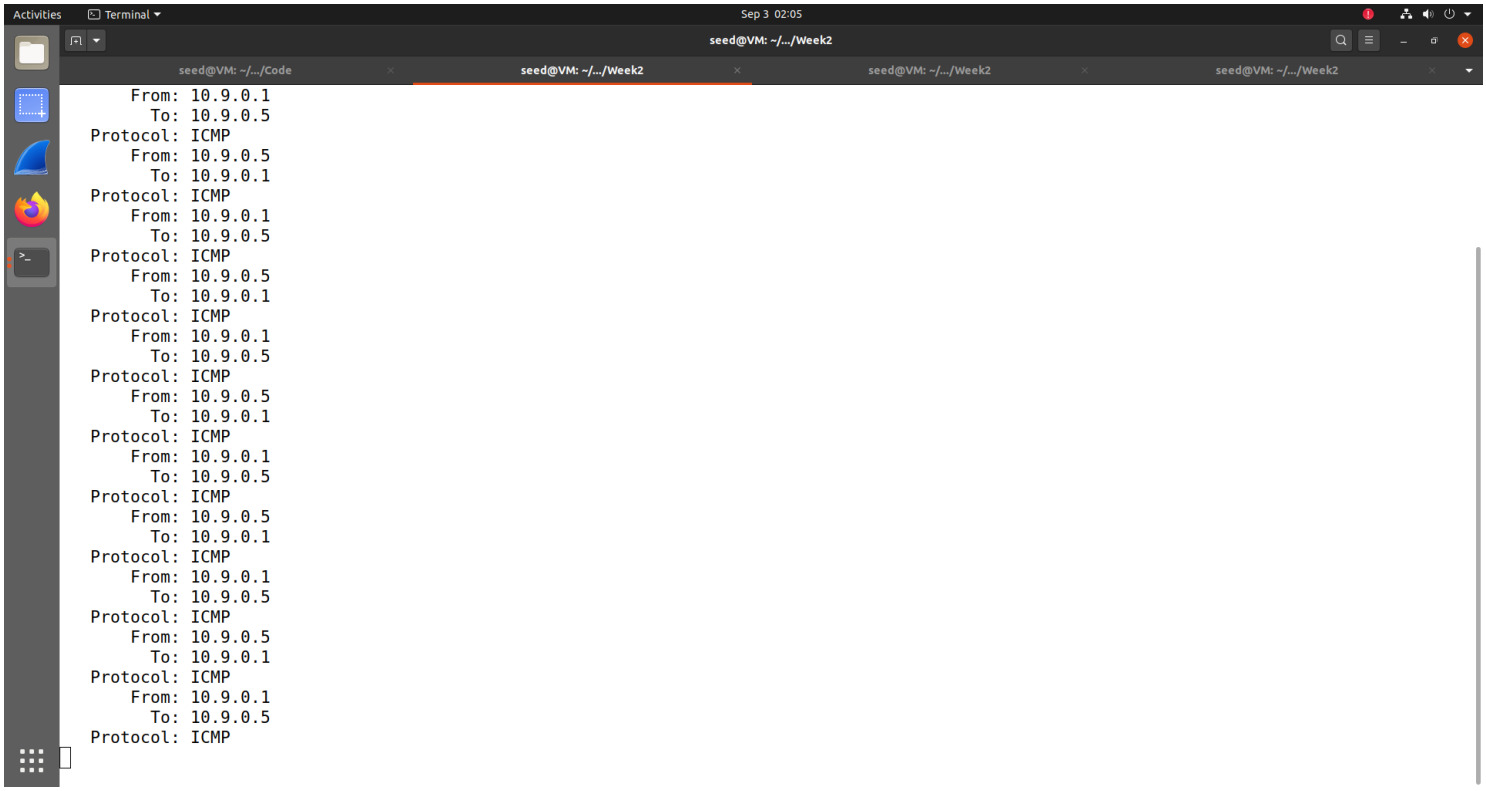
H M Mythreya
PES2UG20CS130
CNS-Lab Week-2

