SCAPY

Đỗ Thanh Nghị
Mai Ngọc Hải, Nguyễn Lữ Khánh Duy
dtnghi@cit.ctu.edu.vn

12-2019

Giới thiệu Scapy

- Thư viện Scapy cài đặt trong Python
- Philippe Biondi đề xuất
- Hỗ trợ khai thác gói tin: gửi, nghe lén, phân tích và giả mạo các gói tin trong mạng
- Hỗ trợ xây dựng các công cụ để thăm dò, quét hoặc nghiên cứu an ninh mạng

Giới thiệu Scapy

- Install Scapy:
  pip install scapy

- Chế độ giao tiếp
  >>>

- Lập trình
  from scapy.all import *

Lệnh **ls()**

- Hiển thị tất cả gói tin được hỗ trợ trong Scapy

```
>>> ls()
AH                  : AH
AKMSuite            : AKM suite
ARP                 : ARP
ASN1P_INTEGER       : None
ASN1P_OID           : None
ASN1P_PRIVSEQ       : None
...
```

Lệnh **ls(<packet>)**

- Hiển thị kiểu & giá trị mặc định của gói tin

```
>>> ls(ARP)
hwtype    : XShortField              = (1)
ptype     : XShortEnumField          = (2048)
hwlen     : FieldLenField            = (None)
plen      : FieldLenField            = (None)
op        : ShortEnumField           = (1)
hwsrc     : MultipleTypeField        = (None)
...
```

Lệnh **conf**

- Hiển thị - Thay đổi thông tin cấu hình trong Scapy

```
>>> conf
USBsocket  = None
auto_crop_tables = True
auto_fragment = 1
...
>>> conf.iface
'wlp3s0'
>>> conf.iface = 'eth0'
>>> conf.iface
'eth0'
```

Lệnh **help(<func>)**

- Hiển thị trợ giúp cho hàm

```
>>> help(rdpcap)

Help on function rdpcap in module scapy.utils:

rdpcap(filename, count=-1)
    Read a pcap or pcapng file and return a packet list
    count: read only <count> packets
(END)
```

Gói tin **IP()**


```
>>> packet = IP()
>>> packet.show()
###[ IP ]###
    version= 4
    ihl= None
    tos= 0x0
    len= None
    id= 1
    flags=
    frag= 0
    ttl= 64
    proto= hopopt
    chksum= None
    src= 127.0.0.1
    dst= 127.0.0.1
    \options\
```

Gói tin **IP()**

9

```
>>> packet = IP()
>>> packet.src = "192.168.20.40"
>>> packet.dst = "170.30.35.70"
      hoặc packet.dst = "example.com"
      hoặc packet.dst = "192.168.2.0/24"
>>> packet.ttl = 10
>>> packet
<IP  ttl=10 src=192.168.20.40 dst=170.30.35.70 |>
>>> del(packet.ttl)
>>> packet
<IP  src=192.168.20.40 dst=170.30.35.70 |>
>>> packet.ttl
```

Gói tin **IP()**


```
>>> packet = IP(src="192.168.20.40",
dst="172.30.35.70", ttl=10)
>>> packet
<IP  ttl=10 src=192.168.20.40 dst=172.30.35.70 |>
>>> packet = IP(src=str(RandIP()))
>>> packet.src
'249.118.159.175'
```

Gói tin **ICMP()**


```
>>> packet = ICMP()
>>> packet.show()
###[ ICMP ]###
     type= echo-request (hoặc echo-reply)
     code= 0
     chksum= None
     id= 0x0
     seq= 0x0
```

Gói tin **TCP()**


```
>>> packet = TCP()
>>> packet.show()
###[ TCP ]###
     sport= ftp_data
     dport= http
     seq= 0
     ack= 0
     dataofs= None
     reserved= 0
     flags= S
     window= 8192
     chksum= None
     urgptr= 0
     options= []
```

Gói tin **UDP()**


```
>>> packet = UDP()
>>> packet.show()
###[ UDP ]###
     sport= domain
     dport= domain
     len= None
     chksum= None
```

