데이터 분석을 통해 원격 액세스 VPN 설정을 최적화하는 프로그래밍 방식

목차

소개

이 문서에서는 현재 사용 가능한 프로그래밍 모듈과 오픈 소스 툴을 통해 설정된 원격 액세스 VPN을 모니터링하고 최적화하는 방법에 대해 설명합니다.유용한 정보를 얻기 위해 활용할 수 있는 가장 작은 네트워크에서도 오늘날 많은 데이터가 생성됩니다.수집된 데이터에 대한 분석을 적용하면 팩트를 바탕으로 보다 빠르고 정확한 정보를 바탕으로 비즈니스 의사 결정을 내릴 수 있습니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 원격 액세스 VPN
- 기본 Python 프로그래밍 개념

사용되는 구성 요소

이 문서는 특정 Cisco ASA 또는 FTD 소프트웨어 및 하드웨어 버전으로 제한되지 않습니다.

> **참고:**Pandas, Streamlit, CSV 및 Matplotlib는 사용되는 몇 가지 Python 라이브러리입니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.네트워크가 작동 중인 경우 모든 명령 및 python 스크립트의 잠재적인 영향을 이해해야 합니다.

문제

전체 직원 중 다수에 대해 재택 근무 모델을 도입하는 기업이 많아 VPN에 의존하여 업무를 수행하는 사용자의 수가 크게 증가했습니다.이로 인해 VPN 집선 장치에 대한 로드가 급격히 증가하여 관리자는 VPN 설정을 다시 생각하고 다시 계획할 수 있습니다.ASA Concentrator의 로드를 줄이기 위해 정보에 근거한 결정을 내리려면 일정 기간 동안 디바이스에서 다양한 정보를 수집하고 해당 정보를 평가해야 합니다. 이는 복잡한 작업이며 수동으로 작업을 수행하는 경우 상당한 시간이 필요합니다.

솔루션

현재 네트워크 프로그래밍 및 데이터 분석에 사용할 수 있는 몇 가지 Python 모듈과 오픈 소스 툴을 통해 프로그래밍을 통해 VPN 설정의 데이터 수집 및 분석, 계획 및 최적화에 큰 도움이 될 수 있습니다.

VPN 사용자 및 동시 연결을 기반으로 하는 초기 분석

분석을 시작하려면 연결 사용자 수, 설정된 동시 연결 수 및 대역폭에 미치는 영향을 확인합니다.다음 Cisco ASA 명령 출력은 다음 세부 정보를 제공합니다.

- **vpn-sessiondb anyconnect 표시**
- **conn 표시**

Python 모듈 **Netmiko**를 사용하여 디바이스에 ssh를 수행하고 명령을 실행하고 출력을 구문 분석할 수 있습니다.

```
cisco_asa_device = {

       "host": host,

       "username": username,

       "password": password,

       "secret": secret,

       "device_type": "cisco_asa",

   }

   net_conn = ConnectHandler(**cisco_asa_device)

   command = "show vpn-sessiondb anyconnect"

   command_output = net_conn.send_command(command)
```

목록에서 VPN 사용자 수 및 연결 수를 일정한 간격으로(2시간마다 올바른 시작이 될 수 있음) 수집하고 하루 최대 일일 카운트를 가져옵니다.

```
#list1 is the list of user counts collected in a day
#list2 is the list of connection counts in a day
list1.sort()
max_vpn_user = list1[-1]

list2.sort()
max_conn = list2[-1]

df1.append([max_vpn_user,max_conn])
```

Pandas는 효율적인 데이터 분석 및 조작 라이브러리이며, 분석된 모든 데이터는 판다에게 시리즈 또는 데이터 프레임으로 저장될 수 있어 데이터에 대한 작업을 쉽게 수행할 수 있습니다.

```
import pandas as pd

df = pd.DataFrame(df1, columns=['Max Daily VPN Users Count','Max Daily Concurrent
Connections'],index=<date range>)
```

