



配置eStreamer

创建一个客户端应用之后，您可以将其连接至 eStreamer 服务器，启动 eStreamer 服务，开始交换数据。



注释

*eStreamer 服务器*是运行 eStreamer 服务的 管理中心 或受管设备（版本 4.9 或更高版本）。

请执行以下任务以管理 eStreamer 和客户端交互：

1. 在 eStreamer 服务器上启用 eStreamer。
有关允许访问 eStreamer 服务器、添加客户端以及生成身份验证凭证以建立已验证连接的信息，请参阅[在 eStreamer 服务器上配置 eStreamer](#)，第 6-1 页。
2. 如需要，请手动运行 eStreamer 服务(eStreamer)。您可以停止、启动以及查看服务的状态，并使用命令行选项调试客户端-服务器的通信。
有关详细信息，请参阅[管理 eStreamer 服务](#)，第 6-4 页。
3. 或者，要使用 eStreamer 标准客户端对连接或数据流进行故障排除，请在准备用于运行客户端的计算机上设置标准客户端。
请参阅[配置 eStreamer 标准客户端](#)，第 6-5 页。

在 eStreamer 服务器上配置 eStreamer

许可证：任意

在您想要用作 eStreamer 服务器的管理中心或受管设备可以开始将事件流传输到客户端应用之前，您必须配置用于向客户端发送事件的 eStreamer 服务器，提供关于客户端的信息，并生成一套要在建立通信时使用的身份验证凭证。您可以从管理中心或受管设备用户界面执行所有这些任务。

有关详细信息，请参阅以下各节：

- [配置 eStreamer 事件类型](#)，第 6-2 页
- [为 eStreamer 客户端添加身份验证](#)，第 6-3 页

配置 eStreamer 事件类型

许可证：任意

您可以控制 eStreamer 服务器能够向客户端应用传输其所请求的事件的类型。

受管设备或管理中心上的可用事件类型包括：

- 入侵事件
- 入侵事件数据包数据
- 入侵事件额外数据

管理中心上的可用事件类型包括：

- 发现事件（这也会启用连接事件）
- 关联并允许列表事件
- 影响标志警报 (Impact flag alerts)
- 用户活动事件
- 恶意事件
- 文件事件

请注意，堆叠 3D9900 对中的主设备和辅助设备像独立受管设备一样向管理中心报告入侵事件。如果在 3D9900 堆栈中的主设备上配置与 eStreamer 客户端通信，则也需要在辅助设备上配置该客户端；客户端配置不会复制。同样，如果要删除该客户端，请将主设备和辅助设备上的该客户端都删除。如果以堆栈配置为管理 3D9900 的管理中心配置 eStreamer 客户端，请注意，管理中心会报告从两个受管设备收到的所有事件，即使两个设备报告的是同一事件。

如果以高可用性配置在管理中心上配置 eStreamer 客户端，客户端配置将不从主管理中心复制至辅助管理中心。

要配置 eStreamer 捕获的事件类型，请执行以下操作：

访问权限：管理员

步骤 1 选择系统 (System) > 集成 (Integration) > eStreamer。

步骤 2 单击 eStreamer。

系统将显示 eStreamer 页面和 eStreamer 事件配置 (eStreamer Event Configuration) 菜单。

步骤 3 选中想要 eStreamer 捕获并转发至请求客户端的事件类型旁的复选框。请注意，如果现在不选中该复选框，则其对应的数据不会被捕获。取消选中复选框不会删除已捕获的数据。

在管理中心或受管设备上，可选择以下任何或全部事件：

- 入侵事件 (Intrusion Events)，以传输受管设备生成的入侵事件。
- 入侵事件数据包数据 (Intrusion Event Packet Data)，以传输与入侵事件关联的数据包。
- 入侵事件额外数据 (Intrusion Event Extra Data)，以传输与入侵事件关联的额外数据，如与通过 HTTP 代理或负载均衡器连接至 Web 服务器的客户端的源 IP 地址关联的 URI。

在管理中心上，还可选择以下任何或全部事件：

- 发现事件 (Discovery Events)，以传输主机发现事件。
- 关联事件 (Correlation Events)，以传输关联并允许列表事件。
- Impact Flag Alerts，以传输管理中心生成的影响警报。
- 用户活动事件 (User Activity Events)，以传输用户事件。
- 入侵事件额外数据 (Intrusion Event Extra Data)，以传输入侵事件的额外数据，如与通过 HTTP 代理或负载均衡器连接至 Web 服务器的客户端的源 IP 地址关联的 URI。



