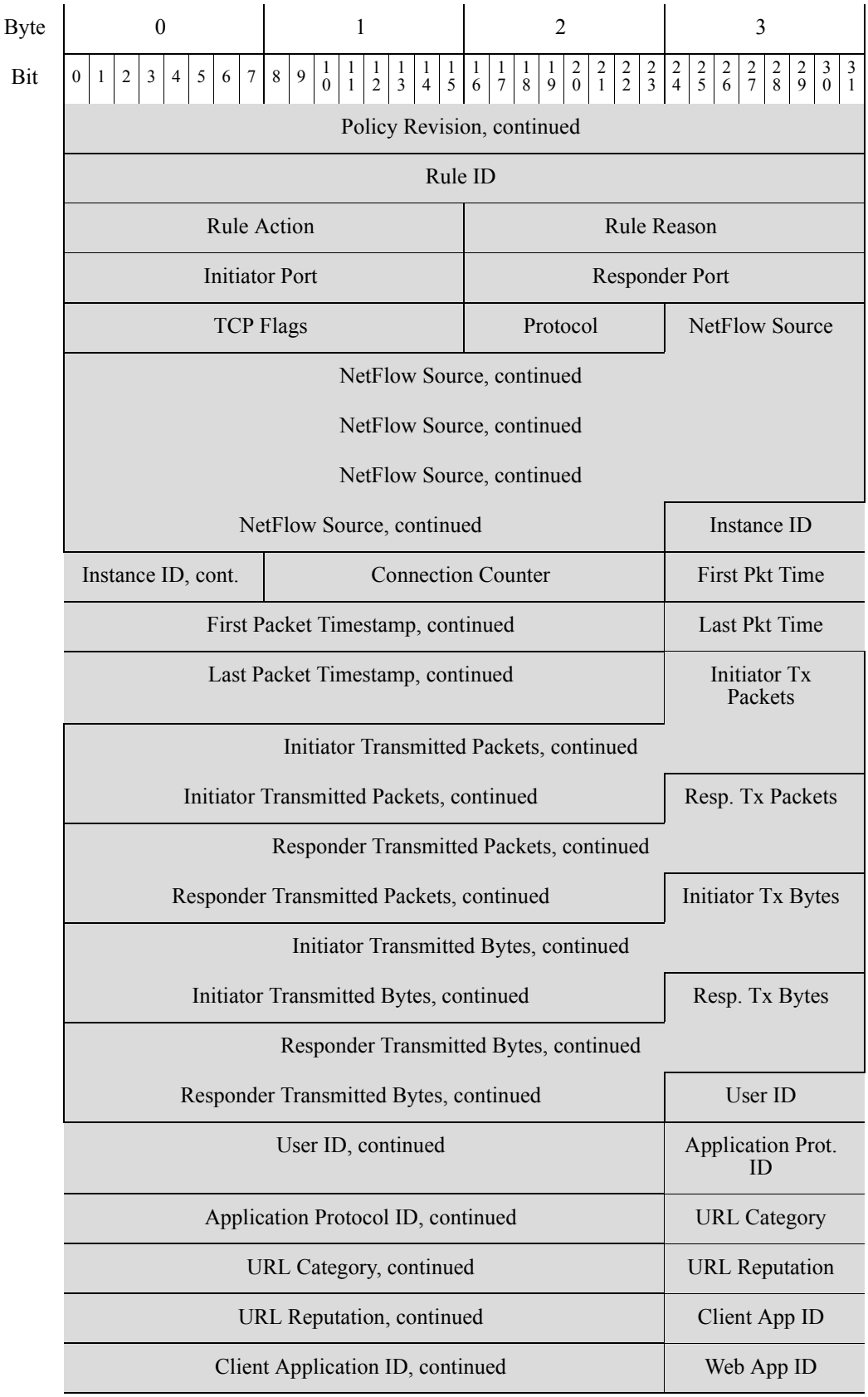
You request connection event records by setting the extended event flag—bit 30 in the Request Flags field—in the request message with an event version of 12 and an event code of 71. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

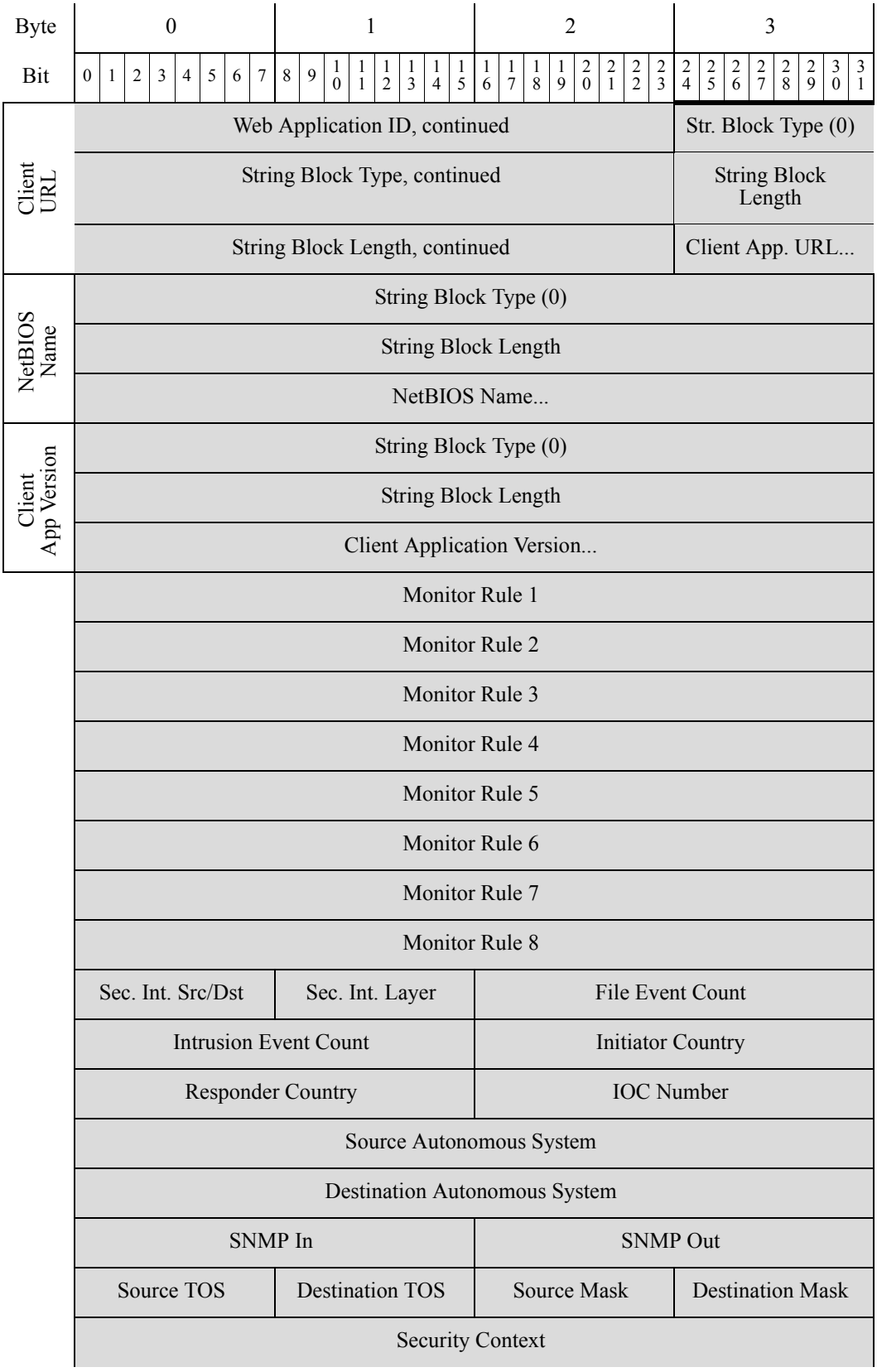
For more information on the Connection Statistics Data message, see Connection Statistics Data Message, page 4-47.

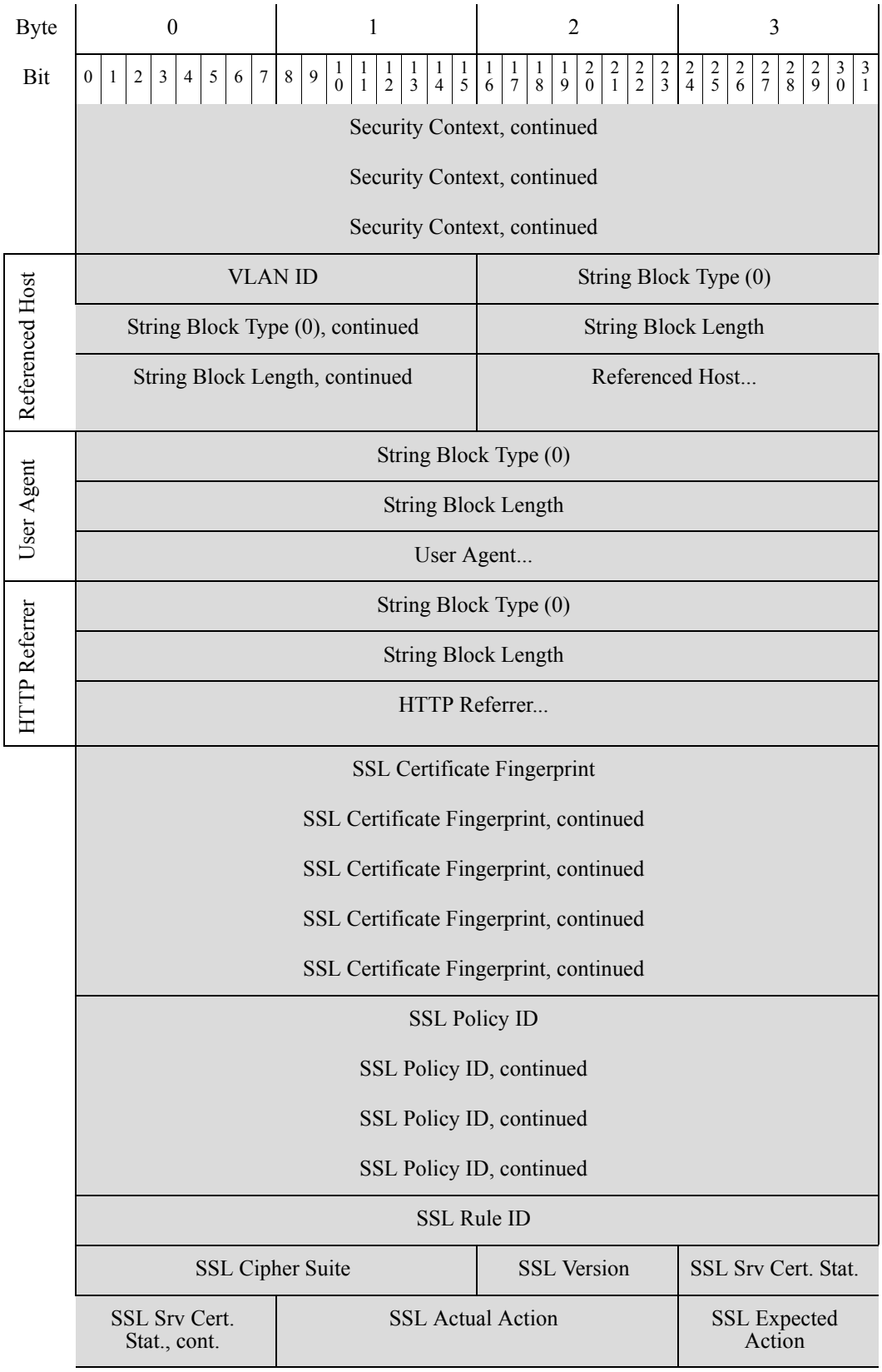
The following diagram shows the format of a Connection Statistics data block for 5.4+:

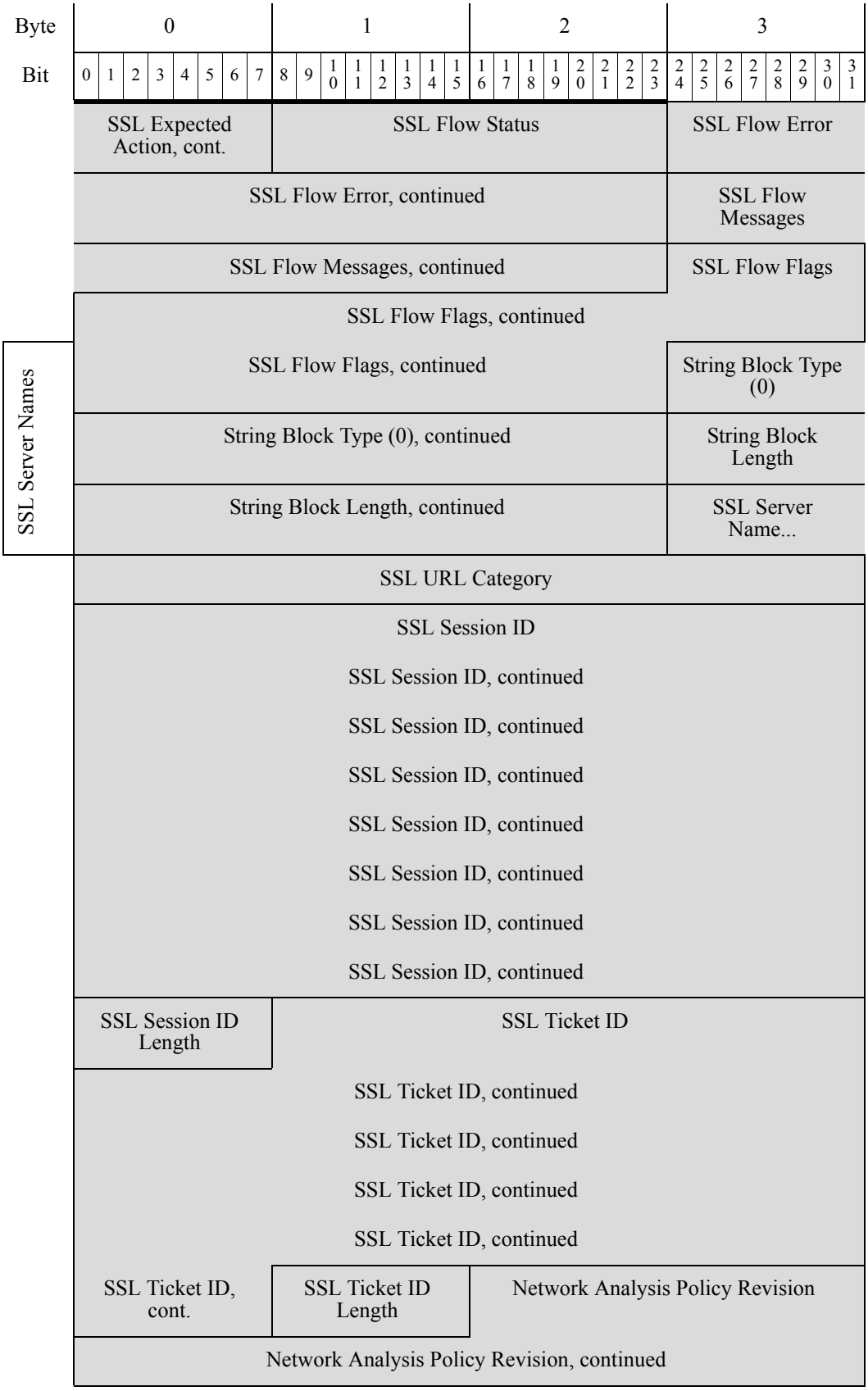


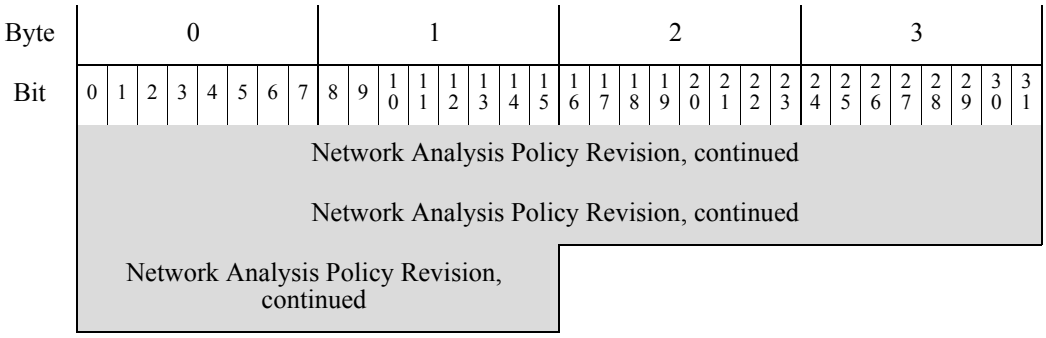
Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ingress Zone, continued																																
Ingress Zone, continued																																
Ingress Zone, continued																																
Egress Zone																																
Egress Zone, continued																																
Egress Zone, continued																																
Egress Zone, continued																																
Ingress Interface																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Ingress Interface, continued																																
Egress Interface																																
Egress Interface, continued																																
Egress Interface, continued																																
Egress Interface, continued																																
Initiator IP Address																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Initiator IP Address, continued																																
Responder IP Address																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Responder IP Address, continued																																
Policy Revision																																
Policy Revision, continued																																
Policy Revision, continued																																


