

CYBR371

LAB 3

Olivia Fletcher
300534281
fletcholiv

Denial of Service Attacks

Question A1 [0.5 marks]

What are the meaning of -s and -c parameters and what are the differences between two of the two commands?

- '-s' stdin reads commands from standard input
- In this lab it use it to -s to set the SYN tcp flag on
- '-c' reads commands from the command_string instead of from the standard input
- In the lab using the -c command directly giving the number of packets we attempt to ping with e.g ping <ubuntu ip> -c 10 (ping 10 packets to the ubuntu terminal)

Question A2 [0.5 marks]

Why aren't new operating systems susceptible of Ping of Death attack? (250 words)

- In a Ping of Death attack, another subset of DDoS attack, the attacker sends large-scale packets to the victims system than the maximum packet size the connection can handle thus directly causing the device to significantly slow down or crash.
- The Ping of Death attack originally surfaced in the 1950's and (most) devices since 1998 have been generally protected against it except for a recent attack emerging in 2013 (& 2020) where there was an exploit found on Windows XP and Windows server 2013. Many websites have tools in place to stop this type of attack by blocking ICMP ping messages.
- Since these cases adjustments have been made to the OS and server softwares such as;
 - Use of a larger memory buffer to avoid buffer overflow.
 - The router and firewall level have tools in place that filter malicious packets out of the network;
 - As the packets from PoD attacks are sent in fragments the system on the router/firewall level scans the fragments to check that the fragments do not exceed the maximum size for received packets (65535 bytes).
 - The IP header of each IP fragment is checked that the requirements are met as such; "Fragment Offset + Total length <= 65535 Bytes" otherwise the packet is rejected.

- New Operating systems are good at detecting if the packets are larger than the system can handle using the above method but there are still ways this attack can happen on modern OS's and here are a few extra ways we can protect ourselves:
 - Ensuring system and applications are all up to date
 - Block all segmented pings from accessing your network/system
 - Allow access for large data bits **after** packets received to avoid crashing
 - Use an overflow buffer to allow for larger memory packets to be received

Question A3 [0.5 marks]

How can you make the Ping of Death packets effective against a target (these days)? (250 words)

- To this day the Ping of Death attack is still a cause of concern as some legacy devices (outdated systems or devices such as medical devices or workplace systems) are vulnerable due to being outdated and use unsupported interfaces that are continuing to be relied on for use. These devices that have yet to be patched can cause significant damage to a network if attacked
- A more recent flaw in 2020 has been noted in the Windows component TCPIP.sys which is a kernel driver which could reach the core of any Windows OS if exploited
- A more recent development of the Ping of Death attack is using an ICMP (Internet Control Message Protocol) flood attack. All computers use an ICMP echo-reply message system (ping). Ping commands are limited to a maximum size of 65,535 bytes, an attacker manipulates this system and does an ICMP flood attack by;
 - Attacker sends out pings from source machine to a victim's target system and waits for an ICMP echo reply
 - Once a connection between the source and target is intact the attacker then overloads the connection with packets
- Most computers are safe against basic PoD and ICMP flood attacks but there is still risk for those who rely on outdated devices

Question B1 [0.5 marks]

