

# Firepower eNcore 操作指南

**首次发布日期:** 2017 年 11 月 11 日

**最后更新日期:** 2021 年 7 月 6 日



# 目录

关于本指南.....	3
修订历史记录.....	4
约定.....	4
1 简介.....	5
2 文档目的.....	5
2.1 背景.....	5
2.2 应用摘要.....	5
3 Cisco eNcore CLI.....	6
3.1 VUM CLI 必备条件.....	6
3.1.1 Python 2.7 或 Python 3.6+ 安装.....	8
3.1.2 pyOpenSSL 安装.....	8
3.1.3 EPEL RHEL 的存储库依赖关系.....	8
3.1.4 在 Azure 上运行 eNcore CLI.....	8
3.1.5 在 Windows 上运行 eNcore CLI.....	13
3.2 安装 eStreamer eNcore CLI.....	13
3.2.1 从源代码构建 eNcore 客户端.....	13
3.2.2 创建 PKCS12 文件.....	13
3.2.3 安装 PKCS12 文件.....	14
3.2.4 测试.....	14
3.2.5 运行 eNcore CLI.....	16

3.3	eStreamer eNcore CLI 配置	17
3.3.1	订用服务器	18
3.3.2	输出器	19
3.3.3	记录数	19
3.3.4	启用	20
3.3.5	执行	20
3.3.6	日志记录	22
4	Cisco Sentinel 的 eStreamer eNcore	23
4.1	将数据发送到 Sentinel	23
4.1.1	配置 Encore 以流传输 UDP	23
4.1.2	创建 Sentinel 工作空间	23
4.1.3	设置 CEF 数据连接器	25
5	适用于 Splunk 8.1+ 的思科 eStreamer eNcore 附加组件 (TA-eStreamer)	28
5.1	必备条件	29
5.2	安装	29
5.2.1	安装 Splunk 的 eNcore 附加组件 (TA-eStreamer)	29
5.2.2	安装 Splunk 的 eNcore 控制面板 (eStreamer 控制面板)	30
5.3	用于 Splunk 设置配置的 eNcore 插件	30
5.3.1	启用数据输入	30
5.3.2	启用脚本	31
5.3.3	eNcore 附加组件设置配置	31
5.4	操作	35
6	适用于 Splunk 的 Firepower 控制面板	35

6.1	入站/出站子网配置 .....	35
6.2	记录数 .....	35
6.3	监控 .....	36
6.4	开始时间 .....	37
6.5	输出器 .....	38
6.6	性能调优 .....	39
6.7	批次大小 .....	40
6.8	持久连接 .....	40
6.9	主机 .....	40
6.10	高级配置设置 .....	41
7	故障排除 .....	44
7.1	错误消息 .....	44
7.2	常见的 eNcore 问题 .....	44
7.3	常见问题解答 .....	48
8	思科支持 .....	54
9	链接和资源 .....	55
9.1	有用链接 .....	55
10	附录 .....	56
10.1	Firepower 管理中心 eStreamer 客户端证书创建 .....	56
10.2	示例配置文件 .....	58
11	免责声明 .....	61

## 关于本指南

作者	Seyed Khadem (skhademd)
更改权限	Cisco 系统高级服务, 安全与协作 IDT, 实施美洲
内容 ID	
项目 ID	

## 修订历史记录

修订	日期	名或用户 ID	备注
1.0	00/00/2021		首次公开发布

## 约定

本文档使用下列约定。

约定	指示
体 字体	命令和关键字及用户输入的文本以 体显示。
体 字体	文档标题、新出现或强调的术语，以及要为其提供数值的参数以 体显示。
[ ]	方括号中的元素是可选项。
{x   y   z}	必需的备选关键字集中在大括号内，以竖线分隔。
[ x   y   z ]	可选的备选关键字集中在方括号内，以竖线分隔。
字符串	不加引号的字符集。请勿将字符串用引号引起来，否则会将字符串和引号视为一个整体。
courier 字体	系统显示的终端会话和信息以 courier 字体显示。
< >	非打印字符（如密码）括在尖括号中。
[ ]	系统提示的默认回复括在方括号中。
!, #	代码行开头的感叹号 (!) 或井字号 (#) 表示注释行。

**注意：** 表示 读者需要注意的地方。“注释”中包含有用的建议或本文档未涵盖材料的引用信息。

**注意：** 表示 读者应当小心处理。在这种情况下，操作可能会导致设备损坏或数据丢失。

# 1 简介

## 2 文档目的

本文档概述了 CLI、Splunk 和 Sentinel 的 eStreamer eNcore 客户端的背景和用途，以帮助用户进行安装和执行。

### 2.1 背景

Cisco 事件流转换器 (eStreamer) 允许用户将系统入侵、发现和连接数据从 Firepower 管理中心或受管设备 (eStreamer 服务器) 传输到外部客户端应用。eStreamer 以简洁、紧凑、二进制编码的消息响应客户端请求，从而促进高性能。

过去，eStreamer SDK 封装了一些额外的代码来创建单独的 Perl 应用 (例如，Cisco eStreamer for Splunk 应用和 CEF 代理)。

### 2.2 应用摘要

eStreamer eNcore 是与 Firepower 管理中心版本 6.0 及更高版本兼容的多平台、多进程 Python 应用。

eNcore 是一个通用客户端，它从 eStreamer 请求所有可能的事件，解析二进制内容，并以各种格式输出事件以支持其他安全信息和事件管理工具 (SIEM)。eNcore 在 Python 中从头开始构建，具有可扩展且快速的多进程架构。它支持 Firepower 管理中心 6.0 版本。它是在 CentOS 7 上构建和测试的，但应与任何支持必备条件的 Linux 发行版配合使用。该软件将在 Windows 上运行，但不受支持。

有三个与 eStreamer eNcore 关联的软件包：

- eNcore CLI
- 适用于 Splunk 的 eNcore 插件
- eNcore Splunk 控制面板

本指南介绍这三个软件包。

## 3 Cisco eNcore CLI

eNcore CLI 是 eStreamer eNcore 的命令行界面。它作为独立应用运行，从 Firepower 管理中心 eStreamer 服务器请求事件并以以下格式之一输出这些事件：

- 旨在与以前的 Splunk 收集器保持兼容性的键值对
- JSON
- Arcsight 的 CEF 与以前的 cef 代理保持向后兼容性。

输出可以流传输到文件、TCP 或 UDP 网络端口或 stdout。

### 3.1 VUM CLI 必备条件

eNcore CLI 可与任何支持必备条件的 Linux 发行版配合使用。它将在 Windows 上运行，但尚未实现生产就绪。

安装 eNcore 的平台有两个主要前提条件：

- Python 2.7 或 Python 3.6+
- pyOpenSSL

eNcore 的 CLI 版本可以在 Python 2.7 或 Python 3.6+ 上运行。您还必须有一种拆分 Firepower 管理中心的 PKCS12 文件的方法。默认方法是安装 pyOpenSSL，让 eNcore 为您完成工作。

**注意：** 如果您希望立即开始操作，encore.sh 脚本应指导您完成所有这些要点，但值得在安装之前熟悉这些要点。

要检查 Python 2.7 是否存在，请使用以下命令：

```
which python
```

要测试 Python 2.7 的位置，请使用以下命令：

```
whereis python
```

如果已安装 Python，则 which Python 命令提供安装目录的路径。例如，如果命令的输出为 /usr/bin/python，则表示已安装 Python。要确定安装的 Python 是否为 v2.7，请列出安装目录（在上面的示例中为 /usr/bin 目录）的父目录的内容。例如，假设列表显示的条目如下所示：

```
lrwxrwxrwx 1 root root 9 Dec 9 2015 python-> python2.7 *
```

此条目显示 `python` 是指向安装 | 有 Python v2.7 的 `python2.7` 目录的链接。另一个命令，`whereis python`，也可用于显示是否存在 `python2.7` 目录。

**注意：** 如果在运行 Splunk 的设备上安装 CLI 版本，则值得注意的是 Splunk 具有自己的 Python 版本。Splunk Python 的编译方式与正常发行版不同-具体而言，它是使用 PyUnicodeUCS2 构建的。`encore.sh` 脚本将检测到此情况并发出警告。如果遇到此问题，则需要创建新用户并以该用户身份运行 `eStreamer-eNcore`。您应考虑运行 Splunk 加载项。

要检查 `pyOpenSSL`，请使用以下命令：

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

或者，使用 `python3` 版本将不再需要 `pyUnicodeUS4` 复杂性。要访问 `python3` 分支，请执行以下操作：`git checkout python3`



### 3.1.1 Python 2.7 或 Python 3.6+ 安装

要在 CentOS 上安装 Python，请使用以下命令：

```
sudo yum install python
```

### 3.1.2 pyOpenSSL 安装

pyOpenSSL 可能已作为 Python 2.7 安装的一部分已安装。要检查是否已安装，请使用以下命令：

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

如果未安装 pip，可使用此命令将其安装在 CentOS 上：

```
sudo python get-pip.py
```

使用以下命令安装 pyOpenSSL：

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

如果使用的是 python3 分支，请运行以下命令：

```
sudo pip3 install pyOpenSSL
```

### 3.1.3 EPEL RHEL 的存储库依赖关系

如果您在安装这些软件包时遇到问题，则可能需要启用 EPEL 存储库。有关安装和启用 EPEL 存储库的说明，请访问互联网。

Red Hat 的 EPEL 指南：

<https://access.redhat.com/solutions/3358>

<https://www.redhat.com/en/blog/whats-epel-and-how-do-i-use-it>

### 3.1.4 在 Azure 上运行 eNcore CLI

- 1 创建新的 Linux 资源，例如 Ubuntu 18.04 LTS：

Azure services













Recent resources

Name	Type	Last Viewed
 encore-demo-2	Virtual machine	a week ago
 sentinencore2	Log Analytics workspace	a week ago
 d8e3a9d7-7798-47c4-9d89-d38857c5bfe7	Subscription	2 weeks ago

Navigate


 Subscriptions
  Resource groups
  All resources
  Dashboard

Tools

 Microsoft Learn <sup>2</sup>  
Learn Azure with free online training from Microsoft
  Azure Monitor  
Monitor your apps and infrastructure
  Security Center  
Secure your apps and infrastructure
  Cost Management  
Analyze and optimize your cloud spend for free

Microsoft Azure
Search resources, services, and docs (G+)

Home > New >

Ubuntu Server 18.04 LTS 

Canonical


**Ubuntu Server 18.04 LTS**  Save for later  
 Canonical  
[Create](#) [Start with a pre-set configuration](#)  
 Deploy with Resource Manager [\(change to Classic\)](#)

Overview Plans

Ubuntu Server 18.04 LTS amd64 Public Azure, Azure Germany, Azure China. Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 18.04 will be available until April 2023. Ubuntu Server is the perfect virtual machine (VM) platform for all workloads from web applications to NoSQL databases and Hadoop. For more information see [Ubuntu on Azure](#) and [using Juju to deploy your workloads](#).

Legal Terms

By clicking the Create button, I acknowledge that I am getting this software from Canonical and that the [legal terms](#) of Canonical apply to it. Microsoft does not provide rights for third-party software. Also see the [privacy statement](#) from Canonical.

