



# eStreamer eNcore for Splunk Operations Guide v4.0.9

First Published: June 1, 2017

Last Updated: September 15, 2020

## Table of Contents

Table of Contents.....	2
About This eStreamer eNcore for Splunk Operations Guide v4.0.9.....	4
Revision History .....	4
Conventions .....	4
1 Introduction .....	6
1.1 Document Purpose .....	6
1.2 Background .....	6
1.3 Application Summary.....	6
1.3.1 Cisco eStreamer eNcore add-on for Splunk (TA-eStreamer).....	6
1.3.2 Cisco eStreamer eNcore Dashboard for Splunk (eStreamer Dashboard).....	6
2 Prerequisites .....	7
3 Installation .....	7
3.1 Installing the eNcore add-on for Splunk (TA-eStreamer) .....	7
3.2 Installing the eNcore Dashboard for Splunk (eStreamer Dashboard) .....	7
4 eNcore Add-on for Splunk Setup Configuration .....	8
4.1 Enable Data Inputs.....	8
4.2 Enable Scripts.....	8
4.3 eNcore Add-on Setup Page Configuration .....	9
5 Operation .....	16
6 Configuration Options.....	17
6.1 The Monitor .....	17
6.2 Start Time.....	18
6.3 Outputters.....	18
6.4 Performance and the workerProcesses Option.....	18
6.5 Batch Size .....	19
7 Troubleshooting and questions .....	20
7.1 Error messages.....	20
7.2 Frequently Asked Questions .....	21
8 Cisco Support .....	21
9 Appendix A: Client Certificate Creation and Installation .....	22

Table of Contents

9.1 Firepower Management Center eStreamer Client Certificate Creation .....	22
10 Appendix B: List of Configuration Options .....	25
11 Appendix C: Example Configuration File .....	28
Trademarks and Disclaimers .....	31

## About This eStreamer eNcore for Splunk Operations Guide v4.0.9

Author	Seyed Khadem (skhademd)
Change Authority	Cisco Systems Advanced Services, Security & Collaboration IDT, Implementation Americas
Content ID	
Project ID	868408

## Revision History

Revision	Date	Name or User ID	Comments
1.0	06/01/2017	Michelle Jenkins	Initial Release
3.0	08/25/2017	Sam Strachan	
3.5	10/15/2018	Richard Clendenning	Updated for v3.5
4.0.9	9/15/2020	Seyed Khadem	Updated for Splunk 8.0

## Conventions

This document uses the following conventions.

Convention	Indication
<b>bold font</b>	Commands and keywords and user-entered text appear in bold font.
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	Elements in square brackets are optional.
{x   y   z}	Required alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
String	A non-quoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
<code>courier font</code>	Terminal sessions and information the system displays appear in <code>courier font</code> .
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.

Conventions

Convention	Indication
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

**Note:** Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

**Caution:** Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

**Warning:** IMPORTANT SAFETY INSTRUCTIONS

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

**Regulatory:** Provided for additional information and to comply with regulatory and customer requirements.

## 1 Introduction

### 1.1 Document Purpose

This document outlines the background and usage of the eStreamer eNcore for Splunk client to assist users with installation and execution.

### 1.2 Background

The Cisco Event Streamer (eStreamer) allows users to stream system intrusion, discovery, and connection data from a Firepower Management Center or managed device (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).

### 1.3 Application Summary

eStreamer eNcore is a multi-platform, multi-process Python application that is compatible with Firepower Management Center versions 6.0 and above.

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 Security Information and Event Management tools (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 but is not supported.

#### 1.3.1 Cisco eStreamer eNcore add-on for Splunk (TA-eStreamer)

The eStreamer eNcore add-on for Splunk is a technology add-on that contains the core eNcore eStreamer client code as well as:

- Data inputs (inputs.conf) for data, logs and status
- Parsing hints (props.conf)
- An extension which allows eNcore to live and die with Splunk

Note: The eNcore for Splunk add-on is not supported on Splunk for Windows.

#### 1.3.2 Cisco eStreamer eNcore Dashboard for Splunk (eStreamer Dashboard)

This is an application which includes the same user interface elements as the old Cisco eStreamer for Splunk app (<https://splunkbase.splunk.com/app/1629/>). The application contains no code or collector elements, however. It is simply a UI application with some pre-defined searches, macros, eventtypes and workflow actions.

## 2 Prerequisites

## 2 Prerequisites

The eNcore add-on for Splunk and the eNcore Dashboard for Splunk do not require any special prerequisites. They are downloadable from Splunkbase and they install in the search head in the same way as other add-ons and applications.

The eNcore add-on for Splunk requires Python 3.6+ and openssl, in the latest Splunk 8.0 release Python3 is included, however the python mods for openssl are not included, which require additional configuration steps outlined in this update. If the Splunk install was customized and is missing one or both of these components, they will need to be installed for the add-on to function.

## 3 Installation

Note: The eNcore for Splunk add-on is not supported on Splunk for Windows.