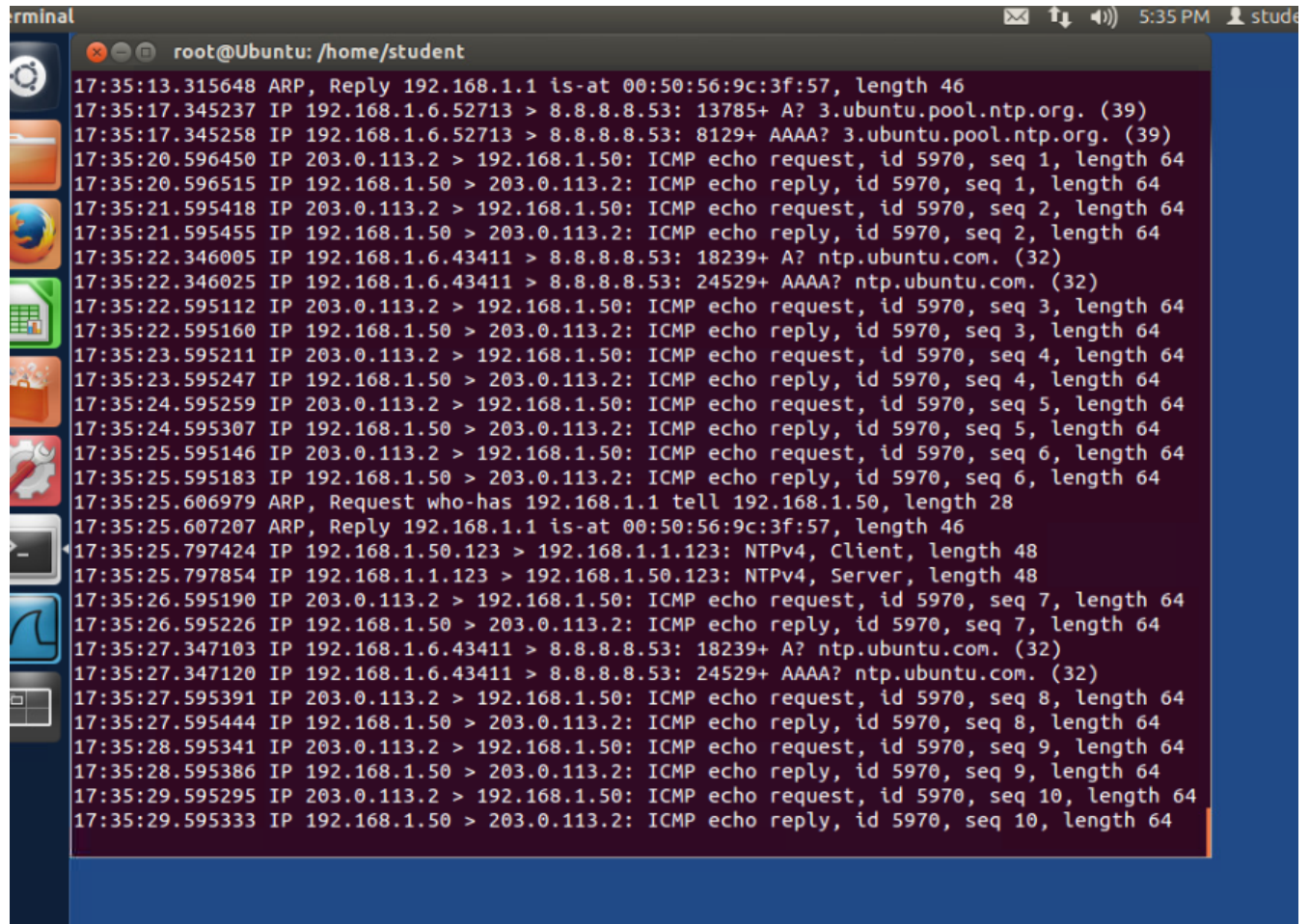
Briefly explain the countermeasures to stop and defend against a smurf attack? (250 words)

- A smurf attack, being an attack which exploits IP and ICMP vulnerabilities, renders the target's network inoperable. Prevention has to be done on two different levels, one being that your network itself must avoid being attacked and also your network must avoid being used to launch an attack due to the intermediary of the attack being that the source-spoofed IP packet has left a given network.
- To avoid being used to launch an attack you;
 - Should disable IP-directed broadcasting on the router
 - Can also apply an outbound filter to the perimeter router
 - Configure hosts and routers to not respond to ICMP echo requests
- Countermeasures anyone can do to stop and defend themselves against a smurf attack;
 - Have networking tools in place to sniff out any odd packet data like volume, size and signatures
 - Use an active up-to-date antivirus, antimalware and configured network firewalls
 - Expand on bandwidth usage to avoid traffic spikes
 - Spread your servers across multiple data centers and use a good system to balance the traffic distribution between centers
 - Using a cloud-based DNS provider where it is designed with DDoS in mind

Question B2 [0.5 marks]

How much did the smurf attack slow down the network? Compare the average ping response time before and during the smurf.

- Before attack;
- Ubuntu system picks up the ping requests from Kali, network traffic speeds of incoming/outgoing is of usual speeds



