Understand eStreamer and Troubleshoot eNcore Integration

Contents

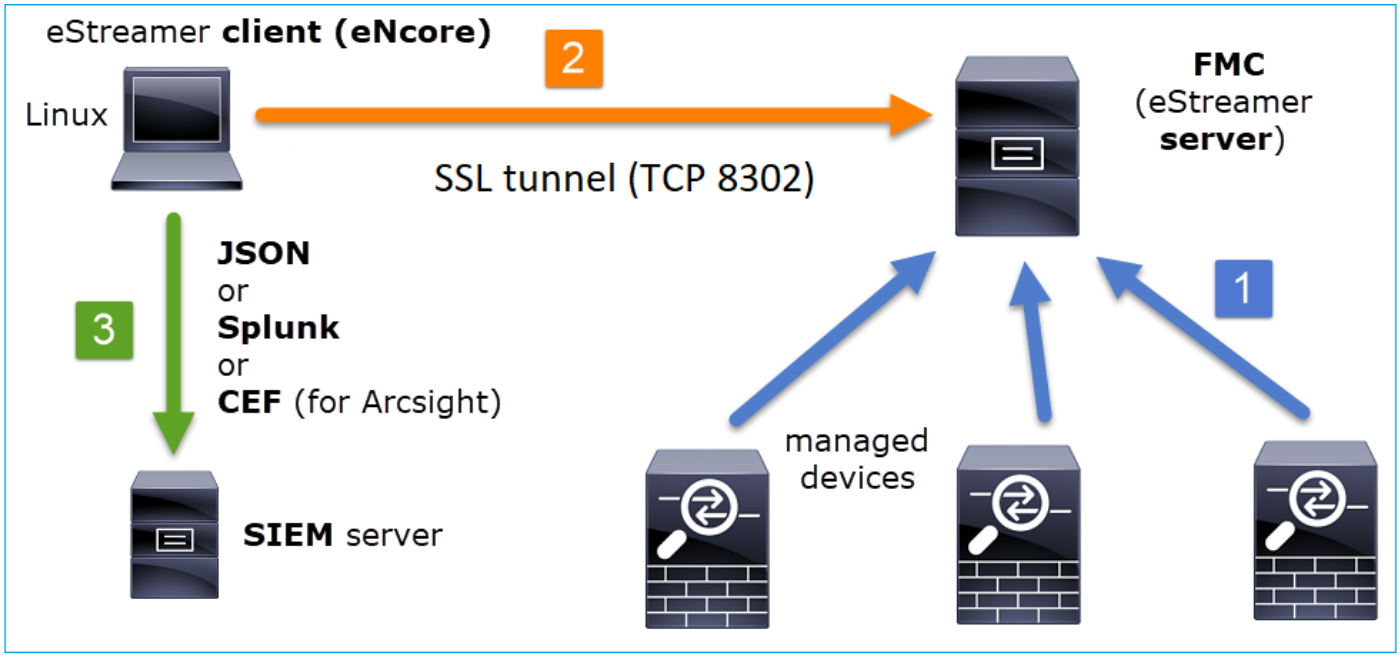
Introduction

This document describes the Cisco Event Streamer (also known as eStreamer) eNcore CLI client. Specifically, it describes the operation and provides troubleshooting information. Additionally, it covers common issues seen by the Cisco Technical Assistance Center (TAC) along with Frequently Asked Questions (FAQ).

Contributed by David Torres Rivas, Mikis Zafeiroudis, Cisco TAC Engineers.

Overview

eNcore is an all-purpose client, which requests all possible events from the eStreamer server (FMC), parses the binary content, and outputs events in various formats to support other Security Information and Event Management tools (SIEMs).



eStreamer Connection Establishment

The client (eNcore) initiates a connection to FMC TCP port 8302 where SSL handshake is performed:

```
1: 11:34:02.901091 192.168.27.100.46538 > 10.48.26.49.8302: S 1607291631:1607291631(0) win 29200
<mss 1460,sackOK,timestamp 2350959 0,nop,wscale 10>
2: 11:34:02.902220 10.48.26.49.8302 > 192.168.27.100.46538: S 2529774236:2529774236(0) ack
1607291632 win 28960 <mss 1380,sackOK,timestamp 940036669 2350959,nop,wscale 7>
3: 11:34:02.902739 192.168.27.100.46538 > 10.48.26.49.8302: . ack 2529774237 win 29
<nop,nop,timestamp 2350959 940036669>
```

The FMC accepts the connection, performs SSL handshake on the same port and verifies the client Common Name (CN):

```
Mar  2 11:45:06 FMC SF-IMS[22601]: [22601] Event Streamer:ConnectionHandler [INFO] Accepted IPv4
connection from 10.48.26.47:46538/tcp
Mar  2 11:45:06 FMC SF-IMS[22601]: [22601] Event Streamer:ConnectionHandler [INFO] Added
10.48.26.47 to host table
Mar  2 11:45:06 FMC SF-IMS[22601]: [22601] Event Streamer:ConnectionHandler [INFO] Added
10.48.26.47(23935) to host table
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):ConnectionHandler
[INFO] Resolved CN 10.48.26.47 to 10.48.26.47
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):ConnectionHandler
[INFO] Matched Certificate CN:10.48.26.47 to 10.48.26.47 (IPv4)
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):sfestreamer [INFO]
Got EVENT_STREAM_REQUEST length 8
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):sfestreamer [INFO]
Publishing service INFO total data size 48
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):sfestreamer [INFO]
Publishing service id:5001 - length size 8
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):sfestreamer [INFO]
Publishing service id:5000 - length size 8
Mar  2 11:45:06 FMC SF-IMS[23935]: [23935] EventStreamer child(10.48.26.47):sfestreamer [INFO]
Publishing service id:6667 - length size 8
```

The eStreamer client then checks its configuration and bookmark file in order to determine what