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Policy Revision, continued
Rule ID
Rule Action
Rule Reason
Initiator Port
Responder Port
TCP Flags
Protocol
NetFlow Source
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
NetFlow Source, continued
Instance ID
Instance ID, cont.
Connection Counter
First Pkt Time
First Packet Timestamp, continued
Last Pkt Time
Last Packet Timestamp, continued
Initiator Tx Packets
Initiator Transmitted Packets, continued
Initiator Transmitted Packets, continued
Resp. Tx Packets
Responder Transmitted Packets, continued
Responder Transmitted Packets, continued
Initiator Tx Bytes
Initiator Transmitted Bytes, continued
Initiator Transmitted Bytes, continued
Resp. Tx Bytes
Responder Transmitted Bytes, continued
Responder Transmitted Bytes, continued
User ID
User ID, continued
Application Prot. ID
Application Protocol ID, continued
URL Category
URL Category, continued
URL Reputation
URL Reputation, continued
Client App ID
Client Application ID, continued
Web App ID


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Client URL
Web Application ID, continued
Str. Block Type (0)
String Block Type, continued
String Block Length
String Block Length, continued
Client App. URL...
NetBIOS Name
String Block Type (0)
String Block Length
NetBIOS Name...
Client App Version
String Block Type (0)
String Block Length
Client Application Version...
Monitor Rule 1
Monitor Rule 2
Monitor Rule 3
Monitor Rule 4
Monitor Rule 5
Monitor Rule 6
Monitor Rule 7
Monitor Rule 8
Sec. Int. Src/Dst
Sec. Int. Layer
File Event Count
Intrusion Event Count
Initiator Country
Responder Country
IOC Number
Source Autonomous System
Destination Autonomous System
SNMP In
SNMP Out
Source TOS
Destination TOS
Source Mask
Destination Mask
Security Context


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Security Context, continued
Security Context, continued
Security Context, continued
Referenced Host
VLAN ID
String Block Type (0)
String Block Type (0), continued
String Block Length
String Block Length, continued
Referenced Host...
User Agent
String Block Type (0)
String Block Length
User Agent...
HTTP Referrer
String Block Type (0)
String Block Length
HTTP Referrer...
SSL Certificate Fingerprint
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Certificate Fingerprint, continued
SSL Policy ID
SSL Policy ID, continued
SSL Policy ID, continued
SSL Policy ID, continued
SSL Rule ID
SSL Cipher Suite
SSL Version
SSL Srv Cert. Stat.
SSL Srv Cert. Stat., cont.
SSL Actual Action
SSL Expected Action


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SSL Expected Action, cont.
SSL Flow Status
SSL Flow Error
SSL Flow Error, continued
SSL Flow Messages
SSL Flow Messages, continued
SSL Flow Flags
SSL Flow Flags, continued
SSL Flow Flags, continued
String Block Type (0)
SSL Server Names
String Block Type (0), continued
String Block Length
String Block Length, continued
SSL Server Name...
SSL URL Category
SSL Session ID
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID, continued
SSL Session ID Length
SSL Ticket ID
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, continued
SSL Ticket ID, cont.
SSL Ticket ID Length
Network Analysis Policy Revision
Network Analysis Policy Revision, continued



The following table describes the fields of the Connection Statistics data block for 5.4+.

Table B-33 Connection Statistics Data Block 5.4+ Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.4+. The value is always `157`.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always `0`.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (`/files/index.html`, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.
Destination Mask	uint8	Destination address prefix mask.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
String Block Type	uint32	Initiates a String data block containing the Referenced Host. This value is always 0.
String Block Length	uint32	The number of bytes included in the Referenced Host String data block, including eight bytes for the block type and header fields plus the number of bytes in the Referenced Host field.
Referenced Host	string	Host name information provided in HTTP or DNS.
String Block Type	uint32	Initiates a String data block containing the User Agent. This value is always 0.
String Block Length	uint32	The number of bytes included in the User Agent String data block, including eight bytes for the block type and header fields plus the number of bytes in the User Agent field.
User Agent	string	Information from the UserAgent header field in the session.
String Block Type	uint32	Initiates a String data block containing the HTTP Referrer. This value is always 0.
String Block Length	uint32	The number of bytes included in the HTTP Referrer String data block, including eight bytes for the block type and header fields plus the number of bytes in the HTTP Referrer field.
HTTP Referrer	string	The site from which a page originated. This is found int he Referred header information in HTTP traffic.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.
SSL Policy ID	uint8[16]	ID number of the SSL policy that handled the connection.
SSL Rule ID	uint32	ID number of the SSL rule or default action that handled the connection.
SSL Cipher Suite	uint16	Encryption suite used by the SSL connection. The value is stored in decimal format. See `www.iana.org/assignments/tls-parameters/tls-parameters.xhtml` for the cipher suite designated by the value.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Version	uint8	The SSL or TLS protocol version used to encrypt the connection.
SSL Server Certificate Status	uint16	The status of the SSL certificate. Possible values include: • 0 — Not checked — The server certificate status was not evaluated. • 1 — Unknown — The server certificate status could not be determined. • 2 — Valid — The server certificate is valid. • 4 — Self-signed — The server certificate is self-signed. • 16 — Invalid Issuer — The server certificate has an invalid issuer. • 32 — Invalid Signature — The server certificate has an invalid signature. • 64 — Expired — The server certificate is expired. • 128 — Not valid yet — The server certificate is not yet valid. • 256 — Revoked — The server certificate has been revoked.
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'
SSL Expected Action	uint16	The action which should be performed on the connection based on the SSL Rule. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
SSL Flow Error	uint32	Detailed SSL error code. These values may be needed for support purposes.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
SSL Flow Messages	uint32	The messages exchanged between client and server during the SSL handshake. See http://tools.ietf.org/html/rfc5246 for more information. • 0x00000001 — NSE_MT__HELLO_REQUEST • 0x00000002 — NSE_MT__CLIENT_ALERT • 0x00000004 — NSE_MT__SERVER_ALERT • 0x00000008 — NSE_MT__CLIENT_HELLO • 0x00000010 — NSE_MT__SERVER_HELLO • 0x00000020 — NSE_MT__SERVER_CERTIFICATE • 0x00000040 — NSE_MT__SERVER_KEY_EXCHANGE • 0x00000080 — NSE_MT__CERTIFICATE_REQUEST • 0x00000100 — NSE_MT__SERVER_HELLO_DONE • 0x00000200 — NSE_MT__CLIENT_CERTIFICATE • 0x00000400 — NSE_MT__CLIENT_KEY_EXCHANGE • 0x00000800 — NSE_MT__CERTIFICATE_VERIFY • 0x00001000 — NSE_MT__CLIENT_CHANGE_CIPHER_SPEC • 0x00002000 — NSE_MT__CLIENT_FINISHED • 0x00004000 — NSE_MT__SERVER_CHANGE_CIPHER_SPEC • 0x00008000 — NSE_MT__SERVER_FINISHED • 0x00010000 — NSE_MT__NEW_SESSION_TICKET • 0x00020000 — NSE_MT__HANDSHAKE_OTHER • 0x00040000 — NSE_MT__APP_DATA_FROM_CLIENT • 0x00080000 — NSE_MT__APP_DATA_FROM_SERVER
SSL Flow Flags	uint64	The debugging level flags for an encrypted connection. Possible values include: • 0x00000001 — NSE_FLOW__VALID - must be set for other fields to be valid • 0x00000002 — NSE_FLOW__INITIALIZED - internal structures ready for processing • 0x00000004 — NSE_FLOW__INTERCEPT - SSL session has been intercepted
String Block Type	uint32	Initiates a String data block containing the SSL Server Name. This value is always 0.

Table B-33 Connection Statistics Data Block 5.4+ Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the SSL Server Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the SSL Server Name field.
SSL Server Name	string	Name provided in the server name indication in the SSL Client Hello.
SSL URL Category	uint32	Category of the flow as identified from the server name and certificate common name.
SSL Session ID	uint8[32]	Value of the session ID used during the SSL handshake when the client and server agree to do session reuse
SSL Session ID Length	uint8	Length of the SSL Session ID. While the session ID cannot exceed 32 bytes, it may be less than 32 bytes.
SSL Ticket ID	uint8[20]	Hash of the session ticket used when the client and server agree to use a session ticket.
SSL Ticket ID Length	uint8	Length of the SSL Ticket ID. While the ticket ID cannot exceed 20 bytes, it may be less than 20 bytes.
Network Analysis Policy revision	uint8[16]	Revision of the Network Analysis Policy associated with the connection event.

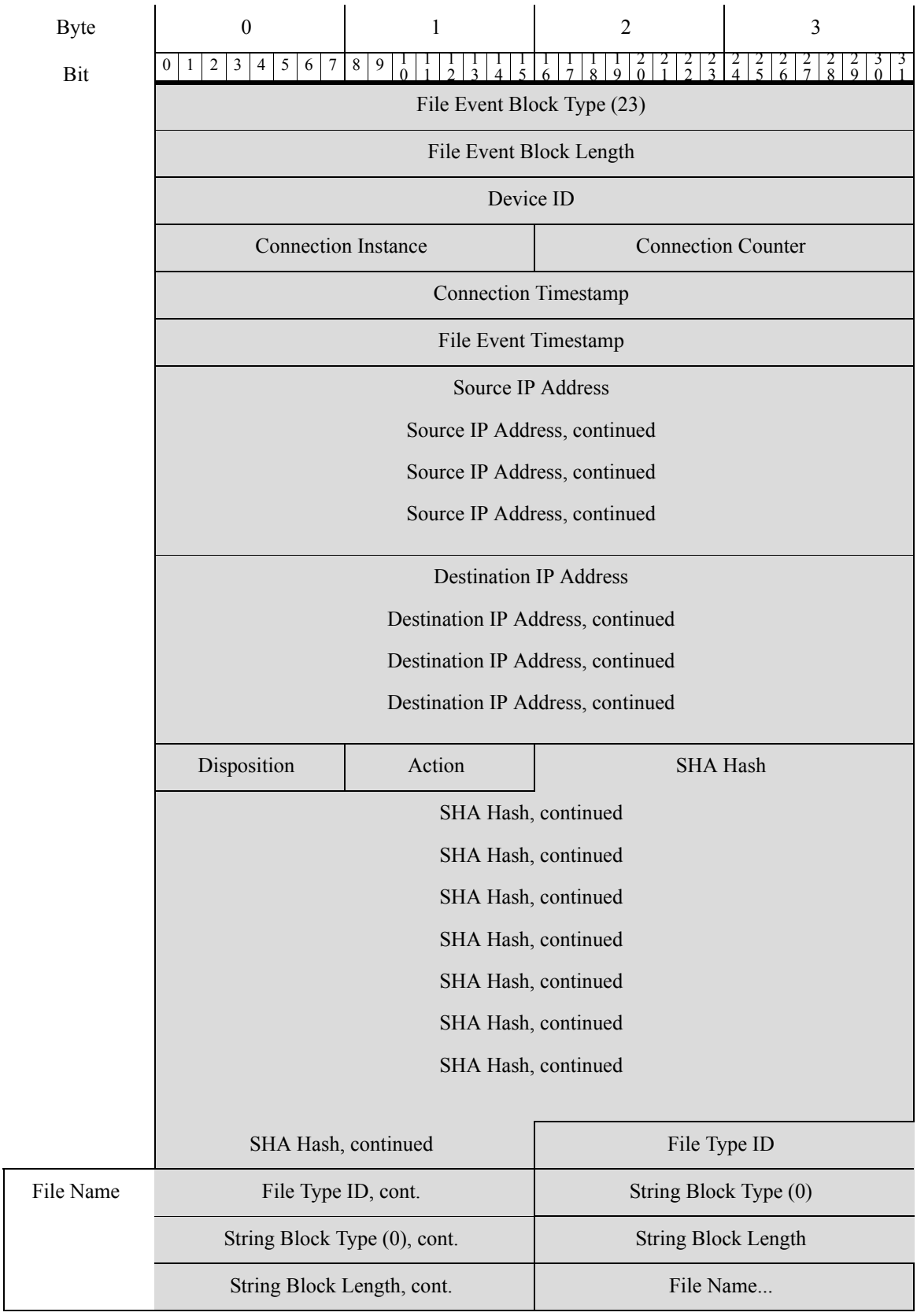
Legacy File Event Data Structures

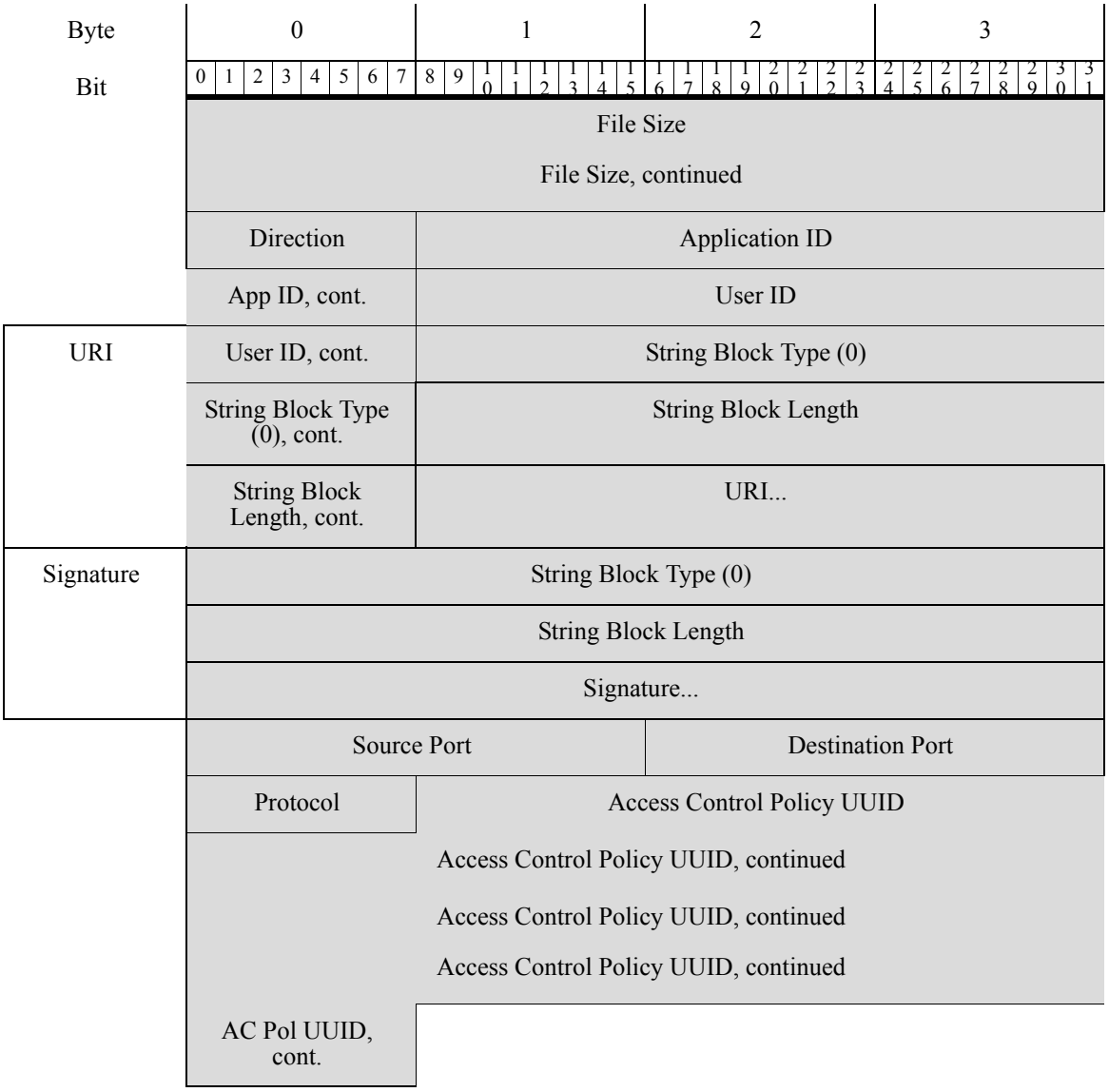
The following topics describe other legacy file event data structures:

File Event for 5.1.1.x

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 23 in the series 2 group of blocks.

The following graphic shows the structure of the File Event data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
File Event Block Type (23)
File Event Block Length
Device ID
Connection Instance
Connection Counter
Connection Timestamp
File Event Timestamp
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Disposition
Action
SHA Hash
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
File Type ID
File Name
File Type ID, cont.
String Block Type (0)
String Block Type (0), cont.
String Block Length
String Block Length, cont.
File Name...



The following table describes the fields in the file event data block:

Table B-34 File Event Data Block Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always `23`.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.

