通过数据分析优化远程访问VPN设置的编程方法

目录

简介

本文档介绍如何通过目前可用的一些编程模块和开源工具监控和优化远程访问VPN设置。如今，即使是最小的网络，也会生成大量数据，这些数据可用于获取有用的信息。对收集的数据应用分析有助于以事实为后盾，做出更快、更明智的业务决策。

先决条件

要求

Cisco 建议您了解以下主题：

- 远程访问 VPN
- 基本Python编程概念

使用的组件

本文档不限于特定Cisco ASA或FTD软件和硬件版本。

> **注意：**Pandas、Streamlit、CSV和Matplotlib是使用的几个Python库。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解任何命令和python脚本的潜在影响。

问题

随着许多公司将大多数员工都采用“在家工作”模式，依靠VPN来工作的用户数量显着增加。这导致VPN集中器的负载突然大幅增加，使管理员重新思考和重新规划其VPN设置。做出明智决策以减少ASA集中器的负载需要在一段时间内从设备收集大量信息并评估此信息，这是一项复杂的任务，如

果手动执行，将需要相当长的时间。

解决方案

借助目前可用于网络可编程性和数据分析的多种Python模块和开源工具，编程在数据收集和分析、VPN设置的规划和优化方面非常有用。

基于VPN用户和并发连接的初始分析

要开始分析，请获取连接的用户数、建立的并发连接数及其对带宽的影响。以下Cisco ASA命令输出将提供以下详细信息：

- show vpn-sessiondb anyconnect
- show conn

Python模块Netmiko可用于ssh到设备、运行命令并解析输出。

```
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)
```

定期收集列表中的VPN用户计数和连接计数（每2小时可以是一个良好的开始），并获取一天的最大每日计数。

```
#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])
```

熊猫是一种高效的数据分析和操作库，所有解析的数据都可以存储为一个系列或一个数据帧，使熊猫对数据的操作变得容易。

```
import pandas as pd

df = pd.DataFrame(df1, columns=['Max Daily VPN Users Count','Max Daily Concurrent
```

```
Connections'],index=<date range>)
```

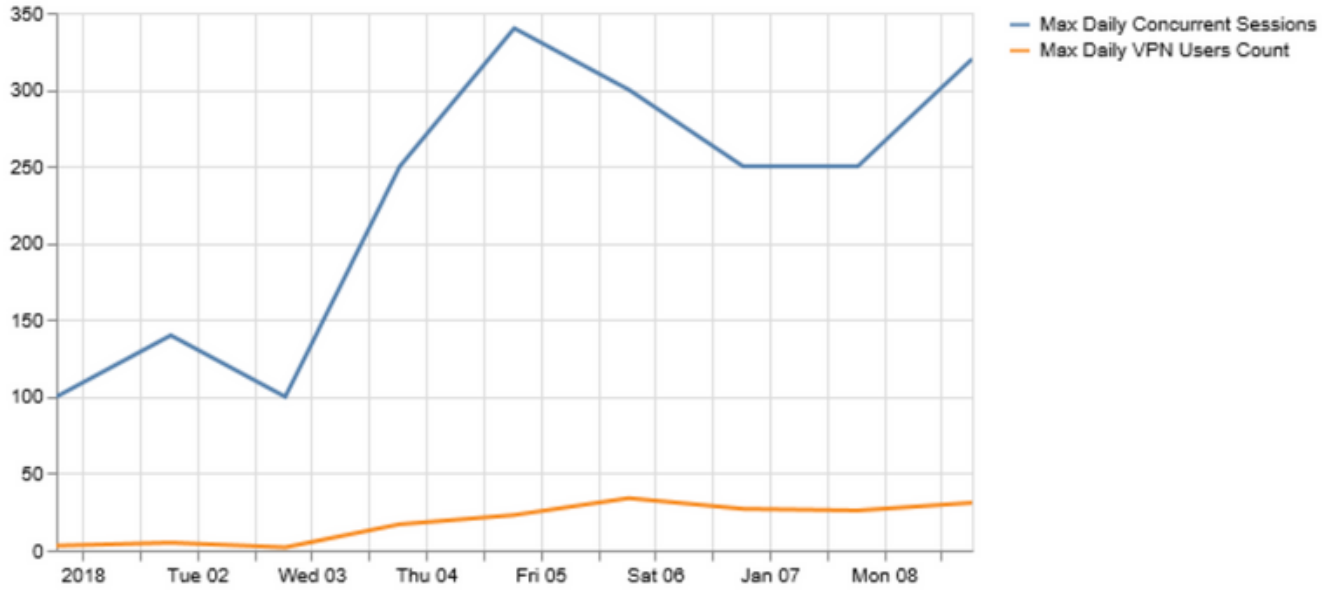
Daily Max VPN user Count - Max concurrent count

