



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 4-7](#) 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 4-40](#) 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 4-44](#) describes the structure of the data record that eStreamer uses for the various host discovery event types.
- [User Data Structures by Event Type, page 4-61](#) describes the structure of the data record that eStreamer uses for the various user event types.
- [Understanding Discovery \(Series 1\) Blocks, page 4-62](#) 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 4-155](#) 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 4-40](#).
- 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 4-63](#) or [User Vulnerability Data Block 5.0+, page 4-155](#).

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 4-45 |
| 11 | 103 | 1 | New TCP Server | Current | Server Messages, page 4-46 |
| 12 | 103 | 1 | New UDP Server | Current | Server Messages, page 4-46 |
| 13 | 4 | 1 | New Network Protocol | Current | New Network Protocol Message, page 4-47 |
| 14 | 4 | 1 | New Transport Protocol | Current | New Transport Protocol Message, page 4-47 |

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 4-47 |
| 16 | 103 | 1 | TCP Server Information Update | Current | Server Messages, page 4-46 |
| 17 | 103 | 1 | UDP Server Information Update | Current | Server Messages, page 4-46 |
| 18 | 53 | 1 | OS Information Update | Current | Operating System Update Messages, page 4-49 |
| 19 | N/A | N/A | Host Timeout | Current | IP Address Reused and Host Timeout/Deleted Messages, page 4-49 |
| 20 | N/A | N/A | Host IP Address Reused | Current | IP Address Reused and Host Timeout/Deleted Messages, page 4-49 |
| 21 | N/A | N/A | Host Deleted: Host Limit Reached | Current | IP Address Reused and Host Timeout/Deleted Messages, page 4-49 |
| 22 | N/A | N/A | Hops Change | Current | Hops Change Message, page 4-50 |
| 23 | N/A | N/A | TCP Port Closed | Current | TCP and UDP Port Closed/Timeout Messages, page 4-50 |
| 24 | N/A | N/A | UDP Port Closed | Current | TCP and UDP Port Closed/Timeout Messages, page 4-50 |
| 25 | N/A | N/A | TCP Port Timeout | Current | TCP and UDP Port Closed/Timeout Messages, page 4-50 |
| 26 | N/A | N/A | UDP Port Timeout | Current | TCP and UDP Port Closed/Timeout Messages, page 4-50 |
| 27 | N/A | N/A | MAC Information Change | Current | MAC Address Messages, page 4-51 |
| 28 | N/A | N/A | Additional MAC Detected for Host | Current | MAC Address Messages, page 4-51 |
| 29 | N/A | N/A | Host IP Address Changed | Current | IP Address Change Message, page 4-48 |
| 31 | N/A | N/A | Host Identified as Router/Bridge | Current | Host Identified as a Bridge/Router Message, page 4-51 |
| 34 | 14 | 1 | VLAN Tag Information Update | Current | VLAN Tag Information Update Messages, page 4-52 |
| 35 | 122 | 1 | Client Application Timeout | Current | Client Application Messages, page 4-47 |
| 42 | 35 | 1 | NetBIOS Name Change | Current | Change NetBIOS Name Message, page 4-52 |
| 44 | N/A | N/A | Host Dropped: Host Limit Reached | Current | IP Address Reused and Host Timeout/Deleted Messages, page 4-49 |
| 45 | 37 | 1 | Update Banner | Current | Update Banner Message, page 4-53 |
| 46 | 55 | 1 | Add Host Attribute | Current | Attribute Messages, page 4-56 |
| 47 | 55 | 1 | Update Host Attribute | Current | Attribute Messages, page 4-56 |
| 48 | 55 | 1 | Delete Host Attribute | Current | Attribute Messages, page 4-56 |

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 4-46 |
| 52 | 103 | 1 | UDP Server Confidence Update | Legacy | Server Messages, page 4-46 |
| 53 | 53 | 1 | OS Confidence Update | Legacy | Operating System Update Messages, page 4-49 |
| 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 4-9 |
| 57 | N/A | N/A | Vulnerability Metadata | Current | Vulnerability Record, page 4-9 |
| 58 | N/A | N/A | Criticality Metadata | Current | Criticality Record, page 4-12 |
| 59 | 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 4-14 |
| 63 | N/A | N/A | Server Metadata | Current | Service Record, page 4-15 |
| 71 | 144 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 5.2.x, page B-135 |
| 71 | 152 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 5.3, page B-150 |
| 71 | 154 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 5.3.1, page B-156 |
| 71 | 155 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 5.4, page B-163 |
| 71 | 157 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 5.4.1, page B-176 |
| 71 | 160 | 1 | Connection Statistics | Legacy | Connection Statistics Data Block 6.0.x, page B-189 |
| 71 | 163 | 1 | Connection Statistics | Current | Connection Statistics Data Block 6.2+, page 4-118 |
| 73 | 136 | 1 | Connection Chunks | Current | Connection Chunk Message, page 4-54 |
| 74 | N/A | N/A | User Set OS | Current | User Server and Operating System Messages, page 4-57 |
| 75 | N/A | N/A | User Set Server | Current | User Server and Operating System Messages, page 4-57 |
| 76 | 83 | 1 | User Delete Protocol | Current | User Protocol Messages, page 4-58 |
| 77 | 60 | 1 | User Delete Client Application | Current | User Client Application Messages, page 4-58 |
| 78 | 78 | 1 | User Delete Address | Current | User Add and Delete Host Messages, page 4-55 |
| 79 | 77 | 1 | User Delete Server | Current | User Delete Server Message, page 4-55 |
| 80 | 80 | 1 | User Set Valid Vulnerabilities | Current | User Set Vulnerabilities Messages for Version 4.6.1+, page 4-54 |

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

| Record Type | Contains Block Type | Series | Description | Record Status | Data Format Described in... |
|-------------|---------------------|--------|--|---------------|---|
| 81 | 80 | 1 | User Set Invalid Vulnerabilities | Current | User Set Vulnerabilities Messages for Version 4.6.1+, page 4-54 |
| 82 | 81 | 1 | User Set Host Criticality | Current | User Set Host Criticality Messages, page 4-56 |
| 83 | 55 | 1 | User Set Attribute Value | Current | Attribute Value Messages, page 4-57 |
| 84 | 82 | 1 | User Delete Attribute Value | Current | Attribute Value Messages, page 4-57 |
| 85 | 78 | 1 | User Add Host | Current | User Add and Delete Host Messages, page 4-55 |
| 86 | N/A | N/A | User Add Server | Current | User Server and Operating System Messages, page 4-57 |
| 87 | 60 | 1 | User Add Client Application | Current | User Client Application Messages, page 4-58 |
| 88 | 83 | 1 | User Add Protocol | Current | User Protocol Messages, page 4-58 |
| 89 | 142 | 1 | User Add Scan Result | Current | Add Scan Result Messages, page 4-59 |
| 90 | N/A | N/A | Source Type Record | Current | Source Type Record, page 4-16 |
| 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 4-61 |
| 93 | 120 | 1 | User Removed Change Event | Current | User Modification Messages, page 4-61 |
| 94 | 120 | 1 | New User Identification Event | Current | User Modification Messages, page 4-61 |
| 95 | 121 | 1 | User Login Change Event | Current | User Information Update Message Block, page 4-62 |
| 96 | N/A | N/A | Source Detector Record | Current | Source Detector Record, page 4-17 |
| 98 | 57 | 2 | User Record | Current | User Record, page 4-19 |
| 101 | N/A | N/A | New OS Event | Current | New Operating System Messages, page 4-59 |
| 102 | 94 | 1 | Identity Conflict Event | Current | Identity Conflict and Identity Timeout System Messages, page 4-60 |
| 103 | 94 | 1 | Identity Timeout Event | Current | Identity Conflict and Identity Timeout System Messages, page 4-60 |
| 106 | N/A | N/A | Third Party Scanner Vulnerability Record | Current | Third Party Scanner Vulnerability Record, page 4-18 |
| 107 | 122 | 1 | Client Application Update | Current | Client Application Messages, page 4-47 |
| 109 | N/A | N/A | Web Application Record | Current | Web Application Record, page 4-21 |
| 114 | 121 | 1 | Failed User Login Event | Current | User Information Update Message Block, page 4-62 |

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

| Record Type | Contains Block Type | Series | Description | Record Status | Data Format Described in... |
|-------------|---------------------|--------|---|---------------|--|
| 115 | N/A | N/A | Security Zone Name Record | Current | Security Zone Name Record, page 3-28 |
| 116 | 14 | 2 | Interface Name Record | Current | Interface Name Record, page 3-30 |
| 117 | 14 | 2 | Access Control Policy Name Metadata | Current | Access Control Policy Name Record, page 3-31 |
| 118 | 14 | 2 | Intrusion Policy Name Record | Current | Intrusion Policy Name Record, page 4-22 |
| 119 | 14 | 2 | Access Control Rule ID Record | Current | Access Control Rule ID Record Metadata, page 3-32 |
| 120 | N/A | N/A | Access Control Rule Action Record | Current | Access Control Rule Action Record Metadata, page 4-23 |
| 121 | N/A | N/A | URL Category Record | Current | URL Category Record Metadata, page 4-24 |
| 122 | N/A | N/A | URL Reputation Metadata | Current | URL Reputation Record Metadata, page 4-25 |
| 124 | 21 | 2 | Access Control Rule Reason Metadata | Current | Access Control Rule Reason Metadata, page 4-26 |
| 145 | 64 | 2 | Access Control Policy Metadata | Current | Access Control Policy Metadata, page 4-27 |
| 146 | 64 | 2 | Prefilter Policy Metadata | Current | Prefilter Policy Metadata, page 4-29 |
| 147 | 21 | 2 | Tunnel or Prefilter Rule Metadata | Current | Tunnel or Prefilter Rule Metadata, page 4-30 |
| 160 | 7 | 1 | Host IOC Set Messages | Current | Host IOC Set Messages, page 4-60 |
| 161 | 39 | 2 | IOC Name Data Block for 5.3+ | Current | IOC Name Data Block for 5.3+, page 4-36 |
| 162 | 148 | 1 | User Host IOC Delete | Current | User IOC Change Data Block 5.3+, page 4-80 |
| 163 | 148 | 1 | User Host IOC Enable | Current | User IOC Change Data Block 5.3+, page 4-80 |
| 164 | 148 | 1 | User Host IOC Disable | Current | User IOC Change Data Block 5.3+, page 4-80 |
| 170 | 95 | 1 | VPN User Login Event | Current | User Information Update Message Block, page 4-62 |
| 171 | 95 | 1 | VPN User Logoff Event | Current | User Information Update Message Block, page 4-62 |
| 280 | 22 | 2 | Security Intelligence Category Metadata | Current | Security Intelligence Category Metadata, page 4-31 |
| 281 | N/A | N/A | Security Intelligence Source/Destination Record | Current | Security Intelligence Source/Destination Record, page 4-33 |

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 2-40](#). For important information on how eStreamer streams metadata records, see [Metadata Transmission, page 2-40](#).

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 4-9](#)
- [Vulnerability Record, page 4-9](#)
- [Criticality Record, page 4-12](#)
- [Network Protocol Record, page 4-12](#)
- [Attribute Record, page 4-13](#)
- [Scan Type Record, page 4-14](#)
- [Service Record, page 4-15](#)
- [Source Type Record, page 4-16](#)
- [Source Application Record, page 4-16](#)
- [Source Detector Record, page 4-17](#)
- [Third Party Scanner Vulnerability Record, page 4-18](#)
- [User Record, page 4-19](#)
- [Web Application Record, page 4-21](#)
- [Intrusion Policy Name Record, page 4-22](#)
- [Access Control Rule Action Record Metadata, page 4-23](#)
- [URL Category Record Metadata, page 4-24](#)
- [URL Reputation Record Metadata, page 4-25](#)
- [Access Control Rule Reason Metadata, page 4-26](#)
- [Security Intelligence Category Metadata, page 4-31](#)
- [Security Intelligence Source/Destination Record, page 4-33](#)

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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | OS Version Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | OS Version... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (55) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Application ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------------------------------|---|---|---|---|---|---|---|--------|---|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (57) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Vulnerability ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Impact | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Exploits | | | | | | | | Remote | | | | | | | | Entry Date Length | | | | | | | | | | | | | | | |
| | Entry Date Length Cont. | | | | | | | | | | | | | | | | Entry Date... | | | | | | | | | | | | | | | |
| | Published Date Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Published Date... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Modified Date Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Modified Date... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Title Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Title... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Short Description Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Short Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Description Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Technical Description Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Technical Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Solution Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Solution... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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: <ul style="list-style-type: none"> 0 — Yes 1 — No |
| Remote | uint8 | Indicates whether the vulnerability can be exploited across a network. Possible values include: <ul style="list-style-type: none"> 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. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (58) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Criticality ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (59) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Network Protocol ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (60) | | | | | | | | | | | | | | | |
| | Record 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 |
| Attribute ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (61) | | | | | | | | | | | | | | | | |
| Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scan Type ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| Name Length | uint32 | The number of bytes included in the scan type name. |
| Name | string | The name of the scan type. |

Service Record

The eStreamer service transmits metadata containing service information for an event within a Service record, the format of which is shown below. The application ID of the service's application protocol provides the cross-reference to the metadata. (Service 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 Service 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 (63) | | | | | | | | | | | | | | | | |
| Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Application ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields in the Service record.

Table 4-9 Service Record Fields

| Field | Data Type | Description |
|----------------|-----------|--|
| Application ID | uint32 | The application ID number of the application protocol. This field is the unique key for this record. |
| Name Length | uint32 | The number of bytes included in the service 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (90) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (91) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Application ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (96) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Detector 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 |
| Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (106) | | | | | | | | | | | | | | | | |
| Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vulnerability ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scanner Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Title Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Title... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 |
| CVE ID Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CVE ID... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BugTraQ Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BugTraQ ID... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field, along with Scanner Type, makes up the unique key for this record. |
| Scanner Type | uint32 | The third-party scanner type. This field, along with Vulnerability ID, makes up the unique key for this record. |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 |
| | 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. This field is the unique key for this record. |
| Protocol | uint32 | Protocol used to detect or report the user. Possible values are: <ul style="list-style-type: none"> 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. |

Table 4-14 User Record Fields (continued)

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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.

| 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 (109) | | | | | | | | | | | | | | | | |
| Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Application ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| 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 (118) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy Name Data Block (14) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy Name Data Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Intrusion Policy Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |

Table 4-16 *Intrusion Policy Name Data Block Fields (continued)*

| Field | Data Type | Description |
|-----------------------|-----------|--|
| 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.

| 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 (120) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Rule Action ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |

Table 4-17 Access Control Rule Action Record Fields (continued)

| Field | Data Type | Description |
|-------------|-----------|--|
| Name Length | uint32 | The number of bytes included in the name. |
| Name | string | The firewall rule action name. Possible values include: <ul style="list-style-type: none"> • 1 — 'Pending' • 2 — 'Allow' • 3 — 'Trust' • 4 — 'Block' • 5 — 'Block with Reset' • 6 — 'Monitor' • 7 — 'Interactive Block' • 8 — 'Interactive Block with Reset' • 14 — 'FastPath' • 22 — 'Domain Not Found' • 23 — 'Sinkhole' |

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.

| 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 (121) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | URL Category ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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.

| 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 (122) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | URL Reputation ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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. This field is the unique key for this record. |
| 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 6.0+, page 4-196](#)). The Access Control Rule Reason data block is block type 59 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 (124) | | | | | | | | | | | | | | | | |
| Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access Control Rule Reason Block Type (59) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access Control Rule Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Access Control Rule Reason | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| String Block Type (0). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| String Block Length. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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 59. 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. |

Table 4-20 Access Control Rule Reason Metadata Fields (continued)

| Field | Data Type | Description |
|----------------------------|-----------|---|
| Access Control Rule Reason | uint32 | <p>The reason the Access Control rule logged the connection. This field is the unique key for this record.</p> <p>The number of the reason for the rule that triggered the event.</p> <p>Rule reasons are a binary bitmap in which multiple bits may be set. There may be several reasons for a rule. The bit values are as follows:</p> <ul style="list-style-type: none"> • 1 — IP Block • 2 — IP Monitor • 4 — User Bypass • 8 — File Monitor • 16 — File Block • 32 — Intrusion Monitor • 64 — Intrusion Block • 128 — File Resume Block • 256 — File Resume Allow"] • 512 — File Custom Detection • 1024 — SSL Block • 2048 — DNS Block • 4096 — DNS Monitor • 8192 — URL Block • 16384 — URL Monitor • 32768 — Content Restriction • 65536 — Intelligent App Bypass • 131072 — WSA Threat |
| 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. |

Access Control Policy Metadata

The eStreamer service transmits metadata containing information about the access control policy that triggered an intrusion event or connection event within an Access Control Policy Metadata record, the format of which is shown below. Access control rule policy 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 145, indicating

an Access Control Policy Metadata record. It contains an Access Control Policy Metadata Block (as documented in [Access Control Policy Metadata Block 6.0+, page 4-201](#)). The Access Control Policy Metadata block is block type 64 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 (145) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy Metadata Block Type (64) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy Metadata Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AC Poli cy UI D | Access Control Policy UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sensor ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Poli cy Na me | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Policy Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-21 Access Control Policy Metadata Fields

| Field | Data Type | Description |
|---|-----------|--|
| Access Control Policy Metadata Block Type | uint32 | Initiates an Access Control Policy Metadata block. This value is always 64. This is a series 2 data block. |
| Access Control Policy Metadata Block Length | uint32 | Total number of bytes in the Access Control Policy Metadata block, including eight bytes for the Access Control Policy Metadata 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. This field is the unique key for this record. |
| Sensor ID | uint32 | ID Number of the Sensor associated with the Access Control policy. This field is the unique key for this record. |

Table 4-21 Access Control Policy Metadata Fields (continued)

| Field | Data Type | Description |
|---------------------|-----------|--|
| String Block Type | uint32 | Initiates a String data block containing the descriptive name associated with 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. |

Prefilter Policy Metadata

The eStreamer service transmits metadata containing information about the prefilter policy that triggered an intrusion event or connection event within a Prefilter Policy record, the format of which is shown below. Prefilter Policy 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 146, indicating an Prefilter Policy Metadata record. It contains an Access Control Policy Metadata Block (as documented in [Access Control Policy Metadata Block 6.0+, page 4-201](#)). The Access Control Policy Metadata block is block type 64 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 (146) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy Metadata Block Type (64) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy Metadata Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AC Poli cy UI D | Access Control Policy UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sensor 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 |
| Policy Name | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Policy Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields in the Prefilter Policy Metadata block.

Table 4-22 Prefilter Policy Metadata Fields

| Field | Data Type | Description |
|-------------------------------|-----------|--|
| Prefilter Policy Block Type | uint32 | Initiates an Prefilter Policy block. This value is always 64. This is a series 2 data block. |
| Prefilter Policy Block Length | uint32 | Total number of bytes in the Prefilter Policy block, including eight bytes for the Prefilter Policy 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. This field, along with the Sensor ID, makes up the unique key for this record. |
| Sensor ID | uint32 | ID Number of the Sensor associated with the Access Control policy This field, along with the Access Control Policy UUID, makes up the unique key for this record. |
| String Block Type | uint32 | Initiates a String data block containing the descriptive name associated with the prefilter 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 prefilter policy. |

Tunnel or Prefilter Rule Metadata

The eStreamer service transmits metadata containing information about the reason a tunnel or prefilter rule triggered an intrusion event or connection event within a Tunnel or Prefilter Rule Reason record, the format of which is shown below. Tunnel or Prefilter 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 147, indicating a Tunnel or Prefilter Rule Reason record.

As they are identical in content, it contains an Access Control Rule Reason Block (as documented in [Access Control Rule Data Block, page 4-195](#)). The Access Control Rule Reason data block is block type 59 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 (147) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Tunnel or Prefilter Rule Metadata Block Type (15) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Tunnel or Prefilter Rule Metadata Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Tunnel or Prefilter Rule ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields in the Tunnel or Prefilter Rule metadata block.

Table 4-23 Tunnel or Prefilter Rule Reason Metadata Fields

| Field | Data Type | Description |
|---------------------------------------|-----------|---|
| Tunnel or Prefilter Rule Block Type | uint32 | Initiates an Access Control Rule block. This value is always 15. Notice that this block is used for Tunnel and Prefilter rules in addition to Access Control rules. |
| Tunnel or Prefilter Rule Block Length | uint32 | Total number of bytes in the Tunnel or Prefilter Rule block, including eight bytes for the Tunnel or Prefilter block type and length fields, plus the number of bytes of data that follows. |
| Tunnel or Prefilter Rule ID | uint32 | The internal Cisco identifier for the tunnel or prefilter rule. |
| String Block Type | uint32 | Initiates a String data block containing the descriptive name associated with the tunnel or prefilter rule UUID and tunnel or prefilter 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. |

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. Security Intelligence Category 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 4-198](#)). 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 4-24 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 block list or allow list triggered by the connection. This field, along with Access Control Policy UUID, makes up the unique key for this record. |

Table 4-24 Security Intelligence Category Metadata Fields (continued)

| Field | Data Type | Description |
|---------------------------------|-----------|---|
| Access Control Policy UUID | uint8[16] | The UUID of the access control policy configured for Security Intelligence. This field, along with Security intelligence List ID, makes up the unique key for this record. |
| String Block Type | uint32 | Initiates a String data block containing the descriptive name associated with the Security Intelligence List. 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 block list or allow list 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.

| 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 (281) | | | | | | | | | | | | | | | |
| | Record Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence Source/Destination ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence Source/Destination Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence Source/Destination... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-25 Security Intelligence Source/Destination Record Fields

| Field | Data Type | Description |
|--|-----------|---|
| Security Intelligence Source/ Destination ID | uint32 | The Security Intelligence source/destination ID number. This field is the unique key for this record. |
| 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:

| 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 |
| | IOC State Block Type (150) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IOC State Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IOC ID Number | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Disabled | | | | | | | | First Seen | | | | | | | | | | | | | | | | | | | | | | | |
| | First Seen, continued | | | | | | | | First Event ID | | | | | | | | | | | | | | | | | | | | | | | |
| | First Event ID, cont. | | | | | | | | First Device ID | | | | | | | | | | | | | | | | | | | | | | | |
| | First Device ID, cont. | | | | | | | | First Instance ID | | | | | | | | | | | | | | | | First Connection Time | | | | | | | |
| | First Connection Time, cont. | | | | | | | | | | | | | | | | First Counter | | | | | | | | | | | | | | | |
| | First Counter, cont. | | | | | | | | Last Seen | | | | | | | | | | | | | | | | | | | | | | | |
| | Last Seen, cont. | | | | | | | | Last Event ID | | | | | | | | | | | | | | | | | | | | | | | |
| | Last Event ID, cont. | | | | | | | | Last Device ID | | | | | | | | | | | | | | | | | | | | | | | |
| | Last Device ID, cont. | | | | | | | | Last Instance ID | | | | | | | | | | | | | | | | Last Connection Time | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-----------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|
| 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 Connection Time, cont. | | | | | | | | | | | | | | | | | | | | | | | | Last Counter | | | | | | | |
| | Last Counter, cont. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

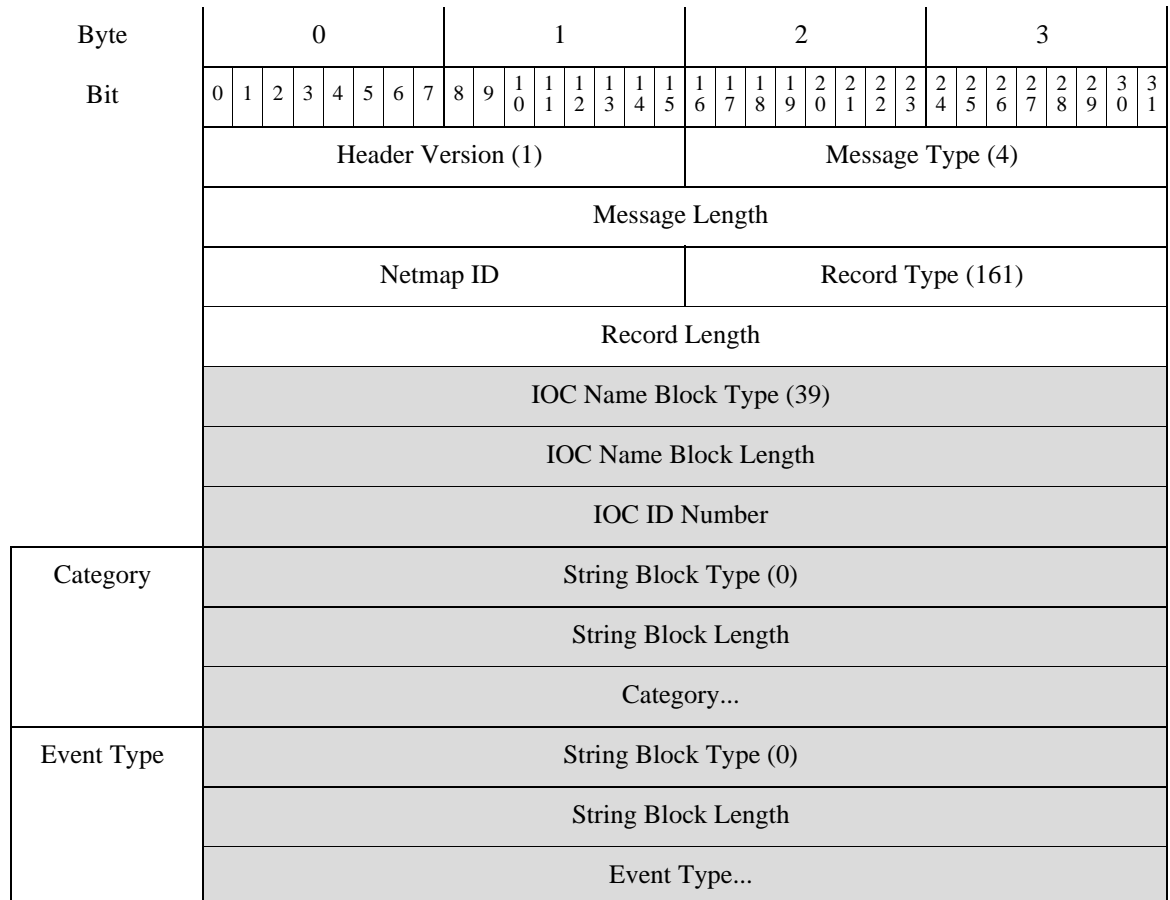
Table 4-26 IOC State Data Block Fields

| 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: <ul style="list-style-type: none"> 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 4-27 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 4-27 *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: <ul style="list-style-type: none"> • 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 4-27 *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 4-27 IOC Name Data Block Fields (continued)

| Field | Data Type | Description |
|------------|-----------|---|
| Event Type | string | <p>The event type for the compromise. Possible values include:</p> <ul style="list-style-type: none"> • 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 4-44](#). 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) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the discovery event header.

