Approche programmatique pour optimiser la configuration du VPN d'accès à distance via l'analyse des données

Contenu

[Introduction](#)
[Conditions préalables](#)
[Conditions requises](#)
[Components Used](#)
[Problème](#)
[Solution](#)
[Analyse initiale basée sur les utilisateurs VPN et les connexions simultanées](#)
[Identifier la tendance du trafic vers les réseaux internes ou externes](#)
[Utilisation de la fonction de fractionnement en canaux](#)
[Utilisateurs VPN non conformes individuels d'identité](#)

Introduction

Ce document décrit comment surveiller et optimiser le VPN d'accès à distance configuré via certains des modules de programmation et outils open source disponibles aujourd'hui. Aujourd'hui, de nombreuses données sont générées dans les plus petits réseaux, même ceux qui peuvent être exploités pour obtenir des informations utiles. L'application d'analyses sur ces données collectées permet de prendre des décisions commerciales plus rapides et plus éclairées, étayées par des faits.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- VPN d’accès à distance
- Concepts de base de programmation Python

Components Used

Ce document n'est pas limité à des versions spécifiques du logiciel et du matériel Cisco ASA ou FTD.

> **Note**: Pandas, Streamlit, CSV et Matplotlib sont quelques bibliothèques Python utilisées.

The information in this document was created from the devices in a specific lab environment. All of

the devices used in this document started with a cleared (default) configuration. Si votre réseau est actif, assurez-vous de comprendre l'impact potentiel de n'importe quel script python et de commande.

Problème

Avec l'adoption par de nombreuses entreprises du modèle Travail à domicile pour une majorité de leurs employés partout dans le monde, le nombre d'utilisateurs qui comptent sur un VPN pour effectuer leur travail a considérablement augmenté. Ceci a conduit à une augmentation soudaine et considérable de la charge sur les concentrateurs VPN, conduisant les administrateurs à repenser et replanifier leurs configurations VPN. Pour prendre des décisions éclairées en vue de réduire la charge sur les concentrateurs ASA, il faut recueillir un large éventail d'informations auprès des périphériques sur une certaine période de temps et évaluer ces informations, ce qui est une tâche complexe et nécessiterait un temps considérable si elles sont effectuées manuellement.

Solution

Avec plusieurs modules Python et des outils open source disponibles aujourd'hui pour la programmabilité du réseau et l'analyse des données, la programmation peut s'avérer très utile pour la collecte et l'analyse des données, la planification et l'optimisation de la configuration VPN.

Analyse initiale basée sur les utilisateurs VPN et les connexions simultanées

Pour commencer l'analyse, obtenez le nombre d'utilisateurs connectés, les connexions simultanées établies et leur impact sur la bande passante. Les sorties de commande Cisco ASA suivantes fournissent les détails suivants :

- **show vpn-sessiondb anyconnect**
- **show conn**

Le module Python **Netmiko** peut être utilisé pour ssh sur le périphérique, exécuter les commandes et analyser les sorties.

```
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)
```

Collectez le nombre d'utilisateurs VPN et le nombre de connexions à intervalles réguliers (toutes les 2 heures peuvent être un bon début) dans une liste et obtenez le nombre journalier maximum pour une journée.

```
#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 est une bibliothèque efficace d'analyse et de manipulation de données et toutes les données analysées peuvent être stockées sous forme de séries ou de trames de données dans les pandas, ce qui facilite les opérations sur les données.