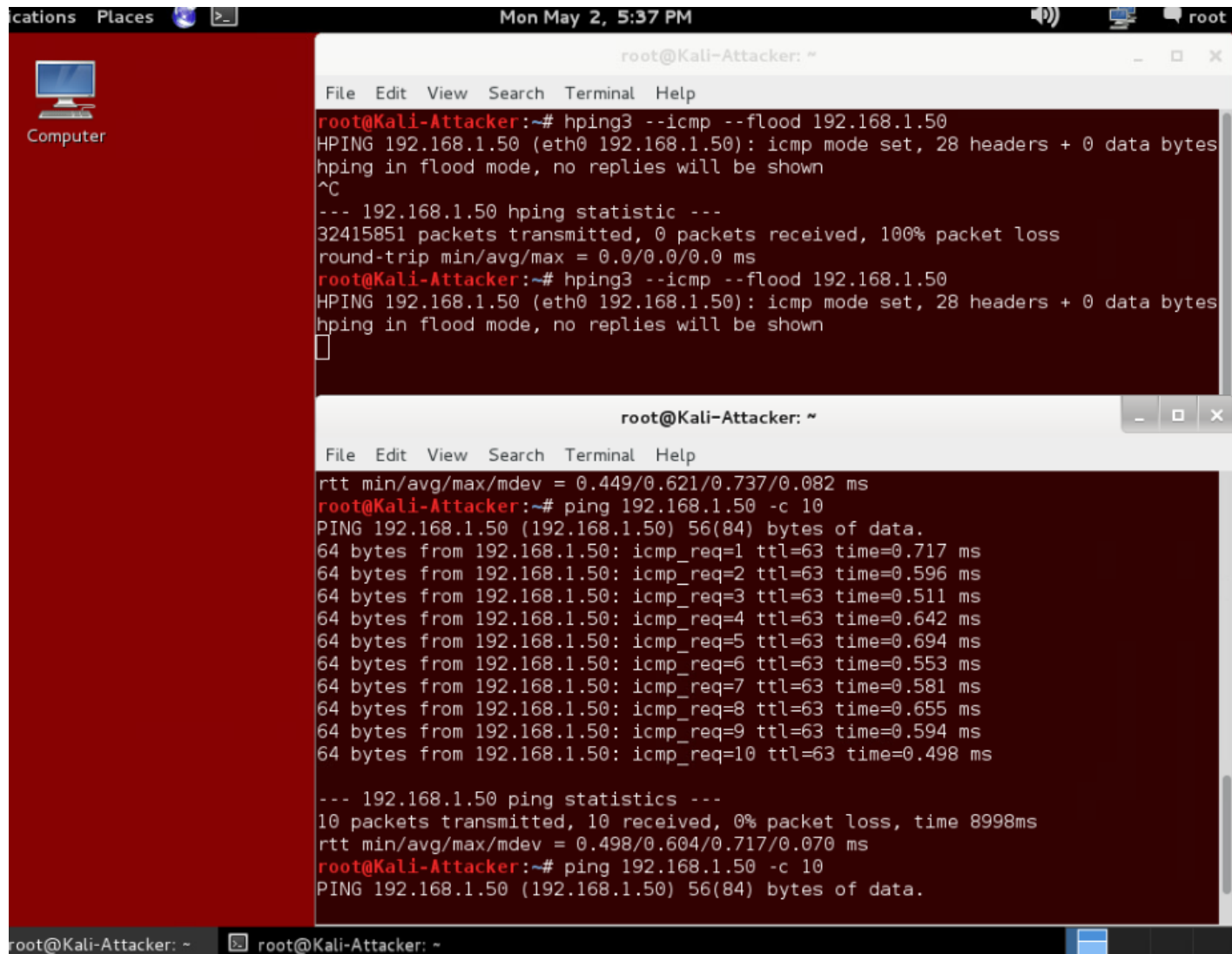
The screenshot shows a terminal window titled 'terminal' with the prompt 'root@Ubuntu: /home/student'. The terminal displays a series of network traffic logs. The logs show a mix of normal network activity and a smurf attack. The attack is characterized by a large volume of ICMP echo requests (ping requests) originating from the IP address 203.0.113.2 and being sent to the IP address 192.168.1.50. The logs also show the corresponding ICMP echo replies being sent back to the source IP address 203.0.113.2. The attack is identified as a smurf attack because the source IP address in the requests is spoofed to be the same as the destination IP address (192.168.1.50). The logs also show other network traffic, including ARP requests and replies, and NTPv4 Client and Server messages.

```
17:35:13.315648 ARP, Reply 192.168.1.1 is-at 00:50:56:9c:3f:57, length 46
17:35:17.345237 IP 192.168.1.6.52713 > 8.8.8.8.53: 13785+ A? 3.ubuntu.pool.ntp.org. (39)
17:35:17.345258 IP 192.168.1.6.52713 > 8.8.8.8.53: 8129+ AAAA? 3.ubuntu.pool.ntp.org. (39)
17:35:20.596450 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 1, length 64
17:35:20.596515 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 1, length 64
17:35:21.595418 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 2, length 64
17:35:21.595455 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 2, length 64
17:35:22.346005 IP 192.168.1.6.43411 > 8.8.8.8.53: 18239+ A? ntp.ubuntu.com. (32)
17:35:22.346025 IP 192.168.1.6.43411 > 8.8.8.8.53: 24529+ AAAA? ntp.ubuntu.com. (32)
17:35:22.595112 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 3, length 64
17:35:22.595160 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 3, length 64
17:35:23.595211 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 4, length 64
17:35:23.595247 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 4, length 64
17:35:24.595259 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 5, length 64
17:35:24.595307 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 5, length 64
17:35:25.595146 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 6, length 64
17:35:25.595183 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 6, length 64
17:35:25.606979 ARP, Request who-has 192.168.1.1 tell 192.168.1.50, length 28
17:35:25.607207 ARP, Reply 192.168.1.1 is-at 00:50:56:9c:3f:57, length 46
17:35:25.797424 IP 192.168.1.50.123 > 192.168.1.1.123: NTPv4, Client, length 48
17:35:25.797854 IP 192.168.1.1.123 > 192.168.1.50.123: NTPv4, Server, length 48
17:35:26.595190 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 7, length 64
17:35:26.595226 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 7, length 64
17:35:27.347103 IP 192.168.1.6.43411 > 8.8.8.8.53: 18239+ A? ntp.ubuntu.com. (32)
17:35:27.347120 IP 192.168.1.6.43411 > 8.8.8.8.53: 24529+ AAAA? ntp.ubuntu.com. (32)
17:35:27.595391 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 8, length 64
17:35:27.595444 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 8, length 64
17:35:28.595341 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 9, length 64
17:35:28.595386 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 9, length 64
17:35:29.595295 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 5970, seq 10, length 64
17:35:29.595333 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 5970, seq 10, length 64
```