Table 4-28 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 1-4 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 4-44 for a list of available event types. |
| Event Subtype | uint32 | Event subtype. See Host Discovery Structures by Event Type, page 4-44 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 | uint8[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 4-29 *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 4-29 *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 4-29 *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 |
| Failed User Login | 1004 | 5 |
| VPN User Login | 1004 | 8 |
| VPN User Logoff | 1004 | 9 |
| Host IOC Set Type | 1008 | 1 |
| 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 4-44.

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:

- [New Host and Host Last Seen Messages](#), page 4-45
- [Server Messages](#), page 4-46
- [New Network Protocol Message](#), page 4-47
- [New Transport Protocol Message](#), page 4-47
- [Client Application Messages](#), page 4-47
- [IP Address Change Message](#), page 4-48
- [Operating System Update Messages](#), page 4-49
- [IP Address Reused and Host Timeout/Deleted Messages](#), page 4-49
- [Hops Change Message](#), page 4-50
- [Hops Change Message](#), page 4-50
- [TCP and UDP Port Closed/Timeout Messages](#), page 4-50
- [MAC Address Messages](#), page 4-51

- [Host Identified as a Bridge/Router Message](#), page 4-51
- [VLAN Tag Information Update Messages](#), page 4-52
- [Change NetBIOS Name Message](#), page 4-52
- [Update Banner Message](#), page 4-53
- [Policy Control Message](#), page 4-53
- [Connection Statistics Data Message](#), page 4-53
- [Connection Chunk Message](#), page 4-54
- [User Set Vulnerabilities Messages for Version 4.6.1+](#), page 4-54
- [User Add and Delete Host Messages](#), page 4-55
- [User Delete Server Message](#), page 4-55
- [User Set Host Criticality Messages](#), page 4-56
- [Attribute Messages](#), page 4-56
- [Attribute Value Messages](#), page 4-57
- [User Server and Operating System Messages](#), page 4-57
- [User Protocol Messages](#), page 4-58
- [User Client Application Messages](#), page 4-58
- [Add Scan Result Messages](#), page 4-59
- [New Operating System Messages](#), page 4-59
- [Identity Conflict and Identity Timeout System Messages](#), page 4-60
- [Host IOC Set Messages](#), page 4-60

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

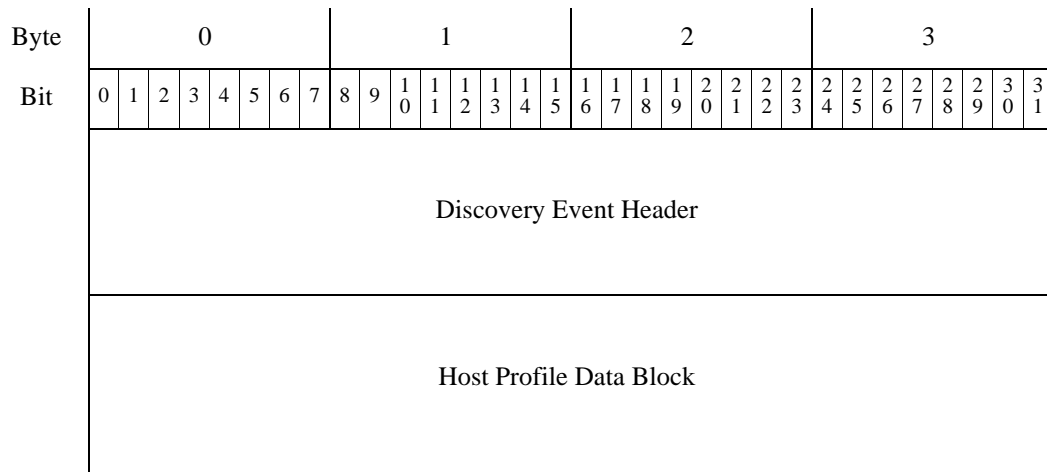
New Host and Host Last Seen event messages have a standard discovery event header and a Host Profile data block (as documented in [Host Profile Data Block for 5.2+](#), page 4-160). The Host Profile data block is block type 139 in series 1.

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-267.



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 4-40) followed by a Server data block (as documented in [Host Server Data Block 4.10.0+](#), page 4-136, block type 103 in series 1):

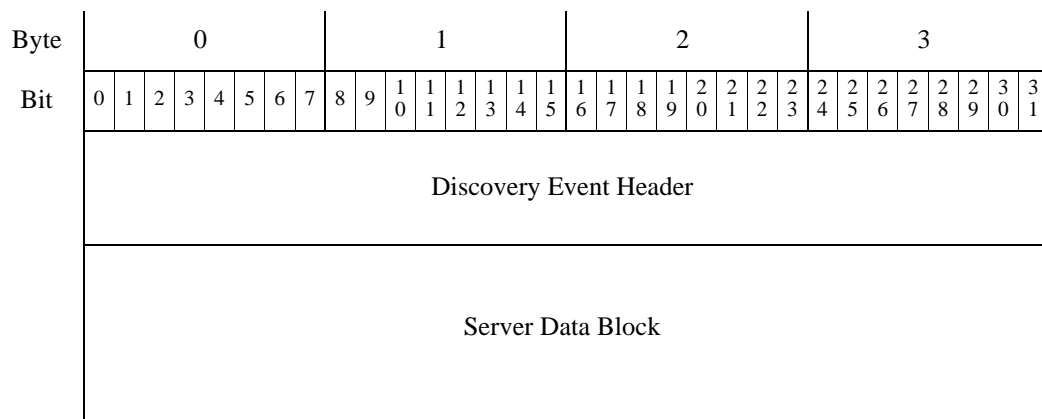
- New TCP Server
- New UDP Server
- TCP Server Information Update
- UDP Server Information Update
- TCP Server Confidence Update
- UDP Server Confidence Update



Note

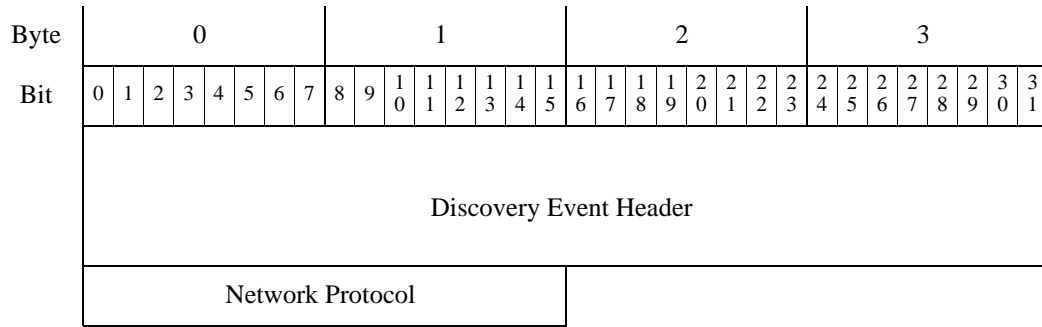
The Server data block differs depending on which system version created the message. For information on the legacy versions of the Server data block, see [Understanding Legacy Data Structures](#), page B-1.

Each of these events use the following format:



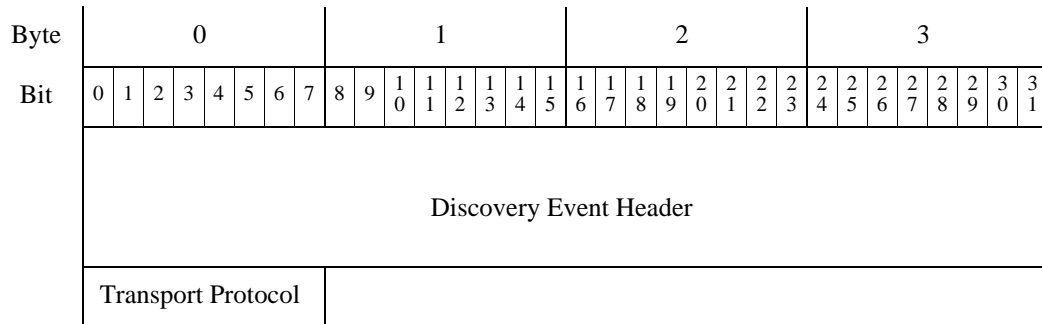
New Network Protocol Message

A New Network Protocol event message has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) 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 4-40](#), block type 4 in series 1) and a one-byte field for the transport protocol number (using values described in following table).



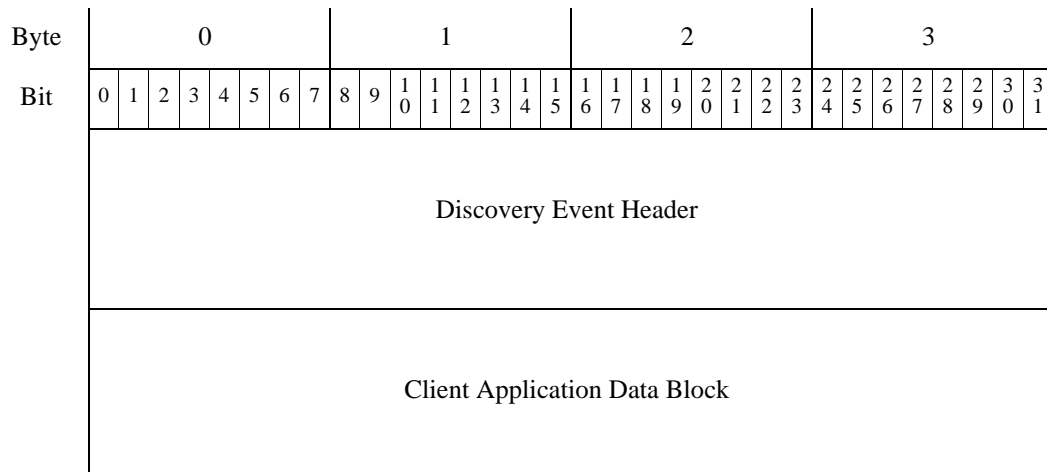
Client Application Messages

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 4-40](#)) followed by a Client Application data block (see [Host Client Application Data Block for 5.0+, page 4-153](#), 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](#).

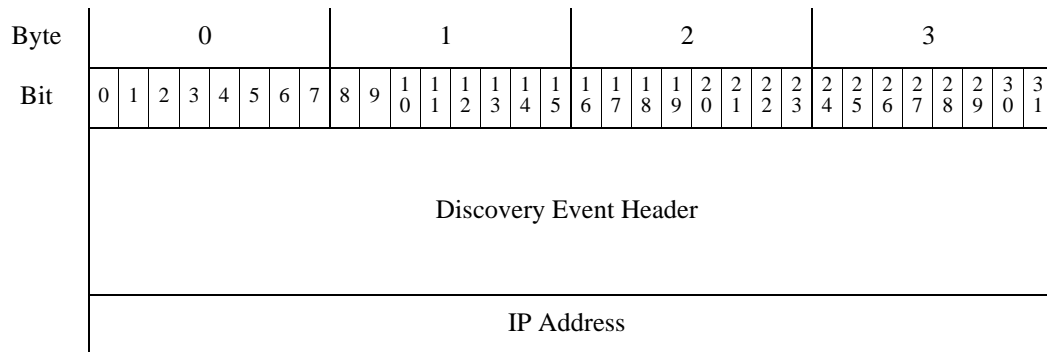


IP Address Change Message

The following host discovery messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) and two different forms, structures, one with four bytes for the IP address and one with 16 bytes for the IP address.

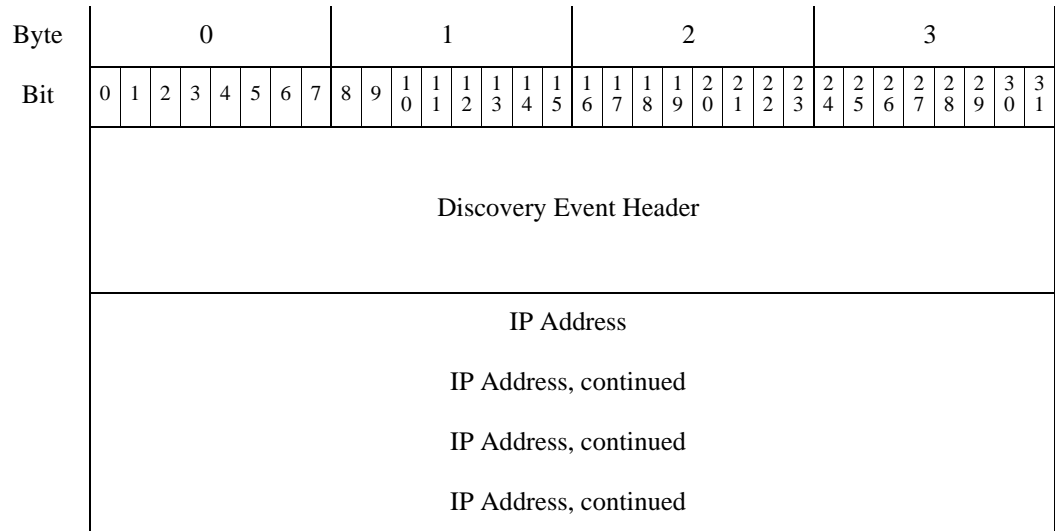
Four bytes are used for the IP address (in IP address octets) in the following case:

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



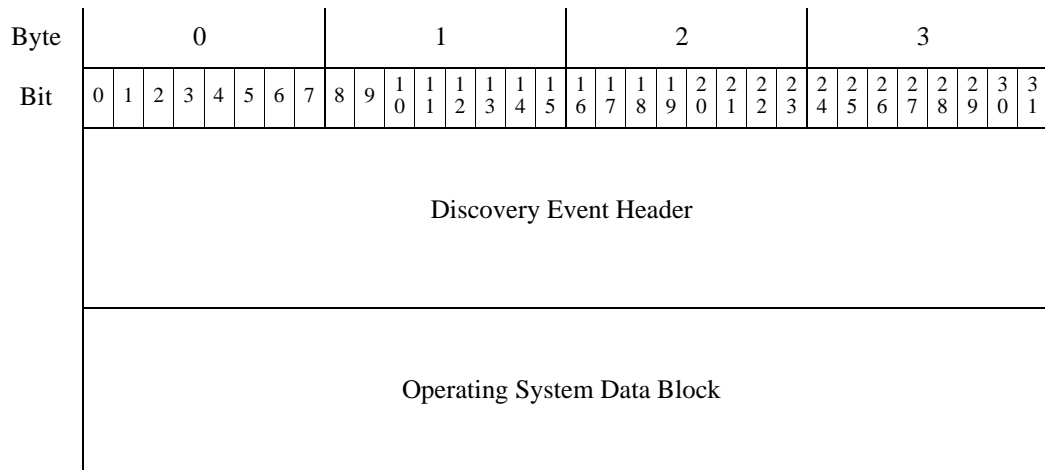
16 bytes are used for the IP address in the following cases:

- 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 4-40](#)) followed by an Operating System data block (as documented in [Operating System Data Block 3.5+, page 4-86](#), block type 53 in series 1).

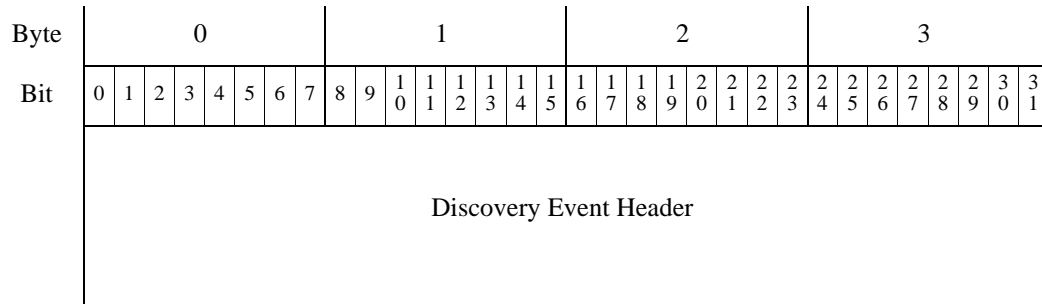


IP Address Reused and Host Timeout/Deleted Messages

The following host event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) with no other data:

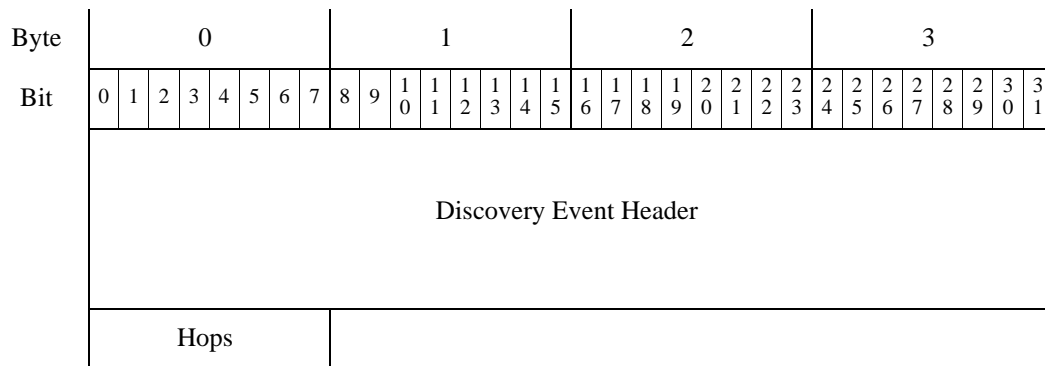
- Host IP Address Reused
- Host Timeout
- Host Deleted: Host Limit Reached

- Host Dropped: Host Limit Reached



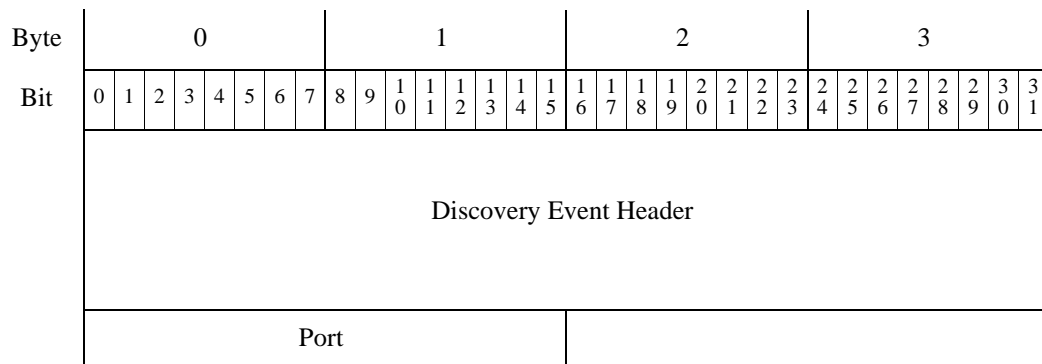
Hops Change Message

A Hops Change event message has a standard discovery event header (as documented in [Discovery Event Header 5.2+](#), page 4-40) 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 4-40) followed by a two-byte field for the port number.



MAC Address Messages

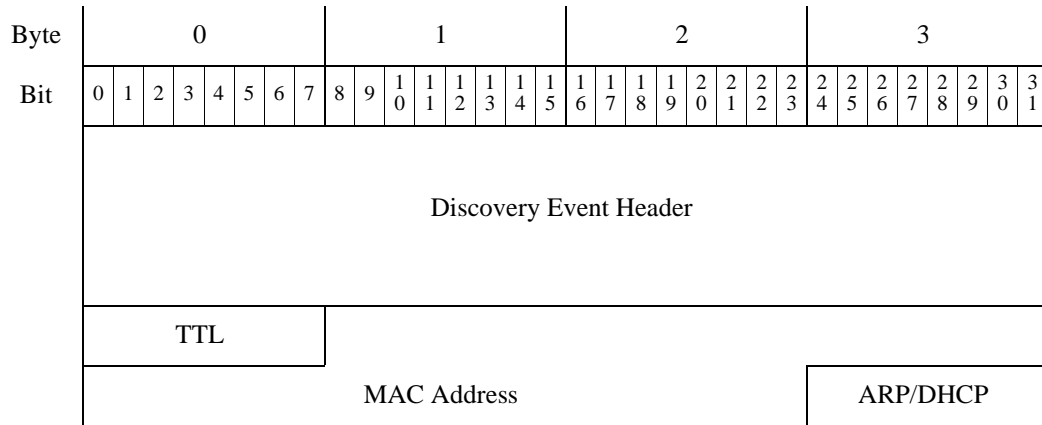
MAC Information Change and Additional MAC Detected for Host messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)), 1 byte for the TTL value, 6 bytes for the MAC address, and 1 byte to indicate whether the MAC address was detected via ARP/DHCP traffic 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 4-51](#). If the data block is 12 bytes in length (20 bytes with the header), see [Host MAC Address 4.9+, page 4-115](#).