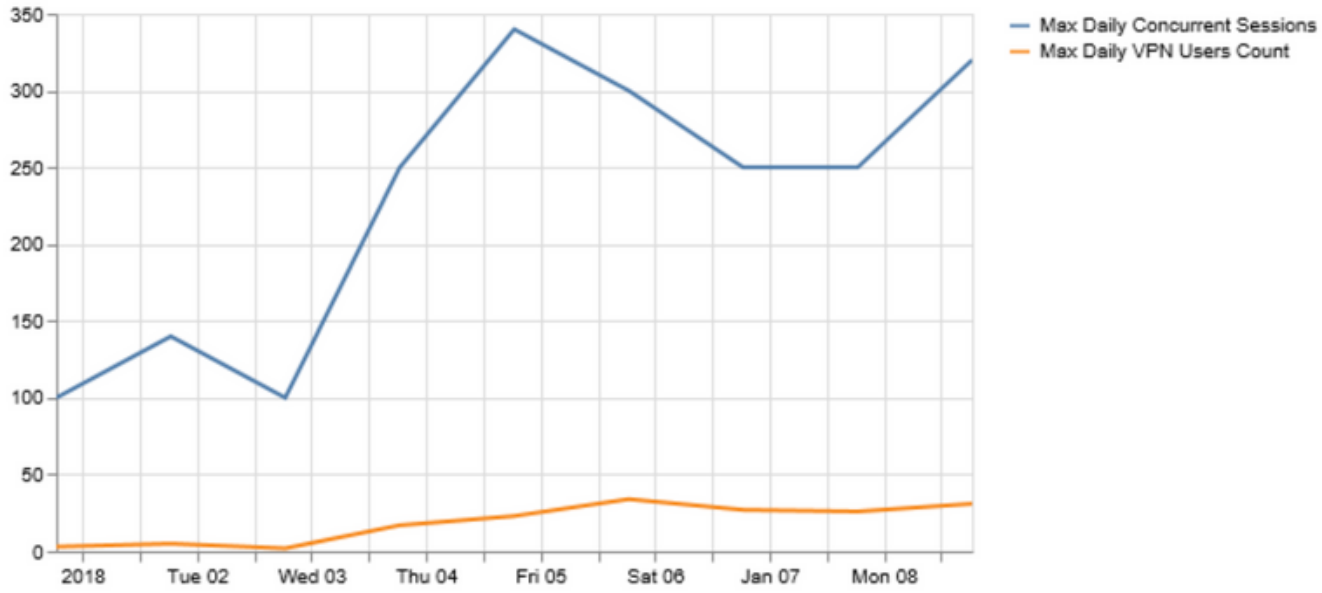
Daily Max VPN user Count - Max concurrent count

	Max Daily VPN Users Count	Max Daily Concurrent Sessions
Jan 1, 2018	3	100
Jan 2, 2018	5	140
Jan 3, 2018	2	100
Jan 4, 2018	17	250
Jan 5, 2018	23	340
Jan 6, 2018	34	300
Jan 7, 2018	27	250
Jan 8, 2018	26	250
Jan 9, 2018	31	320

VPN 설정을 최적화하는 데 도움이 되는 **일일 최대 VPN 사용자**와 **최대 동시 연결**을 분석합니다.

여기 이미지에 표시된 대로 팬더 및 마트플로립 라이브러리에서 플롯 기능을 사용합니다.

```
df.plot()

matplotlib.pyplot.show()
```



VPN 사용자 또는 동시 연결 수가 VPN 헤드엔드의 용량에 가까워지면 다음과 같은 문제가 발생할 수 있습니다.

- 삭제되는 새 VPN 사용자입니다.

- ASA를 통한 새 데이터 연결이 삭제되고 사용자가 리소스에 액세스할 수 없습니다.
- 높은 CPU 및/또는 메모리

일정 기간 동안의 트렌드로 인해 상자가 임계값에 도달하는지 확인할 수 있습니다.

