프라이빗 클라우드의 이벤트 스트림 문제 해결

목차

소개

이 문서에서는 Advanced Malware Protection Secure Endpoint Private Cloud에서 이벤트 스트림을 트러블슈팅하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 항목에 대해 알고 있는 것이 좋습니다.

- 보안 엔드포인트 프라이빗 클라우드

- API 쿼리

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

설정

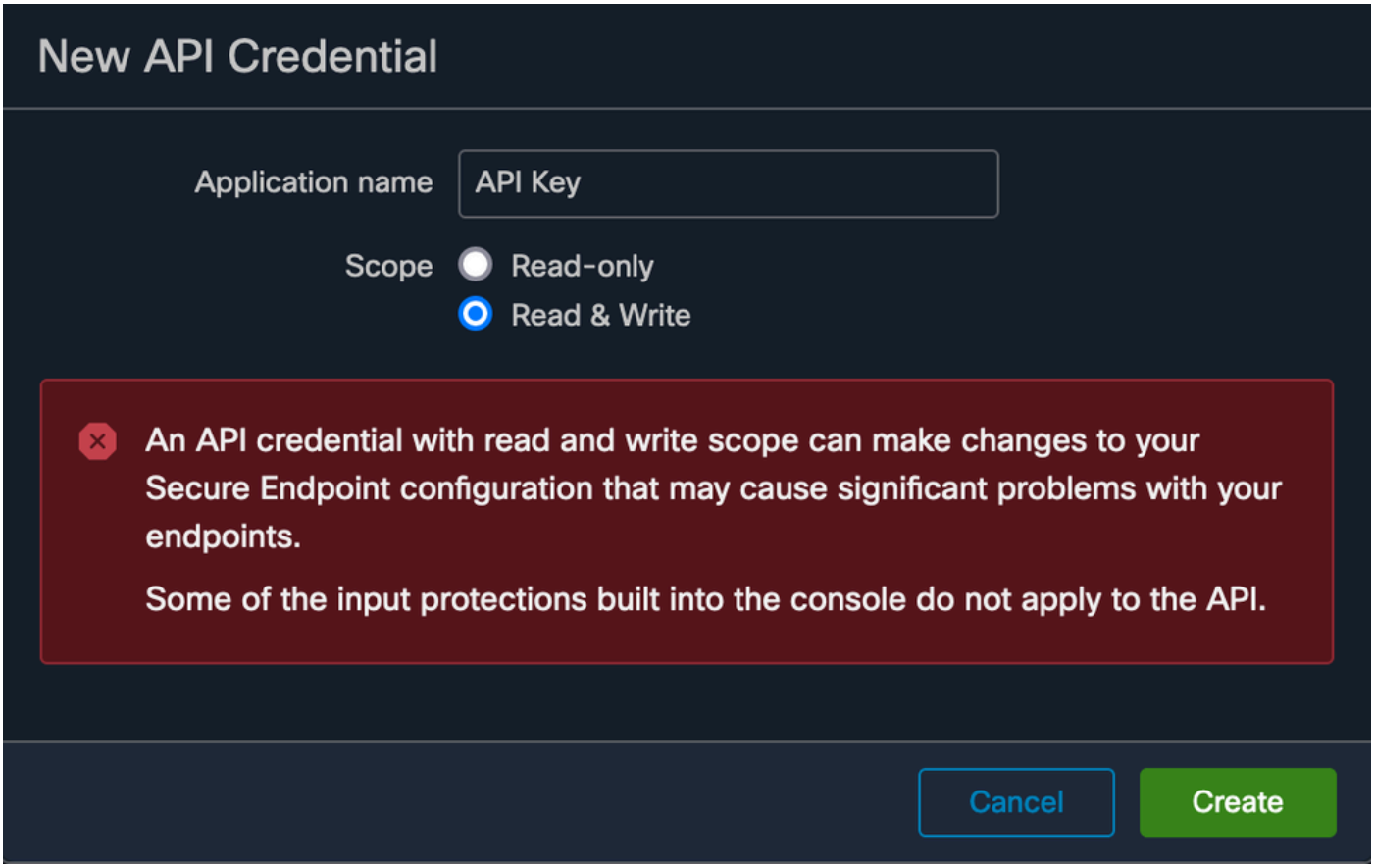
API 키 생성

1단계. Private Cloud Console에 로그인합니다.

2단계. 탐색 Accounts > API Credentials.

3단계. 클릭 New API Credential.