	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

分析每**日最大VPN用**户数**和最大并**发连接数，以帮助确定优化VPN设置的需要。

使用pandas和matplotlib库中的**绘图**功能，如图所示。

```
df.plot()

matplotlib.pyplot.show()
```



如果VPN用户数或并发连接数接近VPN前端的容量，则可能会导致以下问题：

- 正在丢弃新的VPN用户。
- 通过ASA的新数据连接被丢弃，用户无法访问资源。
- 高CPU和/或内存。

一段时间内的趋势有助于确定设备是否达到其阈值。

确定流向内部网络或外部网络的流量趋势

Cisco ASA上的Show conn输出可提供其他详细信息，例如流量是流向内部网络还是外部网络，以

及每个流通过防火墙的数据量（以字节为单位）。

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

使用Netaddr python模块，可以轻松地将获取的连接表拆分为流向外部网络和流向内部网络的流。

```
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]
```

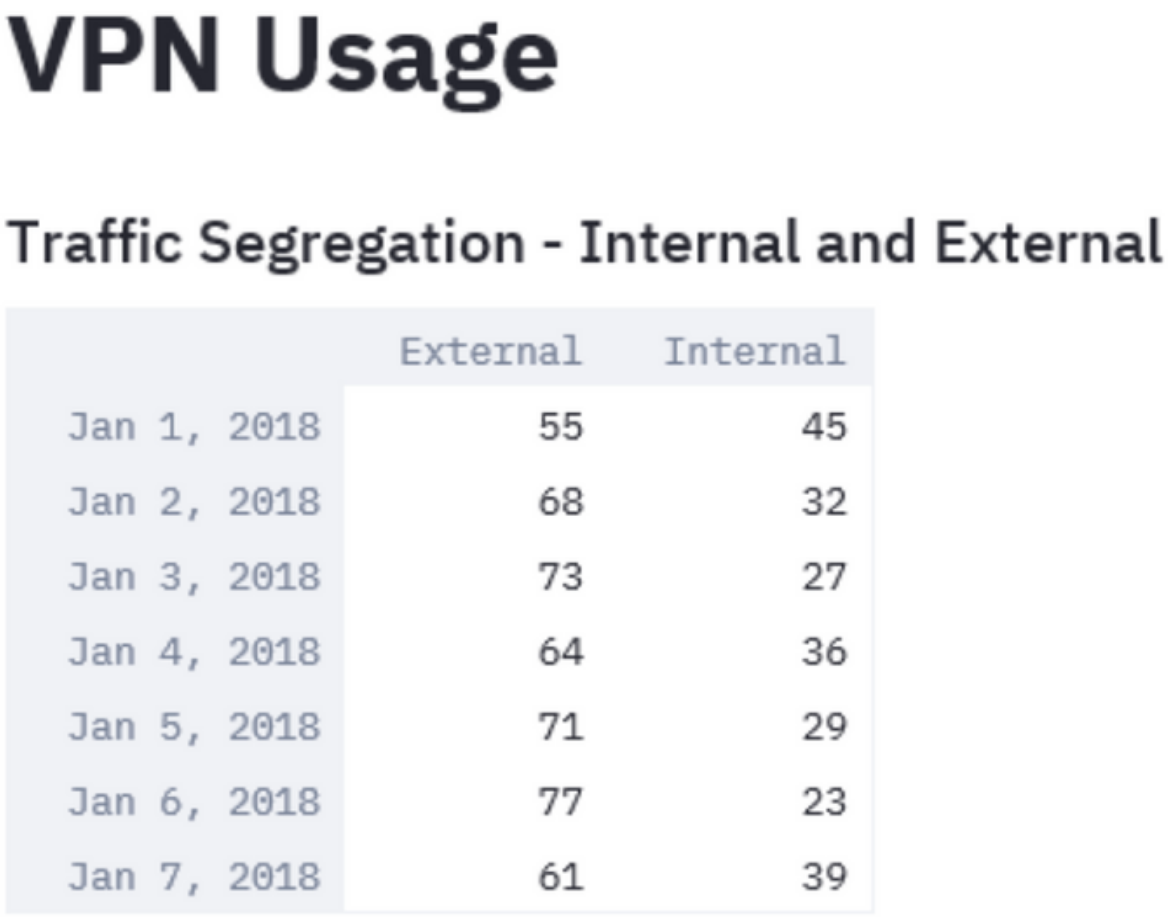
这是内部流量的映像。

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

这是外部流量的映像。

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

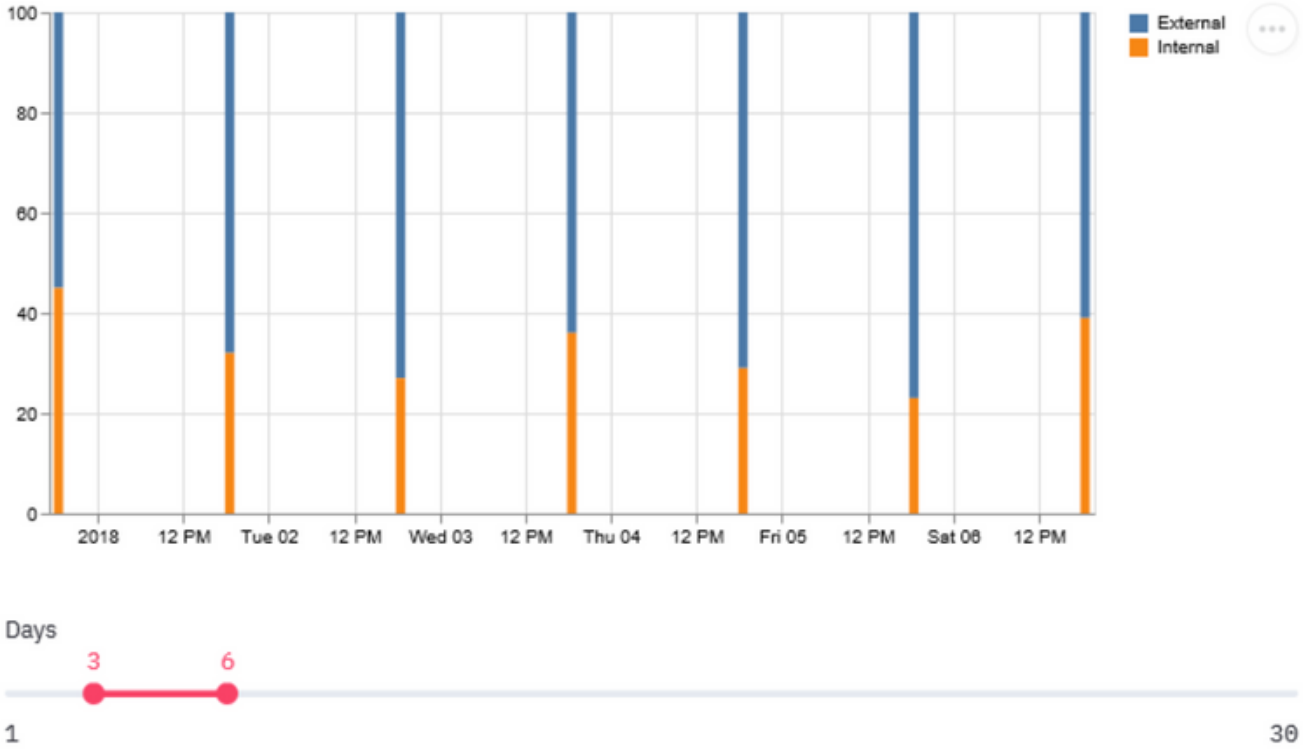
因此，可以深入了解VPN流量中发往内部网络的百分比，以及其中有多少流向互联网。在一段时间内收集此信息并分析其趋势有助于确定VPN流量主要是外部流量还是内部流量。



