Programmatische benadering voor het optimaliseren van VPN-instelling voor externe toegang via gegevensanalyses

Inhoud

[Inleiding](#)
[Voorwaarden](#)
[Vereisten](#)
[Gebruikte componenten](#)
[Probleem](#)
[Oplossing](#)
[Eerste analyse gebaseerd op VPN-gebruikers en gelijktijdige verbindingen](#)
[Identificeer verkeerstrends is gericht op intern netwerk of externe netwerken](#)
[Gebruik de optie Split-tunneling](#)
[Identiteit individuele niet-conforme VPN-gebruikers](#)

Inleiding

Dit document beschrijft hoe u de VPN-instellingen voor externe toegang kunt bewaken en optimaliseren, die u hebt ingesteld met behulp van een aantal programmeermodules en opensource-tools die momenteel beschikbaar zijn. Er wordt vandaag veel data gegenereerd in zelfs de kleinste netwerken die kunnen worden gebruikt om nuttige informatie te verkrijgen. Door de analyse van deze verzamelde gegevens toe te passen, kunnen snellere, meer geïnformeerde bedrijfsbeslissingen worden genomen, die op feiten zijn gebaseerd.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Remote Access VPN
- Basisconcepten van de Python-programmering

Gebruikte componenten

Dit document is niet beperkt tot specifieke Cisco ASA of FTD software en hardwareversies.

> Opmerking: Pandas, Streamlit, CSV en Matplotlib zijn een paar Python-bibliotheken die gebruikt worden.

De informatie in dit document is gebaseerd op de apparaten in een specifieke

laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht en python scripts begrijpt.

Probleem

Omdat veel bedrijven voor het grootste deel van hun werknemers het Work From-model overnemen is het aantal gebruikers dat op VPN vertrouwt om hun baan te volbrengen aanzienlijk toegenomen. Dit heeft geleid tot een plotselinge en aanzienlijke toename van de lading op de VPN-concentrators, wat de beheerders ertoe heeft gebracht hun VPN-instellingen opnieuw te overwegen en te plannen. Het nemen van gefundeerde beslissingen om de belasting voor de ASA-concentrators te verminderen vereist dat er een breed scala aan informatie van de apparaten over een bepaalde periode wordt verzameld en dat de informatie wordt beoordeeld, wat een complexe taak is en een aanzienlijke hoeveelheid tijd zou vergen indien het handmatig wordt gedaan.

Oplossing

Nu er verschillende Python-modules en opensource-tools beschikbaar zijn voor netwerkprogrammeerbaarheid en gegevensanalyse, kan een programmering zeer nuttig blijken voor het verzamelen en analyseren van gegevens, het plannen en het optimaliseren van de VPN-instellingen.

Eerste analyse gebaseerd op VPN-gebruikers en gelijktijdige verbindingen

Om de analyse te beginnen verkrijgen het aantal gebruikers die, de gelijktijdige verbindingen gevestigd, en hun impact op bandbreedte verbinden. De volgende Cisco ASA opdracht outputs zullen deze details geven:

- toon vpn-sessiondb anyconnect
- show conn

De Python-module **Netmiko** kan worden gebruikt om de projector aan te passen, de opdrachten te gebruiken en de output te parseren.

```
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)
```

Verzamel het aantal VPN-gebruikers en de tellingen van de verbindingen met regelmatige tussenpozen (elke 2 uur kan een goede start zijn) in een lijst en verkrijgt het maximale aantal dagen per dag.

```
#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 is een efficiënte bibliotheek van gegevensanalyse en manipulatie en alle geparseerde gegevens kunnen worden opgeslagen als een serie of een gegevenskader in panda's die operaties van de gegevens gemakkelijk maken.

