Probleemoplossing voor gebeurtenisstream in Private Cloud

Inhoud

Inleiding

Dit document beschrijft hoe u problemen kunt oplossen met Event Streams in Advanced Malware Protection Secure Endpoint Private Cloud.

Voorwaarden

Vereisten

Cisco raadt u aan kennis te hebben van de onderwerpen:

- Secure Endpoint voor privé-cloud
- API-query

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

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

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Configuratie

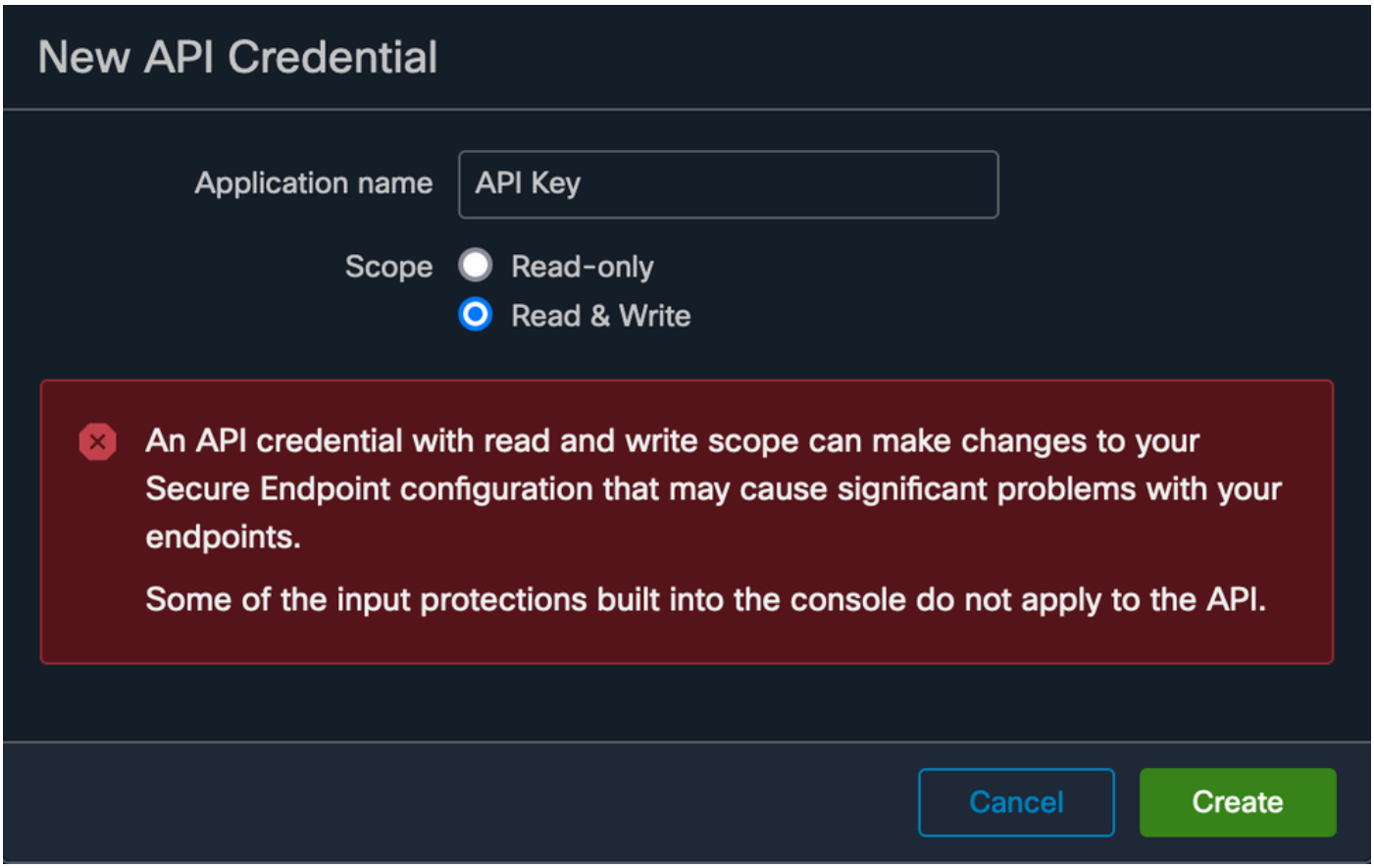
API-sleutel maken

Stap 1. Meld u aan bij Private Cloud Console.