Table B-34 File Event Data Block Fields (continued)

Field	Data Type	Description
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Cisco cloud for a disposition. • 5 — NO_CLOUD_RESP — The Cisco cloud services did not respond to the request.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.

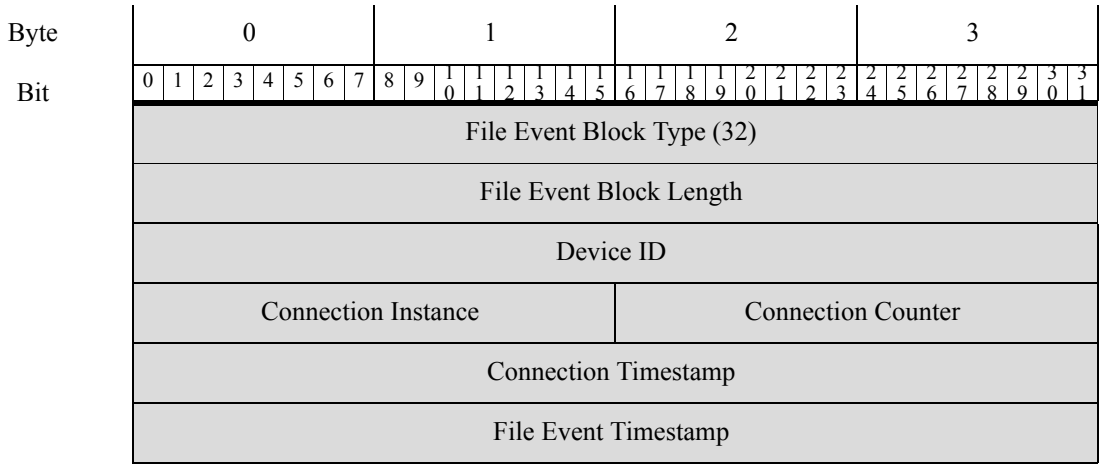
Table B-34 File Event Data Block Fields (continued)

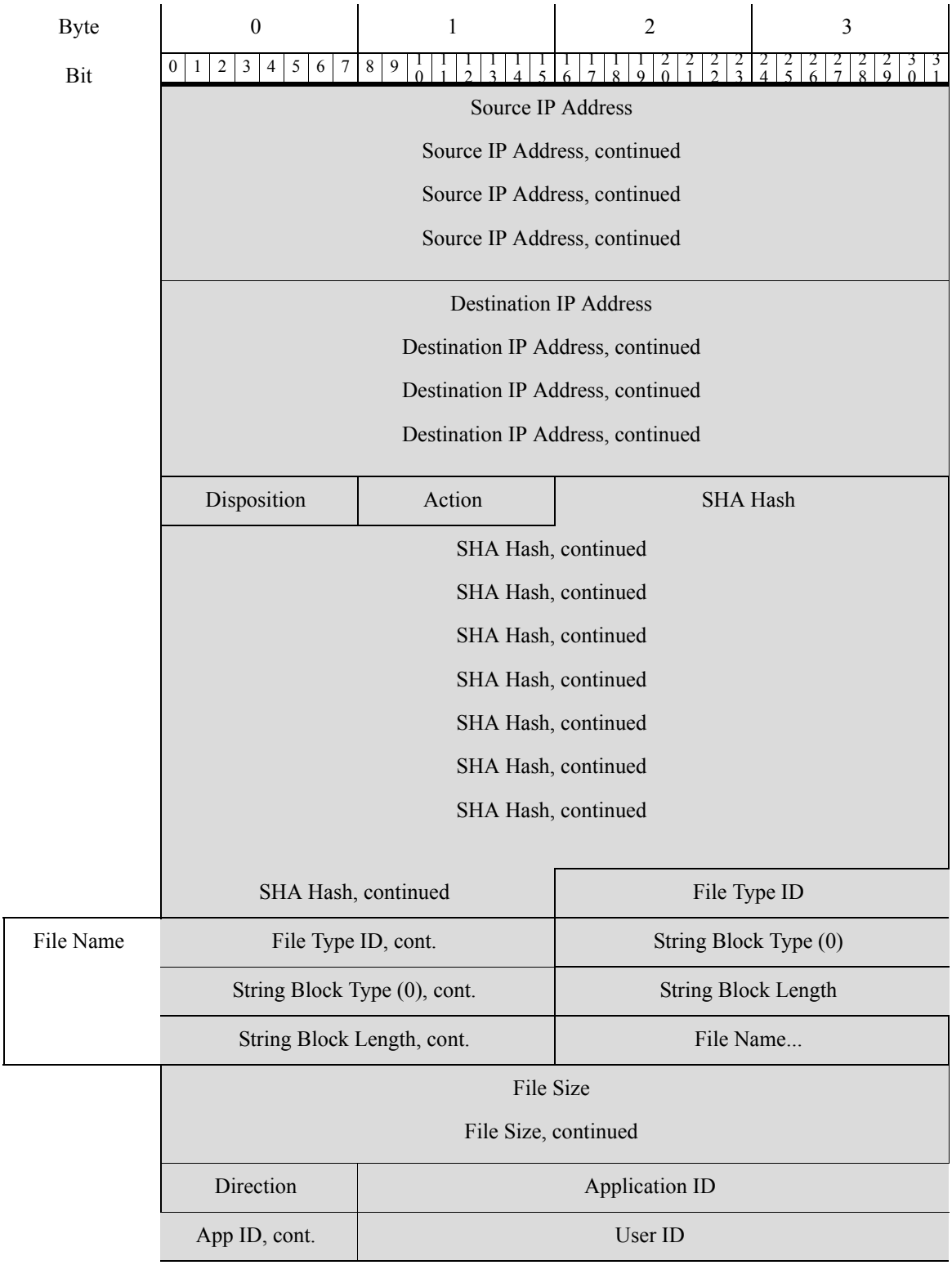
Field	Data Type	Description
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.

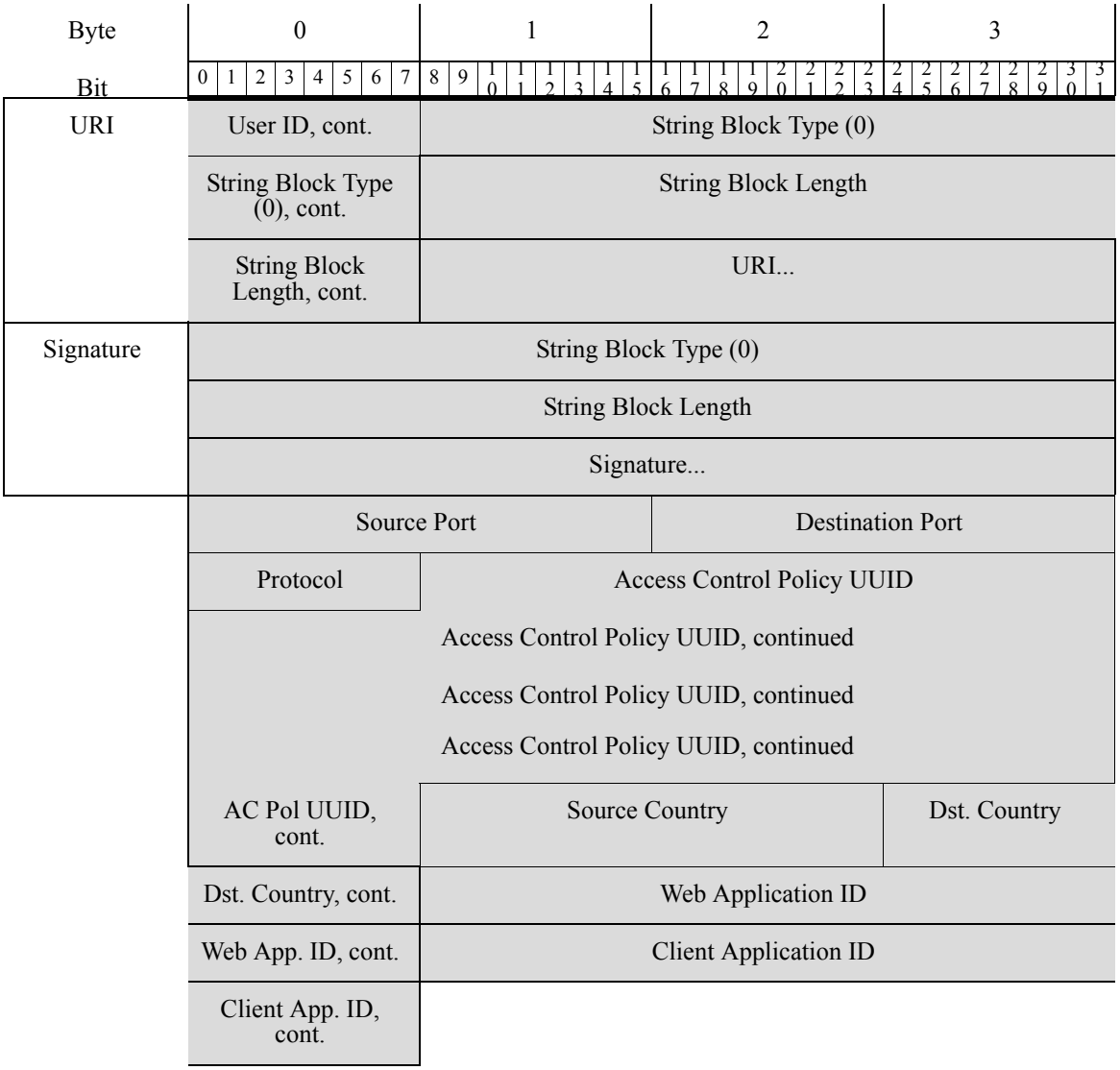
File Event for 5.2.x

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 32 in the series 2 group of blocks. It supersedes block type 23. New fields have been added to track source and destination country, as well as the client and web application instances.

The following graphic shows the structure of the File Event data block:




Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued
Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued
Disposition
Action
SHA Hash
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
File Type ID
File Name
File Type ID, cont.
String Block Type (0)
String Block Type (0), cont.
String Block Length
String Block Length, cont.
File Name...
File Size
File Size, continued
Direction
Application ID
App ID, cont.
User ID



The following table describes the fields in the file event data block:

Table B-35 File Event Data Block Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always `23`.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.

Table B-35 File Event Data Block Fields (continued)

Field	Data Type	Description
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — NEUTRAL — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.

Table B-35 File Event Data Block Fields (continued)

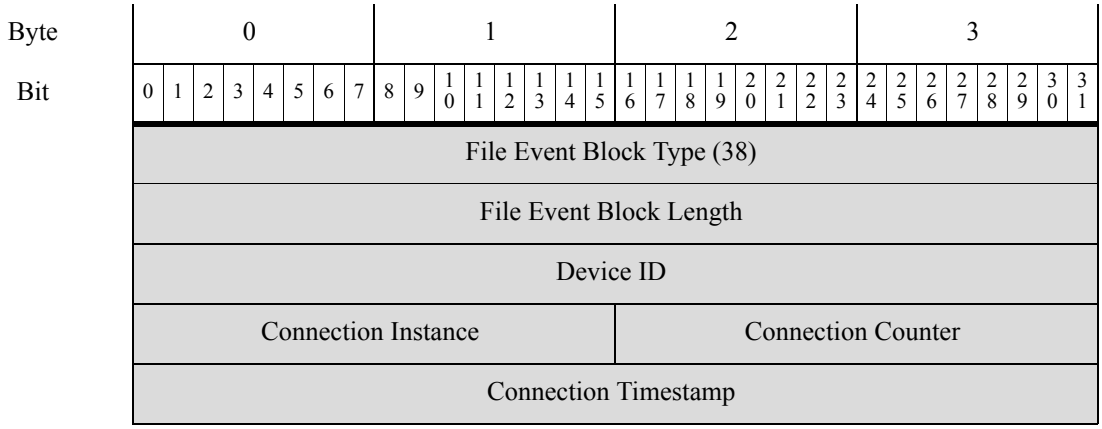
Field	Data Type	Description
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.

File Event for 5.3

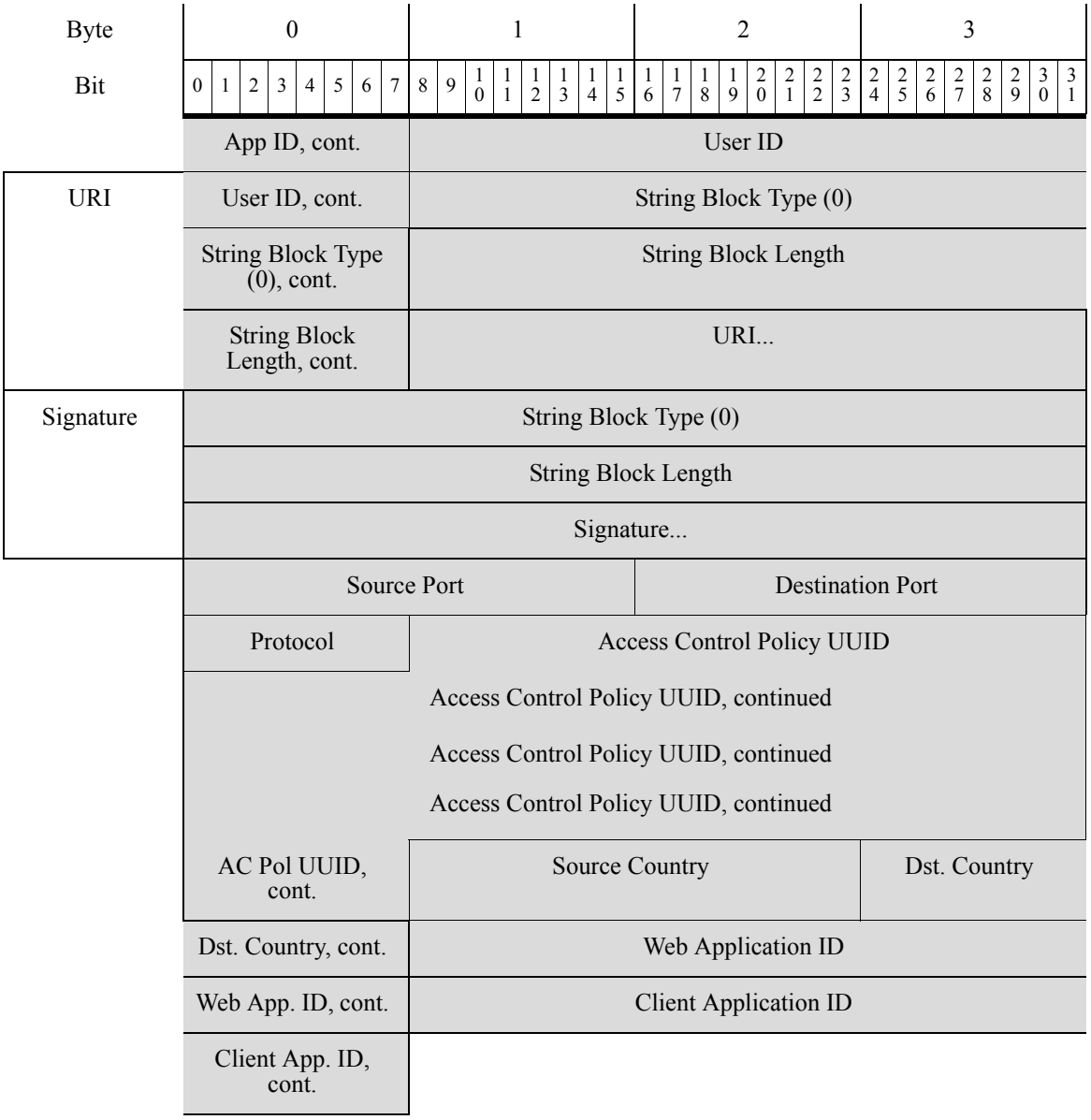
The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 38 in the series 2 group of blocks. It supersedes block type 32. New fields have been added to track dynamic file analysis and file storage.

You request file event records by setting the file event flag—bit 30 in the Request Flags field—in the request message with an event version of 3 and an event code of 111. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

The following graphic shows the structure of the File Event data block.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	File Event Timestamp																															
	Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																															
	Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																															
	Disposition								SPERO Disposition								File Storage Status								File Analysis Status							
	Archive File Status								Threat Score								Action								SHA Hash							
	SHA Hash, continued SHA Hash, continued SHA Hash, continued SHA Hash, continued SHA Hash, continued SHA Hash, continued SHA Hash, continued																															
	SHA Hash, continued																								File Type ID							
File Name	File Type ID, cont.																								String Block Type (0)							
	String Block Type (0), cont.																								String Block Length							
	String Block Length, cont.																								File Name...							
	File Size File Size, continued																															
	Direction								Application ID																							



The following table describes the fields in the file event data block.

Table B-36 File Event Data Block Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always 23.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.

Table B-36 File Event Data Block Fields (continued)

Field	Data Type	Description
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
SPERO Disposition	uint8	Indicates whether the SPERO signature was used in file analysis. If the value is 1, 2, or 3, SPERO analysis was used. If there is any other value SPERO analysis was not used.

