eStreamer eNcore

eStreamer eNcore Operations Guide v1.0

May 1, 2017

Version 1.0

Cisco Systems, Inc.
Corporate Headquarters
170 West Tasman Drive
San Jose, CA 95134-1706 USA
http://www.cisco.com
Tel: 408 526-4000 Toll Free: 800 553-NETS (6387)
Fax: 408 526-4100

Contents

List of Figures and Tables

About This eStreamer eNcore Operations Guide v1.0

Author	Sam Strachan (sastrach)
Change Authority	Cisco Systems Advanced Services, Security & Collaboration IDT, Implementation Americas
Content ID	585637
Project ID	852716

Revision History

Revision	Date	Name or User ID	Comments
0.1	04/28/2017	Sam Strachan	Initial Draft
0.2	04/30/2017	Michelle Jenkins; Huxley Barbee	Technical Writer Review; Technical Peer Review
1.0	04/30/2017	Michelle Jenkins	Initial Release

Document Conventions

📝 Alerts readers to take note. Notes contain helpful suggestions or references to material not covered in the document.

⚠ Alerts readers to be careful. In this situation, you might do something that could result in equipment damage or loss of data.

❗ Alerts readers of a situation that could cause bodily injury. They need to be aware of the hazards involved with electrical circuitry and familiarize themselves with standard practices for preventing accidents.

⏰ Alerts the reader that they can save time by performing the action described in the paragraph affixed to this icon.

ℹ Alerts the reader that the information affixed to this icon will help them solve a problem. The information might not be troubleshooting or even an action, but it could be useful information similar to a Timesaver.

1 Introduction

1.1 Document Purpose

This document seeks to outline the background and usage of the eStreamer eNcore client in order to assist users with installation and execution.

1.2 Background

The Cisco Event Streamer (i.e., eStreamer) allows users to stream system intrusion, discovery, and connection data from Firepower Management Center or managed device (i.e., the eStreamer server) to external client applications. eStreamer responds to client requests with terse, compact, binary encoded messages that facilitate high performance.

Historically, the eStreamer SDK has been wrapped with some additional code to create separate perl applications (e.g., the Cisco eStreamer for Splunk app and the CEF agent).

eStreamer eNcore is a completely new, multi-platform, multi-process Python application that is compatible with FMC versions 6.0 and above.

1.3 Application Summary

eNcore is an all-purpose client, which requests all possible events from eStreamer, parses the binary content, and outputs events in various formats to support other SIEMs. eNcore was built from scratch in Python with a scalable and fast multi-process architecture. It supports version 6.0 of Firepower Management Center. It was built and tested on CentOS 7, but should work with any Linux distribution that supports the pre-requisites. The software will run on Windows, although, it has not been made production-ready with a Windows Service yet.

2 Pre-requisite

To install eStreamer-eNcore, the following pre-requisites are required.

1. Python 2.7
2. pyOpenSSL - a thin wrapper around a subset of the OpenSSL library

The **encore.sh** script should guide you through all these points if you wish to get going immediately, but it is worth being familiar with these points prior to install.

To check whether Python2.7 is present, use following command.

```
which python
```

To test whether Python2.7 is present, use the following command.

```
whereis python
```

If you are installing on a device running Splunk, then it is worth noting that Splunk has its own version of Python. The Splunk Python has been compiled differently from the normal distribution – specifically, it is built with PyUnicodeUCS2. The **encore.sh** script will detect this and warn you. If you encounter this problem, then you will need to create a new user and run eStreamer-eNcore as that user.

To check for pyOpenSSL, use the following command

```
pip list | grep -i pyOpenSSL
```

2.1 Installation

2.1.1 Python 2.7 Installation

Use the following command to install Python on CentOS.

```
sudo yum install python
```

2.1.2 pyOpenSSL

Install pyOpenSSL as follows.

```
sudo yum install python-pip python-devel openssl-devel gcc
sudo pip install pyOpenSSL
```

2.1.3 EPEL Repo Dependency for RHEL

If you are having problems installing these packages, then you may need to enable the EPEL repository.

```
wget http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-9.noarch.rpm
sudo rpm -ivh epel-release-7-9.noarch.rpm
```

2.1.4 Dependencies for Windows

Windows is not yet supported for production execution; the code must be wrapped in a Windows Service. If, however, you wish to attempt an install, then you will need to run the following commands.

```
pip install pyOpenSSL
pip install win-inet-pton
```

3 Operations

3.1 Installation

3.1.1 Download eStreamer-eNcore-X.YY.tar.gz

Use the following command to copy the file from your local machine to the target device.

```
scp /path/to/eStreamer-eNcore-X.YY.tar.gz user@host:/path/in/device
```

3.1.2 Extract Files

```
tar -xf eStreamer-eNcore-X.YY.tar.gz
```

3.1.3 Create (or copy existing) PKCS12 file

See appendix for instructions.

3.1.4 Install the PKCS12 File

```
scp /path/host.pkcs12 user@host:/path/eStreamer_eNcore-X.YY/client.pkcs12
```

3.1.5 Test

Change the working directory to eStreamer-eNcore-X.YY using the following command.

```
cd eStreamer-eNcore-X.YY
```

Then, run the encore shell script – you will be guided through any additional configuration.

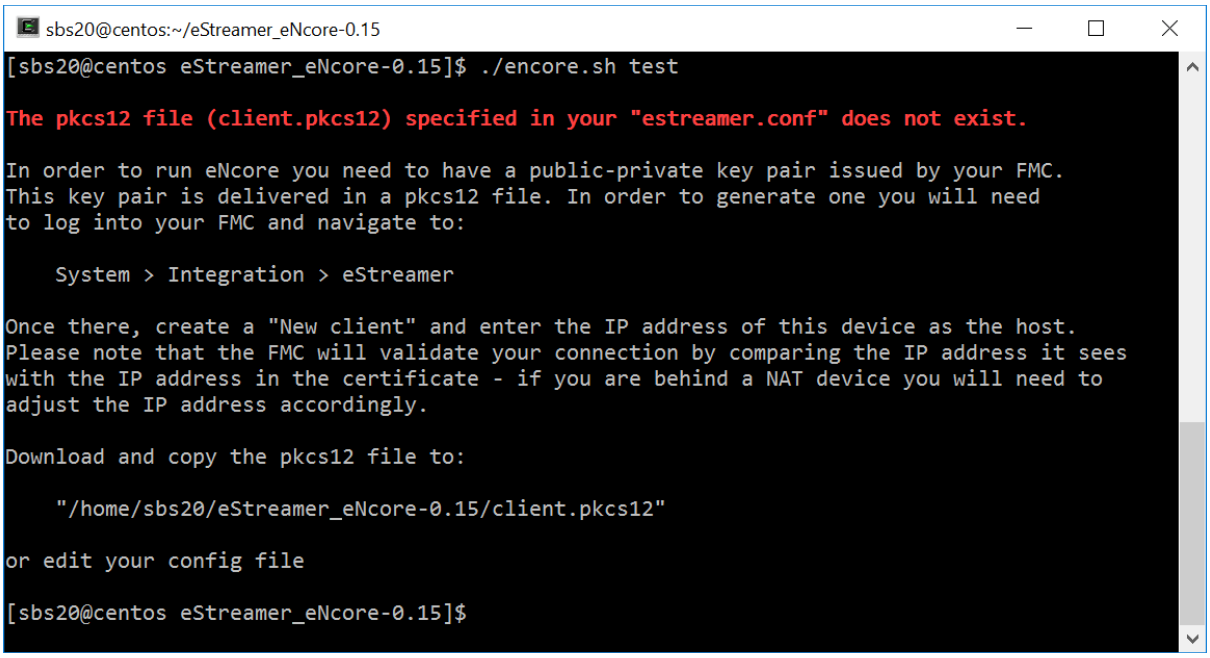
```
./encore.sh test
```

The script will verify that you have the pre-requisites installed, notably:

- Python 2.7
- the correct build of Python
- pyOpenSSL
- a client.pkcs12 file
- a valid host

If there are any missing items, you will be presented with an explanation. An example explanation is in the following figure.

