Approccio programmatico per ottimizzare la configurazione della VPN ad accesso remoto tramite l'analisi dei dati

Sommario

[Introduzione](#)
[Prerequisiti](#)
[Requisiti](#)
[Componenti usati](#)
[Problema](#)
[Soluzione](#)
[Analisi iniziale basata sugli utenti VPN e sulle connessioni simultanee](#)
[Individua tendenza traffico verso rete interna o reti esterne](#)
[Utilizzo della funzione di tunneling ripartito](#)
[Identity Individual Non-Compliant VPN Users](#)

Introduzione

In questo documento viene descritto come monitorare e ottimizzare la VPN ad accesso remoto configurata tramite alcuni dei moduli di programmazione e degli strumenti open source attualmente disponibili. Molti dati vengono generati oggi anche nelle più piccole reti che possono essere sfruttate per ottenere informazioni utili. L'applicazione di analisi su questi dati raccolti consente di prendere decisioni aziendali più informate e più rapide, supportate dai fatti.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- VPN ad accesso remoto
- Concetti base della programmazione Python

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware Cisco ASA o FTD.

> **Nota:** Pandas, Streamlit, CSV e Matplotlib sono alcune librerie Python che vengono utilizzate.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata

ripristinata ai valori predefiniti. Se la rete è operativa, accertarsi di comprendere il potenziale impatto di qualsiasi comando e script Python.

Problema

Con l'adozione del modello "Work From Home" da parte di molte aziende per la maggior parte dei propri dipendenti in tutto il mondo, il numero di utenti che si affidano a VPN per svolgere il proprio lavoro è aumentato notevolmente. Ciò ha portato a un aumento improvviso e considerevole del carico sui concentratori VPN, che ha portato gli amministratori a ripensare e ripianificare le proprie configurazioni VPN. Per prendere decisioni informate e ridurre il carico sui concentratori ASA, è necessario raccogliere un'ampia gamma di informazioni dai dispositivi in un determinato periodo di tempo e valutare le informazioni; si tratta di un'attività complessa che, se eseguita manualmente, richiederebbe molto tempo.

Soluzione

Con diversi moduli Python e strumenti open source disponibili oggi per la programmabilità della rete e l'analisi dei dati, la programmazione può rivelarsi molto utile nella raccolta e nell'analisi dei dati, nella pianificazione e nell'ottimizzazione della configurazione VPN.

Analisi iniziale basata sugli utenti VPN e sulle connessioni simultanee

Per avviare l'analisi, ottenere il numero di utenti che si connettono, le connessioni simultanee stabilite e il loro impatto sulla larghezza di banda. I seguenti output del comando Cisco ASA forniscono questi dettagli:

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

Il modulo Python **Netmiko** può essere usato per eseguire il protocollo SSH sul dispositivo, eseguire i comandi e analizzare gli output.

```
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)
```

Raccogliere il numero di utenti VPN e il numero di connessioni a intervalli regolari (ogni 2 ore può essere un buon inizio) in un elenco e ottenere il numero massimo giornaliero per un giorno.

```
#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 è un'efficiente raccolta di analisi e manipolazione dei dati e tutti i dati analizzati possono essere memorizzati come una serie o un frame di dati in panda rendendo le operazioni sui dati facili.

