Project
Due: July 29, 2024 @5PM
100 Points

Dmytro Poliak
Student Number: 100443432
Section: S50
Date: 29.07.2024

Deliverable:

- PDF file as per the instructions below uploaded into Moodle

Objectives:

- Learn how to configure VM machines
- Learn how to install multiple Linux OSes and create a new user_id and use root id.
- Learn Linux software installation
- Learn software firewalls configuration:
  - Start and stop
  - Configure firewalls rules based on network:
    - Ports
    - IP addresses
- Learn using network monitoring tools such as Wireshark
  - View TCP/IP packets live
  - Trace TCP/IP packets
  - Identify packets valuable information

NOTES:

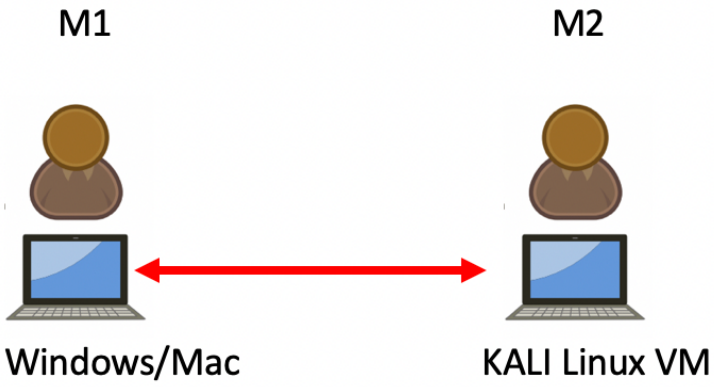
- No late submission will be accepted

Tasks:

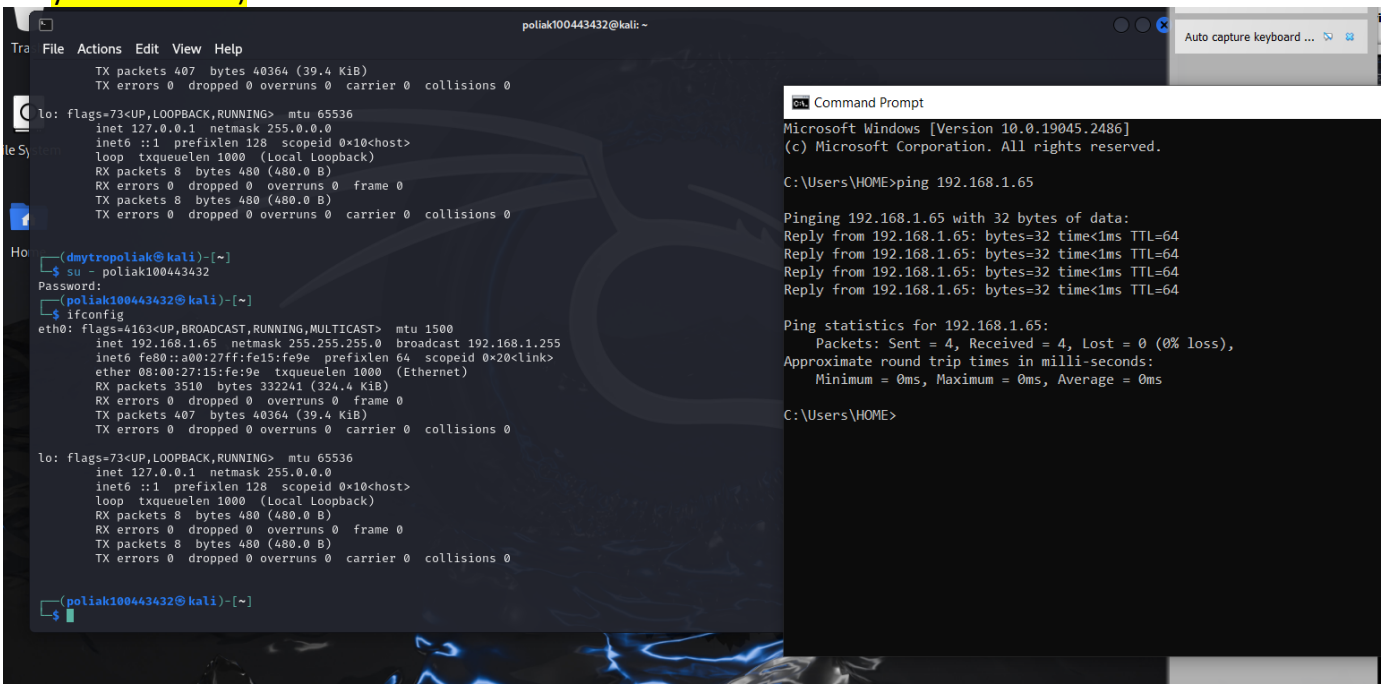
1. Install Kali Linux on M1
2. [10] Create a User_ID similar to your KPU ID on M2 using the command line. (provide SC with the command used)