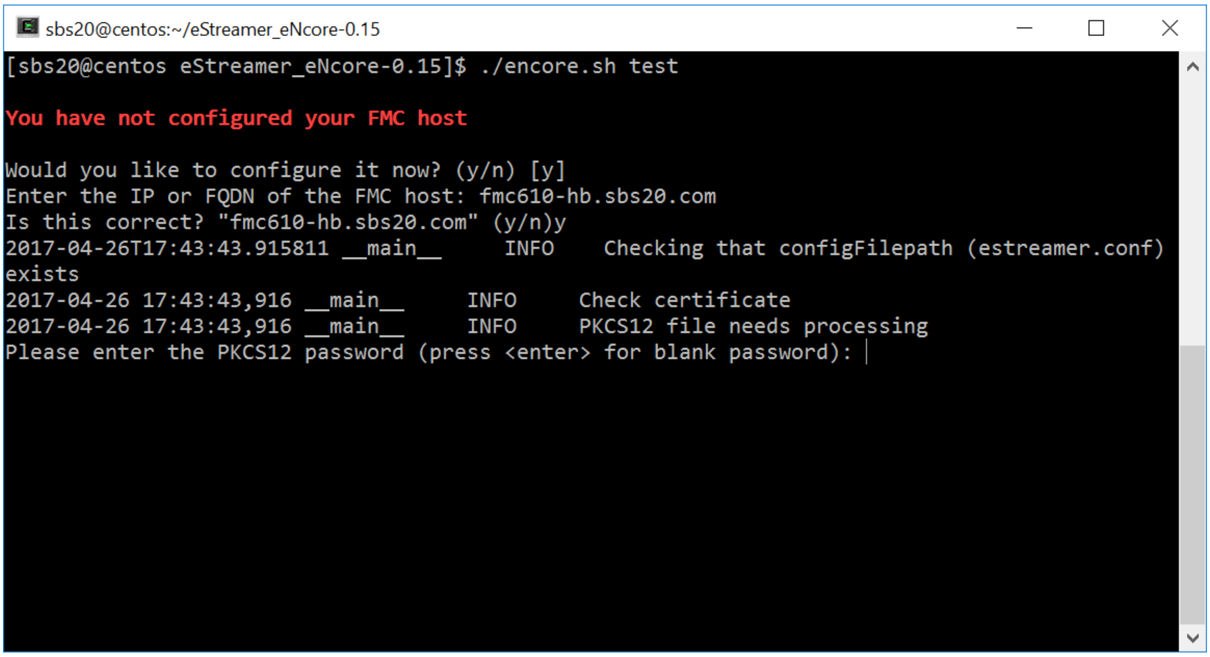
Figure 1: Missing pkcs12 File



```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh test

The pkcs12 file (client.pkcs12) specified in your "estreamer.conf" does not exist.

In order to run eNcore you need to have a public-private key pair issued by your FMC.
This key pair is delivered in a pkcs12 file. In order to generate one you will need
to log into your FMC and navigate to:

    System > Integration > eStreamer

Once there, create a "New client" and enter the IP address of this device as the host.
Please note that the FMC will validate your connection by comparing the IP address it sees
with the IP address in the certificate - if you are behind a NAT device you will need to
adjust the IP address accordingly.

Download and copy the pkcs12 file to:

    "/home/sbs20/eStreamer_eNcore-0.15/client.pkcs12"

or edit your config file

[sbs20@centos eStreamer_eNcore-0.15]$
```

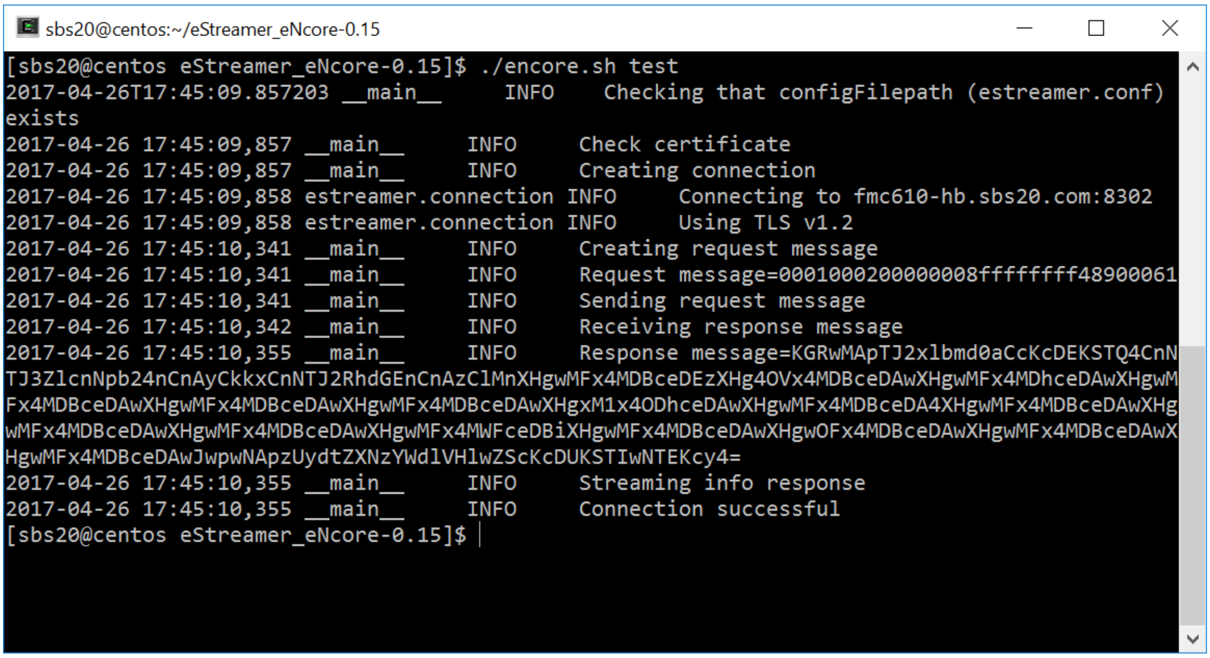
You will then be prompted to enter the IP / FQDN of the FMC and the PKCS12 file password.

