Introduction

The Cisco Event Streamer (also known as eStreamer) allows you to stream FireSIGHT System intrusion, discovery, and connection data from the Cisco Defense Center or managed device (also referred to as the eStreamer server) to external client applications.

Note that eStreamer is not supported on virtual devices. To stream events from a virtual device, you can configure eStreamer on the Defense Center that the device reports to.

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

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

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

This guide provides the information you need to successfully create and run an eStreamer Version 5.3.1 client application.

Major Changes in eStreamer Version 5.3.1

If you are upgrading your FireSIGHT System deployment to Version 5.3.1, please note the following changes, some of which may require you to update your eStreamer client:

- Fixed errors in the following blocks:
  - Fixed Correlation Event for 5.1+, page 3-36 with correct information for the handling of IPv4 addresses.
  - Fixed Discovery Event Header 5.2+, page 4-32 with correct information for the handling of IPv4 addresses.
  - Fixed User Login Information Data Block 5.1+, page 4-171 with correct information for the handling of IPv4 addresses.

- Replaced the following blocks:
  - Replaced Intrusion Impact Alert Data, page B-29 with Intrusion Impact Alert Data 5.3+, page 3-12, which has an IPv6 field.
  - Replaced Malware Event Data Block 5.3, page B-48 with Malware Event Data Block 5.3.1+, page 3-64, which has a security context field.
  - Replaced File Event for 5.3, page B-116 with File Event for 5.3.1+, page 3-57, which has a security context field.
  - Replaced Intrusion Event Record 5.3, page B-17 with Intrusion Event Record 5.3.1+, page 3-6, which has a security context field.
  - Replaced Connection Statistics Data Block 5.3, page B-102 with Connection Statistics Data Block 5.3.1+, page 4-111, which has a security context field.

Using this Guide

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

- Intrusion event data and event extra data
- Correlation (compliance) event data
- Discovery event data
- User event data
- Metadata for events
- Host information
- Malware event data

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

- Understanding the eStreamer Application Protocol, page 2-1, which provides an overview of eStreamer communications, details some of the requirements for writing eStreamer client applications, and describes the four types of messages used to send commands to and receive data from the eStreamer service.
- Understanding Intrusion and Correlation Data Structures, page 3-1, which documents the data formats used to return event data generated by the intrusion detection and correlation components and the data formats used to represent the intrusion and correlation events.
- Understanding Discovery & Connection Data Structures, page 4-1, which documents the data formats used to return discovery, user, and connection event data.
- Understanding Host Data Structures, page 5-1, which documents the data formats that eStreamer uses to return full host information data when it receives a host information request message.
- Configuring eStreamer, page 6-1, which documents how to configure the eStreamer on a Defense Center or managed device. The chapter also documents the eStreamer command-line switches and provides instructions for manually starting and stopping the eStreamer service and for configuring the Defense Center or managed device to start eStreamer automatically.
- Data Structure Examples, page A-1, which provides examples of eStreamer message packets in binary format.

- Understanding Legacy Data Structures, page B-1, which documents the structure of legacy data structures that are no longer in use by the currently shipping product but may be used by older clients.

Prerequisites

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

Product Versions for FireSIGHT System Releases

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

Table 1-1 FireSIGHT System Product Versions

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

Table 1-1 FireSIGHT System Product Versions (continued)

Release	Defense Center Version	Master Defense Center Version	Intrusion Sensor Version	Sensor Version	Managed Device Version
3D System 4.10.1	Defense Center 4.10.1	Master Defense Center 4.10.1	N/A	N/A	4.10.1
3D System 4.10.2	Defense Center 4.10.2	Master Defense Center 4.10.2	N/A	N/A	4.10.2
3D System 4.10.3	Defense Center 4.10.3	Master Defense Center 4.10.3	N/A	N/A	4.10.3
3D System 5.0	Defense Center 5.0	N/A	N/A	N/A	5.0
3D System 5.1	Defense Center 5.1	N/A	N/A	N/A	5.1
3D System 5.1.1	Defense Center 5.1.1	N/A	N/A	N/A	5.1.1
3D System 5.2	Defense Center 5.2	N/A	N/A	N/A	5.2
3D System 5.3	Defense Center 5.3	N/A	N/A	N/A	5.3
3D System 5.3.1	Defense Center 5.3.1	N/A	N/A	N/A	5.3.1

Document Conventions

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

Table 1-2 eStreamer Message Data Type Conventions

Data Type	Description
nn-bit field	Bit field of nn bits
byte	8-bit byte containing data of arbitrary format
int8	Signed 8-bit byte
uint8	Unsigned 8-bit byte
int16	Signed 16-bit integer
uint16	Unsigned 16-bit integer
int32	Signed 32-bit integer
uint32	Unsigned 32-bit integer
uint64	Unsigned 64-bit integer
string	Variable length field containing character data
[n]	Array subscript following any of the above data types to indicate n instances of the indicated data type, for example, uint8[4]
variable	Collection of various data types
BLOB	Binary object of unspecified type, typically raw data as captured from a packet

IP Addresses

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