注释

请注意，这可以控制 eStreamer 服务器可传输的事件。您的客户端应用还必须明确请求您希望其接收的事件类型。有关详细信息，请参阅 [请求标志](#)，第 2-12 页。

步骤 4 点击保存。

系统会保存您的设置，并且在收到请求时，会将您选择的事件转发至 eStreamer 客户端。

为 eStreamer 客户端添加身份验证

许可证：任意

只有先将客户端添加至 eStreamer 服务器的对等数据库，eStreamer 才能向客户端发送事件。还必须将 eStreamer 服务器生成的身份验证证书复制至客户端。

要添加 eStreamer 客户端，请执行以下操作：

访问权限：管理员

步骤 1 选择系统 (System) > 集成 (Integration) > eStreamer。

系统将显示 eStreamer 页面。

步骤 2 点击创建客户端 (Create Client)。

系统将显示“创建客户端”(Create Client) 页面。

步骤 3 在主机名 (Hostname) 字段中，输入运行 eStreamer 客户端的主机的主机名称或 IP 地址。



注释

如果使用主机名，则主机输入服务器必须能够将主机解析为 IP 地址。如果尚未配置 DNS 解析，应先配置解析或使用 IP 地址。

步骤 4 如果要对证书文件进行加密，请在密码 (Password) 字段中输入密码。

步骤 5 点击保存 (Save)。

eStreamer 服务器允许客户端计算机访问管理中心上的 8302 端口，并创建在客户端-服务器身份验证过程中使用的身份验证证书。系统再次显示“eStreamer 客户端”(eStreamer Client) 页面，新的客户端将在 eStreamer 客户端 (eStreamer Client) 下列出。

步骤 6 点击证书文件旁边的下载图标 (↓)。

步骤 7 将证书文件保存至客户端计算机用于 SSL 身份验证的目录。

客户端现在可以连接到管理中心。



提示

要撤消客户端的访问权限，请点击想要移除的主机旁的删除图标 (🗑️)。请注意，无需重启管理中心上的主机输入服务；访问权限将立即撤消。

管理 eStreamer 服务

许可证：任意

您可以从用户界面管理 eStreamer 服务。您也可以使用命令行启动和停止服务。以下部分介绍 eStreamer 命令行选项：

- [启动和停止 eStreamer 服务](#)，第 6-4 页介绍如何启动和停止 eStreamer 服务。
- [eStreamer 服务选项](#)，第 6-4 页介绍可用于 eStreamer 服务的命令行选项及其使用方法。

启动和停止 eStreamer 服务

许可证：任意

您可以用 `manage_estreamer.pl` 脚本管理 eStreamer 服务，通过该脚本，您可以启动、停止、重新加载以及重新启动服务。



提示

您也可以在 eStreamer 初始化脚本中添加命令行选项。有关详细信息，请参阅[eStreamer 服务选项](#)，第 6-4 页。

下表介绍您可以在管理中心或受管设备上使用的 `manage_estreamer.pl` 脚本中的选项。

表 6-1 eStreamer 管理选项

选项	说明	选择选项编号...
enable	启动服务。	3
disable	停止服务。	2
restart	重新启动服务。	4
status	表示服务是否正在运行。	1

eStreamer 服务选项


许可证：任意

eStreamer 提供许多允许您对服务进行故障排除的服务选项。您可以使用下表中描述的服务选项。

表 6-2 eStreamer 服务选项

选项	说明
<code>--debug</code>	运行 eStreamer，并进行调试级日志记录。系统将错误保存到系统日志中并在屏幕上显示错误（与 <code>--nodaemon</code> 一起使用时）。

表 6-2 eStreamer 服务选项

选项	说明
--nodaemon	将 eStreamer 作为前台进程运行。系统在屏幕上显示错误。
--nohostcheck	运行 eStreamer，并禁用主机名检查。即，当客户端主机名与客户端证书 <code>subjectAltName:dNSName</code> 条目中包含的主机名不匹配时，仍允许访问。 <code>nohostcheck</code> 选项在网络 DNS 和/或 NAT 配置阻止主机名成功检查时有用。请注意，系统会执行所有其他安全检查。
	