Figure 2: Enter Password



```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh test

You have not configured your FMC host

Would you like to configure it now? (y/n) [y]
Enter the IP or FQDN of the FMC host: fmc610-hb.sbs20.com
Is this correct? "fmc610-hb.sbs20.com" (y/n)y
2017-04-26T17:43:43.915811 __main__     INFO    Checking that configFilepath (estreamer.conf)
exists
2017-04-26 17:43:43,916 __main__     INFO     Check certificate
2017-04-26 17:43:43,916 __main__     INFO     PKCS12 file needs processing
Please enter the PKCS12 password (press <enter> for blank password):
```

Figure 3: Successful Test

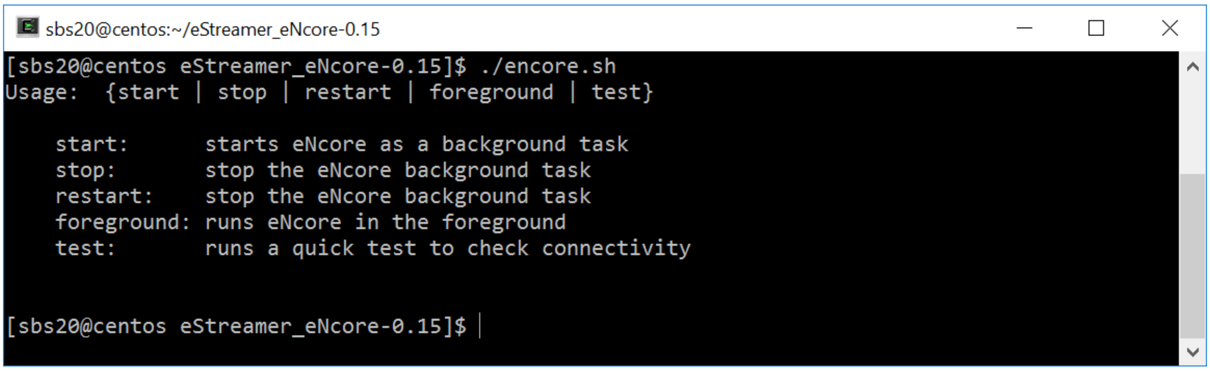


```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh test
2017-04-26T17:45:09.857203 __main__     INFO    Checking that configFilepath (estreamer.conf)
exists
2017-04-26 17:45:09,857 __main__     INFO     Check certificate
2017-04-26 17:45:09,857 __main__     INFO     Creating connection
2017-04-26 17:45:09,858 estreamer.connection INFO     Connecting to fmc610-hb.sbs20.com:8302
2017-04-26 17:45:09,858 estreamer.connection INFO     Using TLS v1.2
2017-04-26 17:45:10,341 __main__     INFO     Creating request message
2017-04-26 17:45:10,341 __main__     INFO     Request message=0001000200000008ffffffff48900061
2017-04-26 17:45:10,341 __main__     INFO     Sending request message
2017-04-26 17:45:10,342 __main__     INFO     Receiving response message
2017-04-26 17:45:10,355 __main__     INFO     Response message=KGRwMApTJ2xlbmd0aCcKcDEKSTQ4CnN
TJ3ZlcnNpb24nCnAyCkkxCnNTJ2RhdGEnCnAzClMnXHgwMFx4MDBceDEzXHg4OVx4MDBceDAwXHgwMFx4MDhceDAwXHgwM
Fx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgxM1x4ODhceDAwXHgwMFx4MDBceDA4XHgwMFx4MDBceDAwXHg
wMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MWFceDBiXHgwMFx4MDBceDAwXHgwOFx4MDBceDAwXHgwMFx4MDBceDAwX
HgwMFx4MDBceDAwJwpwNApzUydtZXNzYWdlVHlwZScKcDUKSTIwNTEKcy4=
2017-04-26 17:45:10,355 __main__     INFO     Streaming info response
2017-04-26 17:45:10,355 __main__     INFO     Connection successful
[sbs20@centos eStreamer_eNcore-0.15]$
```

3.1.6 Run

If you run **encore.sh** without any parameters, you will be presented with brief instructions.

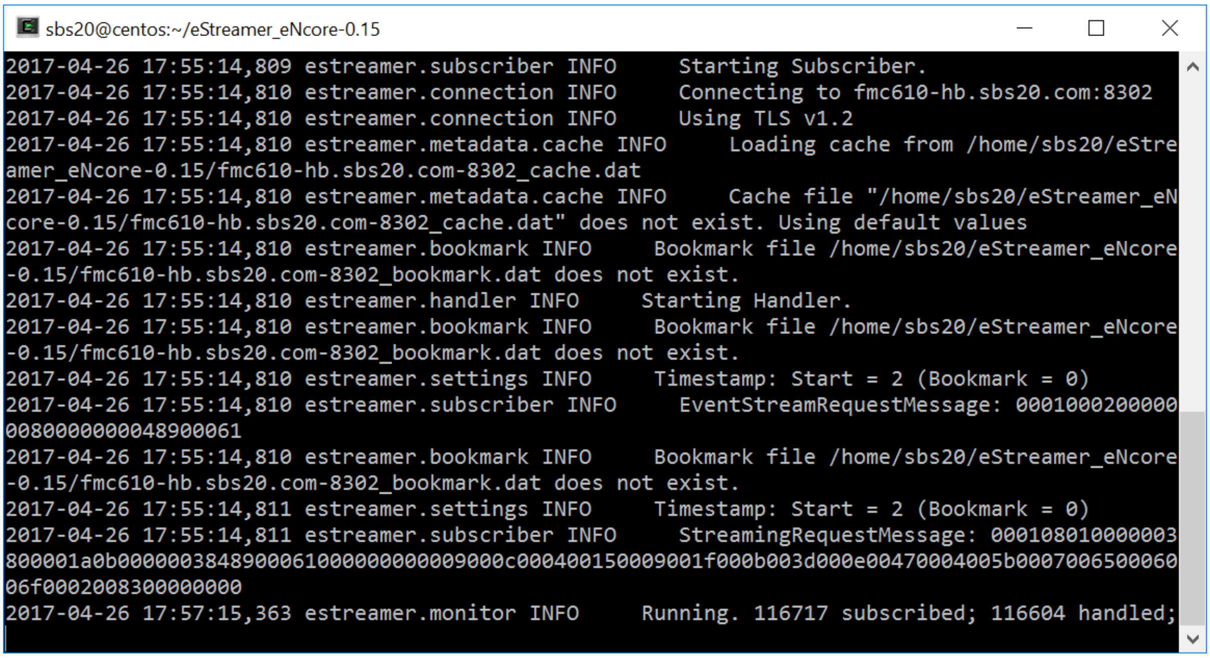
Figure 4: Help Screen



```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh
Usage:  {start | stop | restart | foreground | test}

    start:      starts eNcore as a background task
    stop:       stop the eNcore background task
    restart:    stop the eNcore background task
    foreground: runs eNcore in the foreground
    test:       runs a quick test to check connectivity


[sbs20@centos eStreamer_eNcore-0.15]$
```

For your first run, it is recommended to run it in the foreground so you can see what is happening. Every two minutes, the screen will update with a note of how many records have been processed. If you wish to change the update frequency, then see the **monitor.period** configuration setting.