```
import pandas as pd

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

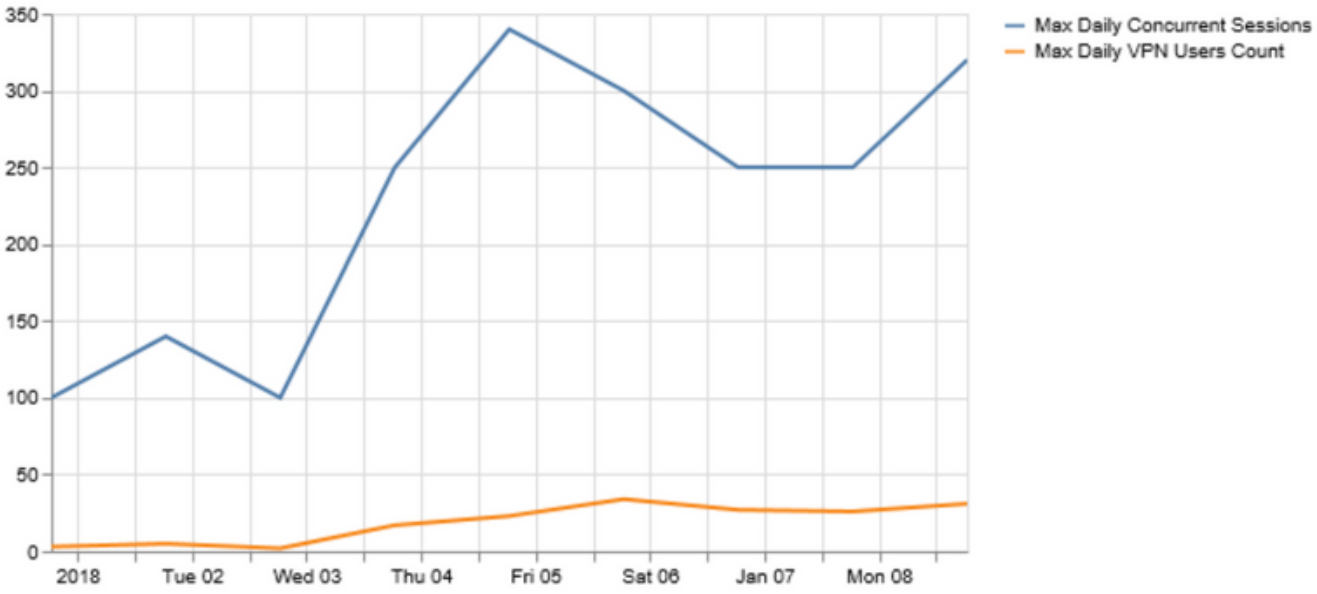
Daily Max VPN user Count - Max concurrent count

	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

Analyser le **nombre maximal d'utilisateurs VPN** et **le nombre maximal de connexions simultanées** qui peuvent aider à déterminer la nécessité d'optimiser les paramètres VPN.

Utilisez la fonction de traçage dans pandas et la bibliothèque **matplotlib**, comme l'illustre l'image ici.

```
df.plot()

matplotlib.pyplot.show()
```



Si le nombre d'utilisateurs VPN ou de connexions simultanées se rapproche de la capacité de la tête de réseau VPN, cela peut provoquer ces problèmes :

- Nouveaux utilisateurs VPN abandonnés.
- De nouvelles connexions de données via l'ASA sont abandonnées et les utilisateurs ne peuvent pas accéder aux ressources.
- CPU et/ou mémoire élevés.

La tendance sur une période de temps peut aider à déterminer si la boîte atteint son seuil.

Identifier la tendance du trafic vers les réseaux internes ou externes

Show conn output sur Cisco ASA peut fournir des détails supplémentaires tels que si le trafic est destiné à des réseaux internes ou externes et combien de données par flux sont transmises par le pare-feu.

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

L'utilisation du module **Netaddr** python facilite le fractionnement de la table de connexion obtenue

en flux vers des réseaux externes et internes.

```
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]
```

Voici l'image du trafic interne.

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

Voici l'image du trafic externe.

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

Par conséquent, fournir une idée du pourcentage de trafic VPN destiné aux réseaux internes et de la quantité de trafic sortant d'Internet. La collecte de ces informations sur une période donnée et l'analyse de leur tendance peuvent aider à déterminer si le trafic VPN est principalement externe ou interne.