```
import pandas as pd

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

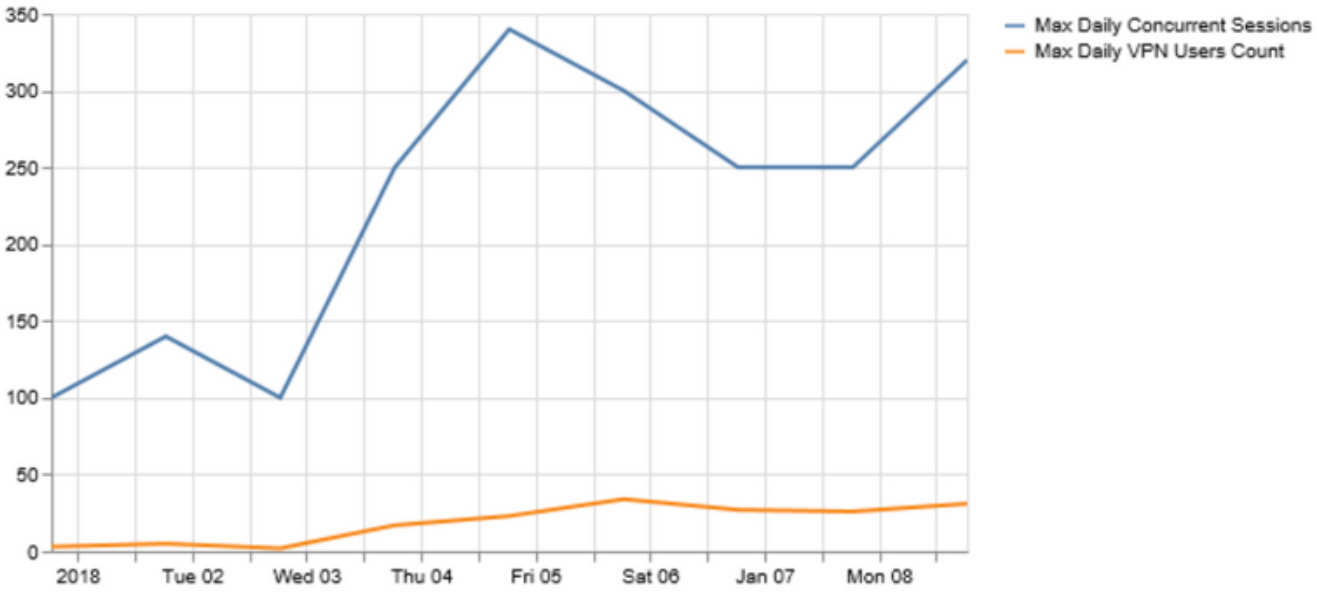
Daily Max VPN user Count - Max concurrent count

	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

Analizza il **numero massimo giornaliero di utenti VPN** e il **numero massimo di connessioni simultanee** che possono aiutare a determinare la necessità di ottimizzare le impostazioni VPN.

Utilizzate la funzione di plottaggio nella libreria pandas e **matplotlib**, come mostrato nell'immagine.

```
df.plot()

matplotlib.pyplot.show()
```



Se il numero di utenti VPN o connessioni simultanee si avvicina alla capacità dell'headend VPN, potrebbero verificarsi i seguenti problemi:

- Eliminazione di nuovi utenti VPN.
- Le nuove connessioni dati tramite l'ASA vengono eliminate e gli utenti non possono accedere alle risorse.
- CPU e/o memoria elevate.

La tendenza in un periodo di tempo può aiutare a determinare se la casella sta raggiungendo la soglia.

Individua tendenza traffico verso rete interna o reti esterne

Show conn output on Cisco ASA può fornire dettagli aggiuntivi come se il traffico sia diretto a reti interne o esterne e la quantità di dati in byte per flusso passata attraverso il firewall.

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'utilizzo del modulo Python di **Netaddr** semplifica la suddivisione della tabella delle connessioni

ottenute in flussi verso reti esterne e verso reti interne.

```
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]
```

L'immagine mostra il traffico interno.

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

Immagine del traffico esterno.

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

Fornendo così una visione di quale percentuale del traffico VPN è destinata alle reti interne e di quanto va verso Internet. La raccolta di queste informazioni in un periodo di tempo e l'analisi della relativa tendenza possono aiutare a determinare se il traffico VPN è prevalentemente esterno o interno.