- Takes only 0.717ms for a successful ping from the Kali system to the Ubuntu system

```
root@Kali-Attacker: ~  
File Edit View Search Terminal Help  
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.  
64 bytes from 192.168.1.50: icmp_req=1 ttl=63 time=0.737 ms  
64 bytes from 192.168.1.50: icmp_req=2 ttl=63 time=0.563 ms  
64 bytes from 192.168.1.50: icmp_req=3 ttl=63 time=0.659 ms  
64 bytes from 192.168.1.50: icmp_req=4 ttl=63 time=0.700 ms  
64 bytes from 192.168.1.50: icmp_req=5 ttl=63 time=0.606 ms  
64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.672 ms  
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.648 ms  
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.609 ms  
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.449 ms  
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.573 ms  
--- 192.168.1.50 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 8998ms  
rtt min/avg/max/mdev = 0.449/0.621/0.737/0.082 ms  
root@Kali-Attacker:~# ping 192.168.1.50 -c 10  
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.  
64 bytes from 192.168.1.50: icmp_req=1 ttl=63 time=0.717 ms  
64 bytes from 192.168.1.50: icmp_req=2 ttl=63 time=0.596 ms  
64 bytes from 192.168.1.50: icmp_req=3 ttl=63 time=0.511 ms  
64 bytes from 192.168.1.50: icmp_req=4 ttl=63 time=0.642 ms  
64 bytes from 192.168.1.50: icmp_req=5 ttl=63 time=0.694 ms  
64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.553 ms  
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.581 ms  
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.655 ms  
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.594 ms  
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.498 ms  
--- 192.168.1.50 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 8998ms  
rtt min/avg/max/mdev = 0.498/0.604/0.717/0.070 ms  
root@Kali-Attacker:~#
```