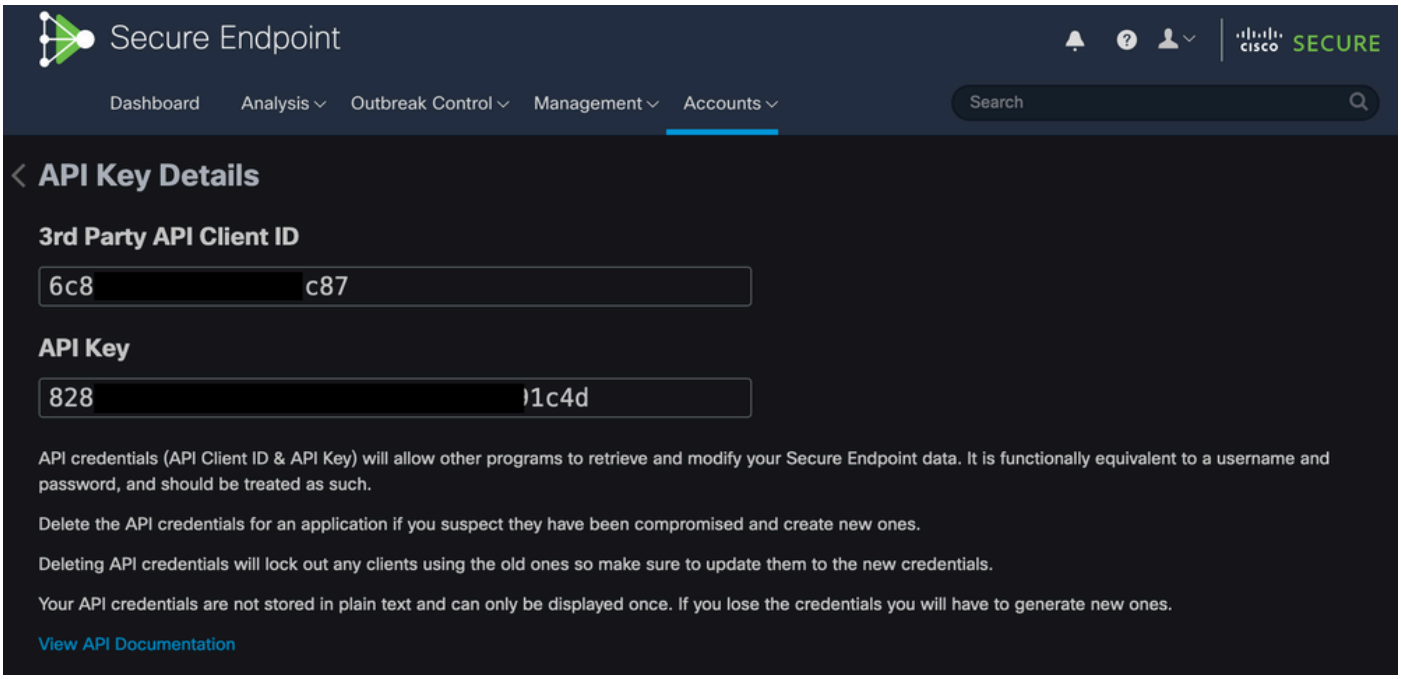
4단계. 추가 Application name 을 클릭하고 Read & Write 범위.



API 키 생성

5단계. 클릭 Create.

6단계. API 자격 증명을 저장합니다.



API 키

> 주의: 이 페이지에서 나가면 API 키를 복구할 수 없습니다.

이벤트 스트림 생성

이렇게 하면 이벤트 정보에 대한 새 AMQP(Advanced Message Queuing Protocol) 메시지 스트림이 생성됩니다.

지정된 이벤트 유형 및 그룹에 대한 이벤트 스트림을 생성할 수 있습니다.

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

다음을 통해 모든 이벤트 유형 및 모든 그룹에 대한 이벤트 스트림을 생성할 수 있습니다.

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

MacOS/Linux

다음을 사용하여 MacOS/Linux에서 이벤트 스트림을 생성할 수 있습니다.

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

창

다음을 사용하여 Windows에서 이벤트 스트림을 만들 수 있습니다.

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

응답

```
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"
  }
}
```

이벤트 스트림 목록

프라이빗 클라우드에서 생성된 이벤트 스트림 목록이 표시됩니다.

MacOS/Linux

다음을 사용하여 MacOS/Linux에서 이벤트 스트림을 나열할 수 있습니다.

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

창

다음을 사용하여 Windows에서 이벤트 스트림을 나열할 수 있습니다.

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

응답

```
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"
  }
}
```

이벤트 스트림 삭제

활성 이벤트 스트림을 삭제합니다.

MacOS/Linux

다음을 사용하여 MacOS/Linux에서 이벤트 스트림을 삭제할 수 있습니다.

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

창

다음을 사용하여 Windows에서 이벤트 스트림을 삭제할 수 있습니다.

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

응답

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

다음을 확인합니다.

1단계. Python 스크립트를 디바이스에 복사하고 다른 이름으로 저장합니다. 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()
```

2단계. 터미널에서 실행 python3 EventStream.py.

3단계. 이벤트 스트림 큐에 추가된 모든 이벤트를 트리거합니다.

4단계. 터미널에 이벤트가 나타나는지 확인합니다.

문제 해결

이러한 명령을 실행하려면 SSH를 통해 프라이빗 클라우드에 로그인해야 합니다.

AMQP 서비스 확인

서비스가 활성화되었는지 확인합니다.

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

서비스가 실행 중인지 확인합니다.

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

이벤트 스트림 수신기에 대한 연결 확인

다음 명령을 실행합니다.

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

연결 설정:

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

연결이 닫힘:

```
=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
```

대기열의 이벤트 확인

대기열의 이벤트는 연결이 설정된 후 이 이벤트 스트림에서 수신자에게 전송될 준비가 되었습니다. 이 예에서는 이벤트 스트림 ID 23에 대한 14개의 이벤트가 있습니다.

<#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
```

네트워크 트래픽 파일 수집

프라이빗 클라우드의 이벤트 스트림 트래픽을 확인하기 위해 tcpdump 툴:

1단계. 프라이빗 클라우드에 SSH를 적용합니다.

2단계. 다음 명령을 실행합니다.

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

3단계. 다음 방법으로 캡처 중지 Ctrl+C (Windows) Command-C (Mac).

4단계. 추출 pcap 파일을 다운로드합니다.

관련 정보

- AMP for Endpoints 이벤트 스트림 기능 구성
- 기술 지원 및 문서 – Cisco Systems

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.