Figure 5: Running in the Foreground with Monitor Status



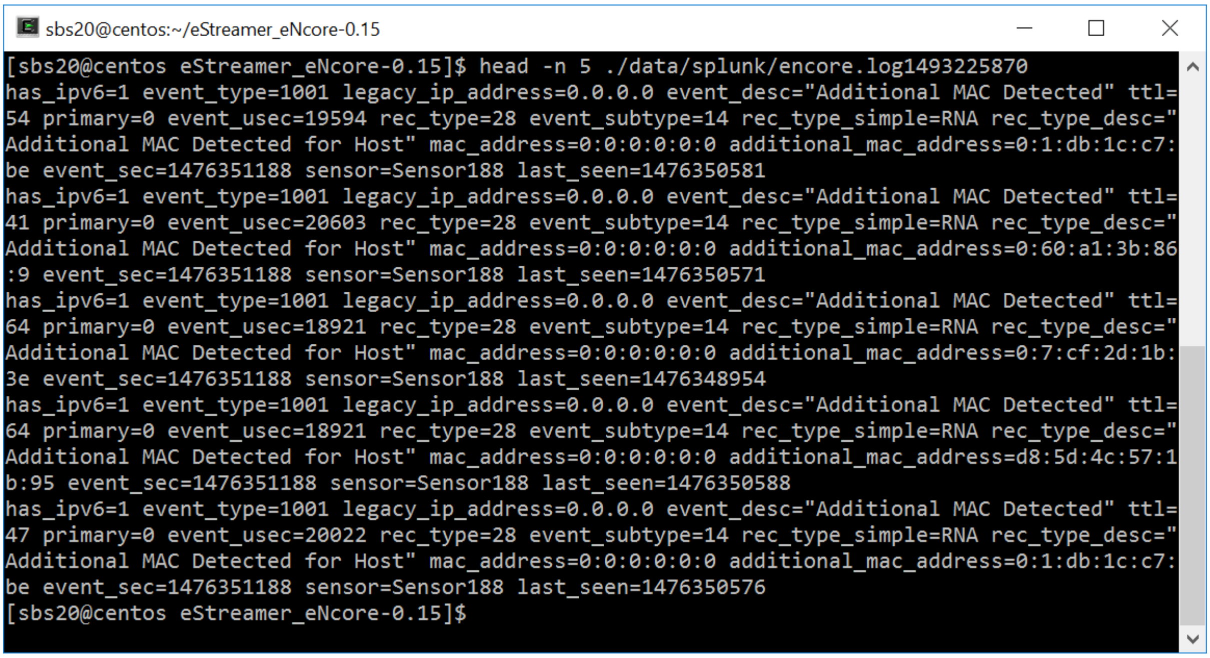
```
2017-04-26 17:55:14,809 estreamer.subscriber INFO     Starting Subscriber.
2017-04-26 17:55:14,810 estreamer.connection INFO     Connecting to fmc610-hb.sbs20.com:8302
2017-04-26 17:55:14,810 estreamer.connection INFO     Using TLS v1.2
2017-04-26 17:55:14,810 estreamer.metadata.cache INFO     Loading cache from /home/sbs20/eStre
amer_eNcore-0.15/fmc610-hb.sbs20.com-8302_cache.dat
2017-04-26 17:55:14,810 estreamer.metadata.cache INFO     Cache file "/home/sbs20/eStreamer_eN
core-0.15/fmc610-hb.sbs20.com-8302_cache.dat" does not exist. Using default values
2017-04-26 17:55:14,810 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer_eNcore
-0.15/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-04-26 17:55:14,810 estreamer.handler INFO     Starting Handler.
2017-04-26 17:55:14,810 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer_eNcore
-0.15/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-04-26 17:55:14,810 estreamer.settings INFO     Timestamp: Start = 2 (Bookmark = 0)
2017-04-26 17:55:14,810 estreamer.subscriber INFO     EventStreamRequestMessage: 0001000200000
0080000000048900061
2017-04-26 17:55:14,810 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer_eNcore
-0.15/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-04-26 17:55:14,811 estreamer.settings INFO     Timestamp: Start = 2 (Bookmark = 0)
2017-04-26 17:55:14,811 estreamer.subscriber INFO     StreamingRequestMessage: 000108010000003
800001a0b000000384890006100000000000009000c000400150009001f000b003d000e00470004005b00070065000600
06f0002008300000000
2017-04-26 17:57:15,363 estreamer.monitor INFO     Running. 116717 subscribed; 116604 handled;
```

To stop the foreground process, press **ctrl-c.**

3.1.7 Finding Your Data

By default, eNcore outputs Splunk compatible key-value pair text files to a relative subdirectory **data/splunk**. This is easily changed by editing the configuration file.

Figure 6: Output Data



```
[sbs20@centos eStreamer_eNcore-0.15]$ head -n 5 ./data/splunk/encore.log1493225870
has_ipv6=1 event_type=1001 legacy_ip_address=0.0.0.0 event_desc="Additional MAC Detected" ttl=
54 primary=0 event_usec=19594 rec_type=28 event_subtype=14 rec_type_simple=RNA rec_type_desc="
Additional MAC Detected for Host" mac_address=0:0:0:0:0:0 additional_mac_address=0:1:db:1c:c7:
be event_sec=1476351188 sensor=Sensor188 last_seen=1476350581
has_ipv6=1 event_type=1001 legacy_ip_address=0.0.0.0 event_desc="Additional MAC Detected" ttl=
41 primary=0 event_usec=20603 rec_type=28 event_subtype=14 rec_type_simple=RNA rec_type_desc="
Additional MAC Detected for Host" mac_address=0:0:0:0:0:0 additional_mac_address=0:60:a1:3b:86
:9 event_sec=1476351188 sensor=Sensor188 last_seen=1476350571
has_ipv6=1 event_type=1001 legacy_ip_address=0.0.0.0 event_desc="Additional MAC Detected" ttl=
64 primary=0 event_usec=18921 rec_type=28 event_subtype=14 rec_type_simple=RNA rec_type_desc="
Additional MAC Detected for Host" mac_address=0:0:0:0:0:0 additional_mac_address=0:7:cf:2d:1b:
3e event_sec=1476351188 sensor=Sensor188 last_seen=1476348954
has_ipv6=1 event_type=1001 legacy_ip_address=0.0.0.0 event_desc="Additional MAC Detected" ttl=
64 primary=0 event_usec=18921 rec_type=28 event_subtype=14 rec_type_simple=RNA rec_type_desc="
Additional MAC Detected for Host" mac_address=0:0:0:0:0:0 additional_mac_address=d8:5d:4c:57:1
b:95 event_sec=1476351188 sensor=Sensor188 last_seen=1476350588
has_ipv6=1 event_type=1001 legacy_ip_address=0.0.0.0 event_desc="Additional MAC Detected" ttl=
47 primary=0 event_usec=20022 rec_type=28 event_subtype=14 rec_type_simple=RNA rec_type_desc="
Additional MAC Detected for Host" mac_address=0:0:0:0:0:0 additional_mac_address=0:1:db:1c:c7:
be event_sec=1476351188 sensor=Sensor188 last_seen=1476350576
[sbs20@centos eStreamer_eNcore-0.15]$
```

3.2 Configuration Options

3.2.1 Essential Configuration

The default configuration file is set up to run out of the box. Following is a brief explanation of each setting in case you wish to customize.