내부 네트워크 또는 외부 네트워크로 향하는 트래픽 트렌드 식별

Cisco ASA의 **Show conn** 출력에서는 트래픽이 내부 또는 외부 네트워크에 연결되는지 여부, 플로우당 전달되는 데이터의 양(바이트)과 같은 추가 세부 정보를 제공할 수 있습니다.

Soure IP	Destination IP	Service	Bytes
10.10.1.1	10.30.2.2	tcp/445	1234
10.10.1.2	40.5.2.3	tcp/443	2341
10.10.1.4	42.4.2.33	tcp/80	5432
10.10.2.3	52.3.2.34	tcp/443	1223
10.10.6.5	10.30.22.2	tcp/80	212
10.10.3.2	10.30.2.3	udp/389	1212
10.10.3.4	32.3.22.2	tcp/443	2123

Netaddr **Python** 모듈을 사용하면 가져온 연결 테이블을 외부 네트워크와 내부 네트워크로 쉽게 분할할 수 있습니다.

```
for f in df['Responder IP']:
   private.append(IPAddress(f).is_private())

df['private'] = private

df_ext = df[df['private'] == False]

df_int = df[df[‘private’] == True]
```

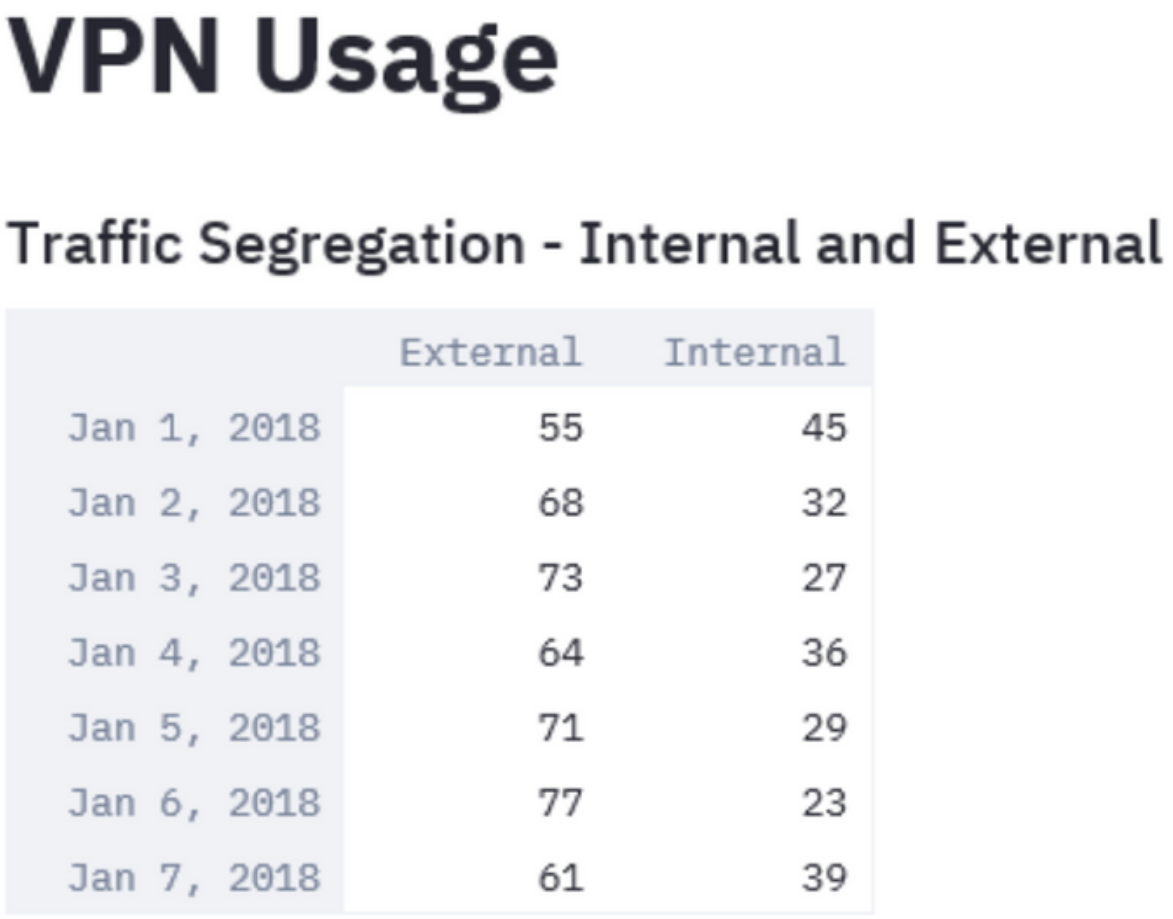
내부 트래픽의 이미지입니다.