Task 2.1A: Sniffing Packets

It is important to turn on promiscuous mode so that the packets not addressed to the attacker's machine is also received and can be sniffed.

```
handle = pcap_open_live("br-ea9efb963438", BUFSIZ, 1, 1000, errbuf);
                                                   ^
                                    1 = promiscuous is on
```

Output



```
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
```

Important library calls:

Question1)

pcap_open_live() → Used to obtain a packet capture handle in order to look at/sniff at packets in the network

pcap_compile() → Used to compile a string that contains the filter into a filter program

Question2)

In root mode, the code doesn't work since the function being called needs root access to work.

Without root:

```
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.1
  Protocol: ICMP
      From: 10.9.0.1
        To: 10.9.0.5
  Protocol: ICMP
^C
root@VM:/volumes# su seed
seed@VM:/volumes$ ./sniff
Segmentation fault (core dumped)
seed@VM:/volumes$ su root
Password:
root@VM:/volumes#
```

Segmentation Fault Error.

Without Promiscuous mode:

```
root@VM:/volumes# ./sniff
```

```
root@c2ac12fc334b:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.078 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.061 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.061 ms
64 bytes from 10.9.0.6: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.9.0.6: icmp_seq=7 ttl=64 time=0.056 ms
64 bytes from 10.9.0.6: icmp_seq=8 ttl=64 time=0.064 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.052 ms
64 bytes from 10.9.0.6: icmp_seq=10 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=11 ttl=64 time=0.058 ms
64 bytes from 10.9.0.6: icmp_seq=12 ttl=64 time=0.055 ms
64 bytes from 10.9.0.6: icmp_seq=13 ttl=64 time=0.054 ms
64 bytes from 10.9.0.6: icmp_seq=14 ttl=64 time=0.061 ms
```

Task 2.1B: Writing Filters

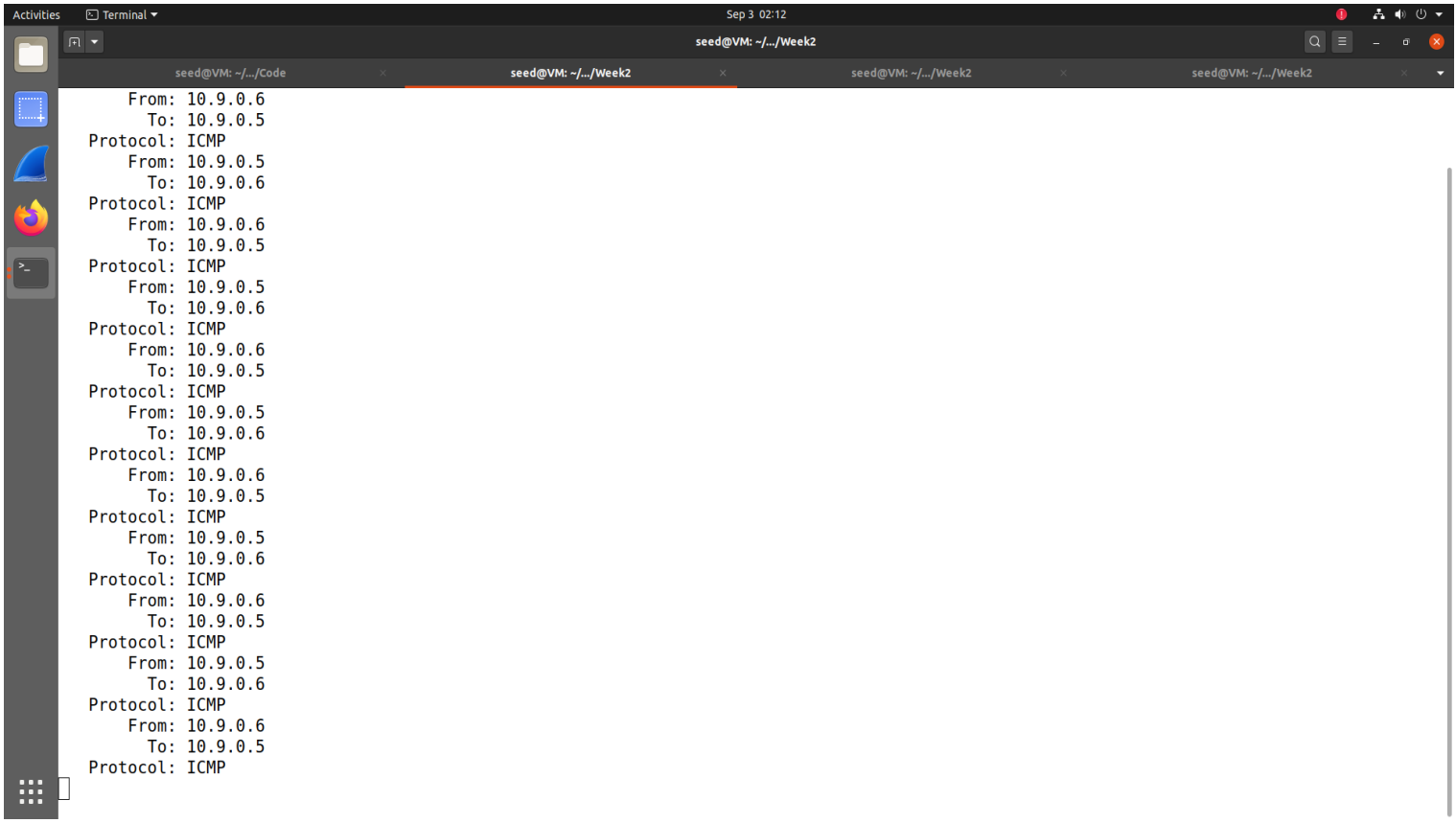
Filters:

In the real world, a network will have thousands if not hundreds of thousands of packets flowing to and from. Going through every packet would be impractical, hence we use Filters. Filters only show the relevant packets depending on the filter. For example, using an ICMP filter will only sniff ICMP packets, and so on.

First, Host A pings Host B using the ping command. The attacker then runs the sniffing code with any filter they may desire.

Filter 1) ICMP

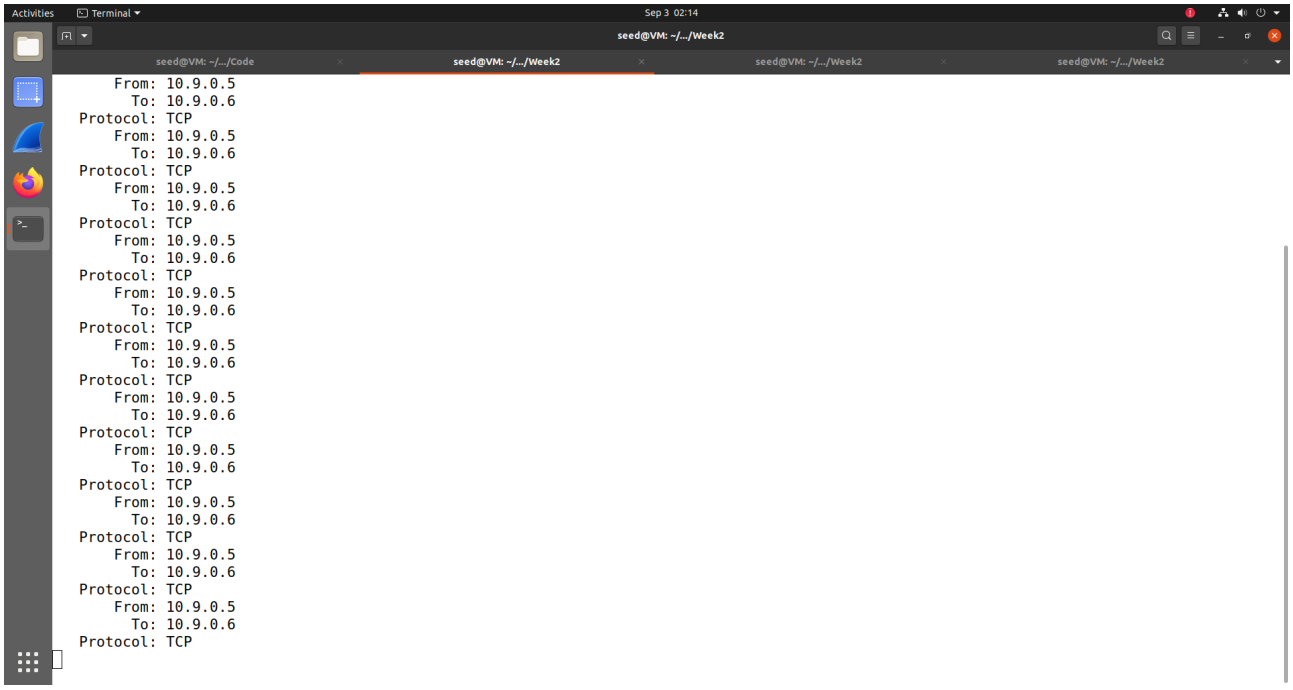
```
char filter_exp[] = "proto ICMP and (host 10.9.0.5 and 10.9.0.6)";
```



```
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.6
  Protocol: ICMP