3.2.1.1 Subscription Server

This is the FMC **host** and associated information. If you encounter TLS difficulties and are willing to downgrade, then you can change **tlsVersion** to 1.0.

Note that downgrading the TLS version is useful for debugging and seeing the software work but it is not a recommended long-term strategy. It is recommended instead to fix the root cause.

Figure 7: Subscription Server Screen

```
    "subscription": {
        "servers": [
            {
                "host": "1.2.3.4",
                "port": 8302,
                "pkcs12Filepath": "client.pkcs12",
                "@comment": "Valid values are 1.0 and 1.2",
                "tlsVersion": 1.2
            }
        ], …
```

3.2.1.2 Monitor

The monitor is a separate thread that runs monitoring and maintenance tasks. By default, it runs every two minutes. It will report the number of events received and handled and will check the status of sub-processes. If there have been any problems, the monitor will place the client into an error state and the client will shut itself down.

Figure 8: Monitor Screen

```
    "monitor": {
        "period": 120,
        "velocity": false,
        "bookmark": false,
        "subscribed": true,
        "handled": true
    },
```

3.2.1.3 Start

The eStreamer server expects requests to state their chosen start time. There are broadly three options.

- 0: Return all data from the earliest point available on the FMC
- 1: Return all data from now onwards
- 2: Use a bookmark to pick up where we left off. First run is from 0

Figure 9: Start Screen

```
    "@startComment": "0 for genesis, 1 for now, 2 for bookmark",
    "start": 2,
```

3.2.1.4 Outputters (Output Data Location)

By default, only the Splunk outputter is enabled. It writes its data to a relative file location, but you may want to output the data to a different location. To change this, alter the **stream.uri** property to **file:///absolute/file/path/filename{0}.ext** where {0} is the timestamp placeholder.

Figure 10: Outputters Screen

```
        "outputters": [
            {
                "name": "Splunk default",
                "adapter": "splunk",
                "enabled": true,
                "stream": {
                    "uri": "relfile:///data/splunk/encore.log{0}",
                    "options": {
                        "rotate": true,
                        "maxLogs": 9999
                    }
                }
            },
```

3.2.2 Advanced Configuration Options

Key	Definition
connectTimeout	The duration in seconds the client will wait for a connection to establish before failing.
responseTimeout	The duration in seconds the client will wait for a response before timing out.
monitor.period	The period in seconds between each execution of monitor tasks. Default is 120. Lower numbers are useful for debugging but will create more log traffic.
monitor.velocity	true | false. True will display the speed at which the client is processing records. A positive value means the client is processing events faster than eStreamer is sending them. Negative is slower. Once up to date, this should hover around zero.
monitor.bookmark	true | false. True will show the last bookmark timestamp. This is useful to see how far behind the eNcore client is.
monitor.subscribed	true | false. True will report the total number of events subscribed.
monitor.handled	true | false. True will report the total number of events written to output.
start	0 specifies oldest data available 1 specifies data as of now 2 specifies use of bookmark
logging.level	Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE, and TRACE. Select the level of logging as per your requirement. It is strongly recommended that you do not use anything above INFO for production environments. DEBUG will generate very large log files and TRACE will significantly affect performance.

Key	Definition
logging.format	This describes the format of the log and how they are stored. Default configuration setting for message format is “{date-time}-{name of module}-{level of logging-message}”.
logging.stdOut	true | false. This determines whether log output is also shown in Standard Output.
logging.filepath	This specifies the location of the application log.
maxQueueSize	Maximum number of messages buffered before throttling takes place. It is essentially a buffer size. The larger this number, the longer it will take to shutdown. Default configuration setting is 100. Do not change.
subscription.servers[]	While this is an array, eNcore can only currently support one server. The array is to support the future ability to connect to multiple hosts.
server.host	The IP address of the FMC (eStreamer Server). Default configuration is 1.2.3.4. If you change the host entry after having run eNcore then new cache, bookmark and metadata files will be generated.
server.port	The server port to connect to. Default 8302.
server.pkcs12Filepath	The PKCS12 filepath location. If you change this having already run eNcore, then you must also delete the cached public and private key otherwise eNcore will continue to use those. They are called {host}-{port}_pkcs.cert and {host}-{port}_pkcs.key.
server.tlsVersion	Valid options are 1.0 and 1.2.
subscription.records	Do not change these values.
handler.records.metadata	true | false. If you wish to exclude the output of metadata (since it has no timestamp information) then set this to false.
handler.records.flows	true | false. If you wish to exclude connection flow records then set this to false.
handler.outputters[]	An array of outputter controllers which define the behavior and format of what gets written by eNcore.
outputter.name	This is a human readable name for your convenience. It is unused by the code.
outputter.adapter	Data is read from eStreamer and stored in a structured internal format. The adapter transforms the data to a desired format. Recognized values are: • splunk • json
outputter.enabled	true | false. You can have more than one outputter specified at once. If you wish to disable a specific outputter, set this flag to false. If all outputters are false (or there are no outputters) then it behaves as a sink.
outputter.passthru	true | false. If true then data flowing through bypasses decoding and metadata processing. It is very fast but of limited use. Its primary purpose is for debugging.
outputter.stream.uri	Specify the location where the output will be stored. You can specify a file URI as normal (e.g., file:///absolute/path/to/file) or a relative filepath (relfile:////relative/path/to/file). Only file URLs are supported currently.
outputter.stream.options	File-based streams require additional options.

Key	Definition

option.rotate	true | false. Set if you want log rotation. Default configuration setting for this is true. Please note that eNcore will not delete any old files. If you wish to do that, you will need to script it separately and schedule it.
	Example:
	Call this from a cron job.
	`#!/bin/bash`
	`find /opt/splunk/etc/apps/eStreamer/log/* -mmin +1440 -exec rm {} \;`

option.maxLogs	Specify the size of the log (number of lines). *Default configuration for this is 9999. You can have fewer, larger files (e.g, 5,0000).*

3.3 Execution

Various shell scripts options are available.

During installation and initial setup – or perhaps for debugging purposes it is useful to run the following commands.

```
./encore.sh test
```