Table B-36 File Event Data Block Fields (continued)

Field	Data Type	Description
File Storage Status	uint8	The storage status of the file. Possible values are: • 1 — File Stored • 2 — File Stored • 3 — Unable to Store File • 4 — Unable to Store File • 5 — Unable to Store File • 6 — Unable to Store File • 7 — Unable to Store File • 8 — File Size is Too Large • 9 — File Size is Too Small • 10 — Unable to Store File • 11 — File Not Stored, Disposition Unavailable

Table B-36 File Event Data Block Fields (continued)

Field	Data Type	Description
File Analysis Status	uint8	Indicates whether the file was sent for dynamic analysis. Possible values are: • 0 — File Not Sent for Analysis • 1 — Sent for Analysis • 2 — Sent for Analysis • 4 — Sent for Analysis • 5 — Failed to Send • 6 — Failed to Send • 7 — Failed to Send • 8 — Failed to Send • 9 — File Size is Too Small • 10 — File Size is Too Large • 11 — Sent for Analysis • 12 — Analysis Complete • 13 — Failure (Network Issue) • 14 — Failure (Rate Limit) • 15 — Failure (File Too Large) • 16 — Failure (File Read Error) • 17 — Failure (Internal Library Error) • 19 — File Not Sent, Disposition Unavailable • 20 — Failure (Cannot Run File) • 21 — Failure (Analysis Timeout) • 22 — Sent for Analysis • 23 — File Not Supported
Archive File Status	uint8	This is always 0.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.

Table B-36 File Event Data Block Fields (continued)

Field	Data Type	Description
File Type ID	uint32	ID number that maps to the file type. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.

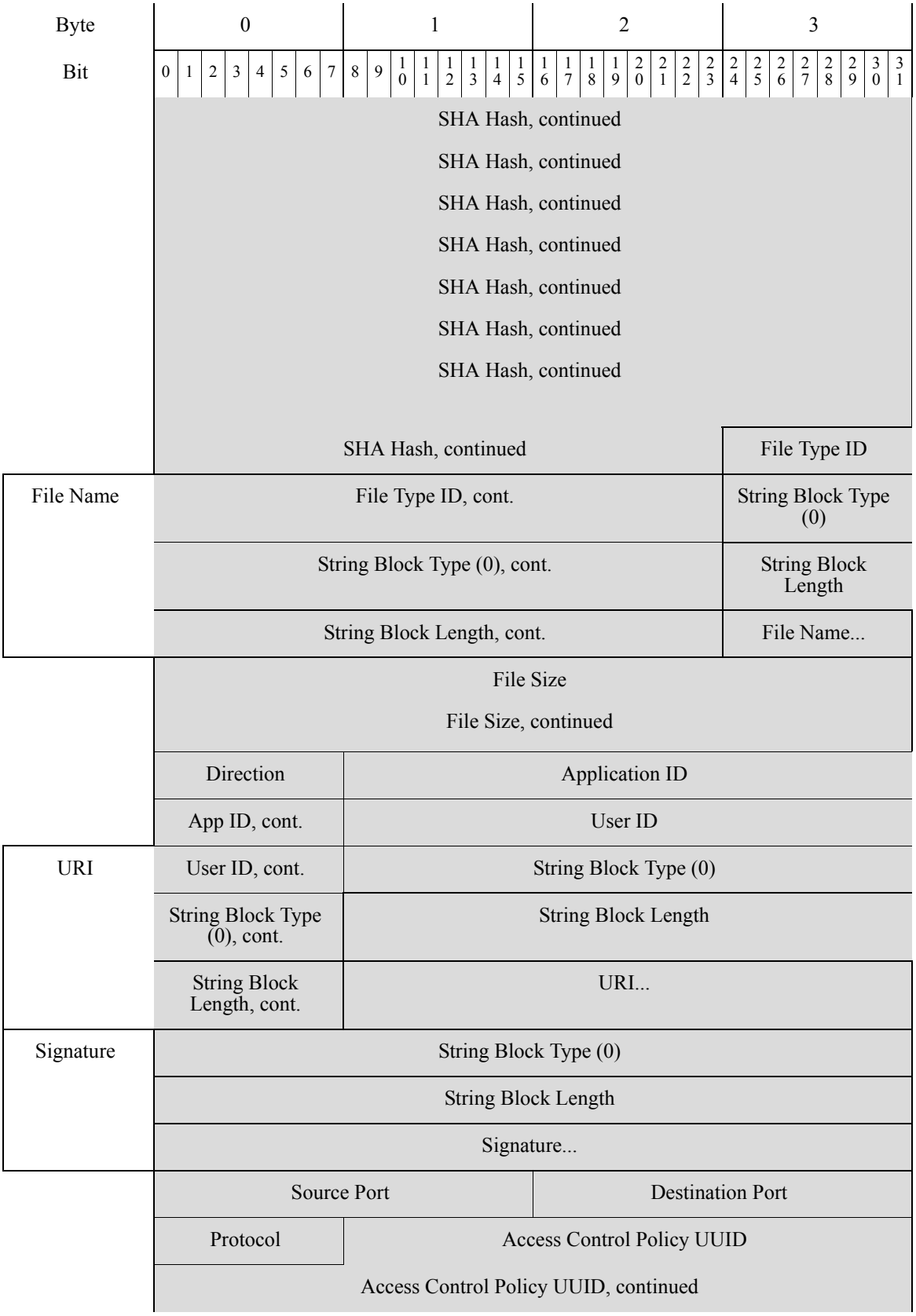
File Event for 5.3.1

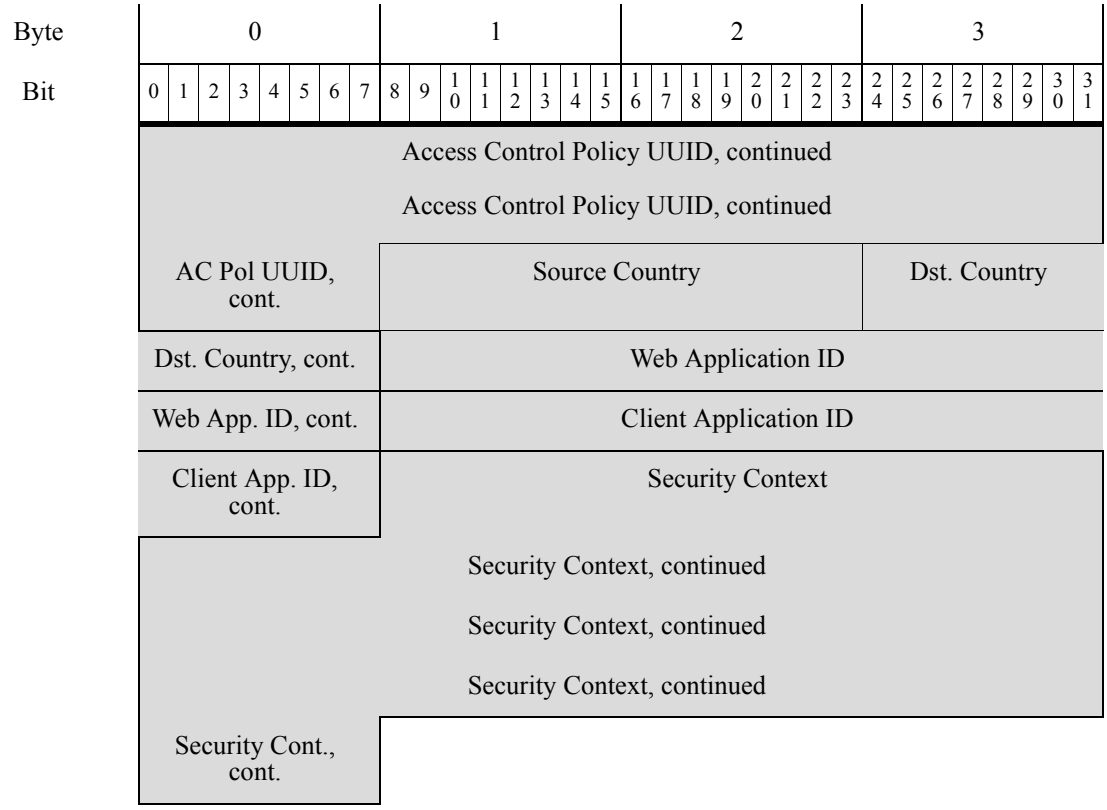
The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 43 in the series 2 group of blocks. It supersedes block type 38. A security context field has been added.

You request file event records by setting the file event flag—bit 30 in the Request Flags field—in the request message with an event version of 4 and an event code of 111. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

The following graphic shows the structure of the File Event data block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
File Event Block Type (43)																																
File Event Block Length																																
Device ID																																
Connection Instance																	Connection Counter															
Connection Timestamp																																
File Event Timestamp																																
Source IP Address Source IP Address, continued Source IP Address, continued Source IP Address, continued																																
Destination IP Address Destination IP Address, continued Destination IP Address, continued Destination IP Address, continued																																
Disposition									SPERO Disposition								File Storage Status								File Analysis Status							
Archive File Status									Threat Score								Action								SHA Hash							


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
SHA Hash, continued
File Type ID
File Name
File Type ID, cont.
String Block Type (0)
String Block Type (0), cont.
String Block Length
String Block Length, cont.
File Name...
File Size
File Size, continued
Direction
Application ID
App ID, cont.
User ID
URI
User ID, cont.
String Block Type (0)
String Block Type (0), cont.
String Block Length
String Block Length, cont.
URI...
Signature
String Block Type (0)
String Block Length
Signature...
Source Port
Destination Port
Protocol
Access Control Policy UUID
Access Control Policy UUID, continued



The following table describes the fields in the file event data block.

Table B-37 File Event Data Block Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always `43`.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.

Table B-37 File Event Data Block Fields (continued)

Field	Data Type	Description
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
SPERO Disposition	uint8	Indicates whether the SPERO signature was used in file analysis. If the value is 1, 2, or 3, SPERO analysis was used. If there is any other value SPERO analysis was not used.
File Storage Status	uint8	The storage status of the file. Possible values are: • 1 — File Stored • 2 — File Stored • 3 — Unable to Store File • 4 — Unable to Store File • 5 — Unable to Store File • 6 — Unable to Store File • 7 — Unable to Store File • 8 — File Size is Too Large • 9 — File Size is Too Small • 10 — Unable to Store File • 11 — File Not Stored, Disposition Unavailable

Table B-37 File Event Data Block Fields (continued)

Field	Data Type	Description
File Analysis Status	uint8	Indicates whether the file was sent for dynamic analysis. Possible values are: • 0 — File Not Sent for Analysis • 1 — Sent for Analysis • 2 — Sent for Analysis • 4 — Sent for Analysis • 5 — Failed to Send • 6 — Failed to Send • 7 — Failed to Send • 8 — Failed to Send • 9 — File Size is Too Small • 10 — File Size is Too Large • 11 — Sent for Analysis • 12 — Analysis Complete • 13 — Failure (Network Issue) • 14 — Failure (Rate Limit) • 15 — Failure (File Too Large) • 16 — Failure (File Read Error) • 17 — Failure (Internal Library Error) • 19 — File Not Sent, Disposition Unavailable • 20 — Failure (Cannot Run File) • 21 — Failure (Analysis Timeout) • 22 — Sent for Analysis • 23 — File Not Supported • 23 —File Transmit File Capacity Handled — File capacity handled (stored on the sensor) because file could not be submitted to the sandbox for analysis • 25 — File Transmit Server Limited Exceeded Capacity Handled — File capacity handled due to rate limiting on server • 26 — Communication Failure — File capacity handled due to cloud connectivity failure • 27 — Not Sent — File not sent due to configuration • 28 — Preclass No Match —File not sent for dynamic analysis since pre-classification didn't find any embedded or suspicious object in the file • 29 — Transmit Sent Sandbox Private Cloud — File sent to the private cloud for dynamic analysis • 30 — Transmit Not Send Sendbox Private Cloud - File not send to the private cloud for analysis

Table B-37 File Event Data Block Fields (continued)

Field	Data Type	Description
Archive File Status	uint8	This is always `0`.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • `1` — Detect • `2` — Block • `3` — Malware Cloud Lookup • `4` — Malware Block • `5` — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • `1` — ICMP • `4` — IP • `6` — TCP • `17` — UDP This is currently only TCP.

Table B-37 File Event Data Block Fields (continued)

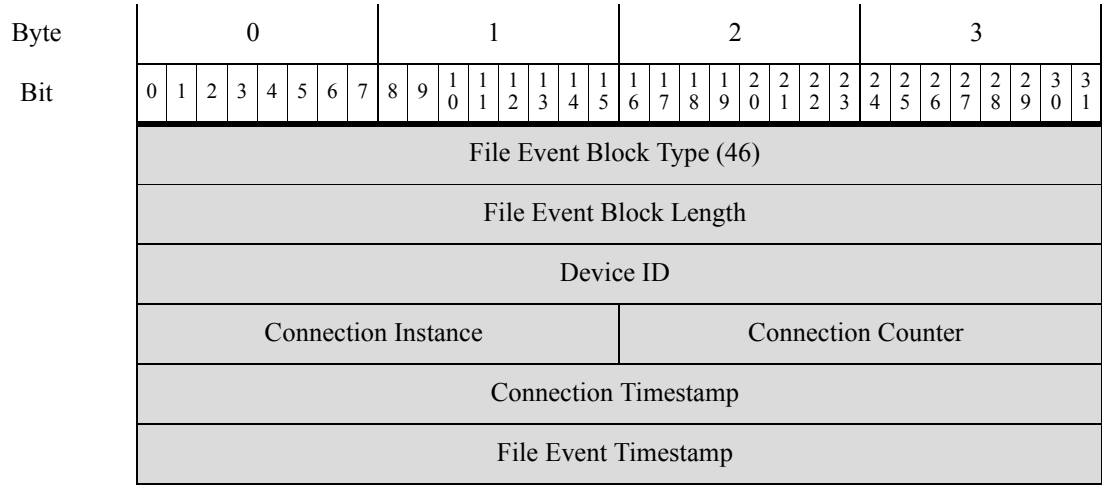
Field	Data Type	Description
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.

File Event for 5.4.x

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 46 in the series 2 group of blocks. It supersedes block type 43. Fields for SSL and file archive support have been added.

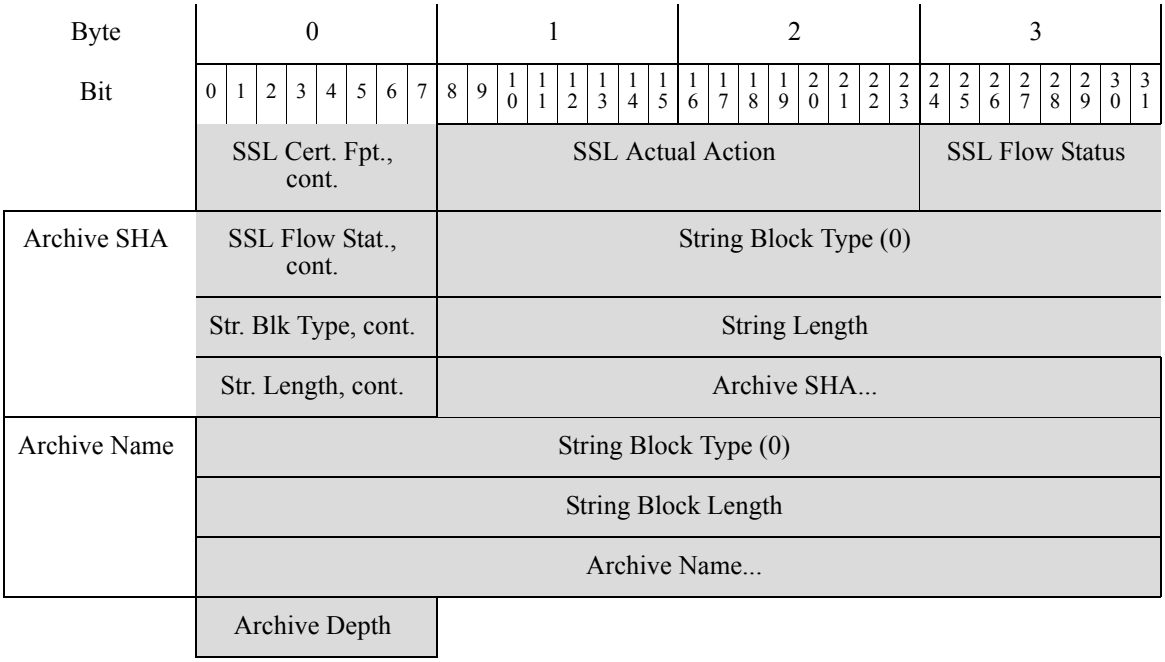
You request file event records by setting the file event flag—bit 30 in the Request Flags field—in the request message with an event version of 5 and an event code of 111. See Request Flags, page 2-11. If you enable bit 23, an extended event header is included in the record.