Soure IP	Destination	Service	Bytes
10.10.1.1	10.30.2.2	tcp/445	1234
10.10.6.5	10.30.22.2	tcp/80	212
10.10.3.2	10.30.2.3	udp/389	1212

외부 트래픽의 이미지입니다.

Soure IP	Destination	Service	Bytes
10.10.1.2	40.5.2.3	tcp/443	2341
10.10.1.4	42.4.2.33	tcp/80	5432
10.10.2.3	52.3.2.34	tcp/443	1223
10.10.3.4	32.3.22.2	tcp/443	2123

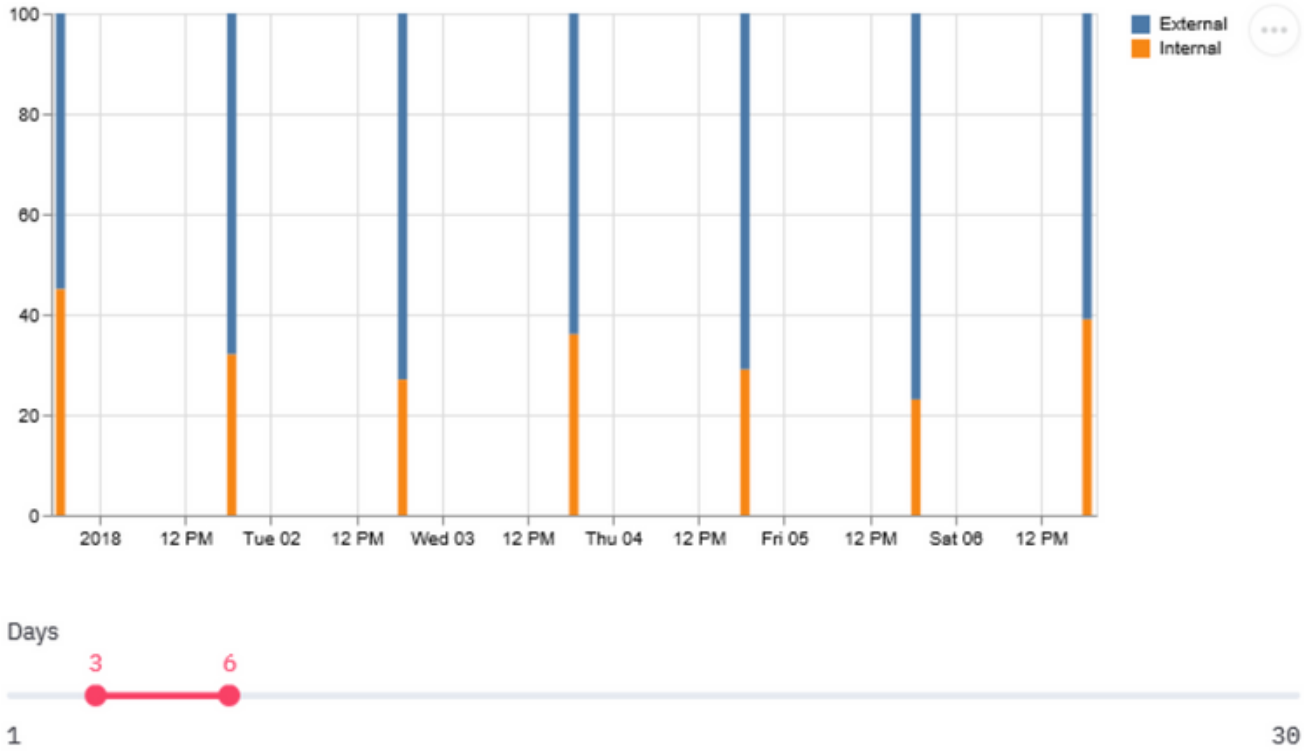
따라서 내부 네트워크로 향하는 VPN 트래픽의 비율 및 그 중 얼마나 많은 트래픽이 인터넷으로 전송되는지 파악할 수 있습니다.일정 기간 동안 이 정보를 수집하고 해당 트렌드를 분석하면 VPN 트래픽이 주로 외부 트래픽인지 내부 트래픽인지를 확인할 수 있습니다.



