

对私有云上的事件流进行故障排除

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[配置](#)

[创建API密钥](#)

[创建事件流](#)

[MacOS/Linux](#)

[Windows 窗口版本](#)

[回复](#)

[事件流列表](#)

[MacOS/Linux](#)

[Windows 窗口版本](#)

[回复](#)

[删除事件流](#)

[MacOS/Linux](#)

[Windows 窗口版本](#)

[回复](#)

[验证](#)

[故障排除](#)

[检查AMQP服务](#)

[检查到事件流接收器的连接](#)

[检查队列中的事件](#)

[收集网络流量文件](#)

[相关信息](#)

简介

本文档介绍如何对高级恶意软件防护安全终端私有云中的事件流进行故障排除。

先决条件

要求

思科建议您了解以下主题：

- 安全终端私有云
- API查询

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 安全终端私有云v3.9.0
- cURL v7.87.0
- cURL v8.0.1

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

配置

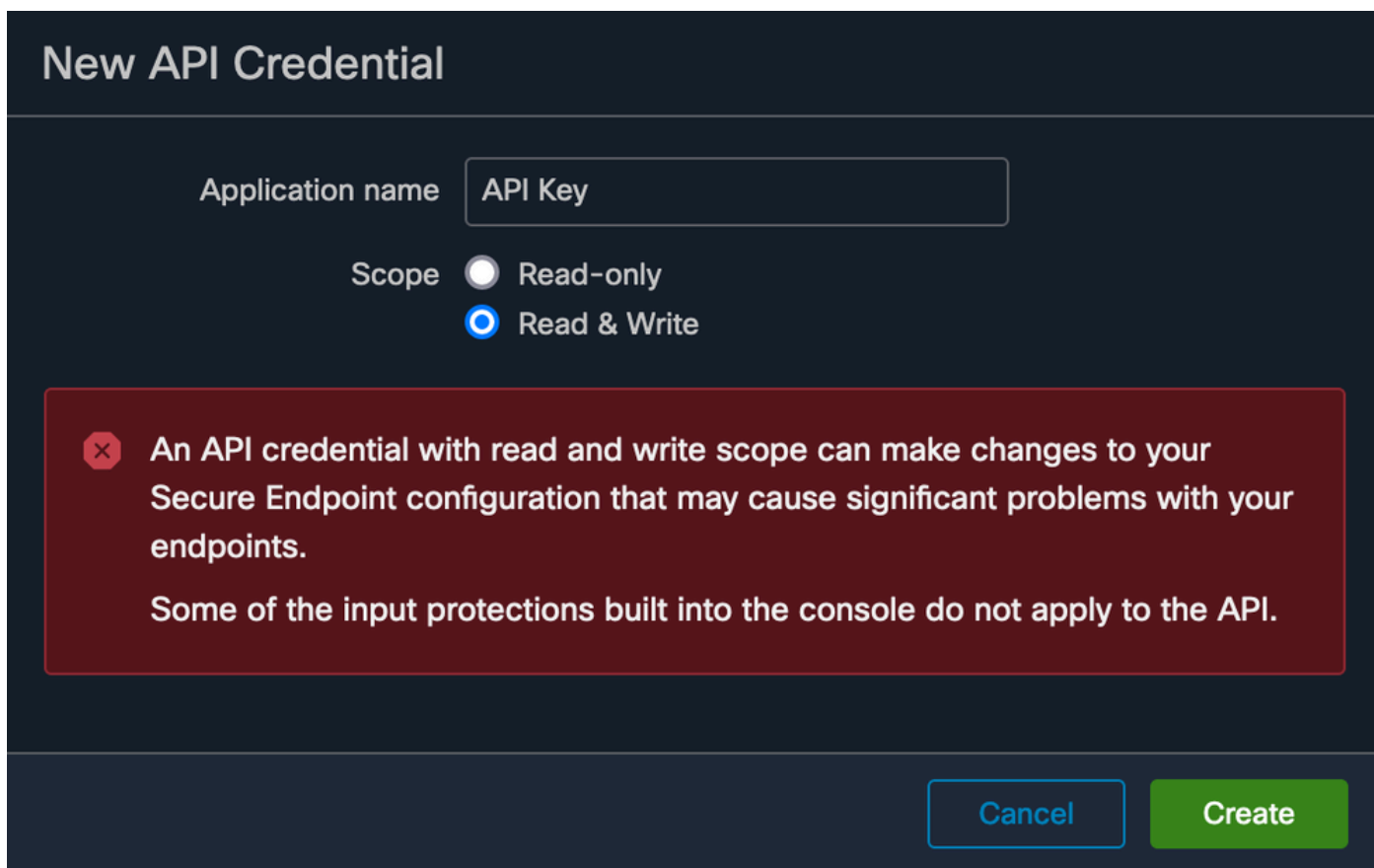
创建API密钥

步骤1:登录私有云控制台。

第二步：导航至 `Accounts > API Credentials`。

第三步：点击 `New API Credential`。

第四步：添加 `Application name` 并点击 `Read & Write` 范围。



New API Credential

Application name

Scope Read-only Read & Write

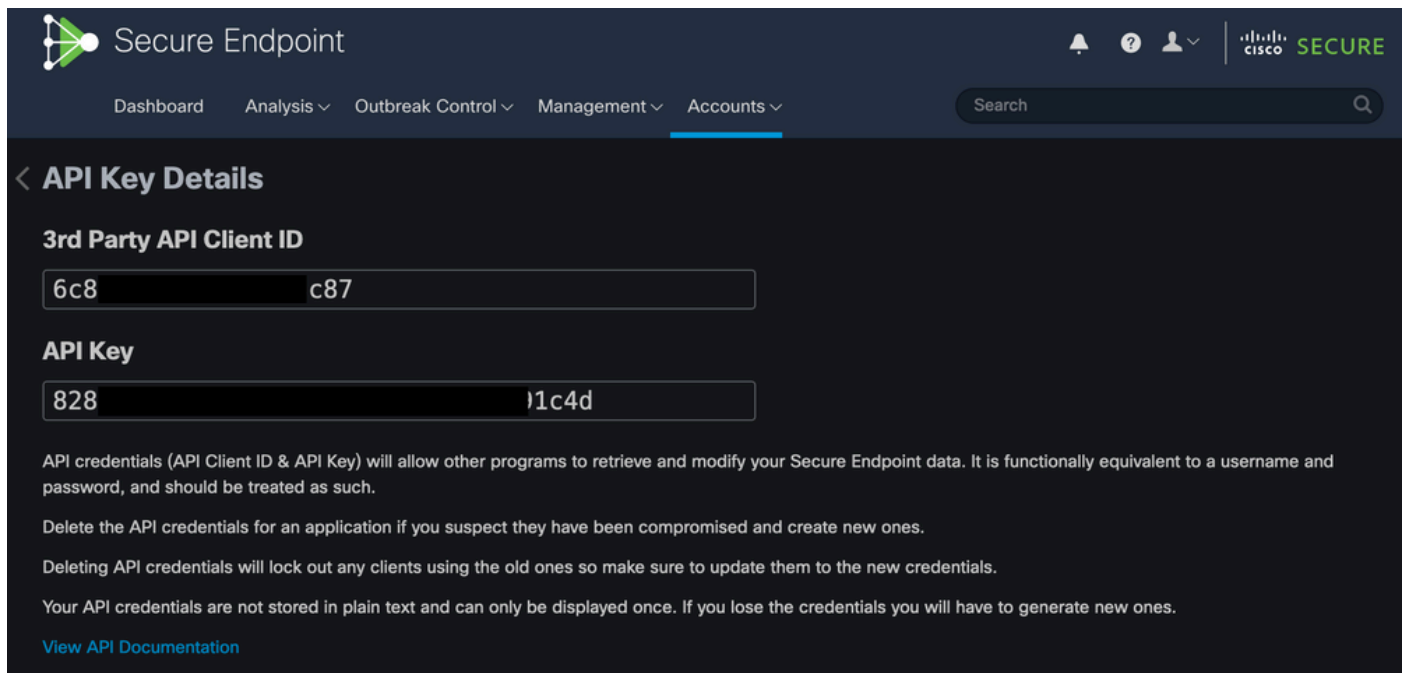
⊗ An API credential with read and write scope can make changes to your Secure Endpoint configuration that may cause significant problems with your endpoints.