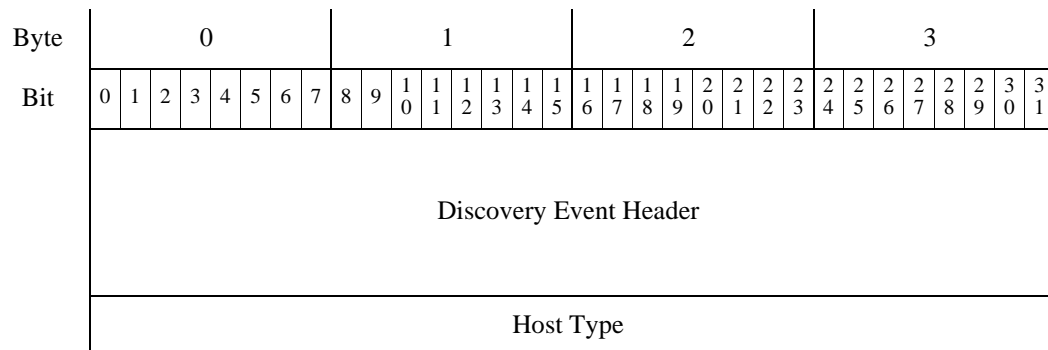
Note that the MAC address data block header is **not** used within MAC Information Change and Additional MAC Detected for Host messages.



Host Identified as a Bridge/Router Message

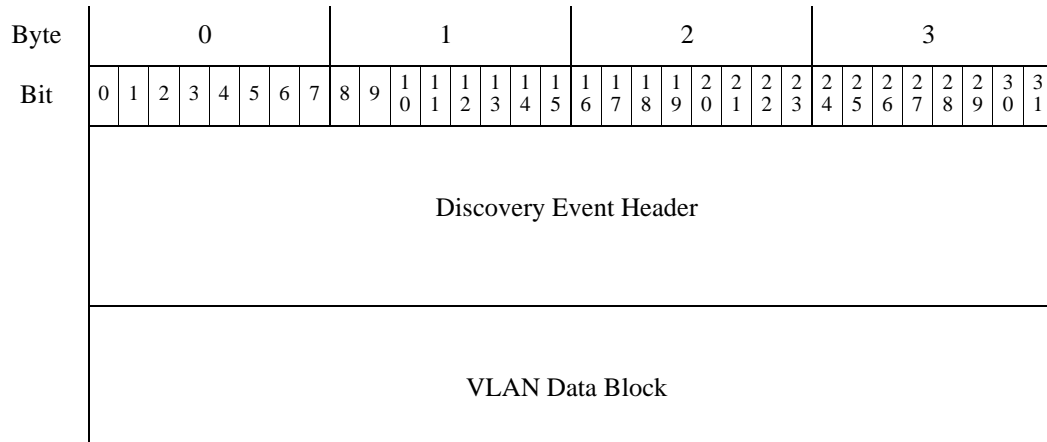
A Host Identified as a Bridge/Router event message has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a four-byte field for the value that matches the host type:

- 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 4-40](#)) followed by VLAN data block (as documented in [VLAN Data Block, page 4-76](#)). The VLAN Data block is block type 14 in the series 1 group of blocks.

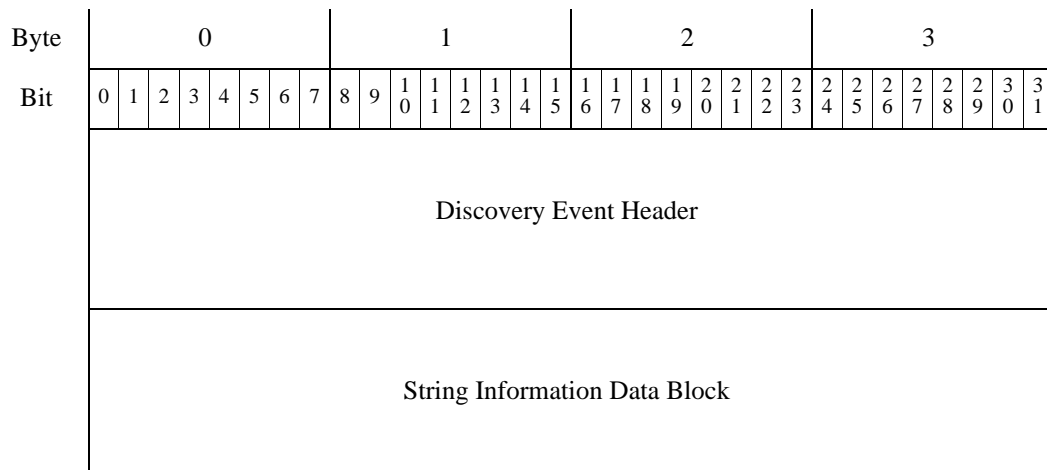


Change NetBIOS Name Message

A Change NetBIOS Name event message has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a String Information data block (as documented in [String Information Data Block, page 4-78](#)). The String Information data block is block type 35 in series 1.

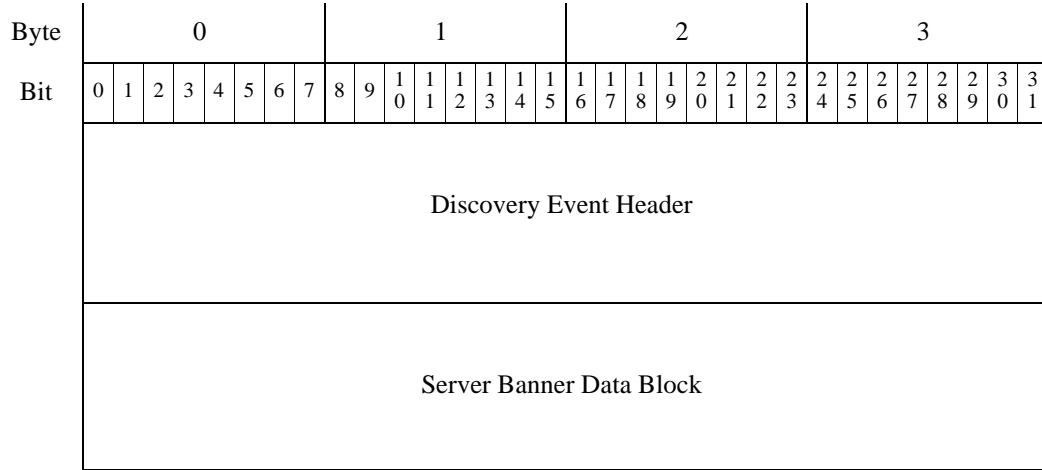

Note

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



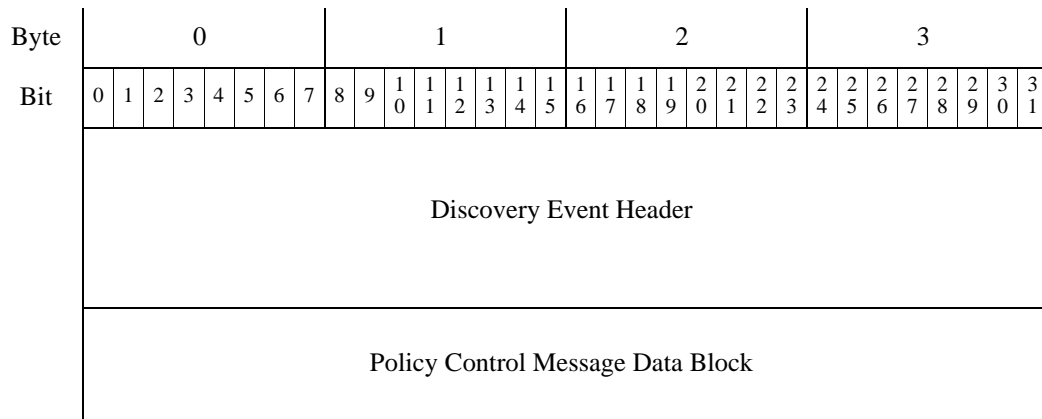
Update Banner Message

An Update Banner event message has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a Server Banner data block (as documented in [Server Banner Data Block, page 4-77](#)). 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 4-40](#)) 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 4-86](#).



Connection Statistics Data Message

The Connection Statistics event has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) 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 6.1+, see [Connection Statistics Data Block 6.2+, page 4-118](#).

**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](#).

| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Connection Statistics Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

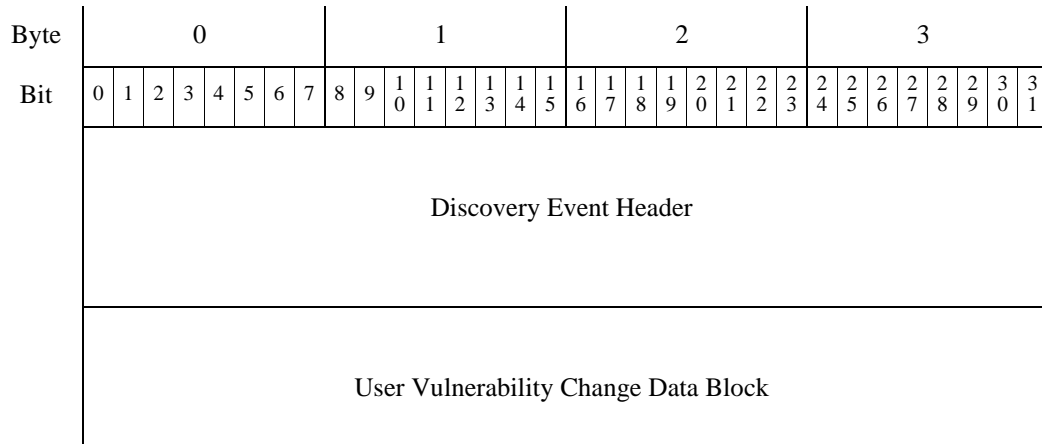
Connection Chunk Message

The Connection Chunk event has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) 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 6.1+, page 4-100](#). The Connection Chunk data block is block type 136 in series 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 |
| Discovery Event Header | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Connection Chunk Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

User Set Vulnerabilities Messages for Version 4.6.1+

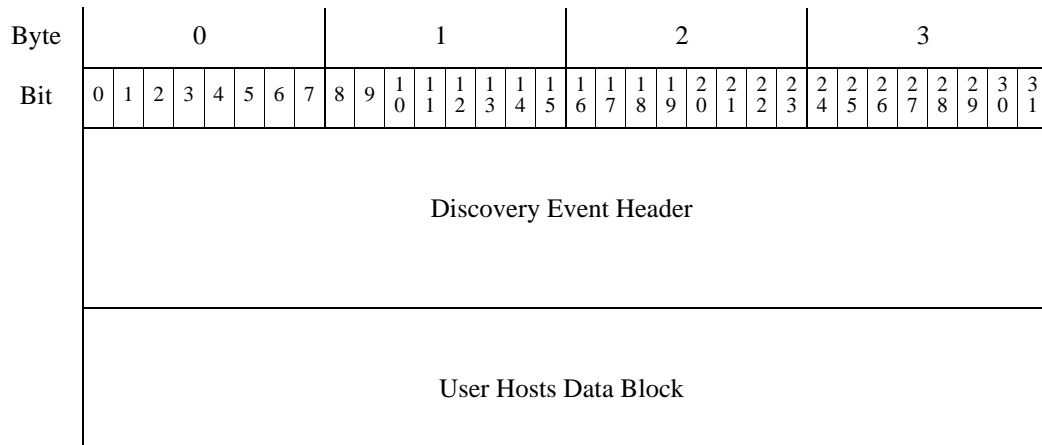
User Set Valid Vulnerabilities, User Set Invalid Vulnerabilities, and User Vulnerability Qualification messages use the same data format: the standard discovery event header (see [Discovery Event Header 5.2+, page 4-40](#)) followed by a User Vulnerability change data block (see [User Vulnerability Change Data Block 4.7+, page 4-106](#), block type 80 in series 1). They are differentiated by record type, event type, and event subtype.



User Add and Delete Host Messages

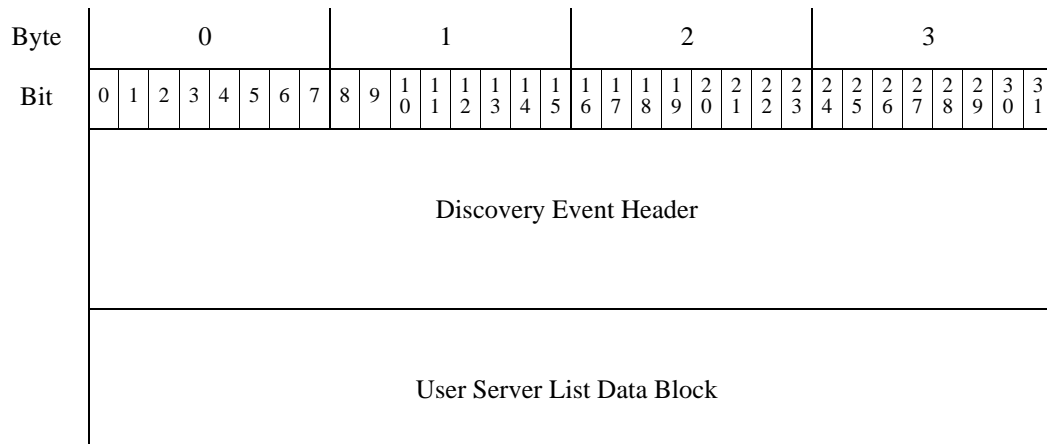
The following host input event messages have the standard discovery event header (see [Discovery Event Header 5.2+](#), page 4-40) followed by a User Hosts data block (see [User Hosts Data Block 4.7+](#), page 4-105, block type 78 in series 1):

- User Delete Address
- User Add Hosts



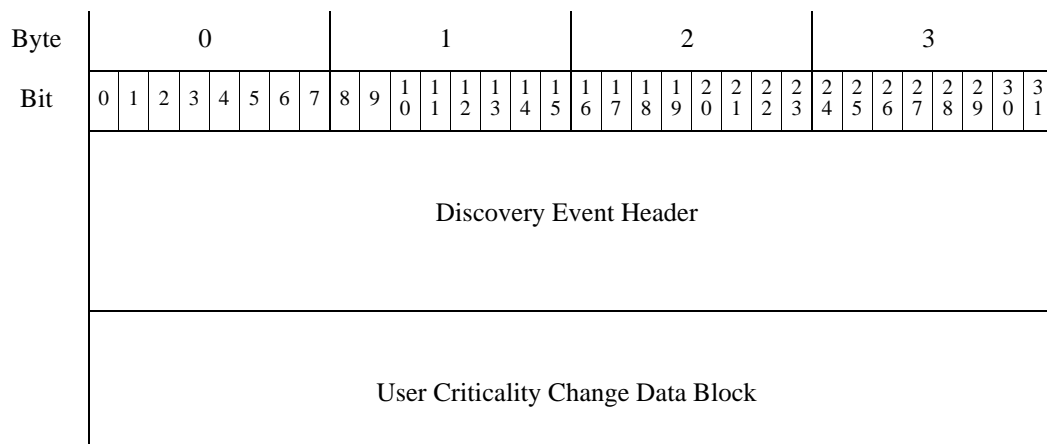
User Delete Server Message

User Delete Server messages have the standard discovery event header (see [Discovery Event Header 5.2+](#), page 4-40) followed by a User Server List data block (see [User Server List Data Block](#), page 4-104). 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 4-40](#)) followed by a User Criticality Change data block (see [User Criticality Change Data Block 4.7+, page 4-108](#)). The User Criticality Change data block is block type 81 in series 1.

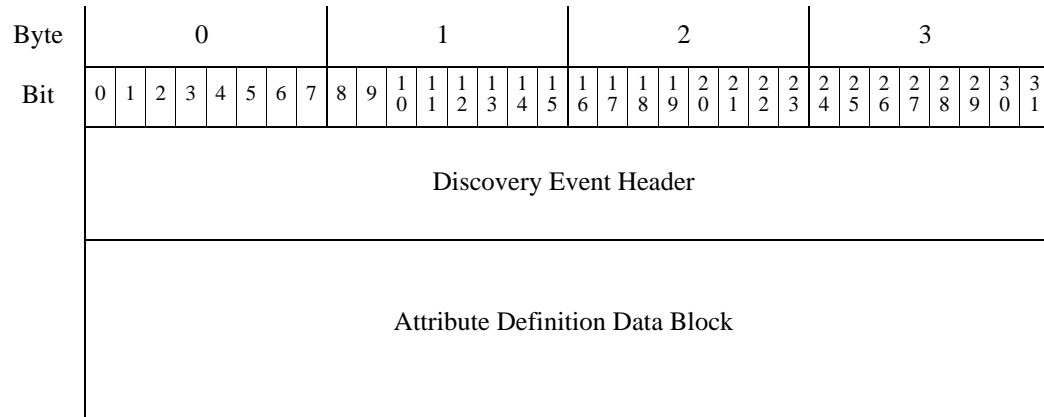


Attribute Messages

The following event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by an Attribute Definition data block (as documented in [Attribute Definition Data Block for 4.7+, page 4-87](#), block type 55 in series 1):

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

Each of these events use the following format:

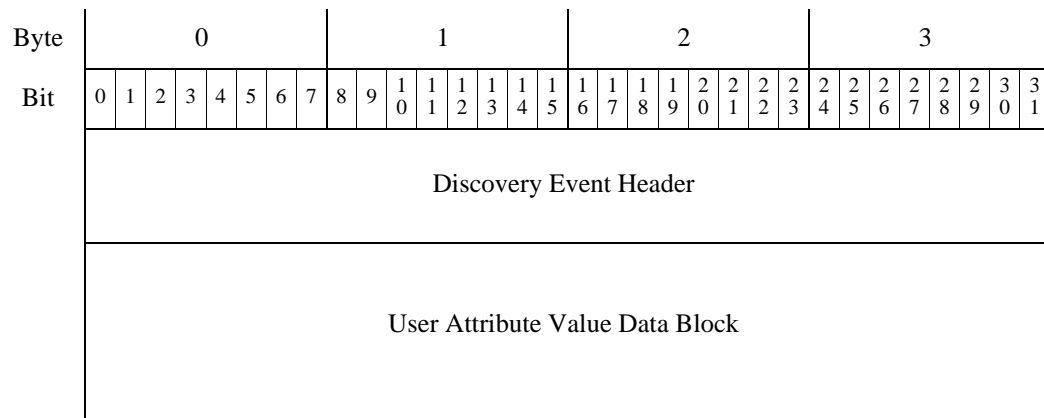


Attribute Value Messages

The following event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a User Attribute Value data block (as documented in [User Attribute Value Data Block 4.7+, page 4-109](#), 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 4-40](#)) followed by a User Product data block (as documented in [User Product Data Block 5.1+, page 4-168](#), block type 60 in series 1):

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

Each of these events use the following format:

| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| User Product Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

User Protocol Messages

The following event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a User Protocol List data block (as documented in [User Protocol List Data Block 4.7+, page 4-111](#), block type 83 in series 1):

- Delete Protocol
- Add Protocol

Each of these events use the following format:

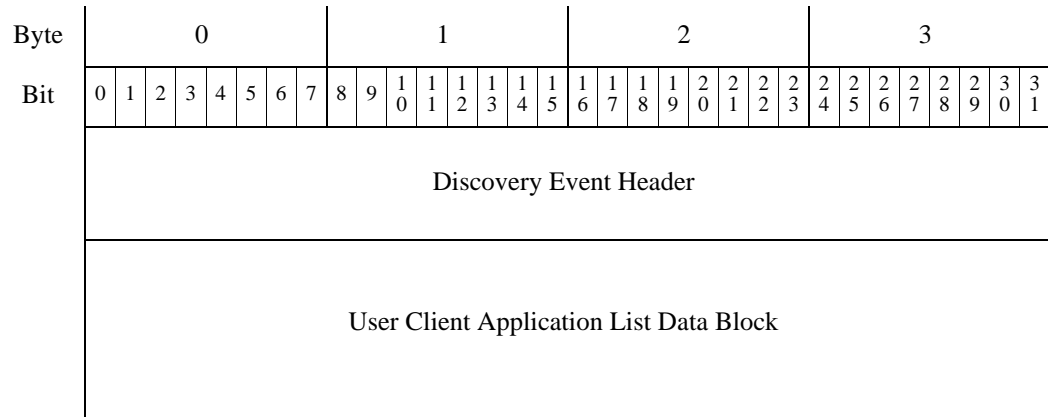
| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|-------------------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| User Protocol List Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

User Client Application Messages

The following event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by a User Client Application List data block (as documented in [User Client Application List Data Block, page 4-93](#), block type 60 in series 1):

- Delete Client Application
- Add Client Application

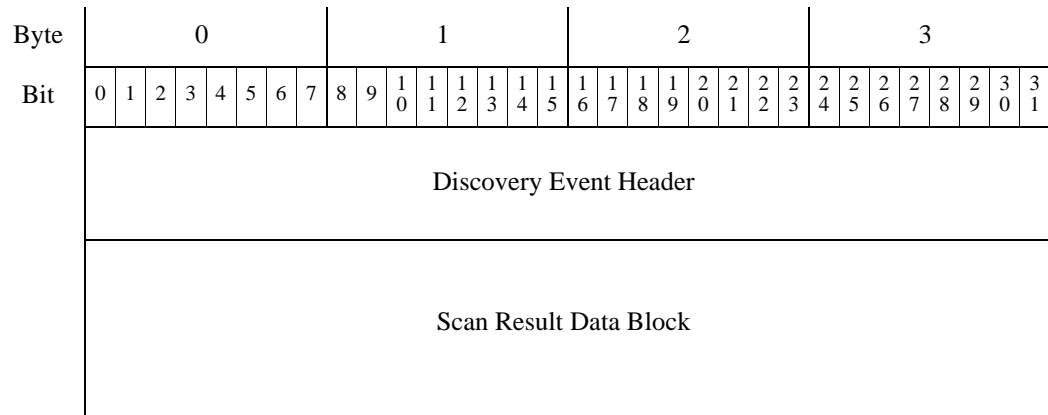
Each of these events use the following format:



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 4-40](#)) followed by a Scan Results data block (as documented in [Scan Result Data Block 5.2+, page 4-134](#)). 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 4-40](#)) followed by an Operating System Fingerprint data block (as documented in [Operating System Fingerprint Data Block 5.1+, page 4-157](#)).

This event uses the following format:

| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Operating System Fingerprint Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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 4-40](#)) followed by an Identity data block (as documented in [Identity Data Block, page 4-113](#)). 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.

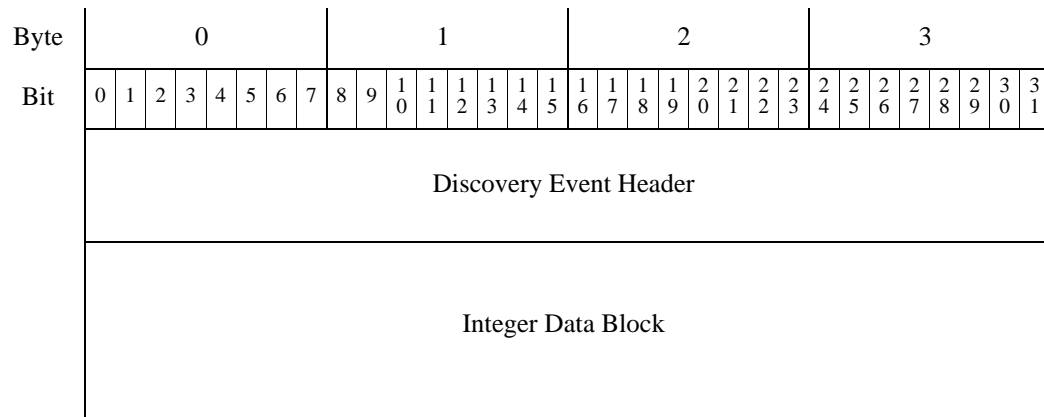
This event uses the following format:

| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Identity Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Host IOC Set Messages

The Host IOC Set message has a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) followed by an integer data block (as documented in [Integer \(INT32\) Data Block, page 4-76](#)). This integer data block contains the ID number of the IOC set for the host.