### 3.1 Installing the eNcore add-on for Splunk (TA-eStreamer)

To install the eNcore add-on for Splunk, either:

- a. Download the add-on from <http://apps.splunk.com/app/3662> and use the "Install app from file" feature in Splunk to then upload and install the add-on, or
- b. Use the "Browse more apps" feature in Splunk and **search for "eNcore"**, then look for Cisco eStreamer Add-on for Splunk in the search results and click Install for that add-on.

You must install a PKCS12 certificate for your Splunk server, which allows the Firepower Management Center to authenticate the identity of the add-on when the eNcore client contacts the Firepower Management Center and establishes a secure tunnel. You create the PKCS12 certificate on the Firepower Management Center, download it, and copy it to this location on the Splunk server (renaming it client.pkcs12):

```
$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/client.pkcs12
```

For more information on creating a PKCS12 certificate and copying it to the Splunk server, see Appendix A.

### 3.2 Installing the eNcore Dashboard for Splunk (eStreamer Dashboard)

To install the eNcore Dashboard for Splunk, either:

- a. Download the app from <http://apps.splunk.com/app/3663> and use the "Install app from file" feature in Splunk to then upload and install the add-on, or
- b. Use the "Browse more apps" feature in Splunk and **search for "eNcore"**, then look for Cisco Firepower eNcore App for Splunk in the search results and click Install for that app.

## 4 eNcore Add-on for Splunk Setup Configuration

### 4.1 Enable Data Inputs

The eNcore add-on for Splunk writes events to log files in the installation's data directory. Splunk must be configured with a Data Input that reads the events from this directory.

To do this, navigate to Settings > Data Inputs > Files & Directories and enable the data input with the path `$(SPLUNK_HOME)/etc/apps/TA-eStreamer/data` and Source type `cisco:estreamer:data`).



**Files & directories**  
Data inputs > Files & directories

Showing 1-10 of 10 items

filter

25 per page

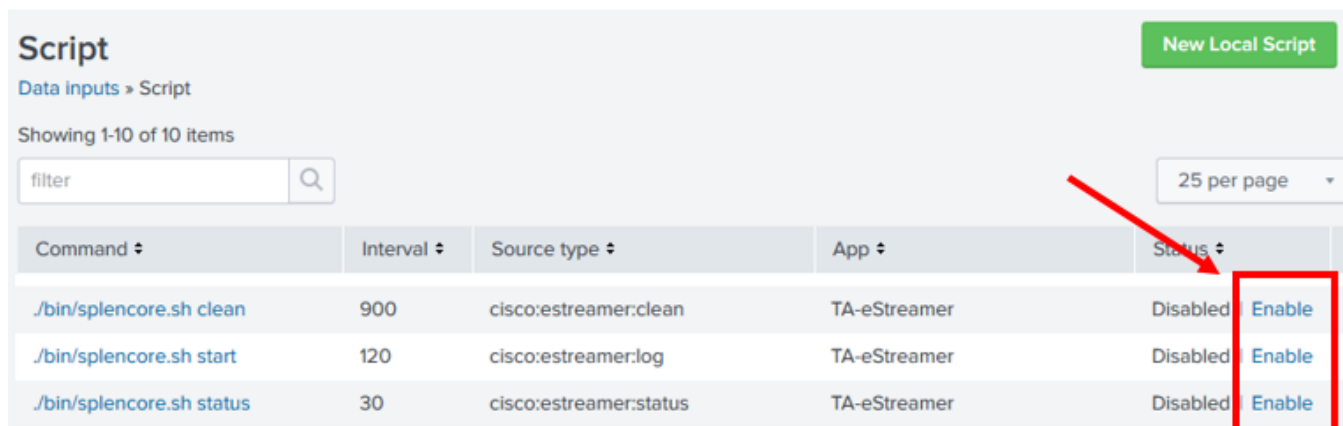
Full path to your data	Set host	Source type	Index	Number of files	App	Status
\$(SPLUNK_HOME)/etc/apps/TA-eStreamer/data	Constant Value	cisco:estreamer:data	default		TA-eStreamer	Disabled <b>Enable</b>

### 4.2 Enable Scripts

The eNcore add-on for Splunk has three scripts that perform important operations:

- `cisco:estreamer:clean` – this script has no output but is used to delete data files older than 12 hours
- `cisco:estreamer:log` – this script uses the stdout of eNcore to take program log data. This becomes very useful where things are not going to plan. More importantly, it is the script which starts the eStreamer eNcore process.
- `cisco:estreamer:status` – this script runs periodically to maintain a clear status of whether the program is running

These scripts must be enabled by navigating to Settings > Data Inputs > Scripts and clicking enable for the three TA-eStreamer scripts.