Useful Links

- [Linux VM Documentation](#)
- [Ubuntu Documentation](#)
- [FAQ](#)
- [Pricing Details](#)

Microsoft Azure Search resources, services, and docs (G+)

Home > New > Ubuntu Server 18.04 LTS >

## Create a virtual machine

**Basics** | Disks | Networking | Management | Advanced | Tags | Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

**Subscription \***  ⓘ  ▼

**Resource group \***  ⓘ  ▼  
[Create new](#)

### Instance details

**Virtual machine name \***  ⓘ  ✓

**Region \***  ⓘ  ▼

**Availability options**  ⓘ  ▼

**Image \***  ⓘ  ▼  
[Browse all public and private images](#)

**Azure Spot instance**  ⓘ  Yes  No

**Size \***  ⓘ  ▼  
[Select size](#)

**Administrator account**

Authentication type  SSH public key  Password

**Username \***

SSH public key source

Key pair name \*

**Inbound port rules**

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \*  None  Allow selected ports

Select inbound ports \*

**Review + create** < Previous Next : Disks >

Home &gt; New &gt; Ubuntu Server 18.04 LTS &gt; Create a virtual machine

## Select a VM size

Search by VM size... Display cost: Monthly vCPUs: 8-16 RAM (GiB): 16-32 Family: 2 selected Add filter

Most used sizes by Azure users

Showing 6 of 363 VM sizes. Subscription: Azure subscription 1 Region: East US Current size: Standard\_D4s\_v3 Image: Ubuntu Server 18.04 LTS Learn more about VM sizes

VM Size	Family	vCPUs	RAM (GiB)	Data disks	Max IOPS	Temp storage (GiB)	Premium disk	Cost/month
F8s	Compute optimized	8	16	32	25600	32	Supported	\$290.54
F16s	Compute optimized	16	32	64	51200	64	Supported	\$581.08
F8	Compute optimized	8	16	32	32x500	128	Not supported	\$290.54
F16	Compute optimized	16	32	64	64x500	256	Not supported	\$581.08
F8s_v2	Compute optimized	8	16	16	12800	64	Supported	\$246.74
F16s_v2	Compute optimized	16	32	32	25600	128	Supported	\$494.21

- 将 CPU 分配给虚拟实例。eNcore CLI 最多可支持 12 个线程。我们建议使用 8-16 核优化计算机。eNcore CLI 使用 16 CPU F16s\_v2 选项最多每秒可支持 7000 事件。
- 根据组织的预期数量进行扩展，对于小批量（小于 500 个事件/秒）操作，建议的最小 CPU 数量为 4。
- 为您的实例命名并下载 PEM 证书。

The screenshot shows the Azure portal interface for a virtual machine named 'encore-demo-2'. The 'Networking' section is expanded, displaying the following details:

Property	Value
Public IP address	13.68.147.56
Public IP address (IPv6)	-
Private IP address	10.0.0.1
Private IP address (IPv6)	-
Virtual network/subnet	CSTA1-vnet/default
DNS name	Configure

The 'Size' section shows the following details:

Property	Value
Size	Standard D4s v3
vCPUs	4
RAM	16 GB

记下分配给您的实例的公共 IP，您将使用此它在 Firepower 管理中心 eStreamer 中创建证书。

- 5 使用 .pem 文件连接到实例的命令行版本。输入继续安装。 Azure 还有一个启用快速命令行连接的快捷方式。

The screenshot shows the 'Connect' page for the virtual machine 'encore-demo-2'. The 'SSH' tab is selected, and the 'Connect via SSH with client' section is visible. The 'Private key path' field is filled with '~/.ssh/azureuser'. The 'Connect via SSH with client' section includes the following steps:

1. Open the client of your choice, e.g. PuTTY or other clients.
2. Ensure you have read-only access to the private key.
 

```
chmod 400 azureuser.pem
```
3. Provide a path to your SSH private key file.
 

```
Private key path: ~/.ssh/azureuser
```
4. Run the example command below to connect to your VM.
 

```
ssh -i <private key path> azureuser@13.68.147.56
```

`ssh -i<private key path> azureuser@<public ip>`

```
Azure — azureuser@encore-demo-2: ~ — ssh -i ~/Documents/Azure/encore-d...

System information as of Sat Aug 22 05:17:45 UTC 2020

System load:  0.04          Processes:           155
Usage of /:   14.5% of 28.90GB  Users logged in:   0
Memory usage: 4%           IP address for eth0: 10.0.0.5
Swap usage:   0%

* Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
  sudo snap install microk8s --channel=1.19/candidate --classic

  https://microk8s.io/ has docs and details.

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

12 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Wed Aug 12 18:45:34 2020 from 108.40.123.72
azureuser@encore-demo-2:~$ █
```

### 3.1.5 在 Windows 上运行 eNcore CLI

**警告：** Windows 尚未支持生产执行。但是，如果您希望尝试安装 CLI 版本，则需要运行以下命令：  
pip install pyOpenSSL, pip install win-inet-pton.

## 3.2 安装 eStreamer eNcore CLI

### 3.2.1 从源代码构建 eNcore 客户端

使用以下命令将最新版本复制到目标客户端：

git 克 <https://github.com/CiscoSecurity/fp-05-firepower-cef-connector-arcsight>

对于以前的版本：<https://github.com/CiscoSecurity/fp-05-microsoft-sentinel-connector/releases>

### 3.2.2 创建 PKCS12 文件

eStreamer 服务器必须能够对客户端连接进行身份验证和授权。这需要 eStreamer 服务器上的 PKCS12 文件标识 eStreamer 客户端，并且必须将此文件复制到 eNcore 服务器。

有关如何在 Firepower 管理中心创建和下载 PKCS12 文件的说明，请参阅 附录。

### 3.2.3 安装 PKCS12 文件

使用以下命令将 PKCS12 文件安全地复制到 eNcore CLI 安装:

```
6scp -i /path/to/pem/encore-demo-2_key.pem /local/path/<public ip>.pkcs12  
azureuser@<Public Ip>:/tmp/
```

将证书从 /tmp 复制到 Git 项目的运行时路径:

```
cp /tmp/client.pkcs12 ~/fp-05-microsoft-sentinel-connector
```

### 3.2.4 测试

1 使用以下命令更改工作目录至 eStreamer-eNcore:

```
cd ~/fp-05-microsoft-sentinel-connector
```

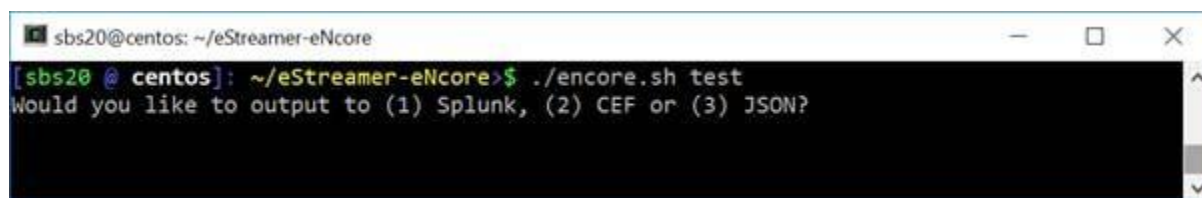
2 运行 `encore.sh` 外壳脚本-系统将指导您完成任何其他配置:

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

该脚本将验证您是否已安装必备组件，特别是:

- Python 2.7、Python 3.6+ 需要来自 Git 的“python3”分支
  - 正确的 Python 构建
  - pyOpenSSL
  - client.pkcs12 文件
  - 有效主机
- 3 选择是输出 Splunk、CEFt 还是 JSON 的数据。在本指南中，我们使用 CEF 输出器，但未来版本可能使用 JSON 或其他自定义格式，具体取决于所使用的 Sentinel 连接器。

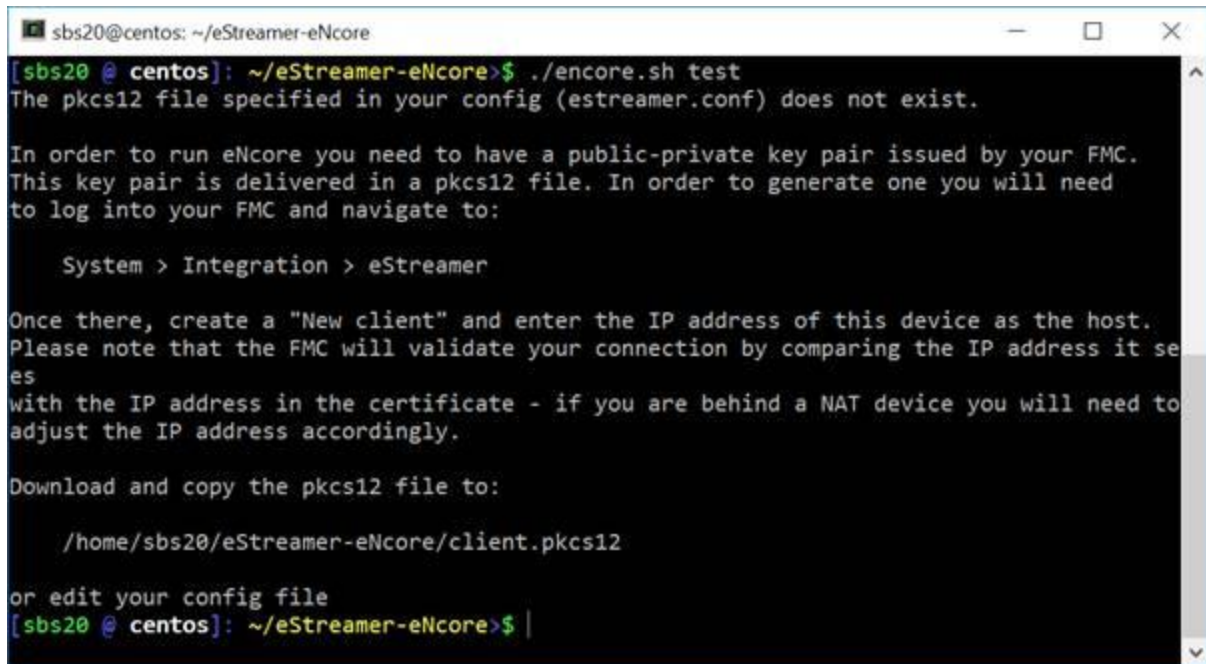
图 1. 选择输出



如果有任何缺失的项目，您将收到说明。下图提供了一个示例说明:

图 2: 缺失的 pkcs12 文件





```
sbs20@centos: ~/eStreamer-eNcore
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
The pkcs12 file specified in your config (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.

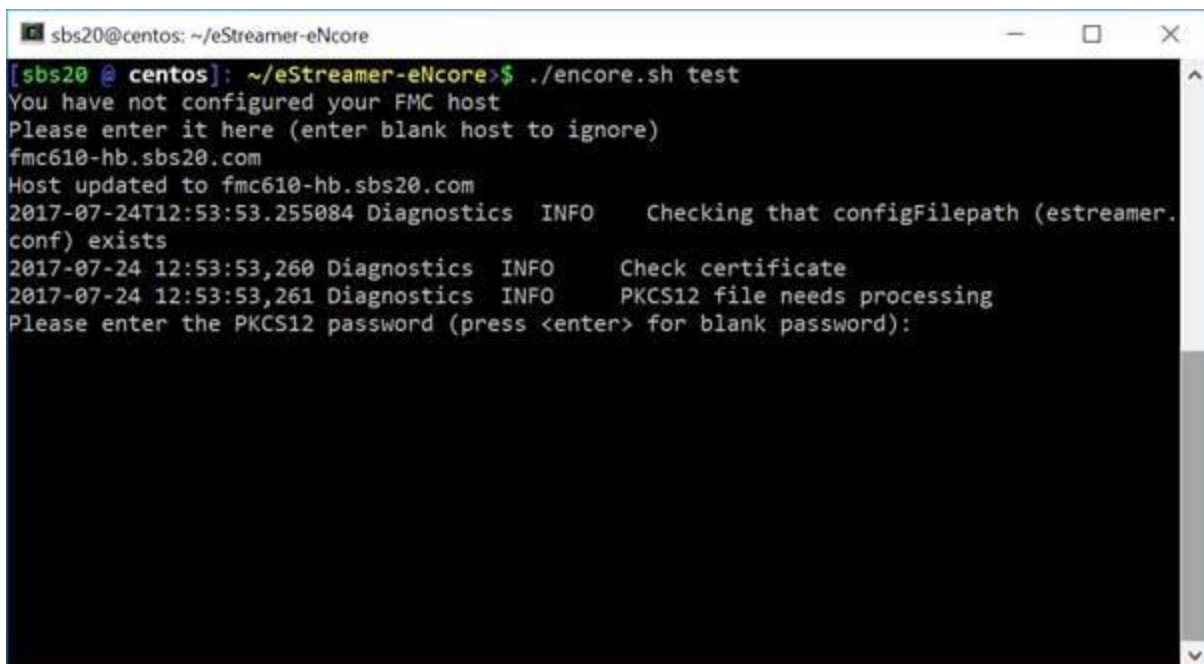
Download and copy the pkcs12 file to:

    /home/sbs20/eStreamer-eNcore/client.pkcs12

or edit your config file
[sbs20 @ centos]: ~/eStreamer-eNcore>$ |
```

4 输入 Firepower 管理中心的 IP / FQDN 和 PKCS12 文件密码。

图 3 - 输入密码



```
sbs20@centos: ~/eStreamer-eNcore
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
You have not configured your FMC host
Please enter it here (enter blank host to ignore)
fmc610-hb.sbs20.com
Host updated to fmc610-hb.sbs20.com
2017-07-24T12:53:53.255084 Diagnostics INFO    Checking that configFilePath (estreamer.
conf) exists
2017-07-24 12:53:53,260 Diagnostics INFO    Check certificate
2017-07-24 12:53:53,261 Diagnostics INFO    PKCS12 file needs processing
Please enter the PKCS12 password (press <enter> for blank password):
```