Stap 2. Naar navigeren Accounts > API Credentials.

Stap 3. Klik op de knop New API Credential.

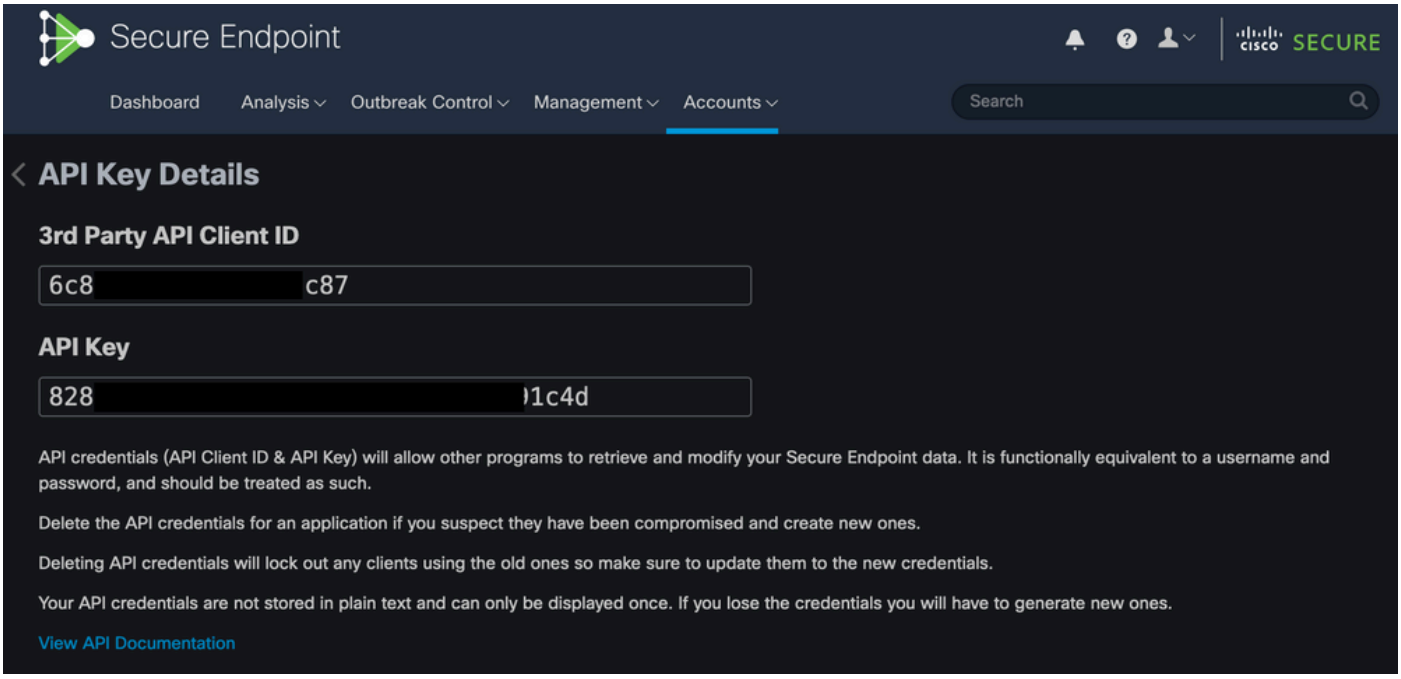
Stap 4. Voeg het Application name en klik op Read & Write toepassingsgebied.



API-sleutel maken

Stap 5. Klik op de knop Create.

Stap 6. API-referenties opslaan.



API-sleutel

Waarschuwing: de API-sleutel kan niet worden hersteld als u deze pagina verlaat.

Gebeurtenisstream maken

Hierdoor wordt een nieuwe Advanced Message Queuing Protocol (AMQP)-berichtstroom voor gebeurtenisinformatie gemaakt.

