

Configurazione di Nexus 9000 come Traffic Generator con SCAPY

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Installazione](#)

[Crea pacchetto](#)

[Traffico di invio](#)

[Verifica](#)

Introduzione

Questo documento descrive Scapy, uno strumento di manipolazione dei pacchetti Python per gli switch N9K, per creare e manipolare i pacchetti con facilità.

Prerequisiti

Scaricare Scapy sul file bootflash dello switch.

Per scaricare Scapy, usa il link da GitHub [GitHub-SCAPY](#)

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Switch Nexus 9000/3000.

Componenti usati

- N9K-C9396PX

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, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Installazione

Scaricare ed estrarre il codice Scapy nella memoria flash di avvio dello switch; sono disponibili FTP, SFTP o SCP.

Abilitare la funzionalità, in questo caso SCP.

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

Copiare il file sullo switch dal laptop.

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

Una volta che l'immagine è nel flash di avvio, deve essere decompressa. Deve abilitare la funzionalità bash e decomprimerla da 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
```

Una volta decompressi, i file possono essere individuati con il comando **dir** sul flash di avvio, i file compressi e non compressi.

```
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
```

Ora Scapy è disponibile.

Notare che è necessario chiamare il programma con i privilegi root e anche passare alla directory 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
>>>
```

Crea pacchetto

Questo è un esempio di come creare un pacchetto IP di base per illustrare la procedura per generare traffico con 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'
```

Un'altra funzionalità è l'invio di un pacchetto da un file pcap precedentemente acquisito. A tale scopo, è possibile utilizzare il comando **rdpcap**.

L'output di questo comando è un elenco Python che contiene tutti i pacchetti acquisiti nel file pcap. In questo esempio, **traffic.pcap** contiene 10 pacchetti e questi pacchetti vengono assegnati all'elenco creato come pacchetti.

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

Nota: il file pcap deve essere memorizzato nella memoria flash di avvio dello switch.

Traffico di invio

Una volta creato il pacchetto, usare il comando **sendp** per iniziare a inviare il pacchetto sull'interfaccia specificata.

```
>>> 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.
```

È quindi possibile scorrere l'elenco dei pacchetti per inviare il traffico sull'interfaccia specificata.

```
>>> 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.
```

Nota: è possibile utilizzare solo la modalità di accesso alle porte dello switch. In caso contrario, viene visualizzato un errore.

Esempio dell'errore:

```
>>> 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
```

Accertarsi che l'interfaccia sia utilizzabile, eseguire il comando **ifconfig**; l'interfaccia deve essere elencata in questa sezione.

```
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

...

Verifica

È possibile utilizzare il comando per controllare qualsiasi pacchetto.

```
>>> 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'
```

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).