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.6
  Protocol: ICMP
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.6
  Protocol: ICMP
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.6
  Protocol: ICMP
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
      From: 10.9.0.5
        To: 10.9.0.6
  Protocol: ICMP
      From: 10.9.0.6
        To: 10.9.0.5
  Protocol: ICMP
```

Filter 2) TCP, destination port range 10-100.

```
char filter_exp[] = "proto TCP and dst portrange 10-100";
```

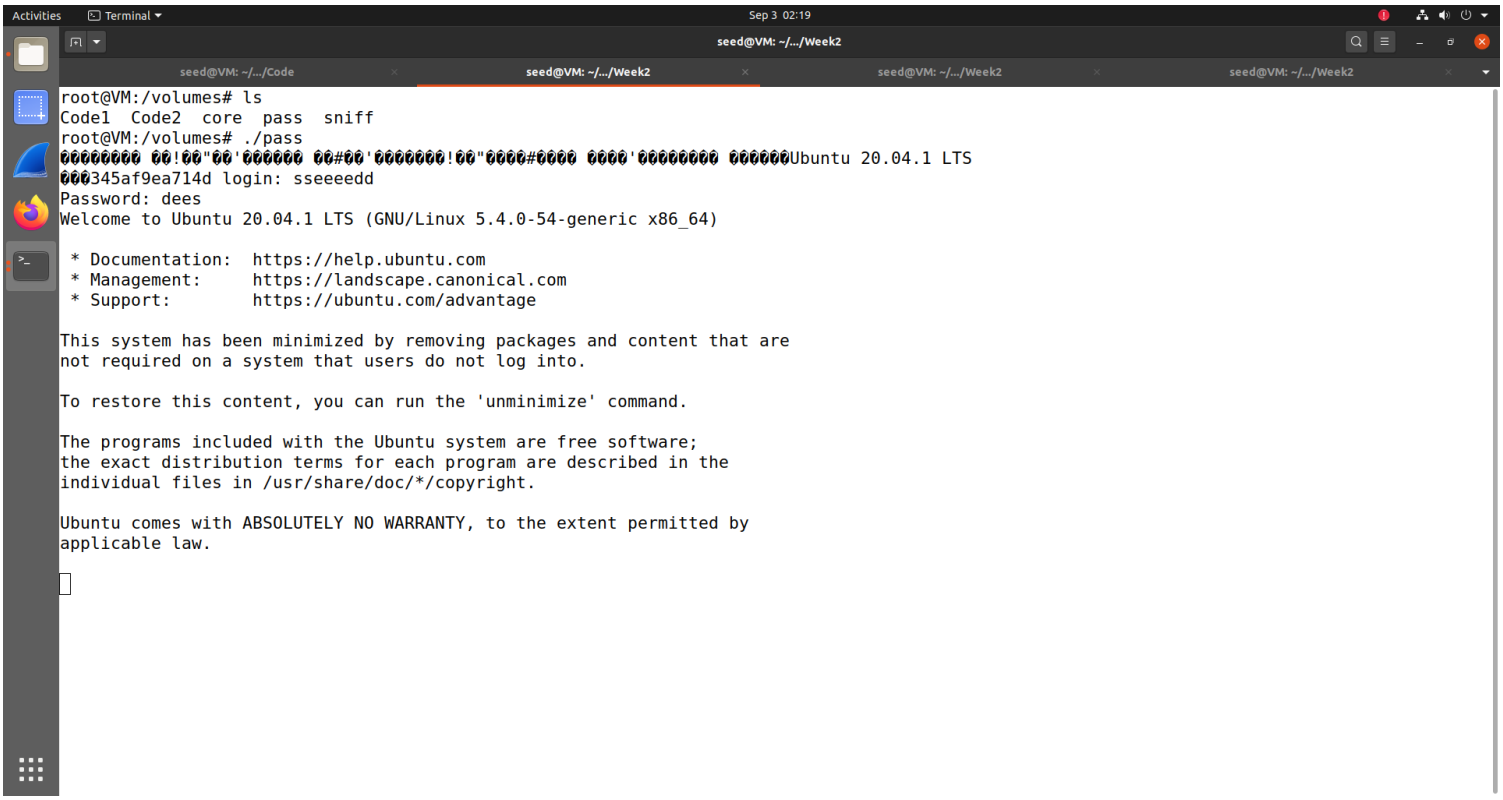


```
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
        From: 10.9.0.5
          To: 10.9.0.6
    Protocol: TCP
```