**Script**  
Data inputs > Script

Showing 1-10 of 10 items

filter

25 per page

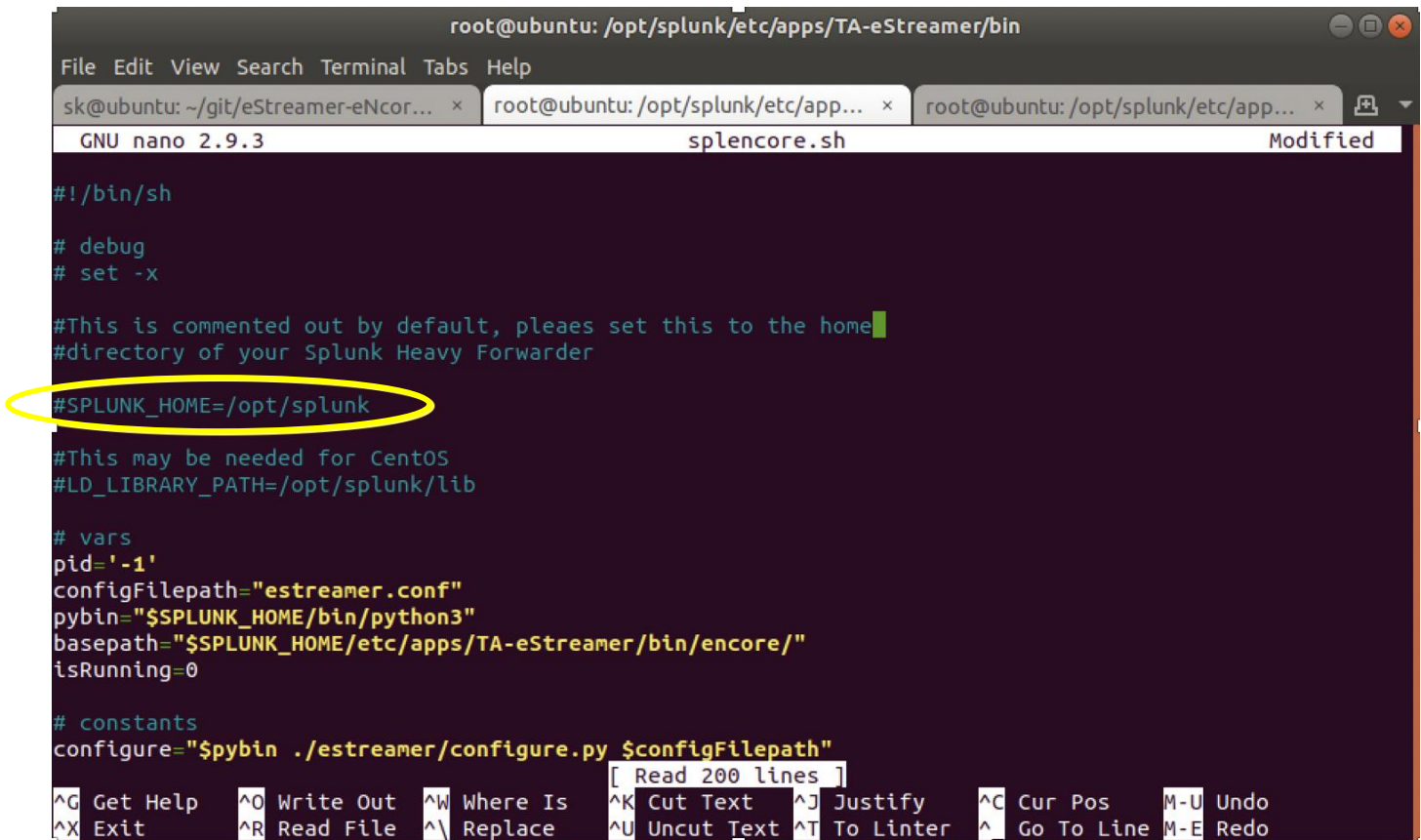
Command	Interval	Source type	App	Status
./bin/splencore.sh clean	900	cisco:estreamer:clean	TA-eStreamer	Disabled <b>Enable</b>
./bin/splencore.sh start	120	cisco:estreamer:log	TA-eStreamer	Disabled <b>Enable</b>
./bin/splencore.sh status	30	cisco:estreamer:status	TA-eStreamer	Disabled <b>Enable</b>



### 4.3 eNcore Add-on Setup Configuration

Navigate to the TA-eStreamer bin directory, located in `$SPLUNK_HOME/etc/apps/TA-eStreamer/bin`, where `$SPLUNK_HOME` represents the home directory of your Splunk Heavy Forwarder installation.

You will want to set the hompath of your `SPLUNK_HOME` install variable, this can be done on the command line by editing the startup script, remove the commented out line for the `SPLUNK_HOME` variable and set it to your install path, in Linux this is typically `/opt/splunk`,



```
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin
File Edit View Search Terminal Tabs Help
sk@ubuntu: ~/git/eStreamer-eNcor... x root@ubuntu: /opt/splunk/etc/app... x root@ubuntu: /opt/splunk/etc/app... x
GNU nano 2.9.3 splencore.sh Modified

#!/bin/sh

# debug
# set -x

#This is commented out by default, please set this to the home
#directory of your Splunk Heavy Forwarder
#SPLUNK_HOME=/opt/splunk
#This may be needed for CentOS
#LD_LIBRARY_PATH=/opt/splunk/lib

# vars
pid='-1'
configfilepath="estreamer.conf"
pybin="$SPLUNK_HOME/bin/python3"
basepath="$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/"
isRunning=0

# constants
configure="$pybin ./estreamer/configure.py $configfilepath"

[ Read 200 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line M-E Redo
```

