配置面向終端的AMP事件流功能

目錄

簡介

本檔案介紹如何設定和使用適用於終端裝置的進階惡意軟體防護(AMP)的事件流功能。

必要條件

需求

思科建議您瞭解以下主題：

- AMP端點版
- Python程式設計基礎知識

採用元件

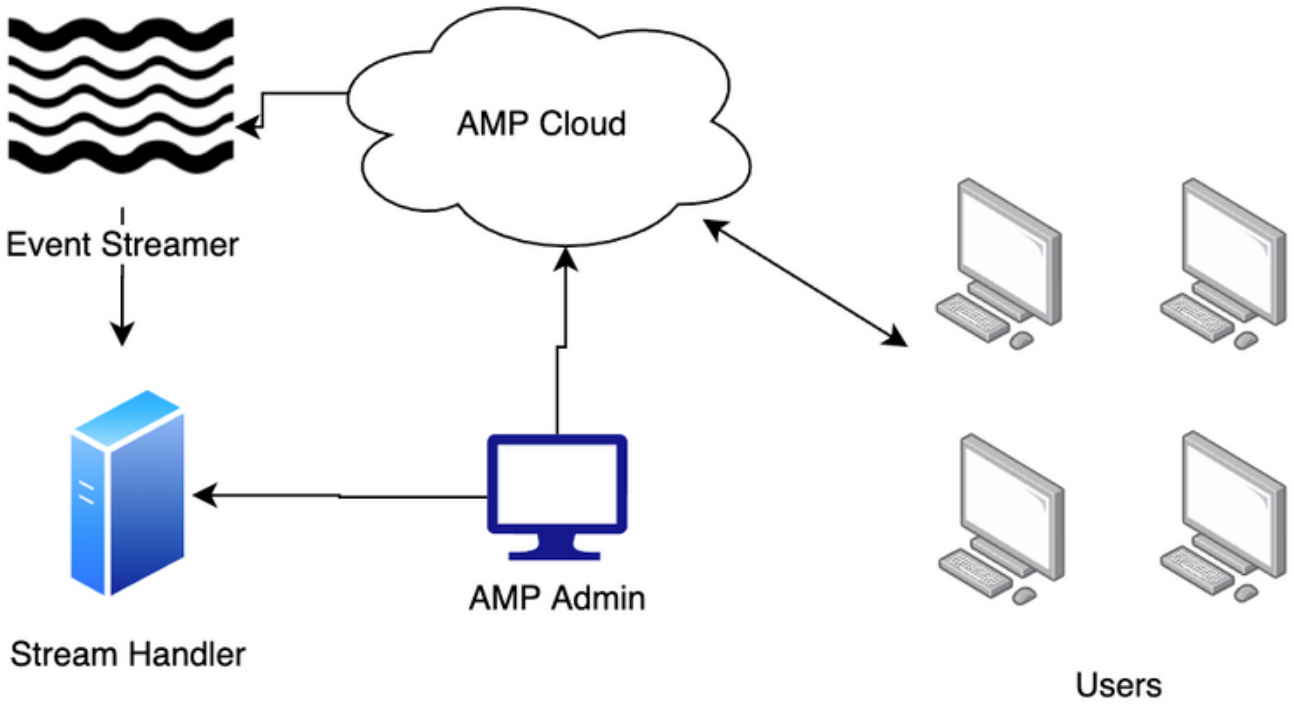
本檔案中的資訊是根據搭載**pika**（版本1.1.0）和**要求**（版本2.22.0）外部庫的Python 3.7。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

網路圖表

此影象提供了事件流排序的示例：



組態

建立API憑據

1. 導航到面向終端的AMP門戶並登入
2. 在**Accounts**下，選擇**API Credentials**
3. 按一下**New API Credentials**
4. 在「應用程式名稱」**欄位中輸入一個值**
5. 選擇**Read & Write** for **Scope**
6. 按一下「**Create**」
7. 將這些憑證儲存在密碼管理器或加密檔案中

建立事件流

1. 開啟Python shell並匯入**json**、**ssl**、**pika**和**requests**庫。

```
import json
import pika
import requests
import ssl
```

2. 儲存url、client_id和api_key的值。 如果您不使用北美雲，您的URL可能會有所不同。 此外，您的client_id和api_key對於您的環境是唯一的。

```
url = "https://api.amp.cisco.com/v1/event_streams"
client_id = "d16aff14860af496e848"
api_key = "d01ed435-b00d-4a4d-a299-1806ac117e72"
```

3. 建立要傳遞給請求的資料對象。 必須包括name ，並且可以包括event_type和group_guid

，以限制流中包含的事件和組。 如果未傳遞group_guid或event_type，則事件流將包括所有組和事件型別。

```
data = {
    "name": "Event Stream for ACME Inc",
    "group_guid": ["5cdf70dd-1b14-46a0-be90-e08da14172d8"],
    "event_type": [1090519054]
}
```

4. 進行POST請求呼叫，並將值儲存在變數中。

```
r = requests.post(
    url = url,
    data = data,
    auth = (client_id, api_key)
)
```

5. 列印狀態代碼。 確認代碼是201。

```
print(r.status_code)
```

6. 將響應的內容載入到json對象，並將該對象儲存在變數中。

```
j = json.loads(r.content)
```

7. 檢視響應資料的內容。

```
for k, v in j.items():
    print(f"{k}: {v}")
```

8. 高級消息隊列協定(AMQP)資料位於響應中。 將資料提取到相應的變數中。

```
user_name = j["data"]["amqp_credentials"]["user_name"]
queue_name = j["data"]["amqp_credentials"]["queue_name"]
password = j["data"]["amqp_credentials"]["password"]
host = j["data"]["amqp_credentials"]["host"]
port = j["data"]["amqp_credentials"]["port"]
proto = j["data"]["amqp_credentials"]["proto"]
```

9. 使用這些引數定義回撥函式。 在此設定中，將事件的正文列印到螢幕。 但是，您可以更改此功能的此內容以適應您的目標。

```
def callback(channel, method, properties, body):
    print(body)
```

10. 根據您建立的變數準備AMQP URL。

```
amqp_url = f"amqps://{user_name}:{password}@{host}:{port}"
```

11. 準備SSL上下文

```
context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
amqp_ssl = pika.SSLOptions(context)
```

12. 使用pika庫方法準備AMQP流。

```
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
)
```

13. 啟動流。

```
channel.start_consuming()
```

14. 溪流現已開通並等待相關事件。

驗證

觸發環境中終端上的事件。 例如，啟動快閃記憶體掃描。 注意資料流將事件資料列印到螢幕。

按**Ctrl+C**(Windows)或**Command-C**(Mac)中斷流。

疑難排解

狀態代碼

- 狀態代碼401表示存在授權問題。 檢查**client_id**和**api_key**，或生成新金鑰。
- 狀態代碼400表示存在錯誤請求問題。 請檢查您沒有使用此名稱建立事件流，或者您未建立超過5個事件流。 狀態代碼400的另一個可能的補救方法是新增以下變數：

```
headers = {
    'content-type': 'application/json'
}
```

並更新您的帖子請求以反映標題宣告：

```
r = requests.post(
    url = url,
    headers = headers,
    data = data,
    auth = (client_id, api_key)
)
```