图 4: 成功的测试



```
sbs20@centos: ~/eStreamer-eNcore
[sbs20 @ centos]: ~/eStreamer-eNcore> ./encore.sh test
2017-07-24T12:54:37.898114 Diagnostics INFO    Checking that configFilepath (estreamer.
conf) exists
2017-07-24 12:54:37,903 Diagnostics INFO    Check certificate
2017-07-24 12:54:37,904 Diagnostics INFO    Creating connection
2017-07-24 12:54:37,904 estreamer.connection INFO    Connecting to fmc610-hb.sbs20.com:
8302
2017-07-24 12:54:37,904 estreamer.connection INFO    Using TLS v1.2
2017-07-24 12:54:38,269 Diagnostics INFO    Creating request message
2017-07-24 12:54:38,269 Diagnostics INFO    Request message=0001000200000008ffffffff48
900061
2017-07-24 12:54:38,269 Diagnostics INFO    Sending request message
2017-07-24 12:54:38,269 Diagnostics INFO    Receiving response message
2017-07-24 12:54:38,286 Diagnostics INFO    Response message=KGRwMMapTJ2x1bmd0aCcKcDEKS
TQ4CnNTJ3ZlcnNpb24nCnAyCkxkxNNTJ2RhdGEEnCnAzClMnXHgwMFx4MDBceDEzXHg4OVx4MDBceDAwXHgwMFx4M
DhceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgxM1x4ODhceDAwXHgwMFx4MDBceDA4X
HgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MwFceDBiXHgwMFx4MDBceDAwXHgwOFx4M
DBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwJwpwNAPzUydtZXNzYwd1VHlwZScKcDUKSTIwNTEKcy4=
2017-07-24 12:54:38,286 Diagnostics INFO    Streaming info response
2017-07-24 12:54:38,286 Diagnostics INFO    Connection successful
[sbs20 @ centos]: ~/eStreamer-eNcore> |
```

如果测试成功，则 eNcore CLI 安装完成。

### 3.2.5 运行 eNcore CLI

如果运行不带任何参数的 `encore.sh`，系统将为您提供简要说明。

图 5: 帮助屏幕

```
sbs20@centos: ~/eStreamer-eNcore
[sbs20 @ centos]: ~/eStreamer-eNcore> ./encore.sh
Usage: {start | stop | restart | foreground | test | setup}

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
setup:     change the output (splunk | cef | json)

[sbs20 @ centos]: ~/eStreamer-eNcore>
```

对于您的第一次运行，请在前台运行它，以便您可以看到正在发生的情况。屏幕每两分钟更新一次，会记录已处理的记录数量。如果希望更改更新频率，请参阅 `monitor.period` 配置设置。

图 6: 使用监控状态在前台运行

```
sbs20@centos: ~/eStreamer-eNcore
2017-07-24 13:03:03,316 estreamer.bookmark INFO      Bookmark file /home/sbs20/eStreamer-eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,316 estreamer.handler INFO      Starting Handler.
2017-07-24 13:03:03,336 estreamer.bookmark INFO      Bookmark file /home/sbs20/eStreamer-eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,337 estreamer.settings.settings INFO      Timestamp: Start = 2 (Bookmark = 0)
2017-07-24 13:03:03,337 estreamer.subscriber INFO      EventStreamRequestMessage: 00010002000000080000000048900061
2017-07-24 13:03:03,357 estreamer.bookmark INFO      Bookmark file /home/sbs20/eStreamer-eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,358 estreamer.settings.settings INFO      Timestamp: Start = 2 (Bookmark = 0)
2017-07-24 13:03:03,358 estreamer.subscriber INFO      StreamingRequestMessage: 00010801000003800001a0b000000384890006100000000009000c00400150009001f000b003d000e00470004005b000700650006006f0002008300000000
2017-07-24 13:03:04,393 estreamer.metadata.cache WARNING Metadata key ('uuid') missing on object ({'recordType': 119, 'blockLength': 8, 'checksum': 0, 'recordLength': 8, 'archiveTimestamp': 0, 'blockType': 15}). Ignoring
2017-07-24 13:05:03,412 estreamer.monitor INFO      Running. 163730 subscribed; 163211 handled;
```

**注意:** 要停止前台进程，请输入 `Ctrl + C`。

### 3.3 eStreamer eNcore CLI 配置

第 2.2 节中介绍的 eNcore CLI 安装过程需要配置基本项目，例如 Firepower 管理中心 IP，才能与 Firepower 管理中心 eStreamer 服务器建立连接。本节介绍应用的一般配置，以使其满足解决方案要求。

配置被存储在 eStreamer-eNcore 目录中的 estreamer.conf 文件中。最初，它包含可根据需要更改的默认设置。该文件为 JSON 格式，包含提供配置信息的密钥。本节详细介绍最可能被更改的密钥和部分。默认配置文件设置为开箱即用。以下是您可以自定义的每个设置的简要说明。

### 3.3.1 订用服务器

这是 Firepower 管理中心主机和相关信息。如果遇到 TLS 困难并愿意降级，则可以将 tlsVersion 更改为 1.0。

**注意：** 降级 TLS 版本对于调试和查看软件工作非常有用，但不是建议的长期策略。建议修复根本原因。

订用密钥包含两个主要子部分：

- 记录部分允许用户选择 eNcore 在连接到 Firepower 管理中心 eStreamer 服务器时请求的事件类型。
- 服务器部分包含 Firepower 管理中心主机 IP 和关联信息。

此密钥及其值的示例如下所示：

图 8: 订用服务器屏幕

```
"subscription": {
  "records": {
    "@comment": [
      "只是因为我们在订用并不意味着服务器正在发送。这也不意味着“，
      “我们正在写入记录。请参阅处理程序。记录[]“
    ],
    "archiveTimestamps": true,
    "eventExtraData": true,
    "extended": true,
    "impactEventAlerts": true,
    "intrusion": true,
    "metadata": true,
    "packetData": true
  },
  "servers": [
    {
      "host": "1.2.3.4",
      "port": 8302,
      "pkcs12Filepath": "client.pkcs12",
      "@comment": "有效值为 1.0 和 1.2",
      "tlsVersion": 1.2
    }
  ], ...
}
```

### 3.3.2 输出器

输出器部分指定 eNcore 如何将事件写入输出。eNcore CLI 可提供以下格式之一的输出：

- Splunk
- JSON
- 用于 Arcsight 的 CEF

输出可通过网络连接发送到 SIEM 或其他收集器或写入文件。

以下示例显示：

- 配置为通过 UDP 将输出发送到 ArcSight 连接器的 ArcSight CEF 输出程序。
- 将相同事件写入本地文件的 ArcSight CEF 输出程序。URI 中的 {0} 表示指定应在文件名中放置 UNIX 时间戳。

```
"outputters": [  
  {  
    "name": "CEF",  
    "adapter": "cef",  
    "enabled": true,  
    "stream": {  
      "uri": "udp://10.0.1.2:514",  
    }  
  },  
  {  
    "name": "CEFfile",  
    "adapter": "cef",  
    "enabled": true,  
    "stream": {  
      "uri": "relfile:///data/data.{0}.cef",  
      "options": {  
        "rotate": false  
        "maxLogs": 9999  
      }  
    }  
  }  
]
```

### 3.3.3 记录数

记录部分指定 eNcore 将处理哪些记录。有两种模式可识别事件以进行处理（或从处理中排除）。

1. 用户可以通过将某个类的值设置为 true 来指定应处理的事件类，例如连接。一个示例是-键值对“链接”：true。相反，用户也可以通过将某类事件的值设置为 false 来指定不应处理该类事件。
2. 用户可以通过将记录类型写入为包含或排除键的值来指定按记录类型处理事件类的例外。多个值应在 JSON 数组中以逗号分隔。例如，要排除记录类型 98 和 170，排除键-值对将显示为：

**“排除”**: [98, 170],

记录键-值对的示例如下所示：

**注意：** 请注意，对于要处理的记录类，必须首先在 Firepower 管理中心 eStreamer 配置中选择它们。还必须在 eNcore 配置的订用部分的记录部分为订阅配置它们。

```
"records": {
  "connections": true,
  "core": true,
  "excl@comment": [
    “这些记录将被排除，无论以上（覆盖‘包括’）”，
    “例如，要排除流和 IPS 事件，请使用[71,400]”
  ],
  "exclude": [],
  "inc@comment": “无论上述内容如何，都将包含这些记录”,
  "include": [],
  "intrusion": true,
  "metadata": false,
  "packets": true,
  "rna": true,
  "rua": true
}
```

### 3.3.4 启用

已启用密钥的值必须设置为 true，eNcore 才能从 Firepower 管理中心请求事件并开始流传输操作。此密钥的示例如下：

```
"enabled": true,
```

### 3.3.5 执行

按照说明完全配置所有项目后，即可使用 eNcore CLI 传输和写入事件。

各种外壳脚本选项可用。

在安装和初始设置期间-或出于调试目的，运行以下命令非常有用：

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

与

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

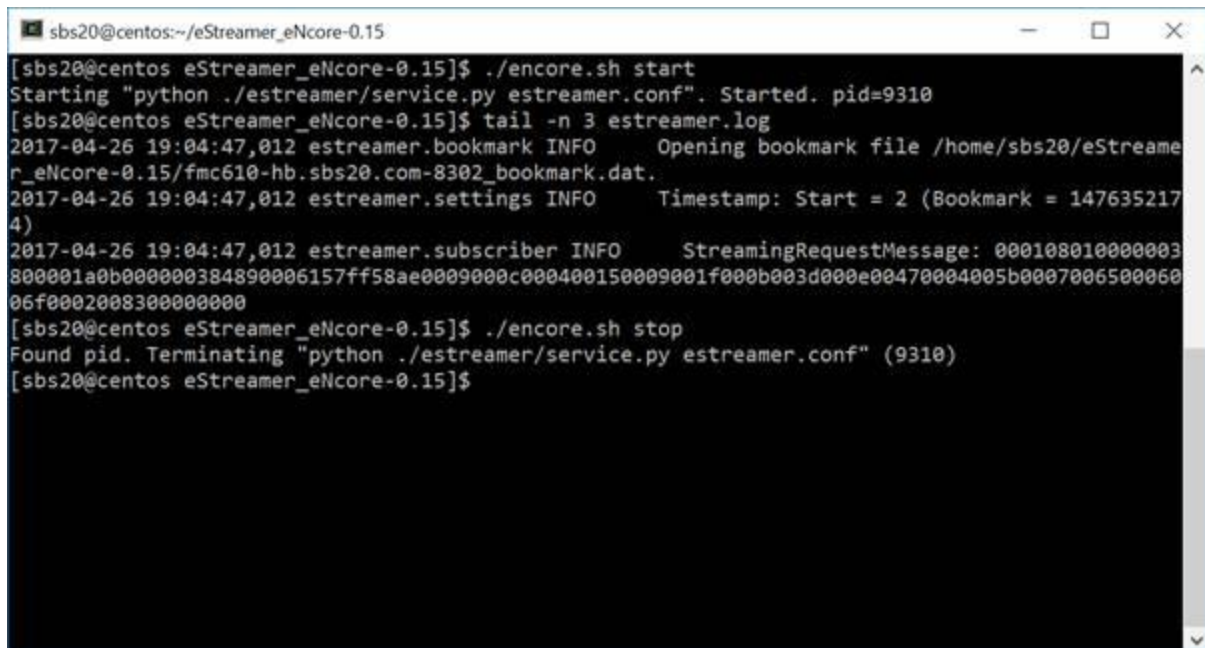
在所有其他情况下，预计 eNcore 将在后台运行，为此需要使用以下命令。

```
./encore.sh start
```

```
./encore.sh stop
```