In CentOS you may see the error,

```
**/opt/splunk/bin/openssl: error while loading shared libraries: libssl.so.1.0.0: cannot open shared object file: No such file or directory**
```

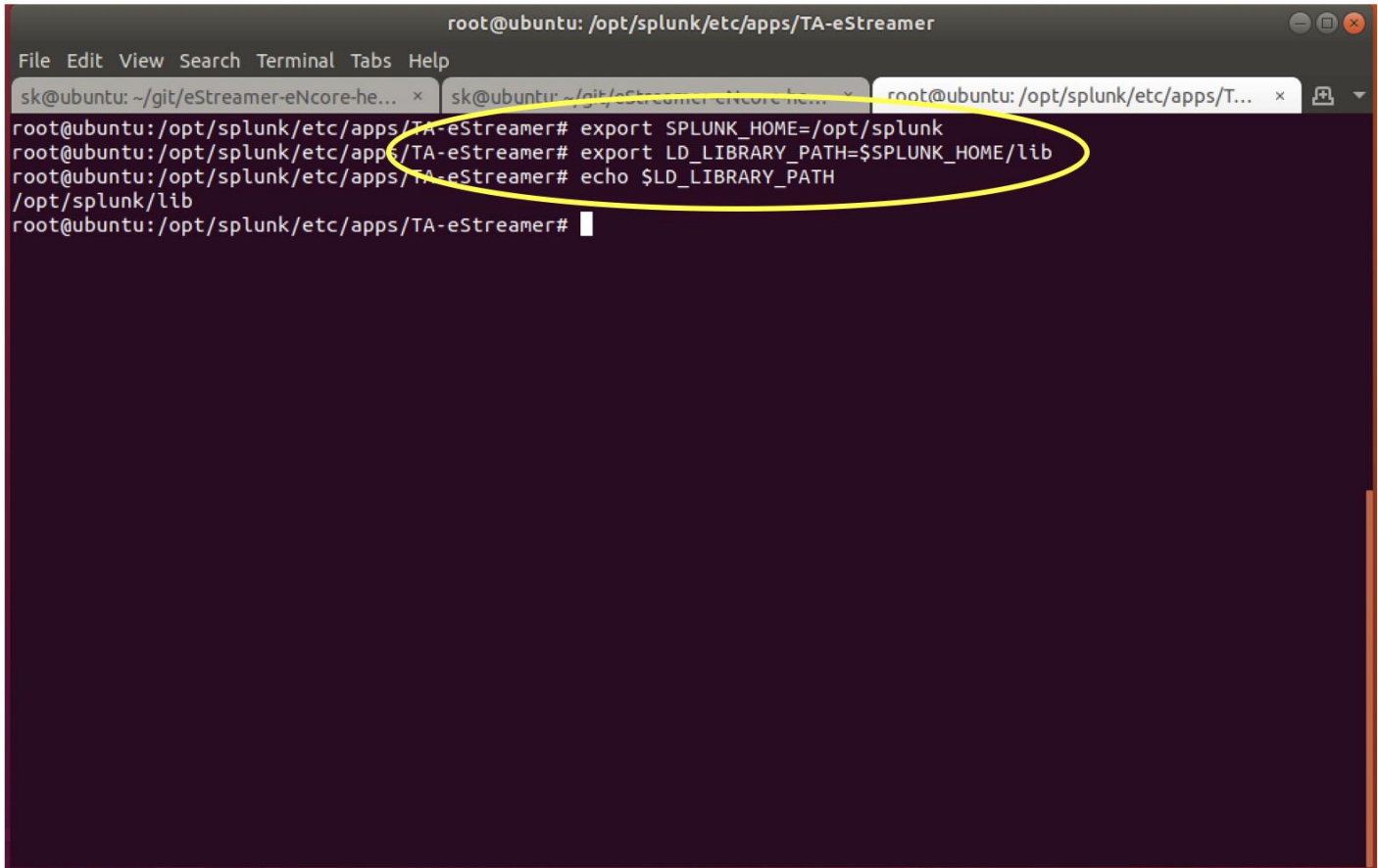
#### 4 eNcore Add-on for Splunk Setup Configuration

To resolve this you need to add one additional setup variable for the Splunk Lib path, it is commented out in the script, you will need to run the following command prior to executing the setup script.

```
export LD_LIBRARY_PATH=$SPLUNK_HOME/lib
```

example: `LD_LIBRARY_PATH=/opt/splunk/lib`

<https://community.splunk.com/t5/Archive/Splunk-OPEN-SSL-unable-to-load-library-files/m-p/229021>

A terminal window screenshot showing the execution of environment variable setup commands. The terminal title is "root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer". The terminal content shows the following commands and output:

```
root@ubuntu:/opt/splunk/etc/apps/TA-eStreamer# export SPLUNK_HOME=/opt/splunk
root@ubuntu:/opt/splunk/etc/apps/TA-eStreamer# export LD_LIBRARY_PATH=$SPLUNK_HOME/lib
root@ubuntu:/opt/splunk/etc/apps/TA-eStreamer# echo $LD_LIBRARY_PATH
/opt/splunk/lib
root@ubuntu:/opt/splunk/etc/apps/TA-eStreamer#
```

The command `export LD_LIBRARY_PATH=$SPLUNK_HOME/lib` and its output `/opt/splunk/lib` are circled in yellow.

