1. Starting Scapy as Root

```
sudo su
```

```
scapy
```

2. Basic Packet Sniffing

Start sniffing:

```
sniff()
```

Generate traffic in a new terminal:

```
ping google.com
```

Stop sniffing: Press Ctrl + C on both ping and Scapy terminals.

Store captured packets:

```
paro = _
```

```
paro.summary()
```

3. Interface-Specific Sniffing

```
sniff(iface="br-internal")
```

Generate traffic:

```
ping 10.6.6.1/24
```

Visit internal page: 10.6.6.23

Stop sniffing: Press Ctrl + C.

Store results:

```
paro2 = _
```

```
paro2.summary()
```

4. ICMP-Filtered Sniffing

Capture only ICMP packets (five packets total):

```
sniff(iface="br-internal", filter="icmp", count=5)
```

Trigger ICMP:

```
ping 10.6.6.23
```

Stop terminals: Press Ctrl + C on ping and Scapy.

Store captured ICMP packets:

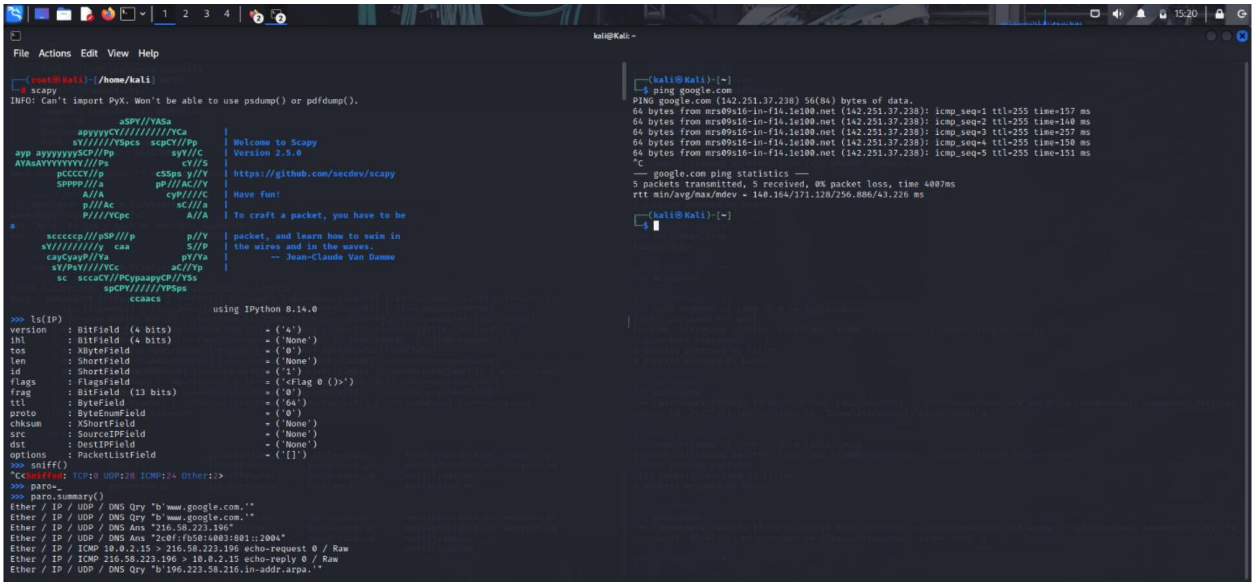
paro3 = _

paro3.summary()

Inspect a specific packet:

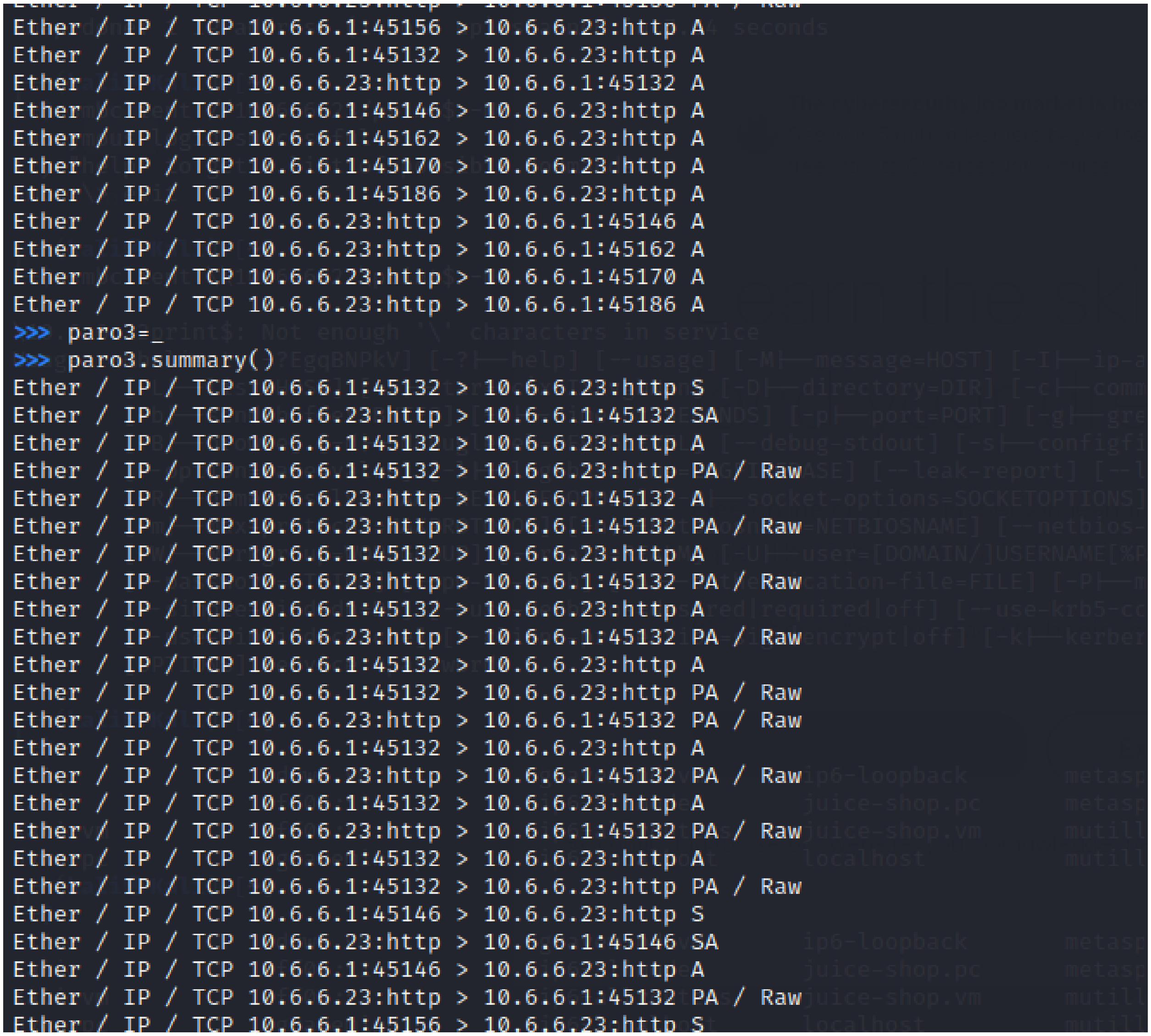
paro3[3]