```
import pandas as pd

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

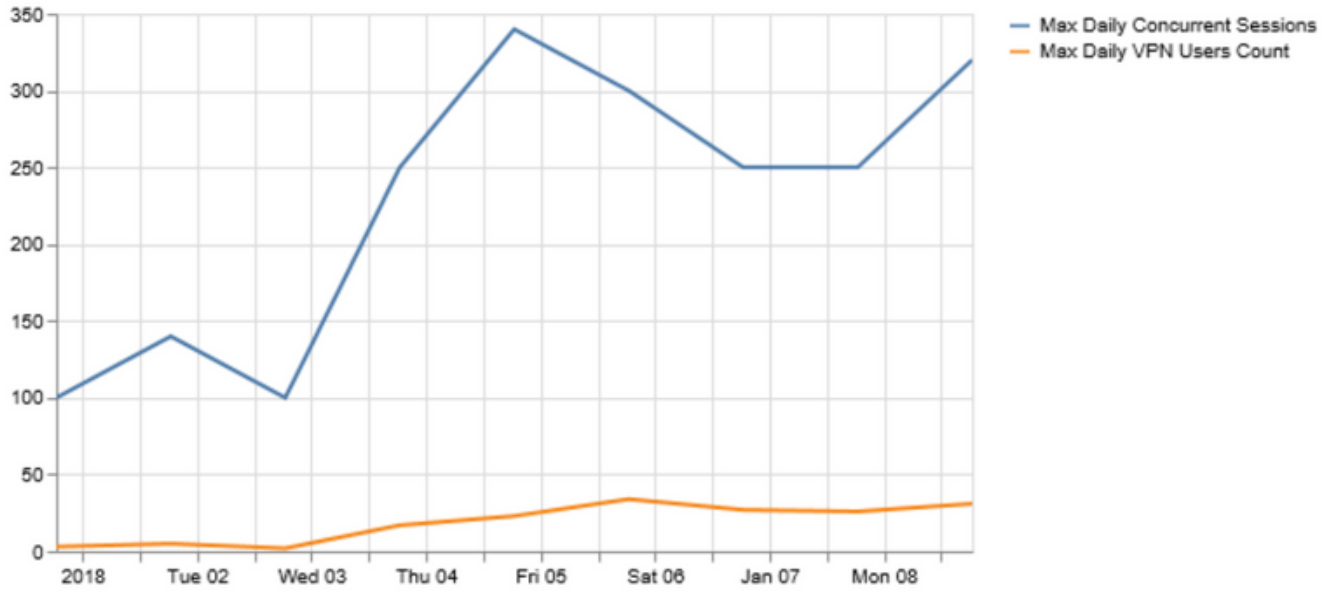
Daily Max VPN user Count - Max concurrent count

	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

Analyse van de **dagelijkse maximum aantal VPN-gebruikers** en **maximale gelijktijdige verbindingen** die kunnen helpen bij het bepalen van de noodzaak om de VPN-instellingen te optimaliseren.

Gebruik de plot functie in pandas en **matplotlib** bibliotheek, zoals hier in de afbeelding wordt getoond.

```
df.plot()

matplotlib.pyplot.show()
```



Als het aantal VPN-gebruikers of gelijktijdige verbindingen dichter bij de capaciteit van het VPN-head-end komt, kunnen deze problemen ontstaan:

- Nieuwe VPN-gebruikers worden verwijderd.
- Nieuwe gegevensverbindingen door de ASA die worden ingetrokken en gebruikers kunnen geen toegang tot de resources krijgen.
- Hoge CPU en/of geheugen.

De trend over een periode kan helpen bepalen of de doos zijn drempel bereikt.

Identificeer verkeerstrends is gericht op intern netwerk of externe netwerken

De uitvoer van verbindingen op Cisco ASA kan extra details geven zoals of het verkeer naar interne of externe netwerken is en hoeveel gegevens in bytes per flow door de firewall wordt doorgegeven.

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

Door gebruik van de pythonmodule van **Netaddr** is het gemakkelijk om de verkregen

verbindingstabel te splitsen in stromen naar externe netwerken en naar interne netwerken.

```
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]
```

Dit is het beeld van het Interne Verkeer.

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

Dit is het beeld van extern verkeer.

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

Hierin inzicht verschaffen in welk percentage van het VPN-verkeer bestemd is voor de interne netwerken en in hoeveel van het internetverkeer. Het verzamelen van deze informatie over een periode en de analyse van de trend ervan kunnen helpen bepalen of het VPN-verkeer voornamelijk extern of intern is.