The following graphic shows the structure of the File Event data block.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Source IP Address			
	Source IP Address, continued			
	Source IP Address, continued			
	Source IP Address, continued			
	Destination IP Address			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Disposition	SPERO Disposition	File Storage Status	File Analysis Status
	Archive File Status	Threat Score	Action	SHA Hash
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			File Type ID
File Name	File Type ID, cont.			String Block Type (0)
	String Block Type (0), cont.			String Block Length
	String Block Length, cont.			File Name...
	File Size			
	File Size, continued			
	Direction	Application ID		
	App ID, cont.	User ID		

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
URI	User ID, cont.								String Block Type (0)																							
	String Block Type (0), cont.								String Block Length																							
	String Block Length, cont.								URI...																							
Signature	String Block Type (0)																															
	String Block Length																															
	Signature...																															
	Source Port																Destination Port															
	Protocol								Access Control Policy UUID																							
	Access Control Policy UUID, continued																															
	Access Control Policy UUID, continued																															
	Access Control Policy UUID, continued																															
	AC Pol UUID, cont.								Source Country																Dst. Country							
	Dst. Country, cont.								Web Application ID																							
	Web App. ID, cont.								Client Application ID																							
	Client App. ID, cont.								Security Context																							
	Security Context, continued																															
	Security Context, continued																															
	Security Context, continued																															
	Security Cont., cont.								SSL Certificate Fingerprint																							
	SSL Certificate Fingerprint, continued																															
	SSL Certificate Fingerprint, continued																															
	SSL Certificate Fingerprint, continued																															
	SSL Certificate Fingerprint, continued																															



The following table describes the fields in the file event data block.

Table B-38 File Event Data Block for 5.4.x Fields

Field	Data Type	Description
File Event Block Type	uint32	Initiates whether file event data block. This value is always 46.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN The file is clean and does not contain malware. • 2 — UNKNOWN It is unknown whether the file contains malware. • 3 — MALWARE The file contains malware. • 4 — UNAVAILABLE The software was unable to send a request to the Cisco cloud for a disposition, or the Cisco cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE The file matches a user-defined hash, and is treated in a fashion designated by the user.
SPERO Disposition	uint8	Indicates whether the SPERO signature was used in file analysis. If the value is 1, 2, or 3, SPERO analysis was used. If there is any other value SPERO analysis was not used.
File Storage Status	uint8	The storage status of the file. Possible values are: • 1 — File Stored • 2 — File Stored • 3 — Unable to Store File • 4 — Unable to Store File • 5 — Unable to Store File • 6 — Unable to Store File • 7 — Unable to Store File • 8 — File Size is Too Large • 9 — File Size is Too Small • 10 — Unable to Store File • 11 — File Not Stored, Disposition Unavailable

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
File Analysis Status	uint8	Indicates whether the file was sent for dynamic analysis. Possible values are: • 0 — File Not Sent for Analysis • 1 — Sent for Analysis • 2 — Sent for Analysis • 4 — Sent for Analysis • 5 — Failed to Send • 6 — Failed to Send • 7 — Failed to Send • 8 — Failed to Send • 9 — File Size is Too Small • 10 — File Size is Too Large • 11 — Sent for Analysis • 12 — Analysis Complete • 13 — Failure (Network Issue) • 14 — Failure (Rate Limit) • 15 — Failure (File Too Large) • 16 — Failure (File Read Error) • 17 — Failure (Internal Library Error) • 19 — File Not Sent, Disposition Unavailable • 20 — Failure (Cannot Run File) • 21 — Failure (Analysis Timeout) • 22 — Sent for Analysis • 23 — File Not Supported

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
Archive File Status	uint8	The status of an archive being inspected. Can have the following values: • 0 — N/A — File is not being inspected as an archive • 1 — Pending — Archive is being inspected • 2 — Extracted — Successfully inspected without any problems • 3 — Failed — Failed to inspect, insufficient system resources • 4 — Depth Exceeded — Successful, but archive exceeded the nested inspection depth • 5 — Encrypted — Partially Successful, Archive was or contains an archive that is encrypted • 6 — Not Inspectable — Partially Successful, File is possibly Malformed or Corrupt
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist • 6 — Cloud Lookup Timeout • 7 — Custom Detection • 8 — Custom Detection Block • 9 — Archive Block (Depth Exceeded) • 10 — Archive Block (Encrypted) • 11 — Archive Block (Failed to Inspect)
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type. The meaning of this field is transmitted in the metadata with this event. See AMP for Endpoints File Type Metadata, page 3-39 for more information.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Security Context	uint8(16)	ID number for the security context (virtual firewall) that the traffic passed through. Note that the system only populates this field for ASA FirePOWER devices in multi-context mode.
SSL Certificate Fingerprint	uint8[20]	SHA1 hash of the SSL Server certificate.

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
SSL Actual Action	uint16	The action performed on the connection based on the SSL Rule. This may differ from the expected action, as the action as specified in the rule may be impossible. Possible values include: • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)'

Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
SSL Flow Status	uint16	Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include: • 0 — 'Unknown' • 1 — 'No Match' • 2 — 'Success' • 3 — 'Uncached Session' • 4 — 'Unknown Cipher Suite' • 5 — 'Unsupported Cipher Suite' • 6 — 'Unsupported SSL Version' • 7 — 'SSL Compression Used' • 8 — 'Session Undecryptable in Passive Mode' • 9 — 'Handshake Error' • 10 — 'Decryption Error' • 11 — 'Pending Server Name Category Lookup' • 12 — 'Pending Common Name Category Lookup' • 13 — 'Internal Error' • 14 — 'Network Parameters Unavailable' • 15 — 'Invalid Server Certificate Handle' • 16 — 'Server Certificate Fingerprint Unavailable' • 17 — 'Cannot Cache Subject DN' • 18 — 'Cannot Cache Issuer DN' • 19 — 'Unknown SSL Version' • 20 — 'External Certificate List Unavailable' • 21 — 'External Certificate Fingerprint Unavailable' • 22 — 'Internal Certificate List Invalid' • 23 — 'Internal Certificate List Unavailable' • 24 — 'Internal Certificate Unavailable' • 25 — 'Internal Certificate Fingerprint Unavailable' • 26 — 'Server Certificate Validation Unavailable' • 27 — 'Server Certificate Validation Failure' • 28 — 'Invalid Action'
String Block Type	uint32	Initiates a String data block containing the Archive SHA. This value is always 0.

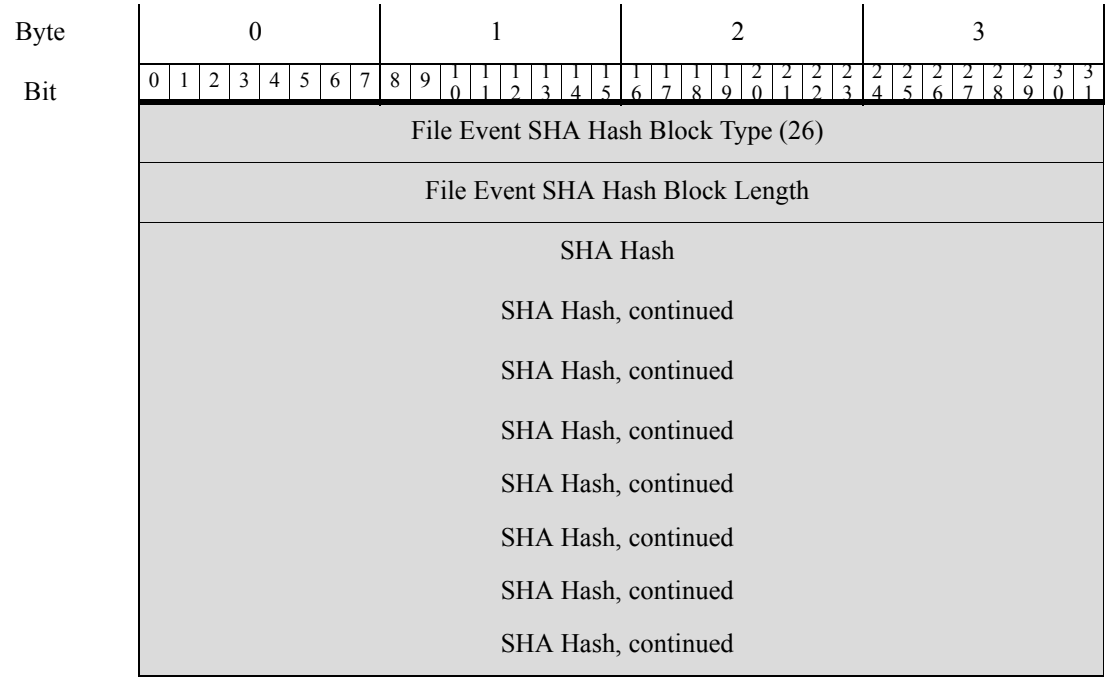
Table B-38 File Event Data Block for 5.4.x Fields (continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Archive SHA String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive SHA	string	SHA1 hash of the parent archive in which the file is contained.
String Block Type	uint32	Initiates a String data block containing the Archive Name. This value is always `0`.
String Block Length	uint32	The number of bytes included in the Archive Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Archive Name	string	Name of the parent archive.
Archive Depth	uint8	Number of layers in which the file is nested. For example, if a text file is in a zip archive, this has a value of `1`.

File Event SHA Hash for 5.1.1-5.2.x

The eStreamer service uses the File Event SHA Hash data block to contain metadata of the mapping of the SHA hash of a file to its filename. The block type is 26 in the series 2 list of data blocks. It can be requested if file log events have been requested in the extended requests—event code 111—and either bit 20 is set or metadata is requested with an event version of 4 and an event code of 21.

The following diagram shows the structure of a file event hash data block:



File Name	String Block Type (0)
	String Block Length
	File Name or Disposition...

The following table describes the fields in the file event SHA hash data block.

Table B-39 File Event SHA Hash 5.1.1-5.2.x Data Block Fields

Field	Data Type	Description
File Event SHA Hash Block Type	uint32	Initiates a File Event SHA Hash block. This value is always `26`.
File Event SHA Hash Block Length	uint32	Total number of bytes in the File Event SHA Hash block, including eight bytes for the File Event SHA Hash block type and length fields, plus the number of bytes of data that follows.
SHA Hash	uint8[32]	The SHA-256 hash of the file in binary format.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the file. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
File Name or Disposition	string	The descriptive name or disposition of the file. If the file is clean, this value is `Clean`. If the file's disposition is unknown, the value is `Neutral`. If the file contains malware, the file name is given.

Legacy Correlation Event Data Structures

The following topics describe other legacy correlation (compliance) data structures:

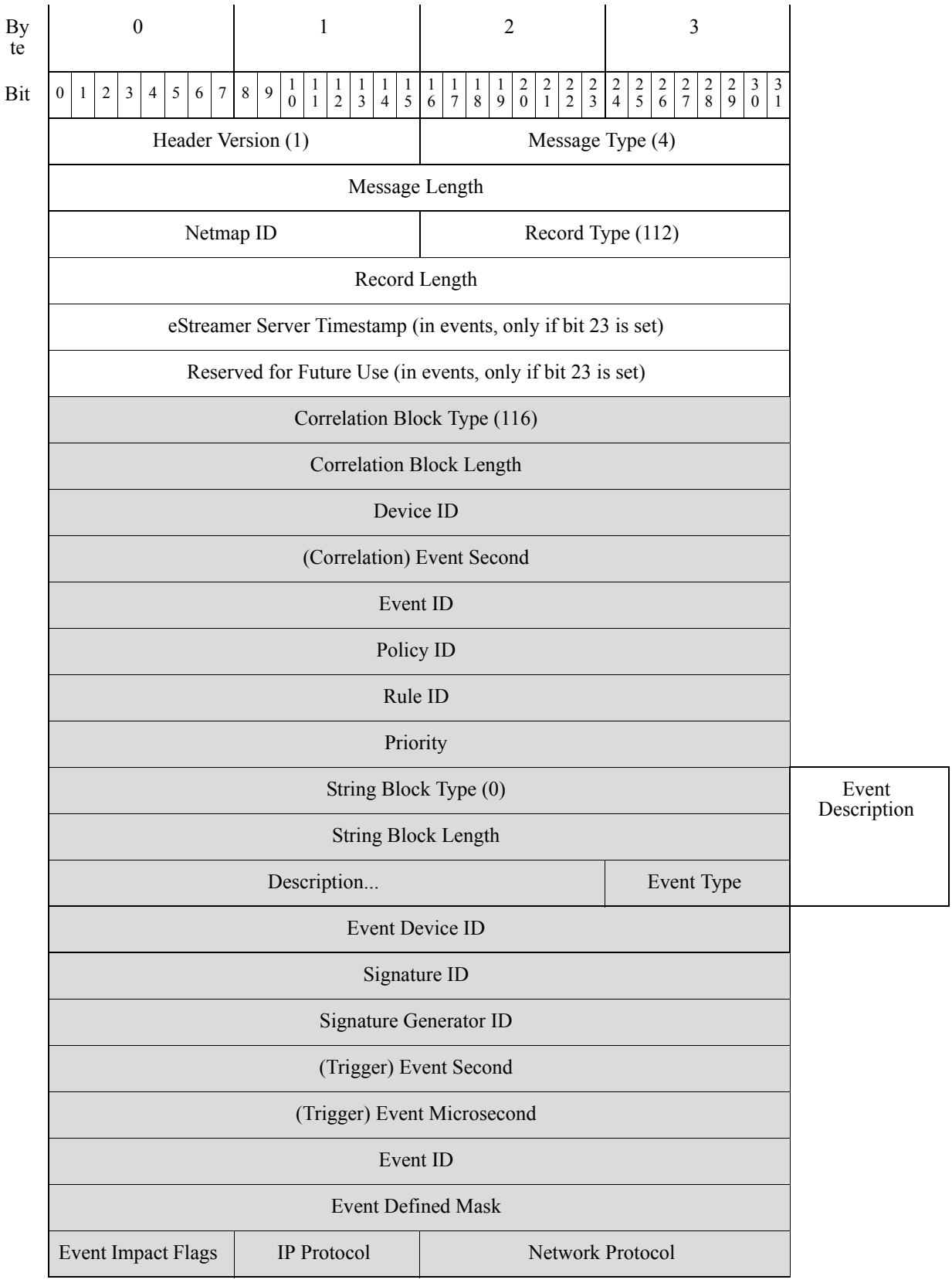
- Correlation Event for 5.0 - 5.0.2, page B-211
- Correlation Event for 5.1-5.3.x, page B-219

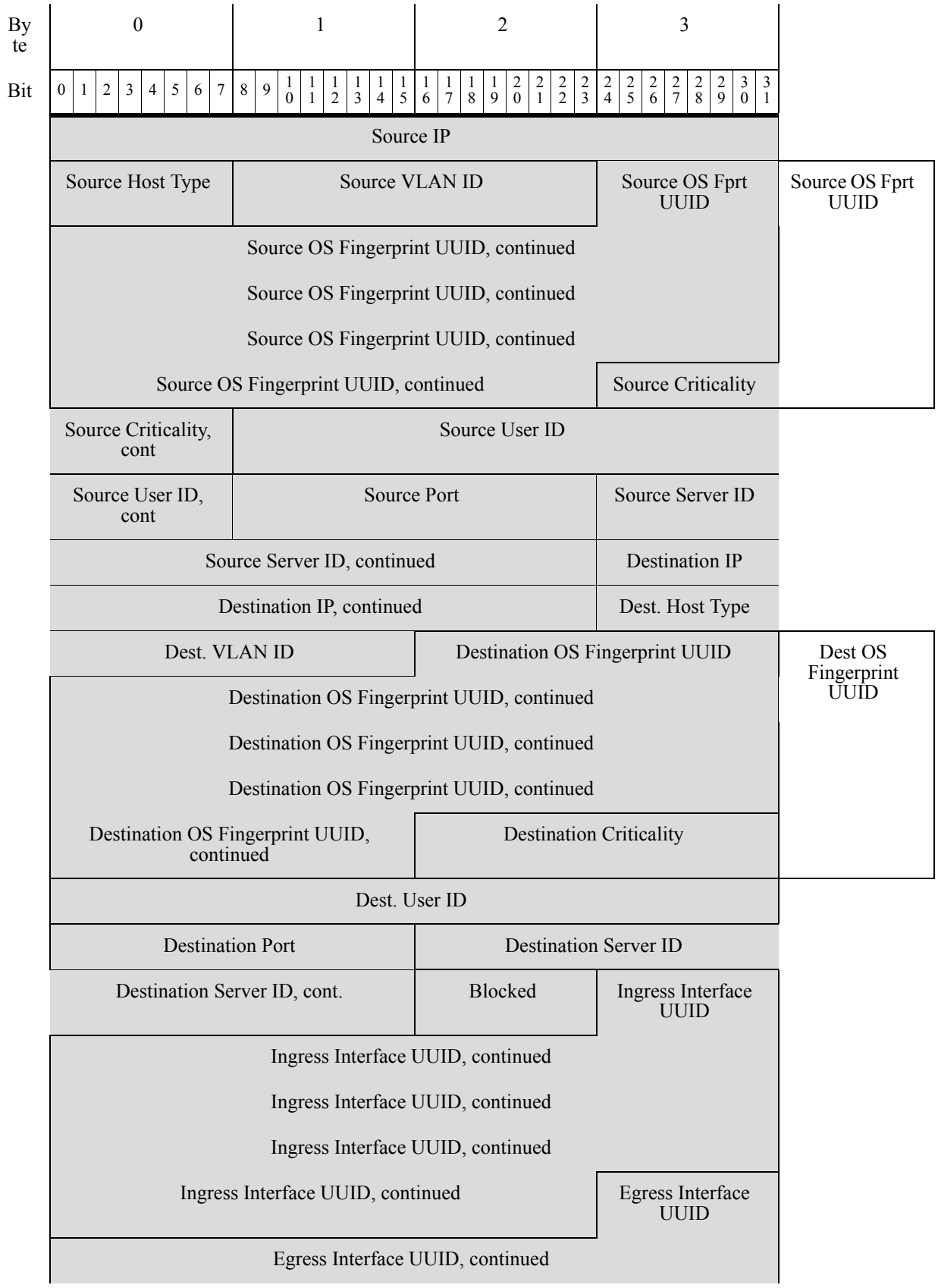
Correlation Event for 5.0 - 5.0.2

Correlation events (called compliance events in pre-5.0 versions) contain information about correlation policy violations. This message uses the standard eStreamer message header and specifies a record type of 112, followed by a correlation data block of type 116. Data block type 116 differs from its predecessor (block type 107) in including additional information about the associated security zone and interface.