小心	启用此选项会对您系统的安全性造成不良影响。

通过首先停止 eStreamer 服务，然后用您想要的选项运行该服务，最后重新启动该服务来使用以上选项。例如，您可以按照在调试模式下运行 eStreamer 服务，第 6-5 页中提供的说明调试 eStreamer 功能。

在调试模式下运行 eStreamer 服务

许可证：任意

您可以在调试模式下运行 eStreamer 服务，以查看该服务在您的终端屏幕上生成的所有状态消息。使用以下程序进行调试。

要在调试模式下运行 eStreamer 服务，请执行以下操作：

访问权限：管理员

-
- 步骤 1 用 SSH 登录到管理中心或受控设备。
 - 步骤 2 使用 `manage_estreamer.pl`，并选择选项 2 来停止 eStreamer 服务。
 - 步骤 3 使用 `./usr/local/sf/bin/sfestreamer --nodaemon --debug` 在调试模式下重新启动 eStreamer 服务。
系统在终端屏幕上显示该服务的状态消息。
 - 步骤 4 调试完成后，通过使用 `manage_estreamer.pl` 和选择选项 4 在正常模式下重新启动该服务。
-

配置 eStreamer 标准客户端

配备 eStreamer SDK 的 *标准客户端* 是一组示例客户端脚本和 Perl 模块，以及 Python 脚本，用于说明如何使用 eStreamer API。您可以运行它们以熟悉 eStreamer 输出，或者使用它们调试您的定制客户端的安装问题。

有关设置标准客户端的详细信息，请参阅以下各节：

- [设置 eStreamer 标准客户端，第 6-6 页](#)
- [运行 eStreamer Perl 标准客户端，第 6-11 页](#)
- [运行 eStreamer Python 标准客户端，第 6-12 页](#)

设置eStreamer 标准客户端

要使用eStreamer 标准客户端，必须先配置示例脚本，使其适合您的环境和要求。

有关详细信息，请参阅以下各节：

- [下载eStreamer 标准客户端](#)，第 6-6 页
- [配置用于 eStreamer 标准客户端的通信](#)，第 6-7 页
- [加载用于 Perl 标准客户端的通用前提条件](#)，第 6-8 页
- [加载用于 Perl SNMP 标准客户端的前提条件](#)，第 6-8 页
- [了解 Perl 测试脚本请求的数据](#)，第 6-8 页
- [修改 Perl 测试脚本请求的数据类型](#)，第 6-9 页
- [创建用于标准客户端的证书](#)，第 6-7 页

下载eStreamer 标准客户端

您可以从[思科支持网站](#)下载包含eStreamer 标准客户端文件的 eStreamerSDK.zip 软件包。eStreamerSDK.zip 软件包包含以下文件：

- SF_CUSTOM_ALERT.MIB
snmp.pm 文件用此 MIB 文件为 SNMP 设置陷阱。
- SFRecords.pm
此 Perl 模块包含发现消息记录块的定义。
- SFStreamer.pm
此 Perl 模块包含 Perl 客户端调用的函数。
- SFPkcs12.pm
此 Perl 模块解析客户端证书并允许客户端连接到 eStreamer 服务器。
- SFRNABlocks.pm
此 Perl 模块包含发现数据块的定义。
- ssl_test.pl
您可以使用此 Perl 脚本测试 SSL 连接上的入侵事件请求。
- OutputPlugins/csv.pm
此 Perl 模块以逗号分隔值 (CSV) 格式打印入侵事件。
- OutputPlugins/print.pm
此 Perl 模块以用户可读的格式打印事件。
- OutputPlugins/snmp.pm
此 Perl 模块将事件发送到指定的 SNMP 服务器。
- OutputPlugins/pcap.pm
此 Perl 模块将数据包捕获存储为 pcap 文件。
- python_client/estreamer_client.py
您可以使用此 Python 脚本测试 SSL 连接上的入侵事件请求。
- python_client/estreamer_connection.py
此 Python 脚本会连接到 eStreamer 服务器。它对于 estreamer_client.py 是必需的。