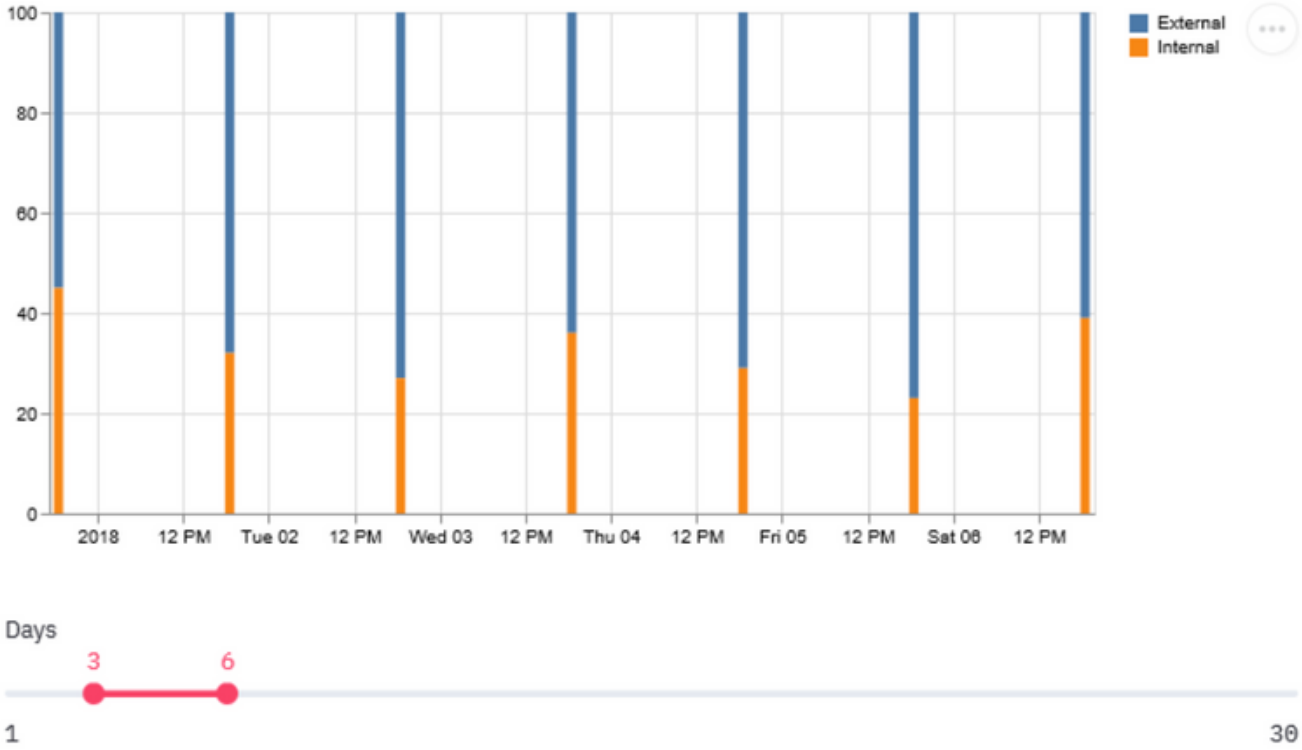
VPN Usage

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

Modules als **Streamlit** maken het mogelijk om niet alleen de tabelgegevens om te zetten naar een grafische voorstelling, maar passen ook aanpassingen toe in real-time om de analyse te ondersteunen. U kunt het tijdvenster van de verzamelde gegevens wijzigen of extra gegevens toevoegen aan de parameters die worden gemonitord.

```
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)
```