U kunt een gebeurtenisstream maken voor opgegeven typen en groepen gebeurtenissen:

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

U kunt een gebeurtenisstream maken voor alle soorten gebeurtenissen en alle groepen door:

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

MacOS/Linux

U kunt een Event Stream maken op MacOS/Linux met behulp van:

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

Windows

U kunt op Windows een gebeurtenisstream maken met behulp van:

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

Reactie

```
HTTP/1.1 201 Created
(...)
"data": {
  "id": 17,
  "name": "EVENT_STREAM_NAME",
  "amqp_credentials": {
    "user_name": "17-1bfXXXXXXXXX",
```

```
    "queue_name": "event_stream_17",
    "password": "3961XXXXXXXXXXXXXXXXXXXX814a77",
    "host": "FMC_SERVICE_URL",
    "port": 443,
    "proto": "https"
  }
}
```

Lijst van gebeurtenisstromen

Dit toont een lijst van gebeurtenisstromen die op Private Cloud zijn gemaakt.

MacOS/Linux

U kunt de Event Streams op MacOS/Linux weergeven met behulp van:

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

Windows

U kunt de Event Streams op Windows weergeven met behulp van:

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

Reactie

```
HTTP/1.1 200 OK
(...)
"data": {
  "id": 17,
  "name": "EVENT_STREAM_NAME",
  "amqp_credentials": {
    "user_name": "17-1bfXXXXXXXXX",
    "queue_name": "event_stream_17",
    "host": "FMC_SERVICE_URL",
    "port": 443,
    "proto": "https"
  }
}
```

Event Streams verwijderen

Verwijdert een actieve gebeurtenisstroom.

MacOS/Linux

U kunt Event Streams op MacOS/Linux verwijderen met behulp van:

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

Windows

U kunt Event Streams op Windows verwijderen met behulp van:

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

Reactie

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

Verifiëren

Stap 1. Kopieer het Python-script naar uw apparaat en bewaar het als 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()
```

Stap 2. Voer het uit in de terminal als `python3 EventStream.py`.

Stap 3. Trigger elke gebeurtenis die wordt toegevoegd aan de wachtrij voor gebeurtenisstromen.

Stap 4. Controleer of de gebeurtenissen in de terminal verschijnen.

Probleemoplossing

Om deze opdrachten uit te voeren moet u via SSH inloggen in de Private Cloud.

Controleer de Advanced Malware Service

Controleer of de service is ingeschakeld:

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

Controleer of de service actief is:

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

Controleer de verbinding met de ontvanger van de gebeurtenisstroom

Voer de opdracht uit:

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

Verbinding tot stand gebracht:

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

De verbinding is gesloten:

```
=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
```

Controleer op gebeurtenissen in de wachtrij

De gebeurtenissen in de rij zijn klaar om op deze gebeurtenisstroom naar de ontvanger worden verzonden nadat de verbinding wordt gevestigd. In dit voorbeeld zijn er 14 gebeurtenissen voor Event Stream ID 23.

<#root>

```
[root@fireamp rabbitmq]# rabbitmqctl list_queues
Listing queues ...
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_60b15rn8mpftaico6or6l8zxavl1usm 26
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_61984nlu8p11eeopmgmtcjra1v8gf5p 26
1acb0eb6-39f7-4b11-bd9b-fc4dd0e3bd77_iesRAgVoOh287mO_DetOx9PdDu8MxkS6kL4oSTeBm9s 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
```

Verzamel netwerkverkeersbestand

Om het verkeer van de gebeurtenisstroom van de Privécloud te verifiëren, kunt u met een tcpdump gereedschap:

Stap 1. SSH in de Private Cloud.

Stap 2. Voer de opdracht uit:

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

Stap 3. Stop de opname met `Ctrl+C` (Windows) of `Command-C` (Mac).

Stap 4. Haal de `pcap` bestand uit de Private Cloud.

Gerelateerde informatie

- AMP configureren voor functie Endpoints Event Stream
- Technische ondersteuning en documentatie – Cisco Systems

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.