配置用于 eStreamer 标准客户端的通信

标准客户端使用安全套接字层 (SSL) 进行数据通信。您必须在打算用作客户端的计算机上安装 OpenSSL，并根据环境对其进行适当配置。



注释

对于 Linux 操作系统上的初始安装，必须将 libssl-dev 组件作为此下载的一部分进行安装。

要在您的客户端上设置 SSL，请执行以下操作：

- 步骤 1 请从 <http://openssl.org/source/> 下载 OpenSSL。
- 步骤 2 将源解压到 /usr/local/src。
- 步骤 3 通过运行配置脚本来配置源。
- 步骤 4 制作并安装编译源。

创建用于标准客户端的证书

许可证：任意

在使用标准客户端之前，您需要在管理中心或受管设备上为您想要其运行客户端的计算机创建一个证书。然后将该证书文件下载到客户端计算机上，并用它创建证书 (server.crt) 和 RSA 密钥文件 (server.key)。

要创建标准客户端的证书，请执行以下操作：

访问权限：管理员

- 步骤 1 选择系统 (System) > 集成 (Integration) > eStreamer。
系统将显示 eStreamer 页面。
- 步骤 2 点击创建客户端 (Create Client)。
系统将显示“创建客户端”(Create Client) 页面。
- 步骤 3 在主机名 (Hostname) 字段中，输入运行 eStreamer 客户端的主机的主机名称或 IP 地址。



注释

如果使用主机名，则主机输入服务器必须能够将主机解析为 IP 地址。如果尚未配置 DNS 解析，应先配置解析或使用 IP 地址。

- 步骤 4 如果要对证书文件进行加密，请在密码 (Password) 字段中输入密码。
- 步骤 5 单击保存。
eStreamer 服务器允许客户端计算机访问管理中心上的 8302 端口，并创建在客户端-服务器身份验证过程中使用的身份验证证书。系统再次显示“eStreamer 客户端”(eStreamer Client) 页面，新的客户端将在“eStreamer 客户端”(eStreamer Client) 下列出。
- 步骤 6 点击证书文件旁边的下载图标 (↓)。
- 步骤 7 将证书文件保存至客户端计算机用于 SSL 身份验证的目录。
客户端现在可以连接到管理中心。



提示

要撤消客户端的访问权限，请点击想要移除的主机旁的删除图标 (🗑️)。请注意，无需重启管理中心上的主机输入服务；访问权限将立即撤消。

加载用于 Python 标准客户端的通用前提条件

在运行 eStreamer Python 参考客户端之前，您必须???

加载用于 Perl 标准客户端的通用前提条件

在运行 eStreamer Perl 标准客户端之前，必须在客户端计算机上安装 IO::Socket::SSL Perl 模块。您可以手动安装该模块或用 cpan 进行安装。



注释

如果客户端计算机上没有安装 Net::SSLLeay 模块，请也安装该模块。与 OpenSSL 进行通信需要使用 Net::SSLLeay。

您也需要安装并配置 OpenSSL，以支持与 eStreamer 服务器的 SSL 连接。有关详细信息，请参阅 [配置用于 eStreamer 标准客户端的通信，第 6-7 页](#)。

加载用于 Perl SNMP 标准客户端的前提条件

在运行 Perl 标准客户端的 eStreamer SNMP 模块之前，必须先在客户端计算机上安装客户端操作系统可用的最新 net-snmp Perl 模块。

下载并解压缩标准客户端

您可以从 [思科支持网站](#) 下载包含 eStreamer 标准客户端的 EventStreamerSDK.zip 文件。

将压缩文件解压到运行 Linux 操作系统的计算机（打算用于运行客户端的计算机）上。

了解 Perl 测试脚本请求的数据

默认情况下，当您使用标准客户端中的 `ssl_test -o` 设置时，您请求下表中所示的数据。

表 6-3 输出插件进行的默认请求

此语法...	调用插件...	并发送...	以请求以下数据...
<code>./ssl_test.pl eStreamerServerName -h HostIPAddresses</code>	不适用	主机请求，消息类型 5，将位 11 设置为 1	主机数据（请参阅 主机数据和多主机数据消息格式，第 2-28 页 ）
<code>./ssl_test.pl eStreamerServerName -d "Global \ domain \ subdomain"</code>	不适用	指定域或子域的事件流请求。	流传输指定域的事件信息（请参阅 域流传输请求消息格式，第 2-32 页 ）

表 6-3 输出插件进行的默认请求 (续)