Demo

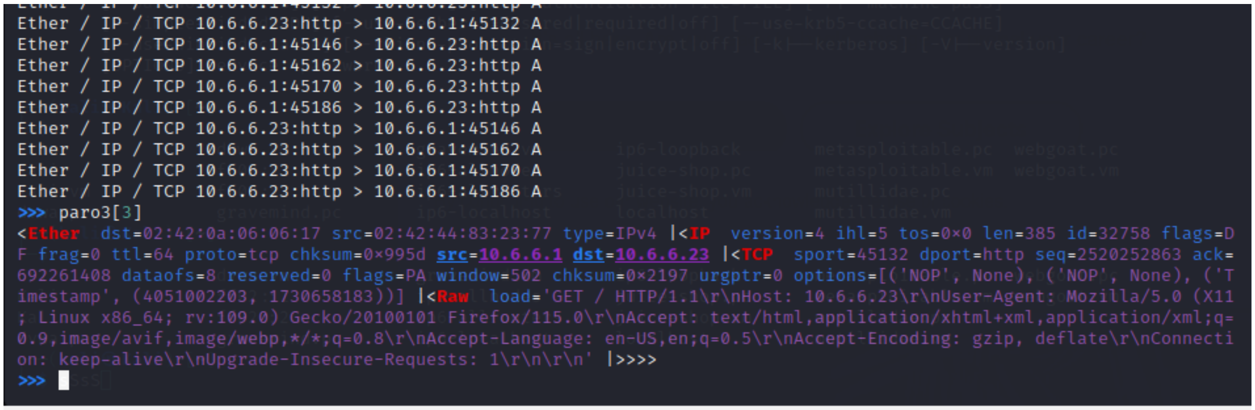


```
┌──(root㉿Kali)-[/home/kali]
└─# scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.5.0
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | To craft a packet, you have to be
a
       scccccp///pSP///p          p//Y   | packet, and learn how to swim in
      sY/////////y  caa           S//P   | the wires and in the waves.
       cayCyayP//Ya              pY/Ya   |        -- Jean-Claude Van Damme
        sY/PsY////YCc          aC//Yp    |
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs
                                       using IPython 8.14.0
>>> ls(IP)
version    : BitField  (4 bits)                    = ('4')
ihl        : BitField  (4 bits)                    = ('None')
tos        : XByteField                            = ('0')
len        : ShortField                            = ('None')
id         : ShortField                            = ('1')
flags      : FlagsField                            = ('<Flag 0 ()>')
frag       : BitField  (13 bits)                   = ('0')
ttl        : ByteField                             = ('64')
proto      : ByteEnumField                         = ('0')
chksum     : XShortField                           = ('None')
src        : SourceIPField                         = ('None')
dst        : DestIPField                           = ('None')
options    : PacketListField                       = ('[]')
>>> sniff()
^C<Sniffed: TCP:0 UDP:28 ICMP:24 Other:2>
>>> paro=_
>>> paro.summary()
Ether / IP / UDP / DNS Qry "b'www.google.com.'"
Ether / IP / UDP / DNS Qry "b'www.google.com.'"
Ether / IP / UDP / DNS Ans "216.58.223.196"
Ether / IP / UDP / DNS Ans "2c0f:fb50:4003:801::2004"
Ether / IP / ICMP 10.0.2.15 > 216.58.223.196 echo-request 0 / Raw
Ether / IP / ICMP 216.58.223.196 > 10.0.2.15 echo-reply 0 / Raw
Ether / IP / UDP / DNS Qry "b'196.223.58.216.in-addr.arpa.'"
```

```
┌──(kali㉿Kali)-[~]
└─$ ping google.com
PING google.com (142.251.37.238) 56(84) bytes of data.
64 bytes from mrs09s16-in-f14.1e100.net (142.251.37.238): icmp_seq=1 ttl=255 time=157 ms
64 bytes from mrs09s16-in-f14.1e100.net (142.251.37.238): icmp_seq=2 ttl=255 time=140 ms
64 bytes from mrs09s16-in-f14.1e100.net (142.251.37.238): icmp_seq=3 ttl=255 time=257 ms
64 bytes from mrs09s16-in-f14.1e100.net (142.251.37.238): icmp_seq=4 ttl=255 time=150 ms
64 bytes from mrs09s16-in-f14.1e100.net (142.251.37.238): icmp_seq=5 ttl=255 time=151 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 140.164/171.128/256.886/43.226 ms

┌──(kali㉿Kali)-[~]
└─$ 
```



```
Ether / IP / TCP 10.6.6.1:45156 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 A
Ether / IP / TCP 10.6.6.1:45146 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45162 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45170 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45186 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45146 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45162 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45170 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45186 A
>>> paro3=_
>>> paro3.summary()
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http S
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 SA
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http PA / Raw
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http PA / Raw
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45132 > 10.6.6.23:http PA / Raw
Ether / IP / TCP 10.6.6.1:45146 > 10.6.6.23:http S
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45146 SA
Ether / IP / TCP 10.6.6.1:45146 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 PA / Raw
Ether / IP / TCP 10.6.6.1:45156 > 10.6.6.23:http S
```



```
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45132 A
Ether / IP / TCP 10.6.6.1:45146 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45162 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45170 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.1:45186 > 10.6.6.23:http A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45146 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45162 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45170 A
Ether / IP / TCP 10.6.6.23:http > 10.6.6.1:45186 A
>>> paro3[3]
<Ether  dst=02:42:0a:06:06:17 src=02:42:44:83:23:77 type=IPv4 |<IP  version=4 ihl=5 tos=0x0 len=385 id=32758 flags=D
F frag=0 ttl=64 proto=tcp chksum=0x995d src=10.6.6.1 dst=10.6.6.23 |<TCP  sport=45132 dport=http seq=2520252863 ack=
692261408 dataofs=8 reserved=0 flags=PA window=502 chksum=0x2197 urgptr=0 options=[('NOP', None), ('NOP', None), ('T
imestamp', (4051002203, 1730658183))] |<Raw  load='GET / HTTP/1.1\r\nHost: 10.6.6.23\r\nUser-Agent: Mozilla/5.0 (X11
; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=
0.9,image/avif,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nConnecti
on: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\n\r\n' |>>>>
>>> 
```