- During attack;
- Kali (in a separate terminal) attempts to ping the Ubuntu system but is taking significantly longer than the previous requests before launching the attack.



```

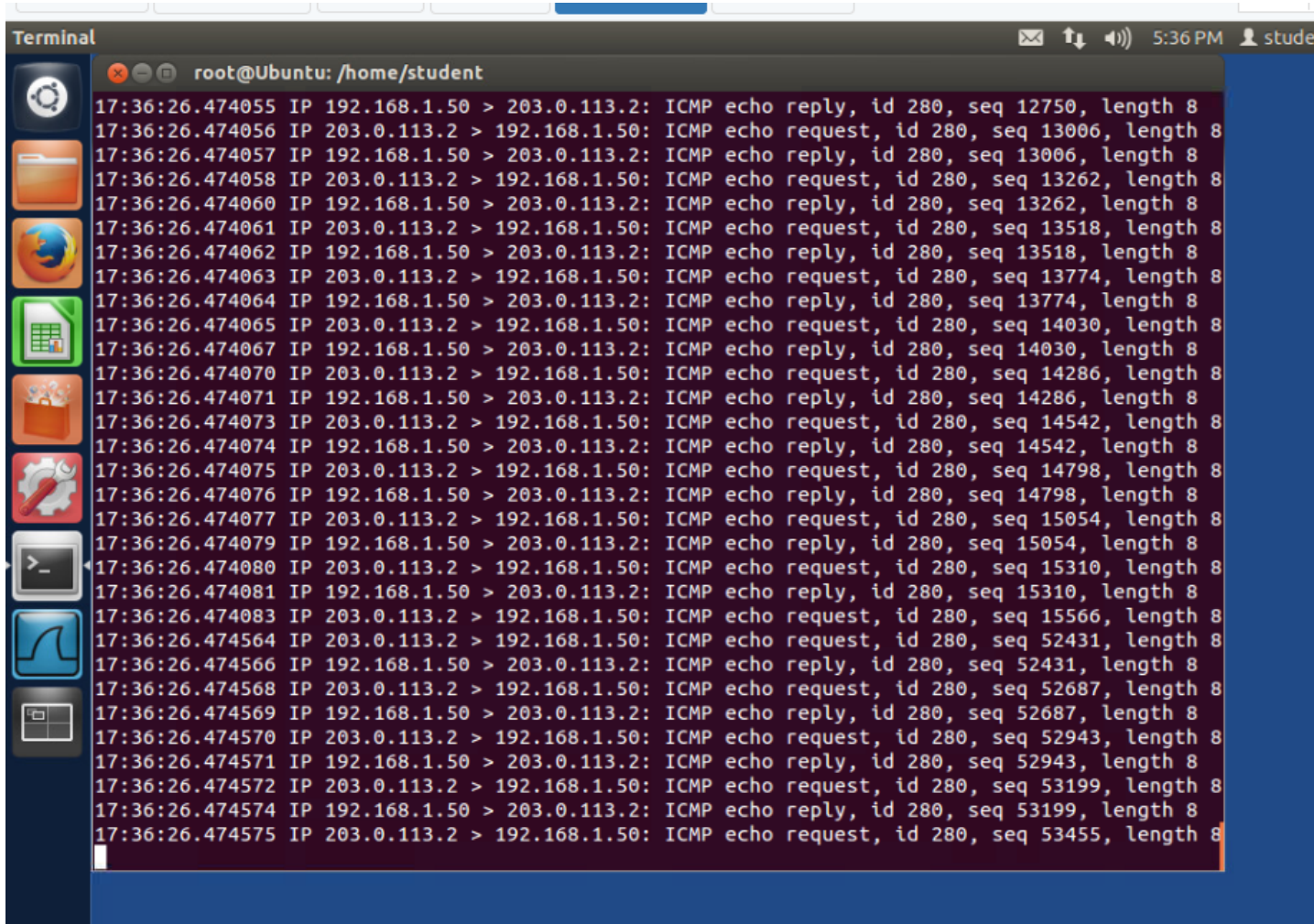
root@Kali-Attacker: ~
File Edit View Search Terminal Help
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.50 hping statistic ---
32415851 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
[ ]

root@Kali-Attacker: ~
File Edit View Search Terminal Help
rtt min/avg/max/mdev = 0.449/0.621/0.737/0.082 ms
root@Kali-Attacker:~# ping 192.168.1.50 -c 10
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.
64 bytes from 192.168.1.50: icmp_req=1 ttl=63 time=0.717 ms
64 bytes from 192.168.1.50: icmp_req=2 ttl=63 time=0.596 ms
64 bytes from 192.168.1.50: icmp_req=3 ttl=63 time=0.511 ms
64 bytes from 192.168.1.50: icmp_req=4 ttl=63 time=0.642 ms
64 bytes from 192.168.1.50: icmp_req=5 ttl=63 time=0.694 ms
64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.553 ms
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.581 ms
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.655 ms
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.594 ms
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.498 ms

--- 192.168.1.50 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8998ms
rtt min/avg/max/mdev = 0.498/0.604/0.717/0.070 ms
root@Kali-Attacker:~# ping 192.168.1.50 -c 10
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.

```


- The Ubuntu terminal had traffic coming in at a significant rate compared to before the attack



The screenshot shows a terminal window titled "Terminal" with the prompt "root@Ubuntu: /home/student". The terminal displays a rapid stream of network traffic logs. Each line represents an ICMP echo request or reply, including the timestamp, IP addresses, and packet details like ID, sequence number, and length. The traffic is bidirectional, with requests from 203.0.113.2 to 192.168.1.50 and replies from 192.168.1.50 to 203.0.113.2. The sequence numbers increase by 20 for each request-reply pair.

```
17:36:26.474055 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 12750, length 8
17:36:26.474056 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 13006, length 8
17:36:26.474057 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 13006, length 8
17:36:26.474058 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 13262, length 8
17:36:26.474060 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 13262, length 8
17:36:26.474061 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 13518, length 8
17:36:26.474062 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 13518, length 8
17:36:26.474063 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 13774, length 8
17:36:26.474064 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 13774, length 8
17:36:26.474065 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 14030, length 8
17:36:26.474067 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 14030, length 8
17:36:26.474070 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 14286, length 8
17:36:26.474071 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 14286, length 8
17:36:26.474073 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 14542, length 8
17:36:26.474074 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 14542, length 8
17:36:26.474075 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 14798, length 8
17:36:26.474076 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 14798, length 8
17:36:26.474077 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 15054, length 8
17:36:26.474079 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 15054, length 8
17:36:26.474080 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 15310, length 8
17:36:26.474081 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 15310, length 8
17:36:26.474083 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 15566, length 8
17:36:26.474564 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 52431, length 8
17:36:26.474566 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 52431, length 8
17:36:26.474568 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 52687, length 8
17:36:26.474569 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 52687, length 8
17:36:26.474570 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 52943, length 8
17:36:26.474571 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 52943, length 8
17:36:26.474572 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 53199, length 8
17:36:26.474574 IP 192.168.1.50 > 203.0.113.2: ICMP echo reply, id 280, seq 53199, length 8
17:36:26.474575 IP 203.0.113.2 > 192.168.1.50: ICMP echo request, id 280, seq 53455, length 8
```