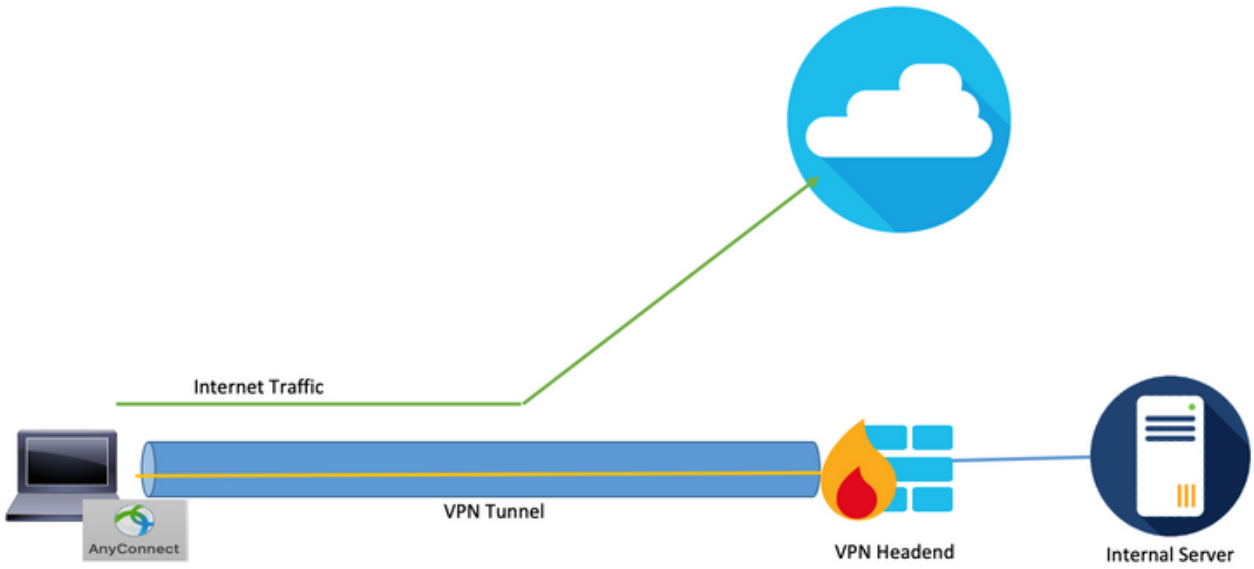


Een trend die richting hoger intern verkeer leidt zou kunnen betekenen dat de meeste VPN-gebruikers toegang hebben tot interne bronnen. Om dit te bereiken, is het, om lading te vergroten, belangrijk om upgrades te plannen aan grotere boxen of de lading te delen met concepten zoals de lading-in evenwicht brengen van VPN.

In sommige gevallen ligt de VPN-capaciteit mogelijk nog steeds onder de drempel, maar een toename in het aantal VPN-gebruikers kan de huidige geconfigureerde VPN-pool uitputten. In dergelijke gevallen, verhoog de VPN IP Pool.

Als echter de trend aantoont dat het grootste deel van het VPN-verkeer extern is, kunt u gesplitste tunneling gebruiken.

Gebruik de optie Split-tunneling



Het is een eigenschap die slechts een specifieke reeks verkeer door de tunnel van het gebruikerssysteem door stuurt en de rest van het verkeer wordt door gestuurd naar de standaardgateway zonder de encryptie van VPN. Om de belasting op de VPN-concentrator te verminderen, kan dus alleen het verkeer dat bestemd is voor het interne netwerk door de tunnel worden geleid en kan internetverkeer via de lokale ISP van de gebruiker worden doorgestuurd. Dit is een effectieve methode die op brede schaal wordt toegepast, maar er zijn een aantal risico's aan verbonden.

Een werknemer heeft toegang tot een aantal sociale media-sites via onbeschermde netwerken om snel zijn laptop te infecteren met malware die zich over het bedrijf verspreidt door een gebrek aan diepgaande beveiligingslagen op de werkvloer. Na infectie zou het gecompromitteerde apparaat een keerpunt kunnen worden van het internet naar het vertrouwde segment, waarbij de perimeter verdediging wordt omzeild.

Eén manier om de risico's terug te dringen terwijl deze optie wordt gebruikt, zou kunnen zijn door alleen splitsingen te gebruiken voor wolkendiensten die voldoen aan strenge veiligheidscriteria, waaronder een goede gegevenshygiëne en compatibiliteit met Duo Security. Het aannemen van dit besluit zal helpen als een groot deel van het eerder waargenomen externe verkeer bestemd is voor deze beveiligde clouddiensten. Dit maakt het noodzakelijk om webtoepassingen te analyseren die toegankelijk zijn voor VPN-gebruikers.

