TECHNION
Azrieli Continuing Education and
External Studies Division

Module 5.8: Intro to Scapy

What Is Scapy?

“Scapy is a Python program that enables the user to send, sniff and dissect and forge network packets. This capability enables the construction of tools that can probe, scan or attack networks.”

So What Can It Do?

- Create packets or sets of packets
- Manipulate the packets
- Send them on the wire
- Sniff packets from the wire
- Preform full protocol lifecycles

What Can It Be Used For?

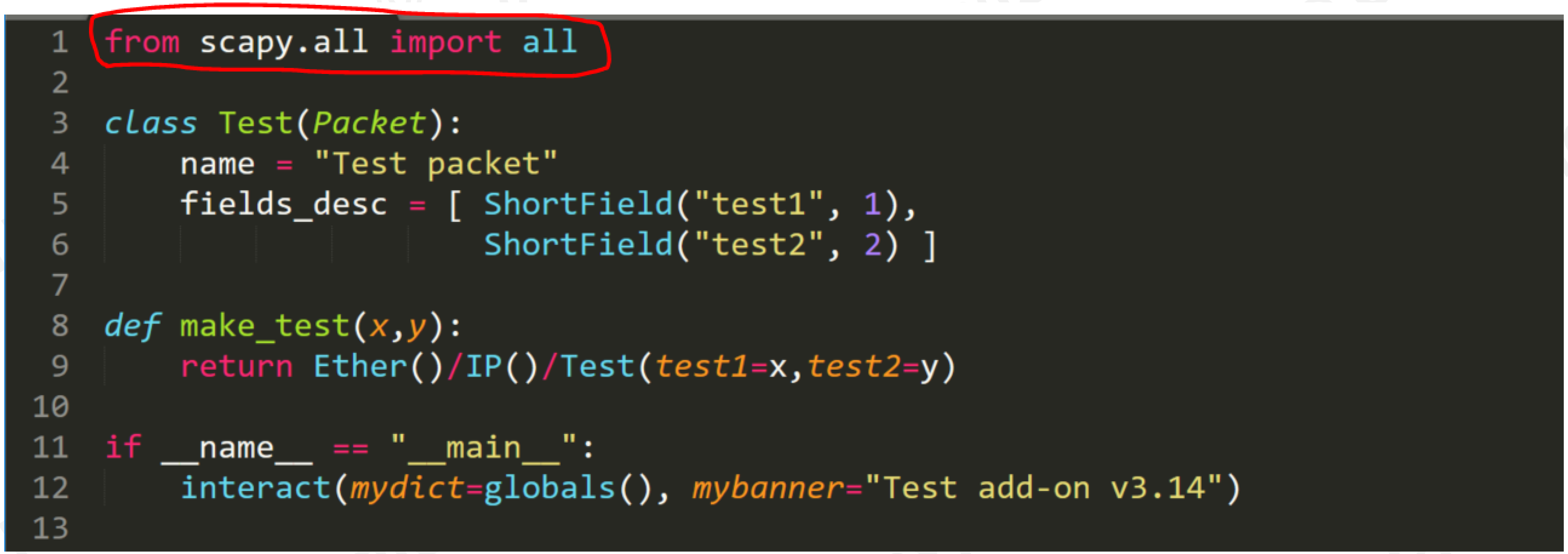
- Testing and research
- Scanning networks and protocols
- Attacks (DoS, ARP poisoning)
- Sniffing