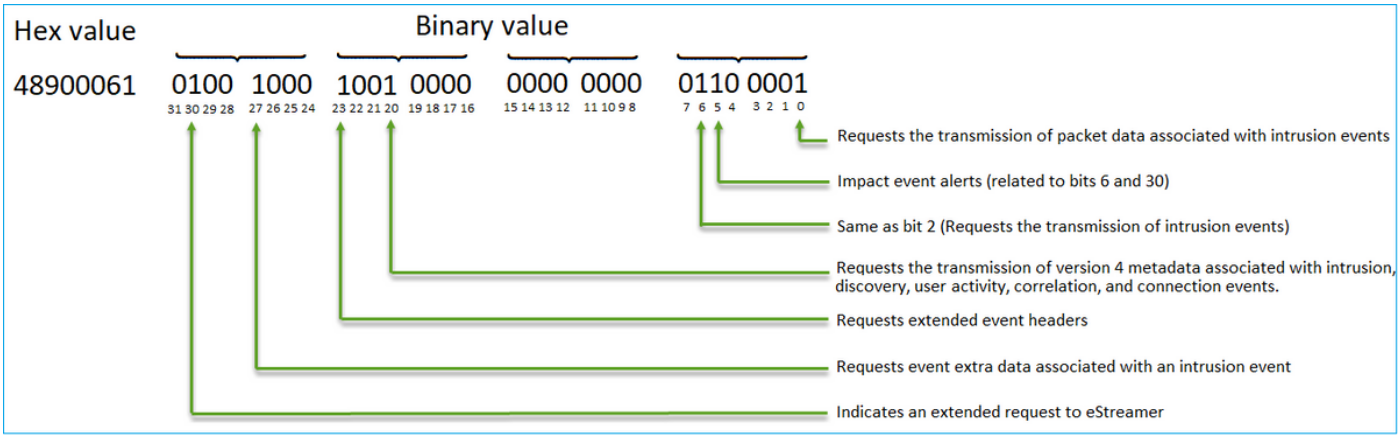
events to request and the start time:

```
2020-03-02 07:18:11,500 Connection   INFO     Connecting to 10.48.26.49:8302
2020-03-02 07:18:11,500 Connection   INFO     Using TLS v1.2
2020-03-02 07:18:11,500 Monitor      INFO     Starting Monitor.
2020-03-02 07:18:11,500 Monitor      INFO     Starting. 0 handled; average rate 0 ev/sec;
2020-03-02 07:18:11,501 Writer       INFO     Starting process.
2020-03-02 07:18:11,506 Transformer  INFO     Starting process.
2020-03-02 07:18:11,985 Bookmark     INFO     Bookmark file /root/eStreamer-eNcore/10.48.26.49-
8302_bookmark.dat does not exist.
2020-03-02 07:18:11,986 Settings     INFO     Timestamp: Start = 2 (Bookmark = 0)
2020-03-02 07:18:11,986 Receiver     INFO     EventStreamRequestMessage:
00010002000000080000000048900061
2020-03-02 07:18:11,986 SubscriberParser INFO     Starting process.
2020-03-02 07:18:11,996 Bookmark     INFO     Bookmark file /root/eStreamer-eNcore/10.48.26.49-
8302_bookmark.dat does not exist.
2020-03-02 07:18:11,996 Settings     INFO     Timestamp: Start = 2 (Bookmark = 0)
2020-03-02 07:18:11,997 Receiver     INFO     StreamingRequestMessage:
000108010000003800001a0b0000003848900061000000000009000c000400150009001f000b003d000e00470004005b
0007006500060060f0002008300000000
```

The EventStreamRequest can be correlated on FMC:

```
Mar  2 12:29:16 FMC SF-IMS[6671]: [6671] EventStreamer child(10.48.26.47):sfestreamer [INFO]
EventStream Request (0x48900061): Since 0 w/ NS Events w/ NS 6.0 Events
w/ Packets w/ Extra IDS Event data w/ Metadata v4 w/ RUA 5.2 Events w/ Impact Alerts w/ 5.3
Impact Alerts w/ Impact Flags w/ RNA 5.3 Events w/ RNA 6.0 Flow w/ Policy 5.4 Events
w/ FireAMP 6.0 Events w/ Filelog 6.0 Events w/ Send archive timestamp w/ Send Detail Request
```

The EventStreamRequest is the hexadecimal representation of the request flags described on Request Flags and must be converted to binary in order to understand if the client requested the required data. This is an example:



> **Note**: Some flag-bits might change the information provided if Extended Requests are initiated.

Based on the Request bits, the FMC pushes the data to the eStreamer client.

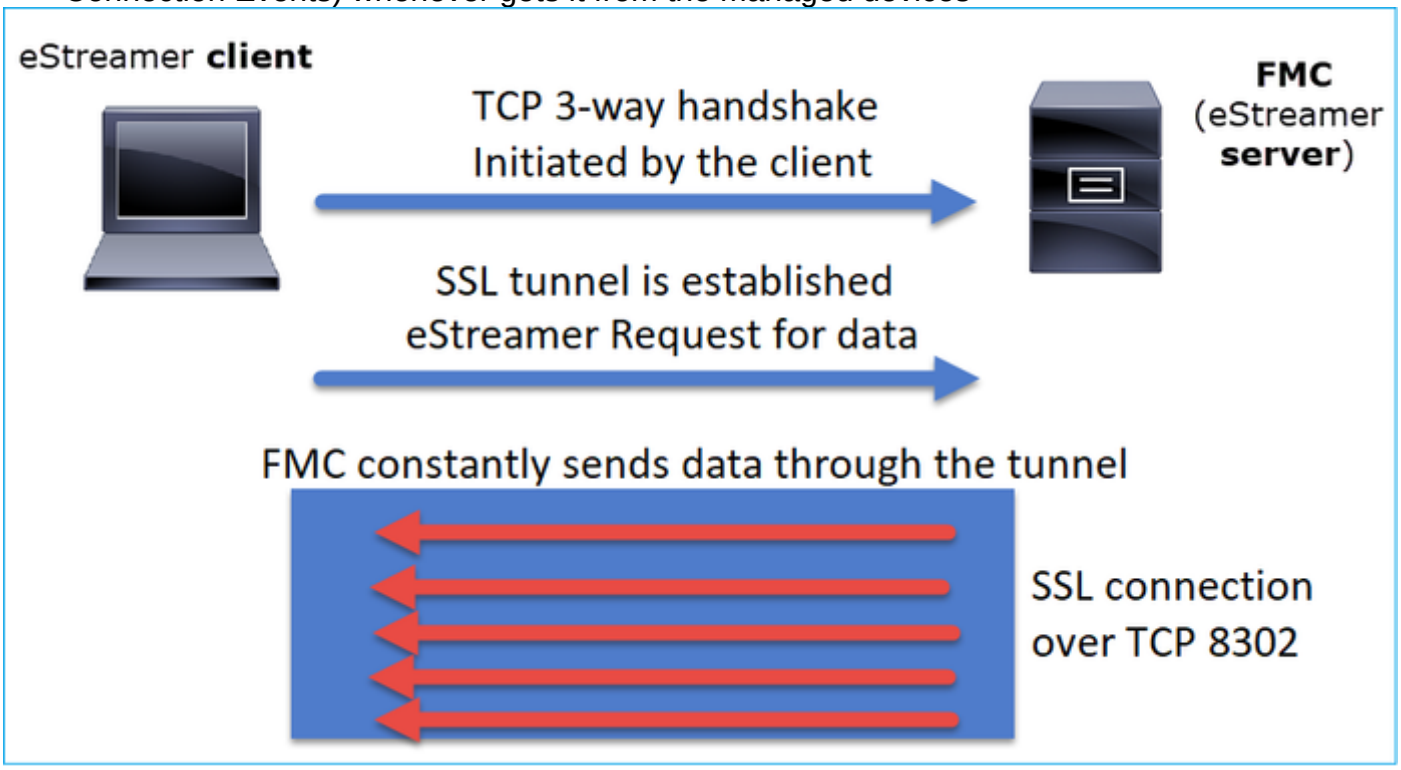
Who initiates the eStreamer connection and data transfer?