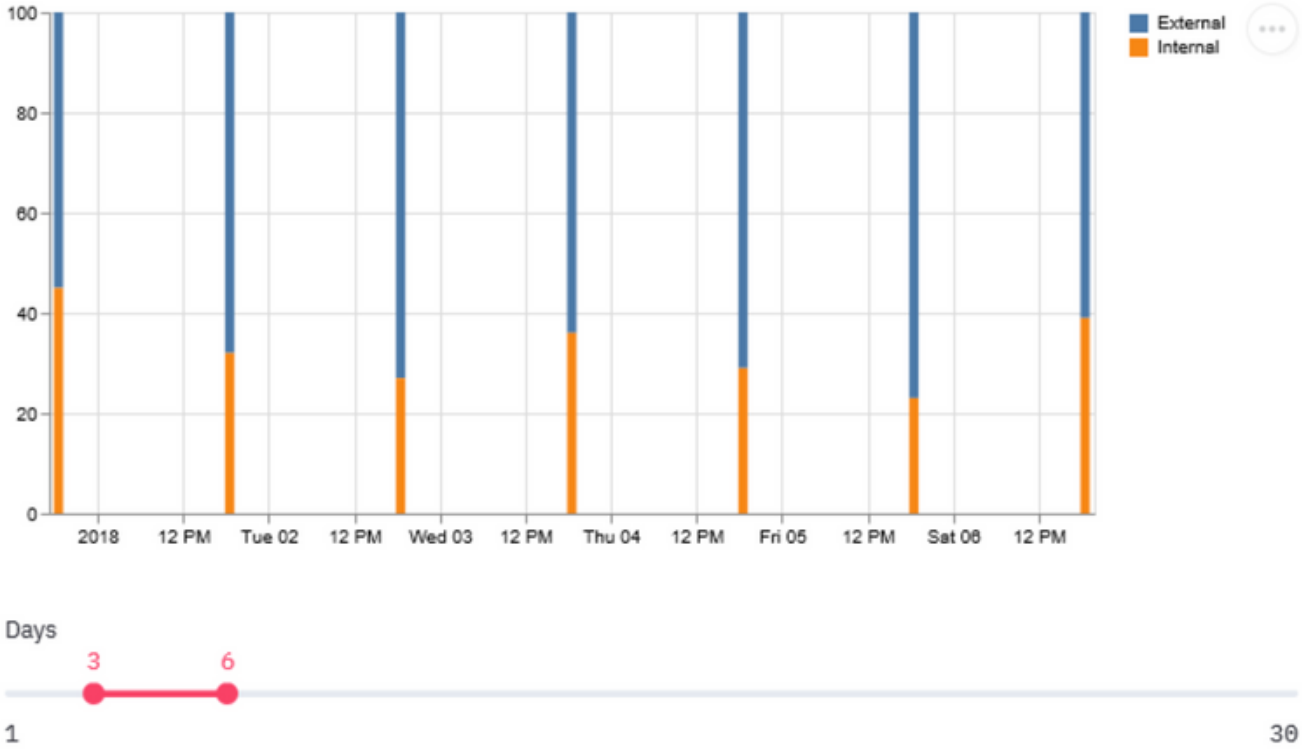
VPN Usage

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

I moduli come **Streamlit** consentono non solo di convertire i dati tabulari in una rappresentazione grafica, ma anche di modificarli in tempo reale per facilitare l'analisi. Può modificare la finestra temporale dei dati raccolti o aggiungere ulteriori dati ai parametri monitorati.

```
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)
```