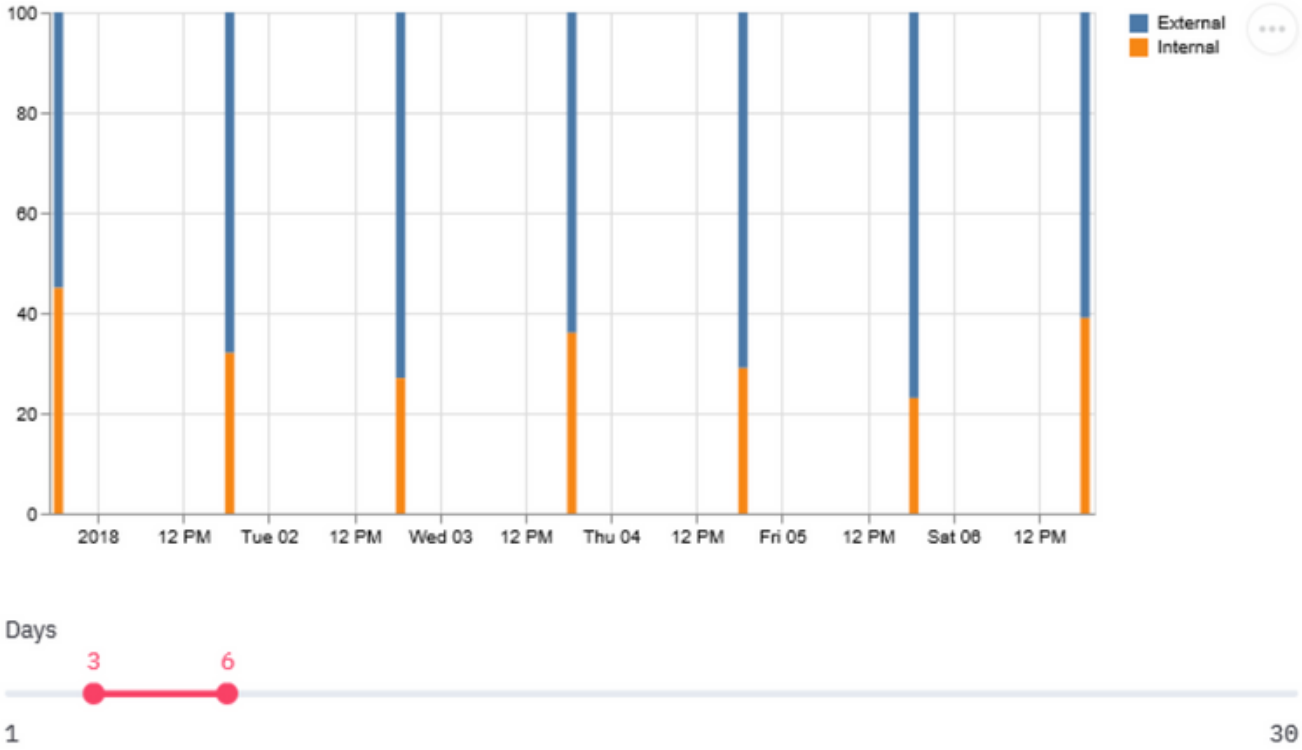
VPN Usage

Traffic Segregation - Internal and External

	External	Internal
Jan 1, 2018	55	45
Jan 2, 2018	68	32
Jan 3, 2018	73	27
Jan 4, 2018	64	36
Jan 5, 2018	71	29
Jan 6, 2018	77	23
Jan 7, 2018	61	39

Des modules comme **Streamlit** permettent non seulement de convertir les données tabulaires en représentation graphique, mais aussi d'y apporter des modifications en temps réel pour faciliter l'analyse. Il peut modifier la fenêtre de temps des données collectées ou ajouter des données supplémentaires aux paramètres surveillés.

```
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)
```