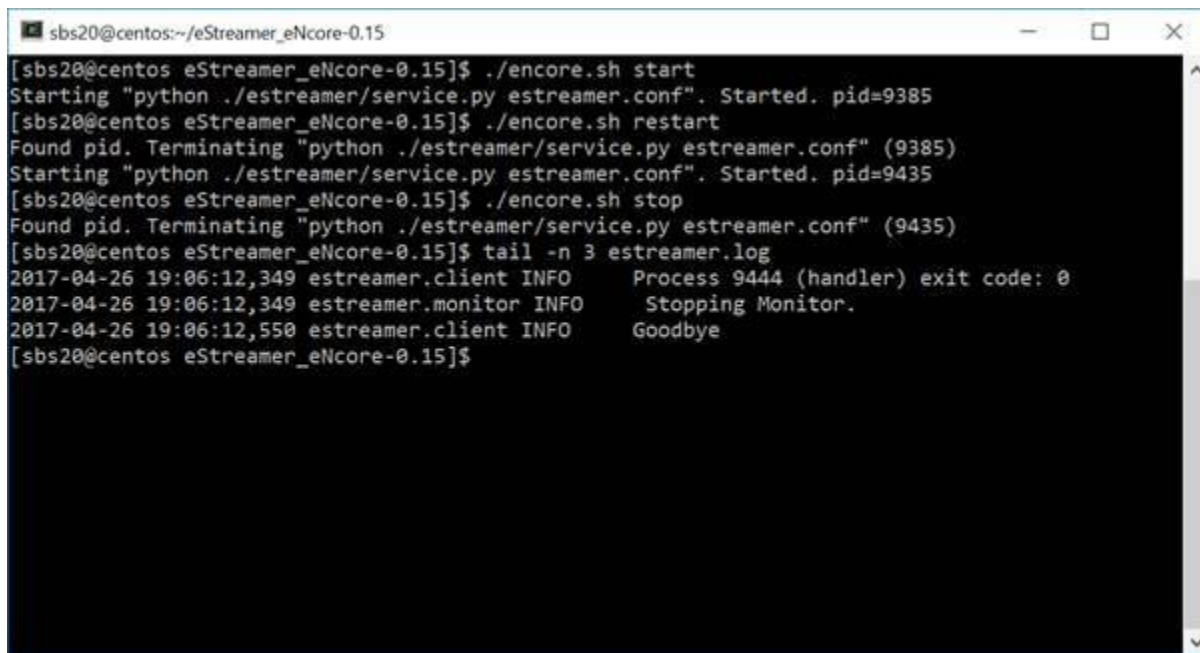
```
./encore.sh restart
```

图 12: 开始、尾部日志、停止



```
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/eStrea
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]$
```

### 3.3.6 日志记录

默认情况下，eNcore 将输出 `estreamer.log` 应用，以 INFO 日志级别登录其工作目录。可以使用 `logging.format` 配置设置来调整日志文件的格式。级别也可以调整。建议保留默认设置以执行生产。

## 4 Cisco Sentinel 的 eStreamer eNcore

### 4.1 将数据发送到 Sentinel

#### 4.1.1 配置 Encore 以流传输 UDP

将 encore 配置为在端口 25226 上使用 UDP 流传输 CEF 数据。如果 encore 已在处理中，请使用 encore.sh stop/start 命令重新启动 encore。

```
"connectTimeout": 10,
"enabled": true,
"handler": {
  "output@comment": "If you disable all outputters it behaves as a sink",
  "outputters": [
    {
      "adapter": "cef",
      "enabled": true,
      "stream": {
        "uri": "udp://127.0.0.1:514"
      }
    }
  ],
  "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": {
```

[ Read 74 lines ]

#### 4.1.2 创建 Sentinel 工作空间

在 Firepower 管理中心和 Azure 实例之间建立有效的 eNcore 连接后，即可使用代理收集器将数据输出路由到 Sentinel。

如果您没有 Sentinel 工作空间，请继续执行以下操作：



[Home](#) > [Azure Sentinel workspaces](#) > [Choose a workspace to add to Azure Sentinel](#) >

## Create Log Analytics workspace

[Basics](#) [Pricing tier](#) [Tags](#) [Review + Create](#)

**i** A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#) ×

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	<input type="text" value="Azure subscription 1"/>
Resource group * ⓘ	<input type="text" value="CSTA1"/>
	<a href="#">Create new</a>

### Instance details

Name * ⓘ	<input type="text" value="SentinelEncore"/>
Region * ⓘ	<input type="text" value="East US"/>

[Review + Create](#)[« Previous](#)[Next : Pricing tier >](#)

Microsoft Azure Search resources, services, and docs (G+/)

Home > Azure Sentinel workspaces > Choose a workspace to add to Azure Sentinel >

## Create Log Analytics workspace

Basics Pricing tier Tags Review + Create

**i** A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

**Project details**  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*   
[Create new](#)

**Instance details**

Name \*

Region \*

[Review + Create](#) [« Previous](#) [Next: Pricing tier >](#)

### 4.1.3 设置 CEF 数据连接器

在 Firepower 管理中心和 Azure 实例之间建立有效的 eNcore 连接后，即可使用代理收集器将数据输出路由到 Sentinel。请参阅官方 Microsoft 指南 (<https://docs.microsoft.com/en-us/azure/sentinel/connect-cef-agent?tabs=rsyslog>)。

Microsoft Azure Search re

Home > Azure Sentinel workspaces > Azure Sentinel | Data connectors >

## Common Event Format (CEF)

首选直接从 Sentinel 访问连接器文档指南，因为文档和预填充命令将包含特定于您的 Azure 实例的工作空间和主键信息。

下述的后续步骤直接来自 Azure Sentinel 设置指南以供参考。

**注意：**最好在 Sentinel 平台上使用直接文档，因为它包含安装代理收集器时需要运行的确切命令和工作空间/主 ID。

运行部署脚本：

1. 从 Azure Sentinel 导航菜单中，点击 **数据连接器**。
2. 从连接器列表中，点击 **通用事件格式 (CEF)** 磁贴，然后点击右下角的 **打开连接器页面** 按钮。
3. 在 1.2 **在 Linux 计算机上安装 CEF 收集器** 下，复制在 **运行以下脚本以安装和应用 CEF 收集器** 下提供的链接，或者从以下文本中复制链接：

```
sudo wget https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/CEF/cef_installer.py&&sudo python cef_installer.py [WorkspaceID] [Workspace Primary Key]
```

4. 在脚本运行时，请检查以确保您没有收到任何错误或警告消息。

**注意：**使用同一台计算机转发系统日志和 CEF 消息。

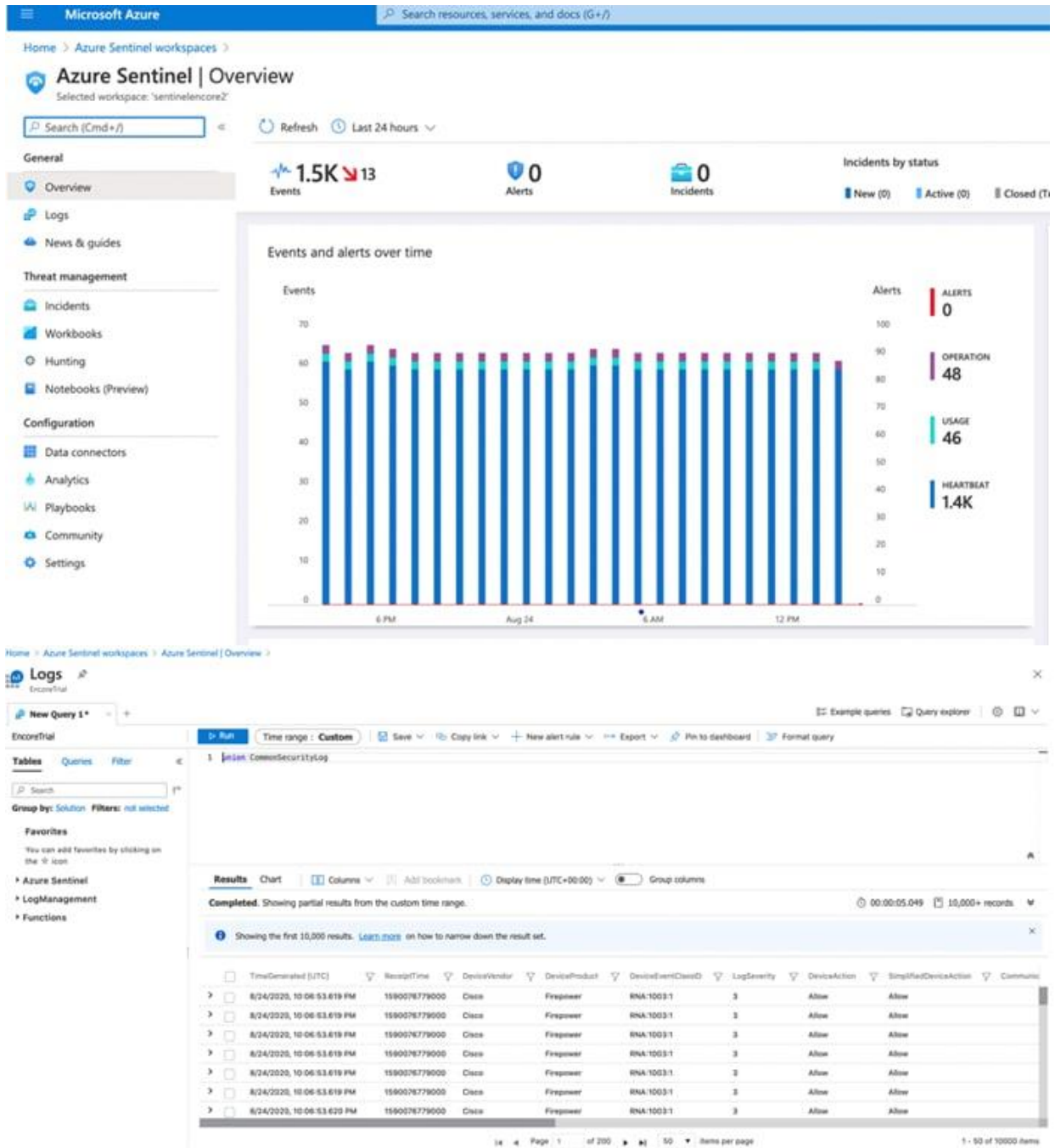
如果计划使用此日志转发器转发 系统日志消息 和 CEF，则为了避免将事件复制到系统日志和 CommonSecurityLog 表，您需要执行以下操作：

- 在以 CEF 格式向转发器发送日志的每台源计算机上，编辑系统日志配置文件以删除用于发送 CEF 消息的设施。这样，在 CEF 中发送的设施也不会系统日志中发送。有关如何执行此操作的详细说明，请参阅 在 Linux 代理上配置系统日志。
- 在这些计算机上运行以下命令，以禁用代理与 Azure Sentinel 中的系统日志配置的同步。这可确保您在上一步中所做的配置更改不会被覆盖。

```
sudo su omsagent -c 'python /opt/microsoft/omsconfig/Scripts/OMS_MetaConfigHelper.py --disable' https://docs.microsoft.com/en-us/azure/sentinel/connect-cef-agent?tabs=rsyslog
```

运行验证脚本后，您应该能够看到数据进入 Azure Sentinel 分析屏幕。





## 5 适用于 Splunk 8.1+ 的思科 eStreamer eNcore 附加组件 (TA-eStreamer)

适用于 Splunk 的 Cisco eStreamer eNcore 附加组件 (TA-eStreamer)

Splunk 的 eStreamer eNcore 插件是一种技术插件，包含核心 eNcore eStreamer 客户端代码以及：

- 用于数据、日志和状态的数据输入 (inputs.conf)
- 解析提示 (props.conf)
- 允许 eNcore 与 Splunk 一起生存和死亡的扩展程序

**注意：** Splunk for Windows 不支持 Splunk 插件的 eNcore。

### 适用于 Splunk 的 Cisco eStreamer eNcore 控制面板 (eStreamer 控制面板)

这是一个包含与旧版 Cisco eStreamer for Splunk 应用 (<https://splunkbase.splunk.com/app/1629/>) 相同的用户界面元素的应用。但是，应用不包含代码或收集器元素。它只是一个包含一些预定义搜索、宏、事件类型和工作流程操作的 UI 应用。

## 5.1 必备条件

Splunk 的 eNcore 插件和 Splunk 的 eNcore 控制面板不需要任何特殊的先决条件。它们可从 Splunkbase 下载，并以与其他加载项和应用相同的方式安装在搜索头中。

Splunk 的 eNcore 插件需要 Python 3.6+ 和 openssl，在最新的 Splunk 8.1 版本中包含 Python3，但不包括适用于 openssl 的 Python mod，这需要本更新中列出的其他配置步骤。如果 Splunk 安装已自定义，并且缺少一个或两个组件，则需要安装它们才能使附加功能正常运行。