此语法...	调用插件...	并发送...	以请求以下数据...
<code>./ssl_test.pl eStreamerServerName -o print -f TextFile</code>	OutputPlugins/ print.pm	事件流请求，消息类型 2，将位 2 和 20-24 设置为 1	事件数据（请参阅事件流请求消息格式，第 2-11 页、关联策略记录，第 3-26 页、关联规则记录，第 3-27 页、发现事件的元数据，第 4-5 页、按事件类型划分的主机发现结构，第 4-42 页和按事件类型划分的用户数据结构，第 4-58 页） eStreamer 传输类型 1 入侵事件，因为已在事件流请求上设置位 2。
<code>./ssl_test.pl eStreamerServerName -o pcap -f TargetPCAPFile</code>	OutputPlugins/ pcap.pm	事件流请求，消息类型 2，将位 0 和 23 设置为 1	数据包数据（请参阅事件数据消息格式，第 2-16 页和数据包记录 4.8.0.2+，第 3-5 页） eStreamer 仅传输数据包数据，因为已在事件流请求上设置位 0。
<code>./ssl_test.pl eStreamerServerName -o csv -f CSVFile</code>	OutputPlugins/ csv.pm	事件流请求，消息类型 2，将位 2 和 23 设置为 1	入侵事件数据（请参阅事件数据消息格式，第 2-16 页和入侵事件记录 7.1+，第 3-7 页） eStreamer 传输类型 1 入侵事件，因为已在事件流请求上设置位 2。
<code>./ssl_test.pl eStreamerServerName -o snmp -f SNMPServer</code>	OutputPlugins/ snmp.pm	事件流请求，消息类型 2，将位 2、20 和 23 设置为 1	入侵事件数据（请参阅事件数据消息格式，第 2-16 页和入侵事件记录 7.1+，第 3-7 页） eStreamer 传输类型 1 入侵事件，因为已在事件流请求上设置位 2。
<code>./ssl_test.pl eStreamerServerName -o syslog</code>	OutputPlugins/ syslog.pm	事件流请求，消息类型 2，将位 2、20 和 23 设置为 1	入侵事件数据（请参阅事件数据消息格式，第 2-16 页和入侵事件记录 7.1+，第 3-7 页） eStreamer 传输类型 1 入侵事件，因为已在事件流请求上设置位 2。
<code>./ssl_test.pl eStreamerServerName json=<filename></code>	不适用	事件流请求，消息类型 2，将位 23 设置为 1，并将所有其他位设置为 0。发送名为 <filename> 的 JSON 文件	提供的 JSON 格式的入侵、连接和文件事件数据。

修改 Perl 测试脚本请求的数据类型

SFStreamer.pm Perl 模块可定义多个您可以在示例脚本中用于请求数据的请求标志变量。下表指出在事件流请求消息中设置每个请求标志需要调用什么请求标志变量。如果您想用其中一个输出模块请求不同的数据，您可以编辑该模块中的 \$FLAG 设置。