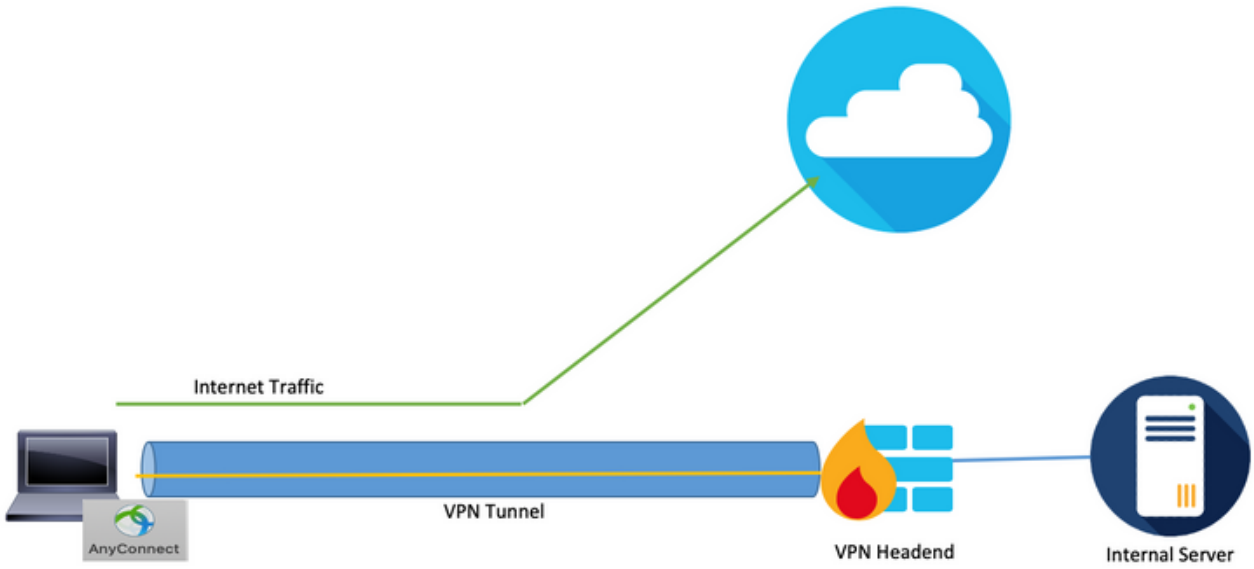


Una tendenza verso un traffico interno più elevato potrebbe significare che la maggior parte degli utenti VPN accede alle risorse interne. Pertanto, per far fronte a questo, aumento del carico, è importante pianificare gli aggiornamenti a caselle più grandi o condividere il carico con concetti come il load balancing VPN.

In alcuni casi, la capacità della VPN potrebbe essere ancora al di sotto della soglia, ma un aumento del numero di utenti VPN può esaurire il pool VPN attualmente configurato. In questi casi, aumentare il pool IP della VPN.

Tuttavia, se la tendenza mostra che la maggior parte del traffico VPN è esterno, è possibile usare il tunneling suddiviso.

Utilizzo della funzione di tunneling ripartito



Èuna funzionalità che inoltra solo uno specifico gruppo di traffico attraverso il tunnel dal sistema utente e il resto del traffico viene inoltrato al gateway predefinito senza crittografia VPN. Pertanto, per ridurre il carico sul concentratore VPN, solo il traffico destinato alla rete interna potrebbe essere indirizzato attraverso il tunnel e il traffico Internet potrebbe essere inoltrato attraverso l'ISP locale dell'utente. Si tratta di un metodo efficace e ampiamente adottato, ma presenta alcuni rischi.

Un dipendente che accede ad alcuni siti di social media su reti non protette per una breve interruzione può infettare il proprio laptop con malware che si diffonde nell'azienda a causa della mancanza di livelli di sicurezza dettagliati per la difesa impostati sul posto di lavoro. Una volta infettato, il dispositivo compromesso potrebbe diventare un punto di rotazione da Internet nel segmento di fiducia, con bypassato le difese perimetrali.

Un modo per ridurre il rischio utilizzando questa funzione sarebbe utilizzare il tunneling ripartito solo per i servizi cloud che superano rigorosi criteri di sicurezza, tra cui una buona igiene dei dati e la compatibilità con Duo Security. L'adozione di questa soluzione aiuterà se una buona parte del traffico esterno osservato in precedenza è destinato a questi servizi cloud sicuri. Questo fa emergere la necessità di analizzare le applicazioni Web a cui accedono gli utenti VPN.