## 5.2 安装

**注意：** Splunk for Windows 不支持 Splunk 插件的 eNcore。

### 5.2.1 安装 Splunk 的 eNcore 附加组件 (TA-eStreamer)

要为 Splunk 安装 eNcore 插件，请执行以下操作之一：

- 从 <http://apps.splunk.com/app/3662> 下载加载项，并使用 Splunk 中的“从文件安装应用”功能上传并安装加载项。
- 使用 Splunk 中的“浏览更多应用”功能并搜索 eNcore，然后在搜索结果中查找 Splunk 的 Cisco eStreamer 附加组件，然后点击该附加组件的安装。

您必须为 Splunk 服务器安装 PKCS12 证书，当 eNcore 客户端联系 Firepower 管理中心并建立安全隧道时，该证书允许 Firepower 管理中心对附加组件的身份进行身份验证：

- 在 Firepower 管理中心上创建 PKCS12 证书。



- 下载证书。
- 将证书复制到 Splunk 服务器上的两个位置 (将其重命名为 client.pkcs12) :  
\$SPLUNK\_HOME/etc/apps/TA-eStreamer/bin/encore/client.pkcs12  
\$SPLUNK\_HOME/etc/apps/TA-eStreamer/bin/client.pkcs12

有关创建 PKCS12 证书并将其复制到 Splunk 服务器的详细信息, 请参阅附录。

## 5.2.2 安装 Splunk 的 eNcore 控制面板 (eStreamer 控制面板)

要安装 Splunk 的 eNcore 控制面板, 请执行以下操作之一:

- 从 <http://apps.splunk.com/app/3663> 下载应用, 并使用 Splunk 中的“从文件安装应用”功能上传和安装加载项。
- 使用 Splunk 中的“浏览更多应用”功能并搜索“eNcore”, 然后在搜索结果中查找适用于 Splunk 的 Cisco Firepower eNcore 应用, 然后点击该应用的安装。

## 5.3 用于 Splunk 设置配置的 eNcore 插件

### 5.3.1 启用数据输入

Splunk 的 eNcore 插件将事件写入安装数据目录中的日志文件。Splunk 必须配置有从此目录读取事件的数据输入。

要执行此操作, 请转至 **设置 (Settings) > 数据输入 (Data Inputs) > 文件和目录 (Files Directories)**, 并使用以下路径启用数据输入: \$SPLUNK\_HOME/etc/apps/TA-eStreamer/data 和源类型 cisco:estreamer:data。



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>

## 5.3.2 启用脚本

Splunk 的 eNcore 插件有三个执行重要操作的脚本：

- `cisco:estreamer:clean` - 没有输出，但用于删除超过 12 小时的数据文件。
- `cisco:estreamer:log` - 使用 eNcore 的标准输出获取程序日志数据。这在不计划的情况下变得非常有用。更重要的是，启动 eStream eNcore 进程的是脚本。
- `cisco:estreamer:status` - 定期运行，以保持程序是否正在运行的明确状态。

要启用脚本，请转至 **设置 (Settings) > 数据输入 (Data Inputs) > 脚本 (Scripts)**，然后为三个 TA-eStream 脚本点击 **启用 (Enable)**。



## 5.3.3 eNcore 附加组件设置配置

转到位于 `$SPLUNK_HOME/etc/apps/TA-eStream/bin` 中的 TA-eStream bin 目录，其中 `$SPLUNK_HOME` 表示 Splunk 重型转发器安装的主目录。

要设置 `SPLUNK_HOME` 安装变量的 `homepath (SPLUNK_HOME)`，请执行以下命令：

```
export SPLUNK_HOME=/opt/splunk
```

其中 `/opt/splunk` 是 Splunk 安装的主位置，如果此更改不同。

然后，如果运行启动/测试脚本，您可能会看到以下错误：

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

要解决此问题，请为 Splunk Lib 路径添加一个额外的设置变量，该变量在脚本中已注释，您需要在执行设置脚本之前运行以下命令：

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



设置 SPLUNK\_HOME 和 LD\_LIBRARY\_PATH 包含在本地终端会话中，要保留这些值，请执行以下操作：

```
GNU nano 4.8 /root/.bash_profile
export SPLUNK_HOME=/opt/splunk #Modify if your SPLUNK_HOME directory is not /opt/splunk
export LD_LIBRARY_PATH=$SPLUNK_HOME/lib
```

```

^G Get Help      ^O Write Out    ^W Where Is    [ Read 2 lines ]
^X Exit          ^R Read File    ^\ Replace     ^K Cut Text    ^J Justify
^_              ^L Go To Line  ^T To Spell    ^C Cur Pos    ^U Undo
^_              ^M Redo       ^A Mark Text   ^M Copy Text

```

在 Ubuntu 上：

- 1 编辑 ~/.bash\_profile file 文件。
- 2 添加上述导出变量，如下所示：
  - 导出 SPLUNK\_HOME = /opt/splunk
  - 导出 LD\_LIBRARY\_PATH=SPLUNK\_HOME/lib
- 3 保存文件，然后运行 ~/.bash\_文件。

```
root@splunk-8-1:~# nano ~/.bash_profile
root@splunk-8-1:~# source ~/.bash_profile
root@splunk-8-1:~# █
```

在 CentOS 上：

- 1 bash 文件可能使用其他别名，请尝试 ~/.profile or ~/.bashrc.
- 2 编辑文件，保存并运行上述源命令

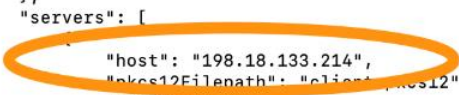
<https://community.splunk.com/t5/Developing-for-Splunk-Enterprise/How-to-get-Splunk-Python-on-CentOS-to-use-SSL-Crypto/m-p/310051>。


修改 estreamer.conf 文件，使其指向 Firepower 管理中心服务器主机 IP 地址

```

GNU nano 4.8 estreamer.conf Modified
"responseTimeout": 2,
"star@comment": "0 for genesis, 1 for now, 2 for bookmark",
"start": 0,
"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": "198.18.133.214",
      "pkcs12Filepath": "client.pkcs12",
      "port": 8302,
      "tls@comment": "Valid values are 1.0 and 1.2",
      "tlsVersion": 1.2
    }
  ]
},
"workerProcesses": 1
}

```





3 编辑设置: false, 将其更改为 true。

4 运行 ./splencore.sh test 命令。

```

root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin# ./splencore.sh test
2021-06-14T19:25:01.680552 Diagnostics INFO Checking that configFilepath (estreamer.conf) exists
2021-06-14 19:25:01,692 Diagnostics INFO Check certificate
2021-06-14 19:25:01,693 Diagnostics INFO PKCS12 file needs processing
Please enter the PKCS12 password (press <enter> for blank password):
2021-06-14T19:25:13.998455 Diagnostics ERROR [no message or attrs]:

# Splunk 8.1+ Python3 does not natively support openssl, please perform the following

Run the following two commands, alternatively you can use the command line version of OpenSSL

$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.key"
$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.cert"

Note: If you are using python3 the command to install OpenSSL is as follows, using python3 with openssl will not require manual commands above and this script will automatically extract and process certificate files

sudo apt install python3-openssl

root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin#

```

5 输入 client.pkcs 证书的密码，它最初会失败并提示您输入以下命令：

命令 #1:

```
$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -nocerts -nodes -out
"/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.key"
```

命令 #2:

```
$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -clcerts -nokeys -out
"/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.cert"
```

**注意:** 用您的 Firepower 管理中心服务器主机 IP 地址替换 19.18.133.214。

```
[root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin#
[root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin# $SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.key"
WARNING: can't open config file: /opt/splunk-home/openssl/openssl.cnf
Enter Import Password:
MAC verified OK
[root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin# $SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.214-8302_pkcs.cert"
WARNING: can't open config file: /opt/splunk-home/openssl/openssl.cnf
Enter Import Password:
MAC verified OK
root@splunk-8-1:/opt/splunk/etc/apps/TA-eStreamer/bin#
```

- 6 在每个提示符后输入 Firepower 管理中心 client.pkcs 证书密码，如果成功，您将在每个命令后看到文本“MAC verify OK”。

```
client.pkcs12 configure_handler.py configure.sh encore setup.xml splencore.sh
root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin# ./splencore.sh test
2021-06-28T16:34:53.884468 Diagnostics INFO Checking that configFilePath (estreamer.conf) exists
2021-06-28 16:34:53,896 Diagnostics INFO Check certificate
2021-06-28 16:34:53,896 Diagnostics INFO PKCS12 file needs processing
Please enter the PKCS12 password (press <enter> for blank password):
2021-06-28T16:34:57.314872 Diagnostics ERROR [no message or attrs]:

# Splunk 8.1+ Python3 does not natively support openssl, please perform the following

Run the following two commands, alternatively you can use the command line version of OpenSSL

$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.194-8302_pkcs.key"
$SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.194-8302_pkcs.cert"

Note: If you are using python3 the command to install OpenSSL is as follows, using python3 with openssl will not require manual commands above and this script will automatically extract and process certificate files

sudo apt install python3-openssl

root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin# $SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -nocerts -nodes -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.194-8302_pkcs.key"
WARNING: can't open config file: /opt/splunk-home/openssl/openssl.cnf
Enter Import Password:
MAC verified OK
root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin# $SPLUNK_HOME/bin/openssl pkcs12 -in "client.pkcs12" -clcerts -nokeys -out "/opt/splunk/etc/apps/TA-eStreamer/bin/encore/198.18.133.194-8302_pkcs.cert"
WARNING: can't open config file: /opt/splunk-home/openssl/openssl.cnf
Enter Import Password:
MAC verified OK
root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin#
```

- 7 运行 ./splencore.sh test 命令，您应该会看到以下内容:

```

root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin# ./splencore.sh test
2021-06-28T16:36:30.492874 Diagnostics INFO    Checking that configFilepath (estreamer.conf) exists
2021-06-28 16:36:30,506 Diagnostics INFO    Check certificate
2021-06-28 16:36:30,506 Diagnostics INFO    Creating connection
2021-06-28 16:36:30,506 Connection INFO    Connecting to 198.18.133.194:8302
2021-06-28 16:36:30,506 Connection INFO    Using TLS v1.2
2021-06-28 16:36:30,544 Diagnostics INFO    Creating request message
2021-06-28 16:36:30,544 Diagnostics INFO    Request message=b'0001000200000008ffffffff48900061'
2021-06-28 16:36:30,544 Diagnostics INFO    Sending request message
2021-06-28 16:36:30,544 Diagnostics INFO    Receiving response message
2021-06-28 16:36:30,553 Diagnostics INFO    Response message=b'gAN9cQAoWAcAAAB2ZXJzaW9ucQFLAVgLAAAAbWVzc2FnZVR5cGVxAK0DCFgGAAAABG
VuZ3RocQNLMFgEAAAAGZGF0YXEEQzAAABOJAAAAACAAAAAATIAAAAAGAAAAAAGsAAAAIAAAAAAABxBXUu'
2021-06-28 16:36:30,554 Diagnostics INFO    Streaming info response
2021-06-28 16:36:30,554 Diagnostics INFO    Connection successful
root@ubuntu-splunk:/opt/splunk/etc/apps/TA-eStreamer/bin# █

```

## 5.4 操作

一旦您按照第 4 部分所述完全配置了所有项目，Splunk 的 eNcore 附加组件将通过选中附加组件设置页面上的 **已启用** 复选框并点击 **保存** 来启动，如第 4.3 部分所述。

执行后，可以通过搜索状态、日志和数据事件来监控加载项的操作：

- 要检查状态，请搜索 `sourcetype="cisco:estreamer:status"`。
- 要检查更详细的日志输出，请搜索 `sourcetype="cisco:estreamer:log"`。
- 要查找 eStreamer 数据，请搜索 `sourcetype="cisco:estreamer:data"`。

要进一步分析 Firepower 事件，请考虑安装适用于 Splunk 的 Cisco Firepower 应用。

# 6 适用于 Splunk 的 Firepower 控制面板

## 6.1 入站/出站子网配置

这是 eNcore 为 `estreamer.conf` 文件中的每个平台提供的默认配置，为许多部署提供最佳配置。但是，在某些情况下，用户可能需要调整某些选项。本节提供有关其中几个选项的详细信息。