有关请求标志及其请求的数据以及与每个标志相对应的产品版本的详细信息，请参阅请求标志，第 2-12 页。

表 6-4 示例脚本中使用的请求标志变量

变量	设置请求标志...	以请求以下数据...
\$FLAG_PKTS	0	数据包数据
\$FLAG_METADATA	1	版本 1 元数据
\$FLAG_IDS	2	类型 1 入侵事件
\$FLAG_RNA	3	版本 1 发现事件
\$FLAG_POLICY_EVENTS	4	版本 1 关联事件
\$FLAG_IMPACT_ALERTS	5	入侵影响警报
\$FLAG_IDS_IMPACT_FLAG	6	类型 7 入侵事件
\$FLAG_RNA_EVENTS_2	7	版本 2 发现事件
\$FLAG_RNA_FLOW	8	版本 1 连接数据
\$FLAG_POLICY_EVENTS_2	9	版本 2 关联事件
\$FLAG_RNA_EVENTS_3	10	版本 3 发现事件
\$FLAG_HOST_ONLY	11	与 \$FLAG_HOST_SINGLE（用于一个主机）或 \$FLAG_HOST_MULTI（用于多个主机）一起发送时，只有主机数据，无事件数据
\$FLAG_RNA_FLOW_3	12	版本 3 连接数据
\$FLAG_POLICY_EVENTS_3	13	版本 3 关联事件
\$FLAG_METADATA_2	14	版本 2 元数据
\$FLAG_METADATA_3	15	版本 3 元数据
\$FLAG_RNA_EVENTS_4	17	版本 4 发现事件
\$FLAG_RNA_FLOW_4	18	版本 4 连接数据
\$FLAG_POLICY_EVENTS_4	19	版本 4 关联事件
\$FLAG_METADATA_4	20	版本 4 元数据
\$FLAG_RUA	21	用户活动事件
\$FLAG_POLICY_EVENTS_5	22	版本 5 关联事件
\$FLAGS_SEND_ARCHIVE_TIMESTAMP	23	包含将事件存档供 eStreamer 服务器处理时应用的时间戳的扩展事件报头
\$FLAG_RNA_EVENTS_5	24	版本 5 发现事件
\$FLAG_RNA_EVENTS_6	25	版本 6 发现事件
\$FLAG_RNA_FLOW_5	26	版本 5 连接数据
\$FLAG_EXTRA_DATA	27	入侵事件额外数据记录
\$FLAG_RNA_EVENTS_7	28	版本 7 发现事件
\$FLAG_POLICY_EVENTS_6	29	版本 6 关联事件
\$FLAG_DETAIL_REQUEST	30	向 eStreamer 发出的扩展请求



小心

在所有事件类型中，在版本 5.x 之前，标准客户端都将 detection engine ID 字段标记为 sensor ID。

运行 eStreamer Perl 标准客户端

eStreamer Perl 标准客户端脚本设计用于配备 Linux 内核的 64 位操作系统，但是，只要客户端计算机满足设置 eStreamer 标准客户端，第 6-6 页中规定的前提条件，该标准客户端应该可以在任何基于 POSIX 的 64 位操作系统上使用。

有关详细信息，请参阅以下各节：

- 用主机请求测试经由 SSL 的客户端连接，第 6-11 页
- 用标准客户端捕获 PCAP，第 6-11 页
- 用标准客户端捕获 CSV 记录，第 6-11 页
- 用标准客户端将记录发送到 SNMP 服务器，第 6-12 页
- 用标准客户端将事件记录到系统日志中，第 6-12 页
- 连接到 IPv6 地址，第 6-12 页

用主机请求测试经由 SSL 的客户端连接

您可以使用 `ssl_test.pl` 脚本测试 eStreamer 服务器与 eStreamer 客户端之间的连接。`ssl_test.pl` 脚本可处理任何记录类型并将其打印到 **STDOUT** 或您指定的输出插件。当您使用不具有输出选项的 `-h` 选项时，它会将指定主机的主机数据流传输到您的终端。



注释

如果不将数据包数据定向到输出插件，则无法使用此脚本来流传输数据包数据，因为将原始数据包数据打印到 **STDOUT** 会干扰您的终端。

通过以下语法用 `ssl_test.pl` 脚本将主机数据发送到标准输出：

```
./ssl_test.pl eStreamerServerIPAddress -h HostIPAddresses
```

例如，通过与 IP 地址为 10.10.0.4 的 eStreamer 服务器的连接测试 10.0.0.0/8 子网中主机的主机数据接收情况：

```
./ssl_test.pl 10.10.0.4 -h 10.0.0.0/8
```

用标准客户端捕获 PCAP

您可以用标准客户端捕获 PCAP 文件中流传输的数据包数据，以查看客户端接收的数据的结构。请注意，使用 `-o pcap` 输出选项时，必须使用 `-f` 来指定目标文件。

通过以下语法用 `ssl_test.pl` 脚本捕获 PCAP 文件中流传输的数据包：

```
./ssl_test.pl eStreamerServerIPAddress -o pcap -f ResultingPCAPFile
```

例如，用 IP 地址为 10.10.0.4 的 eStreamer 服务器流传输的事件创建名为 `test.pcap` 的 PCAP 文件：

```
./ssl_test.pl 10.10.0.4 -o pcap -f test.pcap
```

用标准客户端捕获 CSV 记录

您也可以使用标准客户端捕获 CSV 文件中流传输的入侵事件数据，以查看客户端接收的数据的结构。使用以下语法运行 `streamer_csv.pl` 脚本：

```
./ssl_test.pl eStreamerServerIPAddress -o csv -f ResultingCSVFile
```