Supported Protocols

- Ethernet
- 802.1Q
- 802.11
- 802.3
- LLC
- EAPOL
- EAP
- BOOTP

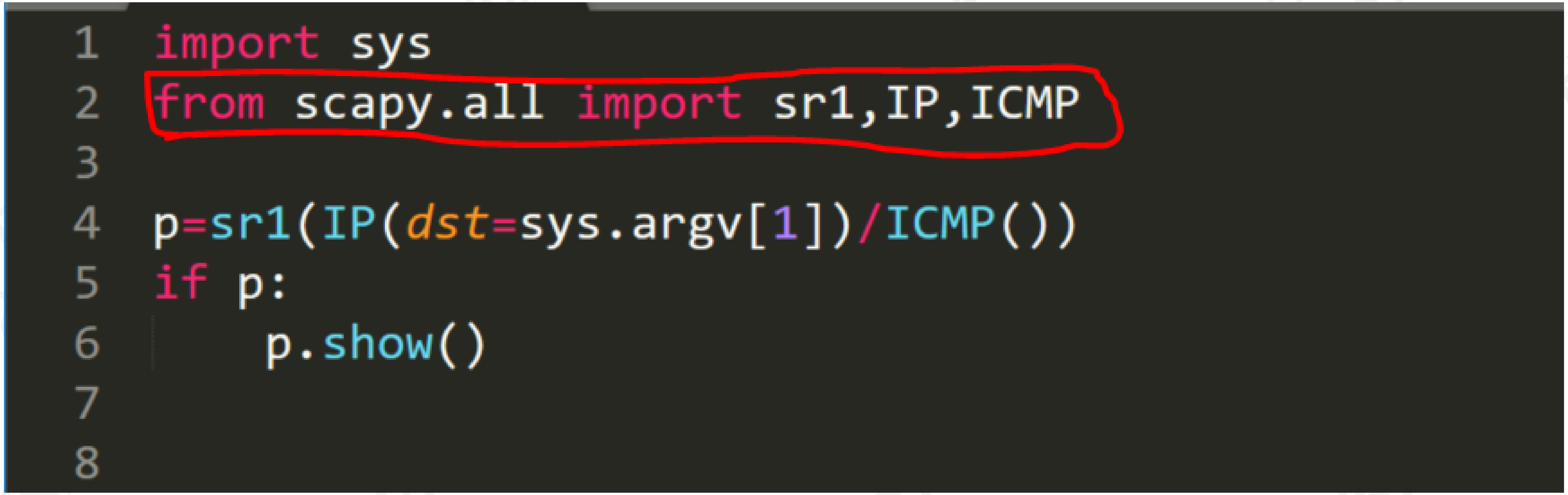
- PPP
- IP
- TCP
- ICMP
- ARP
- STP
- UDP
- DNS