Streamlit와 같은 **모듈**을 사용하면 표 형식 데이터를 그래픽 표현으로 변환할 뿐만 아니라 실시간으로 수정사항을 적용하여 분석할 수 있습니다.수집된 데이터의 타임 윈도우를 수정하거나 모니터링 중인 매개변수에 데이터를 추가할 수 있습니다.

```
import streamlit

#traffic_ptg being a 2D array containing the data collected as in the table above

d = st.slider('Days',1,30,(1,7))

idx = pd.date_range('2018-01-01', periods=7, freq='D')

df = pd.DataFrame(d<subset of the list traffic_ptg based on slider
```

```
value>,columns=['External','Internal'],index=idx)

st.bar_chart(df)
```

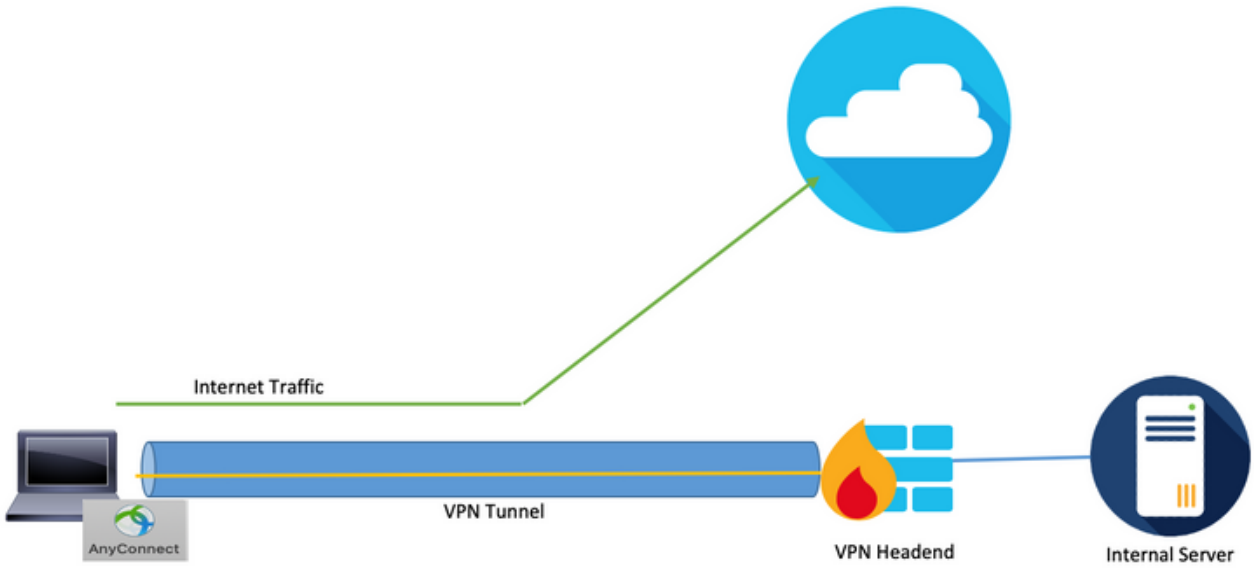


내부 트래픽이 더 높은 것으로 기울어지는 트렌드는 대부분의 VPN 사용자가 내부 리소스에 액세스한다는 것을 의미할 수 있습니다.따라서 로드 증가를 위해 더 큰 박스로의 업그레이드를 계획하거나 VPN 로드 밸런싱과 같은 개념으로 로드를 공유하는 것이 중요합니다.