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:

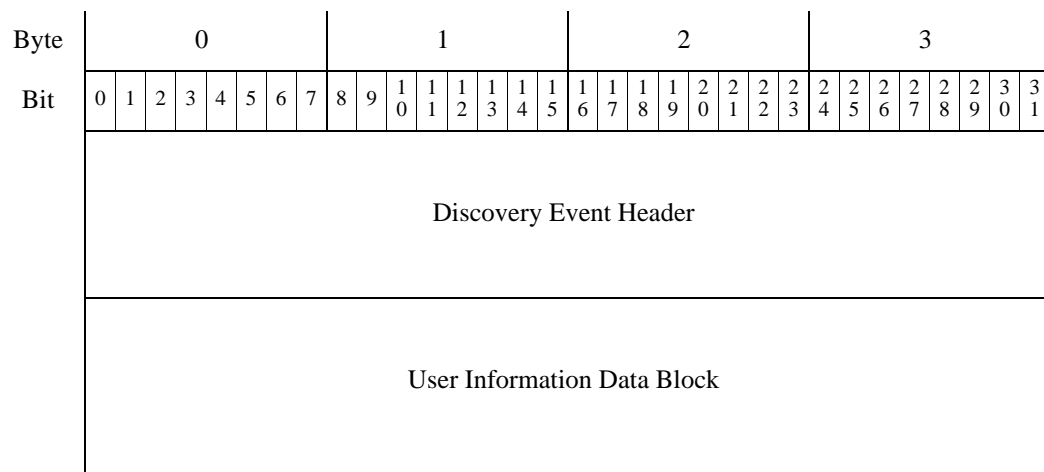
- [User Modification Messages, page 4-61](#)
- [User Information Update Message Block, page 4-62](#)

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 4-40](#)) and a User Information data block (as documented in [User Information Data Block for 6.0+, page 4-185](#)). 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. This block is also used when a user login fails (a failed user login event—event type 1004, subtype 5), when a VPN user logs in (a VPN user login event—event type 1004, subtype 8) or a VPN user logs off (a VPN user logoff event—event type 1004, subtype 9).

User Information Update event messages have a standard discovery event header (as documented in [Discovery Event Header 5.2+, page 4-40](#)) and a User Login Information data block (as documented in [User Login Information Data Block 6.2+, page 4-190](#)). The User Login Information data block is block type 121 in series 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 |
| Discovery Event Header | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| User Login Information Data Block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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.

| 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 |
| Data Block Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Data 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 4-71](#)
- [BLOB Data Block, page 4-72](#)
- [List Data Block, page 4-72](#)
- [Generic List Block, page 4-73](#)

Host Discovery and Connection Data Blocks

For the list of block types in host discovery and connection events, see [Table 4-30 on page 4-63](#). The block types in user events are described in [Table 4-86 on page 4-175](#). 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 4-30 *Host Discovery and Connection Data Block Types*

| Type | Content | Data Block Status | Description |
|------|--------------|-------------------|--|
| 0 | String | Current | Contains string data. See String Data Block, page 4-71 for more information. |
| 1 | Sub-Server | Current | Contains information about a sub-server detected on a server. See Sub-Server Data Block, page 4-74 for more information. |
| 4 | Protocol | Current | Contains protocol data. See Protocol Data Block, page 4-75 for more information. |
| 7 | Integer Data | Current | Contains integer (numeric) data. See Integer (INT32) Data Block, page 4-76 for more information. |
| 10 | BLOB | Current | Contains a raw block of binary data and is used specifically for banners. See BLOB Data Block, page 4-72 for more information. |

Table 4-30 Host Discovery and Connection Data Block Types (continued)

| Type | Content | Data Block Status | Description |
|------|-------------------------------|-------------------|---|
| 11 | List | Current | Contains a list of other data blocks. See List Data Block, page 4-72 for more information. |
| 14 | VLAN | Current | Contains VLAN information. See VLAN Data Block, page 4-76 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 4-73 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 4-78 . |
| 37 | Server Banner | Current | Contains server banner data. See Server Banner Data Block, page 4-77 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 4-81 for more information. |
| 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 4-82 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 4-84 for more information. |
| 53 | Operating System | Current | Contains operating system information for Version 3.5+. See Operating System Data Block 3.5+, page 4-86 for more information. |
| 54 | Policy Engine Control Message | Current | Contains information on user policy control changes. See Policy Engine Control Message Data Block, page 4-86 for more information. |

Table 4-30 *Host Discovery and Connection Data Block Types (continued)*

| Type | Content | Data Block Status | Description |
|------|------------------------------|-------------------|--|
| 55 | Attribute Definition | Current | Contains information on attribute definitions. See Attribute Definition Data Block for 4.7+, page 4-87 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 4-90 for more information. |
| 59 | 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 4-93 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-303 for more information. Superseded by block 141. |
| 62 | Attribute Specification | Current | Contains an attribute name and value. See Attribute Specification Data Block, page 4-96 for more information. |
| 63 | MAC Address Specification | Current | Contains MAC address range specifications. See MAC Address Specification Data Block, page 4-98 for more information. |
| 64 | IP Address Specification | Current | Contains lists of IP and MAC address specification blocks. See Address Specification Data Block, page 4-99 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-141 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 4-102 for more information. |
| 71 | Generic Scan Results | Legacy | Contains results from an Nmap scan (as documented in earlier versions of the product). |

Table 4-30 Host Discovery and Connection Data Block Types (continued)

| Type | Content | Data Block Status | Description |
|------|--------------------------------|-------------------|---|
| 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 4-102 for more information. |
| 77 | User Server List | Current | Contains lists of user server blocks. See User Server List Data Block, page 4-104 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 4-105 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 4-106 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 4-108 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 4-109 for more information. |
| 83 | User Protocol List | Current | Contains lists of protocols for a host or hosts. See User Protocol List Data Block 4.7+, page 4-111 for more information. |
| 85 | Vulnerability List | Current | Contains vulnerabilities that apply to a host. See Host Vulnerability Data Block 4.9.0+, page 4-112 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-123 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). |

Table 4-30 Host Discovery and Connection Data Block Types (continued)

| Type | Content | Data Block Status | Description |
|------|-------------------------|-------------------|---|
| 91 | Host Profile | Legacy | Contains profile information for a host. See Host Profile Data Block for 5.2+ , page 4-160 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 4-113 for more information. |
| 95 | Host MAC Address | Current | Contains MAC address information for a host. See Host MAC Address 4.9+ , page 4-115 for more information. |
| 96 | Secondary Host Update | Current | Contains lists of MAC address information reported by a secondary Secondary Host Update , page 4-116. |
| 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). |
| 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 4-136 for more information. |
| 104 | Full Host Server | Current | Contains server information for a host. See Full Host Server Data Block 4.10.0+ , page 4-138 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 4-142 for more information. The successor block type 117 introduced for 5.0 has an identical structure as block type 105. |

Table 4-30 *Host Discovery and Connection Data Block Types (continued)*

| Type | Content | Data Block Status | Description |
|------|------------------------------|-------------------|---|
| 106 | Full Server Information | Current | Contains information about a server detected on a host. See Full Server Information Data Block , page 4-144 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 4-147 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 4-149. |
| 111 | Full Host Profile | Legacy | Contains complete host profile information. See Full Host Profile Data Block 5.0 - 5.0.2 , page B-268 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 4-152 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-125 for more information. The successor block introduced for version 5.1 has block type 126. |
| 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 4-142 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-141 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 4-153 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 4-117 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 4-155. It supersedes block type 79. |

Table 4-30 *Host Discovery and Connection Data Block Types (continued)*

| Type | Content | Data Block Status | Description |
|------|------------------------------|-------------------|---|
| 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-129 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 4-157 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 4-159 for more information. |
| 132 | Host Profile | Legacy | Contains profile information for a host. See Full Host Profile Data Block 5.2.x, page B-285 for more information. It supersedes block type 91. Superseded by block 139. |
| 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 4-168 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-277 for more information. Supersedes data block 111. |
| 136 | Connection Chunk | Current | Contains connection chunk information. See Connection Chunk Data Block for 6.1+, page 4-100 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-141 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 4-92 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 4-160 for more information. It supersedes block type 132. |

Table 4-30 *Host Discovery and Connection Data Block Types (continued)*

| Type | Content | Data Block Status | Description |
|------|------------------------|-------------------|--|
| 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 4-95 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 Scan Result Data Block 5.2+ , page 4-134. 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 4-97 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-135 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 4-79 for more information. It supersedes block type 38. |
| 148 | User IOC Change | Current | Contains information about user changes to IOCs. See User IOC Change Data Block 5.3+ , page 4-80 for more information. |
| 149 | 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-150 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-156 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-163 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-176 for more information. It supersedes block type 155. |

Table 4-30 Host Discovery and Connection Data Block Types (continued)

| Type | Content | Data Block Status | Description |
|------|-----------------------|-------------------|--|
| 160 | Connection Statistics | Legacy | Contains information for connection events in 5.4.1. See Connection Statistics Data Block 6.0.x , page B-189 for more information. It supersedes block type 157. |
| 163 | Connection Statistics | Current | Contains information for connection events in 6.0+. See Connection Statistics Data Block 6.2+ , page 4-118 for more information. It supersedes block type 160. |

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.

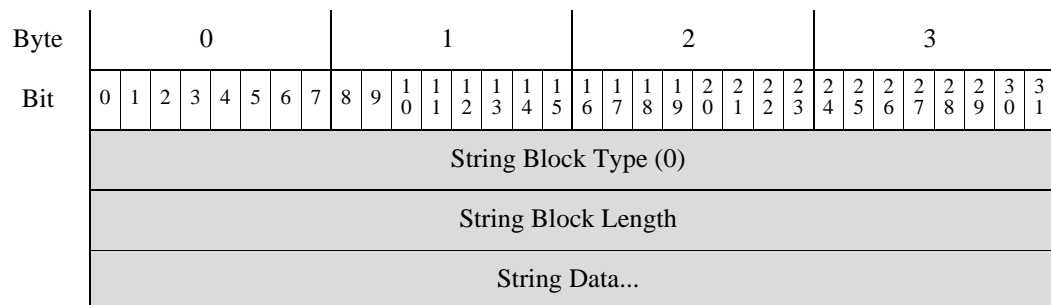
The String data block has a block type of 0 in the series 1 group of blocks.



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.

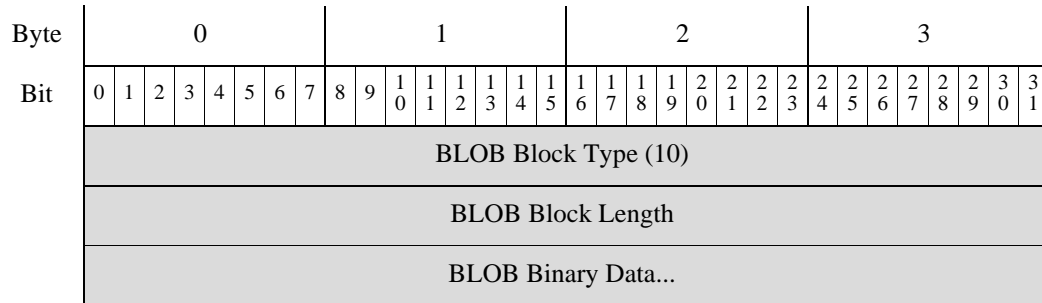
Table 4-31 String Data Block Fields

| 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.

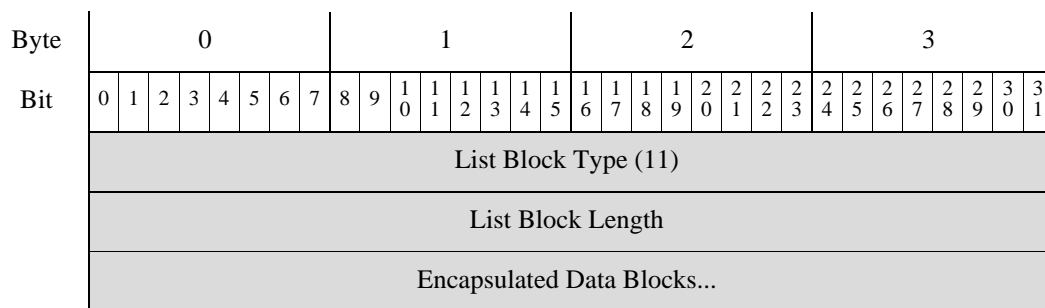
Table 4-32 BLOB Data Block Fields

| 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.

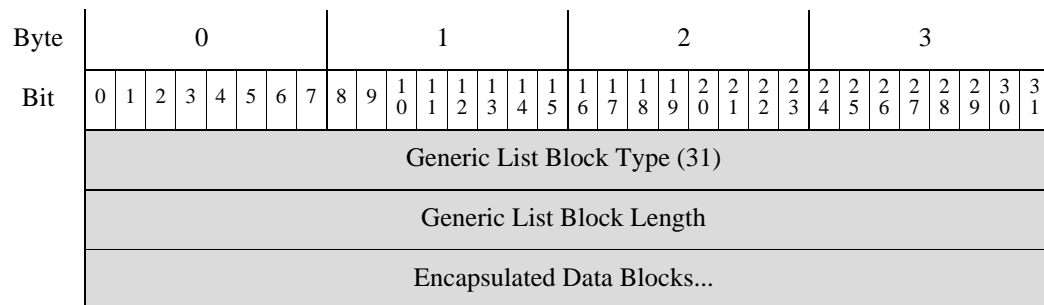
Table 4-33 List Data Block Fields

| 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.

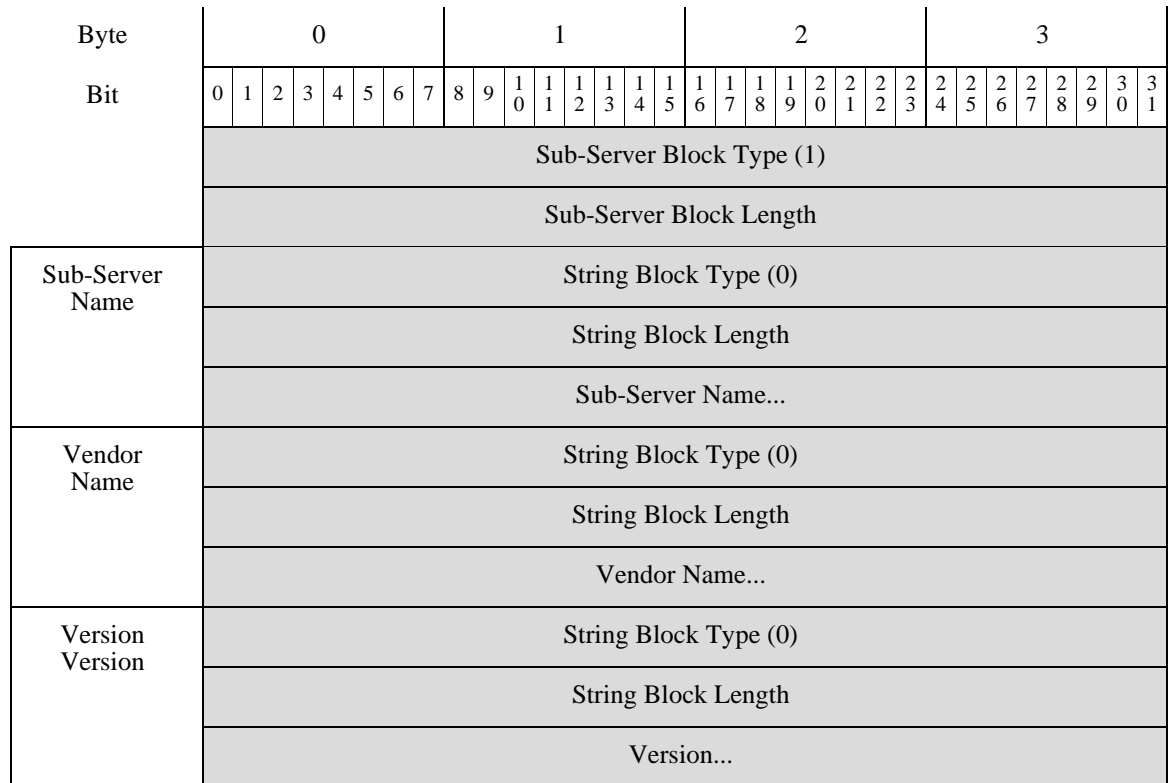
Table 4-34 Generic List Data Block Fields

| 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 4-35 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. |

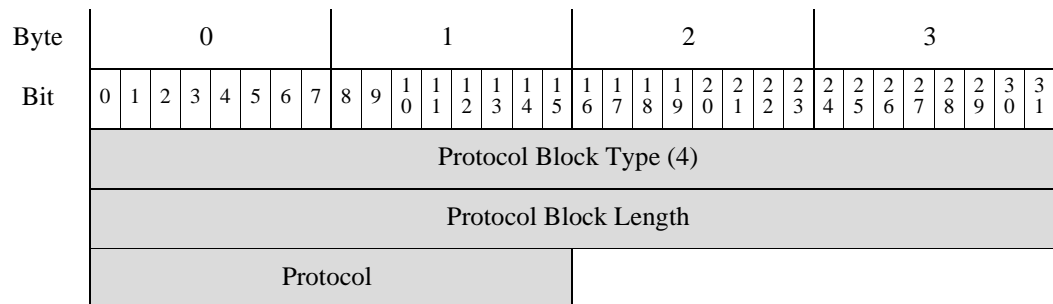
Table 4-35 Sub-Server Data Block Fields (continued)

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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 4-36 Protocol Data Block Fields

| Field | Data Type | Description |
|---------------------|-----------|--|
| Protocol Block Type | uint32 | Initiates a Protocol data block. This value is always 4. |

Table 4-36 Protocol Data Block Fields (continued)

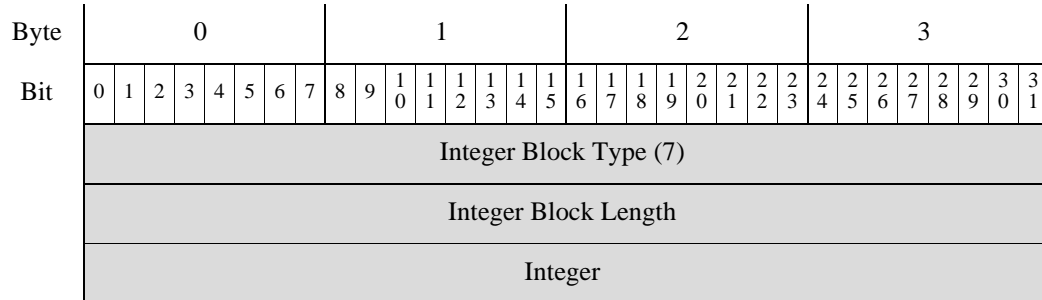
| Field | Data Type | Description |
|-----------------------|-----------|---|
| Protocol Block Length | uint32 | Number of bytes in the Protocol data block. This value is always 10. |
| Protocol | uint16 | IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: <ul style="list-style-type: none"> • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> • 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:



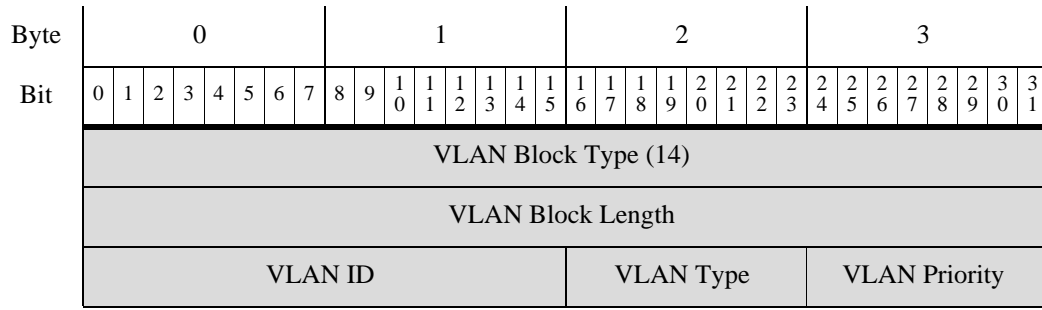
The following table describes the fields of the Integer data block:

Table 4-37 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 4-38 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. <ul style="list-style-type: none"> 0 — Ethernet 1 — Token Ring |
| VLAN Priority | uint8 | Priority value included in the VLAN tag. |

Server Banner Data Block

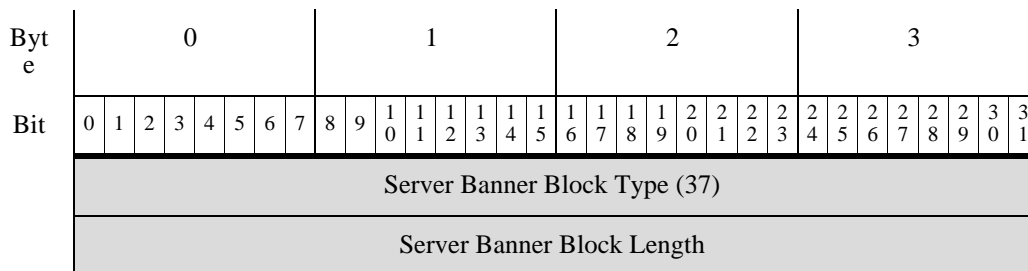
The Server Banner data block provides information about the banner for a server running on a host. It contains the server port, protocol, and the banner data. The Server Banner data block has a block type of 37 in the series 1 group of blocks.

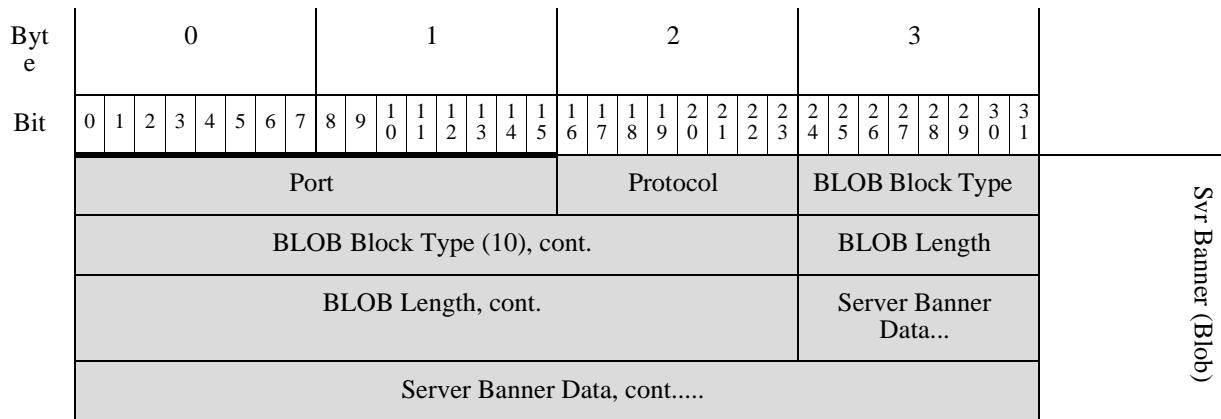
The following diagram shows the format of the Server Banner data block.



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.

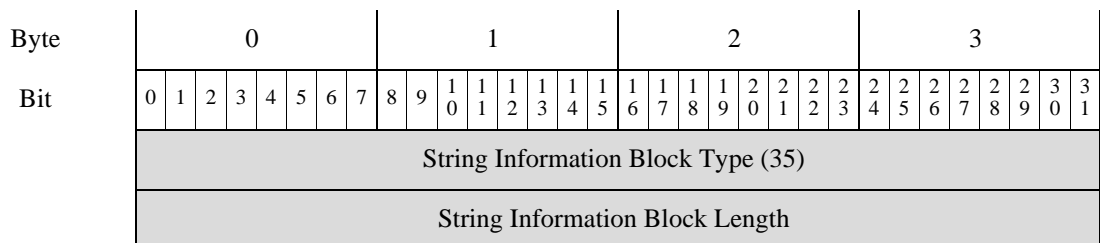
Table 4-39 Server Banner Data Block Fields

| 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 <i>n</i> bytes of the packet involved in the server event, where <i>n</i> 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:



| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 |
| CVE ID | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Value... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-40 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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 Address Block Type (146) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Address Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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 |
| | IP Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Bits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-41 Attribute Address Data Block 5.2+ Fields

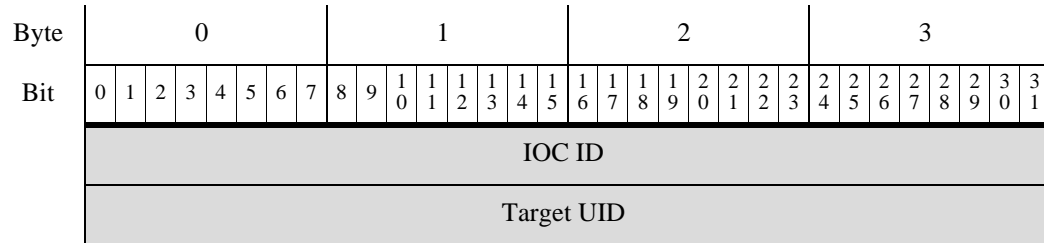
| Field | Data Type | Description |
|--------------------------------|-----------|--|
| Attribute Address Block Type | uint32 | Initiates an Attribute Address data block. This value is always 146. |
| Attribute Address Block Length | uint32 | Number of bytes in the Attribute Address data block, including 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. |

User IOC Change Data Block 5.3+

The User IOC Change data block contains information regarding IOC changes made by a user. It is used within the User Host IOC Delete, User Host IOC Enable, and User Host IOC Disable records. It has a block type of 148 in the series 1 group of blocks.