	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

Streamlit等模**块使**不仅能够将表格数据转换为图形表示，而且可以实时对其应用修改，以帮助分析。它可以修改所收集数据的时间窗口，或向所监控的参数添加额外数据。

```
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)
```

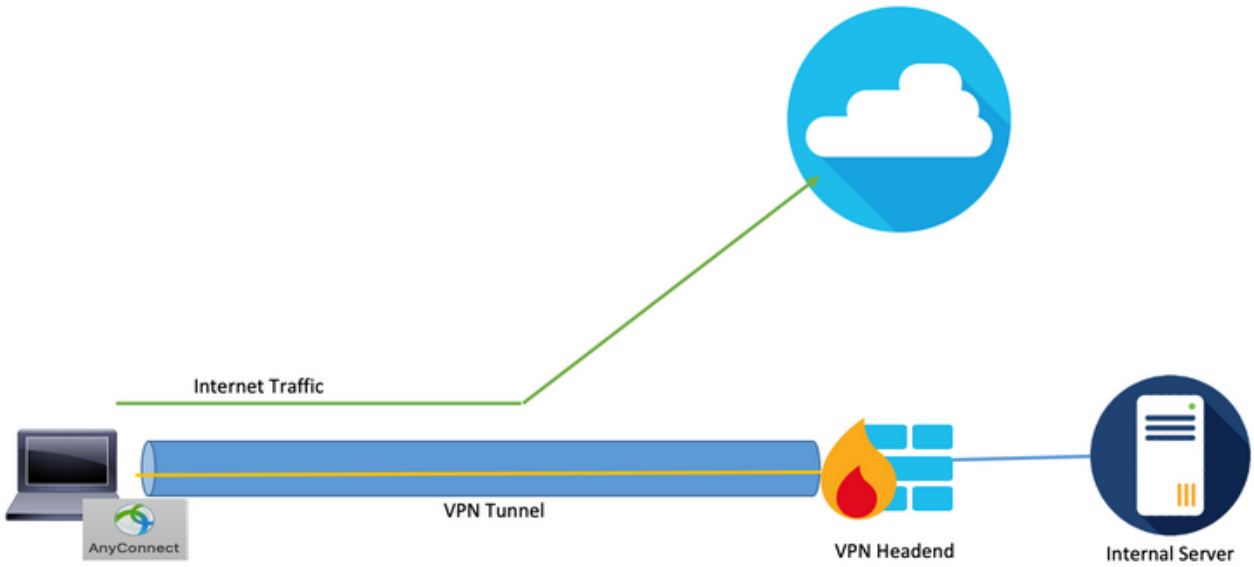


倾向于更高内部流量的趋势可能意味着大多数VPN用户访问内部资源。因此，为了满足这一需求，增加负载，必须计划升级到更大的机箱或使用VPN负载均衡等概念共享负载。

在某些情况下，VPN容量可能仍低于阈值，但VPN用户数量的增加会耗尽当前配置的VPN池。在这种情况下，请增加VPN IP池。

但是，如果趋势显示大部分VPN流量是外部流量，则可以使用分割隧道。

利用分割隧道功能



此功能仅通过隧道从用户系统转发特定流量集，其余流量将转发到默认网关，无需VPN加密。因此，为了减少VPN集中器上的负载，只有发往内部网络的流量才能通过隧道路由，而互联网流量也可以通过用户的本地ISP转发。这是一种有效的方法，被广泛采用，但是它带有一些风险。