경우에 따라 VPN 용량이 여전히 임계값 미만이지만 VPN 사용자 수가 증가하면 현재 구성된 VPN 풀이 소진될 수 있습니다.이 경우 VPN IP 풀을 늘립니다.

그러나 트렌드에서 VPN 트래픽의 대부분이 외부 트래픽인 경우 스플릿 터널링을 사용할 수 있습니다.

스플릿 터널링 기능 활용



사용자 시스템에서 터널을 통해 특정 트래픽 집합만 전달하는 기능이며 나머지 트래픽은 VPN 암호화 없이 기본 게이트웨이로 전달됩니다.따라서 VPN Concentrator의 로드를 줄이기 위해 내부 네트워크로 향하는 트래픽만 터널을 통해 라우팅될 수 있으며, 인터넷 트래픽은 사용자의 로컬 ISP를 통해 전달될 수 있습니다.이것은 효과적인 방법이며 널리 채택되었지만 약간의 위험이 있다.

직원들이 보호되지 않는 네트워크를 통해 일부 소셜 미디어 사이트에 신속하게 접속하면 업무 공간에 설정된 심층 방어 보안 레이어가 없어 회사 전체에 전파되는 악성코드로 랩탑을 감염시킬 수 있습니다.감염된 디바이스는 인터넷에서 경계 방어를 우회하면서 신뢰할 수 있는 세그먼트로 피벗점이 될 수 있습니다.

이 기능을 활용하면서 위험을 줄일 수 있는 한 가지 방법은 데이터 위생과 Duo Security와의 호환성 등 엄격한 보안 기준을 충족하는 클라우드 서비스에만 스플릿 터널링을 사용하는 것입니다.이를 채택하면 이전에 관찰된 많은 외부 트래픽이 이러한 보안 클라우드 서비스로 향할 경우 도움이 됩니다.따라서 VPN 사용자가 액세스하는 웹 애플리케이션을 분석할 필요가 생깁니다.

Cisco FTD(Firepower Threat Defense)와 같은 대부분의 차세대 방화벽은 이벤트와 관련된 애플리케이션 정보를 로그에 포함합니다.python csv **라이브러리** 및 Pandas 데이터 조작 기능으로 이 로그 데이터를 구문 분석 및 정리하면 이와 유사한 데이터 집합을 제공할 수 있으며 여기에 매핑되는 애플리케이션이 추가로 추가될 수 있습니다.

```
#connections.csv contains the connection events from ASA and events_with_app.csv contains
connection events with Application details fromFTD

df1 = pd.read_csv(‘connections.csv’) df2 = pd.read_csv(‘events_with_app.csv') df_merged =
pd.merge(df1,df2,on=['Source IP','Destination IP','Service'])
```

Soure IP	Destination IP	Service	Bytes	Application
10.10.1.1	10.30.2.2	tcp/445	1234	
10.10.1.2	40.5.2.3	tcp/443	2341	Microsoft
10.10.1.4	42.4.2.33	tcp/80	5432	Microsoft
10.10.2.3	52.3.2.34	tcp/443	1223	Office365
10.10.6.5	10.30.22.2	tcp/80	212	
10.10.3.2	10.30.2.3	udp/389	1212	
10.10.3.4	32.3.22.2	tcp/443	2123	Youtube

