データ分析を通じてリモートアクセスVPN設定を最適化するプログラマティックアプローチ

内容

概要

このドキュメントでは、今日の使用可能なプログラミングモジュールとオープンソースツールを使用して、リモートアクセスVPNのセットアップを監視および最適化する方法について説明します。有用な情報を得るために利用できる最小限のネットワークでも、今日では大量のデータが生成されています。この収集されたデータに分析を適用することで、情報に基づく迅速なビジネス上の意思決定が可能になります。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- リモート アクセス VPN
- 基本的なPythonプログラミングの概念

使用するコンポーネント

このドキュメントは、特定のCisco ASAまたはFTDソフトウェアおよびハードウェアバージョンに限定されるものではありません。

> **注**：Pandas、Streamlit、CSV、およびMatplotlibは、使用されるPythonライブラリです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。ネットワークが稼働中の場合は、コマンドやPythonスクリプトが及ぼす潜在的な影響について理解しておく必要があります。

問題

多くの企業が従業員の大半にWork From Homeモデルを採用しており、VPNを利用して仕事を行うユーザの数は大幅に増加しています。これにより、VPNコンセントレータの負荷が急激かつ大幅に増加し、管理者はVPNセットアップを見直して計画し直すことができます。ASAコンセントレータの負荷を軽減するための十分な情報を得るには、デバイスから一定時間にわたって幅広い情報を収集し、複雑なタスクであり、手動で行うと多大な時間が必要になる情報を評価する必要があります。

解決方法

ネットワークプログラマビリティとデータ分析に今日、いくつかのPythonモジュールとオープンソースツールが利用可能なので、プログラミングはVPNセットアップのデータ、計画、最適化の収集と分析に非常に役立ちます。

VPNユーザと同時接続に基づく初期分析

分析を開始するには、接続ユーザ数、同時接続の確立、帯域幅への影響を取得します。次のCisco ASAコマンド出力は、これらの詳細を示します。

- show vpn-sessiondb anyconnect
- show conn

Pythonモジュール**Netmiko**を使用して、デバイスへのssh接続、コマンドの実行、および出力の解析を行うことができます。

```
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)
```

VPNのユーザ数と接続数を定期的に（2時間ごとに）リストから収集し、1日の最大日数を取得します。

```
#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])
```

Pandasは効率的なデータ分析および操作ライブラリであり、解析されたデータはすべてデータの操作を容易にするpandasのシリーズまたはデータフレームとして保存できます。

```
import pandas as pd

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

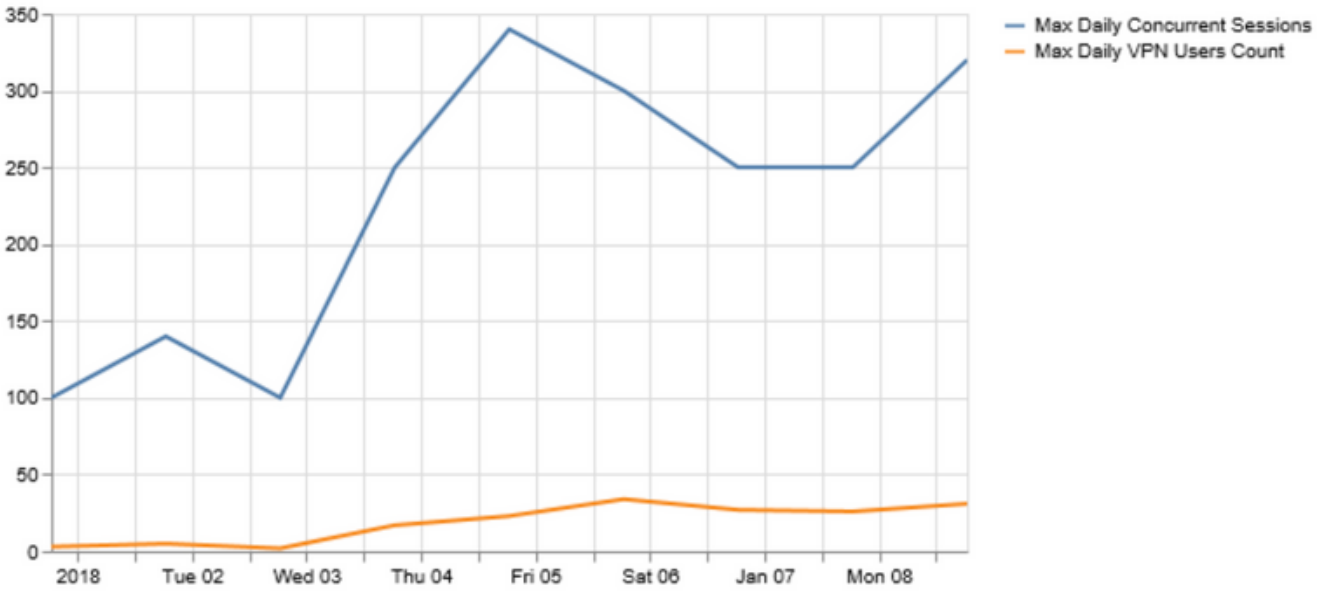
Daily Max VPN user Count - Max concurrent count

	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

毎日の最大**VPNユーザーと最大**同時接続を**分析**し、VPN設定の最適化の必要性を判断します。

図に示すように、pandasおよびmatplotlibライブラリでプロット機能を使用します。