Next we'll need to modify the `estreamer.conf` file, `$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore`. Set the "host" field to the IP or hostname of your FMC, see section 6.7 for more details.

```
    "intrusion": true,  
    "metadata": true,  
    "packetData": true  
  },  
  "servers": [  
    {  
      "host": "64.100.13.164",  
      "pkcs12Filepath": "client.pkcs12",  
      "port": 8302,  
      "tls@comment": "Valid values are 1.0 and 1.2",  
      "tlsVersion": 1.2  
    }  
  ]  
},  
"workerProcesses": 4  
}
```

Next: Edit the settings: false, change it to true

```

root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin
File Edit View Search Terminal Tabs Help
sk@ubuntu: ~/git/eStreamer-eNcor... x root@ubuntu: /opt/splunk/etc/app... x root@ubuntu: /opt/splunk/etc/app... x
{
  "conditions": [
    "splunk"
  ],
  "connectTimeout": 10,
  "enabled": true,
  "handler": {
    "output@comment": "If you disable all outputters it behaves as a sink",
    "outputters": [
      {
        "adapter": "splunk",
        "enabled": true,
        "stream": {
          "options": {
            "maxLogs": 10000,
            "rotate": true
          },
          "uri": "relfile:///../../data/encore.{0}.log"
        }
      }
    ]
  },
  "records": {
    "connections": false,
    "core": true,
    "excl@comment": [
      "These records will be excluded regardless of above (overrides 'include')",

```

Run the `./splncore.sh` test command, and enter the FMC IP Address

```

root@ubuntu:/opt/splunk/etc/apps/TA-eStreamer/bin# ./splncore.sh test
Please Enter the FMC IP:64.100.13.164

```

Copy the two commands and run those each individually, enter the pkcs password when prompted

Command #1 : **Generate the pkcs certificate using Splunk's built-in open ssl**, the test utility will generate this command for you provided. This command can also be run manually using the following format:

```

$SPLUNK_HOME/apps/TA-eStreamer/bin# $SPLUNK_HOME/bin /openssl pkcs12 -in $SPLUNK_HOME /etc/apps/TA-
eStreamer/bin/encore/" client.pkcs12" -nocerts -nodes -out "$SPLUNK_HOME/etc/apps/TA-
eStreamer/bin/encore/$FMC_IP-8302_pkcs.key"

```

*Example:* `/opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/" client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.cert"`

```
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin
File Edit View Search Terminal Tabs Help
sk@ubuntu: ~/git/eStreamer-eNcor... x root@ubuntu: /opt/splunk/etc/app... x root@ubuntu: /opt/splunk/etc/app... x
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# ls
client.pkcs12 configure_handler.py configure.sh encore splencore.sh
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# bash splencore.sh test
Please Enter the FMC IP:64.100.13.164
Please run the following command: /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.cert"
Please run the following command: /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.key"
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.cert"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.key"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin#
```

#### Command #2

Generate the key file, the command will be generated for you on the startup script, but it can also be run manually using the following format,

*Example:* /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302\_pkcs.key"

```
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin
File Edit View Search Terminal Tabs Help
sk@ubuntu: ~/git/eStreamer-eNcor... x root@ubuntu: /opt/splunk/etc/app... x root@ubuntu: /opt/splunk/etc/app... x
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# ls
client.pkcs12 configure_handler.py configure.sh encore splencore.sh
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# bash splencore.sh test
Please Enter the FMC IP:64.100.13.164
Please run the following command: /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.cert"
Please run the following command: /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.key"
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.cert"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/64.100.13.164-8302_pkcs.key"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin#
```

Run the test command to ensure valid SSL handshake is in place

```
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin
File Edit View Search Terminal Tabs Help
sk@ubuntu: ~/git/eStreamer-eNcor... x root@ubuntu: /opt/splunk/etc/app... x root@ubuntu: /opt/splunk/etc/app... x
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/
apps/TA-eStreamer/bin/encore/"client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/
bin/encore/64.100.13.164-8302_pkcs.cert"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# /opt/splunk/bin/openssl pkcs12 -in /opt/splunk/etc/
apps/TA-eStreamer/bin/encore/"client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/b
in/encore/64.100.13.164-8302_pkcs.key"
Enter Import Password:
MAC verified OK
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin# ./splencore.sh test
Please Enter the FMC IP:64.100.13.164
2020-09-15T09:48:43.398890 Diagnostics INFO Checking that configFilepath (estreamer.conf) exists
2020-09-15 09:48:43,413 Diagnostics INFO Check certificate
2020-09-15 09:48:43,413 Diagnostics INFO Creating connection
2020-09-15 09:48:43,413 Connection INFO Connecting to 64.100.13.164:8302
2020-09-15 09:48:43,413 Connection INFO Using TLS v1.2
2020-09-15 09:48:43,511 Diagnostics INFO Creating request message
2020-09-15 09:48:43,511 Diagnostics INFO Request message=b'0001000200000008ffffffff48900061'
2020-09-15 09:48:43,512 Diagnostics INFO Sending request message
2020-09-15 09:48:43,512 Diagnostics INFO Receiving response message
2020-09-15 09:48:43,534 Diagnostics INFO Response message=b'gAN9cQAoWAcAAAB2ZXJzaW9ucQFLAVgLAAAAb
WVzc2FnZVR5cGVxAK0DCFgGAAAAbGVuZ3RocQNLMFgEAAAAZGF0YXEEQzAAABOJAAAACAAAAAAAAAAAAAAATiAAAAgAAAAAAAAAAAA
AGgsAAAAIAAAAAAAAAAABxBXUu'
2020-09-15 09:48:43,535 Diagnostics INFO Streaming info response
2020-09-15 09:48:43,535 Diagnostics INFO Connection successful
root@ubuntu: /opt/splunk/etc/apps/TA-eStreamer/bin#
```