The eStreamer client. Specifically, the client establishes a TCP connection (3-way handshake), then there is an SSL negotiation with Client (mutual) authentication. Finally, through the established tunnel the FMC sends the data whenever there is data to be sent:

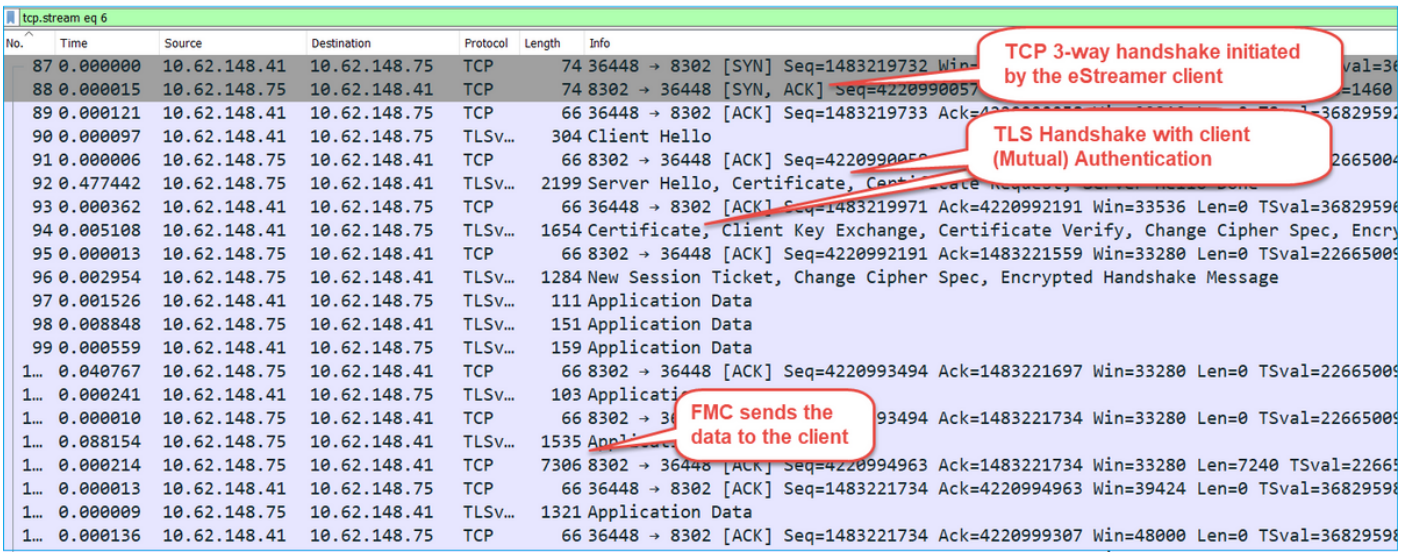
```
root@kali:~/eStreamer-eNcore# ./encore.sh foreground
2020-06-03 20:50:53,365 Monitor      INFO     Running. 100 handled; average rate 0.42 ev/sec;
2020-06-03 20:52:53,488 Monitor      INFO     Running. 100 handled; average rate 0.28 ev/sec;
2020-06-03 20:54:53,601 Monitor      INFO     Running. 100 handled; average rate 0.21 ev/sec;
2020-06-03 20:56:53,725 Monitor      INFO     Running. 100 handled; average rate 0.17 ev/sec;
```

In summary:

- The client initiates the SSL tunnel to request data (pull)
- Once the tunnel is established the tunnel stays UP and the FMC pushes data (e.g. Connection Events) whenever gets it from the managed devices



In this example, the IP 10.62.148.41 is the eStreamer client (eNcore) while the IP 10.62.148.75 is the FMC:



Configure

For details about the eNcore CLI client refer to [eStreamer eNcore CLI Operations Guide v3.5.]

The details of the eStreamer application along with the FMC configuration steps are covered in the [Event Streamer Integration Guide.]

estreamer.conf File Tuning

This section describes what can or must be modified on estreamer.conf in order for the solution to work properly. The estreamer.conf file is located within the *path*/eStreamer-eNcore directory. Here is a sample of the file contents:

```
root@kali:~/eStreamer-eNcore# cat estreamer.conf
{
    "connectTimeout": 10,
    "enabled": true,
    "handler": {
        "output@comment": "If you disable all outputters it behaves as a sink",
        "outputters": [
            {
                "adapter": "json",
                "enabled": true,
                "stream": {
                    "options": {
                        "maxLogs": 10000,
                        "rotate": true
                    },
                    "uri": "relfile:///data/json/encore.{0}.json"
                }
            }
        ],
        "records": {
            "connections": true,
            "core": true,
            "excl@comment": [
                "These records will be excluded regardless of above (overrides 'include')",
                "e.g. to exclude flow and IPS events use [ 71, 400 ]"
            ],
            "exclude": [],
            "inc@comment": "These records will be included regardless of above",
            "include": [],
            "intrusion": true,
            "metadata": true,
            "packets": true,
            "rna": true,
            "rua": true
        }
    },
    "logging": {
        "filepath": "estreamer.log",
        "format": "%(asctime)s %(name)-12s %(levelname)-8s %(message)s",
        "lev@comment": "Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE and TRACE",
        "level": "INFO",
        "stdOut": true
    },
    "monitor": {
        "bookmark": false,
        "handled": true,
        "period": 120,
        "subscribed": true,
        "velocity": false
```

```
    },
    "responseTimeout": 2,
    "star@comment": "0 for genesis, 1 for now, 2 for bookmark",
    "start": 2,
    "subscription": {
        "records": {
            "@comment": [
                "Just because we subscribe doesn't mean the server is sending. Nor does it
mean",
                "we are writing the records either. See handler.records[]"
            ],
            "archiveTimestamps": true,
            "eventExtraData": true,
            "extended": true,
            "impactEventAlerts": true,
            "intrusion": true,
            "metadata": true,
            "packetData": true
        },
        "servers": [
            {
                "host": "10.62.148.75",
                "pkcs12Filepath": "client.pkcs12",
                "port": 8302,
                "tls@comment": "Valid values are 1.0 and 1.2",
                "tlsVersion": 1.2
            }
        ]
    },
    "workerProcesses": 4
```