Answer: sudo adduser poliak100443432

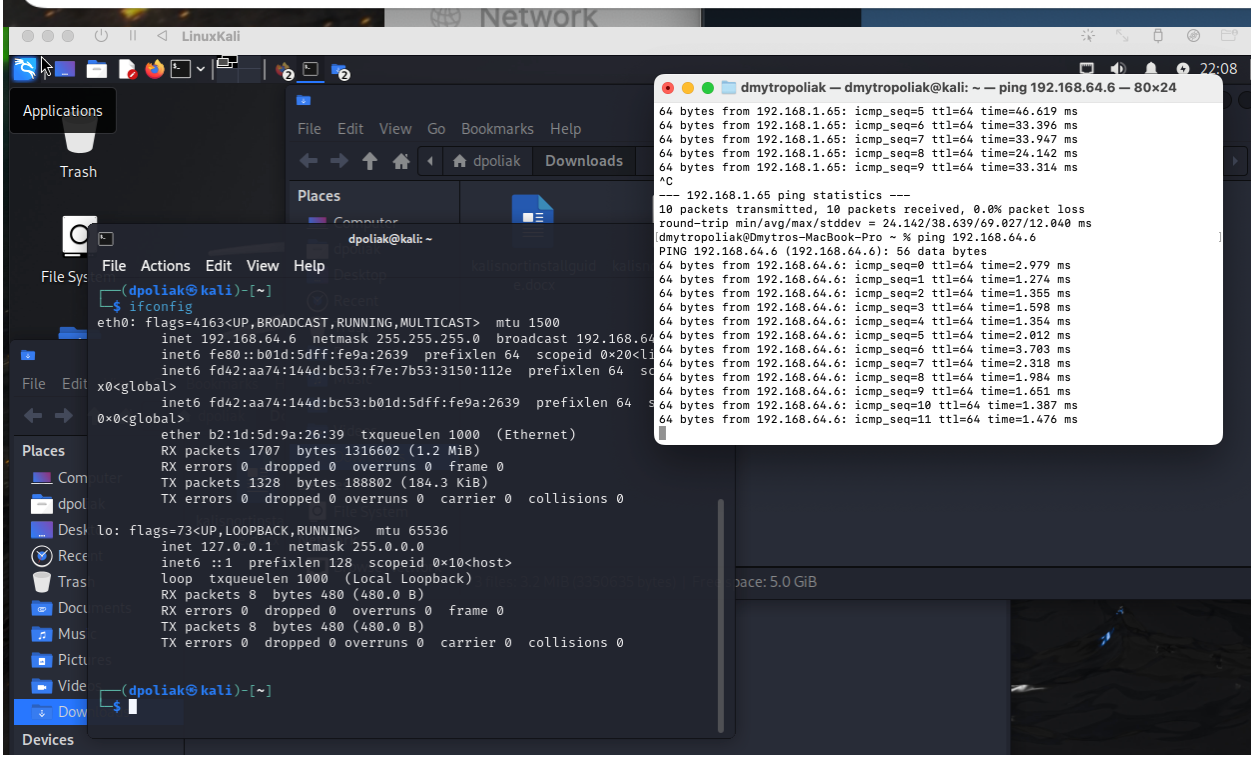
(switch to it for the ping: su – poliak100443432)



