# Ethical Hacking

## Lab. 1 - Packet Sniffing and Spoofing

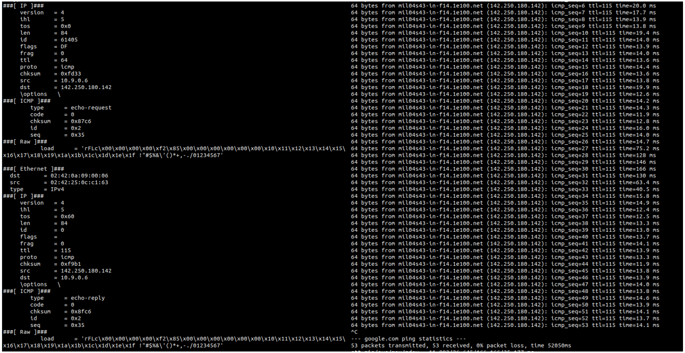
Cosuti Luca
De Faveri Francesco L.
Doria Samuele

Università degli Studi di Padova

# Task 1.1 - Sniffing Packets

User B send **ping google.com**, the attacker execute sniff.py with root privileges.

**Note:** By running the script using a non-root user the program throws a Operation not permitted exception.



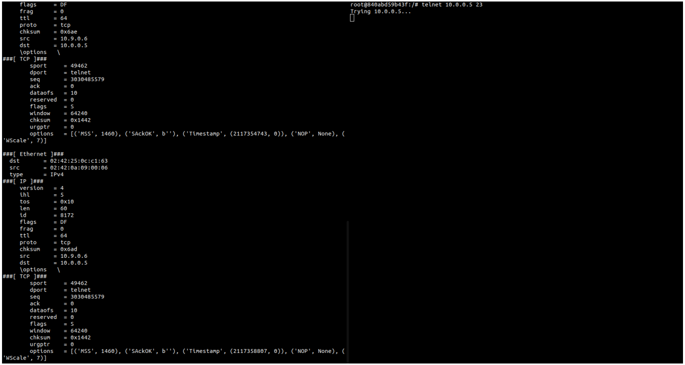
```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 61405
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xfd33
     src       = 10.9.0.6
     dst       = 142.250.180.142
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x87c6
        id        = 0x2
        seq       = 0x35
###[ Raw ]###
           load      = 'rFLc\x00\x00\x00\x00\xf2\x85\x00\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\
x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:06
  src       = 02:42:25:0c:c1:63
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x60
     len       = 84
     id        = 0
     flags     = 
     frag      = 0
     ttl       = 115
     proto     = icmp
     chksum    = 0xf9b1
     src       = 142.250.180.142
     dst       = 10.9.0.6
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x8fc6
        id        = 0x2
        seq       = 0x35
###[ Raw ]###
           load      = 'rFLc\x00\x00\x00\x00\xf2\x85\x00\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\
x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'
```

```
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=6 ttl=115 time=20.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=7 ttl=115 time=17.7 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=8 ttl=115 time=13.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=9 ttl=115 time=13.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=10 ttl=115 time=19.4 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=11 ttl=115 time=14.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=12 ttl=115 time=13.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=13 ttl=115 time=14.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=14 ttl=115 time=13.6 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=15 ttl=115 time=14.4 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=16 ttl=115 time=13.6 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=17 ttl=115 time=13.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=18 ttl=115 time=19.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=19 ttl=115 time=12.6 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=20 ttl=115 time=14.2 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=21 ttl=115 time=14.3 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=22 ttl=115 time=11.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=23 ttl=115 time=12.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=24 ttl=115 time=16.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=25 ttl=115 time=14.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=26 ttl=115 time=14.7 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=27 ttl=115 time=75.2 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=28 ttl=115 time=128 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=29 ttl=115 time=146 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=30 ttl=115 time=166 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=31 ttl=115 time=130 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=32 ttl=115 time=63.4 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=33 ttl=115 time=40.5 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=34 ttl=115 time=15.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=35 ttl=115 time=14.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=36 ttl=115 time=12.4 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=37 ttl=115 time=12.5 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=38 ttl=115 time=13.3 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=39 ttl=115 time=13.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=40 ttl=115 time=13.7 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=41 ttl=115 time=14.1 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=42 ttl=115 time=13.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=43 ttl=115 time=13.3 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=44 ttl=115 time=11.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=45 ttl=115 time=13.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=46 ttl=115 time=13.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=47 ttl=115 time=14.0 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=48 ttl=115 time=13.8 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=49 ttl=115 time=14.6 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=50 ttl=115 time=13.9 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=51 ttl=115 time=14.1 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=52 ttl=115 time=13.7 ms
64 bytes from mil04s43-in-f14.1e100.net (142.250.180.142): icmp_seq=53 ttl=115 time=14.1 ms
^C
--- google.com ping statistics ---
53 packets transmitted, 53 received, 0% packet loss, time 52050ms
```

Live demonstration for the three IP

# Task 1.2 - Spoofing ICMP Packets

User B send **telnet <IP User A> 23**, the attacker is able to catch the packet setting the filter “host <IP> and tcp port 23>”