## 5 Operation

Once you have fully configured all items as described in section 4, the eNcore add-on for Splunk is started by checking the Is enabled? check box on the add-on setup page and clicking Save as described in section 4.3.

Following execution, the operation of the add-on can be monitored by searching for status, log, and data events:

- To check the status, search for `sourcetype=" cisco:estreamer:status"`
- To check more detailed log output, search for `sourcetype=" cisco:estreamer:log"`
- To look for eStreamer data, search for `sourcetype=" cisco:estreamer:data"`

For further analysis of the Firepower events consider installing either:

- Cisco Firepower App for Splunk



## 6 Configuration Options

This default configuration provided by the eNcore for Splunk add-on in the `estreamer.conf` file provides an optimal configuration for many deployments. However, some options that may need to be adjusted by the user in certain circumstances. This section provides details regarding several of these options. The complete list of settings is provided in Appendix B.

### 6.1 The Monitor

The monitor is a separate thread that runs monitoring and maintenance tasks. By default, it runs every two minutes. It writes the number of events handled to the eNcore log and checks the status of subprocesses. If there are any problems with subprocesses, the monitor places the client into an error state and the client shuts itself down.

An example of messages written to the log by the monitor thread is shown below.

```
2018-08-30 05:09:15,026 Monitor      INFO      Running. 2296400 handled; average rate 578.86 ev/sec;
2018-08-30 05:11:15,684 Monitor      INFO      Running. 2296400 handled; average rate 561.87 ev/sec;
2018-08-30 05:13:15,384 Monitor      INFO      Running. 2296400 handled; average rate 545.86 ev/sec;
```

Several aspects of the log messages can be configured in the monitor section of the `estreamer.conf` configuration file, which is located in `$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/estreamer.conf`.

The aspects that can be configured are:

- `period`: The interval in seconds at which the monitor performs its check of subprocesses and writes status messages to the log.
- `bookmark`: If true, the bookmark (the time of the latest event in Unix time format) is included in each monitor log message
- `handled`: If true, the number of events that eNcore has handled since being started.
- `details`: If true, then in addition to the brief status message that the monitor writes to the log, it will also write a detailed message containing many status items related to the operation of the eNcore client.

An example of the configuration of these parameters in the `estreamer.conf` file is shown here:

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

## 6.2 Start Time

The eStreamer server expects the client request to state the start time, which specifies that the Firepower Management Center should only send events that occurred after the start time. There are three options:

- 0: Send all events from the earliest point available on the Firepower Management Center
- 1: Send all events that occur after receiving the client request
- 2: Use a bookmark to pick up where we left off. First run is from 0

An example of the start configuration in the estreamer.conf file is shown here:

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

## 6.3 Outputters

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.

An example of the outputters configuration in the estreamer.conf file is shown here:

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

## 6.4 Performance Tuning

The performance of the eNcore for Splunk add-on has been improved in version 4.0.9 with the addition of multi-processing. By default, four worker processes operate on the incoming messages to achieve higher throughput. While multiple processes can provide significant performance gains, these gains are highly dependent on the platform because for each platform, the processing bottlenecks may be different. Multiple processes also require additional overhead for managing task distribution, so that increasing the number of processes could actually decrease the performance on platforms with a low number of CPU cores.

## 6 Configuration Options

The number of worker processes is configurable through the `workerProcesses` parameter in the `estreamer.conf` configuration file. The number can be set from 1 to 12. Generally, the more capable the platform (i.e., more CPU cores, better I/O, etc.), more throughput is achieved through a higher number of worker processes. However, the only reliable approach is to test performance with various settings such as 1, 2, 4, 8, and 12, and in many cases the best performance may be gained with just one worker process because no process marshalling is required.

One scenario for testing is to:

1. Disable the add-on's Data Input in Splunk, because the same events will be requested multiple times during the testing.
2. Configure a set number of `workerProcesses` (such as 8) and then start eNcore with a `start` parameter of 0 (for genesis) or at least an old start time.
3. Request connection events from the Firepower Management Center (or in some other way request the Firepower Management Center send millions of backlogged events).
4. Observe the event rate reported by the monitor process in the `estreamer.log` file.
5. Repeat the test with a different number of `workerProcesses`.
6. When the optimal number has been determined, set the `workerProcesses` to that number and enable the add-on's Data Input to resume production operations.