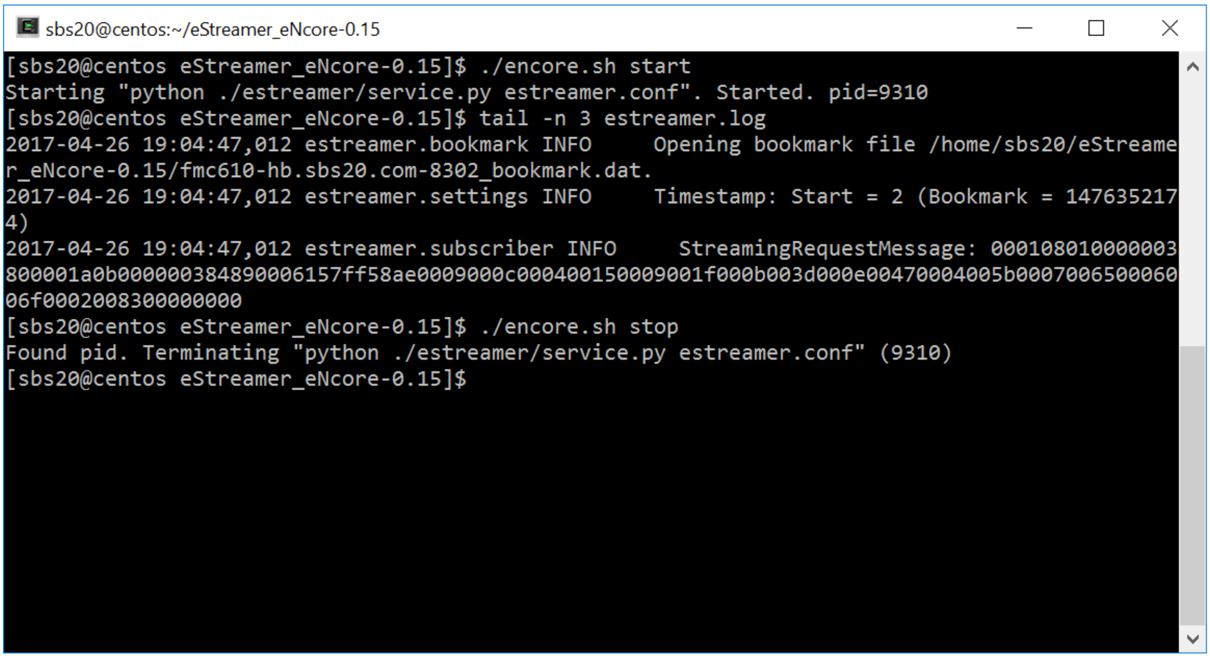
And

```
./encore.sh foreground
```

In all other cases, it is expected that encore will be run in the background, for which the following commands are pertinent.

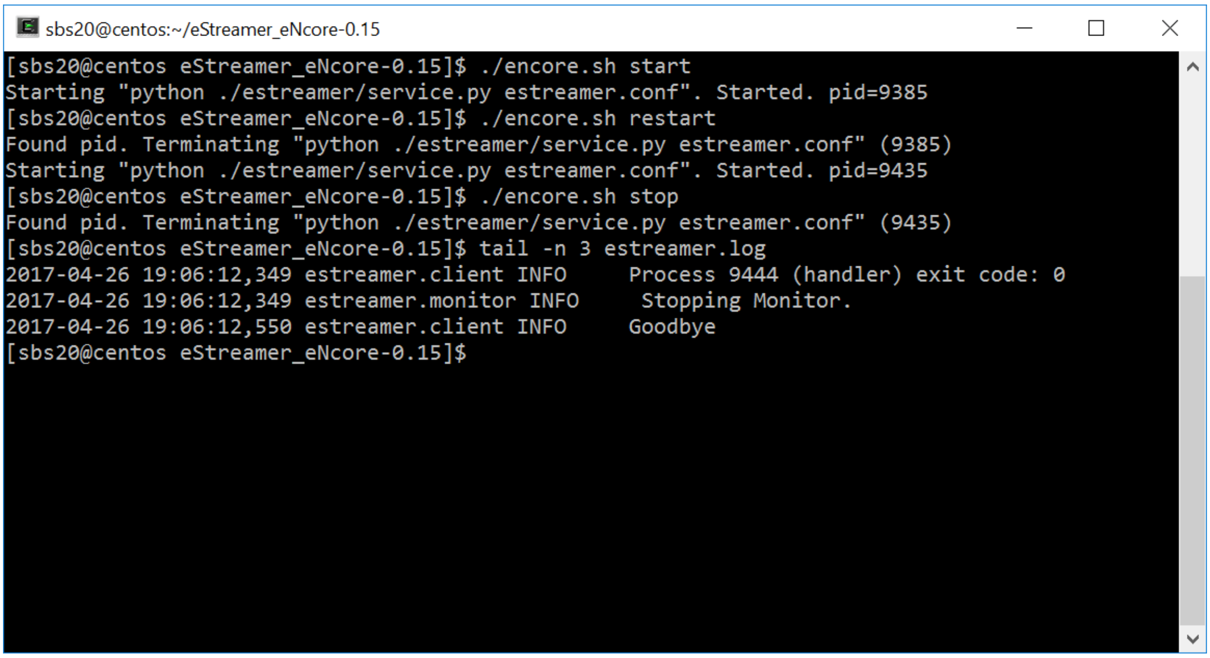
```
./encore.sh start
./encore.sh stop
./encore.sh restart
```

Figure 11: Start, Tail Log, Stop



sbs20@centos:~/eStreamer_eNcore-0.15

```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh start
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9310
[sbs20@centos eStreamer_eNcore-0.15]$ tail -n 3 estreamer.log
2017-04-26 19:04:47,012 estreamer.bookmark INFO     Opening bookmark file /home/sbs20/eStreame
r_eNcore-0.15/fmc610-hb.sbs20.com-8302_bookmark.dat.
2017-04-26 19:04:47,012 estreamer.settings INFO     Timestamp: Start = 2 (Bookmark = 147635217
4)
2017-04-26 19:04:47,012 estreamer.subscriber INFO     StreamingRequestMessage: 000108010000003
800001a0b000000384890006157ff58ae0009000c000400150009001f000b003d000e00470004005b0007006500060
06f0002008300000000
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh stop
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9310)
[sbs20@centos eStreamer_eNcore-0.15]$
```



sbs20@centos:~/eStreamer_eNcore-0.15

```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh start
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9385
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh restart
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9385)
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9435
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh stop
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9435)
[sbs20@centos eStreamer_eNcore-0.15]$ tail -n 3 estreamer.log
2017-04-26 19:06:12,349 estreamer.client INFO     Process 9444 (handler) exit code: 0
2017-04-26 19:06:12,349 estreamer.monitor INFO     Stopping Monitor.
2017-04-26 19:06:12,550 estreamer.client INFO     Goodbye
[sbs20@centos eStreamer_eNcore-0.15]$
```

3.4 Logging

By default, eNcore will output an **estreamer.log** application log in its working directory with a log level of INFO. The format of the log file can be adjusted using the **logging.format** configuration setting. The level can also be adjusted. It is recommended that the default settings are left in place for production execution.

4 Troubleshooting and questions

4.1 Error messages

As far as possible, eNcore has been engineered to provide meaningful error messages. Below is an example error message.

Figure 12: Example Error Message

```
The eStreamer service has closed the connection. There are a number of possible
causes which may show above in the error log.

If you see no errors then this could be that
 * the server is shutting down
 * there has been a client authentication failure (please check that your outbound
IP address matches that associated with your certificate - note that if your device
is subject to NAT then the certificate IP must match the upstream NAT IP)
 * there is a problem with the server. If you are running FMC v6.0, you may need to
install "Sourcefire 3D Defense Center S3 Hotfix AZ 6.1.0.3-1")
```

If you encounter errors that do not make sense or require further explanation, then please contact support so that we can fix the problem and improve the error messages.

4.2 Frequently Asked Questions

Can I output my data to a different server?

Yes. Currently eNcore only writes to the filesystem, but you could mount an NFS or SMB share and specify its path as above. This may impact performance.

Can I run more than one instance?

Yes. Although, currently, the **encore.sh** shell script only supports one instance. The underlying Python program prefixes temporary files (e.g., metadata, certificates, bookmarks) with the host and port. You will also need to update the outputter locations (e.g., [Splunk] … directory = splunk) in order to avoid data collision.