La maggior parte dei firewall di nuova generazione come Cisco Firepower Threat Defense (FTD) contiene informazioni sull'applicazione associate all'evento nei registri. l'analisi e la pulizia dei dati di registro con **le librerie csv** python e le funzionalità di manipolazione dei dati pandas possono fornire un dataset simile a quello riportato sopra con un'aggiunta delle applicazioni a cui si accede mappate su di esso.

```
#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

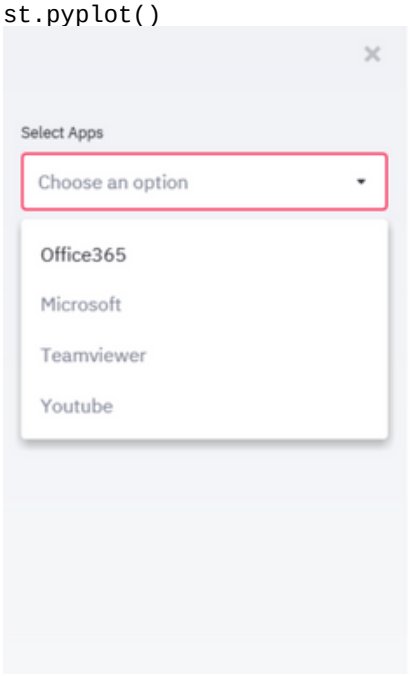
Una volta ottenuto un frame di dati come sopra, è possibile categorizzare il traffico esterno totale in base all'applicazione attraverso i panda.

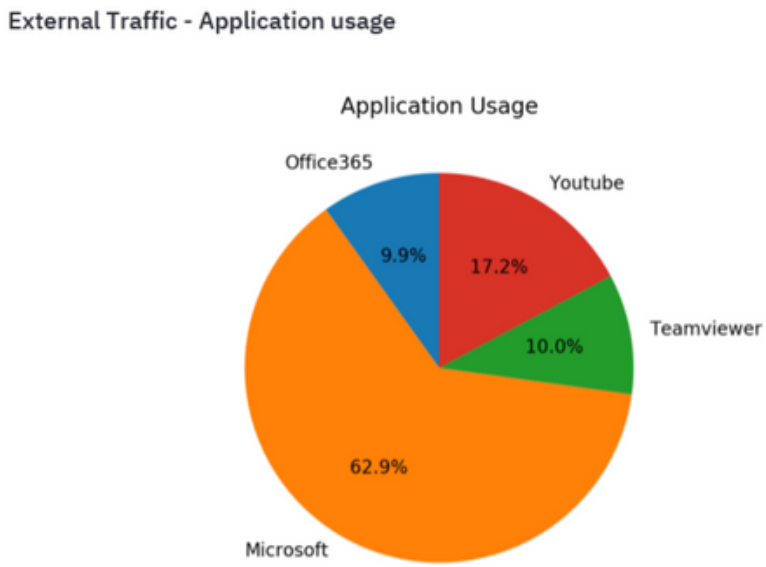
```
df2 = df.groupby('Application')

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

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

L'utilizzo di Streamlit ottiene nuovamente una rappresentazione grafica della quota di ciascuna applicazione nel traffico totale. Consente la flessibilità di modificare la finestra temporale per i dati da includere e di filtrare le applicazioni sull'interfaccia utente stessa senza la necessità di modifiche nel codice, il che rende l'analisi facile e precisa.

```
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()
```