Some of the input protections built into the console do not apply to the API.

创建API密钥

第五步：点击 `Create`。

第六步：保存API凭证。



Secure Endpoint

Dashboard Analysis Outbreak Control Management Accounts

Search

< API Key Details

3rd Party API Client ID

6c8c87

API Key

8281c4d

API credentials (API Client ID & API Key) will allow other programs to retrieve and modify your Secure Endpoint data. It is functionally equivalent to a username and password, and should be treated as such.

Delete the API credentials for an application if you suspect they have been compromised and create new ones.

Deleting API credentials will lock out any clients using the old ones so make sure to update them to the new credentials.

Your API credentials are not stored in plain text and can only be displayed once. If you lose the credentials you will have to generate new ones.

[View API Documentation](#)

API密钥

注意：如果您离开此页面，则无法恢复API密钥。

创建事件流

这将为事件信息创建新的高级消息队列协议(AMQP)消息流。

可以为指定的事件类型和组创建事件流：

```
--data '{"name":"EVENT_STREAM_NAME","event_type":["EVENT_TYPE_1", "EVENT_TYPE_2"],"group_guid":["GROUP_1", "GROUP_2"]}'
```

您可以通过以下方式在所有事件类型和所有组创建事件流：

```
--data '{"name":"EVENT_STREAM_NAME","event_type":[],"group_guid":[]}'
```

MacOS/Linux

您可以使用以下功能在MacOS/Linux上创建事件流：

```
curl -X POST -k -H 'Accept: application/json' -H 'Content-Type: application/json' -u 'CLIENT_ID:API_KEY'
```

Windows 窗口版本

您可以使用以下功能在Windows上创建事件流：

```
curl -X POST -k -H "Accept: application/json" -H "Content-Type: application/json" -u "CLIENT_ID:API_KEY"
```

回复

```
HTTP/1.1 201 Created
```

```
(...)
```

```
"data": {  
  "id": 17,  
  "name": "EVENT_STREAM_NAME",  
  "amqp_credentials": {  
    "user_name": "17-1bfXXXXXXXXXX",  
    "queue_name": "event_stream_17",  
    "password": "3961XXXXXXXXXXXXXXXXXXXXXXXX814a77",  
    "host": "FMC_SERVICE_URL",  
    "port": 443,  
    "proto": "https"  
  }  
}
```

事件流列表

这显示在私有云上创建的事件流的列表。

MacOS/Linux

您可以使用以下命令列出MacOS/Linux上的事件流：

```
curl -k -H 'Accept: application/json' -H 'Content-Type: application/json' -u 'CLIENT_ID:API_KEY' -i 'ht
```

Windows 窗口版本

您可以使用Windows上的Event Streams列出以下内容：

```
curl -k -H "Accept:application/json" -H "Content-Type:application/json" -u "CLIENT_ID:API_KEY" -i "http
```

回复

HTTP/1.1 200 OK

(...)

```
"data": {  
  "id": 17,  
  "name": "EVENT_STREAM_NAME",  
  "amqp_credentials": {  
    "user_name": "17-1bfXXXXXXXXXX",  
    "queue_name": "event_stream_17",  
    "host": "FMC_SERVICE_URL",  
    "port": 443,  
    "proto": "https"  
  }  
}
```