Can I connect to more than one FMC?

Currently, not within a single instance. However, you can configure multiple instances as above.

Can eNcore de-duplicate data to keep my SIEM costs lower?

Not today. It is on the roadmap.

Can I run two instances of eNcore in a HA pair?

Yes and no. It is technically possible to run two side-by-side, but they will be completely ignorant of each other and output double the data. It may be preferable to run them in a hot-stand-by configuration where the primary client's state and configuration data is regularly copied to the secondary client. The state and configuration data in question is estreamer.conf; x.x.x.x-port_bookmark.dat; x.x.x.x-port_cache.dat; x.x.x.x-port_pkcs.cert; x.x.x.x-port_pkcs.key; x.x.x.x-port_status.dat

Can I increase the logging granularity?

Yes, change **logging.level** in the conf file. Please note that while it is possible to increase this level to VERBOSE, the performance impact will be crippling. DEBUG may be useful but slow. We strongly recommend not going above INFO for standard production execution.

5 Cisco Support

Support is provided by Cisco TAC.

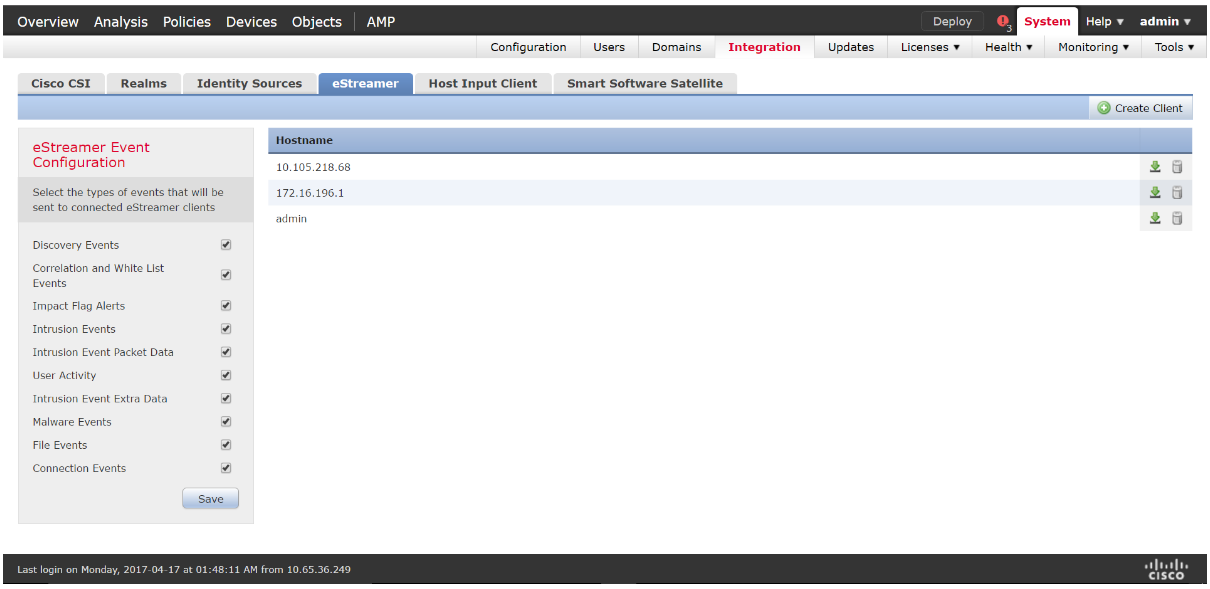
6 Appendix A:

6.1 FMC eStreamer Certificate Creation

Steps to generate an eStreamer client certificate are as follows.

In the FMC 6.x GUI, navigate to **System > Integration > eStreamer**

Figure 13: FMC eStreamer Certificate Creation



eStreamer Event Configuration

Select the types of events that will be sent to connected eStreamer clients

Discovery Events	☑
Correlation and White List Events	☑
Impact Flag Alerts	☑
Intrusion Events	☑
Intrusion Event Packet Data	☑
User Activity	☑
Intrusion Event Extra Data	☑
Malware Events	☑
File Events	☑
Connection Events	☑

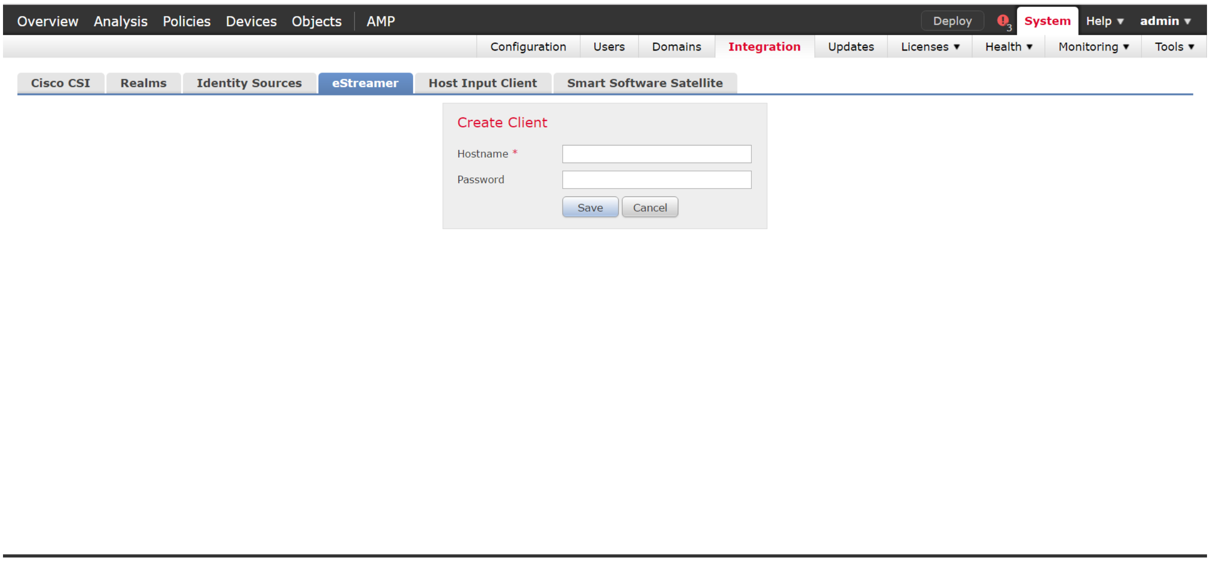
Hostname
10.105.218.68
172.16.196.1
admin

Click **Create Client**. Provide the Hostname and password.

Note: This should be the IP of the client, which will be collecting the event data from the FMC. This password will be required when you first execute eStreamer eNcore.

Please note that the IP address you enter here must be the IP address of the eStreamer-eNcore client **from the perspective of the FMC**. In other words, if the client is behind a NAT device, then the IP address must be that of the upstream NAT interface.