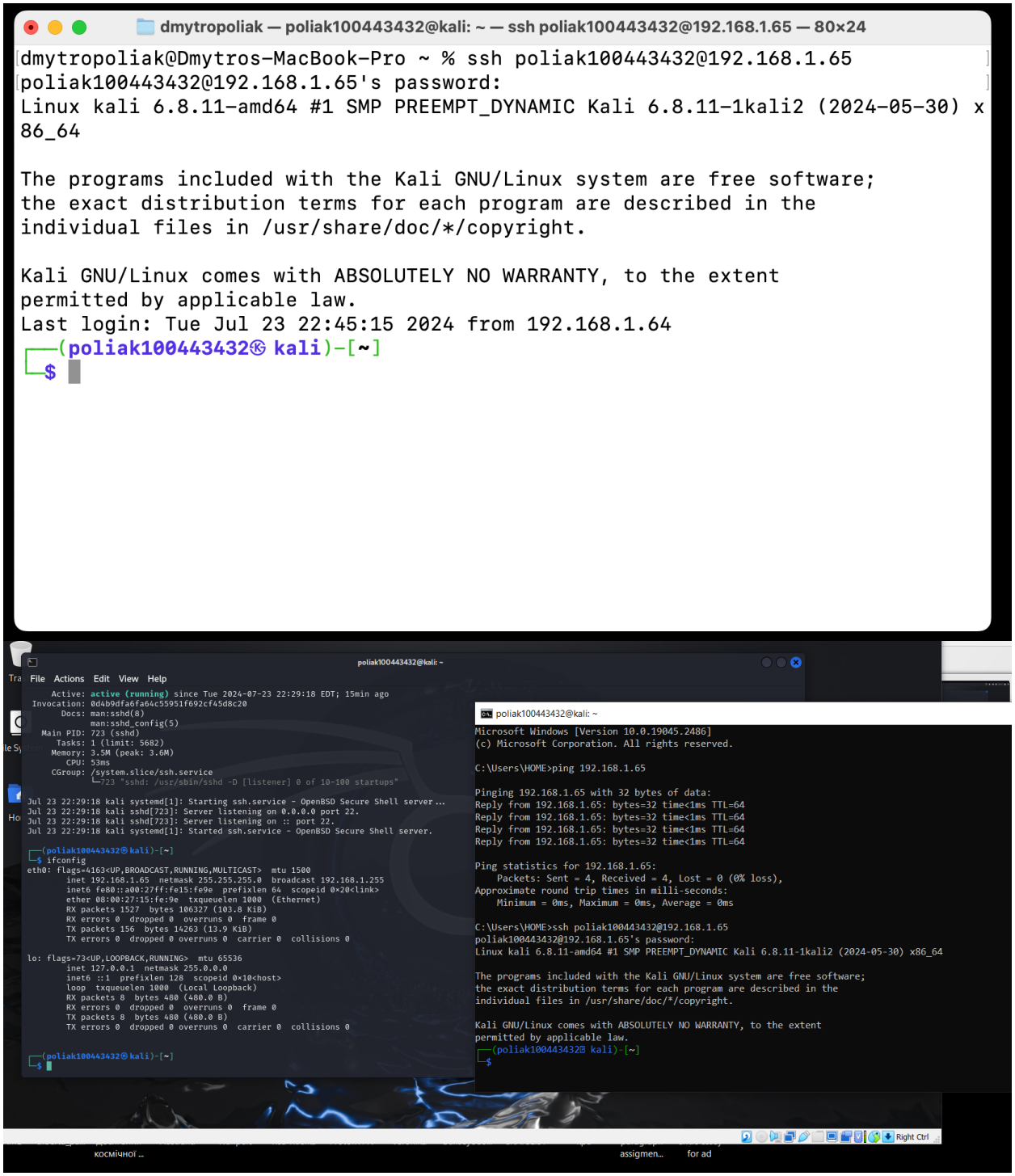
3. Verify the installation by provide a ping screenshot between M1 & M2 (Provide SC showing your ID on M2)



```
[dmytropoliak@Dmytros-MacBook-Pro ~ % ping 192.168.1.65
PING 192.168.1.65 (192.168.1.65): 56 data bytes
64 bytes from 192.168.1.65: icmp_seq=0 ttl=64 time=69.027 ms
64 bytes from 192.168.1.65: icmp_seq=1 ttl=64 time=32.976 ms
64 bytes from 192.168.1.65: icmp_seq=2 ttl=64 time=31.795 ms
64 bytes from 192.168.1.65: icmp_seq=3 ttl=64 time=47.225 ms
64 bytes from 192.168.1.65: icmp_seq=4 ttl=64 time=33.953 ms
64 bytes from 192.168.1.65: icmp_seq=5 ttl=64 time=46.619 ms
64 bytes from 192.168.1.65: icmp_seq=6 ttl=64 time=33.396 ms
64 bytes from 192.168.1.65: icmp_seq=7 ttl=64 time=33.947 ms
64 bytes from 192.168.1.65: icmp_seq=8 ttl=64 time=24.142 ms
64 bytes from 192.168.1.65: icmp_seq=9 ttl=64 time=33.314 ms
^C
--- 192.168.1.65 ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 24.142/38.639/69.027/12.040 ms
dmytropoliak@Dmytros-MacBook-Pro ~ %
```

