配置终端事件数据流功能的AMP

目录

[简介](#)
[先决条件](#)
[要求](#)
[使用的组件](#)
[配置](#)
[网络图](#)
[配置](#)
[创建API凭证](#)
[创建事件数据流](#)
[验证](#)
[故障排除](#)
[状态码](#)

简介

本文描述如何配置和消耗先进的恶意软件保护的(AMP)事件数据流功能终端的。

先决条件

要求

Cisco建议您有以下主题的知识：

- 面向终端的 AMP
- 基础知识Python编程

使用的组件

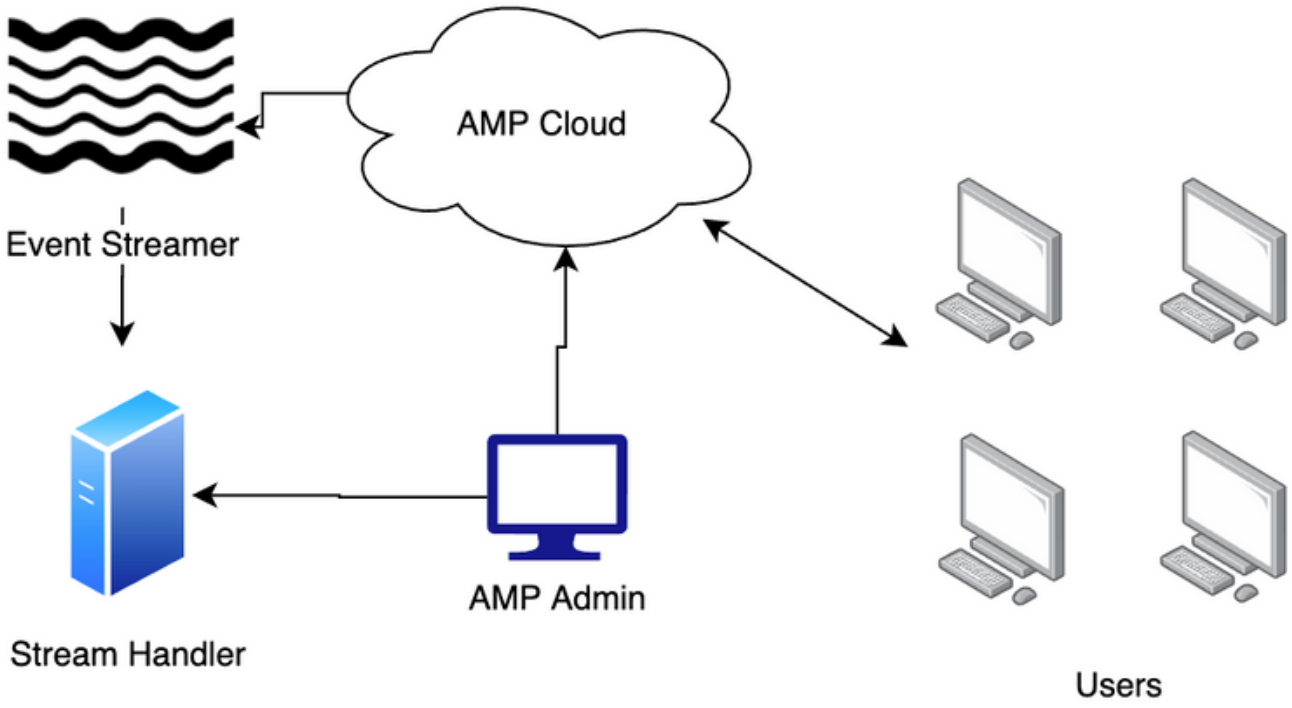
本文档中的信息根据与**pika** (版本1.1.0)和**请求**(版本2.22.0)外部库的Python 3.7。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

配置

网络图

此镜像提供定序的事件数据流示例：



配置

创建API凭证

1. 导航对您的门户的终端的AMP并且登陆
2. 在**帐户下，请**选择**API凭证**
3. 点击**新的API凭证**
4. 在**应用程序名称**名称字段输入一个值
5. 选择**读&**为**范围写入**
6. 单击**创建**
7. 存储在密码管理器或已加密文件的这些凭证

创建事件数据流

1. 打开Python shell并且导入**json**， **ssl**， **pika**并且**请求**库。

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

2. 存储URL、client_id和api_key的值。 如果不使用北美洲网云，您的URL能变化。 并且，您的client_id和api_key对您的环境是唯一。

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

3. 创建数据对象通过到请求。 这必须包括名称，并且能包括event_type和group_guid限制在数据

流和组包括的事件。 如果group_guid或event_type没有通过，事件数据流将包括所有组和事件类型。

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

4. 做POST请求呼叫，并且存储在变量的值。

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

5. 打印状态码。 确认代码是201。

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

6. 装载答复的内容到json对象，并且存储在变量的对象。

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

7. 查看答复数据的内容。

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

8. 先进的消息排队协议(AMQP)数据是在答复里面。 析取数据到各自变量。

```
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. 定义与这些参数的一个回叫功能。 在此设置，您打印事件的正文对屏幕。 然而，您能更改此功能此内容配合您的目标。

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

10. 准备从您创建的变量的AMQP URL。

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

11. 准备SSL上下文

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

12. 准备与pika库方法的AMQP数据流。

```
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. 启动数据流。

```
channel.start_consuming()
```

14. 数据流当前是实际和等候事件。

验证

触发在一个终端的一个事件在您的环境。 例如，请启动一闪存扫描。 注意数据流打印事件数据对屏幕。

按**Ctrl+C** (Windows)或**C命令**(Mac)中断数据流。

故障排除

状态码

- 状态码401指示有与授权的一个问题。 检查您的**client_id**和**api_key**或者生成新的密钥。
- 状态码400指示有一个Bad请求问题。 检查您没有一事件数据流创建与该名称，或者您没有超过创建的5事件数据流。