

# 使用SCAPY將Nexus 9000配置為流量生成器

## 目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[安裝](#)

[建立封包](#)

[傳送流量](#)

[驗證](#)

## 簡介

本文檔介紹Scapy，這是一款用於N9K交換機輕鬆建立和運算元據包的Python資料包操作工具。

## 必要條件

將Scapy下載到交換器bootflash。

要下載Scapy，請使用GitHub [GitHub-SCAPY中的連結](#)

## 需求

思科建議您瞭解以下主題：

- Nexus 9000/3000交換器。

## 採用元件

- N9K-C9396PX

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

## 安裝

將Scapy代碼下載並解壓到您的交換機啟動快閃記憶體；FTP、SFTP或SCP可用。

啟用此功能，本例中為SCP。

```
switch(config)# feature scp-server
switch(config)# sh feature | i scp
scpServer          1          enabled
```

將檔案從筆記型電腦複製到交換器。

```
scp scapy-vxlan-master.zip admin@10.88.164.13:/
```

映像進入開機快閃記憶體後，需要將其解壓縮。它需要啟用bash功能並從bash解壓縮。

```
switch(config)# feature bash
switch(config)# run bash
bash-4.3$ sudo su -
root@switch#cd /bootflash
root@switch#unzip scapy-vxlan-master.zip
```

解壓縮後，可使用dir命令在引導快閃記憶體中查詢檔案，即壓縮和未壓縮檔案。

```
switch# dir bootflash: | i i scapy
 4096    Jul 09 18:00:01 2019  scapy-vxlan-master/
1134096  Jul 19 23:35:26 2023  scapy-vxlan-master.zip
```

現在Scapy已可用。

請注意，您需要以超級使用者許可權呼叫程式，並且還需要導航到Scapy目錄。

```
switch(config)# run bash
Enter configuration commands, one per line. End with CNTL/Z.
```

```
bash-4.2$ sudo su -
root@switch#cd /
root@switch#cd bootflash/scapy-vxlan-master          <<< Move to the scapy folder scapy-vxlan-master
root@switch#python                                  <<< Run python once located inside the folder
Python 2.7.2 (default, Mar  9 2015, 15:52:40)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from scapy.all import *                          <<< Import libraries from scapy
>>>
```

## 建立封包

以下範例顯示如何建立基本IP封包，以說明使用Scapy產生流量的程式。

Create l2 source and destination mac addresses.

```
>>> l2=Ether()
>>> l2.src='00:aa:12:34:12:34'
>>> l2.dst='00:ff:aa:bb:cc:11'
```

Create l3 source and destination IP addresses.

```
>>> l3=IP()
>>> l3.src='10.1.1.1'
>>> l3.dst='10.2.2.2'
```

另一種功能是從以前捕獲的pcap檔案傳送資料包。這是使用rdpcap指令實現的。

該命令的輸出是一個包含在pcap檔案中捕獲的所有資料包的Python清單。在本例中， traffic.pcap包含10個資料包，這些資料包將被分配給pkts建立的清單。

```
>>> pkts = rdpcap('bootflash/traffic.pcap')
>>> len(pkts)
10
>>> type(pkts)
<class 'scapy.plist.PacketList'>
```

---

註:pcap檔案需要儲存在交換機的啟動快閃記憶體中。

---

## 傳送流量

建立資料包後，我們使用命令sendp開始通過指定介面傳送資料包。

```
>>> packet = 12/13.          << packet now have the values for source and destination declared
>>> sendp(packet, iface='Eth1-1'). << Sending the packet through interface eth1/1
.
Sent 1 packets.
```

然後，您可以重複檢視封包清單，透過您指定的介面傳送流量。

```
>>> while True:
...   for i in range(len(pkts)):   <<< It goes through the list pkts with 10 packets and send 1 by
...     sendp(pkts[i], iface='Eth1-1')
...
.
Sent 1 packets.
.
Sent 1 packets.
```

---

注意：只能使用交換機埠模式訪問。否則會顯示錯誤。

---

錯誤示例：

```
>>> sendp(12/13, iface='Eth1-6')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "scapy/sendrecv.py", line 335, in sendp
socket = socket or conf.L2socket(iface=iface, *args, **kargs)
File "scapy/arch/linux.py", line 477, in __init__
set_promisc(self.ins, self.iface)
File "scapy/arch/linux.py", line 165, in set_promisc
mreq = struct.pack("IHH8s", get_if_index(iff), PACKET_MR_PROMISC, 0, b"")
File "scapy/arch/linux.py", line 380, in get_if_index
return int(struct.unpack("I", get_if(iff, SIOCGIFINDEX)[16:20])[0])
File "scapy/arch/common.py", line 59, in get_if
ifreq = ioctl(sck, cmd, struct.pack("16s16x", iff.encode("utf8")))
IOError: [Errno 19] No such device
```

確保介面可用，運行ifconfig命令，必須在該命令中列出介面。

```
bash-4.3$ ifconfig | grep Eth
Eth1-1 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:88
Eth1-2 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:89
Eth1-5 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:8c
Eth1-6 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:8d
```

Eth1-8 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:8f  
Eth1-11 Link encap:Ethernet HWaddr 00:a2:ee:74:4b:c1  
...

## 驗證

您可以使用命令檢查任何給定資料包。

```
>>> pkts[5].show()
###[ Ethernet ]###
  dst      = 01:00:0c:cc:cc:cd
  src=58:97:bd:00:a4:f2
  type     = 0x8100
###[ 802.1Q ]###
  prio     = 6
  id       = 0
  vlan     = 104
  type     = 0x32
###[ LLC ]###
  dsap     = 0xaa
  ssap     = 0xaa
  ctrl     = 3
###[ SNAP ]###
  OUI      = 0xc
  code     = 0x10b
###[ Spanning Tree Protocol ]###
  proto    = 0
  version  = 2
  bpdu type = 2
  bpdu flags = 60
  rootid   = 32872
  rootmac  = 58:97:bd:00:a4:f1
  pathcost = 0
  bridgeid = 32872
  bridgemac = 58:97:bd:00:a4:f1
  portid   = 32769
  age      = 0.0
  maxage   = 20.0
  hellotime = 2.0
  fwdelay  = 15.0
###[ Raw ]###
  load     = '\x00\x00\x00\x00\x02\x00h'
```

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。