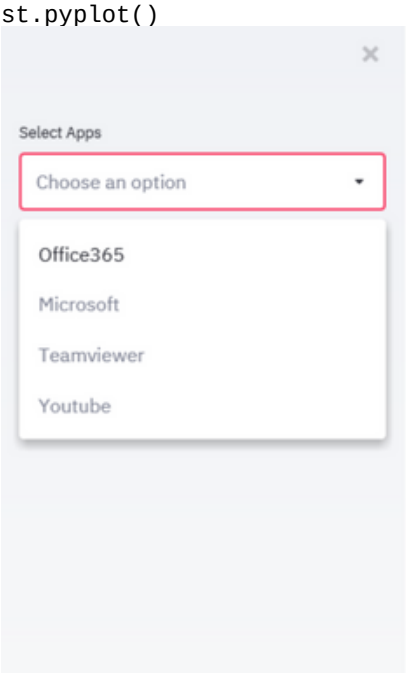
위의 데이터 프레임을 얻은 후에는 애플리케이션을 기반으로 Pandas를 통해 총 외부 트래픽을 분류할 수 있습니다.

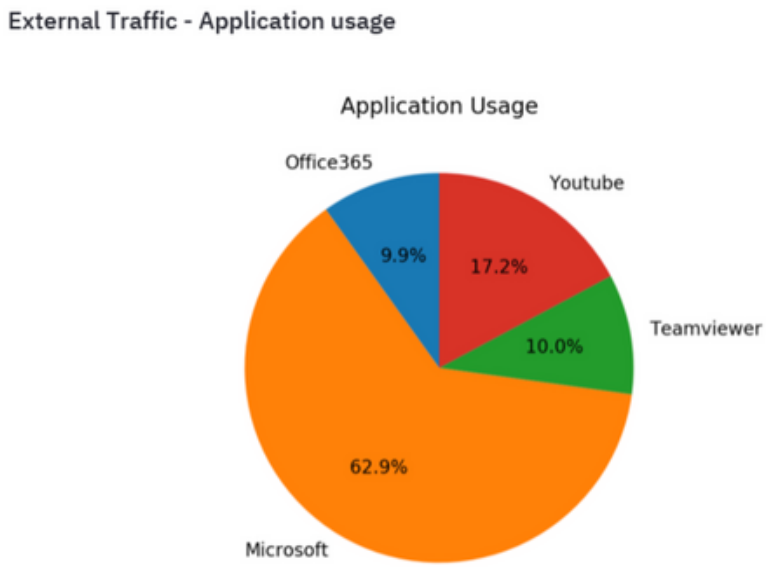
```
df2 = df.groupby('Application')

df3 = df2['Bytes'].sum()
```

```
Application
Microsoft      7773
Office365      1223
Teamviewer     1234
Youtube        2123
Name: Bytes, dtype: int64
```

Streamlit를 다시 사용하면 전체 트래픽에서 각 애플리케이션의 공유를 그래픽으로 볼 수 있습니다 .이 기능을 사용하면 코드를 변경할 필요 없이 사용자 인터페이스 자체에서 애플리케이션을 필터링하고 데이터를 포함할 시간 창을 유연하게 변경할 수 있으므로 분석을 쉽고 정확하게 수행할 수 있습니다.

```
import matplotlib.pyplot as plt

apps = ['Office365', 'Microsoft', 'Teamviewer', 'Youtube']
app_select = st.sidebar.multiselect('Select Apps',activities)

# app_bytes - list containing the applications and bytes

plt.pie(app_bytes, labels=apps)
plt.title('Application Usage')

st.pyplot()
```





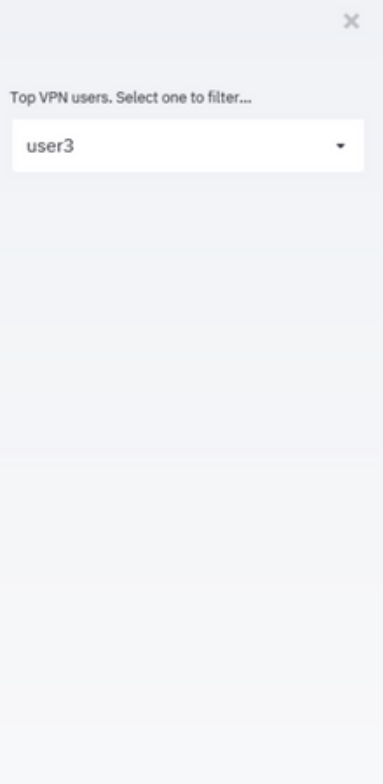
이를 통해 일정 기간 동안 VPN 사용자가 사용하는 상위 웹 애플리케이션을 식별하는 프로세스를 간소화하고 이러한 애플리케이션이 클라우드 서비스를 보호하는지 여부를 파악할 수 있습니다.