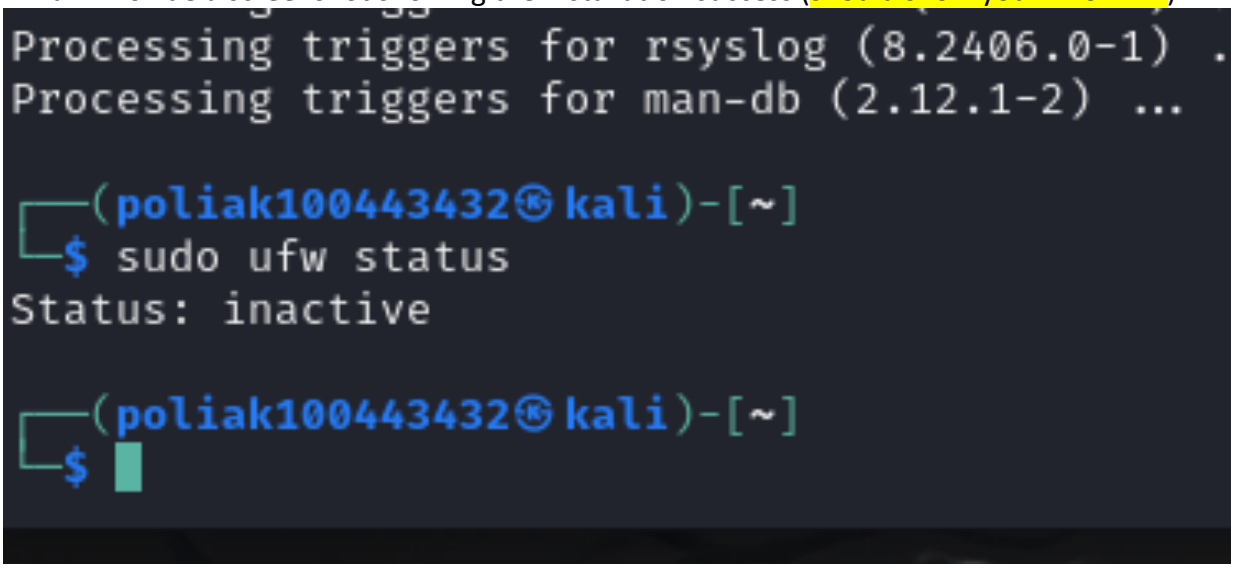


4. [5] Using SSH connect from M1 to M2
   a. Provide a successful SSH connection screenshot from M1 to M2