## 6.2 记录数

记录部分指定 eNcore 将处理哪些记录。有两种模式可识别事件以进行处理（或从处理中排除）：

- 用户可以通过将某个类的值设置为 `true` 来指定应处理的事件类（例如连接）。

一个示例是键-值对"connections": true。相反，用户也可以通过将某类事件的值设置为 false 来指定不应处理该类事件。

- 用户可以通过将记录类型写入为包含或排除键的值来指定按记录类型处理事件类的例外。多个值应在 JSON 数组中以逗号分隔。

例如，要排除记录类型 98 和 170，排除键-值对将显示为：

**“排除”**: [98, 170],

记录键-值对的示例如下所示：

**注意：** 请注意，对于要处理的记录类，必须首先在 Firepower 管理中心 eStreamer 配置中选择它们。还必须在 eNcore 配置的订阅部分的记录部分为订阅配置它们。

```
"记录": {
  "connections": true,
  "core": true,
  "excl@comment": [
    “这些记录将被排除，无论以上（覆盖‘包括’）”，
“例如，要排除流和 IPS 事件，请使用[71, 400]”
  ],
  "exclude": [],
  "inc@comment": “无论上述内容如何，都将包含这些记录”，
  "include": [],
  "intrusion": true,
  "metadata": false,
  "packets": true,
  "rna": true,
  "rua": true
}
```

## 6.3 监控

监控器是运行监控和维护任务的单独线程。默认情况下，它每两分钟运行一次。它将处理的事件数写入 eNcore 日志并检查子进程的状态。如果子进程存在任何问题，监控器会将客户端置于错误状态，然后客户端自行关闭。

监控线程写入日志的消息示例如下所示：

```
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;
```

可以在 estreamer.conf 配置文件的 monitor 部分配置日志消息的多个方面，该配置文件位于：

Splunk: \$SPLUNK\_HOME/etc/apps/TA-eStreamer/bin/encore/estreamer.conf

Sentinel/CEF: / fp-05-firepower-cef-connector-arcsight/estreamer.conf

可配置的方面包括：

- 阶段：监控器对子进程执行检查并将状态消息写入日志的间隔（以秒为单位）。
- 书签：如果为 true，则书签（Unix 时间格式的最新事件时间）包含在每条监控器日志消息中。
- 已处理：如果为 true，则为 eNcore 自启动以来已处理的事件数。
- 详细信息：如果为 true，则除了监控器写入日志的简短状态消息外，它还将写入包含与 eNcore 客户端操作相关的许多状态项的详细消息。

在 estreamer.conf 文件中这些参数的配置示例：

```
"monitor": {  
  "period": 120,  
  "书签": false,  
  "已处理": true,  
  "详细信息": true  
},
```

## 6.4 开始时间

eStreamer 服务器希望客户端请求声明开始时间，指定 Firepower 管理中心应仅发送在开始时间之后发生的事件。有三个选项：

- 0：从 Firepower 管理中心上可用的最早点发送所有事件。
- 1：发送在收到客户端请求后发生的所有事件。
- 2：使用书签从上次停止的地方开始。首次运行是从 0 开始。

estreamer.conf 文件中的启动配置示例如下所示：

```
"@startComment": "0 代表创    ， 1 代表现在， 2 代表书签",  
"开始": 2,
```



## 6.5 输出器

默认情况下，仅启用 Splunk 输出程序。它将数据写入相对文件位置，但您可能希望将数据输出到其他位置。要更改它，请将 `stream.uri` 属性更改为 `file:///absolute/file/path/filename{0}.ext`，其中 `{0}` 是时间戳占位符。

`estreamer.conf` 文件中的输出器配置示例：

```
"输出器": [
  {
    "名称": "Splunk default",
    "适配器": "splunk",
    "enabled": true,
    "stream": {
      "uri": "relfile:///data/splunk/encore.log{0}"
      "options": {
        "轮动": true,
        "maxLogs": 9999
      }
    }
  }
],
```

```
"输出器": [
  {
    "名称": "Arcsight"
    "适配器": "cef",
    "enabled": true,
    "stream": {
      "uri": "relfile:///data/cef/encore{0}.cef",
      "options": {
        "rotate": true,
        "maxLogs": 9999
      }
    }
  }
],
```

## 6.6 性能调优

eNcore for Splunk 附加组件的性能在版本 4.x 中通过添加多处理功能得到了改进。默认情况下，对传入消息运行四个工作进程以实现更高的吞吐量。

虽然多个进程可以提供显著的性能提升，但这些提升很大程度上取决于平台，因为对于每个平台，处理瓶颈可能不同。多个进程还需要额外的开销来管理任务分配，因此增加进程数实际上可能会降低 CPU 核心数量较少的平台上的性能。

工作进程数可通过 `estreamer.conf` 文件中的 `workProcesses` 参数进行配置。该数字可设置为 1 到 12。通常，平台越强大（即，更多的 CPU 核心、更好的 I/O 等），通过更多的工作进程实现更多的吞吐量。但是，唯一可靠的方法是使用各种设置，例如 1、2、4、8 和 12 来测试性能，而且在许多情况下，只需一个工作进程即可获得最佳性能，因为不需要进程编组。

一个测试场景是：

1. 在 Splunk 中禁用加载项的数据输入，因为在测试期间将多次请求相同的事件。
2. 配置一定数量的 `workerProcesses`（例如 8），然后使用开始参数 0（用于起源）或至少旧的开始时间启动 eNcore。
3. 从 Firepower 管理中心请求连接事件（或以其他方式请求 Firepower 管理中心发送数百万个积压事件）。
4. 观察 `estreamer.log` 文件中监控进程报告的事件速率。
5. 对不同数量的 `workerProcesses` 重复此测试。
6. 确定最佳数量后，将 `workerProcesses` 设置为该数量，并启用插件的数据输入以恢复生产操作。

`estreamer.conf` 文件中的 `workProcesses` 配置示例：

事件速率 (每秒)	工作线程	批量大小 ( )
小于 100	1	2
100-2000	1	100 (默认值)
2000-4000	4	100 (默认值)
4000-6000	8	250
8000+	12	500



典型的 Splunk 重转发平均每秒可以处理 4000-5000 个事件，但是，此速率取决于操作系统上的可用资源，其他后台任务，或者其他 TA（技术插件）可能导致较低的性能速率。

如果使用专用虚拟机，规格将主要取决于您希望处理多少卷。每秒处理大约 4000 个事件的典型安装应安装在 8 核 @ 3.6 GHz CPU，32 GB RAM 计算机或（c5.2x 大型 ec2 实例）上。对于每天仅处理少量事件（小于 100 个事件）的轻量级客户端，eNcore 经过测试，已知可使用最低规格 4 核和 1GB RAM。

## 6.7 批次大小

Splunk 插件的 eNcore 还会尝试通过对收到的事件进行批处理来提高性能，并且仅在达到批处理的阈值时才将其写入输出。默认批处理大小为 100 个事件。

如果事件速率非常低，则 100 个事件的批量大小可能会导致 Splunk 中事件出现的意外延迟。

例如，如果入侵事件是唯一处理的事件，且入侵事件速率平均每小时 100 个事件，则在批量处理完成并写入磁盘时，批量处理中的第一个事件通常会延迟一小时或更长时间。要减少此类延迟，可以将 batchSize 设置为较低的值，或者要完全消除它们，可以将 batchSize 设置为 1。

将 batchSize 设置为 1 的缺点是，在高吞吐量环境中，整体事件速率会更低。此外，由于事件不断写入磁盘，因此文件锁定和旋转可能会出现，因此强烈建议将最小批处理大小设置为 2。

estreamer.conf 文件中的 batchSize 配置示例：

```
"batchSize": 50
```

对于大量配置，您可以将 batchSize 设置为最高 500，以实现最佳性能。同样，batchSize 越高，客户端写入磁盘的频率越低，而文件 I/O 越少，意味着额外的计算会以处理事件的轻微延迟为代价。

## 6.8 持久连接

可以无限期保留客户端以侦听来自 Firepower 管理中心的数据流，或者让 eNcore 在 Splunk 中断后自动重新启动，这可以通过将以下配置值设置为 true 来实现：

```
"alwaysAttemptToContinue": true
```

## 6.9 主机

默认情况下，在 estreamer.conf 文件中定义了通用占位符，您需要将其更改为 Firepower 管理中心的 IP 或主机名。截至撰写本文时，仅支持 IPv4 地址。

**“主机”：“1.2.3.4”**

```

GNU nano 4.8 encor
    "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": "198.18.133.214",
    "pkcs12Filepath": "client.pkcs12",
    "port": 8302,
    "tls@comment": "Valid values are 1.0 and 1.2",
    "tlsVersion": 1.2
  }
]
},
"workerProcesses": 1
}

```

<sup>^</sup>G Get Help    <sup>^</sup>O Write Out    <sup>^</sup>W Where Is    <sup>^</sup>K Cut Text    <sup>^</sup>J Justify    <sup>^</sup>C Cur Pos  
<sup>^</sup>X Exit        <sup>^</sup>R Read File    <sup>^</sup>\ Replace     <sup>^</sup>U Paste Text   <sup>^</sup>T To Spell    <sup>^</sup>\_ Go To Line

## 6.10 高级配置设置

下表介绍了 estreamer.conf 文件的关键定义。

密钥	定义
alwaysAttemptToContinue	true   false. 控制即使 CLI 进程已终止，eNcore 客户端是否仍将保留连接。
batchSize	在写入磁盘之前存储在内存中的事件数。默认值为 100，请为低流量调整此值，因为事件将在内存中排队，直到达到此阈值。 为大量实施调整更高，这将限制文件 I/O 访问请求的数量并提高客户端的性能，但事件可能会延迟。延迟系数与您的注入速率相关。例如，如果您的注入速率为 100 事件/秒，batchSize 为 500，则每 5 秒数据将写入磁盘。
已启用	true   false. 控制 eNcore 是否运行。
connectTimeout	客户端在失败之前等待连接建立的持续时间（秒）。

密钥	定义
responseTimeout	客户端在超时之前等待响应的持续时间（秒）。
9monitor.period	每次执行监控任务之间的时间间隔（秒）。默认值为 120。较低的数字可用于调试，但会创建更多日志流量。
monitor.velocity	true   false. True 将显示客户端处理记录的速度。正值表示客户端处理事件的速度比 eStreamer 发送事件的速度快。负值比较慢。一旦更新，此值应保持在零附近。
monitor.bookmark	true   false. True 将显示最后一个书签时间戳。这有助于了解 eNcore 客户端的后退情况。
monitor.subscribed	true   false. True 将报告已订阅的事件总数。
monitor.handled	true   false. True 将报告写入输出的事件总数。
开始	0 指定最旧的可用数据。 1 指定截至目前的数据。 2 指定使用书签。
logging.level	级别包括 FATAL、ERROR、WARNING、INFO、DEBUG、VERBOSE 和 TRACE。根据您的要求选择日志记录级别。强烈建议您不要在生产环境中使用高于 INFO 的任何内容。DEBUG 将生成非常大的日志文件，而 TRACE 将显着影响性能。
logging.format	这描述了日志的格式及其存储方式。消息格式的默认配置设置为“{date-time} {name of module}-{level of logging-message}”。
logging.stdOut	true   false. 这将确定日志输出是否也显示在标准输出中。
logging.filepath	这指定应用日志的位置。
maxQueueSize	在限制发生之前缓冲的最大邮件数。它实际上是一个缓冲区大小。此数字越大，关闭所需的时间越长。默认配置设置为 100。请勿更改。
subscription.servers[]	虽然这是一个阵列，但 eNcore 当前只能支持一个服务器。该阵列支持未来连接到多台主机的功能。
server.host	Firepower 管理中心的 IP 地址 (eStreamer Server)。默认配置为 1.2.3.4。如果在运行 eNcore 后更改主机条目，则将生成新的缓存、书签和元数据文件。
server.port	连接的服务器端口。默认值为 8302。
server.pkcs12Filepath	PKCS12 文件路径位置。如果更改已运行 eNcore，则还必须删除缓存的公钥和私钥；否则，eNcore 将继续使用这些地址。它们分别称