De meeste next-generation firewalls zoals Cisco Firepower Threat Defense (FTD) bevatten toepassingsinformatie die bij de gebeurtenis in weblogs is gekoppeld. Door deze loggegevens te parseren en te reinigen met **bibliotheken** van csv-python en pandas kan de gegevensmanipulatiefunctie een vergelijkbaar dataset als hierboven bieden, waarbij de applicaties die er toegang toe hebben, in kaart worden gebracht.

```
#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

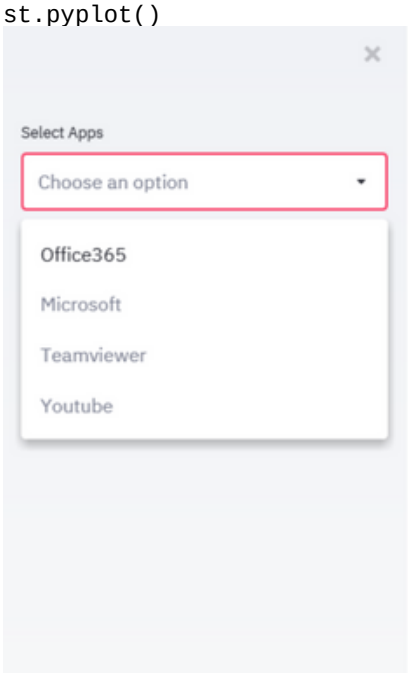
Zodra een gegevenskader zoals hierboven is bereikt, kunt u het totale externe verkeer op basis van de toepassing door panda's categoriseren.

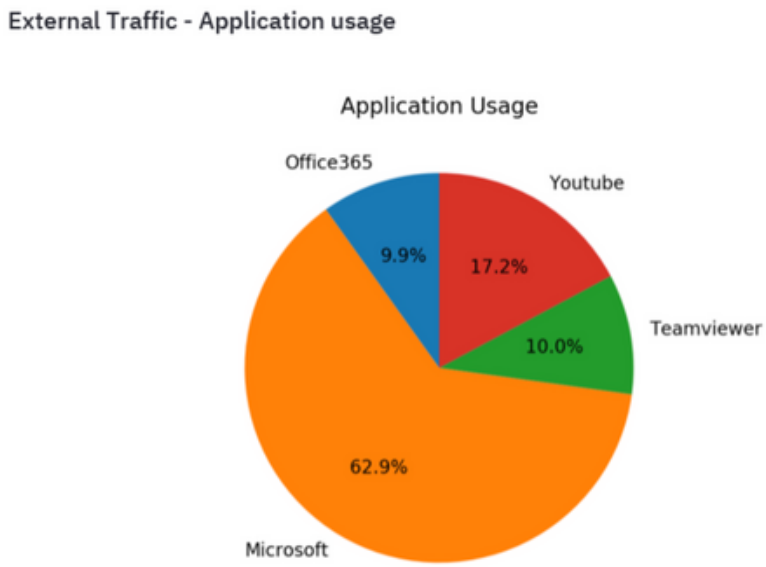
```
df2 = df.groupby('Application')

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

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

Gebruik van Streamlit verkrijgt opnieuw een grafische weergave van het aandeel van elke toepassing in het totale verkeer. Het biedt de flexibiliteit om het tijdvenster voor de opname van gegevens te wijzigen en toepassingen op de gebruikersinterface zelf te filteren zonder dat de code hoeft te worden gewijzigd, wat de analyse eenvoudig en nauwkeurig maakt.

```
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()
```