- After attack;
- Failed to ping Ubuntu system as the attack rendered the network unable to be pinged
- 100% packet loss, reached the packet limit (10 in this case)

```

root@Kali-Attacker: ~
File Edit View Search Terminal Help
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.50 hping statistic ---
32415851 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C

root@Kali-Attacker: ~
File Edit View Search Terminal Help
64 bytes from 192.168.1.50: icmp_req=2 ttl=63 time=0.596 ms
64 bytes from 192.168.1.50: icmp_req=3 ttl=63 time=0.511 ms
64 bytes from 192.168.1.50: icmp_req=4 ttl=63 time=0.642 ms
64 bytes from 192.168.1.50: icmp_req=5 ttl=63 time=0.694 ms
64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.553 ms
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.581 ms
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.655 ms
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.594 ms
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.498 ms

--- 192.168.1.50 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8998ms
rtt min/avg/max/mdev = 0.498/0.604/0.717/0.070 ms
root@Kali-Attacker:~# ping 192.168.1.50 -c 10
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.

--- 192.168.1.50 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9063ms
root@Kali-Attacker:~#

```


- Another test;
- Did this again but using - 100 packets
- First ping request goes through at 20.4ms which is significantly longer time than the 0.717 it took to ping the ubuntu system before Kali attacking

```

Places  Mon May 2, 5:51 PM  root@Kali-Attacker: ~

root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.50 hping statistic ---
32415851 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.50 hping statistic ---
63528140 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
10 packets transmitted, 0 received, 100% packet loss, time 9063ms

root@Kali-Attacker:~# ping 192.168.1.50 -c 100
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data:
64 bytes from 192.168.1.50: icmp_req=28 ttl=63 time=20.4 ms
64 bytes from 192.168.1.50: icmp_req=48 ttl=63 time=3.43 ms
64 bytes from 192.168.1.50: icmp_req=55 ttl=63 time=23.0 ms

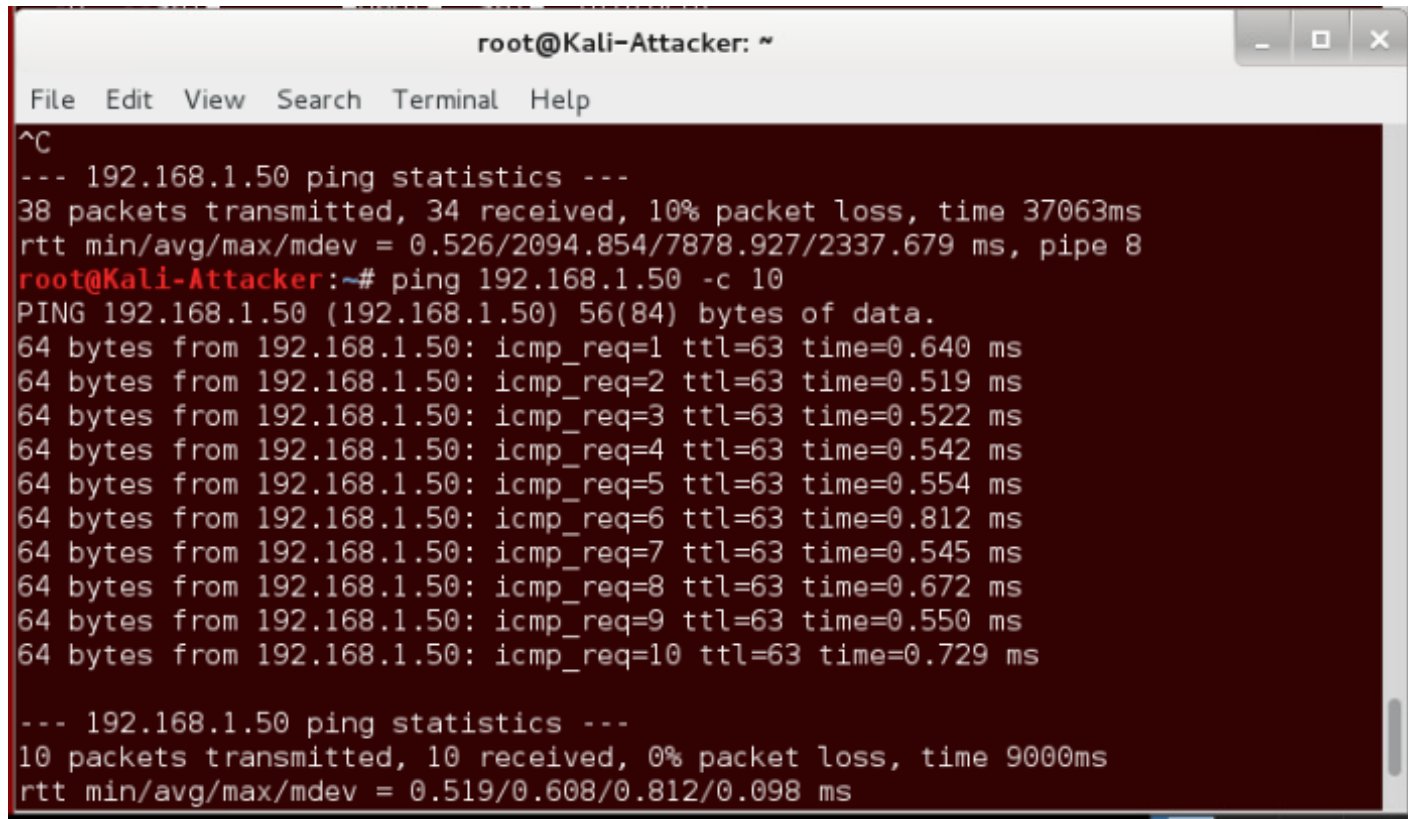
--- 192.168.1.50 ping statistics ---
100 packets transmitted, 3 received, 97% packet loss, time 99766ms
rtt min/avg/max/mdev = 3.430/15.644/23.013/8.698 ms
root@Kali-Attacker:~#