The subscription section

To modify the Event Streamer Request towards the server (FMC), modify the eStreamer.conf subscriptions section. For example, when you set extended requests to false it changes EventStream Request on FMC:

```
"subscription": {
        "records": {
            "@comment": [
                "Just because we subscribe doesn't mean the server is sending. Nor does it
mean",
                "we are writing the records either. See handler.records[]"
            ],
            "archiveTimestamps": true,
            "connection": true,
            "eventExtraData": true,
            "extended": false,
            "impactEventAlerts": true,
            "intrusion": true,
            "metadata": true,
            "packetData": true
        },
```

With extended requests = false:

```
Jun 3 13:48:24 firepower SF-IMS[16084]: [16084] EventStreamer child(10.48.26.47):sfestreamer
[INFO]
EventStream Request (0x08900061): Since 4294967295 w/ NS Events w/ Packets w/ Extra IDS Event
```

```
data w/
Metadata v4 w/ Impact Alerts w/ Impact Flags w/ Send archive timestamp
```

With extended requests = true:

```
Jun 3 13:50:52 firepower SF-IMS[17167]: [17167] EventStreamer child(10.48.26.47):sfestreamer
[INFO]
EventStream Request (0x48900061): Since 1590497346 w/ NS Events w/ NS 6.0 Events w/ Packets w/
Extra IDS Event data w/ Metadata
v4 w/ RUA 5.2 Events w/ Impact Alerts w/ 5.3 Impact Alerts w/ Impact Flags w/ RNA 5.3 Events w/
RNA 6.0 Flow w/ Policy 5.4 Events
v w/ FireAMP 6.0 Events w/ Filelog 6.0 Events w/ Send archive timestamp w/ Send Detail Request
```

The logging section

To enable debugs on eNcore CLI edit the estreamer.conf file and change the log level:

```
 "logging": {
        "filepath": "estreamer.log",
        "format": "%(asctime)s %(name)-12s %(levelname)-8s %(message)s",
        "lev@comment": "Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE and TRACE",
        "level": "DEBUG",
        "stdOut": true
    },
```

The monitor section

To see the number of events/second processed and current bookmark, edit the monitor section on estreamer.conf:

```
  "monitor": {
    "bookmark": true,         #If true, adds date/timestamp (see above)
    "handled": true,           #Number of records processed
    "period": 120,             #How often (in seconds) monitor writes to the log
    "subscribed": true,        #Number of records received
    "velocity": false          #A measure of whether eNcore is keeping up (>=1 is good)
  },
```

Other relevant top-level keys:

```
"connectTimeout": 10,     <- The number of seconds to wait for a response when establishing a
connection to the FMC.

"workerProcesses": 4,     <- The number of processes that eNcore spawns.
```

This value can be set from 2-12. More processes are intended to improve performance but there is an overhead cost with each process. The result is that optimal performance is achieved with the right combination of "number of processes" with the processing capability of the host machine. The best guidelines available are:

- For 2 cores: "workerProcesses": 4
- For 4 or more cores: "workerProcesses": 12

Troubleshoot

For generic eStreamer troubleshooting procedures refer to this document [Troubleshoot Issues Between FireSIGHT System and eStreamer Client (SIEM)]

For testing purposes, you can enable eNcore as a foreground process and verify the communication with FMC

```
root@kali:~/eStreamer-eNcore# ./encore.sh foreground
2020-06-04 11:48:00,048 Controller   INFO     eNcore version: 3.5.4
2020-06-04 11:48:00,049 Controller   INFO     Python version: 2.7.13 (default, Jan 19 2017,
14:48:08) \n[GCC 6.3.0 20170118]
2020-06-04 11:48:00,051 Controller   INFO     Platform version: Linux-4.13.0-kali1-amd64-x86_64-
with-Kali-kali-rolling-kali-rolling
2020-06-04 11:48:00,052 Controller   INFO     Starting client (pid=12374).
2020-06-04 11:48:00,052 Controller   INFO     Sha256:
77ac7e72d0b96e0a4b9c1c4f9a16c2de0b2b5ccf2929dd2857cf94ed96b295e3
2020-06-04 11:48:00,052 Controller   INFO     Processes: 4
2020-06-04 11:48:00,053 Controller   INFO     Settings:
 …
2020-06-04 11:48:00,053 Diagnostics  INFO     Check certificate
2020-06-04 11:48:00,054 Diagnostics  INFO     Creating connection
2020-06-04 11:48:00,054 Connection   INFO     Connecting to 10.62.148.75:8302
2020-06-04 11:48:00,054 Connection   INFO     Using TLS v1.2
2020-06-04 11:48:00,136 Diagnostics  INFO     Creating request message
2020-06-04 11:48:00,137 Diagnostics  INFO     Request message=00010002000000008ffffffff48900061
2020-06-04 11:48:00,137 Diagnostics  INFO     Sending request message
2020-06-04 11:48:00,137 Diagnostics  INFO     Receiving response message
2020-06-04 11:48:00,229 Diagnostics  INFO     Response
message=KGRwMApTJ2xlbmd0aCcKcDEKSTQ4CnNTJ3ZlcnNpb24nCnAyCkkxCnNTJ2RhdGEnCnAzClMnXHgwMFx4MDBceDEz
XHg4OVx4MDBceDAwXHgwMFx4MDhceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgxM1x4ODhceDAw
XHgwMFx4MDBceDA4XHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MWFceDBiXHgwMFx4MDBceDAw
XHgwOFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwJwpwNApzUydtZXNzYWdlVHlwZScKcDUKSTIwNTEKcy4=
2020-06-04 11:48:00,229 Diagnostics  INFO     Streaming info response
2020-06-04 11:48:00,230 Diagnostics  INFO     Connection successful
2020-06-04 11:48:00,230 Monitor      INFO     Starting Monitor.
2020-06-04 11:48:00,236 Decorator    INFO     Starting process.
2020-06-04 11:48:00,236 Transformer  INFO     Starting process.
2020-06-04 11:48:00,237 Connection   INFO     Connecting to 10.62.148.75:8302
2020-06-04 11:48:00,237 Connection   INFO     Using TLS v1.2
2020-06-04 11:48:00,238 Writer       INFO     Starting process.
2020-06-04 11:48:00,639 Bookmark     INFO     Opening bookmark file /root/eStreamer-
eNcore/10.62.148.75-8302_bookmark.dat.
2020-06-04 11:48:00,640 Settings     INFO     Timestamp: Start = 2 (Bookmark = 1591210934)
2020-06-04 11:48:00,640 Receiver     INFO     EventStreamRequestMessage:
00010002000000085ed7f3b648900061
2020-06-04 11:48:00,640 SubscriberParser INFO     Starting process.
2020-06-04 11:48:00,640 Bookmark     INFO     Opening bookmark file /root/eStreamer-
eNcore/10.62.148.75-8302_bookmark.dat.
2020-06-04 11:48:00,646 Bookmark     INFO     Opening bookmark file /root/eStreamer-
eNcore/10.62.148.75-8302_bookmark.dat.
2020-06-04 11:48:00,646 Settings     INFO     Timestamp: Start = 2 (Bookmark = 1591210934)
2020-06-04 11:48:00,647 Receiver     INFO     StreamingRequestMessage:
000108010000003800001a0b0000003848900061 5ed7f3b60009000c000400150009001f000b003d000e00470004005b
0007006500060006f0002008300000000
2020-06-04 11:48:00,653 Monitor      INFO     Running. 0 handled; average rate 1.2 ev/sec;
```