删除事件流

删除活动事件流。

MacOS/Linux

您可以使用以下方式删除MacOS/Linux上的事件流：

```
curl -X DELETE -k -H 'Accept: application/json' -H 'Content-Type: application/json' -u 'CLIENT_ID:API_KEY'
```

Windows 窗口版本

您可以使用以下方法删除Windows上的事件流：

```
curl -X DELETE -k -H "Accept:application/json" -H "Content-Type:application/json" -u "CLIENT_ID:API_KEY"
```

回复

HTTP/1.1 200 OK

(...)

```
"data": {}
```

验证

步骤1:将Python脚本复制到您的设备并将其另存为 `EventStream.py`.

```
import pika
import ssl

user_name = "USERNAME"
queue_name = "QUEUE_NAME"
password = "PASSWORD"
host = "FMC_SERVICE_URL"
port = 443
proto = "https"

def callback(channel, method, properties, body):
    print(body)

amqp_url = f"amqps://{user_name}:{password}@{host}:{port}"

context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
amqp_ssl = pika.SSLOptions(context)

params = pika.URLParameters(amqp_url)
params.ssl_options = amqp_ssl

connection = pika.BlockingConnection(params)
channel = connection.channel()

channel.basic_consume(
    queue_name,
    callback,
    auto_ack = False
)

channel.start_consuming()
```

第二步：在终端中执行它作为 `python3 EventStream.py`.

第三步：触发添加到Event Stream队列中的任何事件。

第四步：检查终端中是否显示事件。

故障排除

要执行这些命令，必须通过SSH登录到私有云。

检查AMQP服务

验证服务是否已启用：

```
[root@fireamp rabbitmq]# amp-ctl service status rabbitmq
running enabled rabbitmq
```

验证服务是否正在运行：

```
[root@fireamp ~]# svstat /service/rabbitmq
/service/rabbitmq: up (pid 25504) 7402137 seconds
```

检查到事件流接收器的连接

执行命令：

```
tail /data/log/rabbitmq/rabbit@fireamp.log
```

已建立连接：

```
=INFO REPORT==== 19-Apr-2023::08:40:12 ===
accepting AMQP connection <0.17588.27> (127.0.0.1:32946 -> 127.0.0.1:5672)
```

连接已关闭：

```
=WARNING REPORT==== 19-Apr-2023::08:41:52 ===
closing AMQP connection <0.17588.27> (127.0.0.1:32946 -> 127.0.0.1:5672):
connection_closed_abruptly
```

检查队列中的事件

建立连接后，队列中的事件已准备就绪，可以按此事件流发送到接收方。在本示例中，事件流ID 23有14个事件。

<#root>

```
[root@fireamp rabbitmq]# rabbitmqctl list_queues
Listing queues ...
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_60b15rn8mpftai6or6l8zxav11usm 26
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_61984nlu8p11eeopmgmtcjra1v8gf5p 26
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_iesRAGVo0h287m0_Det0x9PdDu8MxkS6kL4oSTeBm9s 26
```

```
event_decoration 0
event_log_store 0

event_stream_23 14
```

```
event_streams_api 0
events_delayed 0
events_retry 0
mongo_event_consumer 0
out_events_q1 0
tevent_listener 0
```

收集网络流量文件

为了验证来自私有云的事件流流量，您可以使用 `tcpdump` 工具：

步骤1:通过SSH连接到私有云。

第二步：执行命令：

```
tcpdump -vvv -i eth1 host <Event_Stream_Receiver_IP> -w file.pcap
```

第三步：停止捕获 `Ctrl+C` (Windows)或 `Command-C` (Mac)。

第四步：提取 `pcap` 文件。

相关信息

- [配置面向终端的AMP事件流功能](#)
- [技术支持和文档 - Cisco Systems](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。