You can request 5.0 correlation events from eStreamer only by extended request, for which you request event type code 31 and version code 7 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests). You can optionally enable bit 23 in the flags field of the initial event stream request message, to include the extended event header. You can also enable bit 20 in the flags field to include user metadata.

Note that the record structure includes a String block type, which is a block in series 1. For information about series 1 blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Header Version (1)
Message Type (4)
Message Length
Netmap ID
Record Type (112)
Record Length
eStreamer Server Timestamp (in events, only if bit 23 is set)
Reserved for Future Use (in events, only if bit 23 is set)
Correlation Block Type (116)
Correlation Block Length
Device ID
(Correlation) Event Second
Event ID
Policy ID
Rule ID
Priority
String Block Type (0)
String Block Length
Description...
Event Type
Event Description
Event Device ID
Signature ID
Signature Generator ID
(Trigger) Event Second
(Trigger) Event Microsecond
Event ID
Event Defined Mask
Event Impact Flags
IP Protocol
Network Protocol


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Source IP
Source Host Type
Source VLAN ID
Source OS Fprt UUID
Source OS Fprt UUID
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source Criticality
Source Criticality, cont
Source User ID
Source User ID, cont
Source Port
Source Server ID
Source Server ID, continued
Destination IP
Destination IP, continued
Dest. Host Type
Dest. VLAN ID
Destination OS Fingerprint UUID
Dest OS Fingerprint UUID
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination Criticality
Dest. User ID
Destination Port
Destination Server ID
Destination Server ID, cont.
Blocked
Ingress Interface UUID
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Egress Interface UUID
Egress Interface UUID, continued

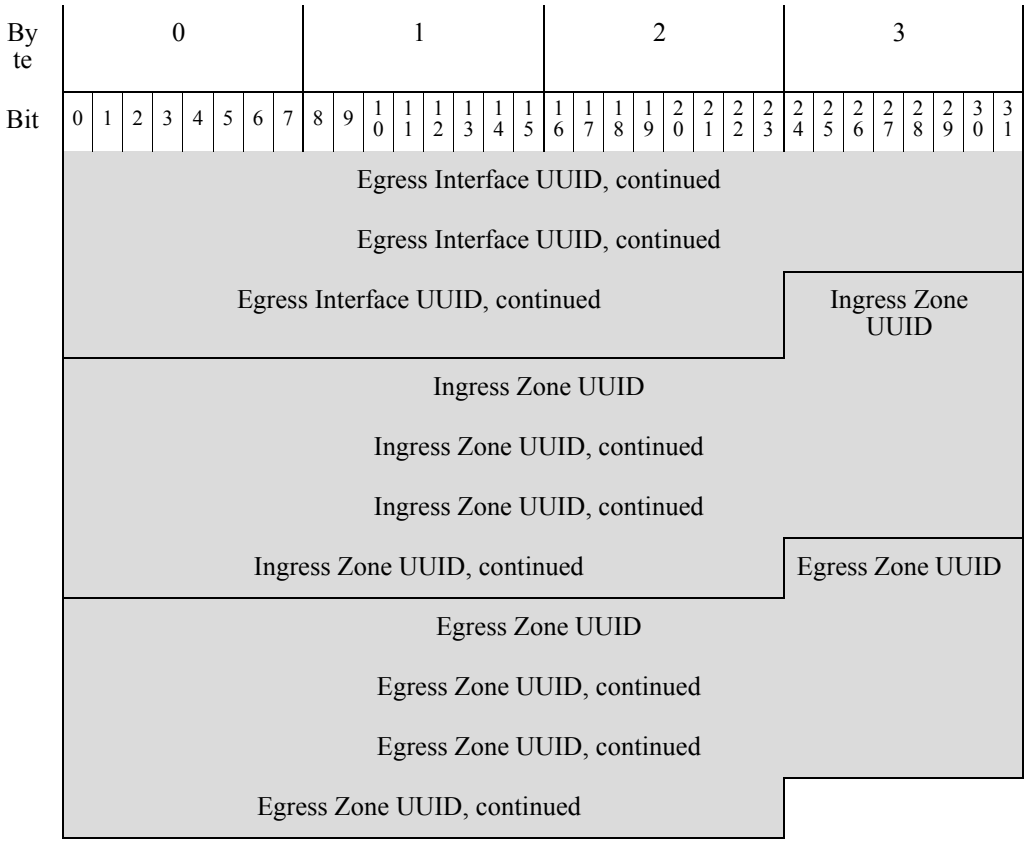


Table B-40 Correlation Event 5.0 - 5.0.2 Data Fields

Field	Data Type	Description
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of `107`. See Understanding Discovery (Series 1) Blocks, page 4-56.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.
Device ID	uint32	Internal identification number of the managed device or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain managed device names by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
(Correlation) Event Second	uint32	UNIX timestamp indicating the time that the correlation event was generated (in seconds from 01/01/1970).
Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.

Table B-40 Correlation Event 5.0 - 5.0.2 Data Fields (continued)

Field	Data Type	Description
Priority	uint32	Priority assigned to the event. This is an integer value from `0` to `5`.
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to `0`. For more information about string blocks, see String Data Block, page 4-64.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, host discovery, or user event: • `1` — Intrusion • `2` — Host discovery • `3` — User
Event Device ID	uint32	Identification number of the device that generated the event that triggered the correlation event. You can obtain device name by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is `0`.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Firepower System preprocessor or rules engine that generated the event.
(Trigger) Event Second	uint32	UNIX timestamp indicating the time of the event that triggered the correlation policy rule (in seconds from 01/01/1970).
(Trigger) Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See Table B-41 on page B-218 for a list of each bit value.

Table B-40 Correlation Event 5.0 - 5.0.2 Data Fields (continued)

Field	Data Type	Description
Event Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • `0x01` (bit 0) — Source or destination host is in a network monitored by the system. • `0x02` (bit 1) — Source or destination host exists in the network map. • `0x04` (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • `0x08` (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • `0x10` (bit 4) — There is a vulnerability mapped to the server detected in the event. • `0x20` (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • `0x40` (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red (bit 6). The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • `0x80` (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be 0 or 1: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
IP Protocol	uint8	Identifier of the IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP	uint8[4]	IP address of the source host in the event, in IP address octets.
Source Host Type	uint8	Source host's type: • `0` — Host • `1` — Router • `2` — Bridge
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.

Table B-40 Correlation Event 5.0 - 5.0.2 Data Fields (continued)

Field	Data Type	Description
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	IP address of the destination host associated with the policy violation (if applicable). This value will be 0 if there is no destination IP address.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Service ID	uint32	Identification number for the server running on the source host.

Table B-40 Correlation Event 5.0 - 5.0.2 Data Fields (continued)

Field	Data Type	Description
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (drop when deployment is inline, switched, or routed) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in inline, switched, or routed deployment.
Ingress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the ingress interface associated with correlation event.
Egress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the egress interface associated with correlation event.
Ingress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the ingress security zone associated with correlation event.
Egress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the egress security zone associated with correlation event.

The following table describes each Event Defined Mask value.

Table B-41 Event Defined Values

Description	Mask Value
Event Impact Flags	0x00000001
IP Protocol	0x00000002
Network Protocol	0x00000004
Source IP	0x00000008
Source Host Type	0x00000010
Source VLAN ID	0x00000020
Source Fingerprint ID	0x00000040
Source Criticality	0x00000080
Source Port	0x00000100
Source Server	0x00000200
Destination IP	0x00000400
Destination Host Type	0x00000800
Destination VLAN ID	0x00001000
Destination Fingerprint ID	0x00002000
Destination Criticality	0x00004000
Destination Port	0x00008000
Destination Server	0x00010000

Table B-41 Event Defined Values (continued)

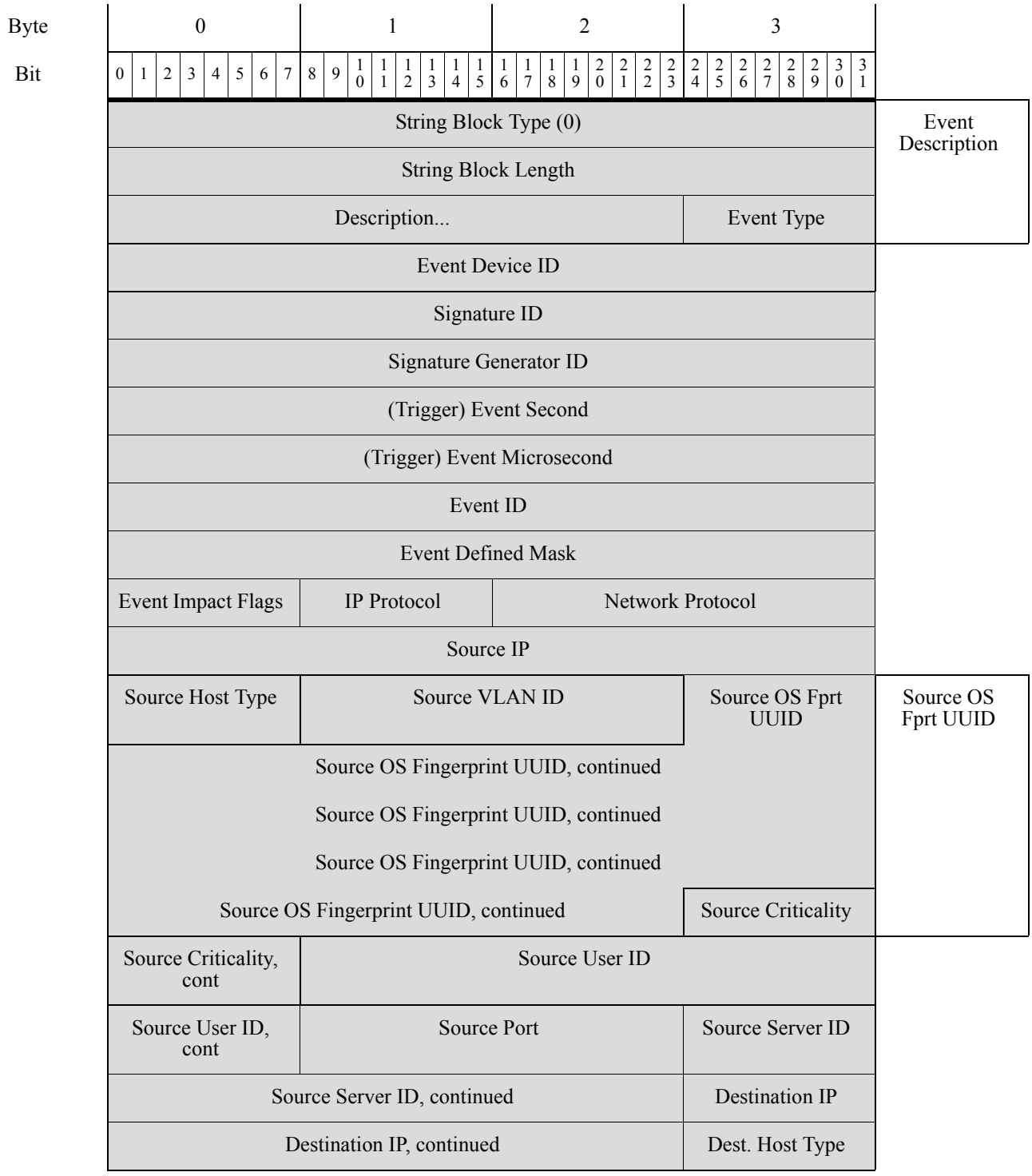
Description	Mask Value
Source User	0x00020000
Destination User	0x00040000

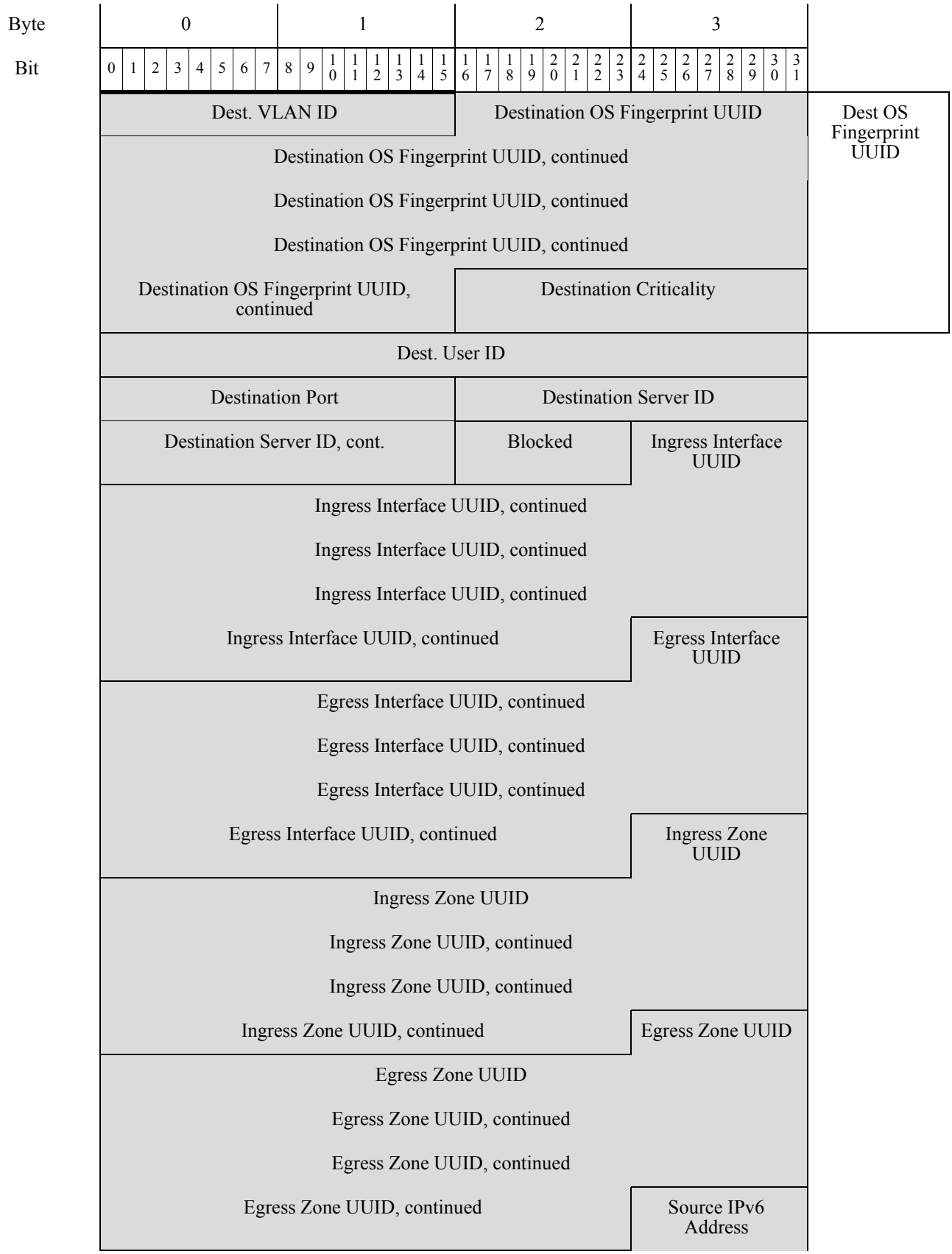
Correlation Event for 5.1-5.3.x

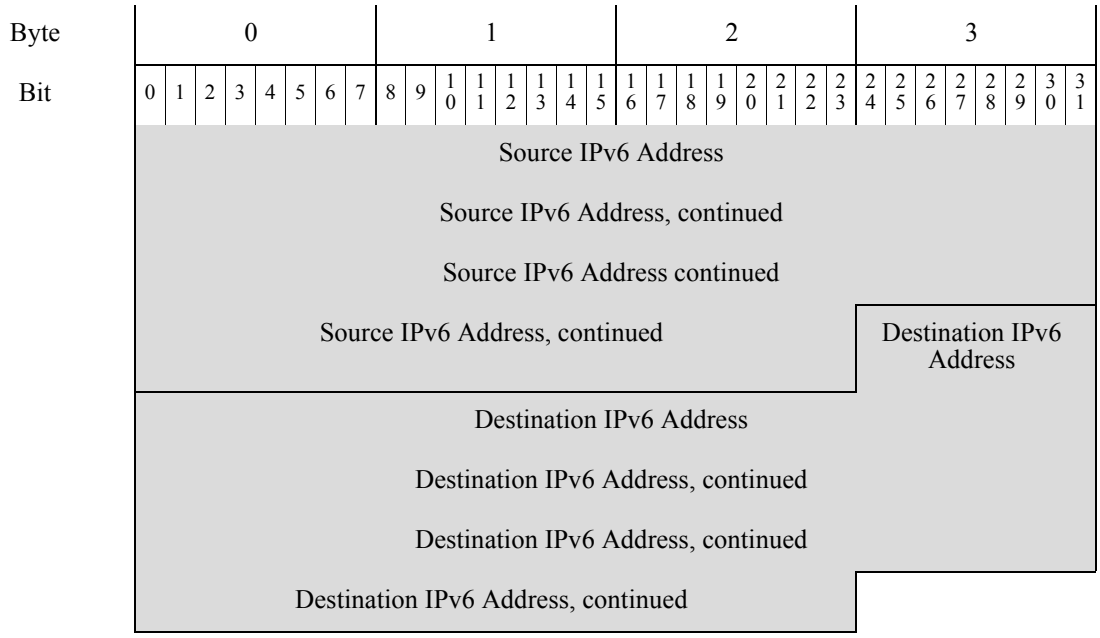
Correlation events (called compliance events in pre-5.0 versions) contain information about correlation policy violations. This message uses the standard eStreamer message header and specifies a record type of 112, followed by a correlation data block of type 128 in the series 1 set of data blocks. Data block type 128 differs from its predecessor (block type 116) in including IPv6 support.