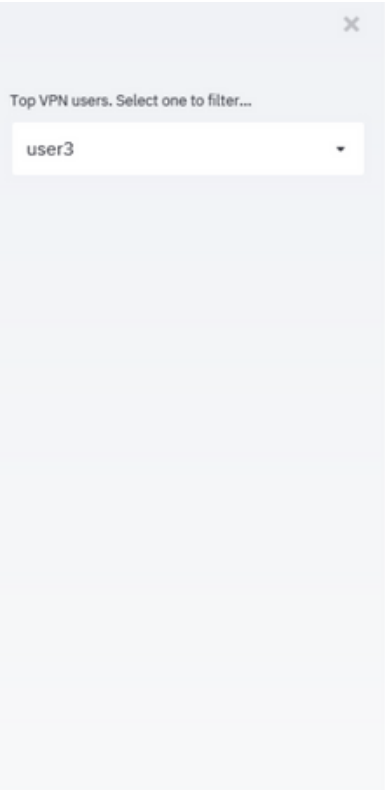
Dit kan het proces vereenvoudigen van identificatie van de top web toepassingen die door VPN-gebruikers over een tijdsperiode worden gebruikt en als deze toepassingen cloudservices moeten beveiligen.

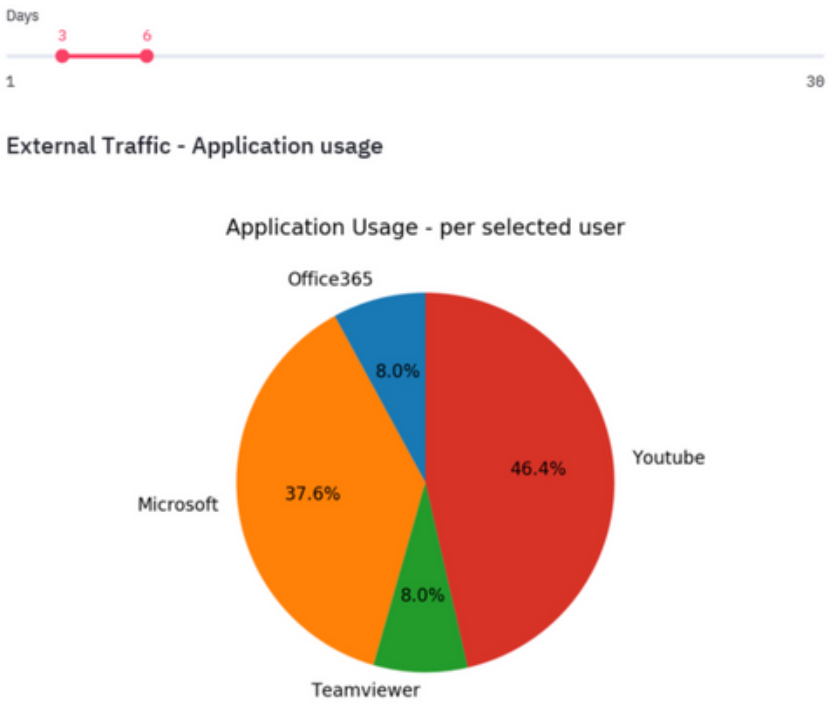
Als de meest grootschalige toepassingen bedoeld zijn om beveiligde cloudservices te

identificeren, kunnen ze worden gebruikt met een gesplitste tunnel en zo de lading op een VPN-concentrator verminderen. Als de bovenste toepassingen echter betrekking hebben op diensten die minder veilig zijn of een risico kunnen opleveren, is het veiliger om ze door de VPN-tunnel te laten passeren. Reden is dat andere netwerkbeveiligingsapparaten het verkeer kunnen verwerken voordat ze dat verkeer toestaan. U kunt dan toegangsbeleid op de firewalls gebruiken om de toegang tot externe netwerken te beperken.

Identiteit individuele niet-conforme VPN-gebruikers

In sommige gevallen zou de toename in verband kunnen worden gebracht met slechts een paar gebruikers die niet aan bepaalde beleidsmaatregelen voldoen. De modules en datasets die hierboven worden gebruikt kunnen opnieuw worden gebruikt om de top VPN-gebruikers en de webtoepassingen te identificeren waartoe ze toegang hebben. Dit kan helpen om deze gebruikers te isoleren en hun effect op de lading van de apparatuur te meten.





In scenario's, waar geen van de methodes past, zouden de beheerders moeten kijken naar de oplossingen van de eindpunt zoals AMP voor de oplossing van Endpoints en de oplossing van Cisco Umbrella om de eindpunten in onbeschermde netwerken te beschermen.