密钥	定义
	为{host}-{port} _pkcs.cert 和{host}-{port} _pkcs.key。
server.tlsVersion	有效的选项为 1.0 和 1.2。
subscription.records	请勿更改这些值。
handler.records.metadata	true   false. 如果要排除元数据的输出（因为它没有时间戳信息），请将此值设置为 false。
handler.records.flows	true   false. 如果要排除连接流记录，请将此值设置为 false。
handler.outputters[]	定义 eNcore 所编写内容的行为和格式的一组输出控制器。
outputter.name	为方便起见，这是人类可读的名称。它未在代码中使用。
outputter.adapter	数据从 eStreamer 读取并以结构化内部格式存储。适配器将数据转换为所需的格式。可识别的值包括： <ul style="list-style-type: none"> <li>• Splunk</li> <li>• json</li> </ul>
outputter.enabled	true   false. 您可以一次指定多个输出器。如果要禁用特定输出器，请将此标志设置为 false。如果所有输出器为 false（或没有输出器），则其表现为接收器。
outputter.passthru	true   false. 如果为 true，则流经的数据会绕过解码和元数据处理。它速度非常快，但用途有限。其主要用途是用于调试。
outputter.stream.uri	表示备份的存储位置。您可以指定文件 URI 为正常（例如 file:///absolute/path/to/file）或相对文件路径（relfile:///relative/path/to/file）。 目前仅支持文件 URL。
outputter.stream.options	基于文件的流需要其他选项。
option.rotate	true   false. 设置是否要循环日志。默认配置设置为 true。请注意，eNcore 不会删除任何旧文件。如果您希望执行此操作，则需要单独编写脚本并进行安排。 示例： 从 Cron 作业调用此命令。 #!/bin/bash find /opt/splunk/etc/apps/eStreamer/log/* -mmin +1440 -exec rm {} \;
option.maxLogs	指定日志的大小（行数）。默认配置为 10,000。您可以拥有更少，

密钥	定义
	更大的文件（例如，50,000）。

## 7 故障排除

### 7.1 错误消息

Splunk 插件的 eNcore 旨在提供有意义的错误消息。以下是错误消息示例：

eStreamer 服务已关闭连接-错误日志中可能显示了许多可能的原因。

如果您没有看到错误，则可能是：

- 服务器正在关闭。
- 客户端身份验证失败（请检查您的出站 IP 地址是否与证书关联的出站 IP 地址匹配-请注意，如果您的设备受 NAT 限制，则证书 IP 必须与上游 NAT IP 匹配）。
- 服务器存在问题。如果您运行的是 Firepower 管理中心 v6.0，则可能需要安装“Sourcefire 3D 防御中心 S3 修补程序 AZ 6.1.0.3-1”）。

如果您遇到没有意义或需要进一步解释的错误，请联系支持部门，以便我们修复问题并改进错误消息。

### 7.2 常见的 eNcore 问题

**问题：** 以下信息将帮助您快速解决使用 Firepower eNcore for Splunk TA 的客户遇到的常见问题。从处理许多报告问题来看，常见主题是稳定性、连接性和配置问题。下面的列表介绍了其中几个场景并提供了快速解决方案，但是，如果您仍然遇到问题，请不要 ，如果您在安装 Microsoft Sentinel 代理时遇到问题，请立即创建 TAC 支持请求单

**建议:**

在 Azure 上安装 Microsoft 代理，然后尝试重新安装 OMS

<https://support.microsoft.com/en-us/help/4131455/how-to-reinstall-operations-management-suite-oms-agent-for-linux>

**问题:** 没有数据进入 Splunk。

**建议:**

- 检查位于以下位置的 Splunk TA 的数据目录:

`$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/data/splunk` (默认配置)

- 搜索缺少的记录，常用方法是 `grep`:

`cat "encore*" | grep "rec_type=400"` (400 为入侵事件)

如果未显示结果，则这可能是证书问题，或者您可能正在过滤某些事件类型，请检查 `estreamer.log` 中是否存在任何错误或磁盘故障指示。

此外，请务必检查 `inputs.conf` 文件，以便监控器指向上述数据目录应如下所示 (默认情况下)：

# 数据写入

```
[monitor://$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/data]
```

```
disabled = 0
```

```
source = encore
```

```
sourcetype = cisco:estreamer:data
```

```
crcSalt = <SOURCE>
```

**问题:** 数据过多，实用程序不足以清除我需要的内容。

**建议:**

如果使用 Splunk，则可以将 `input.conf` 监控节更改为批处理，这将在接收时删除文件：

```
[monitor://$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/data]
```

## 更改

### # 数据写入

```
[batch://$SPLUNK_HOME/etc/apps/TA-eStreamer/bin/encore/data]
```

```
disabled = 0
```

```
source = encore
```

```
sourcetype = cisco:estreamer:data
```

```
crcSalt = <SOURCE>
```

重启 Splunk 以使更改生效。

在 CLI/CEF Arcsight 版本上，`./encore.sh` 清理实用程序可以提供一些缓解，但它非常基本，并且操作系统应采用更稳健的文件保留策略、文件轮换、按给定频率清除的 Cron、或 Apache Kafka 是文件/磁盘管理的替代方案。

**问题：** 无法建立连接。没有数据进入 Splunk。

## 建议：

建立连接有几个步骤：

- 已建立可由 Firepower 管理中心访问的客户端证书。 <这里为培训指南链接>
- 证书到位后，您可以运行例程 `./splncore.sh test` 或 `./encore.sh test` 来确定是否启用了连接。客户端仅与其背后的网络一样好，而且通常会禁用通用连接，或者显示 NAT IP 错误。在这些情况下，请从终端向 Firepower 管理中心运行 ping 脚本，以确保网络上没有任何东西阻止连接。请记住，端口 8042 对 eStreamer 协议开放。

我们在 eNcore 数据分布上看到的另一个问题是，数据正在使用 TCP 转发到负载均衡器或通过网络转发。我们强烈建议您将数据本地保存到客户端，并使用专用服务来确保在网络中分布。通过在 `estreamer.conf` 中使用以下变量，可以在“持久”模式下启用 eNcore：

**“alwaysAttemptContinue”： true**

如果负载均衡器终止连接或在网络上重启 eNcore，此模式将自动启动 Python 进程以重新启动通信。

**问题：** 我在突发中收到大量入侵事件，但几小时或几天内都看不到任何入侵事件。



**建议:**

客户端可能处理的事件数量非常少。默认情况下，eNcore 将事件写入磁盘或默认情况下每 100 个事件配置的任何输出流。它旨在通过限制文件 I/O 操作最大限度地提高性能。这可以在配置中使用 estreamer.conf 中的以下变量进行调整:

**“batchSize”: 2。**

将 batchSize 设置为 2 将导致立即传送事件，但会以性能成本为代价，而反之，增加 batchSize 会提高性能，但会延迟将其写入磁盘的时间。如果接收的事件数超过每秒 2000+，这可能很有用，在这种情况下，batchSize 为 500 可能是实施的更好阈值。

此外，在仅发送少量事件时具有高的批处理大小和工作线程数可能会导致问题，因为数据在内存中排队直至达到批处理阈值，这可能意味着数天甚至数周才能将数据发送到 Splunk。

服务器可以支持的工作进程数取决于处理器核心和服务器上的负载的数量和速度。

事件速率 (每秒)	工作线程	批量大小 ( )
小于 100	1	2
100 - 2000	1	100 (默认值)
2000-4000	4	100 (默认值)
4000-6000	8	250
8000+	12	500

**问:** 我是否应将我的批处理大小设置为 1

**答:** 这是本文档先前版本中所提倡的，但是这可能会造成一些破坏性影响。

将批处理大小设置为 1 将立即将来自 Firepower 管理中心的所有信息写入磁盘/数据流，而在某些环境中这可能是首选，当执行 CLEAN 或 monitor 等其他命令时，这可能会导致文件系统死锁为什么我们建议将此值设置为最小值 2，以便在写入过程之间可以对数据存储执行其他操作。

**问题:** 我发现随着时间的 ，事件数据中经常出现间 。当我 制图形时，我会看到一个 图的图表。

**建议:**



虽然目前 `encore.sh` 外壳脚本仅支持一个实例。底层 Python 程序使用主机和端口为临时文件（例如，元数据、证书、书签）添加前缀。您还需要更新输出器位置（例如 `[Splunk]...directory = splunk`），以避免数据冲突。

如果要运行多个实例，建议提取 `eStreamer-eNcore` 的其他副本并单独配置，以避免更改 `encore.sh`。

### 是否可以连接到超过 1 个 Firepower 管理中心

当前不在单个实例中。但是，您可以按上述配置多个实例。

### Firepower 中有哪些不同的记录类型

Firepower 中有超过 500 种记录类型，下面是完整指南的链接，不过这里有一个快速参考表，方便您参考。

记录类型	块类型	系列	说明	记录状态	...中描述的数据格式
2	不适用	不适用	数据包数据（版本 4.8.0.2+）	当前	<a href="#">数据包记录 4.8.0.2+</a>
4	不适用	不适用	优先级元数据	当前	<a href="#">优先级记录</a>
9	20	1	入侵影响警报	传统	<a href="#">入侵影响警报数据</a>
9	153	1	入侵影响警报	当前	<a href="#">入侵影响警报数据 5.3+</a>
62	不适用	不适用	用户元数据	当前	<a href="#">用户记录</a>
66	不适用	不适用	规则消息元数据（版本 4.6.1+）	当前	<a href="#">用于 4.6.1+ 的规则消息记录</a>
67	不适用	不适用	分类元数据（版本 4.6.1+）	当前	<a href="#">用于 4.6.1+ 的分类记录</a>
69	不适用	不适用	关联策略元数据（版本 4.6.1+）	当前	<a href="#">关联策略记录</a>
70	不适用	不适用	关联规则元数据（版本 4.6.1+）	当前	<a href="#">关联规则记录</a>
104	不适用	不适用	入侵事件 (IPv4) 记录 4.9 - 4.10.x	传统	产品的较早版本
105	不适用	不适用	入侵事件 (IPv6)	传统	产品的较早

			记录 4.9 - 4.10.x		版本
110	4	2	入侵事件额外数据 (版本 4.10.0+)	当前	<u>入侵事件额外数据记录</u>
111	5	2	入侵事件额外数据 元数据 (版本 4.10.0+)	当前	<u>入侵事件 额外数据 元数据</u>
112	128	1	用于 5.1-5.3.x 的 关联事件	传统	<u>用于 5.1- 5.3.x 的关 联事件</u>
112	156	1	用于 5.4+ 的关联 事件	当前	<u>用于 5.4+ 的关联事件</u>
115	14	2	安全区名称元数据	当前	<u>安全区名称 记录</u>
116	14	2	接口名称元数据	当前	<u>接口名称 记录</u>
117	14	2	访问控制策略名称 元数据	当前	<u>访问控制策 略名称记录</u>
118	15	2	入侵策略名称元 数据	当前	<u>入侵策略名 称记录</u>
119	15	2	访问控制规则 ID 元数据	当前	<u>访问控制规 则 ID 记录 元数据</u>
120	不适用	不适用	访问控制规则操作 元数据	当前	<u>访问控制规 则操作记录 元数据</u>
121	不适用	不适用	URL 类别元数据	当前	<u>URL 类别记 录元数据</u>
122	不适用	不适用	URL 信誉元数据	当前	<u>URL 信誉记 录元数据</u>
123	不适用	不适用	受管设备记录元 数据	当前	<u>受管设备记 录元数据</u>
125	不适用	2	恶意软件事件记录 (版本 5.1.1+)	当前	<u>恶意软件事 件记录 5.1.1+</u>

125	24	2	恶意软件事件 (版本 5.1.1+)	当前	<u>恶意软件事件数据块</u> 5.1.1.x
125	33	2	恶意软件事件 (版本 5.2.x)	传统	<u>恶意软件事件数据块</u> 5.2.x
125	35	2	恶意软件事件 (版本 5.3)	传统	<u>恶意软件事件数据块</u> 5.3
125	44	2	恶意软件事件 (版本 5.3.1)	传统	<u>恶意软件事件数据块</u> 5.3.1
125	47	2	恶意软件事件 (版本 5.4+)	当前	<u>恶意软件事件数据块</u> 5.4+
127	14	2	综合安全情报云名称元数据 (版本 5.1+)	当前	<u>综合安全情报云名称元数据</u>
128	不适用	不适用	恶意软件事件类型元数据 (版本 5.1+)	当前	<u>恶意软件事件类型元数据</u>
129	不适用	不适用	恶意软件事件子类型元数据 (版本 5.1+)	当前	<u>恶意软件事件子类型元数据</u>
130	不适用	不适用	FireAMP 检测器类型元数据 (版本 5.1+)	当前	<u>FireAMP 检测器类型元数据</u>
131	不适用	不适用	FireAMP 文件类型元数据 (版本 5.1+)	当前	<u>FireAMP 文件类型元数据</u>
132	不适用	不适用	安全情景名称	当前	<u>安全情景名称</u>
207	不适用	不适用	入侵事件 (IPv4) 记录 5.0.x - 5.1	传统	<u>入侵事件 (IPv4) 记录</u> 5.0.x - 5.1
208	不适用	不适用	入侵事件 (IPv6)	传统	<u>入侵事件</u>