At the same time, on FMC you can see logs like these ones when the eNcore streamer client establishes the connection. Note that the FMC backend timezone is always UTC:

```
root@FMC2000-2:~# tail -f /var/log/messages
Jun  4 09:48:00 FMC2000-2 SF-IMS[4135]: [4135] Event Streamer:ConnectionHandler [INFO] Accepted
```

IPv4 connection from 10.62.148.41:36528/tcp
Jun 4 09:48:00 FMC2000-2 SF-IMS[4135]: [4135] Event Streamer:ConnectionHandler [INFO] **Added 10.62.148.41(8512) to host table**
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):SFUtil [INFO] **Found IPv4 address 10.62.148.41 for ksec-sfvm-win7-3.cisco.com**
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] **Resolved CN ksec-sfvm-win7-3.cisco.com to 10.62.148.41**
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] **Matched Certificate CN:ksec-sfvm-win7-3.cisco.com to 10.62.148.41 (IPv4)**
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Got EVENT_STREAM_REQUEST length 8
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Publishing service INFO total data size 48
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Publishing service id:5001 - length size 8
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Publishing service id:5000 - length size 8
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Publishing service id:6667 - length size 8
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] Got UEC_STREAM_REQUEST length 56
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] requested service [6667] timestamp [1591210934]
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 12, version 9
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 21, version 4
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 31, version 9
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 61, version 11
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 71, version 14
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 91, version 4
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 101, version 7
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 111, version 6
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] Got Event type 131, version 2
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):sfestreamer [INFO] **EventStream Request (0x48900061): Since 1591210934 w/ NS Events w/ NS 6.0 Events w/ Packets w/ Extra IDS Event data w/ Metadata v4 w/ RUA 5.2 Events w/ Impact Alerts w/ 5.3 Impact Alerts w/ Impact Flags w/ RNA 5.3 Events w/ RNA 6.0 Flow w/ Policy 5.4 Events w/ FireAMP 6.0 Events w/ Filelog 6.0 Events w/ Send archive timestamp w/ Send Detail Request**
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):ConnectionHandler [INFO] creating iterator for service [6667] prefix [unified2.] timestamp [1591210934]
Jun 4 09:48:00 FMC2000-2 SF-IMS[8512]: [8512] EventStreamer child(ksec-sfvm-win7-3.cisco.com):**Unified2Iterator [INFO] Opened /var/sf/archive/netmap_2/unified2.1591210800**
Jun 4 09:48:02 FMC2000-2 SF-IMS[4135]: [4135] Event Streamer:ConnectionHandler [INFO] Child with pid 8510 exited with status 5120
Jun 4 09:48:02 FMC2000-2 SF-IMS[4135]: [4135] Event Streamer:ConnectionHandler [INFO] Removed host entry for pid: 8510
Jun 4 09:48:04 FMC2000-2 SF-IMS[22296]: [25092] CloudAgent:url_license [INFO] Peer with active URLFiltering: 310f4c00-a415-11ea-bf5b-a2d6028849fe
Jun 4 09:48:04 FMC2000-2 SF-IMS[22296]: [25092] CloudAgent:url_license [INFO] Peer with active URLFiltering: d637b6f0-a414-11ea-ad97-cc17b6ea4c03
Jun 4 09:48:04 FMC2000-2 SF-IMS[22296]: [25092] CloudAgent:url_license [INFO] Peer with active URLFiltering: 873709b8-78b6-11ea-ae87-b82f93835447
Jun 4 09:48:04 FMC2000-2 SF-IMS[22296]: [25092] CloudAgent:url_license [INFO] Peer with active

```
URLFiltering: c7c0217c-78b6-11ea-a719-b7f0a277eb86
```

Items to Collect Before you Contact Cisco Technical Assistance Center (TAC)

It is highly recommended to collect these items before you contact Cisco TAC:

- The version of eStreamer eNcore
- The version of Python
- The version of host OS
- Do you see events on FMC? Share a screenshot from events + FMC eStreamer configuration
- Enable debug on eNcore CLI (as it is described in the 'logging section')
- Generate a troubleshoot file from FMC
- Provide these files from eNcore:
  estreamer.conf
  estreamer.log

Common Issues

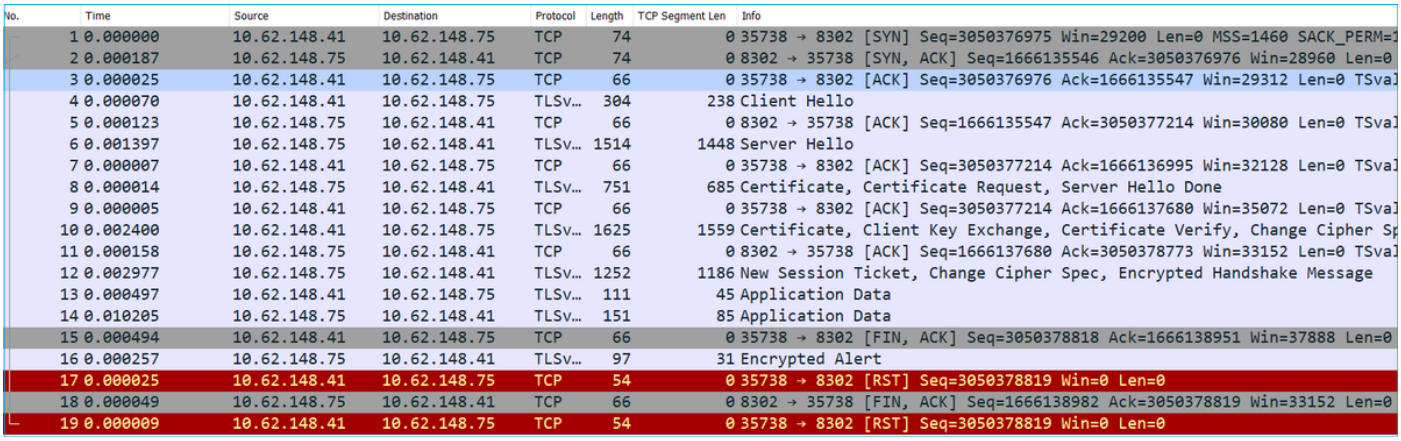
No Connectivity on TCP port 8302

Telnet from the eStreamer client to FMC port 8302 and verify connectivity is established.