You can request 5.1-5.3.x correlation events from eStreamer only by extended request, for which you request event type code 31 and version code 8 in the Stream Request message (see Submitting Extended Requests, page 2-4 for information about submitting extended requests). You can optionally enable bit 23 in the flags field of the initial event stream request message, to include the extended event header. You can also enable bit 20 in the flags field to include user metadata.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header Version (1)																	Message Type (4)															
Message Length																																
Netmap ID																	Record Type (112)															
Record Length																																
eStreamer Server Timestamp (in events, only if bit 23 is set)																																
Reserved for Future Use (in events, only if bit 23 is set)																																
Correlation Block Type (128)																																
Correlation Block Length																																
Device ID																																
(Correlation) Event Second																																
Event ID																																
Policy ID																																
Rule ID																																
Priority																																


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
String Block Type (0)
String Block Length
Description...
Event Type
Event Description
Event Device ID
Signature ID
Signature Generator ID
(Trigger) Event Second
(Trigger) Event Microsecond
Event ID
Event Defined Mask
Event Impact Flags
IP Protocol
Network Protocol
Source IP
Source Host Type
Source VLAN ID
Source OS Fprt UUID
Source OS Fprt UUID
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source OS Fingerprint UUID, continued
Source Criticality
Source Criticality, cont
Source User ID
Source User ID, cont
Source Port
Source Server ID
Source Server ID, continued
Destination IP
Destination IP, continued
Dest. Host Type


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Dest. VLAN ID
Destination OS Fingerprint UUID
Dest OS Fingerprint UUID
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination OS Fingerprint UUID, continued
Destination Criticality
Dest. User ID
Destination Port
Destination Server ID
Destination Server ID, cont.
Blocked
Ingress Interface UUID
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Ingress Interface UUID, continued
Egress Interface UUID
Egress Interface UUID, continued
Egress Interface UUID, continued
Egress Interface UUID, continued
Egress Interface UUID, continued
Ingress Zone UUID
Ingress Zone UUID
Ingress Zone UUID, continued
Ingress Zone UUID, continued
Ingress Zone UUID, continued
Egress Zone UUID
Egress Zone UUID
Egress Zone UUID, continued
Egress Zone UUID, continued
Egress Zone UUID, continued
Source IPv6 Address



Note that the record structure includes a String block type, which is a block in series 1. For information about series 1 blocks, see Understanding Discovery (Series 1) Blocks, page 4-56.

Table B-42 Correlation Event 5.1-5.3.x Data Fields

Field	Data Type	Description
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 128. See Understanding Discovery (Series 1) Blocks, page 4-56.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.
Device ID	uint32	Internal identification number of the managed device or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain managed device names by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
(Correlation) Event Second	uint32	UNIX timestamp indicating the time that the correlation event was generated (in seconds from 01/01/1970).
Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record, page 4-14 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.

Table B-42 Correlation Event 5.1-5.3.x Data Fields (continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to 0. For more information about string blocks, see String Data Block, page 4-64.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, host discovery, or user event: • 1 — Intrusion • 2 — Host discovery • 3 — User
Event Device ID	uint32	Identification number of the device that generated the event that triggered the correlation event. You can obtain device name by requesting Version 3 metadata. See Managed Device Record Metadata, page 3-34 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Firepower System preprocessor or rules engine that generated the event.
(Trigger) Event Second	uint32	UNIX timestamp indicating the time of the event that triggered the correlation policy rule (in seconds from 01/01/1970).
(Trigger) Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the Cisco device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See Table B-41 on page B-218 for a list of each bit value.

Table B-42 Correlation Event 5.1-5.3.x Data Fields (continued)

Field	Data Type	Description
Event Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Firepower System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `x` indicates the value can be 0 or 1: • (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
IP Protocol	uint8	Identifier of the IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP Address	uint8[4]	This field is reserved but no longer populated. The Source IPv4 address is stored in the Source IPv6 Address field. See IP Addresses, page 1-6 for more information.
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge

Table B-42 Correlation Event 5.1-5.3.x Data Fields (continued)

Field	Data Type	Description
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	This field is reserved but no longer populated. The Destination IPv4 address is stored in the Destination IPv6 Address field. See IP Addresses, page 1-6 for more information.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record, page 4-14 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Service ID	uint32	Identification number for the server running on the source host.

Table B-42 Correlation Event 5.1-5.3.x Data Fields (continued)

Field	Data Type	Description
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (drop when deployment is inline, switched, or routed) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in inline, switched, or routed deployment.
Ingress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the ingress interface associated with correlation event.
Egress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the egress interface associated with correlation event.
Ingress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the ingress security zone associated with correlation event.
Egress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the egress security zone associated with correlation event.
Source IPv6 Address	uint8[16]	IP address of the source host in the event, in IPv6 address octets.
Destination IPv6 Address	uint8[16]	IP address of the destination host in the event, in IPv6 address octets.

Legacy Host Data Structures

To request these structures, you must use a Host Request Message. To request a legacy structure, the Host Request Message must use an older format. See Host Request Message Format, page 2-25 for more information.

The following topics describe legacy host data structures, including both host profile and full host profile structures:

Full Host Profile Data Block 5.0 - 5.0.2

The Full Host Profile data block for version 5.0 - 5.0.2 contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures, page 4-1. The Full Host Profile data block a block type value of 111.

Note An asterisk(*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Full Host Profile Data Block (111)																															
	Data Block Length																															
	IP Address																															
	Hops								Generic List Block Type (31)																							
	Generic List Block Type, continued								Generic List Block Length																							
OS Derived Fingerprints	Generic List Block Length, continued								Operating System Fingerprint Block Type (130)*																							
	OS Fingerprint Block Type (130)*, con't								Operating System Fingerprint Block Length																							
	OS Fingerprint Block Length, con't								Operating System Derived Fingerprint Data...																							
	Generic List Block Type (31)																															
	Generic List Block Length																															
Server Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Server Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Client Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Client Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
User Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System User Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Scan Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Scan Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Application Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Application Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Conflict Fingerprint Data...																															
(TCP) Full Server Data	List Block Type (11)...																															
	List Block Length...																															
	(TCP) Full Server Data Blocks (104)*																															
(UDP) Full Server Data	List Block Type (11)																															
	List Block Length																															
	(UDP) Full Server Data Blocks (104)*																															
Network Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Network) Protocol Data Blocks (4)*																															
Transport Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Transport) Protocol Data Blocks (4)*																															
MAC Address Data	List Block Type (11)																															
	List Block Length																															
	Host MAC Address Data Blocks (95)*																															
	Last Seen																															
	Host Type																															
	Business Criticality																VLAN ID															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	VLAN Type								VLAN Priority								Generic List Block Type (31)															
Host Client Data	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Full Host Client Application Data Blocks (112)*															
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name String...																															
Notes Data	String Block Type (0)																															
	String Block Length																															
	Notes String....																															
(VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty Scan Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*																															
Attribute Value Data	List Block Type (11)																															
	List Block Length																															
	Attribute Value Data Blocks *																															

The following table describes the components of the Full Host Profile for 5.0 - 5.0.2 record.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields

Field	Data Type	Description
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the device.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always `31`.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always `11`.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • `0` — Host • `1` — Router • `2` — Bridge • `3` — NAT (network address translation device) • `4` — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always `0`.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.

Table B-43 Full Host Profile Record 5.0 - 5.0.2 Fields (continued)

Field	Data Type	Description
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always `0`.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Cisco-detected IDs. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always `11`.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block, page 4-74 for a description of the data blocks in this list.

Full Host Profile Data Block 5.1.1

The Full Host Profile data block for version 5.1.1 contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures, page 4-1. The Full Host Profile data block a block type value of 135. It deprecates data block 111.

Note An asterisk(*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Full Host Profile Data Block (135)																															
	Data Block Length																															
	IP Address																															
	Hops								Generic List Block Type (31)																							
	Generic List Block Type, continued								Generic List Block Length																							
OS Derived Fingerprints	Generic List Block Length, continued								Operating System Fingerprint Block Type (130)*																							
	OS Fingerprint Block Type (130)*, con't								Operating System Fingerprint Block Length																							
	OS Fingerprint Block Length, con't								Operating System Derived Fingerprint Data...																							
	Generic List Block Type (31)																															
	Generic List Block Length																															
Server Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Server Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Client Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Client Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System VDB Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
User Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System User Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Scan Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Scan Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Application Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Application Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Conflict Fingerprint Data...																															
(TCP) Full Server Data	List Block Type (11)...																															
	List Block Length...																															
	(TCP) Full Server Data Blocks (104)*																															
(UDP) Full Server Data	List Block Type (11)																															
	List Block Length																															
	(UDP) Full Server Data Blocks (104)*																															
Network Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Network) Protocol Data Blocks (4)*																															
Transport Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Transport) Protocol Data Blocks (4)*																															
MAC Address Data	List Block Type (11)																															
	List Block Length																															
	Host MAC Address Data Blocks (95)*																															
	Last Seen																															
	Host Type																															
	Business Criticality																VLAN ID															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	VLAN Type								VLAN Priority								Generic List Block Type (31)																
Host Client Data	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Full Host Client Application Data Blocks (112)*															
NetBIOS Name	String Block Type (0)																															
	String Block Length																															
	NetBIOS Name String...																															
Notes Data	String Block Type (0)																															
	String Block Length																															
	Notes String....																															
(VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty Scan Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*																															
Attribute Value Data	List Block Type (11)																															
	List Block Length																															
	Attribute Value Data Blocks *																															
	Mobile								Jailbroken								VLAN Presence															

The following table describes the components of the Full Host Profile for 5.1.1 record.

Table B-44 Full Host Profile Record 5.1.1 Fields

Field	Data Type	Description
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.

Table B-44 Full Host Profile Record 5.1.1 Fields (continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.

Table B-44 Full Host Profile Record 5.1.1 Fields (continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.

Table B-44 Full Host Profile Record 5.1.1 Fields (continued)

Field	Data Type	Description
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — Host • 1 — Router • 2 — Bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.

Table B-44 Full Host Profile Record 5.1.1 Fields (continued)

Field	Data Type	Description
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Cisco-detected IDs. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block, page 4-74 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No

Full Host Profile Data Block 5.2.x

The Full Host Profile data block for version 5.2.x contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures, page 4-1. The Full Host Profile data block a block type value of 140. It supersedes the prior version, which has a block type of 135.

Note An asterisk (*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Full Host Profile Data Block (140)																															
	Data Block Length																															
	Host ID Host ID, continued Host ID, continued Host ID, continued																															
IP Addresses	List Block Type (11)																															
	List Block Length																															
	IP Address Data Blocks (143)*																															
	Hops								Generic List Block Type (31)																							
	Generic List Block Type, continued								Generic List Block Length																							
OS Derived Fingerprints	Generic List Block Length, continued								Operating System Fingerprint Block Type (130)*																							
	OS Fingerprint Block Type (130)*, con't								Operating System Fingerprint Block Length																							
	OS Fingerprint Block Length, con't								Operating System Derived Fingerprint Data...																							
	Generic List Block Type (31)																															
	Generic List Block Length																															
Server Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Server Fingerprint Data...																															
	Generic List Block Type (31)																															


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Generic List Block Length
Client Fingerprints
Operating System Fingerprint Block Type (130)*
Operating System Fingerprint Block Length
Operating System Client Fingerprint Data...
Generic List Block Type (31)
Generic List Block Length
VDB Native Fingerprints 1
Operating System Fingerprint Block Type (130)*
Operating System Fingerprint Block Length
Operating System VDB Fingerprint Data...
Generic List Block Type (31)
Generic List Block Length
VDB Native Fingerprints 2
Operating System Fingerprint Block Type (130)*
Operating System Fingerprint Block Length
Operating System VDB Fingerprint Data...
Generic List Block Type (31)
Generic List Block Length
User Fingerprints
Operating System Fingerprint Block Type (130)*
Operating System Fingerprint Block Length
Operating System User Fingerprint Data...
Generic List Block Type (31)
Generic List Block Length
Scan Fingerprints
Operating System Fingerprint Block Type (130)*
Operating System Fingerprint Block Length
Operating System Scan Fingerprint Data...
Generic List Block Type (31)
Generic List Block Length

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Application Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Application Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Conflict Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Mobile Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Mobile Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
IPv6 Server Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System IPv6 Server Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
Ipv6 Client Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System Ipv6 Client Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ipv6 DHCP Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System IPv6 DHCP Fingerprint Data...																															
	Generic List Block Type (31)																															
	Generic List Block Length																															
User Agent Fingerprints	Operating System Fingerprint Block Type (130)*																															
	Operating System Fingerprint Block Length																															
	Operating System User Agent Fingerprint Data...																															
(TCP) Full Server Data	List Block Type (11)...																															
	List Block Length...																															
	(TCP) Full Server Data Blocks (104)*																															
(UDP) Full Server Data	List Block Type (11)																															
	List Block Length																															
	(UDP) Full Server Data Blocks (104)*																															
Network Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Network) Protocol Data Blocks (4)*																															
Transport Protocol Data	List Block Type (11)																															
	List Block Length																															
	(Transport) Protocol Data Blocks (4)*																															
MAC Address Data	List Block Type (11)																															
	List Block Length																															
	Host MAC Address Data Blocks (95)*																															
	Last Seen																															
	Host Type																															
	Business Criticality																VLAN ID															

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	VLAN Type								VLAN Priority								Generic List Block Type (31)																
Host Client Data	Generic List Block Type, continued																Generic List Block Length															
	Generic List Block Length, continued																Full Host Client Application Data Blocks (112)*															
NetBios Name	String Block Type (0)																															
Name	String Block Length																															
	NetBIOS Name String...																															
Notes Data	String Block Type (0)																															
	String Block Length																															
	Notes String....																															
(VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*																															
3rd Pty Scan Host Vulns	Generic List Block Type (31)																															
	Generic List Block Length																															
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*																															
Attribute Value Data	List Block Type (11)																															
	List Block Length																															
	Attribute Value Data Blocks *																															
	Mobile								Jailbroken																							

The following table describes the components of the Full Host Profile for 5.2.x record.

Table B-45 Full Host Profile Record 5.2.x Fields

Field	Data Type	Description
Host ID	uint8[16]	Unique ID number of the host. This is a UUID.
List Block Type	uint32	Initiates a List data block comprising IP address data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated IP address data blocks.
IP Address	variable	IP addresses of the host and when each IP address was last seen. See Host IP Address Data Block, page 4-89 for a description of this data block.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always 31.

Table B-45 Full Host Profile Record 5.2.x Fields (continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Cisco VDB fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Cisco vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table B-45 Full Host Profile Record 5.2.x Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying mobile device fingerprint data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a mobile device host. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table B-45 Full Host Profile Record 5.2.x Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (IPv6 Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 DHCP) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Agent) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+, page 4-130 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.