An example of the `workerProcesses` configuration in the `estreamer.conf` file is shown here:

```
"workerProcesses": 12
```

### 6.5 Batch Size

The eNcore for Splunk add-on also attempts to improve performance by batching received events and only writing them to output when the threshold for the batch has been reached. The default batch size is 100 events.

If the event rate is very low, then a batch size of 100 events could cause an unwanted delay in the appearance of events in Splunk. For example, if intrusion events are the only events that are handled and the intrusion event rate averages 100 events per hour, then the first event in a batch will often be delayed an hour or more while the batch completes and is written to disk. To reduce such delays the `batchSize` can be set to a lower value, or to eliminate them entirely, the `batchSize` can be set to 1.

The disadvantage of setting `batchSize` to 1, is that in high-throughput environments, the overall event rate will be lower.

An example of the `batchSize` configuration in the `estreamer.conf` file is shown here:

```
"batchSize": 50
```

## 6.6 Persisting a connection

It can be advantage to persist the client indefinitely to listen for a stream of data from the FMC, or have eNcore automatically restart after a Splunk outage, this can be achieved by setting the following configuration value

```
"alwaysAttemptToContinue": true
```

## 6.7 Host

By default a generic place holder is defined within the estreamer.conf, you will want to change this to the IP or HOST name of your FMC. As of this writing only ipv4 addresses are supported

**“host”: “1.2.3.4”**

```
"servers": [  
  {  
    "host": "64.100.13.164",  
    "pkcs12Filepath": "client.pkcs12",  
    "port": 8302,  
    "tls@comment": "Valid values are 1.0 and 1.2",  
    "tlsVersion": 1.2  
  }  
]
```

# 7 Troubleshooting

## 7.1 Error messages

The eNcore for Splunk add-on is engineered to provide meaningful error messages. Below is an 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 Firepower Management Center v6.0, you may need to install " Sourcefire 3D Defense Center S3 Hotfix AZ 6.1.0.3-1")

## 8 Cisco Support

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.

### 7.2 Frequently Asked Questions

Can I connect to more than one Firepower Management Center?

Currently, not within a single instance.

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 are comprised of:

- 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, by changing logging.level in the estreamer.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.

## 8 Cisco Support

Support is provided by Cisco TAC.

## 9 Appendix A: Client Certificate Creation and Installation

### 9.1 Firepower Management Center eStreamer Client Certificate Creation

The steps to generate an eStreamer client certificate are as follows:

1. Navigate to the web interface of the Firepower Management Center at <https://fmc-ip-address> and log in with your Firepower Management Center credentials.
2. In the Firepower Management Center 6.x GUI, navigate to System > Integration > eStreamer.

#### Firepower Management Center eStreamer Certificate Creation

The screenshot displays the Firepower Management Center (FMC) interface for configuring eStreamer clients. The main navigation bar includes 'Overview', 'Analysis', 'Policies', 'Devices', 'Objects', 'AMP', 'Configuration', 'Users', 'Domains', 'Integration', 'Updates', 'Licenses', 'Health', 'Monitoring', and 'Tools'. The user is logged in as 'admin'. The 'Integration' section is active, showing 'eStreamer', 'Host Input Client', and 'Smart Software Satellite' tabs. A 'Create Client' button is visible. The 'eStreamer Event Configuration' section on the left allows selecting event types to be sent to connected clients, with all options checked: 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, and Connection Events. A 'Save' button is at the bottom of this section. The main table lists the following client information:

Hostname	
10.105.218.68	
172.16.196.1	
admin	

At the bottom of the page, it shows 'Last login on Monday, 2017-04-17 at 01:48:11 AM from 10.65.36.249' and the Cisco logo.

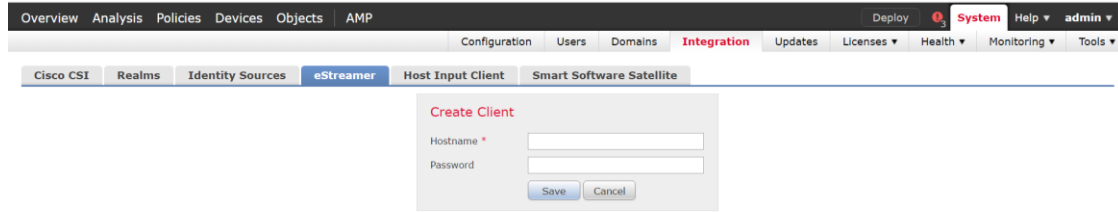
3. Click Create Client.
4. Provide the Hostname and password.

**Note:** This should be the IP of the client, which will be collecting the event data from the Firepower Management Center. The password you enter here 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 Firepower Management Center*. In other words, if the client is behind a NAT device, then the IP address must be that of the upstream NAT interface.