Importing Scapy



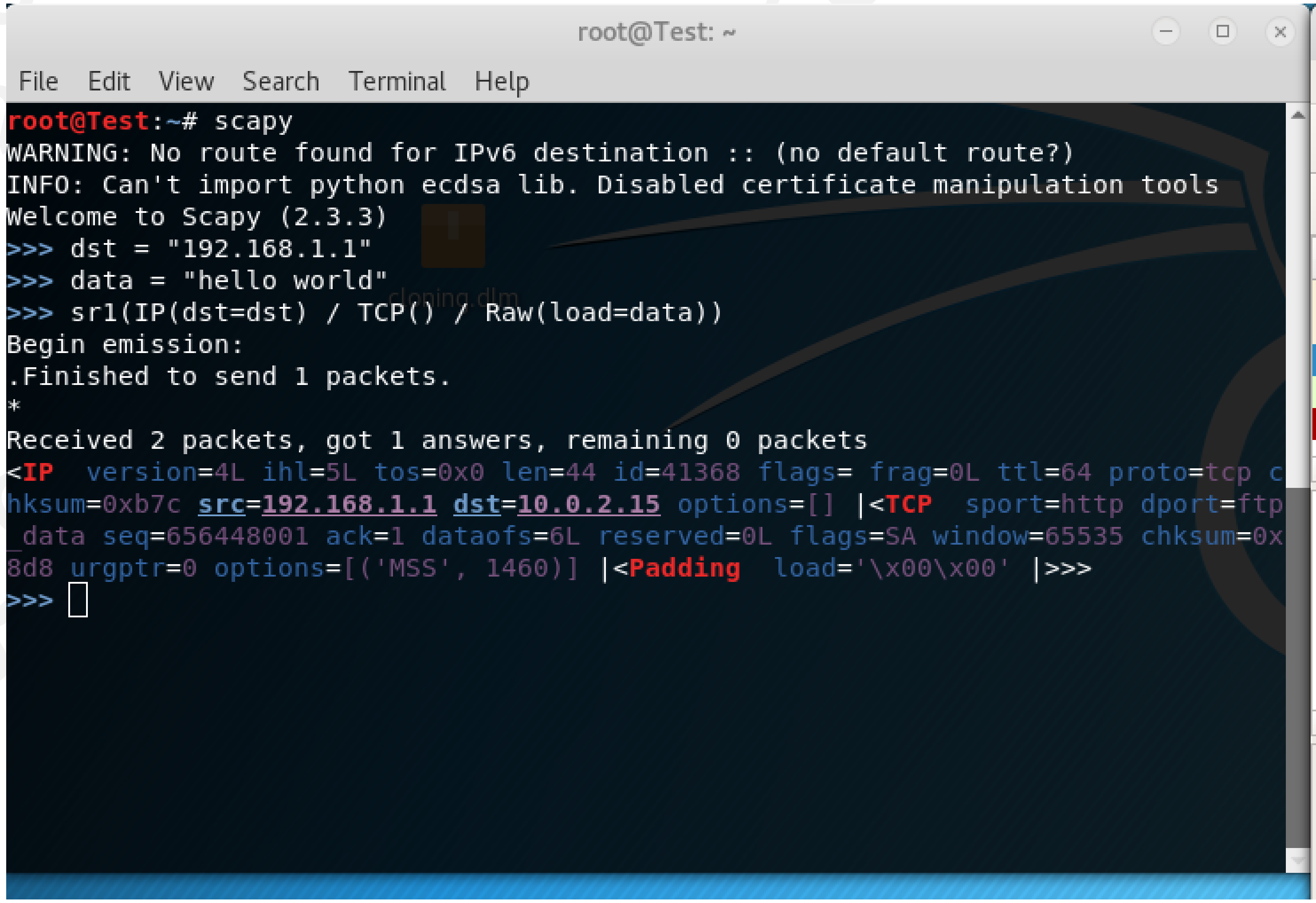
```python
from scapy.all import all

class Test(Packet):
    name = "Test packet"
    fields_desc = [ ShortField("test1", 1),
                    ShortField("test2", 2) ]

def make_test(x,y):
    return Ether()/IP()/Test(test1=x,test2=y)

if __name__ == "__main__":
    interact(mydict=globals(), mybanner="Test add-on v3.14")