The following diagram shows the basic structure of a User IOC 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 IOC Change Block Type (148) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IP Address Ranges | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Range Specification Data Blocks* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



The following table describes the fields of the User IOC Change data block.

Table 4-42 User IOC Change Data Block 5.3+ Fields

| Field | Data Type | Description |
|--------------------------------------|-----------|--|
| User IOC Change Block Type | uint32 | Initiates a User IOC Change data block. This value is always 148. |
| User ID | uint32 | ID number of the user who made the IOC change. |
| Source Type | uint32 | Number that maps to the type of data source: <ul style="list-style-type: none"> 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 <code>nmimport.pl</code> 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 4-95 for a description of this data block. |
| IOC ID | uint32 | ID number of the IOC being changed. |
| Target UID | uint32 | Not used in events supported for eStreamer output. |

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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|-------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 List Item Block Type (39) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute List Item Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Attr Name | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-43 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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 Value Block Type (48) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Value Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Integer Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Data Block (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Value String... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the Attribute Value data block.

Table 4-44 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: <ul style="list-style-type: none"> 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* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

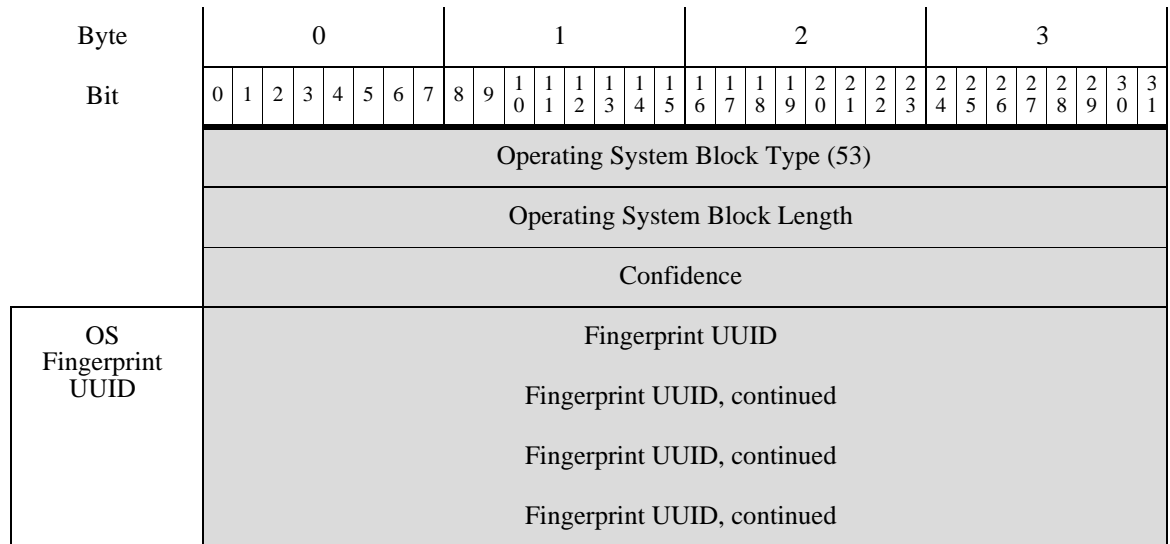
The following table describes the components of the Full Sub-Server data block.

Table 4-45 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 4-112 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 4-112 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+.



The following table describes the fields of the v3.5 operating system data block.

Table 4-46 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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 Engine Control Message Block Type (54) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Policy Engine Control Message Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Control Message | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Control Message... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the Policy Engine Control Message data block.

Table 4-47 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 4-56](#).

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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-----------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------------|--|
| 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 | |
| Address List | Attribute Address Block Type (38) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | List of Attribute Addresses | |
| | Attribute Address Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | List Block Type (11) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute Address List... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-48 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: <ul style="list-style-type: none"> 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 4-48 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 4-81 . |
| 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 4-79 . |

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:

| 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 Protocol Block Type (57) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Protocol Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IP Address Ranges | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Range Specification Data Blocks* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MAC Add. Ranges | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | MAC Range Specification Data Blocks... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Protocol Type | | | | | | | | | | | | | | | | Protocol | | | | | | | | | | | | | | | |

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

Table 4-49 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 4-95 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 4-98 for a description of this data block. |

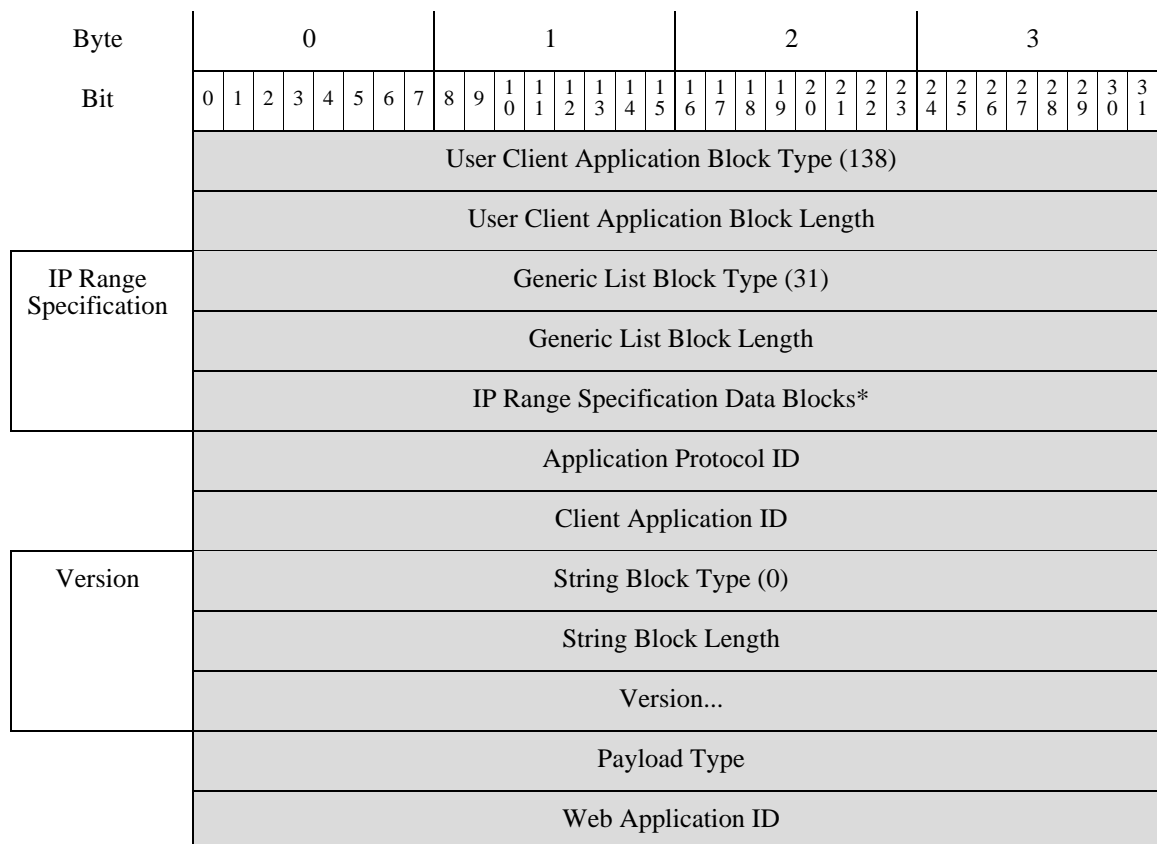
Table 4-49 User Protocol Data Block Fields (continued)

| 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 6.7, 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 59.

The following diagram shows the basic structure of a User Client Application data block:



The following table describes the fields of the User Client Application data block.

Table 4-50 *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 4-95 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. |
| 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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 (60) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Client Application Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| User Client App List Blocks | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Client Application List Data Blocks... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the User Client Application List data block.

Table 4-51 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: <ul style="list-style-type: none"> 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 <code>nmimport.pl</code> 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. |

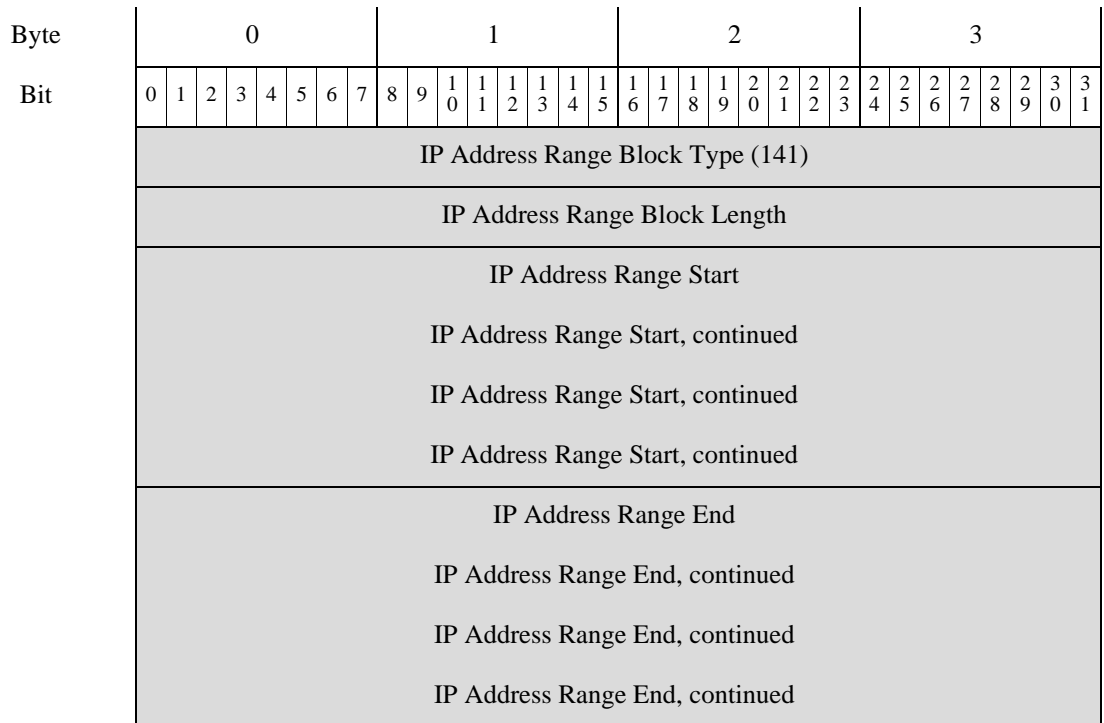
Table 4-51 User Client Application List Data Block Fields (continued)

| 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 4-92. |

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.

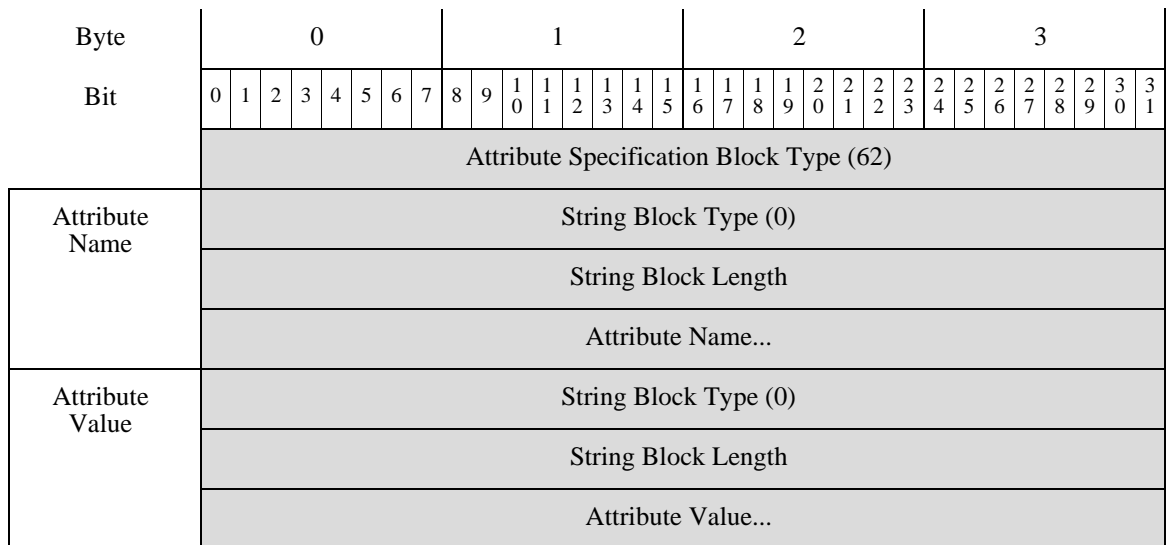
Table 4-52 IP Address Range Data Block Fields

| 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 4-53 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. |

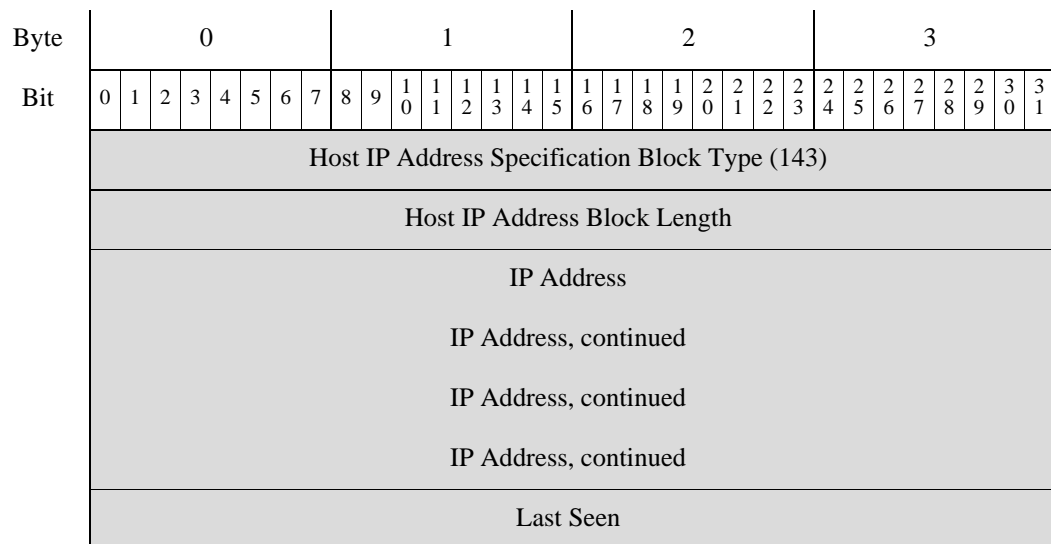
Table 4-53 Attribute Specification Data Block Fields (continued)

| 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.

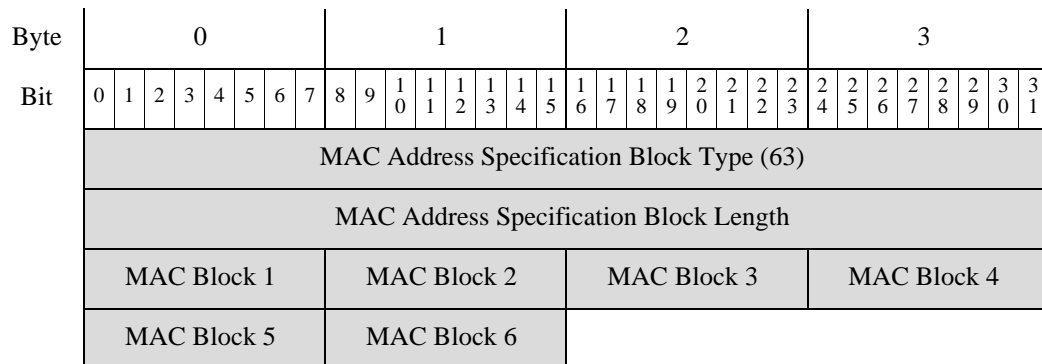
Table 4-54 Host IP Address Data Block Fields

| 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.

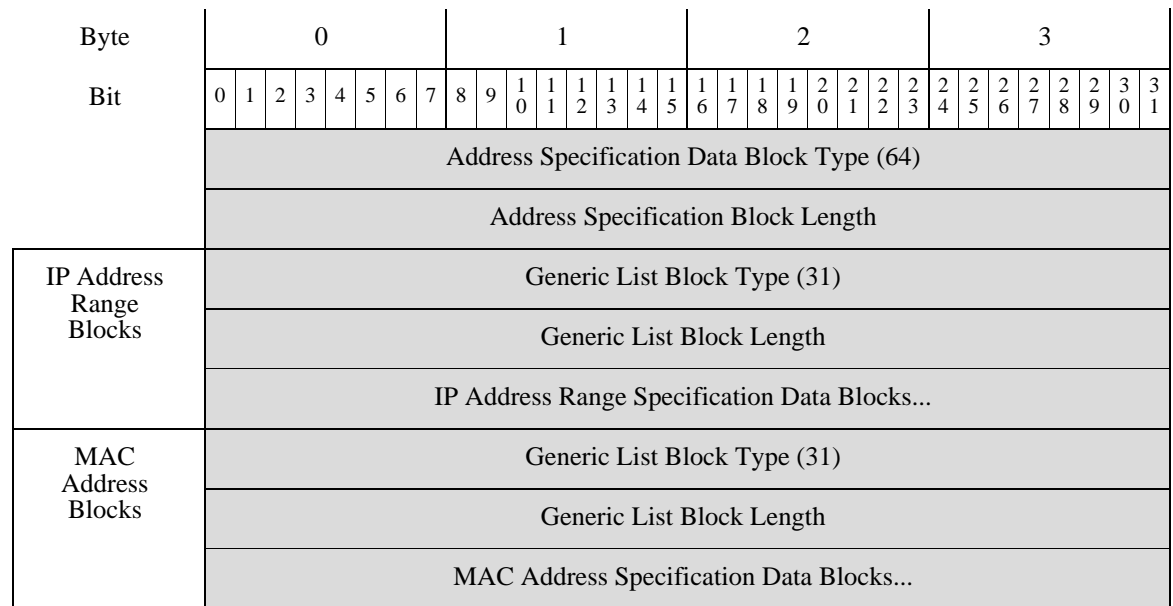
Table 4-55 MAC Address Specification Data Block Fields

| 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 4-56 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 4-56 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 4-95 . |
| 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 4-98 . |

Connection Chunk Data Block for 6.1+

The Connection Chunk data block conveys connection data. It stores connection log data that aggregates over a five-minute period. The version for 6.1+ introduces the new field Original Client IP Address. The Connection Chunk data block has a block type of 164 in the series 1 group of blocks. It supersedes block type 136.

The following diagram shows the format of the Connection Chunk 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 |
| Connection Chunk Block Type (136) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Connection Chunk Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Initiator IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Responder IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Original Client IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Start Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Application Protocol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Responder Port | | | | | | | | | | | | | | | | Protocol | | | | | | | | Connection Type | | | | | | | | |
| NetFlow Detector 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 |
| Packets Sent | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Packets Sent, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Packets Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Packets Received, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bytes Sent | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bytes Sent, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bytes Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bytes Received, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Connections | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the Connection Chunk data block.

Table 4-57 Connection Chunk Data Block Fields

| Field | Data Type | Description |
|-------------------------------|-----------|--|
| Connection Chunk Block Type | uint32 | Initiates a Connection Chunk data block. This value is always 164. |
| 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 original client and responder IP addresses 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 and original client IP addresses to identify identical connections. |
| Original Client IP Address | uint8(4) | IP address of the host behind the proxy that originated the request. This is used with the initiator and responder IP addresses 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. |

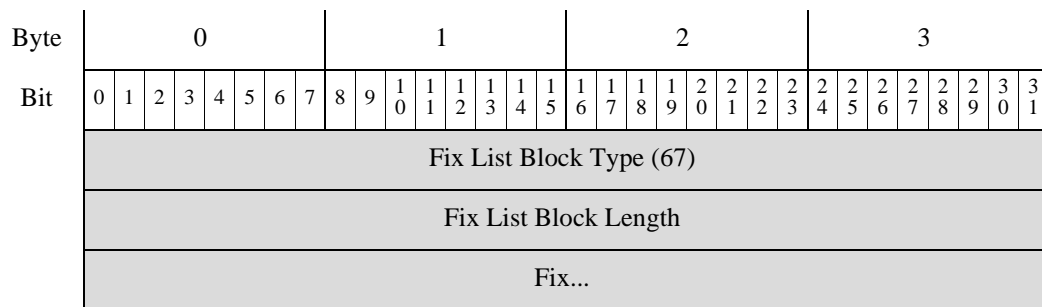
Table 4-57 Connection Chunk Data Block Fields (continued)

| Field | Data Type | Description |
|----------------------------|-----------|--|
| NetFlowDetector 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 4-58 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:

| 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 Data Block Type (76) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Server Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IP Range Specification | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Address Range Specification Data Blocks* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Port | | | | | | | | | | | | | | | | Protocol | | | | | | | | | | | | | | | |

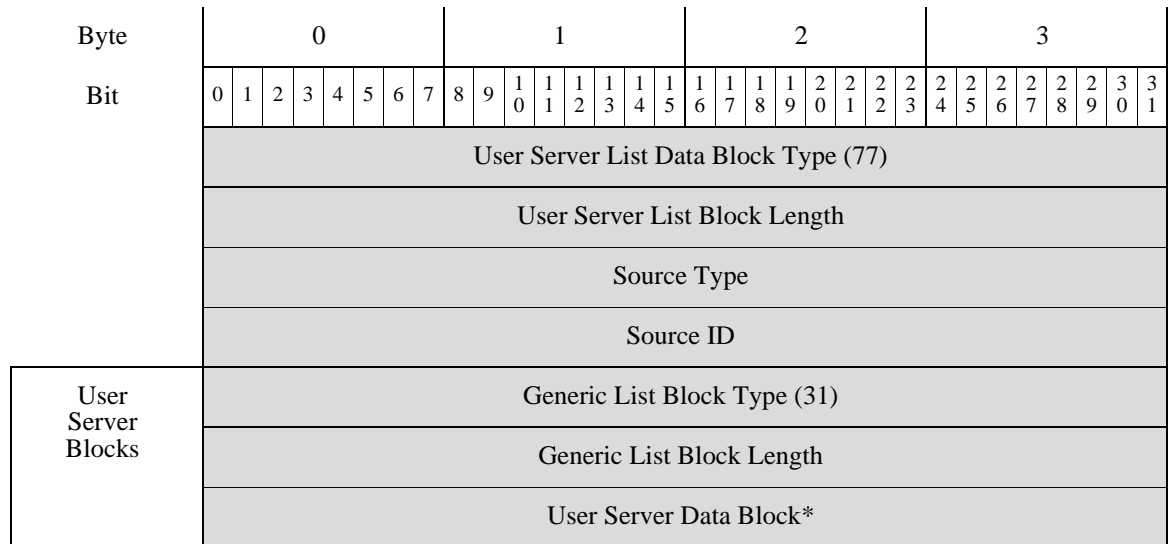
The following table describes the fields of the User Server data block.