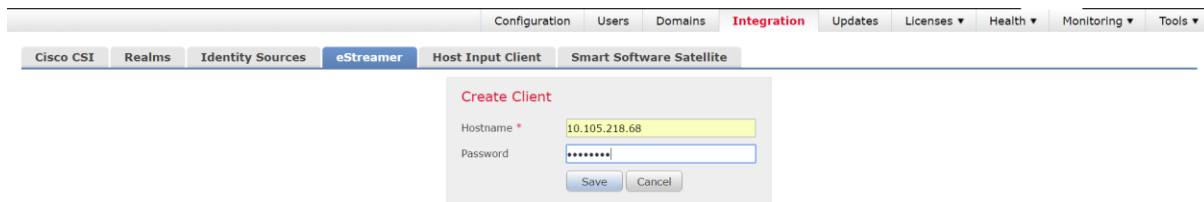
9 Appendix A: Client Certificate Creation and Installation

Create Client Hostname and Password Screen



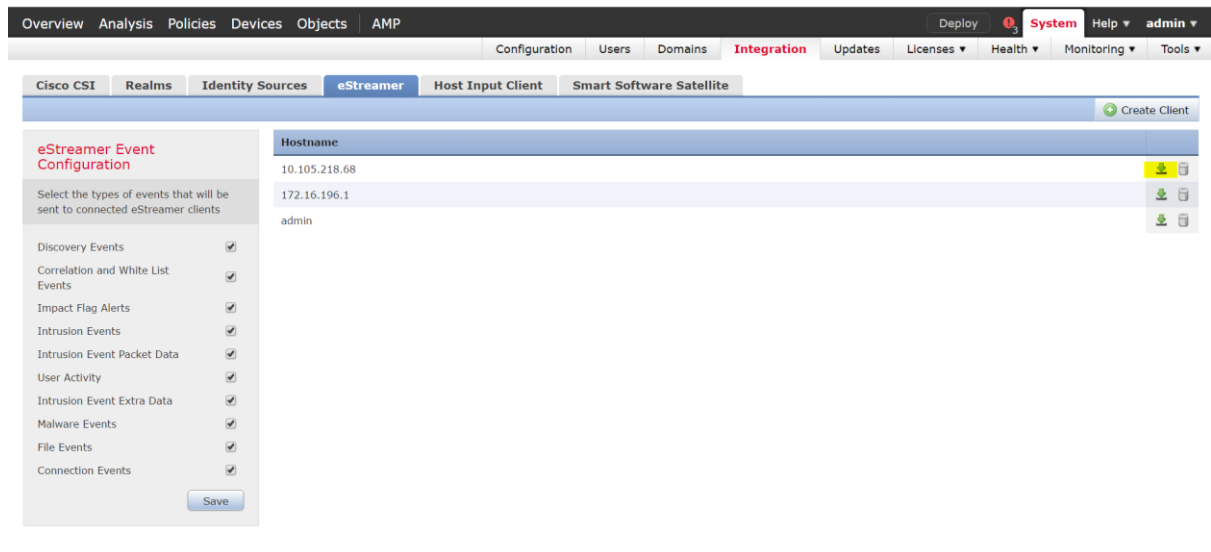
5. Click Save.

Create Client Save Screen



6. Download the PKCS12 file by clicking the Download icon at the right.

### Download Screen



7. Copy the PKCS12 file to the desired location in the target device. By default, eStreamer-eNcore will look for /path/eStreamer\_eNcore/client.pkcs12. If you wish to use a different filename, then you must edit the estreamer.conf file.



## 10 Appendix B: List of Configuration Options

The following table provides a comprehensive list of configuration options for the eNcore for Splunk add-on.

Key	Definition
Enabled	true   false. Controls whether eNcore will run.
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.details	true   false. Controls whether the monitor writes a detailed status message to the log.
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.bookmark	true   false. True will show the last bookmark timestamp. This is useful to see how far behind the eNcore client is.
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.
logging.format	This describes the format of the log and how they are stored. Default configuration setting for message format is “ <b>{date-time}-{name of module}-{level of logging-message}</b> ”.
logging.stdOut	true   false. This determines whether log output is also shown in Standard Output.

Key	Definition
logging.filepath	This specifies the location of the application log.
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 Firepower Management Center (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: <ul style="list-style-type: none"> <li>— splunk</li> <li>— json</li> </ul>

## 10 Appendix B: List of Configuration Options

Key	Definition
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 (relative:///relative/path/to/file).  Only file URLs are supported currently.
outputter.stream.options	File-based streams require additional options.
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). <i>Default configuration for this is 10,000. You can have fewer, larger files (e.g, 50,000).</i>
workerProcesses	Specify the number of worker processes, from 1 to 12, to optimize performance. See section 6.4 for guidance on setting this option.
batchSize	Specify the threshold for a batch. See section 6.5 for guidance on setting the batch size.

## 11 Appendix C: Example Configuration File

The eNcore for Splunk add-on comes with a default estreamer.conf configuration file. For reference purposes, an example configuration file is provided below.

Example estreamer.conf Configuration File

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

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

  "monitor": {
    "period": 120,
    "bookmark": false,
    "handled": true,
    "details": 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
      }
    ]
  },
}
```

## 11 Appendix C: Example Configuration File

```

"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
        }
      }
    }
  ]
},

```

11 Appendix C: Example Configuration File

```
{
  "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.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT 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.

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.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://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. (1110R)

© 2017 Cisco Systems, Inc. All rights reserved.