

# Solucionar problemas de fluxo de eventos na nuvem privada

## Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Configuração](#)

[Criar chave de API](#)

[Criar Fluxo de Eventos](#)

[MacOS/Linux](#)

[Windows](#)

[Resposta](#)

[Lista de fluxos de eventos](#)

[MacOS/Linux](#)

[Windows](#)

[Resposta](#)

[Excluir Fluxos de Eventos](#)

[MacOS/Linux](#)

[Windows](#)

[Resposta](#)

[Verificar](#)

[Troubleshooting](#)

[Verificar o Serviço AMQP](#)

[Verifique a conexão com o receptor de fluxo de eventos](#)

[Verificar os Eventos na Fila](#)

[Coletar arquivo de tráfego de rede](#)

[Informações Relacionadas](#)

## Introdução

Este documento descreve como solucionar problemas de Fluxos de Eventos na Nuvem Privada de Endpoint Seguro da Proteção Avançada contra Malware.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento dos seguintes tópicos:

- Nuvem privada de endpoint segura
- Consulta API

## Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Nuvem privada de endpoint segura v3.9.0
- cURL v7.87.0
- cURL v8.0.1

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Configuração

### Criar chave de API

Etapa 1. Faça login no console da nuvem privada.

Etapa 2. Navegue até `Accounts > API Credentials`.

Etapa 3. Clique em `New API Credential`.

Etapa 4. Adicione o comando `Application name` e clique em `Read & Write` escopo.

# New API Credential

Application name

API Key

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.

Cancel

Create

Criar chave de API

Etapa 5. Clique em **Create**.

Etapa 6. Salvar credenciais de API.

The screenshot shows the Cisco Secure Endpoint console interface. At the top, there is a navigation bar with the following items: Dashboard, Analysis, Outbreak Control, Management, and Accounts (which is currently selected). A search bar is located on the right side of the navigation bar. Below the navigation bar, the main content area is titled "API Key Details". Under this title, there are two sections: "3rd Party API Client ID" and "API Key". Each section has a text input field containing a hexadecimal string. The "3rd Party API Client ID" field contains "6c8" followed by "c87". The "API Key" field contains "828" followed by "1c4d". Below these fields, there is a paragraph of text explaining that API credentials allow other programs to retrieve and modify Secure Endpoint data, and that they should be treated as a username and password. It also includes instructions on how to delete and regenerate API credentials. At the bottom of the page, there is a link to "View API Documentation".

Chave de API

---

Cuidado: a chave de API não poderá ser recuperada se você sair desta página.

---

## Criar Fluxo de Eventos

Isso cria um novo fluxo de mensagens AMQP (Advanced Message Queuing Protocol) para obter informações sobre eventos.

Você pode criar um Fluxo de Eventos para tipos e grupos de eventos especificados:

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

Você pode criar um Fluxo de Eventos para todos os tipos de eventos e todos os grupos:

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

## MacOS/Linux

Você pode criar um fluxo de eventos em MacOS/Linux com o uso de:

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

## Windows

Você pode criar um fluxo de eventos no Windows com o uso de:

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

## Resposta

HTTP/1.1 201 Created

(...)

```
"data": {  
  "id": 17,  
  "name": "EVENT_STREAM_NAME",  
  "amqp_credentials": {  
    "user_name": "17-1bfXXXXXXXXXX",
```

```
    "queue_name": "event_stream_17",
    "password": "3961XXXXXXXXXXXXXXXXXXXXXXXXXXXX814a77",
    "host": "FMC_SERVICE_URL",
    "port": 443,
    "proto": "https"
  }
}
```

## Lista de fluxos de eventos

Isso mostra uma lista de fluxos de eventos criados na nuvem privada.

### MacOS/Linux

Você pode listar os Fluxos de Eventos no MacOS/Linux com o uso de:

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

### Windows

Você pode listar os Fluxos de Eventos no Windows com o uso de:

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

## Resposta

```
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"
  }
}
```

## Excluir Fluxos de Eventos

Exclui um fluxo de eventos ativo.

## MacOS/Linux

Você pode excluir Fluxos de Eventos no MacOS/Linux com o uso de:

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

## Windows

Você pode excluir Fluxos de Eventos no Windows com o uso de:

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

## Resposta

```
HTTP/1.1 200 OK  
(...)  
"data": {}
```

## Verificar

Etapa 1. Copie o script Python no seu dispositivo e salve-o como `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()
```

Etapa 2. Execute-o no terminal como `python3 EventStream.py`.

Etapa 3. Dispare qualquer evento que seja adicionado à fila do Fluxo de Eventos.

Etapa 4. Verifique se os eventos aparecem no terminal.

## Troubleshooting

Para executar esses comandos, você deve fazer login via SSH na nuvem privada.

### Verificar o Serviço AMQP

Verifique se o serviço está habilitado:

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

Verifique se o serviço está em execução:

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

### Verifique a conexão com o receptor de fluxo de eventos

Execute o comando:

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

Conexão estabelecida:

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

A conexão está fechada:

```
=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
```

## Verificar os Eventos na Fila

Os eventos na fila estão prontos para serem enviados nesse fluxo de eventos para o receptor após o estabelecimento da conexão. Neste exemplo, há 14 eventos para a ID do fluxo de eventos 23.

<#root>

```
[root@fireamp rabbitmq]# rabbitmqctl list_queues  
Listing queues ...  
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_60b15rn8mpftaico6or6l8zxav1lusm 26  
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_61984nlu8p11eeopmgmtcjra1v8gf5p 26  
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_iesRAgVo0h287mO_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
```

## Coletar arquivo de tráfego de rede

Para verificar o tráfego do fluxo de eventos da nuvem privada, você pode coletar a captura com um `tcpdump` ferramenta:

Etapa 1. SSH na nuvem privada.

Etapa 2. Execute o comando:



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

Etapa 3. Parar a captura com `Ctrl+C` (Windows) ou `Command-C` (Mac)

Etapa 4. Extraia o `pcap` da nuvem privada.

## Informações Relacionadas

- [Configurar o AMP para o recurso de fluxo de eventos de endpoints](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.