员工通过未受保护的网络访问某些社交媒体站点以快速中断，可能会使其笔记本电脑感染由于缺乏工作场所中设置的深度防御安全层而扩散到整个公司的恶意软件。一旦受感染，受感染设备可能会成为从互联网到受信任网段的一个枢纽点，绕过边界防御。

在使用此功能的同时降低风险的一种方法是仅对符合严格安全标准的云服务使用分割隧道，包括良好的数据卫生和与双核安全的兼容性。如果之前观察到的大量外部流量流向这些安全云服务，则采用此方法将有所帮助。这就需要分析VPN用户正在访问的Web应用。

大多数下一代防火墙(如思科Firepower威胁防御(FTD))都包含与日志中的事件相关的应用信息。使用python csv库和pandas数据处理功能**解析和**清除此日志数据，可以提供与上述类似的数据集，并添加被访问的应用映射到该数据集。

```
#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

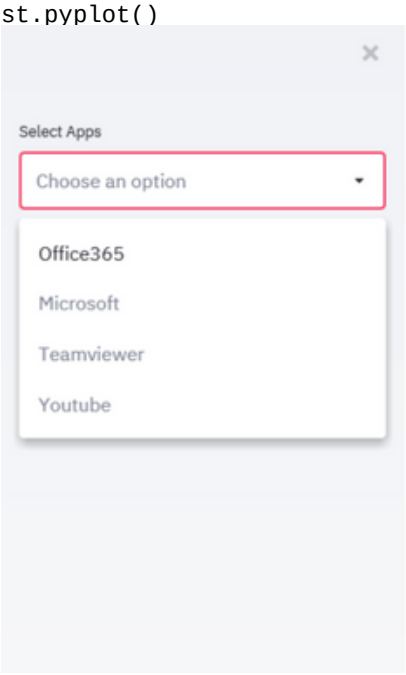
获得上述数据帧后，您可以通过panda根据应用对总外部流量进行分类。

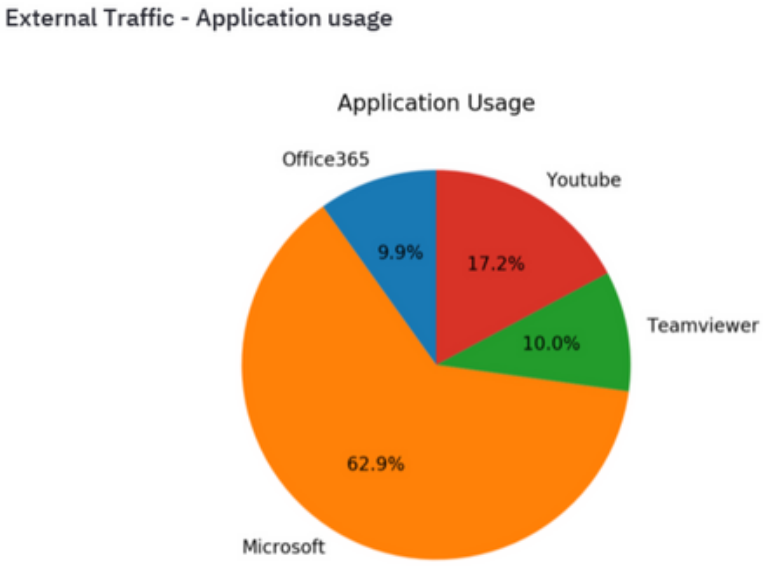
```
df2 = df.groupby('Application')

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

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

使用Streamlit再次获得每个应用在总流量中份额的图形表示。它允许灵活地更改要包含的数据的时间窗口以及过滤用户界面本身上的应用程序，而无需对代码进行任何更改，这使分析变得简单而准确。

```
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()
```



