Risoluzione dei problemi relativi al flusso di eventi nel cloud privato

Sommario

Introduzione

In questo documento viene descritto come risolvere i problemi relativi ai flussi di eventi in Advanced Malware Protection Secure Endpoint Private Cloud.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Secure Endpoint Private Cloud
- query API

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

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

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Configurazione

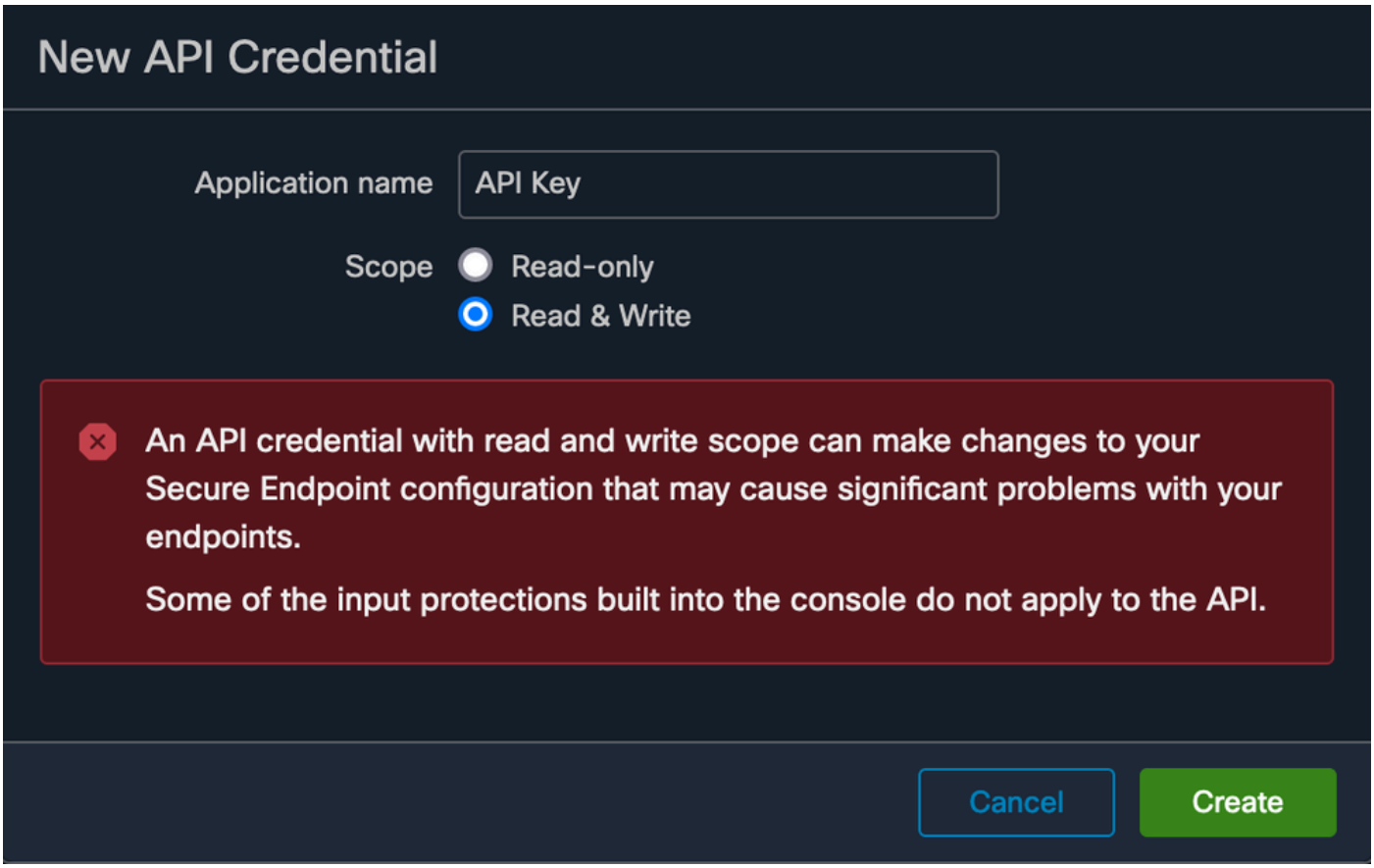
Crea chiave API

Passaggio 1. Accedere alla console cloud privata.

Passaggio 2. Passa a Accounts > API Credentials.

Passaggio 3. Fare clic su New API Credential.

