

# Configure o Nexus 9000 como um gerador de tráfego com SCAPY

## Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Instalação](#)

[Criar um pacote](#)

[Enviar tráfego](#)

[Verificar](#)

## Introdução

Este documento descreve o Scapy, uma ferramenta de manipulação de pacotes Python para que switches N9K criem e manipulem pacotes com facilidade.

## Pré-requisitos

Faça o download do Scapy para o bootflash do switch.

Para baixar o Scapy, use o link de GitHub [GitHub-SCAPY](#)

## Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Switch Nexus 9000/3000.

## Componentes Utilizados

- N9K-C9396PX

As informações neste documento foram criadas a partir de dispositivos em um ambiente de

laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Instalação

Baixe e extraia o código Scapy para o flash de inicialização do switch; FTP, SFTP ou SCP estão disponíveis.

Habilite o recurso, neste caso, SCP.

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

Copie o arquivo do laptop para o switch.

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

Quando a imagem estiver na flash de inicialização, ela precisará ser descompactada. É necessário habilitar o recurso bash e descompactá-lo do bash.

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

Uma vez descompactados, os arquivos podem ser localizados com o comando dir no flash de inicialização, o compactado e descompactado.

```
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
```

Agora o Scapy está disponível.

Observe que você precisa chamar o programa com privilégios root e também precisa navegar para o diretório Scapy.

```
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
>>>
```

## Criar um pacote

Este é um exemplo de como criar um pacote IP básico para ilustrar o procedimento para gerar tráfego usando Scapy.

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'
```

Outro recurso é enviar um pacote de um arquivo pcap capturado anteriormente. Isso é feito com o comando rdpcap.

A saída desse comando é uma lista Python que contém todos os pacotes capturados em seu arquivo pcap. Neste exemplo, traffic.pcap contém 10 pacotes e esses pacotes estão sendo atribuídos à lista criada como pacotes.

```
>>> pkts = rdpcap('bootflash/traffic.pcap')
>>> len(pkts)
```

```
10
>>> type(pkts)
<class 'scapy.plist.PacketList'>
```

---

Observação: o arquivo pcap precisa ser armazenado na flash de inicialização do switch.

---

## Enviar tráfego

Depois que o pacote é criado, usamos o comando `sendp` para começar a enviar nosso pacote pela interface especificada.

```
>>> 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.
```

Você pode então percorrer a lista de pacotes para enviar o tráfego pela interface especificada.

```
>>> 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.
```

---

Observação: somente o modo de acesso das portas do switch está disponível para ser usado. Caso contrário, ele exibirá um erro.

---

Exemplo do erro:

```
>>> 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
```

Certifique-se de que a interface seja utilizável, execute o comando `ifconfig`, a interface deve ser listada nela.

```
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
...
```

## Verificar

Você pode usar o comando para verificar qualquer pacote fornecido.

```
>>> 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
  bpdutype = 2
  bpduflags = 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
fwddelay = 15.0
```

```
###[ Raw ]###
```

```
load = '\x00\x00\x00\x00\x02\x00h'
```

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.