Task 2.1 C: Sniff Passwords on Telnet

```
char filter_exp[] = "port 23";
```

Output

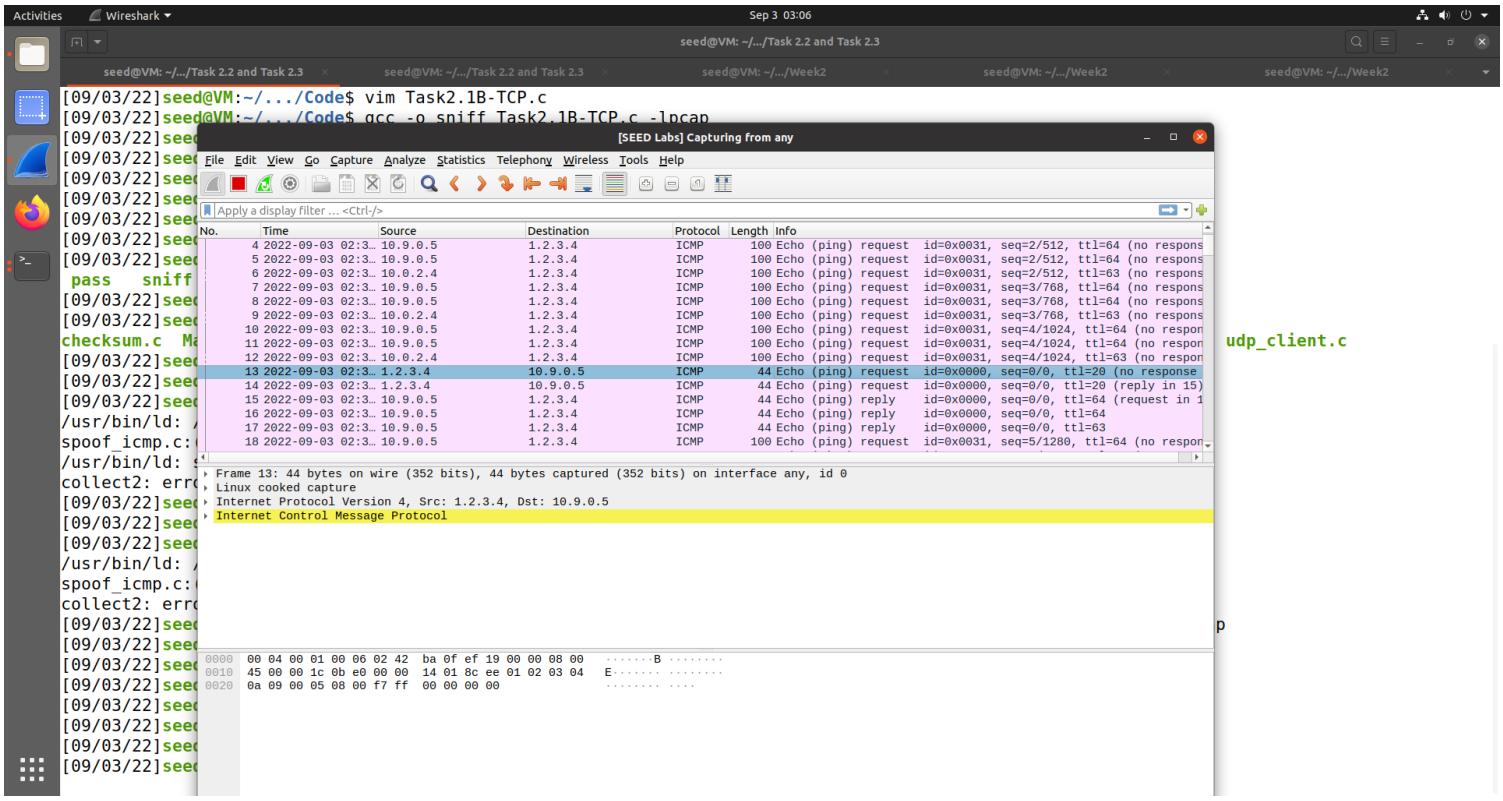


As seen above, the password is visible: dees

Task 2.2B: Spoofing Packets

Spoofing is the process of pretending/assuming someone else's ID to trick the victim into accepting communication with the attacker. The attacker's actual ID is hidden.