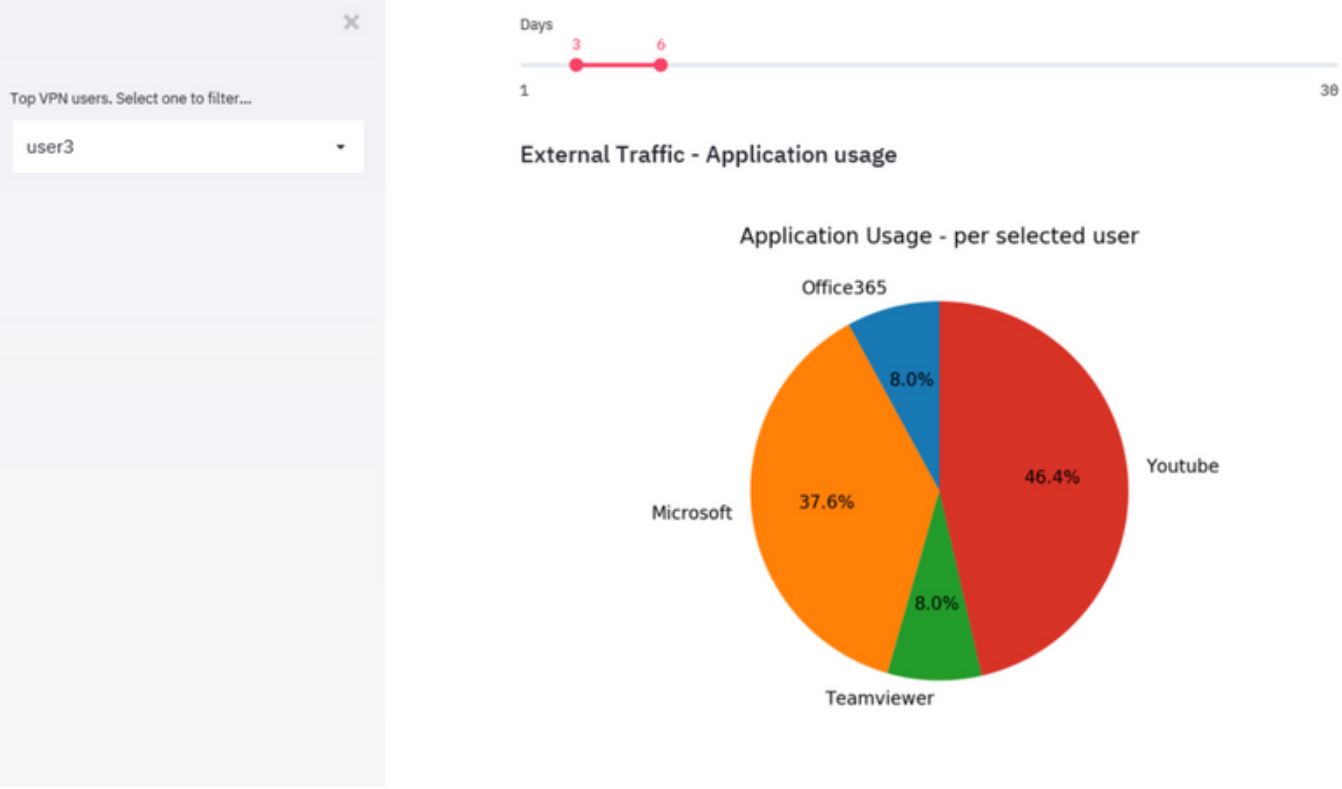


这可以简化识别VPN用户在一段时间内使用的顶级Web应用的过程，以及这些应用是否保护云服务的过程。

如果大多数大量应用的目的是识别安全云服务，则它们可以与拆分隧道一起使用，从而减少VPN集中器上的负载。但是，如果最主要的应用是针对安全性较低或可能带来风险的服务，则通过VPN隧道传递这些应用更安全。原因是其他网络安全设备可以在允许此类流量通过之前处理流量。然后，您可以利用防火墙上的访问策略来限制对外部网络的访问。

身份个人不合规VPN用户

在某些情况下，激增可能只与不符合某些策略的少数用户相关。上述模块和数据集可再次用于识别排名靠前的VPN用户及其访问的Web应用。这有助于隔离此类用户并观察其对设备负载的影响。



在所有方法都不适合的情况下，管理员应查看终端安全解决方案，如面向终端的AMP解决方案和思科Umbrella解决方案，以保护未受保护网络中的终端。