例如，用 IP 地址为 10.10.0.4 的 eStreamer 服务器流传输的事件创建名为 `test.csv` 的 CSV 文件：

```
./ssl_test.pl 10.10.0.4 -o csv -f test.csv
```

用标准客户端将记录发送到 SNMP 服务器

您也可以使用标准客户端将入侵事件数据流传输到 SNMP 服务器。使用 `-f` 选项指示应接收事件的 SNMP 陷阱服务器的名称。请注意，此输出方法需要路径中有一个名为 `snmptrapd` 的二进制文件，因此只能用于 UNIX 类系统。

使用以下语法将入侵事件发送到 SNMP 服务器：

```
./ssl_test.pl eStreamerServerIPAddress -o snmp -f SNMPServerName
```

例如，用 IP 地址为 10.10.0.4 的 eStreamer 服务器流传输的事件将事件发送到 IP 地址为 10.10.0.3 的 SNMP 服务器：

```
./ssl_test.pl 10.10.0.4 -o snmp -f 10.10.0.3
```

用标准客户端将事件记录到系统日志中

您也可以使用标准客户端将入侵事件流传输到客户端上的本地系统日志服务器。

使用以下语法将事件发送到系统日志：

```
./ssl_test.pl eStreamerServerIPAddress -o syslog
```

例如，记录 IP 地址为 10.10.0.4 的 eStreamer 服务器流传输的事件：

```
./ssl_test.pl 10.10.0.4 -o syslog
```

连接到 IPv6 地址

您可以使用标准客户端通过主管理接口连接到具有 IPv6 地址的管理中心。必须在客户端计算机上安装 `Socket6` 和 `IO::Socket::INET6` Perl 模块，并使用 `-ipv6` 选项或其简称 `-i`。

通过以下语法用 `ssl_test.pl` 脚本指定 IPv6 地址：

```
./ssl_test.pl -ipv6 eStreamerServerIPAddress
```

或

```
./ssl_test.pl -i eStreamerServerIPAddress
```

例如，要连接到 IPv6 地址为 `2001:470:e09c:20:7c1e:5248:1bf7:2ea0` 的管理中心，请使用以下命令：

```
./ssl_test.pl -ipv6 2001:470:e09c:20:7c1e:5248:1bf7:2ea0
```

运行 eStreamer Python 标准客户端

eStreamer Python 标准客户端脚本演示了一种从 Cisco Secure Firewall 系统管理中心 eStreamer 服务获取事件数据的更简单的新机制。事件信息不会以二进制数据形式返回，而是以 JSON 或 CSV 等格式的完全限定文本形式返回。

此 API 仅支持请求三种事件类型的信息：连接事件、入侵事件和文件事件。对于所有其他事件，您必须使用单独的客户端和《eStreamer 集成指南》中介绍的常规方法。

Python 代码提供了一个使用新机制的简单客户端示例。Perl 示例客户端代码也已修改为可选择性地使用此新机制（使用 `json=<filename>` 命令行参数），但 Python 示例更容易理解，因为它仅支持新机制。

示例用法:

```
./estreamer_client.py --server 192.168.1.1 --configfile json_request.json --pkcs12_file 192.168.1.2_8.pkcs12 --start all
```

表 6-5 Python 脚本参数

此参数...	以下...
-h, --help	是否显示了此帮助消息并退出。
--server SERVER	指定 eStreamer 服务器的 IP 地址。此 IP 地址必须可从运行客户端的计算机进行访问。
--port PORT	指定 eStreamer 服务器的端口。默认值为 8302
--configfile CONFIGFILE	提供 JSON 格式的配置文件。有关详细信息，请参阅 JSON 文件的格式，第 2-5 页 。
--pkcs12_file PKCS12_FILE	向 eStreamer 服务器提供用于身份验证的 Pkcs12 文件。
--pkcs12_password PKCS12_PASSWORD	如有必要，提供 Pkcs12 密码。
--debug	启用调试模式。
--start {now,all,bookmark}	流媒体事件的开始时间
--outfile OUTFILE	用于存储事件的输出文件。默认值为打印到标准输出

当地语言翻译版本说明

思科可能会在某些地方提供本内容的当地语言翻译版本。请注意，翻译版本仅供参考，如有任何不一致之处，以本内容的英文版本为准。