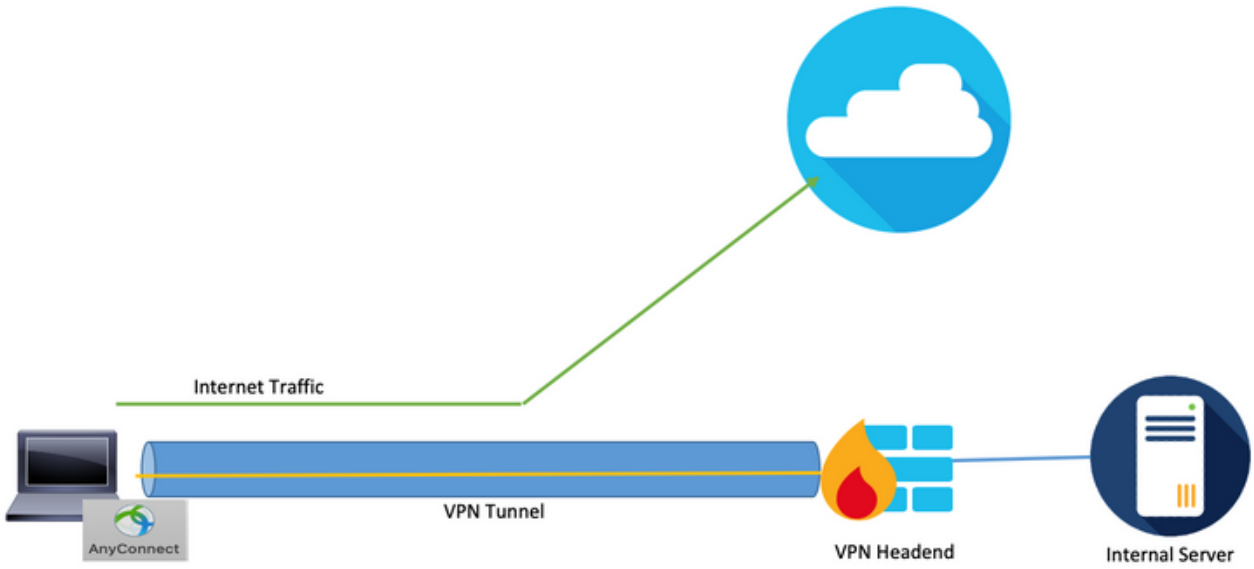


Une tendance vers un trafic interne plus élevé pourrait signifier que la majorité des utilisateurs VPN accèdent aux ressources internes. Par conséquent, pour répondre à cela, augmenter la charge, il est important de planifier des mises à niveau vers des boîtes plus grandes ou de partager la charge avec des concepts tels que l'équilibrage de charge VPN.

Dans certains cas, la capacité VPN peut être encore inférieure au seuil, mais une augmentation du nombre d'utilisateurs VPN peut épuiser le pool VPN configuré actuel. Dans ce cas, augmentez le pool d'adresses IP VPN.

Cependant, si la tendance indique que la majorité du trafic VPN est externe, vous pouvez utiliser la transmission tunnel partagée.

Utilisation de la fonction de fractionnement en canaux



Il s'agit d'une fonctionnalité qui transfère uniquement un ensemble spécifique de trafic via le tunnel à partir du système utilisateur et le reste du trafic est transféré à la passerelle par défaut sans cryptage VPN. Par conséquent, pour réduire la charge sur le concentrateur VPN, seul le trafic destiné au réseau interne peut être acheminé via le tunnel et le trafic Internet peut être transmis via le FAI local de l'utilisateur. Il s'agit d'une méthode efficace et largement adoptée, mais elle comporte certains risques.

Un employé accède à certains sites de médias sociaux sur des réseaux non protégés pour une pause rapide peut infecter son ordinateur portable par un programme malveillant qui se propage dans l'entreprise en raison d'un manque de couches de sécurité de défense en profondeur configurées sur le lieu de travail. Une fois infecté, le périphérique compromis pourrait devenir un point de pivot d'Internet vers le segment de confiance, avec contournement des défenses périmétriques.

Une façon de réduire les risques tout en utilisant cette fonctionnalité serait d'utiliser la transmission tunnel partagée uniquement pour les services cloud qui répondent à des critères de sécurité rigoureux, notamment une bonne hygiène des données et une compatibilité avec Duo Security. L'adoption de cette approche vous aidera si une bonne partie du trafic externe observé précédemment est destinée à ces services cloud sécurisés. Il est donc nécessaire d'analyser les applications Web auxquelles accèdent les utilisateurs VPN.