Table 4-59 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: <ul style="list-style-type: none"> 6 — TCP 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> 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:



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

Table 4-60 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: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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. |

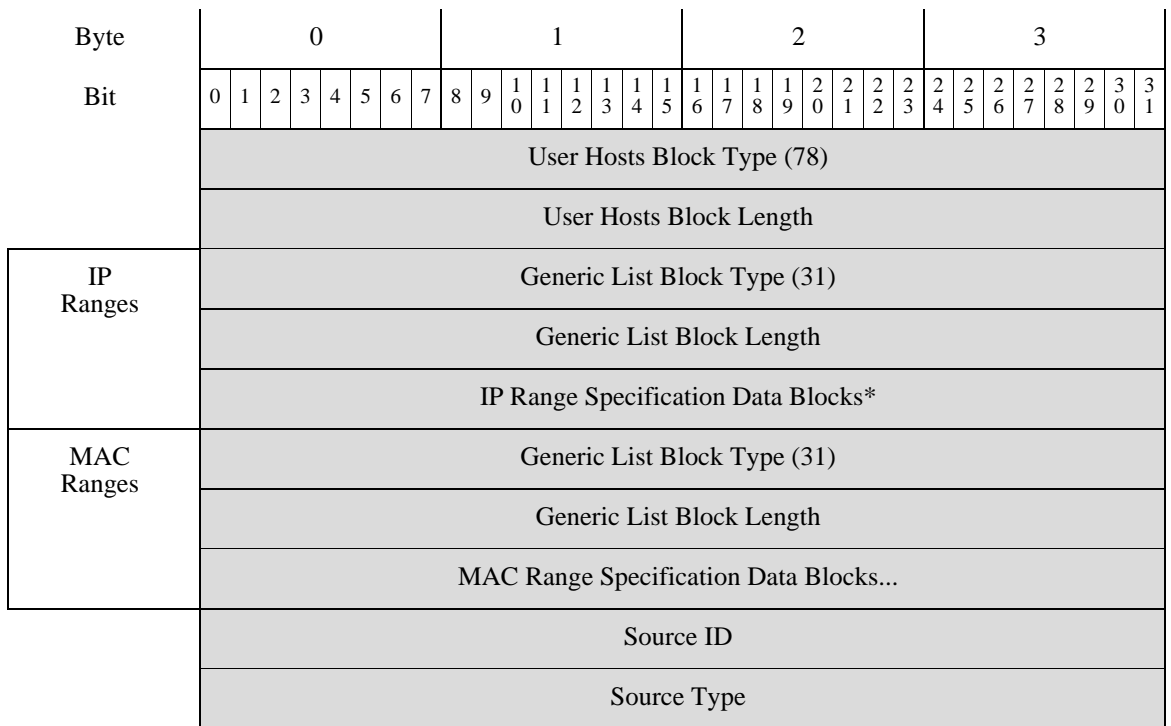
Table 4-60 User Server List Data Block Fields (continued)

| 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 4-55](#) 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 4-61 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 4-95 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 4-98 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: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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 4-54.

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 4-62 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: <ul style="list-style-type: none"> 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 <code>nmimport.pl</code> 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 4-62 User Vulnerability Change Data Block Fields (continued)

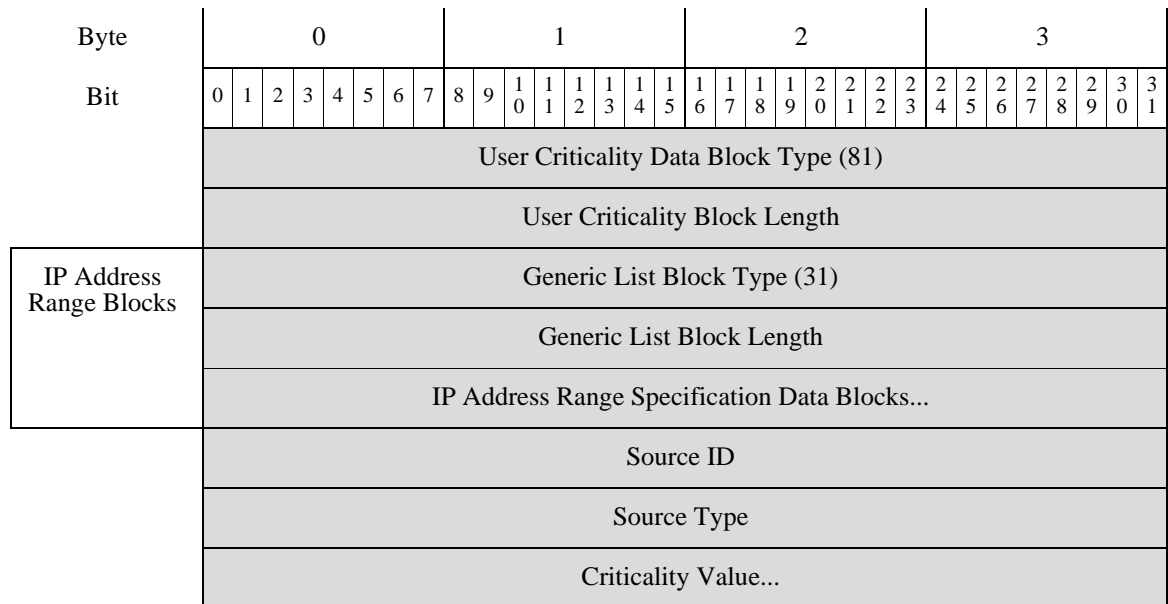
| Field | Number of Bytes | Description |
|--------------------------------|-----------------|--|
| Generic List Block Length | uint32 | Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number 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 4-155. |

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 4-56.

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 4-63 *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: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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:

| 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 Attribute Value Data Block Type (82) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Attribute Value Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IP Address Range Blocks | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Address Range Specification Data Blocks... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Attribute ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | BLOB Block Type (10) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | BLOB Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Value... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the User Attribute Value data block.

Table 4-64 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 4-64 User Attribute Value Data Block Fields (continued)

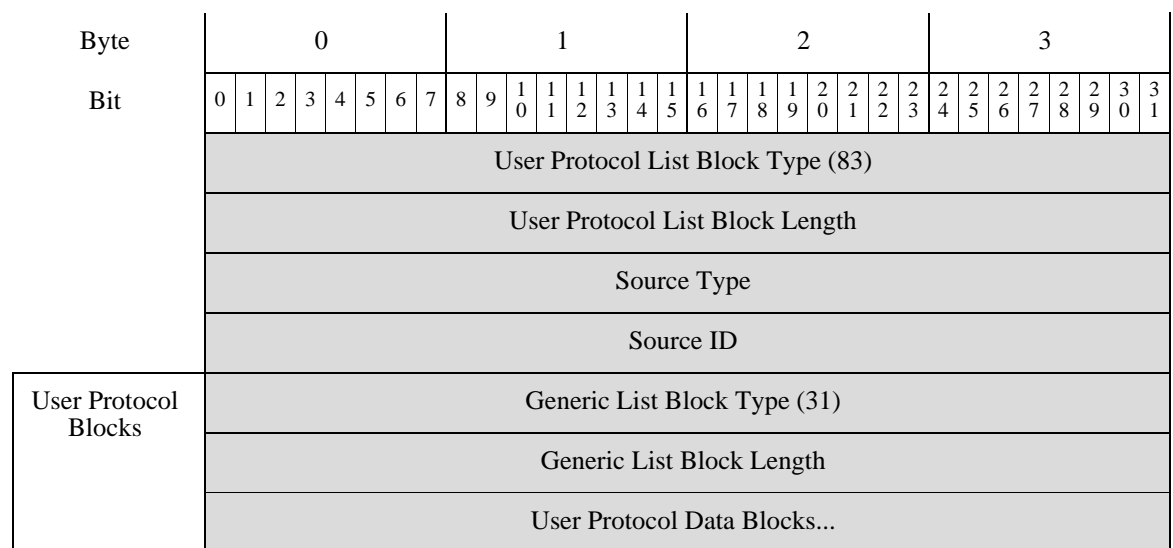
| Field | Number of Bytes | Description |
|-------------------|-----------------|--|
| Source Type | uint32 | Number that maps to the type of data source: <ul style="list-style-type: none"> • 0 if the user attribute value was provided by RNA • 1 if the user attribute value was provided by a user • 2 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 <code>nmimport.pl</code> 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 4-90](#).

The User Protocol List data block is used in user protocol messages, as documented in [User Protocol Messages, page 4-58](#).

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.

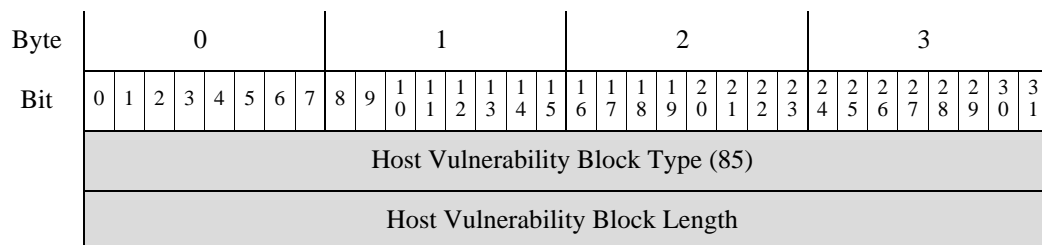
Table 4-65 User Protocol List Data Block Fields

| 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: <ul style="list-style-type: none"> 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 <code>nmimport.pl</code> 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:



| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-----------------------|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 Vulnerability ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Invalid Flags | | | | | | | | Type | | | | | | | | | | | | | | | | | | | | | | | |
| | Type (cont.) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the Host Vulnerability data block.

Table 4-66 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 4-60.

The following diagram shows the format of an identity data block for 4.9+.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|-------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 |
| | Identity Data Block Type (94) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity Data Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity Data Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity Data Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|------------------|--------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| Identity UUID | Identity UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Identity UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Port | | | | | | | | | | | | | | | | Protocol | | | | | | | | | | | | | | | | |
| Server Map ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the Cisco identity data block.

Table 4-67 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: <ul style="list-style-type: none"> 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 <code>nmimport.pl</code> 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. |

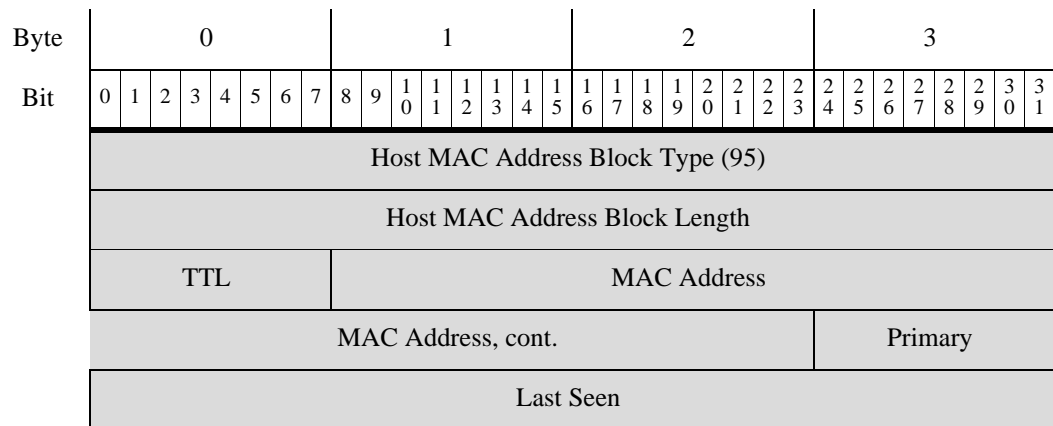
Table 4-67 Identity Data Block Fields (continued)

| Field | Data Type | Description |
|---------------|-----------|---|
| Protocol | uint16 | <p>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.</p> <p>Transport layer protocols are identified by the IANA protocol number. For example:</p> <ul style="list-style-type: none"> • 6 — TCP • 7 — UDP <p>Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example:</p> <ul style="list-style-type: none"> • 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.

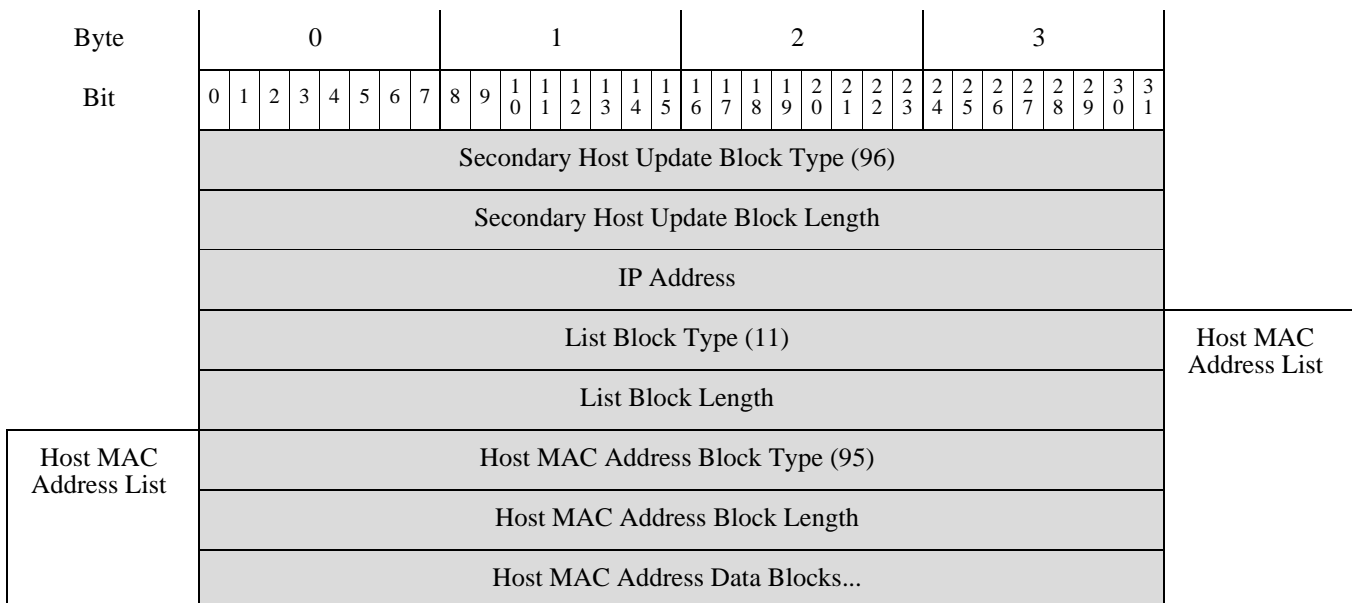
Table 4-68 Host MAC Address Data Block Fields

| 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.

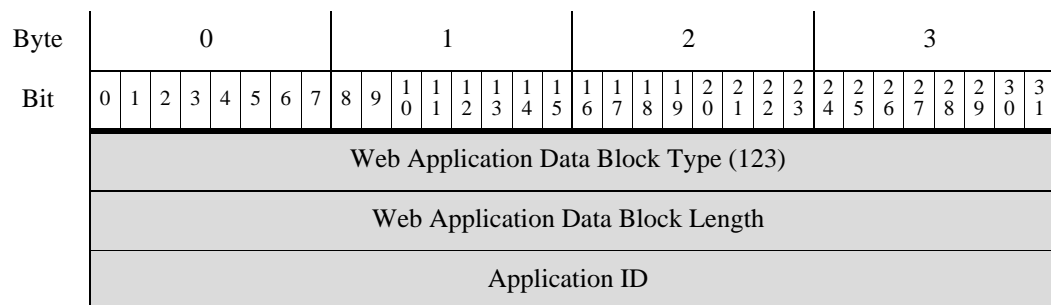
Table 4-69 Secondary Host Update Data Block Fields

| 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 4-70 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 6.2+

The connection statistics data block is used in connection data messages. A third Security Intelligence field has been added to Connection Statistics Data Block for 6.2+. The connection statistics data block for version 6.2+ has a block type of 168 in the series 1 group of blocks. It supersedes block type 163, [Connection Statistics Data Block 6.1.x, page B-204](#).

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 4-53](#).

The following diagram shows the format of a Connection Statistics data block for 6.2+:

| 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 Statistics Data Block Type (168) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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 |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Original Client IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Original Client IP Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Original Client IP Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Original Client IP Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Revision | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Revision, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Revision, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Revision, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Rule ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Tunnel Rule 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 |
| | Rule Action | | | | | | | | | | | | | | | | Rule Reason | | | | | | | | | | | | | | | |
| | Rule Reason, cont. | | | | | | | | | | | | | | | | Initiator Port | | | | | | | | | | | | | | | |
| | Responder Port | | | | | | | | | | | | | | | | TCP Flags | | | | | | | | | | | | | | | |
| | Protocol | | | | | | | | NetFlow Source | | | | | | | | | | | | | | | | | | | | | | | |
| | NetFlow Source, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Init. Tx Pkt, cont. | | | | | | | | Responder Transmitted Packets | | | | | | | | | | | | | | | | | | | | | | | |
| | Responder Transmitted Packets, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Resp. Tx Pkt, cont. | | | | | | | | Initiator Transmitted Bytes | | | | | | | | | | | | | | | | | | | | | | | |
| | Initiator Transmitted Bytes, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Init. Tx Bytes, cont. | | | | | | | | Responder Transmitted Packets | | | | | | | | | | | | | | | | | | | | | | | |
| | Responder Transmitted Bytes, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Resp. Tx. Bytes, cont. | | | | | | | | Initiator Packets Dropped | | | | | | | | | | | | | | | | | | | | | | | |
| | Initiator Packets Dropped, continued. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Init. Pkt. Drop, cont. | | | | | | | | Responder Packets Dropped | | | | | | | | | | | | | | | | | | | | | | | |
| | Responder Packets Dropped, continued. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Resp. Pkt. Drop, cont. | | | | | | | | Initiator Bytes Dropped | | | | | | | | | | | | | | | | | | | | | | | |

| 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 Bytes Dropped, continued. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Init. Byte Drop, cont. | | | | | | | | Responder Bytes Dropped | | | | | | | | | | | | | | | | | | | | | | | |
| | Rsp. Byte Drop, cont. | | | | | | | | Responder Bytes Dropped, continued. | | | | | | | | | | | | | | | | | | | | | | | |
| | QOS Intf., cont. | | | | | | | | QOS Applied Interface | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | QOS Applied Interface, continued | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | QOS Applied Interface, continued | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | QOS Applied Interface, continued | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | QOS Rule ID | | | | | | | | | | | | | | | | | | | | | | | |
| | QOS Rule ID, cont. | | | | | | | | User ID | | | | | | | | | | | | | | | | | | | | | | | |
| | User ID, cont. | | | | | | | | 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. | | | | | | | | Str. Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | |
| | Str. Block Type, cont. | | | | | | | | String Block Length | | | | | | | | | | | | | | | | | | | | | | | |
| | Str. Block Len., cont. | | | | | | | | 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|-------------------------------------|-----------------------|---|---|---|---|---|---|------------------|---|---|----|----|----|----|----|-------------------------------|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|----|
| | 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 |
| 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 | | | | | | | | | | | | | | | | Original Client Country | | | | | | | | | | | | | | | | |
| IOC Number | | | | | | | | | | | | | | | | Source Autonomous System | | | | | | | | | | | | | | | | |
| Source Autonomous System, continued | | | | | | | | | | | | | | | | Destination Autonomous System | | | | | | | | | | | | | | | | |
| Destination Autonomous System | | | | | | | | | | | | | | | | SNMP In | | | | | | | | | | | | | | | | |
| SNMP Out | | | | | | | | | | | | | | | | Source TOS | | | | | | | | Destination TOS | | | | | | | | |
| Source Mask | | | | | | | | Destination Mask | | | | | | | | Security Context | | | | | | | | | | | | | | | | |
| Security Context | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Security Context, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Security Context, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Security Context, continued | | | | | | | | | | | | | | | | VLAN ID | | | | | | | | | | | | | | | | |
| Referenced Host | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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 Actual Action, cont. | | | | | | | | SSL Expected Action | | | | | | | | | | | | | | | | SSL Flow Status | | | | | | | | |
| SSL Flow Status, cont. | | | | | | | | 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 |
| | Security Group ID, continued | | | | | | | | | | | | | | | | Location IPv6 | | | | | | | | | | | | | | | |
| | Location IPv6, continued | | | | | | | | | | | | | | | | Location IPv6, continued | | | | | | | | | | | | | | | |
| | Location IPv6, continued | | | | | | | | | | | | | | | | Location IPv6, continued | | | | | | | | | | | | | | | |
| | Location IPv6, continued | | | | | | | | | | | | | | | | Location IPv6, continued | | | | | | | | | | | | | | | |
| | Location IPv6, continued | | | | | | | | | | | | | | | | HTTP Response | | | | | | | | | | | | | | | |
| DNS Query | HTTP Response, continued | | | | | | | | | | | | | | | | String Block Type (0) | | | | | | | | | | | | | | | |
| | String Block Type (0), continued | | | | | | | | | | | | | | | | String Block Length | | | | | | | | | | | | | | | |
| | String Block Length, continued | | | | | | | | | | | | | | | | DNS Query... | | | | | | | | | | | | | | | |
| | DNS Record Type | | | | | | | | | | | | | | | | DNS Response Type | | | | | | | | | | | | | | | |
| | DNS TTL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sinkhole UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sinkhole UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sinkhole UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sinkhole UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence List 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence List 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Security Intelligence List 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the Connection Statistics data block for 6.2+.

