

Fehlerbehebung für Event Stream in der Private Cloud

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konfiguration](#)

[API-Schlüssel erstellen](#)

[Ereignisstream erstellen](#)

[MacOS/Linux](#)

[Windows](#)

[Antwort](#)

[Liste der Event Streams](#)

[MacOS/Linux](#)

[Windows](#)

[Antwort](#)

[Löschen von Ereignisströmen](#)

[MacOS/Linux](#)

[Windows](#)

[Antwort](#)

[Überprüfung](#)

[Fehlerbehebung](#)

[Überprüfen Sie den AMQP-Service](#)

[Überprüfen Sie die Verbindung zum Event Stream Receiver.](#)

[Suchen nach Ereignissen in der Warteschlange](#)

[Netzwerkdatenverkehrsdatei erfassen](#)

[Zugehörige Informationen](#)

Einleitung

In diesem Dokument wird die Fehlerbehebung bei Event Streams in Advanced Malware Protection Secure Endpoint Private Cloud beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie die folgenden Themen kennen:

- Sichere Endpunkt-Private-Cloud
- API-Abfrage

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Secure Endpoint Private Cloud v3.9.0
- cURL v7.87.0
- cURL v8.0.1

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Konfiguration

API-Schlüssel erstellen

Schritt 1: Melden Sie sich bei der Private Cloud-Konsole an.

Schritt 2: Navigieren Sie zu `Accounts > API Credentials`.

Schritt 3: Klicken Sie auf `New API Credential`.

Schritt 4: Fügen Sie `Application name` und klicke auf `Read & Write Umfang`.

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

API-Schlüssel erstellen

Schritt 5: Klicken Sie auf **Create**.

Schritt 6: API-Anmeldeinformationen speichern.

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 displays the 'API Key Details' page. The page has a back arrow on the left and the title 'API Key Details'. Under the 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 a space and 'c87'. The 'API Key' field contains '828' followed by a space and '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 provides instructions on how to delete and regenerate API credentials. At the bottom of the page, there is a link to 'View API Documentation'.

API-Schlüssel

Achtung: Der API-Schlüssel kann nicht wiederhergestellt werden, wenn Sie diese Seite verlassen.

Ereignisstream erstellen

Dadurch wird ein neuer AMQP-Meldungsstream (Advanced Message Queuing Protocol) für Ereignisinformationen erstellt.

Sie können eine Ereignisüberwachung für bestimmte Ereignistypen und Gruppen erstellen:

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

Sie können eine Ereignisüberwachung für alle Ereignistypen und Gruppen erstellen, indem Sie:

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

MacOS/Linux

Sie können einen Event Stream unter MacOS/Linux erstellen, indem Sie:

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

Windows

Sie können unter Windows eine Ereignisüberwachung erstellen, indem Sie Folgendes verwenden:

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

Antwort

HTTP/1.1 201 Created

(...)

```
"data": {  
  "id": 17,  
  "name": "EVENT_STREAM_NAME",  
  "amqp_credentials": {
```

```
    "user_name": "17-1bfXXXXXXXXXX",
    "queue_name": "event_stream_17",
    "password": "3961XXXXXXXXXXXXXXXXXXXXXXXXX814a77",
    "host": "FMC_SERVICE_URL",
    "port": 443,
    "proto": "https"
  }
}
```

Liste der Event Streams

Hier sehen Sie eine Liste der Event-Streams, die in der Private Cloud erstellt wurden.

MacOS/Linux

Sie können die Event Streams unter MacOS/Linux auflisten, indem Sie:

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

Windows

Sie können die Ereignisstreams unter Windows auflisten, indem Sie Folgendes verwenden:

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

Antwort

```
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"
  }
}
```

Löschen von Ereignisströmen

Löscht einen aktiven Ereignisstream.

MacOS/Linux

Sie können Event Streams unter MacOS/Linux löschen, indem Sie:

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

Windows

Sie können Ereignisstreams unter Windows löschen, indem Sie:

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

Antwort

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

Überprüfung

Schritt 1: Kopieren Sie das Python-Skript auf Ihr Gerät, und speichern Sie es unter `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()
```

Schritt 2: Führen Sie es im Terminal aus als `python3 EventStream.py`.

Schritt 3: Trigger für alle Ereignisse, die der Ereignisdatenverkehrwarteschlange hinzugefügt werden.

Schritt 4: Überprüfen Sie, ob die Ereignisse im Terminal angezeigt werden.

Fehlerbehebung

Um diese Befehle auszuführen, müssen Sie sich über SSH bei der Private Cloud anmelden.

Überprüfen Sie den AMQP-Service

Überprüfen Sie, ob der Dienst aktiviert ist:

```
[root@fireamp rabbitmq]# ampctl service status rabbitmq
running enabled rabbitmq
```

Überprüfen Sie, ob der Dienst ausgeführt wird:

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

Überprüfen Sie die Verbindung zum Event Stream Receiver.

Führen Sie den folgenden Befehl aus:

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

Verbindung wird hergestellt:

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

Verbindung ist geschlossen:

```
=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
```

Suchen nach Ereignissen in der Warteschlange

Ereignisse in der Warteschlange können nach dem Herstellen der Verbindung über diesen Ereignisstream an den Empfänger gesendet werden. In diesem Beispiel gibt es 14 Ereignisse für die Ereignisstream-ID 23.

<#root>

```
[root@fireamp rabbitmq]# rabbitmqctl list_queues  
Listing queues ...  
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_60b15rn8mpftaico6or6l8zxav1l1usm 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
```

Netzwerkdatenverkehrsdatei erfassen

Um den Ereignisstream-Datenverkehr aus der Private Cloud zu überprüfen, können Sie die Erfassung mit einem `tcpdump` Tool:

Schritt 1: SSH in die Private Cloud.

Schritt 2: Führen Sie den folgenden Befehl aus:

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

Schritt 3: Aufnahme stoppen mit `Ctrl+C` (Windows) `Command-C` (Mac)

Schritt 4: Extrahieren Sie die `pcap` aus der Private Cloud.

Zugehörige Informationen

- [Konfigurieren der AMP für Endgeräte-Event-Stream-Funktion](#)
- [Technischer Support und Dokumentation für Cisco Systeme](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.