```

Importing Specific Modules

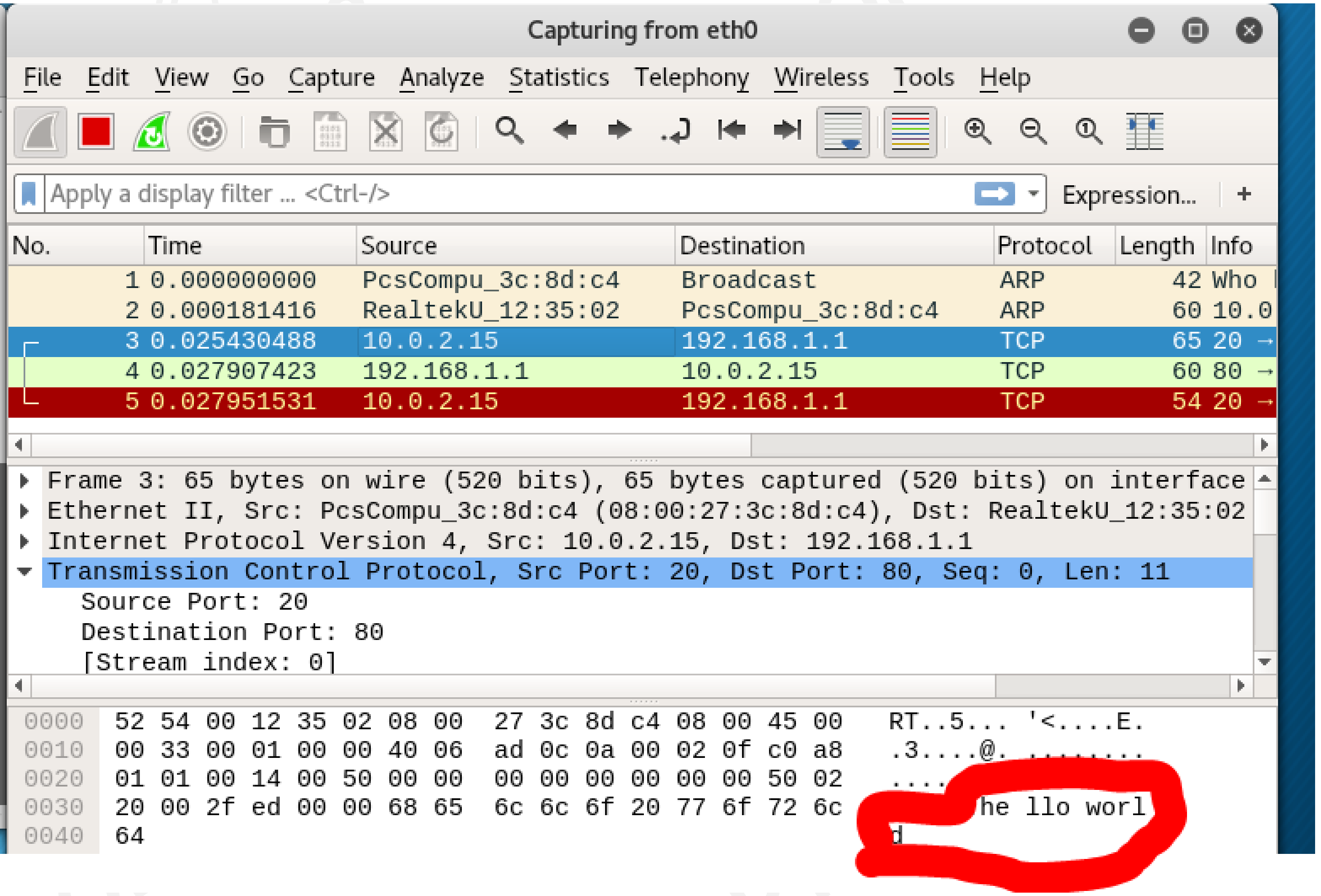


```python
import sys
from scapy.all import sr1,IP,ICMP

p=sr1(IP(dst=sys.argv[1])/ICMP())
if p:
    p.show()


```

Simple TCP Request



```
root@Test:~# scapy
WARNING: No route found for IPv6 destination :: (no default route?)
INFO: Can't import python ecdsa lib. Disabled certificate manipulation tools
Welcome to Scapy (2.3.3)
>>> dst = "192.168.1.1"
>>> data = "hello world"
>>> sr1(IP(dst=dst) / TCP() / Raw(load=data))
Begin emission:
.Finished to send 1 packets.
*
Received 2 packets, got 1 answers, remaining 0 packets
<IP  version=4L ihl=5L tos=0x0 len=44 id=41368 flags= frag=0L ttl=64 proto=tcp c
hksum=0xb7c src=192.168.1.1 dst=10.0.2.15 options=[] |<TCP  sport=http dport=ftp
_data seq=656448001 ack=1 dataofs=6L reserved=0L flags=SA window=65535 chksum=0x
8d8 urgptr=0 options=[('MSS', 1460)] |<Padding  load='\x00\x00' |>>>
>>> 
```



Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

```
No.  Time         Source              Destination         Protocol  Length Info
   1 0.000000000  PcsCompu_3c:8d:c4   Broadcast           ARP           42 Who
   2 0.000181416  RealtekU_12:35:02   PcsCompu_3c:8d:c4   ARP           60 10.0
   3 0.025430488  10.0.2.15           192.168.1.1         TCP           65 20 →
   4 0.027907423  192.168.1.1         10.0.2.15           TCP           60 80 →
   5 0.027951531  10.0.2.15           192.168.1.1         TCP           54 20 →
```

```
▸ Frame 3: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface
▸ Ethernet II, Src: PcsCompu_3c:8d:c4 (08:00:27:3c:8d:c4), Dst: RealtekU_12:35:02
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.1.1
▾ Transmission Control Protocol, Src Port: 20, Dst Port: 80, Seq: 0, Len: 11
    Source Port: 20
    Destination Port: 80
    [Stream index: 0]
```

```
0000  52 54 00 12 35 02 08 00  27 3c 8d c4 08 00 45 00   RT..5... '<....E.
0010  00 33 00 01 00 00 40 06  ad 0c 0a 00 02 0f c0 a8   .3....@. ........
0020  01 01 00 14 00 50 00 00  00 00 00 00 00 00 50 02   ....
0030  20 00 2f ed 00 00 68 65  6c 6c 6f 20 77 6f 72 6c     he llo worl
0040  64                                                 d
```

Request Structure

```
packet = sr1(IP(dst=“192.168.1.1”)/TCP()/Raw(“Hello World”))
```

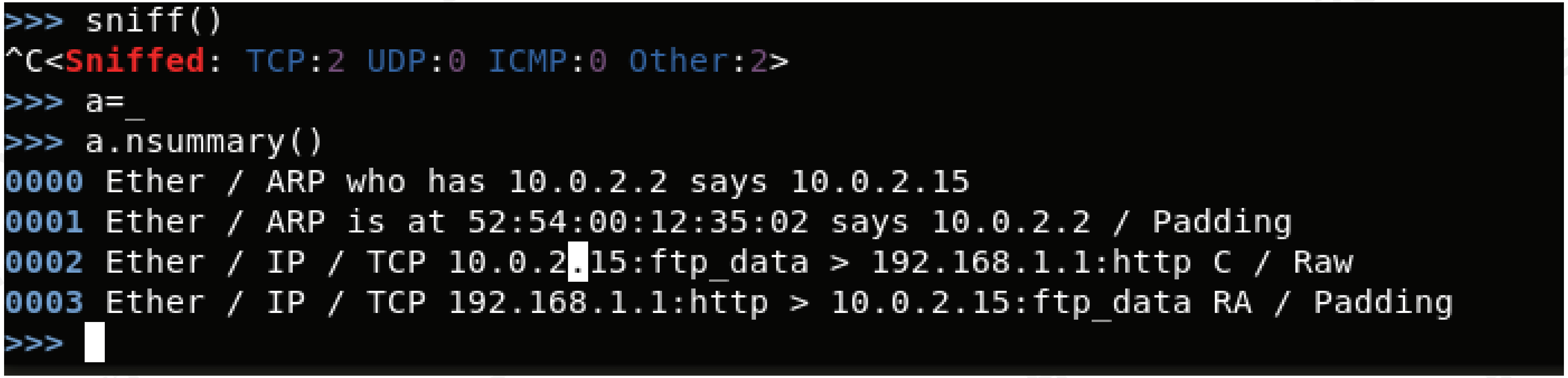
Let’s Add Flags

```
packet = sr1(IP(dst=“192.168.1.1”)/TCP(flags=“S”)/Raw(“Hello World”))
```

Types of Requests

- sr1 – Will send packets and receive only first the answer. (L3)
- sr – Will send packets and receive all answers. (L3)
- srp1 – Will send packets and receive only the first answer. (L2)
- srp - Will send packets and receive all answers. (L2)

Sniffing



```
>>> sniff()
^C<Sniffed: TCP:2 UDP:0 ICMP:0 Other:2>
>>> a=_
>>> a.nsummary()
0000 Ether / ARP who has 10.0.2.2 says 10.0.2.15
0001 Ether / ARP is at 52:54:00:12:35:02 says 10.0.2.2 / Padding
0002 Ether / IP / TCP 10.0.2.15:ftp_data > 192.168.1.1:http C / Raw
0003 Ether / IP / TCP 192.168.1.1:http > 10.0.2.15:ftp_data RA / Padding
>>> 
```

Sniffing Filters

```
sniff(filter=“tcp port 110”)
```

Questions?