La plupart des pare-feu de nouvelle génération, tels que Cisco Firepower Threat Defense (FTD), contiennent des informations d'application associées à l'événement dans les journaux. L'analyse et le nettoyage de ces données de journal avec les fonctions python **csv** et de manipulation de données pandas peuvent fournir un jeu de données similaire à celui ci-dessus avec un ajout des applications auxquelles on accède mappées.

```
#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

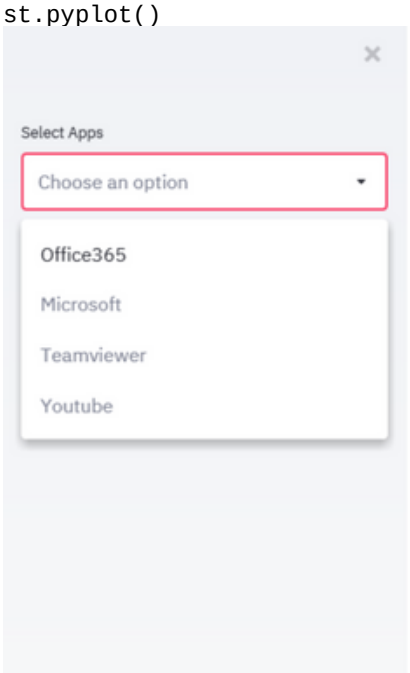
Une fois qu'une trame de données comme ci-dessus est obtenue, vous pouvez catégoriser le trafic externe total en fonction de l'application via pandas.

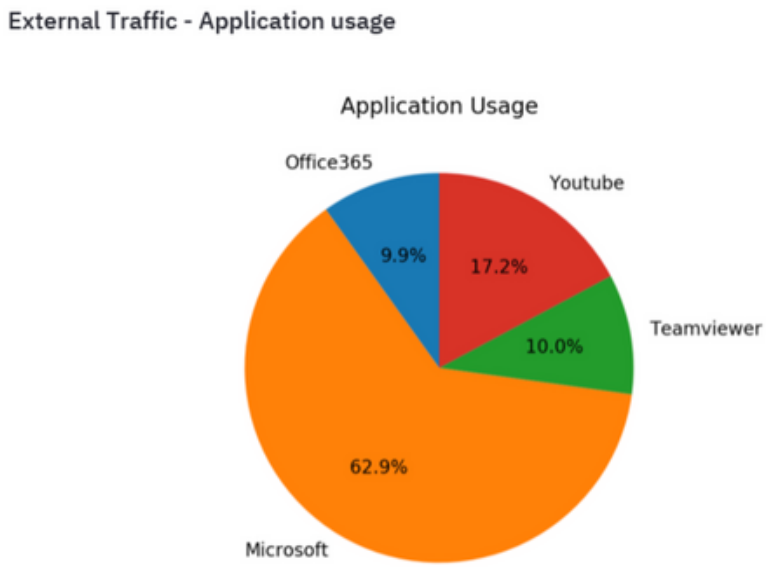
```
df2 = df.groupby('Application')

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

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

L'utilisation de Streamlit obtient à nouveau une représentation graphique de la part de chaque application dans le trafic total. Il permet de modifier la fenêtre de temps pour l'inclusion des données et de filtrer les applications sur l'interface utilisateur elle-même sans avoir à modifier le code, ce qui rend l'analyse facile et précise.

```
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()
```