Gói tin **Ether()**


```
>>> packet = Ether()
>>> packet.show()
WARNING: Mac address to reach destination not found. Using broadcast.
###[ Ethernet ]###
     dst= ff:ff:ff:ff:ff:ff
     src= c4:8e:8f:46:95:77
     type= 0x9000
```

Gói tin **Ether()**


```
>>> packet = Ether(src=RandMAC())
>>> packet
<Ether  src=b2:3d:93:59:92:f1 |>
```

Ghép gói tin

- Sử dụng dấu / để ghép các gói tin lại với nhau

```
>>> IP()
<IP  |>

>>> IP()/TCP()
<IP  frag=0 proto=tcp |<TCP  |>>

>>> Ether()/IP()/TCP()
<Ether  type=0x800 |<IP  frag=0 proto=tcp |<TCP  |
>>>
```

Gửi gói tin

- Gửi gói tin ở tầng 3
  send(pkt, inter=0, loop=0, count=1, iface=N)

- Gửi gói tin ở tầng 2
  sendp(pkt, inter=0, loop=0, count=1, iface=N)

Gửi gói tin


```
>>> send(IP(dst="192.168.2.104")/UDP(dport=53))
.
Sent 1 packets.

>>>
sendp(Ether()/IP(dst="192.168.2.104")/UDP(dport=53))
.
Sent 1 packets.
```

Gửi gói tin

- Gửi gói tin với độ dài nhất định

```
>>> payload = "XXXXXXXXXX"
>>> packet = IP()/ICMP()/payload

>>> packet = IP()/ICMP()/("X"*60000)
```

Đọc file .pcap


```
sudo tcpdump -w packets.pcap
>>> a = rdpcap("/home/dtnghi/packets.pcap")
>>> a
<packets.pcap: TCP:21 UDP:11 ICMP:0 Other:0>
>>> a.show()
0000 Ether / IP / UDP / DNS Qry "_googlecast._tcp.local."
0001 Ether / IP / UDP 192.168.2.100:54055 > 172.217.27.14:443 / Raw
0002 Ether / IP / UDP / DNS Qry "_googlecast._tcp.local."
0003 Ether / IP / UDP 172.217.27.14:443 > 192.168.2.100:54055 / Raw
...
```

Nghe lén


```
>>> sniff(iface = "wlp3s0", prn = lambda x: x.summary(), filter = "udp", store = 0)
Ether / IP / UDP 192.168.2.104:34612 > 239.255.255.250:1900 / Raw
Ether / IP / UDP 192.168.2.104:34612 > 239.255.255.250:1900 / Raw
Ether / IP / UDP 192.168.2.104:34612 > 239.255.255.250:1900 / Raw
Ether / IP / UDP 192.168.2.109:54904 > 239.255.255.250:1900 / Raw
Ether / IP / UDP 192.168.2.109:54904 > 239.255.255.250:1900 / Raw
Ether / IP / UDP 192.168.2.109:54904 > 239.255.255.250:1900 / Raw
...
<Sniffed: TCP:0 UDP:0 ICMP:0 Other:0>
...
```

Chương trình gửi gói tin TCP


```
from scapy.all import *

target_ip = "192.168.1.x"
target_port = 80
payload = "toto"

source_ip = RandIP()      #random Ip address

tcp_packet = IP(src=source_ip, dst=target_ip) /
TCP(dport=target_port) / payload

send(tcp_packet)
```

Chương trình ping


```
from scapy.all import *

target_ip = "192.168.1.x"

response = sr1(IP(dst=target_ip)/ICMP(), timeout=2)

if response:
  print("Response received from " + target_ip)
else:
  print("No response received from " + target_ip)
```

Chương trình nghe lén


```
from scapy.all import *

def packet_callback(packet):
    print(packet.show())

sniff(prn=packet_callback, store=0)
```

Chương trình nghe lén dịch vụ web


```
from scapy.all import *
from scapy.layers.http import *

def packet_callback(packet):
    if packet.haslayer(HTTP):
        print(packet.show())

sniff(prn=packet_callback, filter="tcp port 80", store=0)
```