

Nexus 9000 configureren als verkeersgenerator met SCAPY

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Installatie](#)

[Een pakket maken](#)

[Verkeer verzenden](#)

[Verifiëren](#)

Inleiding

Dit document beschrijft Scapy, een Python-pakketmanipulatiETOOL voor N9K-switches om eenvoudig pakketten te maken en te manipuleren.

Voorwaarden

Download Scapy naar de switch bootflash.

Als u Scapy wilt downloaden, gebruikt u de link van GitHub [GitHub-SCAPY](#)

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Nexus 9000/3000 Switch

Gebruikte componenten

- N9K-C9396PX switch

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 live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Installatie

Download en haal de Scapy code naar uw switch boot flash; FTP, SFTP, of SCP zijn beschikbaar.

Schakel de optie, in dit geval SCP, in.

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

Kopieer het bestand van de laptop naar de switch.

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

Zodra het beeld in de laarsflits is, moet het worden gedeprimeerd. Het moet mogelijk maken functie bash en unzip het uit 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
```

Zodra gedeprimeerd, kunnen de bestanden worden gevonden met de **dir** opdracht op de boot flitser, de gecprimeerde en niet gecprimeerde.

```
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
```

Nu is Scapy beschikbaar.

Bericht dat u het programma met wortelvoorrechten moet roepen en u moet ook aan de folder van Scapy navigeren.

```
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
>>>
```

Een pakket maken

Dit is een voorbeeld van hoe u een standaard IP-pakket maakt om de procedure te illustreren om verkeer te genereren met 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'
```

Een andere mogelijkheid is om een pakket te verzenden vanuit een pcap-bestand dat eerder is opgenomen. Dit wordt bereikt met de opdracht **rdpcap**.

De uitvoer van die opdracht is een Python-lijst met alle pakketten die in uw pcap-bestand zijn opgenomen. In dit voorbeeld bevat **traffic.pcap** 10 pakketten en die pakketten worden toegewezen aan de lijst die als pakketten wordt gemaakt.

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

Opmerking: het pcap-bestand moet worden opgeslagen in de opstartflitsers van de switch.

Verkeer verzenden

Zodra het pakket is gemaakt, gebruiken we het opdrachtverzend om te beginnen met het verzenden van ons pakket via de opgegeven interface.

```
>>> packet = l2/l3.          << packet now have the values for source and destination declared o
>>> sendp(packet, iface='Eth1-1').  << Sending the packet through interface eth1/1
.
Sent 1 packets.
```

U kunt dan door de lijst van pakketten herhalen om het verkeer over de interface te verzenden u specificeert.

```
>>> 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.
```

Opmerking: alleen toegang tot de modus switch-poorten is beschikbaar voor gebruik. Anders geeft het een fout weer.

Voorbeeld van de fout:

```
>>> sendp(l2/l3, 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
```

Zorg ervoor dat de interface bruikbaar is, voer de **ifconfig**-opdracht uit, de interface moet daar worden vermeld.

```
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

...

Verifiëren

U kunt de opdracht gebruiken om een bepaald pakket te controleren.

```
>>> 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'
```

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.