```
df.plot()

matplotlib.pyplot.show()
```



VPNユーザまたは同時接続の数がVPNヘッドエンドのキャパシティに近づいている場合、次の問

題が発生する可能性があります。

- ドロップされる新しいVPNユーザ。
- ASA経由の新しいデータ接続がドロップされ、ユーザがリソースにアクセスできない。
- 高CPUおよび/またはメモリ。

ある期間にわたる傾向は、ボックスがしきい値に達しているかどうかを判断するのに役立ちます。

内部ネットワークまたは外部ネットワークに対するトラフィックトレンドの特定

Cisco ASAでのshow connの出力は、トラフィックが内部ネットワークか外部ネットワークか、およびフローあたりのデータ量（バイト）がファイアウォールを通過するといった詳細を提供できます。

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モジュールを使用すると、取得した接続テーブルを外部ネットワークおよび内部ネットワークへのフローに簡単に分割できます。

```
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]
```

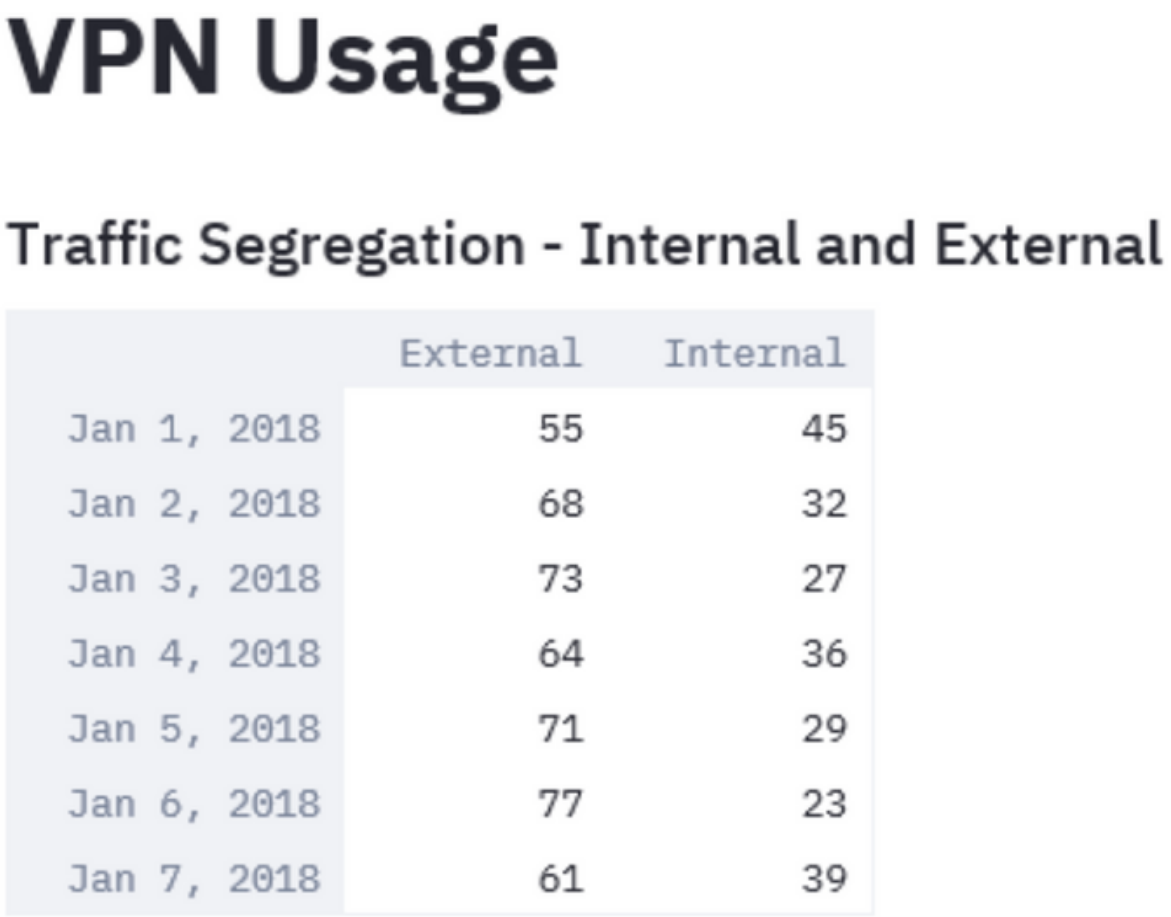
これは、内部トラフィックのイメージです。

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

これは、外部トラフィックのイメージです。

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

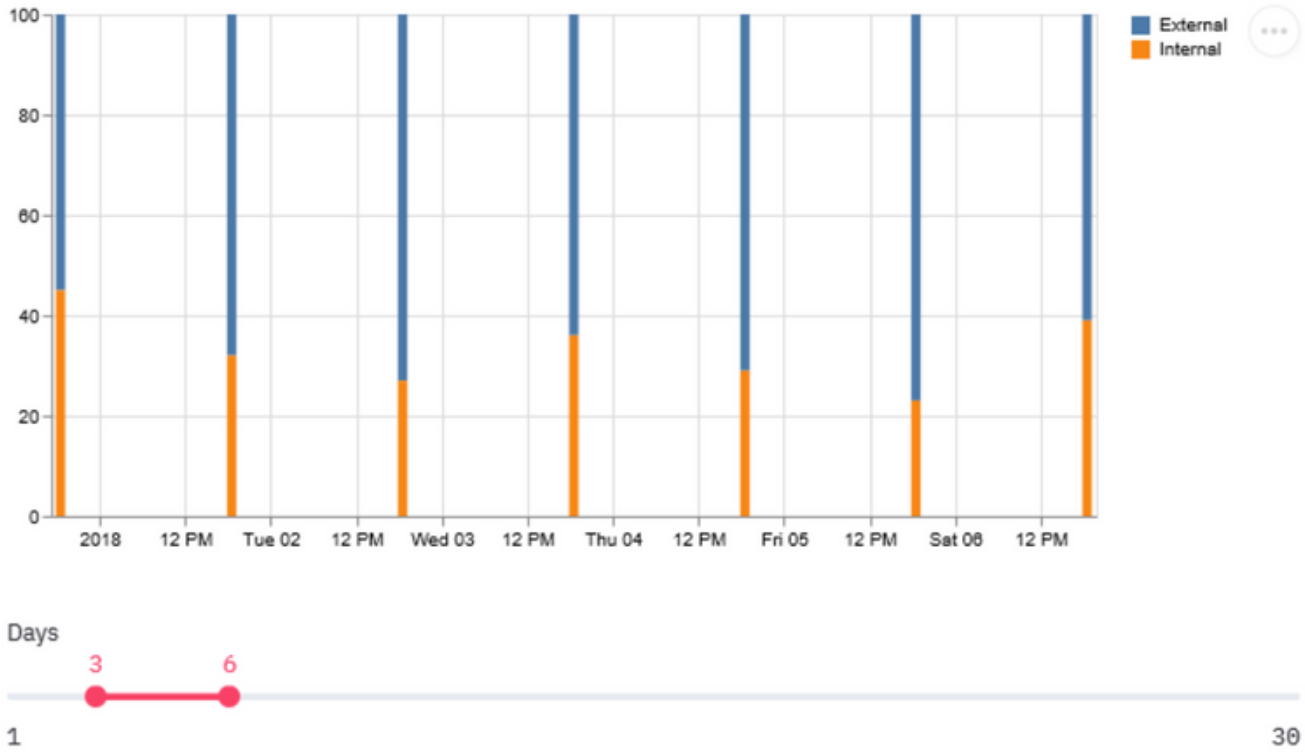
これにより、VPNトラフィックの内部ネットワーク宛の割合と、インターネットに送信されるトラフィックの量を把握できます。この情報を一定期間にわたって収集し、その傾向を分析することで、VPNトラフィックが主に外部または内部のどちらであるかを判断できます。



Streamlitなどのモジュールを使用すると、表形式データをグラフィカル表示に変換するだけでなく、リアルタイムで修正を適用して分析を支援できます。収集されたデータの時間枠を変更したり、監視対象のパラメータにデータを追加したりできます。