```

Question B3 [0.5 marks]

Write an hping3 command to launch a smurf attack against the Ubuntu machine and randomly spoof the source IP addresses

- Before the attack we get the following ping times;



```
root@Kali-Attacker: ~  
File Edit View Search Terminal Help  
^C  
--- 192.168.1.50 ping statistics ---  
38 packets transmitted, 34 received, 10% packet loss, time 37063ms  
rtt min/avg/max/mdev = 0.526/2094.854/7878.927/2337.679 ms, pipe 8  
root@Kali-Attacker:~# ping 192.168.1.50 -c 10  
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.  
64 bytes from 192.168.1.50: icmp_req=1 ttl=63 time=0.640 ms  
64 bytes from 192.168.1.50: icmp_req=2 ttl=63 time=0.519 ms  
64 bytes from 192.168.1.50: icmp_req=3 ttl=63 time=0.522 ms  
64 bytes from 192.168.1.50: icmp_req=4 ttl=63 time=0.542 ms  
64 bytes from 192.168.1.50: icmp_req=5 ttl=63 time=0.554 ms  
64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.812 ms  
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.545 ms  
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.672 ms  
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.550 ms  
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.729 ms  
  
--- 192.168.1.50 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9000ms  
rtt min/avg/max/mdev = 0.519/0.608/0.812/0.098 ms
```

- To launch the attack I used the following command;
- hping3 -icmp -flood 192.168.1.50 -a 196.168.176.255
- The -a flag in this command line allows for spoofing the IP

```

Mon May 2, 6:41 PM
root@Kali-Attacker: ~

File Edit View Search Terminal Help

-U --urg      set URG flag
-X --xmas     set X unused flag (0x40)
-Y --ymas     set Y unused flag (0x80)
--tcpexitcode use last tcp->th_flags as exit code
--tcp-mss     enable the TCP MSS option with the given value
--tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime
Common
-d --data     data size (default is 0)
-E --file     data from file
-e --sign     add 'signature'
-j --dump     dump packets in hex
-J --print    dump printable characters
-B --safe     enable 'safe' protocol
-u --end      tell you when --file reached EOF and prevent rewind
-T --traceroute traceroute mode (implies --bind and --ttl 1)
--tr-stop     Exit when receive the first not ICMP in traceroute mode
--tr-keep-ttl Keep the source TTL fixed, useful to monitor just one hop
--tr-no-rtt   Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
--apd-send    Send the packet described with APD (see docs/APD.txt)
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50 -a 196.176.255
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown

64 bytes from 192.168.1.50: icmp_req=6 ttl=63 time=0.812 ms
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.545 ms
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.672 ms
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.550 ms
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.729 ms

--- 192.168.1.50 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.519/0.608/0.812/0.098 ms
root@Kali-Attacker:~# ping 192.168.1.50 -c 100
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.