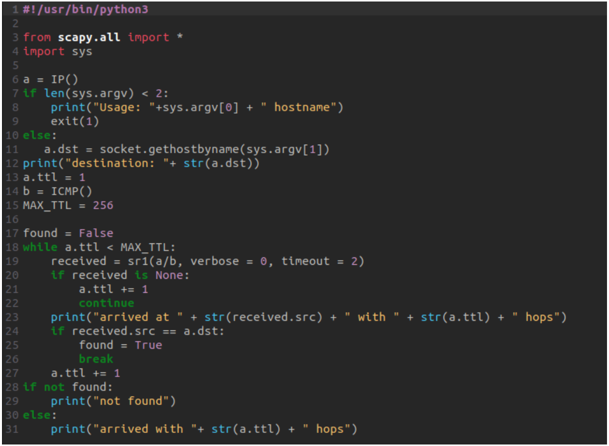


```
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x6ae
     src       = 10.9.0.6
     dst       = 10.0.0.5
     \options   \
###[ TCP ]###
        sport     = 49462
        dport     = telnet
        seq       = 3030485579
        ack       = 0
        dataofs   = 10
        reserved  = 0
        flags     = S
        window    = 64240
        chksum    = 0x1442
        urgptr    = 0
        options   = [('MSS', 1460), ('SAckOK', b''), ('Timestamp', (2117354743, 0)), ('NOP', None), (
'WScale', 7)]

###[ Ethernet ]###
  dst       = 02:42:25:0c:c1:63
  src       = 02:42:0a:09:00:06
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 60
     id        = 8172
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x6ad
     src       = 10.9.0.6
     dst       = 10.0.0.5
     \options   \
###[ TCP ]###
        sport     = 49462
        dport     = telnet
        seq       = 3030485579
        ack       = 0
        dataofs   = 10
        reserved  = 0
        flags     = S
        window    = 64240
        chksum    = 0x1442
        urgptr    = 0
        options   = [('MSS', 1460), ('SAckOK', b''), ('Timestamp', (2117358807, 0)), ('NOP', None), (
'WScale', 7)]
```

```
root@840abd59b43f:/# telnet 10.0.0.5 23
Trying 10.0.0.5...
```

# Task 1.3 - Traceroute

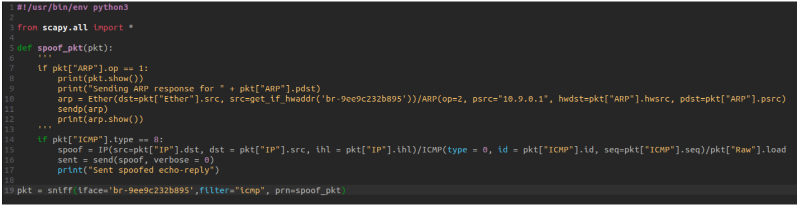
Snippet of the code.



```python
#!/usr/bin/python3

from scapy.all import *
import sys

a = IP()
if len(sys.argv) < 2:
    print("Usage: "+sys.argv[0] + " hostname")
    exit(1)
else:
   a.dst = socket.gethostbyname(sys.argv[1])
print("destination: "+ str(a.dst))
a.ttl = 1
b = ICMP()
MAX_TTL = 256

found = False
while a.ttl < MAX_TTL:
    received = sr1(a/b, verbose = 0, timeout = 2)
    if received is None:
        a.ttl += 1
        continue
    print("arrived at " + str(received.src) + " with " + str(a.ttl) + " hops")
    if received.src == a.dst:
        found = True
        break
    a.ttl += 1
if not found:
    print("not found")
else:
    print("arrived with "+ str(a.ttl) + " hops")
```

# Task 1.4 - Sniffing and-then Spoofing

Fix 10.9.0.99 with arp spoofing



```python
#!/usr/bin/env python3

from scapy.all import *

def spoof_pkt(pkt):
    '''
    if pkt["ARP"].op == 1:
        print(pkt.show())
        print("Sending ARP response for " + pkt["ARP"].pdst)
        arp = Ether(dst=pkt["Ether"].src, src=get_if_hwaddr('br-9ee9c232b895'))/ARP(op=2, psrc="10.9.0.1", hwdst=pkt["ARP"].hwsrc, pdst=pkt["ARP"].psrc)
        sendp(arp)
        print(arp.show())
    '''
    if pkt["ICMP"].type == 8:
        spoof = IP(src=pkt["IP"].dst, dst = pkt["IP"].src, ihl = pkt["IP"].ihl)/ICMP(type = 0, id = pkt["ICMP"].id, seq=pkt["ICMP"].seq)/pkt["Raw"].load
        sent = send(spoof, verbose = 0)
        print("Sent spoofed echo-reply")

pkt = sniff(iface='br-9ee9c232b895',filter="icmp", prn=spoof_pkt)
```

# Task 2.1A - Understanding Sniffer

*Q*1

*Q*2 We need to execute the program as root because we interact with the promiscuous mode and the network devices to listen to the traffic. Without it we get a "Segmentation fault (core dumped)".

*Q*3 Setting the mode to "not promiscuous" the program is still working and able to sniff the network because of our configuration which is not using promiscuous mode at all, instead is set to "host mode" on docker. As a matter of fact, the promiscuity in the attacker VM is set to 0.

# Task 2.1B - Writing Filters

Live demonstration.

# Task 2.1C - Sniffing Passwords

Live demonstration.

Figure: GRAZIE PER L'ATTENZIONE!!11!!!!11!1!1! CAPYBARAAA