Additionally, you can use the eNcore test option to test the connectivity:

```
root@kali:~/eStreamer-eNcore# ./encore.sh test
2020-05-28T16:02:56.931919 Diagnostics  INFO    Checking that configFilepath (estreamer.conf)
exists
2020-05-28 16:02:56,935 Diagnostics  INFO     Check certificate
2020-05-28 16:02:56,936 Diagnostics  INFO     Creating connection
2020-05-28 16:02:56,936 Connection   INFO     Connecting to 10.62.148.75:8302
2020-05-28 16:02:56,936 Connection   INFO     Using TLS v1.2
2020-05-28 16:02:56,946 Diagnostics  INFO     Creating request message
2020-05-28 16:02:56,946 Diagnostics  INFO     Request message=00010002000000008ffffffff48900061
2020-05-28 16:02:56,946 Diagnostics  INFO     Sending request message
2020-05-28 16:02:56,946 Diagnostics  INFO     Receiving response message
2020-05-28 16:02:56,957 Diagnostics  INFO     Response
message=KGRwMApTJ2xlbmd0aCcKcDEKSTQ4CnNTJ3ZlcnNpb24nCnAyCkkxCnNTJ2RhdGEnCnAzClMnXHgwMFx4MDBceDEz
XHg4OVx4MDBceDAwXHgwMFx4MDhceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgxM1x4ODhceDAw
XHgwMFx4MDBceDA4XHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MWFceDBiXHgwMFx4MDBceDAw
XHgwOFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwJwpwNApzUydtZXNzYWdlVHlwZScKcDUKSTIwNTEKcy4=
2020-05-28 16:02:56,957 Diagnostics  INFO     Streaming info response
2020-05-28 16:02:56,957 Diagnostics  INFO     Connection successful
```

This is a successful connection attempt as it is seen in Wireshark (10.62.148.41 is the eNcore IP while 10.62.148.75 is the FMC):



No.	Time	Source	Destination	Protocol	Length	TCP Segment Len	Info
1	0.000000	10.62.148.41	10.62.148.75	TCP	74	0	35738 → 8302 [SYN] Seq=3050376975 Win=29200 Len=0 MSS=1460 SACK_PERM=1
2	0.000187	10.62.148.75	10.62.148.41	TCP	74	0	8302 → 35738 [SYN, ACK] Seq=1666135546 Ack=3050376976 Win=28960 Len=0
3	0.000025	10.62.148.41	10.62.148.75	TCP	66	0	35738 → 8302 [ACK] Seq=3050376976 Ack=1666135547 Win=29312 Len=0 TSval
4	0.000070	10.62.148.41	10.62.148.75	TLSv…	304	238	Client Hello
5	0.000123	10.62.148.75	10.62.148.41	TCP	66	0	8302 → 35738 [ACK] Seq=1666135547 Ack=3050377214 Win=30080 Len=0 TSval
6	0.001397	10.62.148.75	10.62.148.41	TLSv…	1514	1448	Server Hello
7	0.000007	10.62.148.41	10.62.148.75	TCP	66	0	35738 → 8302 [ACK] Seq=3050377214 Ack=1666136995 Win=32128 Len=0 TSval
8	0.000014	10.62.148.75	10.62.148.41	TLSv…	751	685	Certificate, Certificate Request, Server Hello Done
9	0.000005	10.62.148.41	10.62.148.75	TCP	66	0	35738 → 8302 [ACK] Seq=3050377214 Ack=1666137680 Win=35072 Len=0 TSval
10	0.002400	10.62.148.41	10.62.148.75	TLSv…	1625	1559	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Sp
11	0.000158	10.62.148.75	10.62.148.41	TCP	66	0	8302 → 35738 [ACK] Seq=1666137680 Ack=3050378773 Win=33152 Len=0 TSval
12	0.002977	10.62.148.75	10.62.148.41	TLSv…	1252	1186	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
13	0.000497	10.62.148.41	10.62.148.75	TLSv…	111	45	Application Data
14	0.010205	10.62.148.75	10.62.148.41	TLSv…	151	85	Application Data
15	0.000494	10.62.148.41	10.62.148.75	TCP	66	0	35738 → 8302 [FIN, ACK] Seq=3050378818 Ack=1666138951 Win=37888 Len=0
16	0.000257	10.62.148.75	10.62.148.41	TLSv…	97	31	Encrypted Alert
17	0.000025	10.62.148.41	10.62.148.75	TCP	54	0	35738 → 8302 [RST] Seq=3050378819 Win=0 Len=0
18	0.000049	10.62.148.75	10.62.148.41	TCP	66	0	8302 → 35738 [FIN, ACK] Seq=1666138982 Ack=3050378819 Win=33152 Len=0
19	0.000009	10.62.148.41	10.62.148.75	TCP	54	0	35738 → 8302 [RST] Seq=3050378819 Win=0 Len=0

Certificate CN Does Not Match the Remote Host

If the eStreamer client is behind NAT, the certificate must be generated with the upstream IP address or errors like these are seen:

```
Mar  2 11:30:01 FMC SF-IMS[16921]: [16921] Event Streamer:ConnectionHandler [INFO] Accepted IPv4
connection from 10.48.26.47:46529/tcp
Mar  2 11:30:01 FMC SF-IMS[16921]: [16921] Event Streamer:ConnectionHandler [INFO] Added
10.48.26.47 to host table
Mar  2 11:30:01 FMC SF-IMS[16921]: [16921] Event Streamer:ConnectionHandler [INFO] Added
10.48.26.47(17659) to host table
Mar  2 11:30:01 FMC SF-IMS[17659]: [17659] EventStreamer child(192.168.27.100):ConnectionHandler
[INFO] Resolved CN 192.168.27.100 to 192.168.27.100
Mar  2 11:30:01 FMC SF-IMS[17659]: [17659] EventStreamer child(192.168.27.100):ConnectionHandler
[ERROR] Certificate Common Name 192.168.27.100 does not match remote host: 10.48.26.47. It was
issued to a different client.
Mar  2 11:30:02 FMC SF-IMS[16921]: [16921] Event Streamer:ConnectionHandler [INFO] Child with
pid 17659 exited with status 0
Mar  2 11:30:02 FMC SF-IMS[16921]: [16921] Event Streamer:ConnectionHandler [INFO] Removed host
entry for pid: 17659
```