Table 4-71 Connection Statistics Data Block 6.2+ Fields

| Field | Data Type | Description |
|---|-----------|--|
| Connection Statistics Data Block Type | uint32 | Initiates a Connection Statistics data block for 6.2+. The value is always 168. |
| 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 4-71 Connection Statistics Data Block 6.2+ 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. |
| Original Client IP Address | uint8[16] | IP address of the host behind the proxy that originated the request, 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. |
| Tunnel Rule ID | uint32 | Internal identifier for the tunnel 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 | uint32 | 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. |

Table 4-71 Connection Statistics Data Block 6.2+ Fields (continued)

| Field | Data Type | Description |
|-----------------------------|-----------|--|
| Initiator Transmitted Bytes | uint64 | Number of bytes transmitted by the initiating host. |
| Responder Transmitted Bytes | uint64 | Number of bytes transmitted by the responding host. |
| Initiator Packets Dropped | uint64 | Number of packets dropped from the session initiator due to rate limiting. |
| Responder Packets Dropped | uint64 | Number of packets dropped from the session responder due to rate limiting. |
| Initiator Bytes Dropped | uint64 | Number of bytes dropped from the session initiator due to rate limiting. |
| Responder Bytes Dropped | uint64 | Number of bytes dropped from the session responders due to rate limiting. |
| QOS Applied Interface | uint8[16] | For rate-limited connections, the name of the interface on which rate limiting is applied. |
| QOS Rule ID | uint32 | Internal ID number of the Quality of Service rule applied to the connection, if applicable. |
| 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. |

Table 4-71 Connection Statistics Data Block 6.2+ 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 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 block list. |
| Security Intelligence Layer | uint8 | The IP layer that matched the IP block list. |
| 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. |
| Original Client Country | uint 16 | Code for the country of the host behind the proxy which originated the request. |
| 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 4-71 Connection Statistics Data Block 6.2+ 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 in the 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 4-71 Connection Statistics Data Block 6.2+ Fields (continued)

| Field | Data Type | Description |
|-------------------------------|-----------|---|
| SSL Version | uint8 | The SSL or TLS protocol version used to encrypt the connection. |
| SSL Server Certificate Status | uint32 | The status of the SSL certificate. Possible values include: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 0 — 'Unknown' • 1 — 'Do Not Decrypt' • 2 — 'Block' • 3 — 'Block With Reset' • 4 — 'Decrypt (Known Key)' • 5 — 'Decrypt (Replace Key)' • 6 — 'Decrypt (Resign)' |

Table 4-71 Connection Statistics Data Block 6.2+ Fields (continued)

| Field | Data Type | Description |
|-----------------|-----------|--|
| SSL Flow Status | uint16 | <p>Status of the SSL Flow. These values describe the reason behind the action taken or the error message seen. Possible values include:</p> <ul style="list-style-type: none"> • 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 4-71 Connection Statistics Data Block 6.2+ Fields (continued)

| Field | Data Type | Description |
|-------------------|-----------|--|
| SSL Flow Messages | uint32 | <p>The messages exchanged between client and server during the SSL handshake. See http://tools.ietf.org/html/rfc5246 for more information.</p> <ul style="list-style-type: none"> • 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 | <p>The debugging level flags for an encrypted connection. Possible values include:</p> <ul style="list-style-type: none"> • 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 4-71 Connection Statistics Data Block 6.2+ 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. |
| DNS Response Type | uint16 | The numerical value for the type of DNS response. |
| 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 three Security Intelligence lists associated with the connection. |

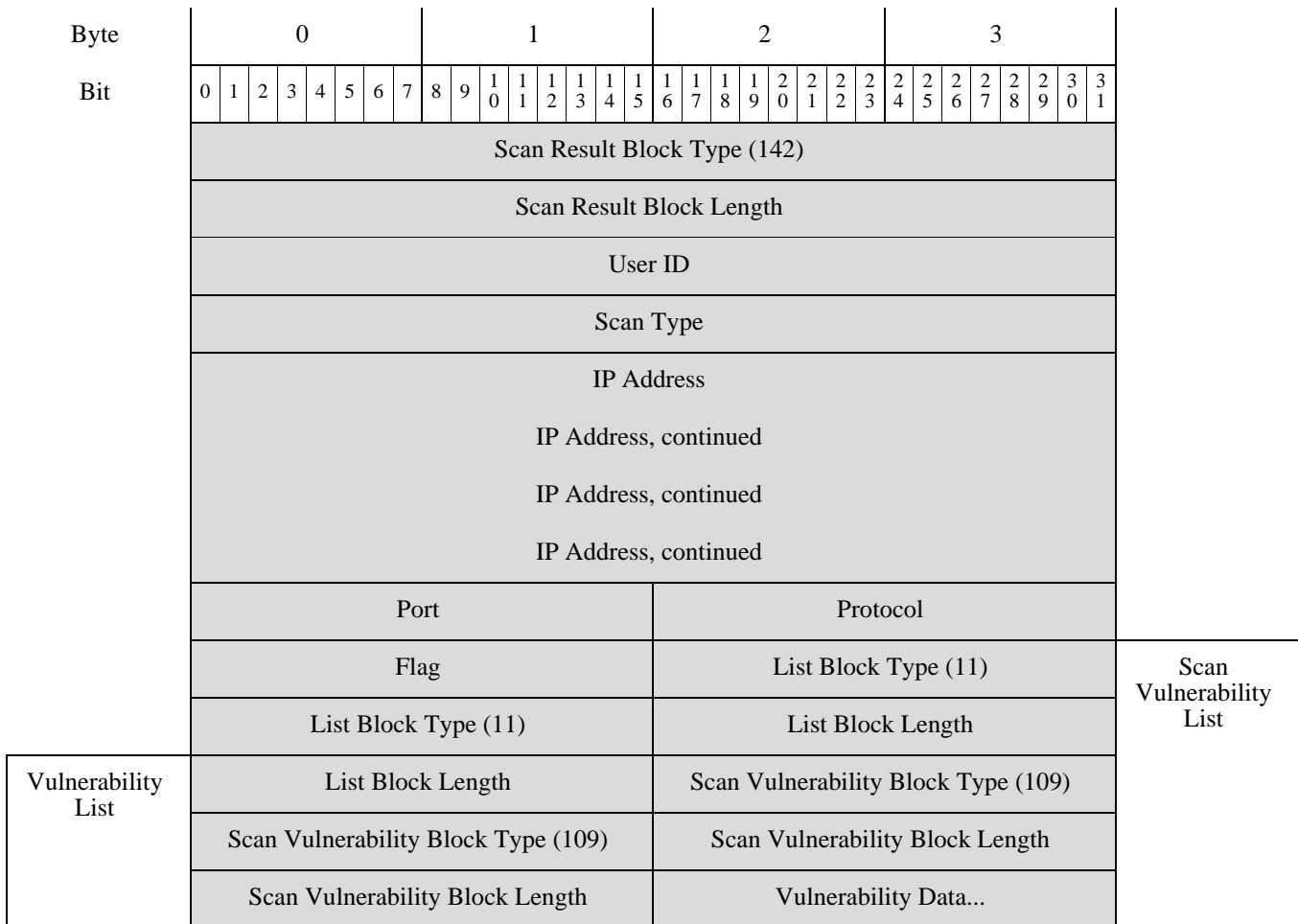
Table 4-71 Connection Statistics Data Block 6.2+ Fields (continued)

| Field | Data Type | Description |
|------------------------------|-----------|--|
| 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 three Security Intelligence lists associated with the connection. |
| Security Intelligence List 3 | uint32 | Security Intelligence List associated with the event. This maps to a Security Intelligence list in associated metadata. There may be three 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:



| 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) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Generic Scan Results List | |
| | List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scan Results List | Generic Scan Results Block Type (108) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic Scan Results Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic Scan Results... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| User Product List | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Product Data Blocks* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the Scan Result data block.

Table 4-72 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: <ul style="list-style-type: none"> 6 — TCP 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> 2048 — IP |
| Flag | uint16 | Reserved |

Table 4-72 Scan Result Data Block Fields (continued)

| Field | Data Type | Description |
|-----------------------------------|-----------|---|
| 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. |
| 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 4-168 for a description of this data block. |

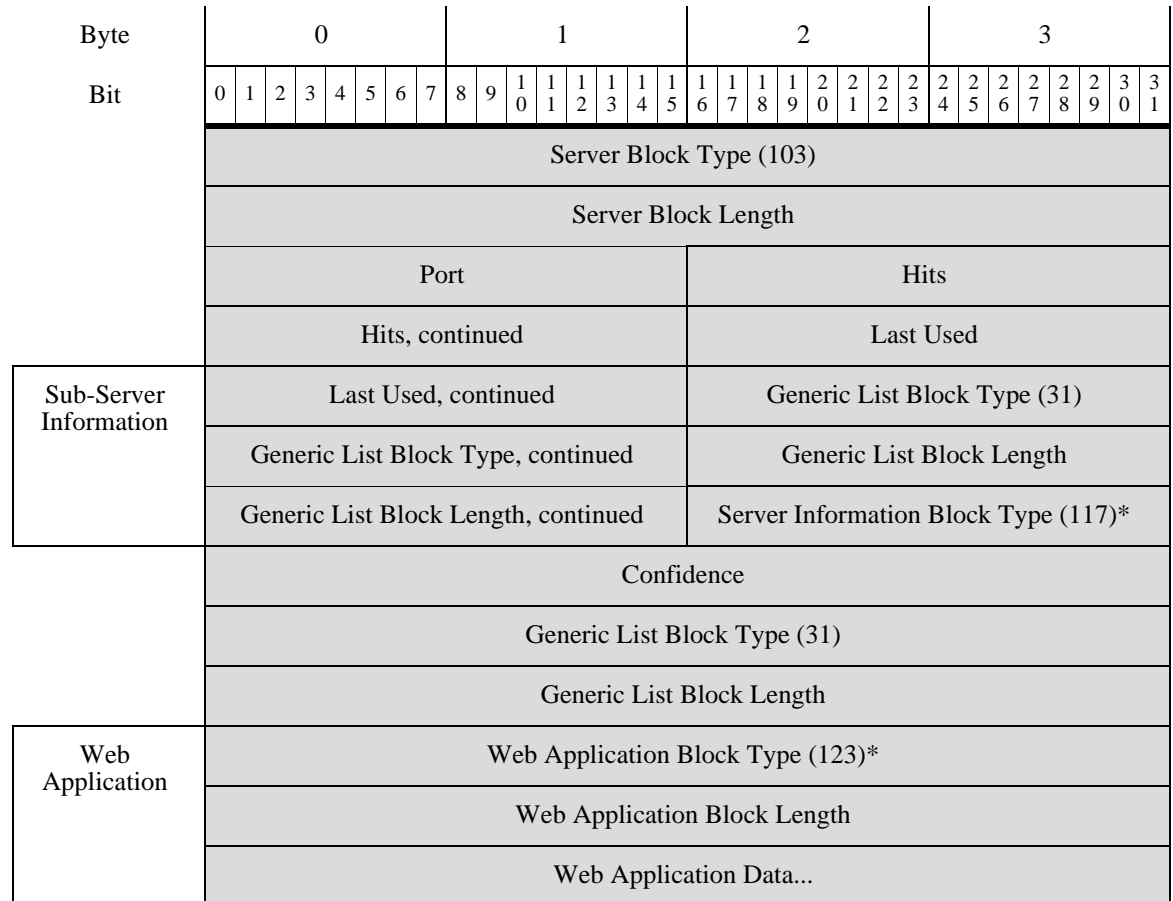
Host Server Data Block 4.10.0+

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 4-46. 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 4-73 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. |

Table 4-73 Host Server Data Block 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 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 4-142 . |
| Confidence | uint32 | Confidence percentage. |
| Generic List Block Type | uint32 | Initiates a Generic data block. This value is always 31. |
| 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 4-117 . |

Full Host Server Data Block 4.10.0+

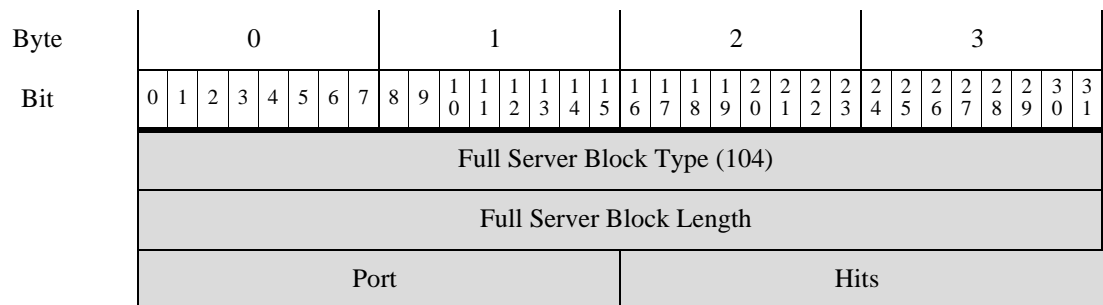
The Full Host Server data block conveys information about a server, including the server port, the frequency of use and most recent update, confidence of data accuracy, and Cisco and third-party vulnerabilities related to that server for the host. The Full Host Server data block contains a Full Sub-Server 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 | | | | | | | |
|------------------------------|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 - Cisco | Hits, continued | | | | | | | | | | | | | | | | Generic List Block Type (31) | | | | | | | | | | | | | | | |
| | Generic List Block Type, continued | | | | | | | | | | | | | | | | Generic List Block Length | | | | | | | | | | | | | | | |
| | Generic List Block Length, continued | | | | | | | | | | | | | | | | Full Server Information Data Blocks (106)* | | | | | | | | | | | | | | | |
| Sub-Servers - User | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Full Server Information Data Block Type (106)* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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)* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 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 | 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 4-74 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 4-144 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 4-144 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. |

Table 4-74 Full Server Data Block 4.10.0+ Fields (continued)

| Field | Data Type | Description |
|--|------------------|---|
| 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 4-144 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. |
| 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 4-144 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[<i>n</i>] | First <i>n</i> bytes of the packet involved in the server event, where <i>n</i> 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 4-112 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. |
| (ThirdParty/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 4-112 for a description of this data block. |

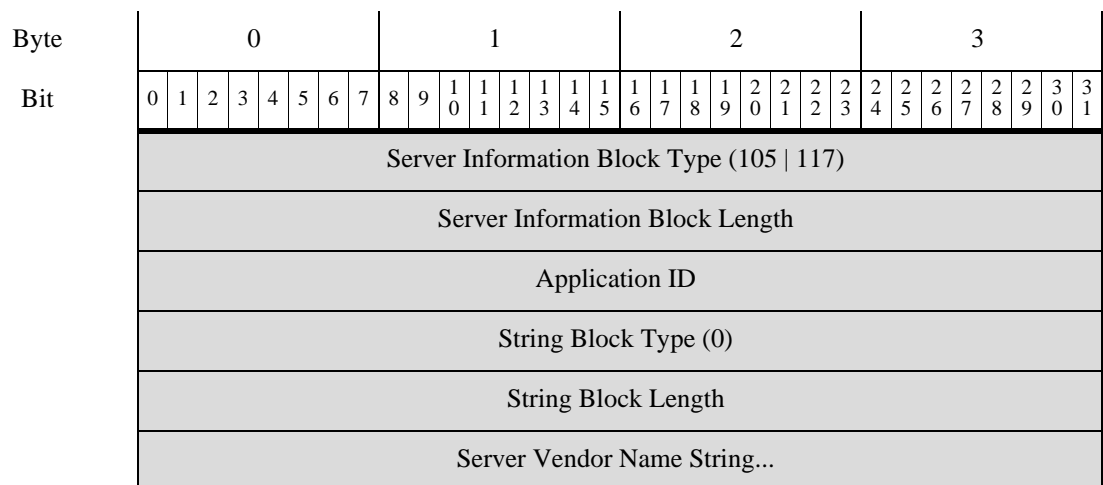
Table 4-74 Full Server Data Block 4.10.0+ Fields (continued)

| Field | Data Type | Description |
|---|-----------|---|
| 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 4-112 for a description of this data block. |
| 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 4-136](#) and [Full Host Server Data Block 4.10.0+, page 4-138](#).

The following diagram shows the format of the 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 |
| | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Server Version String... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Last Used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | List Block Type (11) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sub-Servers | Sub-Server Block Type (1) * | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sub-Server Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sub-Server Data... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the Server Information data block.

Table 4-75 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. |

Table 4-75 Server Information Data Block Fields (continued)

| Field | Data Type | Description |
|-------------------------|-----------|--|
| 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. |
| Source Type | uint32 | Number that maps to the type of data source: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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 4-74 . |

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 4-84](#)). 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 |
| | Full Server Block Type (106) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Full Server Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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 4-76 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. |

Table 4-76 Full Server Information Data Block Fields (continued)

| Field | Data Type | Description |
|-------------------------------|-----------|--|
| 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. |
| 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: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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 4-84 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 [Scan Result Data Block 5.2+](#), page 4-134. 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:

| 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 Scan Results Data Block Type (108) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic Scan Results Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Port | | | | | | | | | | | | | | | | Protocol | | | | | | | | | | | | | | | |
| Scan Result Sub-Servers | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Scan Result Sub-Server String... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scan Result Value | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Scan Result Value... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scan Result Sub-Server | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Scan Result Sub-Server (unformatted) String... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scan Result Value | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Scan Result Value... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the Generic Scan Results data block.

Table 4-77 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. |
| 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: <ul style="list-style-type: none"> • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> • 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. |

Table 4-77 Generic Scan Result Data Block Fields (continued)

| Field | Number of Bytes | Description |
|---------------------|-----------------|--|
| 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 [Scan Result Data Block 5.2+, page 4-134](#) and [Add Scan Result Messages, page 4-59](#). 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 | | | | | | | |
|-------------|-------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| Bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 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 |
| Description Clean | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Description Clean... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 4-78 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: <ul style="list-style-type: none"> 6 — TCP 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> 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. |

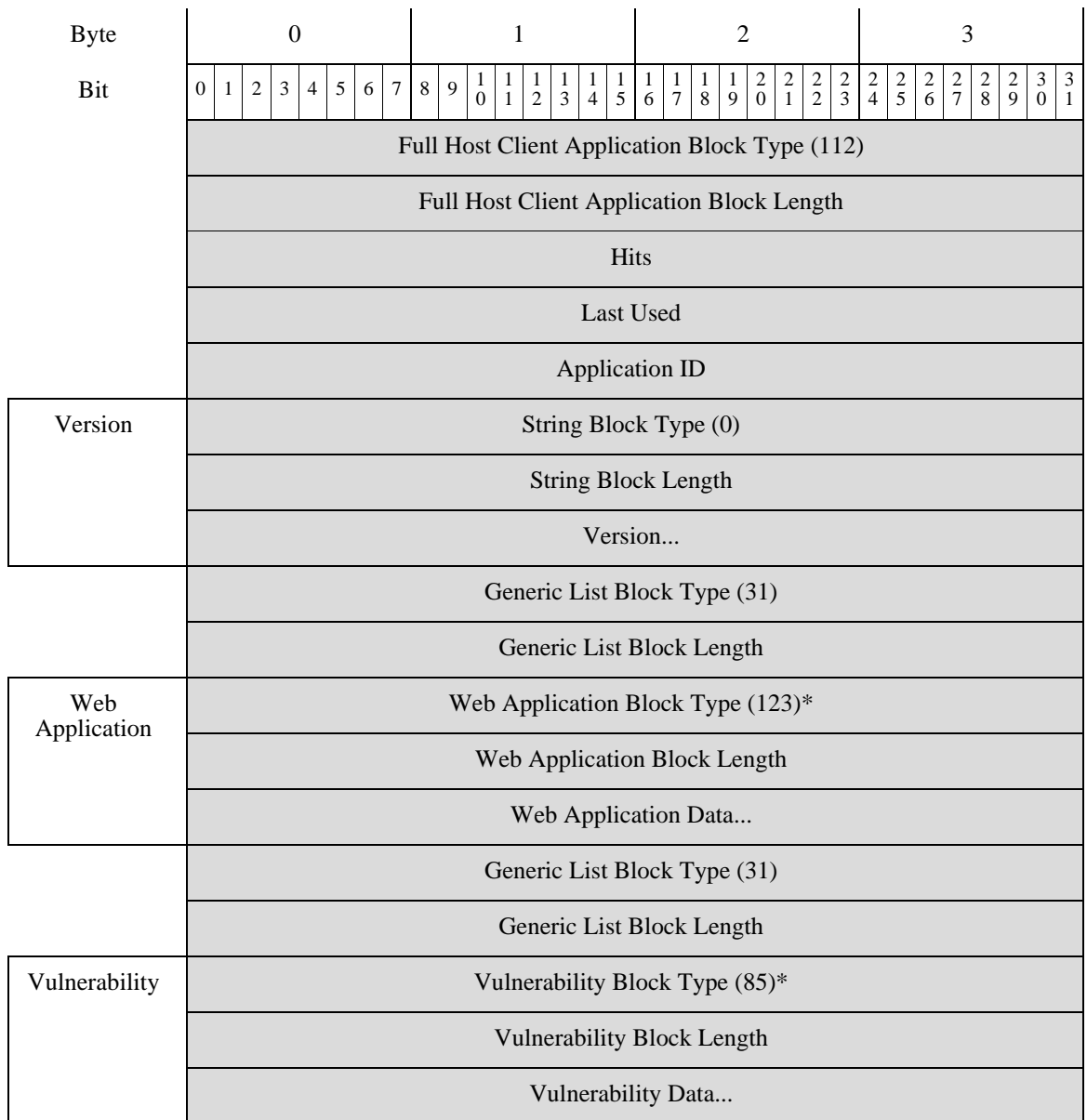
Table 4-78 Scan Vulnerability Data Block Fields (continued)

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |
| 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 4-76 . |
| 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 4-78 . |

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+:



The following table describes the fields of the Full Host Client Application data block.

Table 4-79 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. |
| 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+:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 Client Application Block Type (122) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Host Client Application Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Hits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Last Used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Application Protocol 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the Host Client Application data block.

Table 4-80 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. |
| Application Protocol ID | uint32 | The internal identification number for the application protocol, if applicable. |

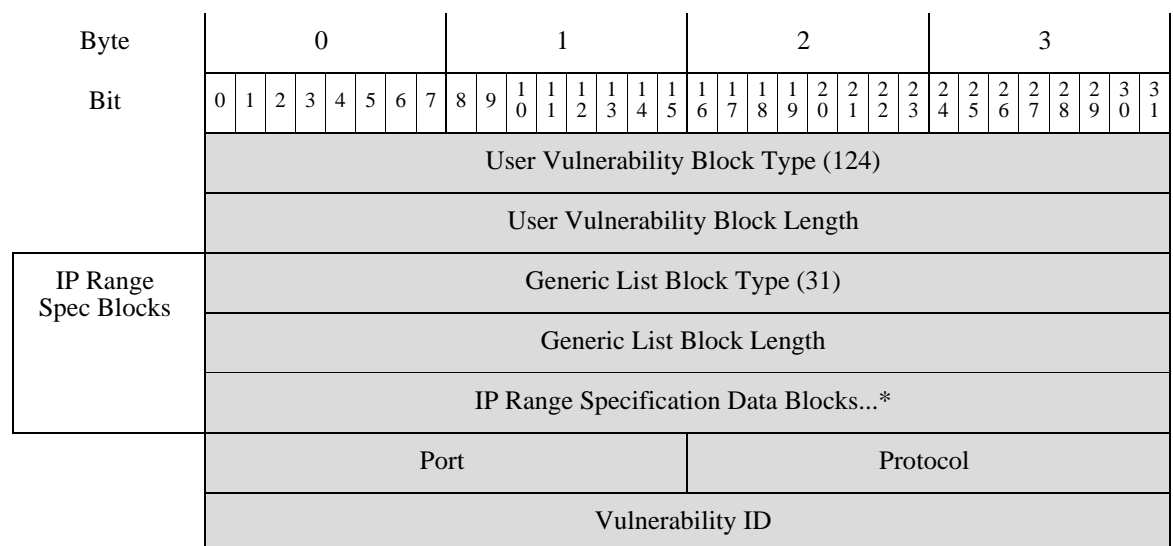
Table 4-80 Host Client Application Data Block Fields (continued)

| Field | Data Type | Description |
|-----------------------------|-----------|--|
| 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 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 4-117 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 4-106](#).

The following diagram shows the format of a User Vulnerability data block:



| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|------------------------|--------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 4-81 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 4-95 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 4-81 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: <ul style="list-style-type: none"> • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> • 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 | | | | | | | | 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 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* | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the operating system fingerprint data block.

Table 4-82 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 4-82 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 4-159 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.

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|--------------------|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| | Mobile Device Information Block Type (131) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Mobile Device Information Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mobile Device Data | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Mobile Device String Data... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Mobile Device Last Seen | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Mobile | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Jailbroken | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The describes the fields of the Mobile Device Information data block returned by 5.1+.

Table 4-83 Mobile Device Information Data Block 5.1+ Fields

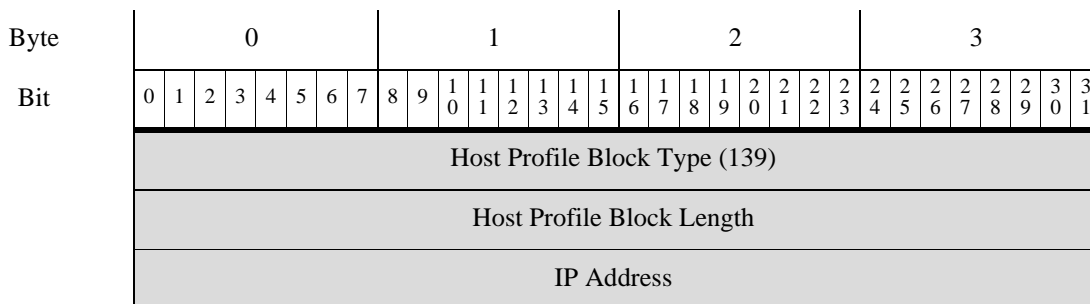
| Field | Data Type | Description |
|--|-----------|---|
| Mobile Device Information Block Type (131) | uint32 | Initiates the operating system data block. This value is always 131. |
| Mobile Device Information Block Length | uint32 | Number of bytes in the Mobile Device Information data block, including 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* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Host Discovery and Connection Data Blocks

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | | |
|---------------------------|-------------------------------------|---|---|---|---|---|---|------------|---|---|----|----|----|----|----|---------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|-----------------------------|
| | 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 | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | | |
|-----------------|-------------------------------------|---|---|---|---|---|---|---|-----------|---|----|----|----|----|----|----|--------------------------------|----|----|----|----|----|----|----|------------------------------|----|----|----|----|----|----|----|-----------------------------|
| | 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 Data | VLAN ID, cont. | | | | | | | | VLAN Type | | | | | | | | VLAN Priority | | | | | | | | Generic List Block Type (31) | | | | | | | | List of Client Applications |
| | Generic List Block Type (31), cont. | | | | | | | | | | | | | | | | Generic List Block Length | | | | | | | | | | | | | | | | |
| | Generic List Block Length, cont. | | | | | | | | | | | | | | | | Client Application Data Blocks | | | | | | | | | | | | | | | | |
| NetBIOS Name | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | NetBIOS String Data... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the host profile data block returned by 5.2+.