Table B-45 Full Host Profile Record 5.2.x Fields (continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always `11`.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • `0` — Host • `1` — Router • `2` — Bridge • `3` — NAT (network address translation device) • `4` — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.

Table B-45 Full Host Profile Record 5.2.x Fields (continued)

Field	Data Type	Description
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Cisco vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Cisco-detected IDs. See Host Vulnerability Data Block 4.9.0+, page 4-104 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.

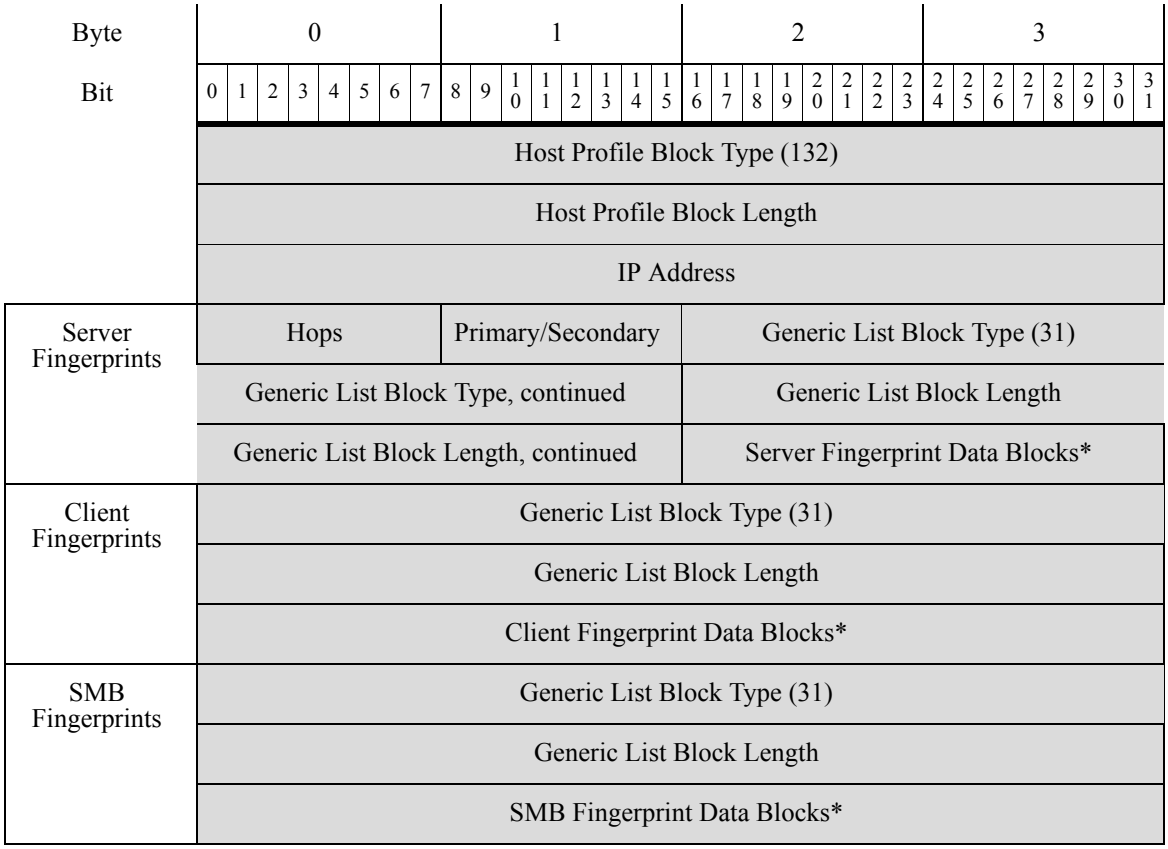
Table B-45 Full Host Profile Record 5.2.x Fields (continued)

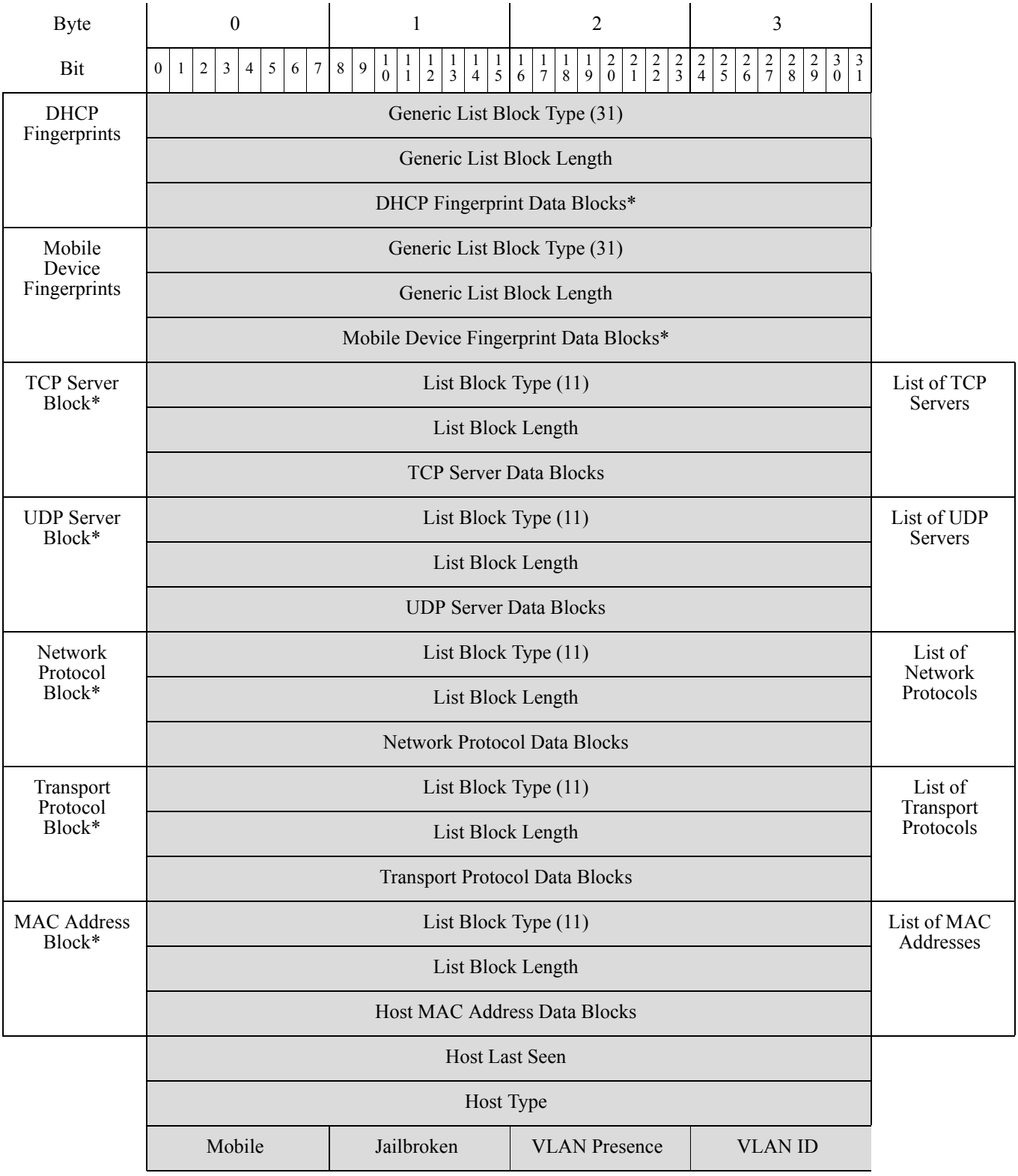
Field	Data Type	Description
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block, page 4-74 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.

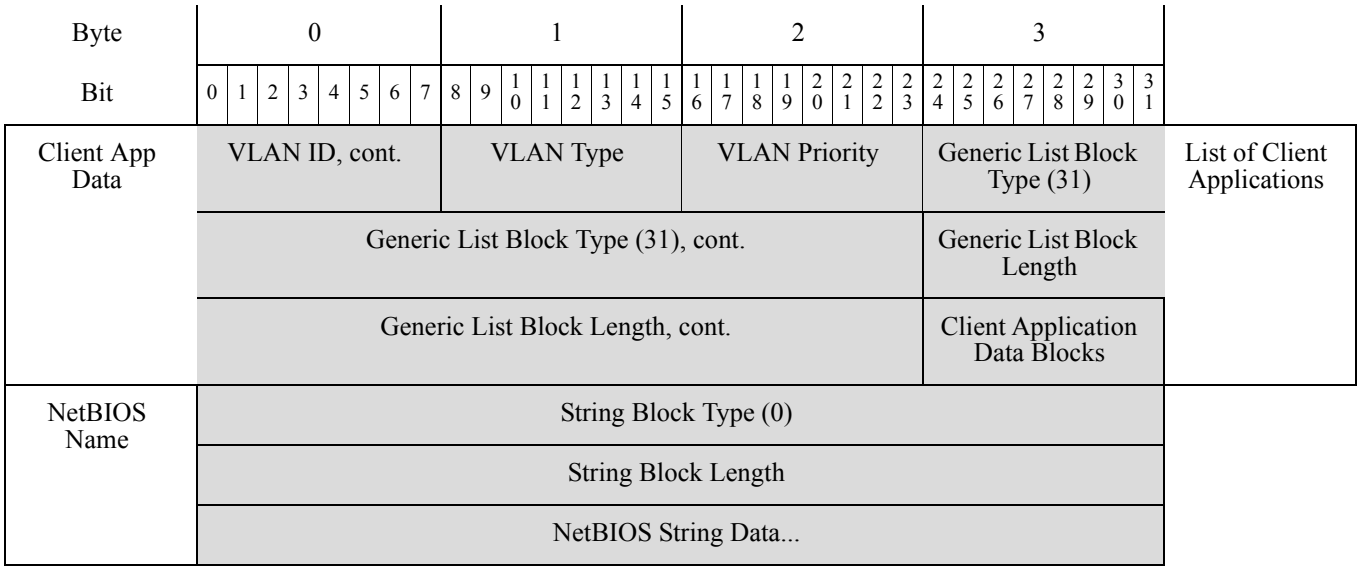
Host Profile Data Block for 5.1.x

The following diagram shows the format of a Host Profile data block. The data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a data block can convey a NetBIOS name for the host. The Host Profile data block has a block type of 132.

Note An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.




Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
DHCP Fingerprints
Generic List Block Type (31)
Generic List Block Length
DHCP Fingerprint Data Blocks*
Mobile Device Fingerprints
Generic List Block Type (31)
Generic List Block Length
Mobile Device Fingerprint Data Blocks*
TCP Server Block*
List Block Type (11)
List Block Length
TCP Server Data Blocks
List of TCP Servers
UDP Server Block*
List Block Type (11)
List Block Length
UDP Server Data Blocks
List of UDP Servers
Network Protocol Block*
List Block Type (11)
List Block Length
Network Protocol Data Blocks
List of Network Protocols
Transport Protocol Block*
List Block Type (11)
List Block Length
Transport Protocol Data Blocks
List of Transport Protocols
MAC Address Block*
List Block Type (11)
List Block Length
Host MAC Address Data Blocks
List of MAC Addresses
Host Last Seen
Host Type
Mobile
Jailbroken
VLAN Presence
VLAN ID



The following table describes the fields of the host profile data block returned by version 5.1.x

Table B-46 Host Profile Data Block 5.1.x Fields

Field	Data Type	Description
Host Profile Block Type	uint32	Initiates the Host Profile data block for 5.1.x. This value is always `132`.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8[4]	IP address of the host described in the profile, in IP address octets.
Hops	uint8	Number of hops from the host to the device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • `0` — Host is in the primary network. • `1` — Host is in the secondary network.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always `31`.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.

Table B-46 Host Profile Data Block 5.1.x Fields (continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Table B-46 Host Profile Data Block 5.1.x Fields (continued)

Field	Data Type	Description
Operating System Fingerprint (Mobile Device Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a mobile device fingerprint. See Operating System Fingerprint Data Block 5.1+, page 4-149 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
TCP Server Data Blocks	variable	Host server data blocks describing a TCP server (as documented for earlier versions of the product).
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
UDP Server Data Blocks	uint32	Host server data blocks describing a UDP server (as documented for earlier versions of the product).
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Network Protocol Data Blocks	uint32	Protocol data blocks describing a network protocol. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always `11`.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Transport Protocol Data Blocks	uint32	Protocol data blocks describing a transport protocol. See Protocol Data Block, page 4-68 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always `11`.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.

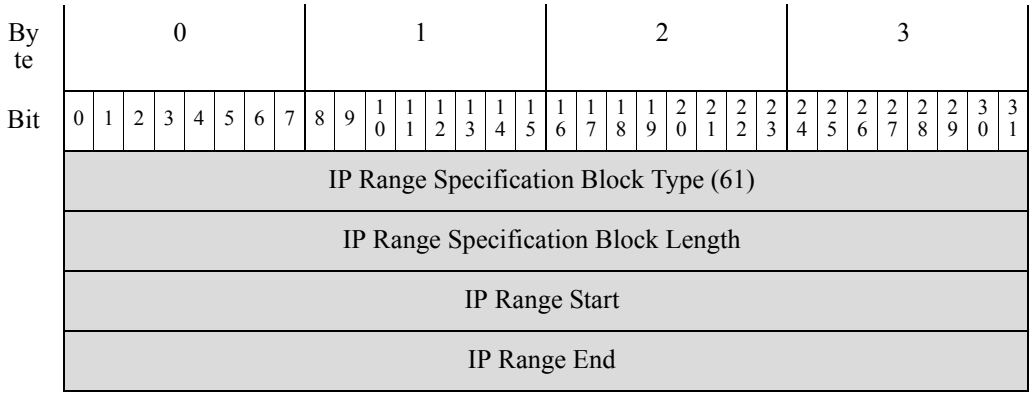
Table B-46 Host Profile Data Block 5.1.x Fields (continued)

Field	Data Type	Description
Host MAC Address Data Blocks	uint32	Host MAC Address data blocks describing a host MAC address. See Host MAC Address 4.9+, page 4-107 for a description of this data block.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — Host • 1 — Router • 2 — Bridge • 3 — NAT device • 4 — LB (load balancer)
Mobile	uint8	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint8	True-false flag indicating whether the host is a mobile device that is also jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Client Application data blocks conveying client application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated client application data blocks.
Client Application Data Blocks	uint32	Client application data blocks describing a client application. See Full Host Client Application Data Block 5.0+, page 4-143 for a description of this data block.
String Block Type	uint32	Initiates a string data block for the NetBIOS name. This value is set to 0 to indicate string data.
String Block Length	uint32	Indicates the number of bytes in the NetBIOS name data block, including eight bytes for the string block type and length, plus the number of bytes in the NetBIOS name.
NetBIOS String Data	Variable	Contains the NetBIOS name of the host described in the host profile.

IP Range Specification Data Block for 5.0 - 5.1.1.x

The IP Range Specification data block conveys a range of IP addresses. IP Range Specification data blocks are used in User Protocol, User Client Application, Address Specification, User Product, User Server, User Hosts, User Vulnerability, User Criticality, and User Attribute Value data blocks. The IP Range Specification data block has a block type of 61.

The following diagram shows the format of the IP Range Specification data block:



The following table describes the components of the IP Range Specification data block.

Table B-47 IP Range Specification Data Block Fields

Field	Data Type	Description
IP Range Specification Block Type	uint32	Initiates a IP Range Specification data block. This value is always `61`.
IP Range Specification Block Length	uint32	Total number of bytes in the IP Range Specification data block, including eight bytes for the IP Range Specification block type and length fields, plus the number of bytes of IP range specification data that follows.
IP Range Specification Start	uint32	The starting IP address for the IP address range.
IP Range Specification End	uint32	The ending IP address for the IP address range.

Access Control Policy Rule Reason Data Block

The eStreamer service uses the Access Control Rule Policy Rule Reason Data block to contain information about access control policy rule IDs. This data block has a block type of 21 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Access Control Policy Rule Reason Data Block Type (21)																															
	Access Control Policy Rule Reason Data Block Length																															
Description	Reason																String Block Type (0)															
	String Block Type (0), continued																String Block Length															
	String Block Length, continued																Description...															

The following table describes the fields in the Access Control Policy Rule ID metadata block.

Table B-48 Access Control Policy Rule Reason Data Block Fields

Field	Data Type	Description
Access Control Policy Rule Reason Data Block Type	uint32	Initiates an Access Control Policy Rule Reason data block. This value is always `21`.
Access Control Policy Rule Reason Data Block Length	uint32	Total number of bytes in the Access Control Policy Rule Reason data block, including eight bytes for the Access Control Policy Rule Reason data block type and length fields, plus the number of bytes of data that follows.
Reason	uint16	The number of the reason for the rule that triggered the event.
String Block Type	uint32	Initiates a String data block containing the description of the access control policy rule reason. This value is always `0`.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the reason for the rule.

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W