FMC DNS Resolution for the eStreamer Client is Incorrect

In case FMC has wrong DNS entries for the eStreamer client the events do not reach the client. To identify if this is the issue take a capture on FMC. In this example, the FMC receives a TCP SYN packet from streamer client host ksec-sfvm-win7-3.cisco.com:

```
root@FMC2000-2:/var/sf/archive/netmap_2# tcpdump -i eth0 port 8302
HS_PACKET_BUFFER_SIZE is set to 4.
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:32:45.453401 IP ksec-sfvm-win7-3.cisco.com.36428 > FMC2000-2.8302: Flags [S], seq 2427598184,
win 29200, options [mss 1460,sackOK,TS val 3681355935 ecr 0,nop,wscale 7], length 0
18:32:45.453425 IP FMC2000-2.8302 > ksec-sfvm-win7-3.cisco.com.36428: Flags [S.], seq
1996800475, ack 2427598185, win 28960, options [mss 1460,sackOK,TS val 2264897265 ecr
3681355935,nop,wscale 7], length 0
18:32:45.453539 IP ksec-sfvm-win7-3.cisco.com.36428 > FMC2000-2.8302: Flags [.], ack 1, win 229,
options [nop,nop,TS val 3681355935 ecr 2264897265], length 0
```

You can use the **-n** flag to see the resolved IP:

```
root@FMC2000-2:/var/sf/archive/netmap_2# tcpdump -i eth0 port 8302 -n
HS_PACKET_BUFFER_SIZE is set to 4.
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:34:58.015971 IP 10.62.148.41.36434 > 10.62.148.75.8302: Flags [S], seq 713101140, win 29200,
options [mss 1460,sackOK,TS val 3681488496 ecr 0,nop,wscale 7], length 0
```

Alternatively, you can use the **nslookup** command tool from the FMC CLI:

```
root@FMC2000-2:/var/sf/archive/netmap_2# nslookup ksec-sfvm-win7-3.cisco.com
Server:         1.2.3.4
Address:        1.2.3.4#53

Name: ksec-sfvm-win7-3.cisco.com Address: 10.62.148.41
```

eStreamer Communication Issue due to SSL Certificate Error

Ensure that the eStreamer client uses the correct FMC SSL Certificate. If the certificate is incorrect on FMC /var/log/message files you see these events:

```
Jun 11 14:15:33 FMC SF-IMS[14211]: [14211] Event Streamer:ConnectionHandler [INFO]
estreamer.c:2149:AcceptConnections(): Accepted IPv4 connection from 192.0.2.100:42143/tcp
Jun 11 14:15:33 FMC SF-IMS[14211]: [14211] Event Streamer:ConnectionHandler [INFO]
estreamer.c:389:allowConnection(): Added 192.0.2.100 to host table
Jun 11 14:15:33 FMC SF-IMS[14211]: [14211] Event Streamer:ConnectionHandler [INFO]
estreamer.c:334:rememberPid(): Added 192.0.2.100(13687) to host table
Jun 11 14:15:33 FMC SF-IMS[14211]: [14211] Event Streamer:ConnectionHandler [DEBUG]
estreamer.c:1347:AcceptConnection(): Created new estreamer child with src 192.0.2.100 : pid
13615
Jun 11 14:15:34 FMC SF-IMS[13687]: [13615] Event Streamer:ConnectionHandler [ERROR]
estreamer.c:1116:AcceptConnection(): SSL_accept failed, SSL_get_error reports SSL_ERROR_SYSCALL
```

You can delete the eStreamer client on FMC and reconfigure it. This regenerates the SSL Certificate. Import the new Certificate into the eStreamer client.

Wrong IP Address Configured on eStreamer for ASA SFR Module Integration

On eStreamer client, you must use the SFR module IP. On ASA run the command **show sfr module details** to see the module IP.

ArcSight Common Event Format (CEF)

The Arcsight Common Event Format Standard defines the key-value pairs that must be sent from eNcore CLI. If there is inconsistency on data received on Arcisght, ie: missing fields, out of order, or some data is not parsed correctly on Arcsight client, it is useful to modify the configuration to write to a log file by setting. This helps to determine where the problem lies.

```
    "handler": {
        "output@comment": "If you disable all outputters it behaves as a sink",
        "outputters": [
            {
```

```
            "adapter": "cef",
            "enabled": true,
            "stream": {
                "uri": "relfile:///data/data.{0}.cef"
            }
        }
    ],
```

RAW CEF events are written in a line with each field separated by pipe “|”:

```
<13>May 26 09:31:39 kali2 CEF:0|Cisco|Firepower|6.0|RNA:1003:1|CONNECTION STATISTICS|3|act=Allow
app=STUN bytesOut=820 cs1=test cs1Label=fwPolicy
cs2=Default Action cs2Label=fwRule cs3=INSIDE cs3Label=ingressZone cs4=OUTSIDE
cs4Label=egressZone cs5Label=secIntelCategory deviceExternalId=1
deviceInboundInterface=inside deviceOutboundInterface=outside dpt=9000 dst=216.151.129.103
dvchost=10.48.26.45 dvcpid=2 end=1590497212000 externalId=50850
proto=17 reason=N/A requestClientApplicatio
```

eStreamer Client does not Show All Logs

This is often due to eStreamer client oversubscription (too many events sent by the FMC). Run this command on the eStreamer client-side and check if the Recv-Q counter is high. This is the count of bytes not copied by the user program connected to this socket. In this example there are 143143 Bytes pending on the client-side:

```
root@kali:~# netstat -an | egrep "8302|Recv-Q"
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp   143143      0 10.62.148.41:36732      10.62.148.75:8302       ESTABLISHED
```

Check the events per second received by the eStreamer client. This provides you an indication of the events per second rate:

```
root@kali:~/eStreamer-eNcore# cat estreamer.log | grep "ev/sec"
```

Try to lower the amount of data requested by the eStreamer client, or the types of events sent by the FMC. Alternatively, you can try to increase the amount of resources allocated on the eStreamer client side.

Frequently Asked Questions (FAQ)

Where to get the eNcore-cli package?

- Check the FMC software download page, **Firepower System Tools and APIs - eNcore for CEF**
- Alternatively, you can get the latest eNcore file from https://github.com/CiscoSecurity/fp-05-firepower-cef-connector-arcsight/tree/master/assets

When there is an FMC full backup in progress the eStreamer does not generate events. Is this normal?

Yes, it is expected behavior. From the FMC Config Guide When to Back Up:

While the system collects backup data, there may be a temporary pause in data correlation (FMC only), and you may be prevented from changing configurations related to the backup.

Are there any special licenses required for FMC integration with eStreamer client (e.g. Qradar)?

No

Where do eStreamer events get sourced from?

The FMC. Specifically, the FMC gets the events from the managed devices (FTD) and forwards them to the eStreamer client(s) like eNcore, ArcSight, Splunk, QRadar, LogRhythm, etc.

Is there any compatibility matrix between Splunk and eNcore?

Check the Splunk docs for compatibility information. For example, to see which Splunk versions are compatible with eNcore version 3.6.8 check https://splunkbase.splunk.com/app/3662/

Can eStreamer eNcore consume data from multiple FMCs?

At the time of this writing, no. Check enhancement request CSCvq14351

What are the recommended options to configure eStreamer for FMC High Availability (HA) setup?