			记录 5.0.x - 5.1		<u>(IPv6) 记录 5.0.x - 5.1</u>
260	19	2	ICMP 类型数据 数据块	当前	<u>ICMP 类型数 据块</u>
270	20	2	ICMP 代码数据块	当前	<u>ICMP 代码数 据块</u>
400	34	2	入侵事件记录 5.2.x	传统	<u>入侵事件记录 5.2.x</u>
400	41	2	入侵事件记录 5.3	传统	<u>入侵事件记录 5.3</u>
400	42	2	入侵事件记录 5.3.1	传统	<u>入侵事件记录 5.3.1</u>
400	45	2	入侵事件记录 5.4+	当前	<u>入侵事件记录 5.4+</u>
500	32	2	文件事件 (版本 5.2.x)	传统	<u>用于 5.2 的文 件事件</u>
500	38	2	文件事件 (版本 5.3)	传统	<u>用于 5.3 的文 件事件</u>
500	43	2	文件事件 (版本 5.3.1)	传统	<u>用于 5.3.1 的 文件事件</u>
500	46	2	文件事件 (版本 5.4+)	当前	<u>用于 5.4+ 的 文件事件</u>
502	32	2	文件事件 (版本 5.2.x)	传统	<u>用于 5.2 的文 件事件</u>
502	38	2	文件事件 (版本 5.3)	传统	<u>用于 5.3 的文 件事件</u>
502	43	2	文件事件 (版本 5.3.1)	传统	<u>用于 5.3.1 的 文件事件</u>
502	46	2	文件事件 (版本 5.4+)	当前	<u>用于 5.4+ 的 文件事件</u>

不适用	27	2	用于 5.3+ 的文件事件 SHA 散列	当前	<u>用于 5.3+ 的文件事件 SHA 散列</u>
510	不适用	不适用	用于 5.3+ 的文件类型 ID 元数据	当前	<u>用于 5.2+ 的规则文档数据块</u>
511	40	2	用于 5.3+ 的文件事件 SHA 散列	当前	<u>用于 5.3+ 的文件事件 SHA 散列</u>
520	28	2	用于 5.2+ 的地理位置数据块	当前	<u>用于 5.2+ 的地理位置数据块</u>
530	不适用	不适用	文件策略名称 (File Policy Name)	当前	<u>文件策略名称 (File Policy Name)</u>
600	不适用	不适用	SSL 策略名称	当前	<u>SSL 策略名称</u>
602	不适用	不适用	SSL 密码套件 (SSL Cipher Suite)	当前	<u>SSL 加密套件</u>
604	不适用	不适用	SSL 版本 (SSL Version)	当前	<u>SSL 版本</u>
605	不适用	不适用	SSL 服务器证书状态	当前	<u>SSL 服务器证书状态 (SSL Server Certificate Status)</u>
606	不适用	不适用	SSL 实际操作 (SSL Actual Action)	当前	<u>SSL 实际操作</u>
607	不适用	不适用	SSL 预期操作	当前	<u>SSL 预期操作</u>
608	不适用	不适用	SSL 流状态 (SSL Flow Status)	当前	<u>SSL 流状态</u>
613	不适用	不适用	SSL URL 类别	当前	<u>SSL URL 类别</u>



614	50	2	用于 5.4+ 的 SSL 证书详细信息数据块	当前	<a href="#">用于 5.4+ 的 SSL 证书详细信息数据块</a>
700	不适用	不适用	网络分析策略记录	当前	<a href="#">网络分析策略名称记录</a>

## Firepower 事件 eStreamer 集成指南

<https://www.cisco.com/c/en/us/support/security/defense-center/products-programming-reference-guides-list.html>

## eNcore 是否可以对数据进行重复数据删除以降低 SIEM 成本

不。但是，请了解 Splunk 中的 dedup 命令，并可能创建可消除重复条目的命令，这当然是手动密集型操作，容易出错。

Splunk 重复数据删除事件指南：

[https://](https://docs.splunk.com/Documentation/SCS/current/SearchReference/DedupCommandExamples)

[docs.splunk.com/Documentation/SCS/current/SearchReference/DedupCommandExamples](https://docs.splunk.com/Documentation/SCS/current/SearchReference/DedupCommandExamples)

## 是否可以在高可用性对中运行两个 eNcore 实例

是与否。从技术上讲，可以并行运行两个数据包，但它们将完全相互忽略，并输出两倍的数据。最好在热备份配置中运行它们，其中主客户端的状态和配置数据定期复制到辅助客户端。

有问题的状态和配置数据是 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.da

## 是否可以增加日志记录的 度

是，更改 .conf 文件中的 logging.level 。

请注意，虽然可以将此级别提高到 VERBOSE，但对性能的影响会非常严重。DEBUG 可能有用但很慢。对于标准生产执行，我们强烈建议不要超过 INFO。

# 8 思科支持

支持均由 Cisco TAC 提供

该应用可免费使用，且受社区支持。如有问题，可通过邮件发送至 [encore-community@cisco.com](mailto:encore-community@cisco.com)。

## 9 链接和资源

以下是 CSTA 合作伙伴的完整最新列表：

[https://www.cisco.com/c/m/en\\_us/products/security/technical-alliance-partners.html](https://www.cisco.com/c/m/en_us/products/security/technical-alliance-partners.html)

### 9.1 有用链接

- Splunkbase: <https://splunkbase.splunk.com/>
- 适用于 Firepower 6.x 客户的 eNcore TA: <https://splunkbase.splunk.com/app/3662/>
- Firepower App for Splunk (2019) : <https://splunkbase.splunk.com/app/4388/#/overview>
- 适用于 Firepower 5.4 (2014) 的 FTA 和控制面板: <https://splunkbase.splunk.com/app/1629/>
- FTD TA: <https://splunkbase.splunk.com/app/3955/>
- FTD 控制面板: <https://splunkbase.splunk.com/app/4010/>
- Cisco 安全套件: <https://splunkbase.splunk.com/app/525/>
- 适用于 Firepower 5.4 的 Sourcefire TA: <https://splunkbase.splunk.com/app/1808/>
- eNcore CLI 版本: <https://github.com/CiscoSecurity/fp-05-firepower-cef-connector-arcsight>
- 适用于 Splunk 的 Firepower 应用在 SalesConnect 上的概述:  
<https://www.cisco.com/c/dam/en/us/products/collateral/security/solution-overview-c22-741993.pdf>

## 10 附录

### 10.1 Firepower 管理中心 eStreamer 客户端证书创建

生成 eStreamer 客户端证书的步骤如下:

- 1 转到 Firepower 管理中心的 Web 界面 (<https://fmc-ip-address>)，并使用 Firepower 管理中心凭证登录。
- 2 在 Firepower 管理中心 6.x GUI 中，转至 **系统 (System)** > **集成 (Integration)** > eStreamer。



- 3 点击 **创建客户端 (Create Client)**。
- 4 提供主机名和密码。

**注意：** 这应该是将从 Firepower 管理中心收集事件数据的客户端 IP。首次执行 eStreamer eNcore 时，需要输入您在此处输入的密码。

请注意，从 *Firepower 管理中心的角度来看*，您在此处输入的 IP 地址必须是 eStreamer-eNcore 客户端的 IP 地址。换句话说，如果客户端位于 NAT 设备后面，则 IP 地址必须是上游 NAT 接口的 IP 地址。

- 5 创建客户端主机名和密码屏幕。



6 点击 **保存**。

7 创建客户端保存屏幕。



8 点击右侧的 **下载** 图标，下载 PKCS12 文件。

9 下载屏幕



10 将 PKCS12 文件复制到目标设备中的所需位置。

默认情况下，eStreamer-eNcore 将查找 `/path/eStreamer_eNcore/client.pkcs12`。如果您希望使用其他文件名，则必须编辑 `estreamer.conf` 文件。

## 10.2 示例配置文件

Splunk 插件的 eNcore 随附默认 `estreamer.conf` 文件。为了便于参考，下面提供了一个示例配置文件：

```
{
  "connectTimeout": 10
  "responseTimeout": 10
  "@startComment": "0 代表创    ， 1 代表现在， 2 代表书签”，
  "开始": 2
  "monitor": {
    "period": 120,
    "bookmark": false,
    "handled": true,
    "details": true
  },
  "logging": {
    "@comment": "级别包括 FATAL、ERROR、WARNING、INFO、DEBUG、
    VERBOSE 和 TRACE.",
    "level": "INFO",
    "format": "%(asctime)s %(name)-12s %(levelname)-8s %(message)s",
    "stdOut": true,
```

```

    "filepath" : "estreamer.log"
  },
  "@queueComment" :
    "在限制发生之前缓冲的最大邮件数。越强大”，
    “您的 CPU 和 RAM 越多，此数字越大。它本 上是一个“”，
    "buffer size. 超出特定大小后，您将不会看到任何性能提升，它将“，
    “只是需要更长的时间才能停止”
  ],
  "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" : [
      “只是因为我们订阅并不意味着服务器正在发送。这也不意味着“，
      “我们正在写入记录。请参阅 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":“无论上述内容如何，都将包含这些记录”，
    "include": [],
    "@excludeComment": [
      “这些记录将被排除，无论以上（覆盖'包括'）”，
      “例如，要排除流和 IPS 事件，请使用[71, 400]”
    ],
    "exclude": []
  },
  "@comment":“如果禁用所有输出器，则其表现为接收器”，
  "outputters": [
    {
      "名称": "Splunk default",
      "适配器r": "splunk",
      "enabled": true,
      "stream": {
        "uri": "relfile:///data/splunk/encore.log{0}"
        "options": {
          "rotate": true,
          "maxLogs": 9999
        }
      }
    }
  ],
  {
    "名称": "JSON",
    "适配器": "json",
    "enabled": false,
    "stream": {
      "uri": "relfile:///data/json/log{0}.json",
      "options": {
        "rotate": true,
        "maxLogs": 9999
      }
    }
  }
]
}
}

```



# 11 免责声明

本手册中有关产品的规格和信息如有更改，恕不另行通知。本手册中的所有声明、信息和建议均准确可靠，但我们不为其提供任何明示或暗示的担保。用户必须承担使用产品的全部责任。

随附产品的软件许可和有限担保在随产品一起提供的信息包中提供，且构成本文的一部分。如果您无法找到软件许可或有限担保，请与思科代表联系以获取副本。

思科所采用的 TCP 信头压缩是加州大学伯克莱分校 (UCB) 开发的一个程序的改版，是 UCB 的 UNIX 操作系统公共域版本的一部分。保留所有权利。© 1981，加州大学董事会。

无论在该手册中是否作出了其他担保，来自这些供应商的所有文档文件和软件都按“原样”提供且仍有可能存在缺陷。思科和上述供应商不承诺所有明示或暗示的担保，包括（但不限于）对特定用途的适销性、适用性、非侵权性以及因交易、使用或商业惯例所衍生的担保。

在任何情况下，对于任何间接、特殊、连带发生或偶发的损坏，包括（但不限于）因使用或无法使用本手册而导致的任何利润损失或数据损失或损坏，思科及其供应商概不负责，即使思科及其供应商已获知此类损坏的可能性也不例外。

本档中使用的任何互联网协议 (IP) 地址和电话号码并非实际地址和电话号码。本档中所含的任何示例、命令显示输出、网络拓扑图和其他图形仅供说明之用。说明性内容中用到的任何真实 IP 地址或电话号码纯属巧合，并非有意使用。

所有打印副本和软拷贝均被视为非受控副本，应以原始在线版本为最新版本。

思科在全球设有 200 多个办事处。[www.cisco.com/go/offices](http://www.cisco.com/go/offices) 中列有各办事处的地址、电话和传真。

思科和思科徽标是思科和/或其附属公司在美国和其他国家/地区的商标或注册商标。要查看思科商标列表，请转至此 URL：[www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks)。文中提及的第三方商标为其相应所有者的财产。“合作伙伴”一词的使用并不意味着思科和任何其他公司之间存在合作关系。(1110R)

© 2021 思科系统公司。版权所有。