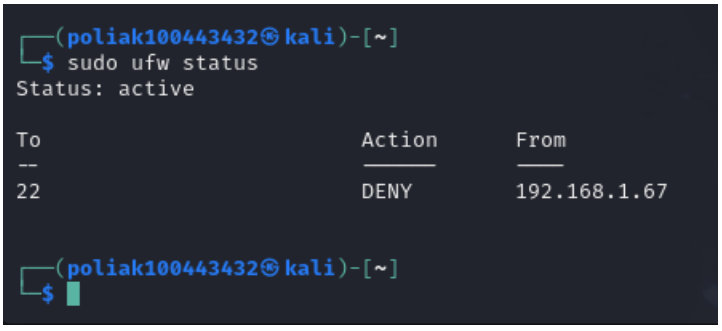
   i. Did you get an error? How did you fix it? – Yes, It says that new user is not in sudoers file. Went to sudo visudo. Add line poliak100443432 ALL=(ALL:ALL) AL. Save it and exit.

5. [5] Install UFW Linux firewall on M2
    a. Provide a screenshot showing the installation success (should show your ID on M2)



6. [5]Using UFW, prevent M1 from accessing M2 using SSH
    a. Provide SSH connection screenshot from M1

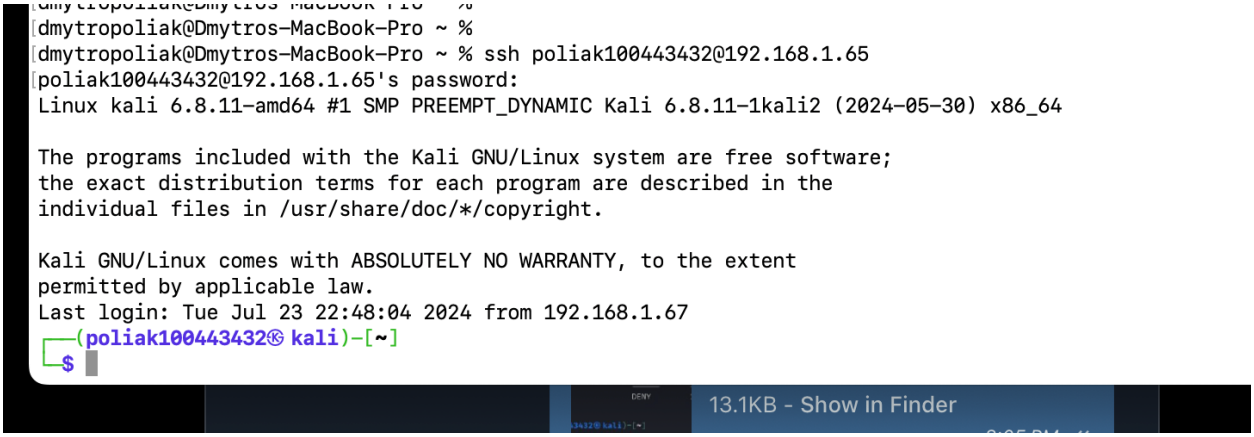
```
dmytropoliak@Dmytros-MacBook-Pro ~ %
dmytropoliak@Dmytros-MacBook-Pro ~ % ssh poliak100443432@192.168.1.65
ssh: connect to host 192.168.1.65 port 22: Operation timed out
dmytropoliak@Dmytros-MacBook-Pro ~ %
```



    b. Write the command you used below?