The recommendation is to configure only the active FMC unit for eStreamer. If you configure both FMC units for eStreamer the SIEM receives duplicates events because the standby FMC responds to eStreamer request. Related enhancement request: CSCvi95944

Does an FMC upgrade require to manually generate new eStreamer certificates?

No

Do Security Intelligence events get sent to eStreamer client? Is it possible to select Security Intelligence events as a separate category and send them to an eStreamer client?

The Security Intelligence (SI) events are included under the category of Connection events and not as a separate category. Because of this, there is no separate SI event that is sent to the streamer. Related enhancement request: CSCva39052

Is it possible to specify on FMC the sensors/managed devices that have their eStreamer events sent to the eStreamer client?

With only one FMC domain currently, this is not possible. Related enhancement request CSCvt31270. Alternatively, you configure on FMC two different domains. In the first domain, you add all the managed devices that you want to enable eStreamer for and configure the eStreamer client. For the second domain, you add the rest of the devices and don't configure eStreamer.

What is the version of eStreamer on the Firepower? I need this information for the SIEM configuration (e.g. LogRhythm)

To check the Firepower (FMC) version from the FMC UI navigate to **Help** (top right corner) **> About > Software version**

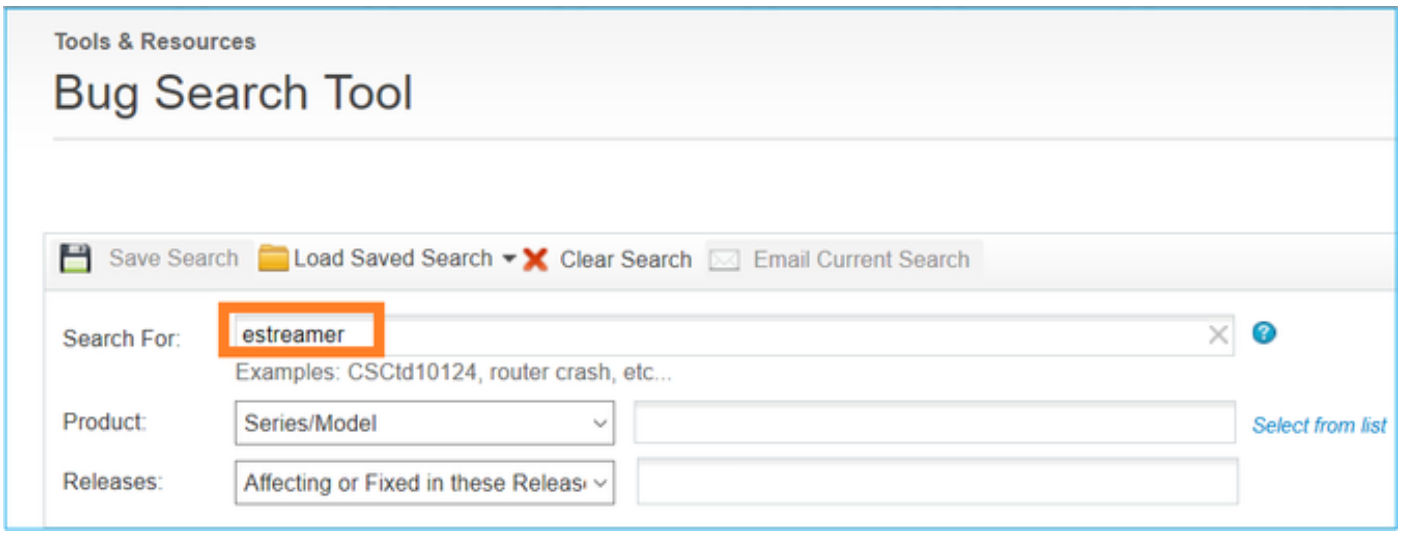
When FMC is configured with domains how to see the domain info in the FMC eStreamer data?

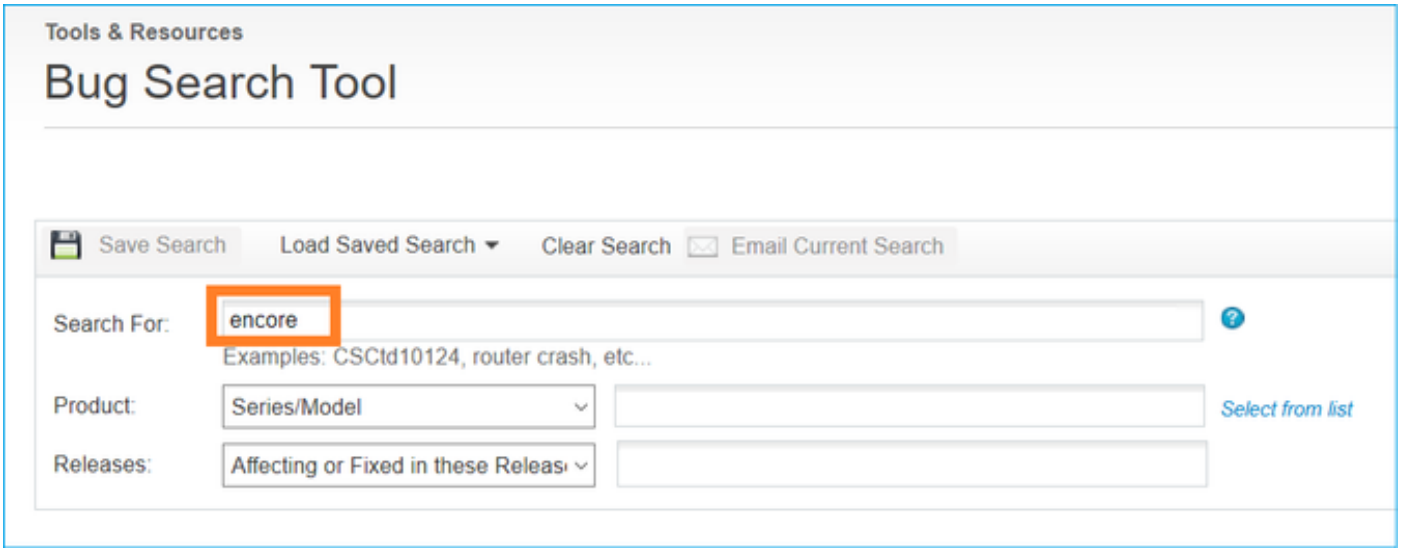
In the eStreamer Integration Guide check the **Netmap ID** number next to the Record Type in the header section of many different record types. The Netmap ID number can be converted into Domain or Device name using **Netmap Domain Metadata** (Record Type 350) and **Managed Device Record Metadata** (Record Type 123), respectively.

The client application must interpret the binary data and metadata according to the information provided in the eStreamer Integration Guide.

Known Issues

Open the Bug Search Tool and search for streamer and encore issues, e.g.





Related Information

- eStreamer Server Streaming