Figure 14: Create Client Hostname and Password Screen



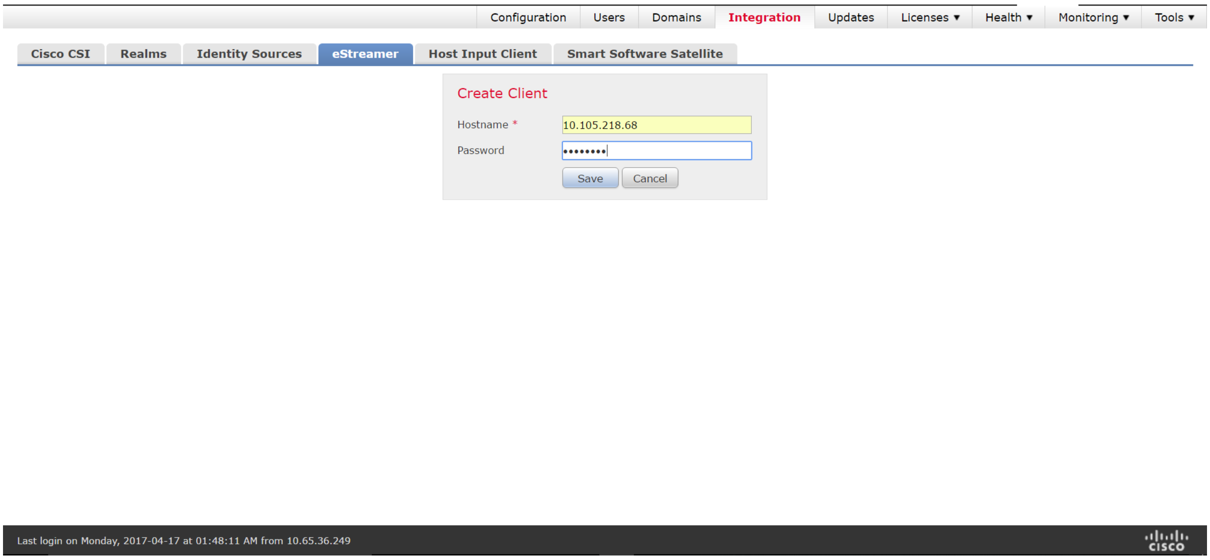
Create Client

Hostname *

Password

Click **Save**.

Figure 15: Create Client Save Screen



Create Client

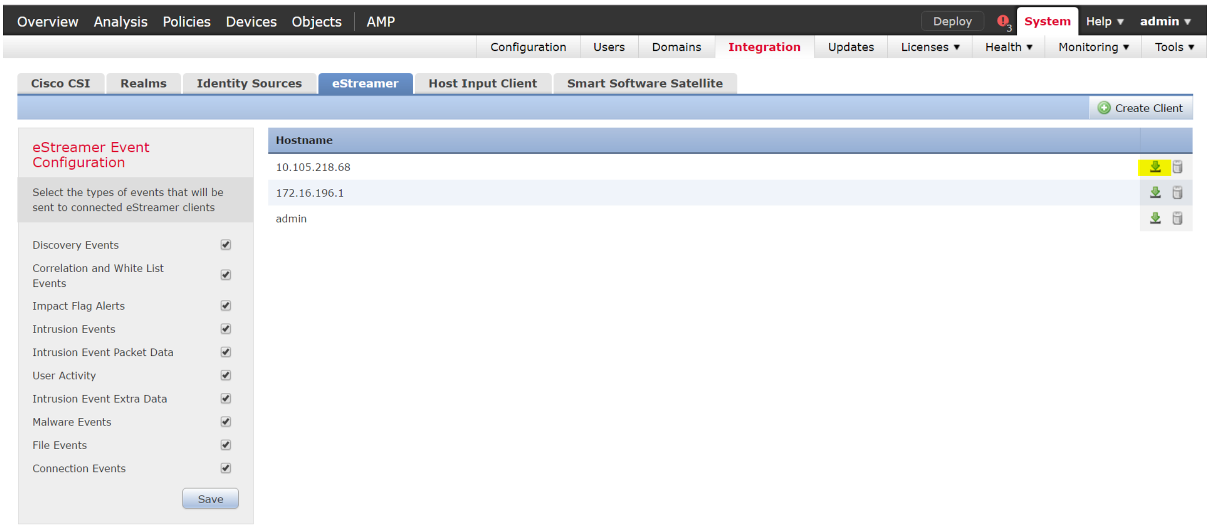
Hostname * 10.105.218.68

Password ••••••••

Last login on Monday, 2017-04-17 at 01:48:11 AM from 10.65.36.249

Download the pkcs12 file.

Figure 16: Download Screen



eStreamer Event Configuration

Select the types of events that will be sent to connected eStreamer clients

Discovery Events	☑
Correlation and White List Events	☑
Impact Flag Alerts	☑
Intrusion Events	☑
Intrusion Event Packet Data	☑
User Activity	☑
Intrusion Event Extra Data	☑
Malware Events	☑
File Events	☑
Connection Events	☑

Hostname
10.105.218.68
172.16.196.1
admin

Copy the pkcs12 file to the desired location in the target device. By default, eStreamer-eNcore will look for **/path/eStreamer_eNcore.X.YY/client.pkcs12**. If you wish to use a different filename, then you must edit the **estreamer.conf** file.

6.2 Example Configuration File

Figure 17: Example Configuration File

```
{
    "connectTimeout":  10,
    "responseTimeout":  10,

    "@startComment": "0 for genesis, 1 for now, 2 for bookmark",
    "start": 2,

    "monitor": {
        "period": 120,
        "velocity": false,
        "bookmark": false,
        "subscribed": true,
        "handled": true
    },

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

    "@queueComment": [
        "Maximum number of messages buffered before throttling takes place. The
more powerful",
        "your CPU and more RAM you have, the larger this number can be. It's
essentially a",
        "buffer size. Beyond a certain size you won't see any performance gain and
it will",
        "just take longer to stop"
    ],
    "maxQueueSize": 100,

    "subscription": {
        "servers": [
            {
                "host": "1.2.3.4",
                "port": 8302,
                "pkcs12Filepath": "client.pkcs12",
                "@comment": "Valid values are 1.0 and 1.2",
                "tlsVersion": 1.2
            }
        ],

        "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[]"
            ],
            "packetData": true,
            "extended": true,
            "metadata": true,
            "eventExtraData": true,
            "impactEventAlerts": true,
            "intrusion": true,
            "archiveTimestamps": true
        }
    },

    "handler": {
        "records": {
            "core": true,
            "metadata": true,
            "flows": true,
            "packets": true,
            "intrusion": true,
            "rua": true,
            "rna": true,

            "@includeComment": "These records will be included regardless of 
above",
            "include": [],

            "@excludeComment": [
                "These records will be excluded regardless of above (overrides 
'include')",
                "e.g. to exclude flow and IPS events use [ 71, 400 ]"
            ],
            "exclude": []
        },

        "@comment": "If you disable all outputters it behaves as a sink",
        "outputters": [
            {
                "name": "Splunk default",
                "adapter": "splunk",
                "enabled": true,
                "stream": {
                    "uri": "relfile:///data/splunk/encore.log{0}",
                    "options": {
                        "rotate": true,
                        "maxLogs": 9999
                    }
                }
            },
            {
                "name": "JSON",
                "adapter": "json",
                "enabled": false,
                "stream": {
                    "uri": "relfile:///data/json/log{0}.json",
                    "options": {
                        "rotate": true,
                        "maxLogs": 9999
                    }
                }
            }
        ]
    }
}
```

Trademarks and Disclaimers

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THIRD PARTY SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THIRD PARTY SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.