```
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)
```

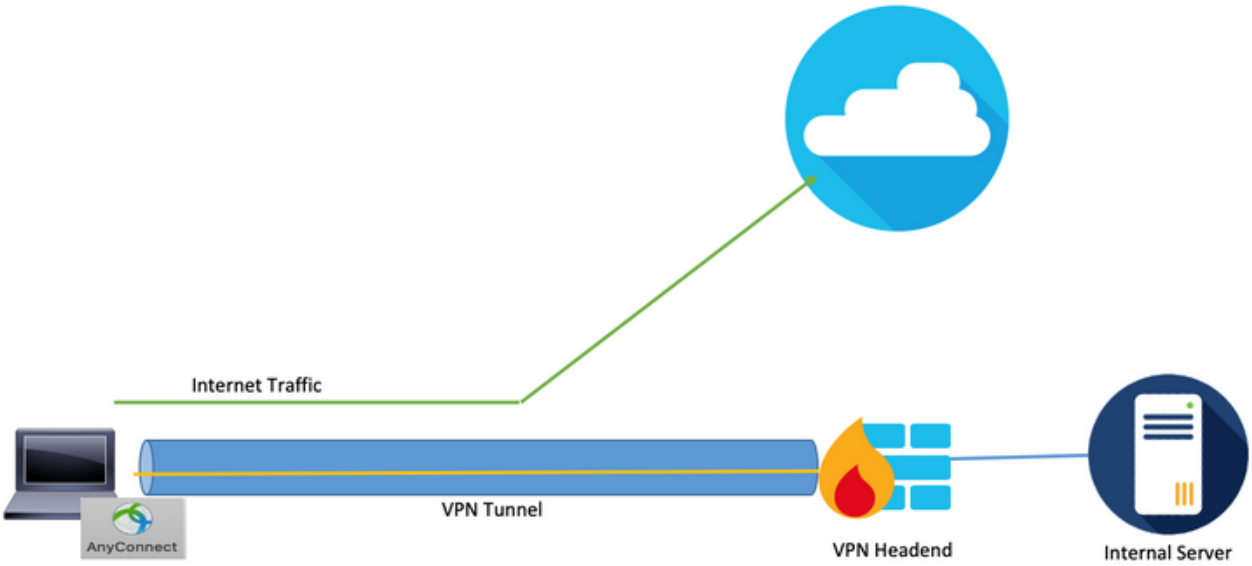


内部トラフィックの増加に偏っている傾向は、VPNユーザの大半が内部リソースにアクセスすることを意味します。そのため、負荷を増やすには、より大きなボックスへのアップグレードを計画するか、VPNロードバランシングなどの概念で負荷を共有することが重要です。

場合によっては、VPNのキャパシティがしきい値未満のままである可能性がありますが、VPNユーザ数の増加によって、現在の設定済みVPNプールが枯渇する可能性があります。このような場合、VPN IPプールを増やします。

ただし、VPNトラフィックの大部分が外部であることが傾向で示されている場合は、スプリットトンネリングを使用できます。

スプリットトンネリング機能の使用



これは、ユーザシステムからトンネルを通過する特定のトラフィックセットだけを転送し、残りのトラフィックはVPN暗号化なしでデフォルトゲートウェイに転送する機能です。したがって、VPNコンセントレータの負荷を軽減するために、内部ネットワーク宛てのトラフィックだけがトンネルを経由してルーティングされ、インターネットトラフィックはユーザのローカルISPを経由して転送されます。これは効果的な方法であり、広く採用されていますが、いくつかのリスクがあります。