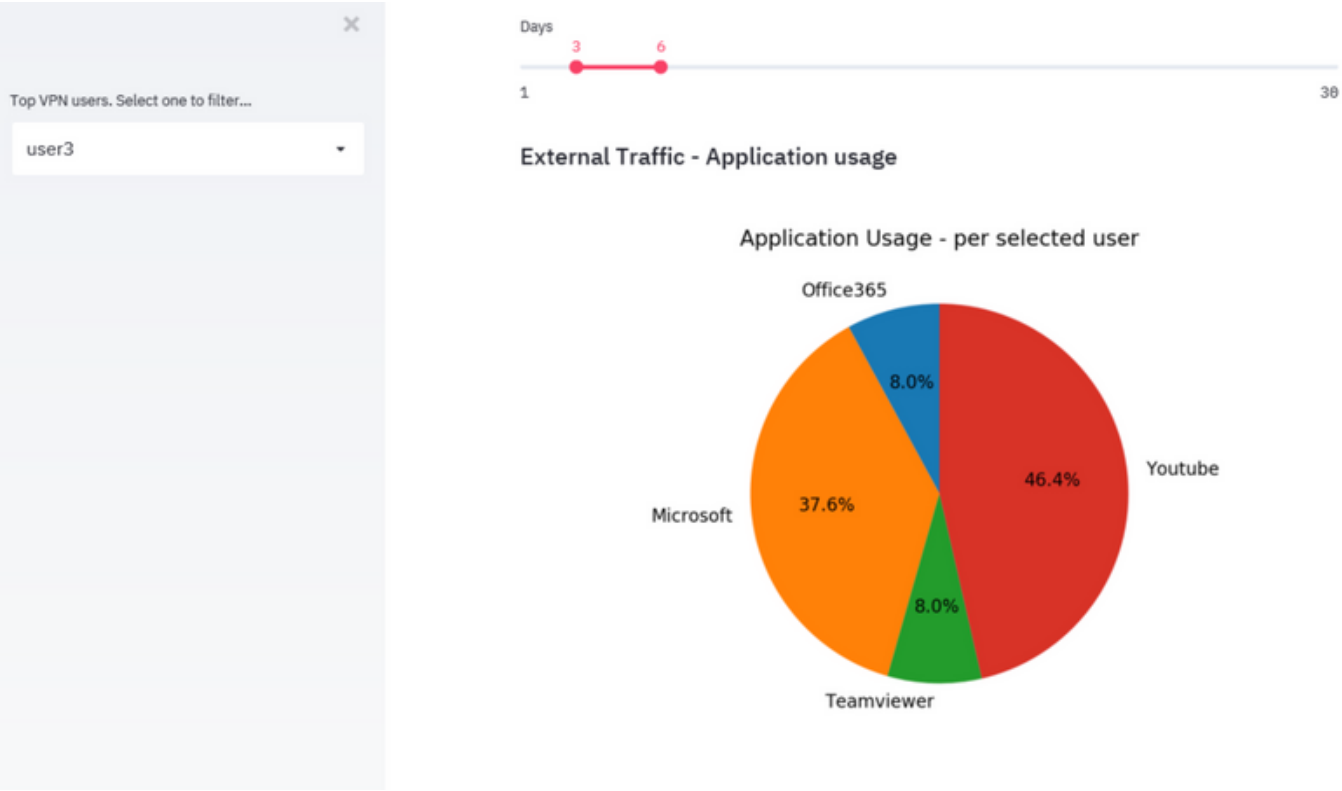
Cela peut simplifier le processus d'identification des principales applications Web utilisées par les utilisateurs VPN sur une certaine période et si ces applications doivent sécuriser les services cloud ou non.

Si les applications les plus volumineuses sont destinées à identifier des services cloud sécurisés,

elles peuvent être utilisées avec un tunnel partagé, réduisant ainsi la charge sur un concentrateur VPN. Cependant, si les principales applications sont destinées à des services moins sécurisés ou présentant un risque, il est plus sûr de les passer par le tunnel VPN. La raison en est que d'autres périphériques de sécurité réseau peuvent traiter le trafic avant qu'il ne permette le passage de ce trafic. Vous pouvez ensuite utiliser des stratégies d'accès sur les pare-feu pour limiter l'accès aux réseaux externes.

Utilisateurs VPN non conformes individuels d'identité

Dans certains cas, la poussée pourrait être associée à quelques utilisateurs seulement qui ne respectent pas certaines politiques. Les modules et les ensembles de données utilisés ci-dessus peuvent être à nouveau utilisés pour identifier les principaux utilisateurs VPN et les applications Web auxquelles ils accèdent. Cela peut aider à isoler ces utilisateurs et à observer leur effet sur la charge du périphérique.



Dans les scénarios où aucune des méthodes ne convient, les administrateurs doivent examiner les solutions de sécurité des points d'extrémité telles que la solution AMP for Endpoints et la solution Cisco Umbrella pour protéger les points d'extrémité dans les réseaux non protégés.