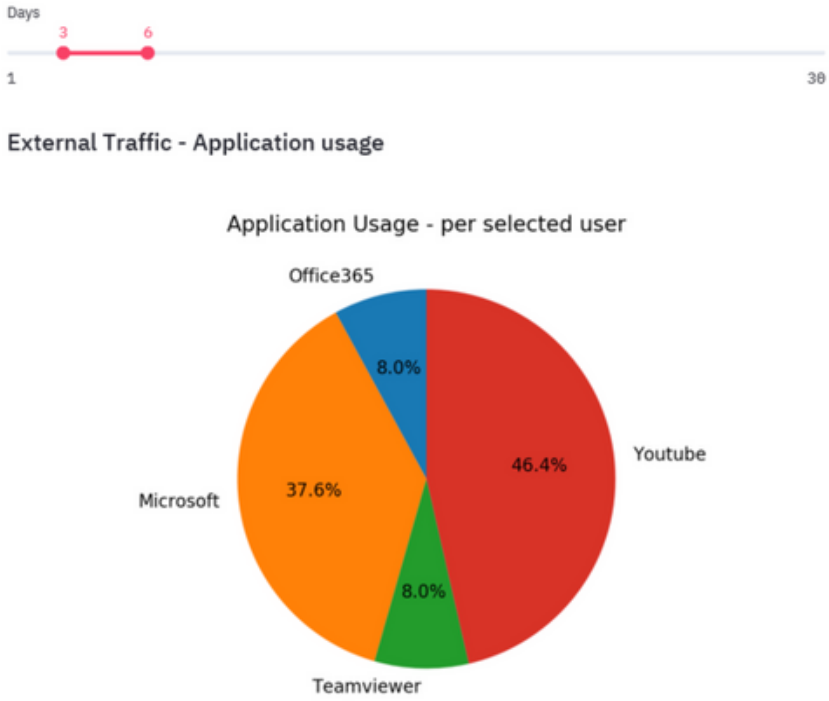
많은 양의 애플리케이션이 보안 클라우드 서비스를 식별하도록 지정된 경우 분할 터널과 함께 사용할 수 있으므로 VPN 집중 장치의 로드가 줄어듭니다.그러나 상위 애플리케이션이 덜 안전하거나

위험을 초래할 수 있는 서비스에 있다면 VPN 터널을 통해 이를 전달하는 것이 더 안전합니다.다른 네트워크 보안 디바이스가 이러한 트래픽을 통과하기 전에 트래픽을 처리할 수 있는 이유입니다.그런 다음 방화벽에서 액세스 정책을 활용하여 외부 네트워크에 대한 액세스를 제한할 수 있습니다.

개별 비준수 VPN 사용자 ID

경우에 따라 이러한 급증은 특정 정책을 준수하지 않는 소수의 사용자에게만 적용될 수 있습니다 .위에서 사용한 모듈과 데이터 세트를 다시 사용하여 상위 VPN 사용자와 사용자가 액세스하는 웹 애플리케이션을 식별할 수 있습니다.이를 통해 이러한 사용자를 격리하고 디바이스 로드에 미치는 영향을 관찰할 수 있습니다.





어떤 방법에도 맞지 않는 시나리오에서 관리자는 AMP for Endpoints 솔루션 및 Cisco Umbrella 솔루션과 같은 엔드포인트 보안 솔루션을 검토하여 보호되지 않는 네트워크의 엔드포인트를 보호해야 합니다.