従業員が保護されていないネットワークを介して一部のソーシャルメディアサイトに迅速にアクセスすると、職場に設定されている多層防御のセキュリティレイヤが不足して、ラップトップが会社全体に広がるマルウェアに感染する可能性があります。感染すると、侵害されたデバイスはインターネットから信頼できるセグメントへのピボットポイントになり、境界防御がバイパスされる可能性があります。

この機能を利用しながらリスクを軽減する1つの方法は、優れたデータの衛生やDuo Securityとの互換性など、厳しいセキュリティ基準を通過するクラウドサービスに対してのみ、スプリットトンネリングを使用することです。この方法を採用すると、以前に確認された大量の外部トラフィックが、これらのセキュアなクラウドサービスに送信される場合に役立ちます。これにより、VPNユーザがアクセスするWebアプリケーションを分析する必要が生じます。

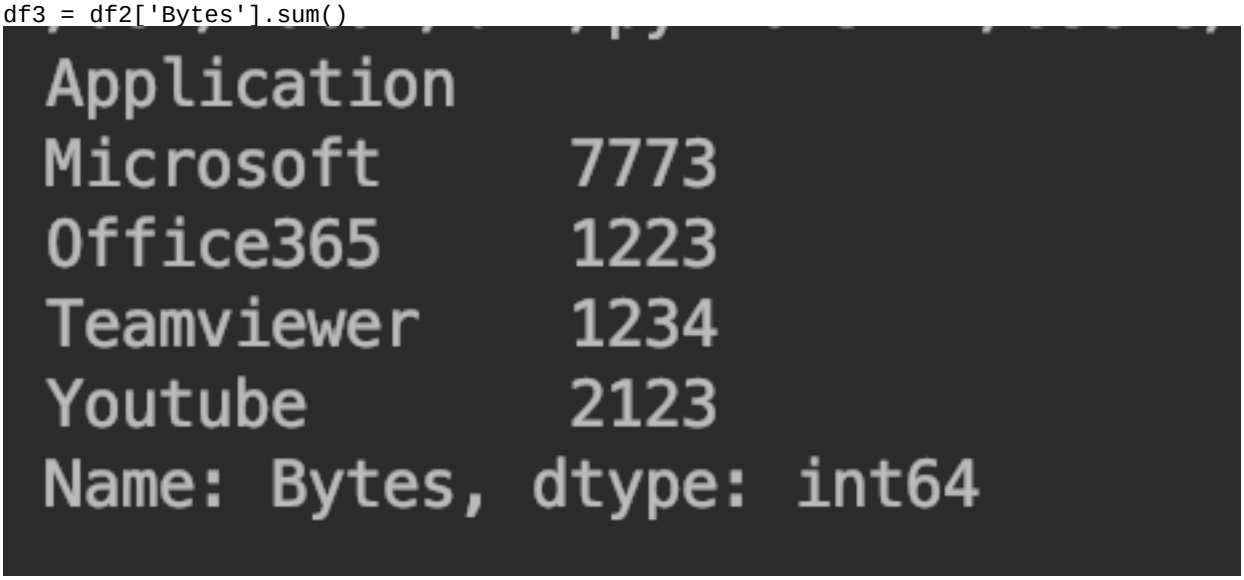
Cisco Firepower Threat Defense(FTD)などの次世代ファイアウォールのほとんどは、イベントに関連するアプリケーション情報をログに記録しています。pythonの**csvライブラリとpandasデータ操作機能を使用して、このログデータを解析およびクリーニングする**と、上記と同様のデータセットを提供でき、アクセスするアプリケーションをマッピングできます。