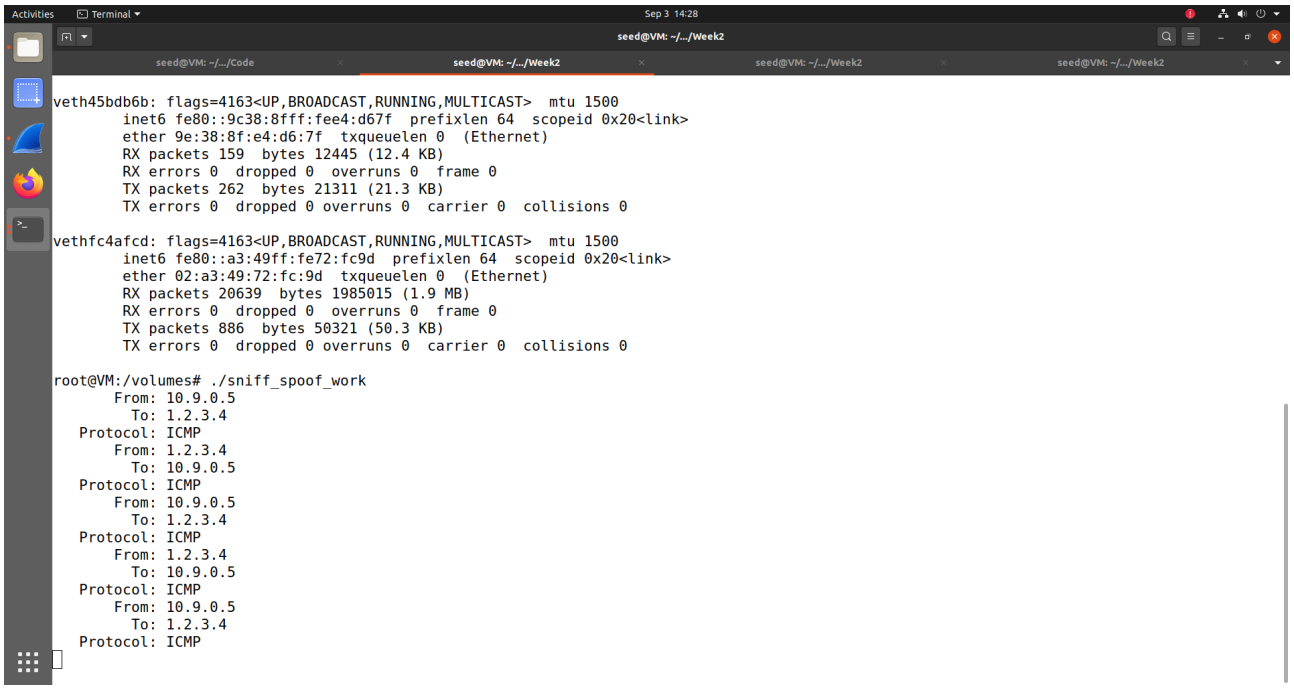
Output

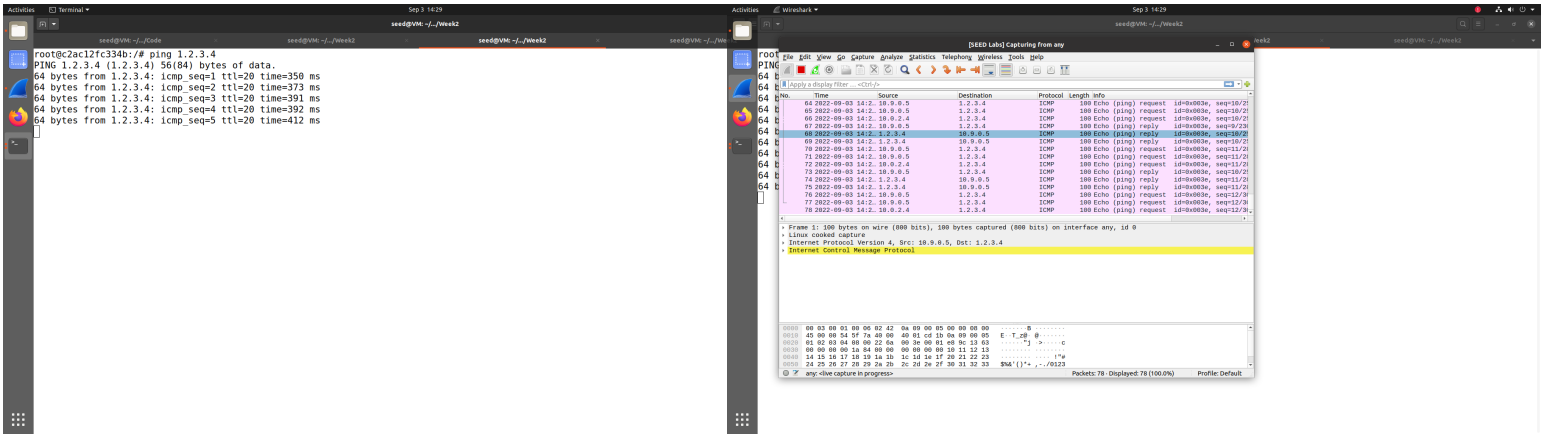


Question 4: Yes you have to calculate the checksum.

Question 5: Raw sockets require root privileges because creating a raw socket require kernel commands and requires direct interaction with the network card.

Task 2.3 Sniff and then Spoof





As seen above, the Host A receives a response even though 1.2.3.4 doesn't exist. By sniffing and spoofing, the attacker has claimed the identity of 1.2.3.4 and established a communication with Host A.