```

- Below is the ubuntu terminal during the attack using the 196.168.176.255 spoofed IP

```
Terminal
root@Ubuntu: /home/student
8
18:42:08.190261 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 21873, length 8
18:42:08.190262 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 21873, length 8
18:42:08.190263 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 22129, length 8
18:42:08.190265 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 22129, length 8
18:42:08.190266 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 22385, length 8
18:42:08.190267 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 22385, length 8
18:42:08.190268 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 22641, length 8
18:42:08.190269 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 22641, length 8
18:42:08.190271 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 22897, length 8
18:42:08.190272 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 22897, length 8
18:42:08.190275 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 23153, length 8
18:42:08.190276 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 23153, length 8
18:42:08.190277 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 23409, length 8
18:42:08.190278 IP 192.168.1.50 > 196.176.0.255: ICMP echo reply, id 16198, seq 23409, length 8
18:42:08.190279 IP 196.176.0.255 > 192.168.1.50: ICMP echo request, id 16198, seq 23665, length 8
```

- Below is showing the attempted pings during the Kali attack and the times being significantly higher than before the attack
- Before attack first ping: 0.640ms
- During attack first ping: 312ms

```

Places  Mon May 2, 6:45 PM  root
root@Kali-Attacker: ~
File Edit View Search Terminal Help
--tr-keep-ttl    Keep the source TTL fixed, useful to monitor just one hop
--tr-no-rtt      Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
--apd-send       Send the packet described with APD (see docs/APD.txt)
root@Kali-Attacker:~# hping3 --icmp --flood 192.168.1.50 -a 196.176.255
HPING 192.168.1.50 (eth0 192.168.1.50): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
^C
--- 192.168.1.50 hping statistic ---
96363660 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@Kali-Attacker:~#

root@Kali-Attacker: ~
File Edit View Search Terminal Help
64 bytes from 192.168.1.50: icmp_req=7 ttl=63 time=0.545 ms
64 bytes from 192.168.1.50: icmp_req=8 ttl=63 time=0.672 ms
64 bytes from 192.168.1.50: icmp_req=9 ttl=63 time=0.550 ms
64 bytes from 192.168.1.50: icmp_req=10 ttl=63 time=0.729 ms

--- 192.168.1.50 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.519/0.608/0.812/0.098 ms
root@Kali-Attacker:~# ping 192.168.1.50 -c 100
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.
64 bytes from 192.168.1.50: icmp_req=19 ttl=63 time=312 ms
64 bytes from 192.168.1.50: icmp_req=34 ttl=63 time=17.6 ms
64 bytes from 192.168.1.50: icmp_req=44 ttl=63 time=421 ms
64 bytes from 192.168.1.50: icmp_req=65 ttl=63 time=17.2 ms
64 bytes from 192.168.1.50: icmp_req=66 ttl=63 time=17.1 ms

--- 192.168.1.50 ping statistics ---
100 packets transmitted, 5 received, 95% packet loss, time 99488ms
rtt min/avg/max/mdev = 17.185/157.084/421.026/174.520 ms
root@Kali-Attacker:~#

```

Question C1 [1 marks]

Briefly explain the countermeasures to stop and defend against a SYN-flood attack? (250 words)

- Three processes (three way handshake) must take place for a TCP protocol to be successful between a client and server connection, this process is initiated when the client sends a SYN message to the server, the server then receives the message and responds with a SYN-ACK message back to the client, then the client confirms the connection with the ACK message. The SYN flood happens when the three-way process (handshake) above is manipulated so the attacker can then rapidly initiate a connection to the server without confirming the connection (last process of the handshake) with the ACK. The server then expends an excess of resources waiting for the opened connections thus resulting in the system becoming unresponsive to, other, legitimate traffic.
- Countermeasures in place to avoid a SYN-flood attack are;
 - Ensure firewall softwares, OS, and antiviruses are up to date
 - Installing an IPS to sniff out unusual packet data like traffic patterns, volume, size and signatures
 - Configure the firewall for SYN flood protection and SYN attack thresholds
 - Use tools that allow for easier access to view all network traffic to ensure simpler constant analysis across the entire network
 - Similar to smurf attack, expand on bandwidth usage to avoid traffic spikes and manage larger scale attacks