```
#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

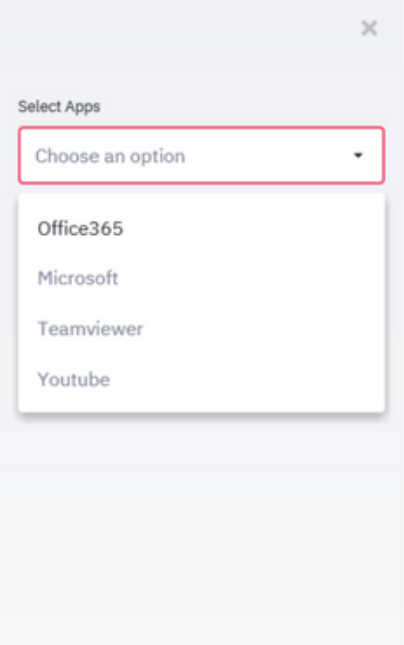
上記のデータフレームが取得されたら、pandasを介したアプリケーションに基づいて外部トラフィックの合計を分類できます。

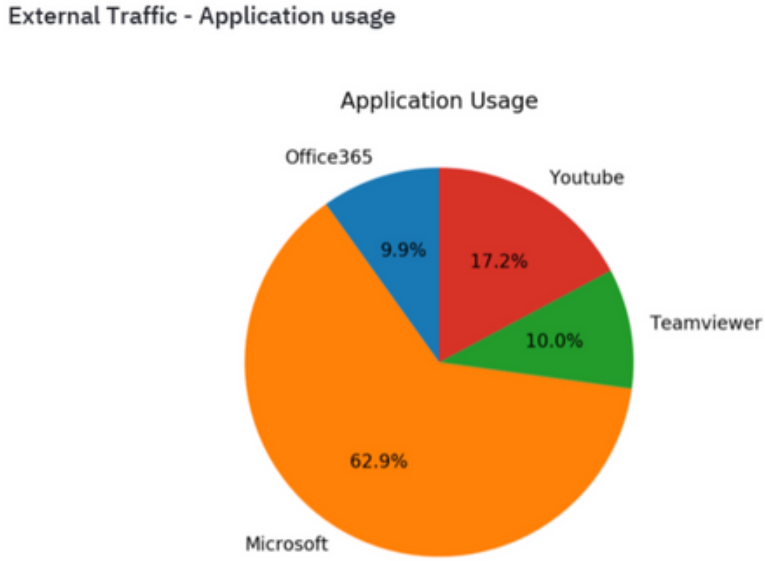
```
df2 = df.groupby('Application')

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



Streamlitを使用すると、トラフィック全体における各アプリケーションの共有をグラフィカルに表示できます。これにより、データを含めるための時間枠を柔軟に変更したり、コードを変更することなくユーザインタフェース自体のアプリケーションをフィルタで除外したりできるため、分析が簡単で正確になります。