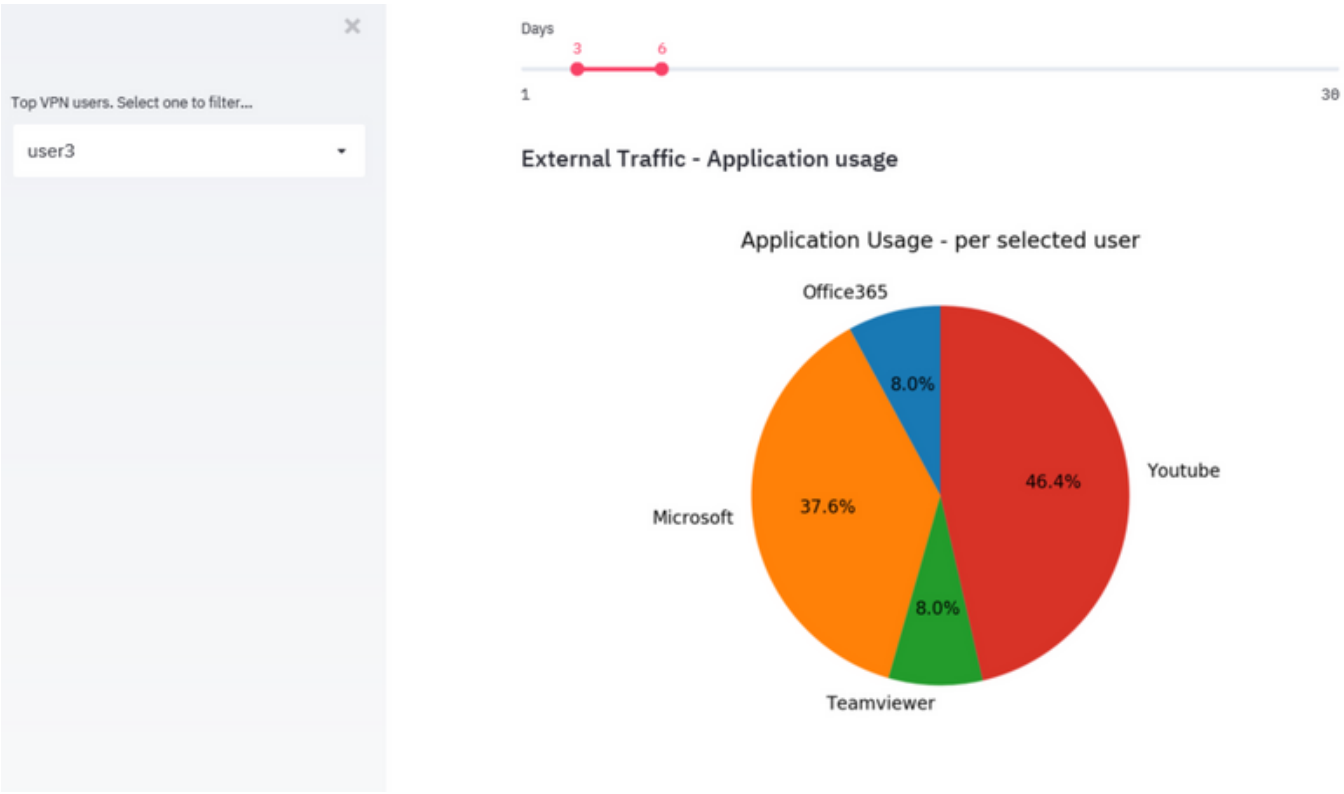
In questo modo è possibile semplificare il processo di identificazione delle principali applicazioni Web utilizzate dagli utenti VPN per un determinato periodo di tempo e se tali applicazioni sono destinate o meno alla protezione dei servizi cloud.

Se le applicazioni più voluminose sono destinate a identificare servizi cloud sicuri, possono essere

utilizzate con un tunnel suddiviso, riducendo così il carico su un concentratore VPN. Tuttavia, se le applicazioni principali sono destinate a servizi meno sicuri o che possono rappresentare un rischio, è più sicuro passarle attraverso il tunnel VPN. Il motivo è che altri dispositivi di sicurezza di rete possono elaborare il traffico prima di consentirne il passaggio. È quindi possibile utilizzare i criteri di accesso sui firewall per limitare l'accesso alle reti esterne.

Identity Individual Non-Compliant VPN Users

In alcuni casi, l'aumento potrebbe essere associato solo ad alcuni utenti che non rispettano determinate politiche. I moduli e i set di dati utilizzati in precedenza possono essere utilizzati nuovamente per identificare i principali utenti VPN e le applicazioni Web a cui accedono. Ciò può aiutare nell'isolamento di tali utenti e osservare il loro effetto sul carico del dispositivo.



Negli scenari in cui nessuno dei metodi rientra, gli amministratori devono esaminare le soluzioni di sicurezza degli endpoint, ad esempio la soluzione AMP for Endpoints e la soluzione Cisco Umbrella, per proteggere gli endpoint nelle reti non protette.