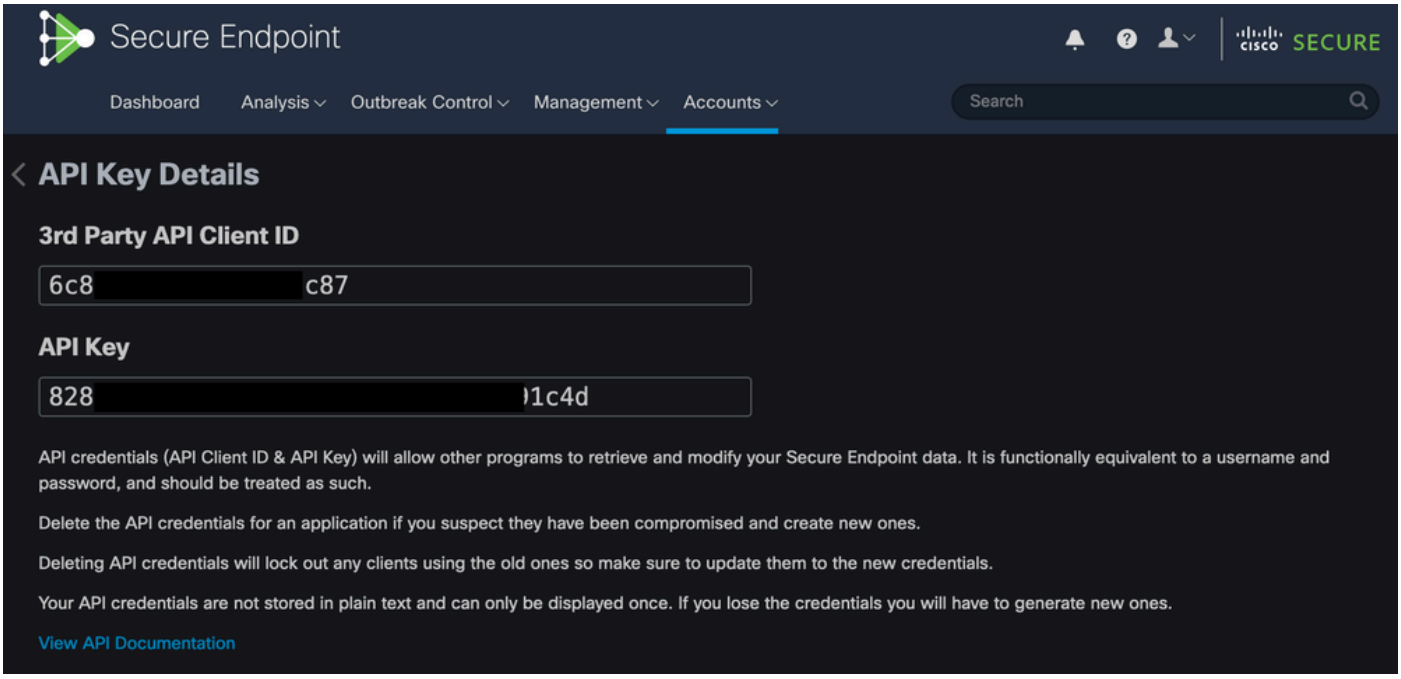
Passaggio 4. Aggiungere la Application name e fare clic su Read & Write ambito.



Crea chiave API

Passaggio 5. Fare clic su Create.

Passaggio 6. Salvare le credenziali API.



Chiave API

Attenzione: se si esce da questa pagina, non è possibile recuperare la chiave API.

Crea flusso eventi

In questo modo viene creato un nuovo flusso di messaggi AMQP (Advanced Message Queuing Protocol) per le informazioni sugli eventi.

È possibile creare un flusso di eventi per i tipi e i gruppi di eventi specificati:

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

È possibile creare un flusso di eventi per tutti i tipi di eventi e per tutti i gruppi nei modi seguenti:

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

MacOS/Linux

È possibile creare un flusso di eventi su MacOS/Linux utilizzando:

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

Windows

È possibile creare un flusso di eventi in Windows utilizzando:

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

Risposta

```
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"
  }
}
```

Elenco di flussi di eventi

In questo modo viene visualizzato un elenco dei flussi di eventi creati nel cloud privato.

MacOS/Linux

È possibile elencare i flussi di eventi su MacOS/Linux con l'utilizzo di:

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

Windows

È possibile elencare i flussi di eventi in Windows utilizzando:

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

Risposta

```
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"
  }
}
```

Elimina flussi di eventi

Elimina un flusso di eventi attivo.

MacOS/Linux

È possibile eliminare i flussi di eventi in MacOS/Linux utilizzando:

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

Windows

È possibile eliminare i flussi di eventi in Windows utilizzando:

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

Risposta

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

Verifica

Passaggio 1. Copiare lo script Python nel dispositivo e salvarlo con nome 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()
```

Passaggio 2. Eseguirlo nel terminale come `python3 EventStream.py`.

Passaggio 3. Attiva qualsiasi evento aggiunto alla coda del flusso di eventi.

Passaggio 4. Verificare che gli eventi vengano visualizzati nel terminale.

Risoluzione dei problemi

Per eseguire questi comandi, è necessario accedere tramite SSH al cloud privato.

Controllare il servizio AMQP

Verificare se il servizio è abilitato:

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

Verificare se il servizio è in esecuzione:

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

Controllare la connessione al ricevitore del flusso di eventi

Eseguire il comando:

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

Connessione stabilita:

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

Connessione chiusa:

```
=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
```

Controllare gli eventi nella coda

Gli eventi nella coda sono pronti per essere inviati su questo flusso di eventi al destinatario dopo aver stabilito la connessione. In questo esempio sono presenti 14 eventi per l'ID flusso evento 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
```

Raccogli file di traffico di rete

Per verificare il traffico del flusso di eventi dal cloud privato, è possibile raccogliere l'acquisizione con un `tcpdump` strumento:

Passaggio 1. SSH nel cloud privato.

Passaggio 2. Eseguire il comando:

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

Passaggio 3. Arresta la cattura con Ctrl+C (Windows) o Command-C (Mac)

Passaggio 4. Estrarre il pcap dal cloud privato.

Informazioni correlate

- Configura funzionalità flusso eventi AMP for Endpoints
- Documentazione e supporto tecnico – Cisco Systems

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).