```
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()
```



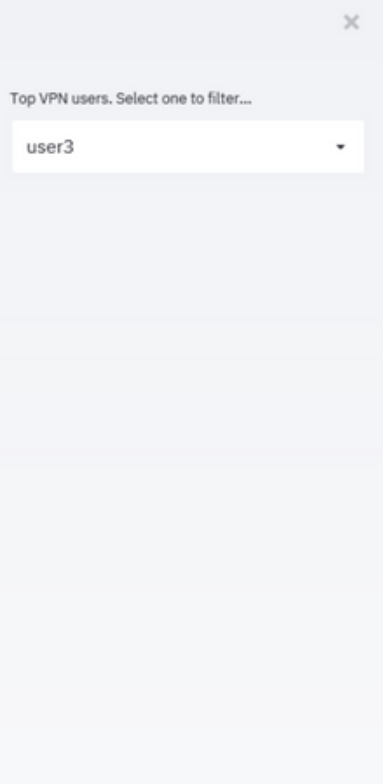


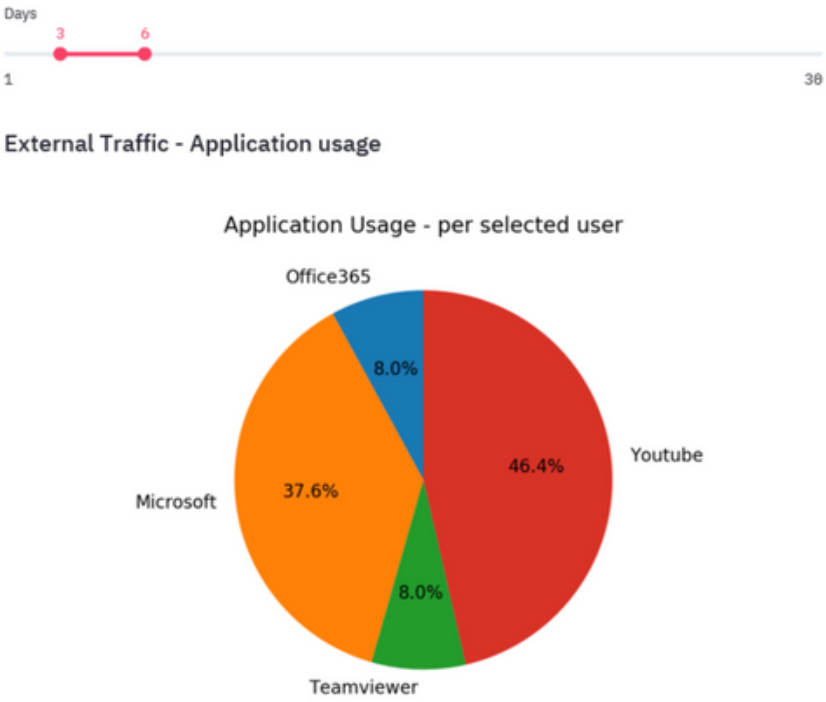
これにより、VPNユーザが一定期間にわたって使用している上位のWebアプリケーションを特定するプロセスが簡素化され、これらのアプリケーションがクラウドサービスの保護を目的としているかどうかが簡単になります。

最も大容量のアプリケーションがセキュアなクラウドサービスを特定する目的で使用される場合、スプリットトンネルで使用できるため、VPNコンセントレータの負荷が軽減されます。ただし、上位のアプリケーションがセキュリティの低いサービスやリスクを引き起こす可能性のあるサービスである場合は、VPNトンネルを通過する方が安全です。その理由は、他のネットワークセキュリティデバイスは、そのようなトラフィックを許可する前にトラフィックを処理できるためです。その後、ファイアウォールのアクセスポリシーを使用して、外部ネットワークへのアクセスを制限できます。

ID個人の非準拠VPNユーザ

場合によっては、サージが特定のポリシーに準拠していない少数のユーザにのみ関連付けられる可能性があります。上記で使用したモジュールとデータセットを再利用して、アクセスする上位VPNユーザとWebアプリケーションを特定できます。これにより、このようなユーザを分離し、デバイスの負荷に対するユーザの影響を確認できます。





いずれの方法にも適合しないシナリオでは、管理者は、AMP for Endpointsソリューションや Cisco Umbrellaソリューションなどのエンドポイントセキュリティソリューションを調べて、保護されていないネットワークのエンドポイントを保護する必要があります。