Table 4-84 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: <ul style="list-style-type: none"> 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 4-157 for a description of this data block. |

Table 4-84 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 4-157 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 4-157 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 4-157 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 4-84 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 4-157 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 4-157 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 4-157 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 4-157 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 4-84 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 4-157 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 4-136 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 4-136 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 4-75 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 4-75 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 4-84 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 4-115 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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 4-152 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 [Scan Result Data Block 5.2+](#), [page 4-134](#) and [User Server and Operating System Messages, page 4-57](#). 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 |
| | User Product Data Block Type (134) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User Product Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Source Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IP Address Ranges | Generic List Block Type (31) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Generic List Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IP Range Specification Data Blocks* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|-----------------------|----------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 | | | | | | | |
|------------------|------------------------------------|---|---|---|---|---|---|---|------------|---|----|----|----|----|----|----|------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 4-85 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: <ul style="list-style-type: none"> • 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 <code>nmimport.pl</code> 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 4-95 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: <ul style="list-style-type: none"> • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: <ul style="list-style-type: none"> • 2048 — IP |
| Drop User Product | uint32 | Indicates whether the user OS definition was deleted from the host: <ul style="list-style-type: none"> • 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 4-85 *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 4-85 *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 4-85 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 4-102 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 4-62](#).

**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 4-86 *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 for 5.0 - 5.0.2, page B-100 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 4-176 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 5.x, page B-114 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 5.x, page B-114 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 4-34 for more information. |
| 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 4-185 for more information. Supersedes block type 120. |

Table 4-86 User Data Block Type (continued)

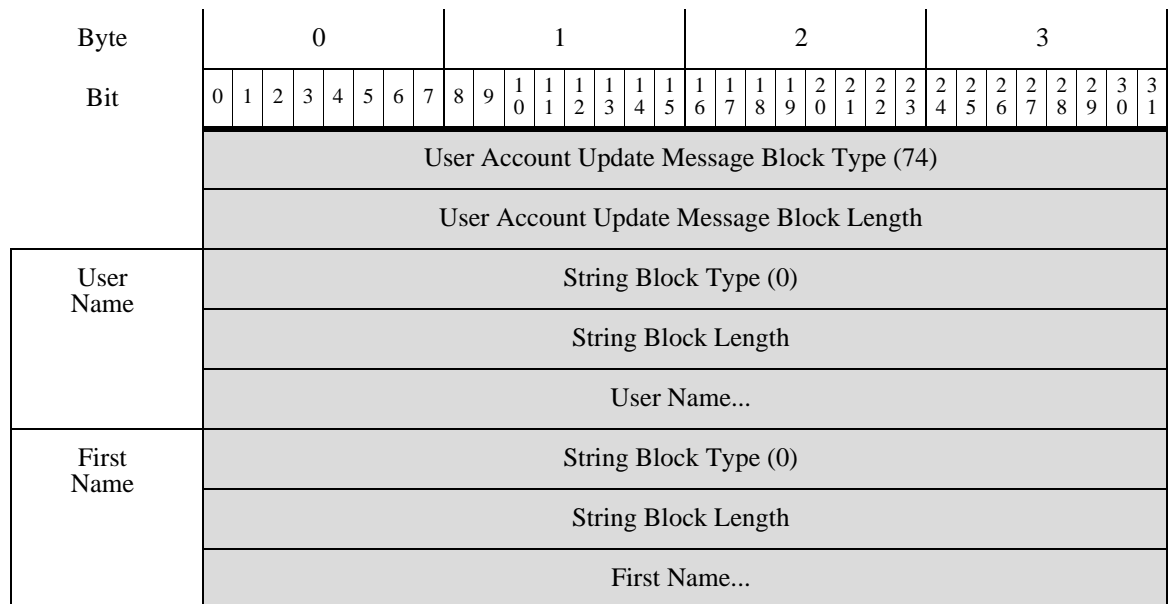
| Type | Content | Data Block Category | Description |
|------|-------------------------|---------------------|---|
| 159 | User Login Information | Legacy | Contains changes in login information for users detected by the system. See User Login Information Data Block 6.0.x, page B-104 for more information. It supersedes block type 127. |
| 165 | User Login Information | Legacy | Contains changes in login information for users detected by the system. See User Login Information Data Block 6.1.x, page B-110 for more information. It supersedes block type 159. It is superseded by block type 167. |
| 166 | VPN Session Information | Current | Contains information on VPN sessions detected by the system. See VPN Session Data Block for 6.2+, page 4-188 for more information. |
| 167 | User Login Information | Current | Contains changes in login information for users detected by the system. See User Login Information Data Block 6.2+, page 4-190 for more information. It supersedes block type 165. |

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 | | | | | | | |
|-----------------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| Middle Initials | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Middle Initials... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|--------------------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| Country/ Region | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Country/Region... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|----------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 |
| Office | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Office... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|---------------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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 4 | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | User 4... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 4-87 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. |

Table 4-87 *User Account Update Message 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 middle initials for the user. This value is always 0. |
| 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. |

Table 4-87 *User Account Update Message Data Block Fields (continued)*

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |
| 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. |

Table 4-87 *User Account Update Message Data Block Fields (continued)*

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |
| 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. |

Table 4-87 *User Account Update Message Data Block Fields (continued)*

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |
| 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. |

Table 4-87 User Account Update Message Data Block Fields (continued)

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |

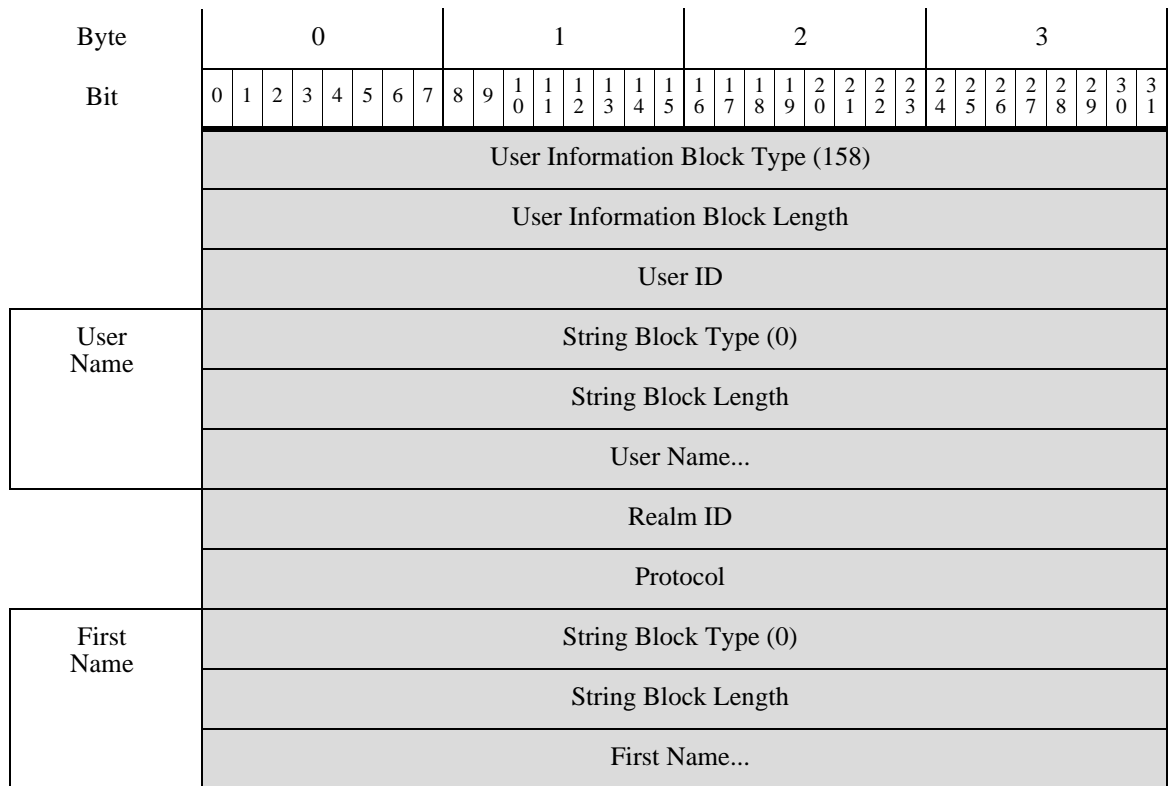
User Information Data Block for 6.0+

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 4-61](#)

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-114](#) for more information.

The following diagram shows the format of the User Information data block.



| Byte | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|----------------------------------|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Endpoint Profile ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Security Group ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Location IPv6 Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Location IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Location IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Location IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the User Information data block.

Table 4-88 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. |

Table 4-88 *User Information Data Block Fields (continued)*

| Field | Data Type | Description |
|---------------------|-----------|--|
| 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. |
| 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. |

Table 4-88 User Information Data Block Fields (continued)

| Field | Data Type | Description |
|-----------------------|-----------|--|
| 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. |

VPN Session Data Block for 6.2+

The VPN Session data block for 6.2+ has a block type of 166 in the series 1 group of blocks. The data block describes VPN Session information.

The following diagram shows the format of a VPN Session data block in 6.2+.

| 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 |
| | VPN Session Data Block Type (166) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | VPN Session Data Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Index | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Group Policy | Type | | | | | | | | | | | | | | | | String Block Type (0) | | | | | | | | | | | | | | | |
| | Str. Blk Type | | | | | | | | | | | | | | | | String Block Length | | | | | | | | | | | | | | | |
| | Str. Blk Length | | | | | | | | | | | | | | | | Group Policy... | | | | | | | | | | | | | | | |
| Connection Profile | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Connection Profile... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Client IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Client IP Address (continued) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Client IP Address (continued) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Client IP Address (continued) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Client Operating System | Client Country | | | | | | | | | | | | | | | | String Block Type (0) | | | | | | | | | | | | | | | |
| | String Block Type (0) (continued) | | | | | | | | | | | | | | | | String Block Length | | | | | | | | | | | | | | | |
| | String Block Length (continued) | | | | | | | | | | | | | | | | Client Operating System... | | | | | | | | | | | | | | | |

| 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 Application | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Client Application... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Connection Duration | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Bytes Transmitted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Bytes Transmitted (continued) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Bytes Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Bytes Received (continued) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the fields of the VPN Session data block.

Table 4-89 VPN Session Data Block Fields

| Field | Data Type | Description |
|-----------------------------|-----------|--|
| VPN Session Data Block Type | uint32 | Initiates the VPN Session data block. This value is always 166. |
| VPN Session Block Length | uint32 | Number of bytes in the VPN Session data block, including eight bytes for the VPN Session data block type and length, plus the number of bytes in the VPN Session data fields that follow. |
| Index | uint32 | A number generated by the VPN device to identify the session. |
| Type | uint8 | Type of VPN session. Possibly values are: <ul style="list-style-type: none"> • 0 - Unknown • 1- Cisco IKEv1 Client • 2- AnyConnect IKEv1 Client • 3 - AnyConnect SSL • 4 - WebVPN Clientless • 5 - Site to Site IKEv2 • 6 - Site to Site IKEv2 • 7 - Generic IKEv2 RA Client |
| String Block Type | uint32 | Initiates a String data block containing the Group Policy for the VPN Session. 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 group policy. |
| Group Policy | string | The name of the group policy assigned to the client when the VPN session is established. |

Table 4-89 VPN Session Data Block Fields (continued)

| Field | Data Type | Description |
|-------------------------|-----------|---|
| String Block Type | uint32 | Initiates a String data block containing the Connection Profile for the VPN session. 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 connection profile. |
| Connection Profile | string | The name of the connection profile (tunnel group) used by the VPN session. |
| Client IP Address | uint8[16] | IP address of the VPN client device. |
| Client Country | uint16 | Code for the country of the VPN client. |
| String Block Type | uint32 | Initiates a String data block containing the operating system used by the client device. 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 operating system name. |
| Client Operating System | string | The operating system for the client device. |
| String Block Type | uint32 | Initiates a String data block containing the VPN application used by the client device. 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 VPN application. |
| Client Application | string | The VPN application for the client device. |
| Connection Duration | uint32 | Duration of the VPN session in seconds. Only specified for VPN logout actions, otherwise the value is 0. |
| Bytes Transmitted | uint64 | Number of bytes transmitted to the VPN client during the VPN session. Only specified for VPN logout actions, otherwise the value is 0. |
| Bytes Received | uint64 | Number of bytes received from the VPN client during the VPN session. Only specified for VPN logout actions, otherwise the value is 0. |

User Login Information Data Block 6.2+

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 4-62](#).

The User Login Information data block has a block type of 167 in the series 1 group of blocks for version 6.2+. It has new fields for VPN support. It supersedes block type 165. See [User Login Information Data Block 6.1.x, page B-107](#) 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 (167) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Protocol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Port | | | | | | | | | | | | | | | | Range Start | | | | | | | | | | | | | | | |
| | Start Port | | | | | | | | | | | | | | | | End Port | | | | | | | | | | | | | | | |
| Email | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Email... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | IPv6 Address, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Location IPv6 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 |
| | 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... | | | | | | | | | | | | | | | |
| Description | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VPN Session | VPN Session Data Block Type (166) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | VPN Session Data Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | VPN Session... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The following table describes the components of the User Login Information data block.

Table 4-90 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 167 for version 6.2+. |
| 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. |
| 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 1-4 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. |

Table 4-90 User Login Information Data Block Fields (continued)

| Field | Data Type | Description |
|-----------------------|-----------|---|
| 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. |
| 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. |
| Protocol | uint32 | Protocol used to detect or report the user. Possible values are: <ul style="list-style-type: none"> • 165 - FTP • 426 - SIP • 547 - AOL Instant Messenger • 683 - IMAP • 710 - LDAP • 767 - NTP • 773 - Oracle Database • 788 - POP3 • 1755 - MDNS |
| Port | uint16 | The port number on which the user was detected. |
| Range Start | uint16 | The start port in the port range used by the TS Agent. |
| Start Port | uint16 | The start port in the range the TS Agent assigned to the individual user. |
| End Port | uint16 | The end port in the range the TS Agent assigned to the individual 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. |
| 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. |

Table 4-90 User Login Information Data Block Fields (continued)

| Field | Data Type | Description |
|-------------------------------|------------------|--|
| Authentication Type | uint8 | Type of authentication used by the user. Values may be: <ul style="list-style-type: none"> • 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. |
| 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 reporter of this activity, such as the name of the Active Directory server. |
| String Block Type | uint32 | Initiates a String data block containing the Description value. This value is always 0. |
| String Block Length | uint32 | Number of bytes in the Description String data block, including eight bytes for the block type and length fields, plus the number of bytes in the Description field. |
| Description | string | The Description of the login or logoff activity. |
| VPN Session Block Type | uint32 | Initiates a VPN Session data block containing the VPN session data. This value is always 166. |
| VPN Session Data Block Length | uint32 | Number of bytes in the VPN Session data block, including eight bytes for the block type and length fields, plus the number of bytes in the VPN Session data block. |
| VPN Session data | VPN Session Data | Information regarding the detected VPN session, if the login was associated with a VPN session. This is only used when there is a VPN session. |

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).

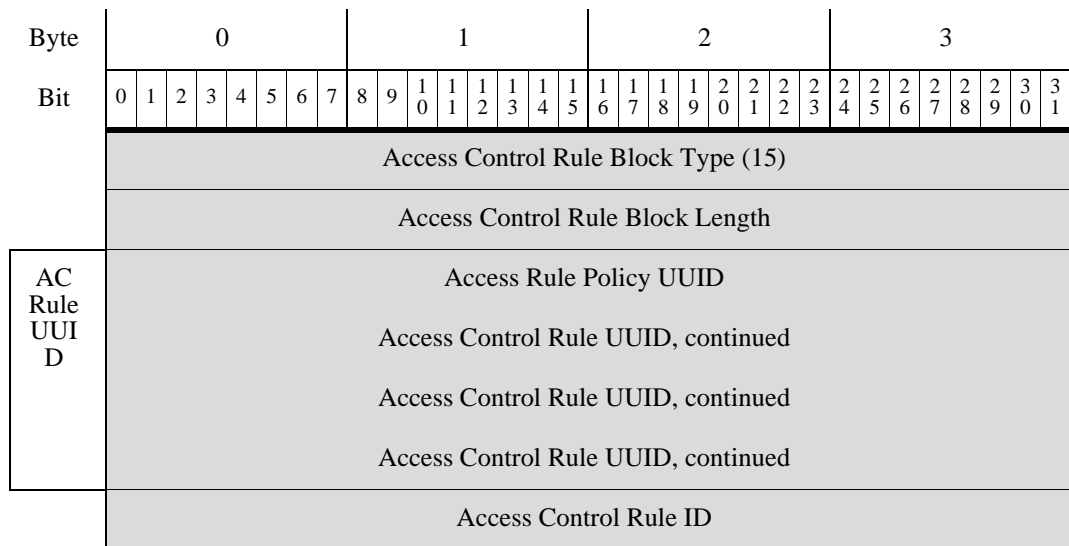
Table 4-91 Discovery and Connection Event Series 2 Block Types

| 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 4-195. |
| 21 | Access Control Rule Reason | Legacy | Used by access control rule metadata messages to map access control rule reasons to a descriptive string. See Access Control Policy Rule Reason Data Block , page B-260. |
| 22 | Security Intelligence Category | Current | Used to store Security Intelligence information. See Security Intelligence Category Data Block 5.1+ , page 4-198. |
| 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 4-200. |
| 59 | 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 6.0+ , page 4-196. |

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:



| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-92 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. This field, along with Access Control Rule ID, makes up the unique key for this record. |
| Access Control Rule ID | uint32 | The internal Cisco identifier for the access control rule. This field, along with Access Control Rule UUID, makes up the unique key for this record. |
| 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 6.0+

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 59 in the series 2 group of blocks. It supersedes block type 21.

The following graphic shows the structure of the Access Control Rule Reason 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 |
| | Access Control Rule Reason Block Type (59) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Rule Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Description | Access Control Rule Reason | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Description... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-93 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 59. |
| 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. |

Table 4-93 Access Control Rule Reason Data Block Fields (continued)

| Field | Data Type | Description |
|----------------------------|-----------|--|
| Access Control Rule Reason | uint32 | <p>The reason the Access Control rule logged the connection. This field is the unique key for this record. The number of the reason for the rule that triggered the event.</p> <p>Rule reasons are a binary bitmap in which multiple bits may be set. There may be several reasons for a rule. The bit values are as follows:</p> <ul style="list-style-type: none"> • 1 — IP Block • 2 — IP Monitor • 4 — User Bypass • 8 — File Monitor • 16 — File Block • 32 — Intrusion Monitor • 64 — Intrusion Block • 128 — File Resume Block • 256 — File Resume Allow"] • 512 — File Custom Detection • 1024 — SSL Block • 2048 — DNS Block • 4096 — DNS Monitor • 8192 — URL Block • 16384 — URL Monitor • 32768 — Content Restriction • 65536 — Intelligent App Bypass • 131072 — WSA Threat |
| 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 4-94 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 block list or allow list triggered by the connection. This field, along with Access Control Policy UUID, makes up the unique key for this record. |
| Access Control Policy UUID | uint8[16] | The UUID of the access control policy configured for Security Intelligence. This field, along with Security Intelligence List ID, makes up the unique key for this record. |
| String Block Type | uint32 | Initiates a String data block containing the descriptive name associated with the Security Intelligence List. This value is always 0. |

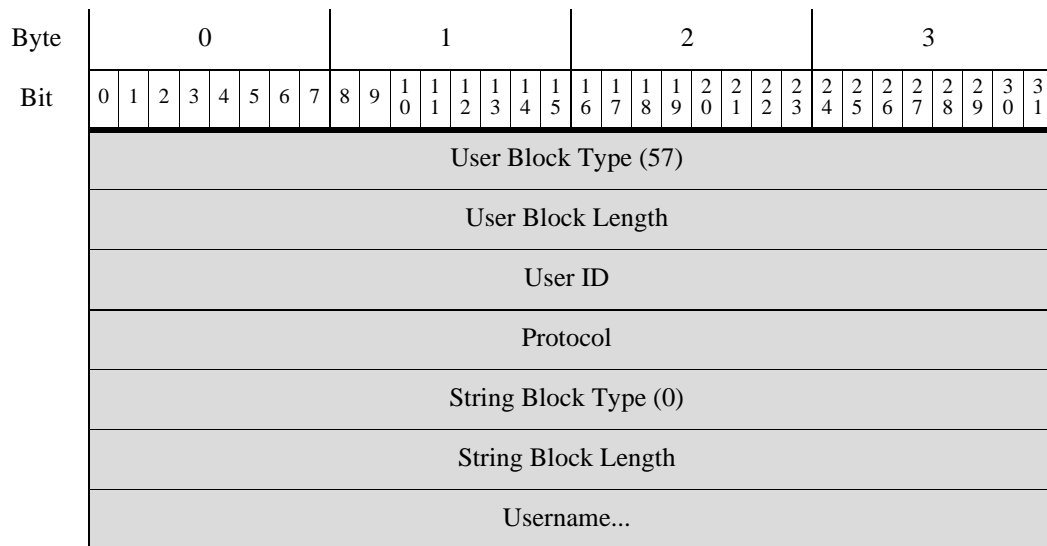
Table 4-94 Security Intelligence Category Data Block fields (continued)

| 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 Security Intelligence List Name field. |
| Security Intelligence List Name | string | The name of the Security Intelligence category IP block list or allow list 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.

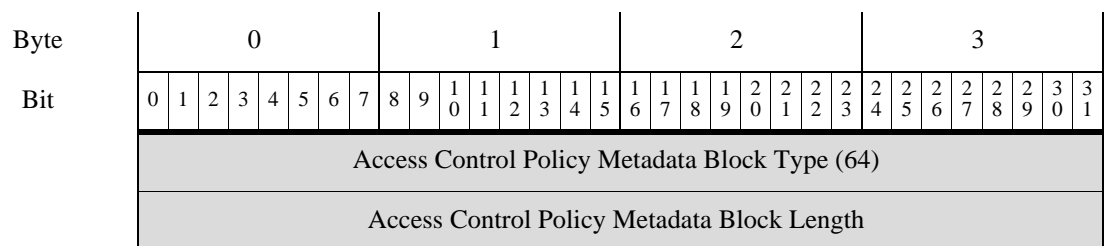
Table 4-95 User Data Block Fields

| 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. This field is the unique key for this record. |
| Protocol | uint32 | Protocol used to detect or report the user. Possible values are: <ul style="list-style-type: none"> • 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 |

Access Control Policy Metadata Block 6.0+

The eStreamer service uses the Access Control Policy Metadata data block in Access Control Policy metadata messages to provide Access Control Policy information. The Access Control Policy metadata block has a block type of 64 in the series 2 group of blocks.

The following graphic shows the structure of the Access Control Policy 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 |
| AC Policy UUID | Access Control Policy UUID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Access Control Policy UUID, continued | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Sensor ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Name | String Block Type (0) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | String Block Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Policy Name... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

Table 4-96 Access Control Policy Metadata Block Fields

| Field | Data Type | Description |
|---|-----------|--|
| Access Control Policy Metadata Block Type | uint32 | Initiates an Access Control Policy metadata block. This value is always 64. |
| Access Control Policy Metadata Block Length | uint32 | Total number of bytes in the Access Control Policy Metadata block, including eight bytes for the Access Control Policy Metadata 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. This field is the unique key for this record. |
| 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 descriptive name associated with 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. |