sudo ufw deny from 192.168.1.67 to any port 22

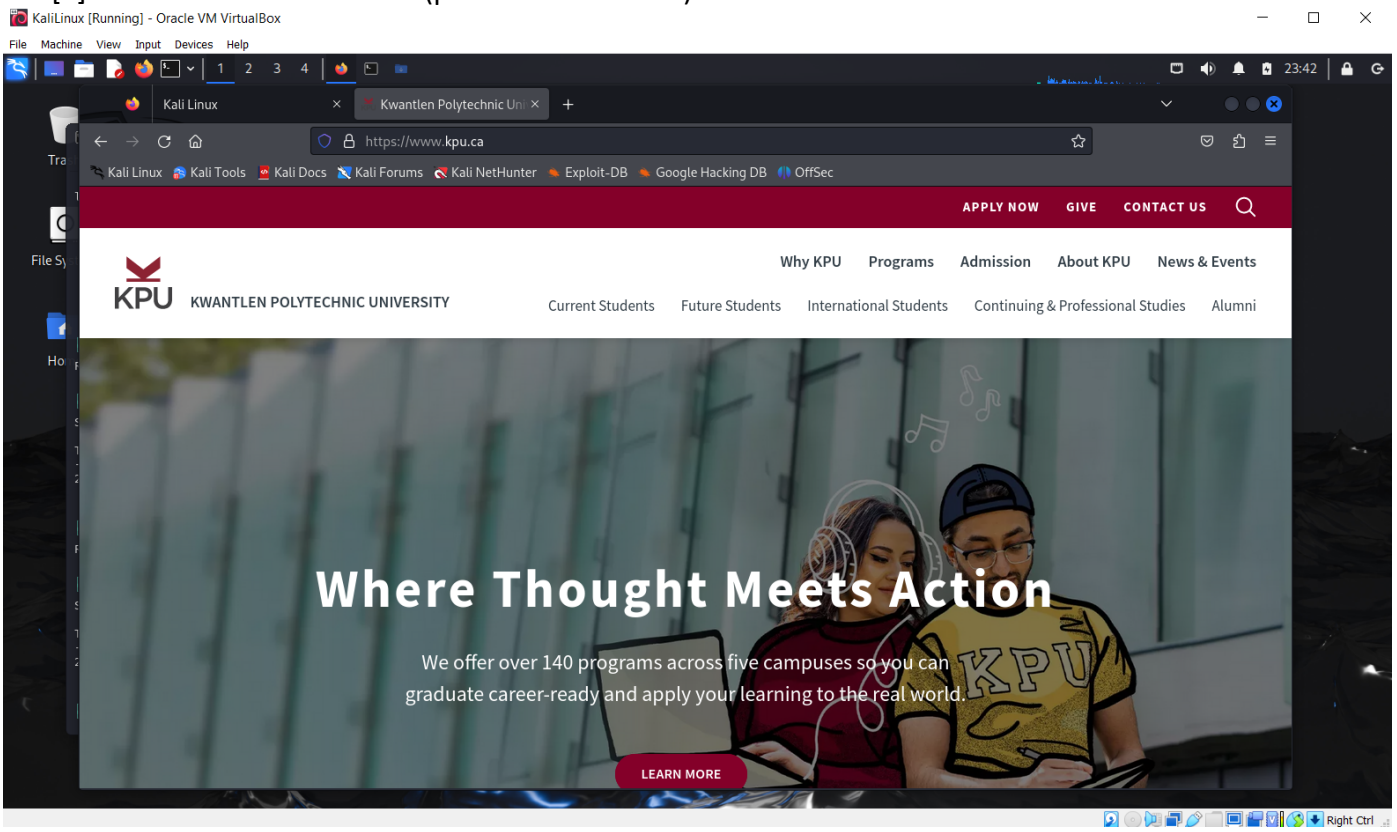
7. [5] Using UFW, allow M1 to access M2 using SSH
    a. Provide SSH connection screenshot from M1



    b. Write the command you used below?

Sudo ufw allow any from 192.168.1.67 to any port 22

8. [5] Access KPU site from M2 (provide screenshot)

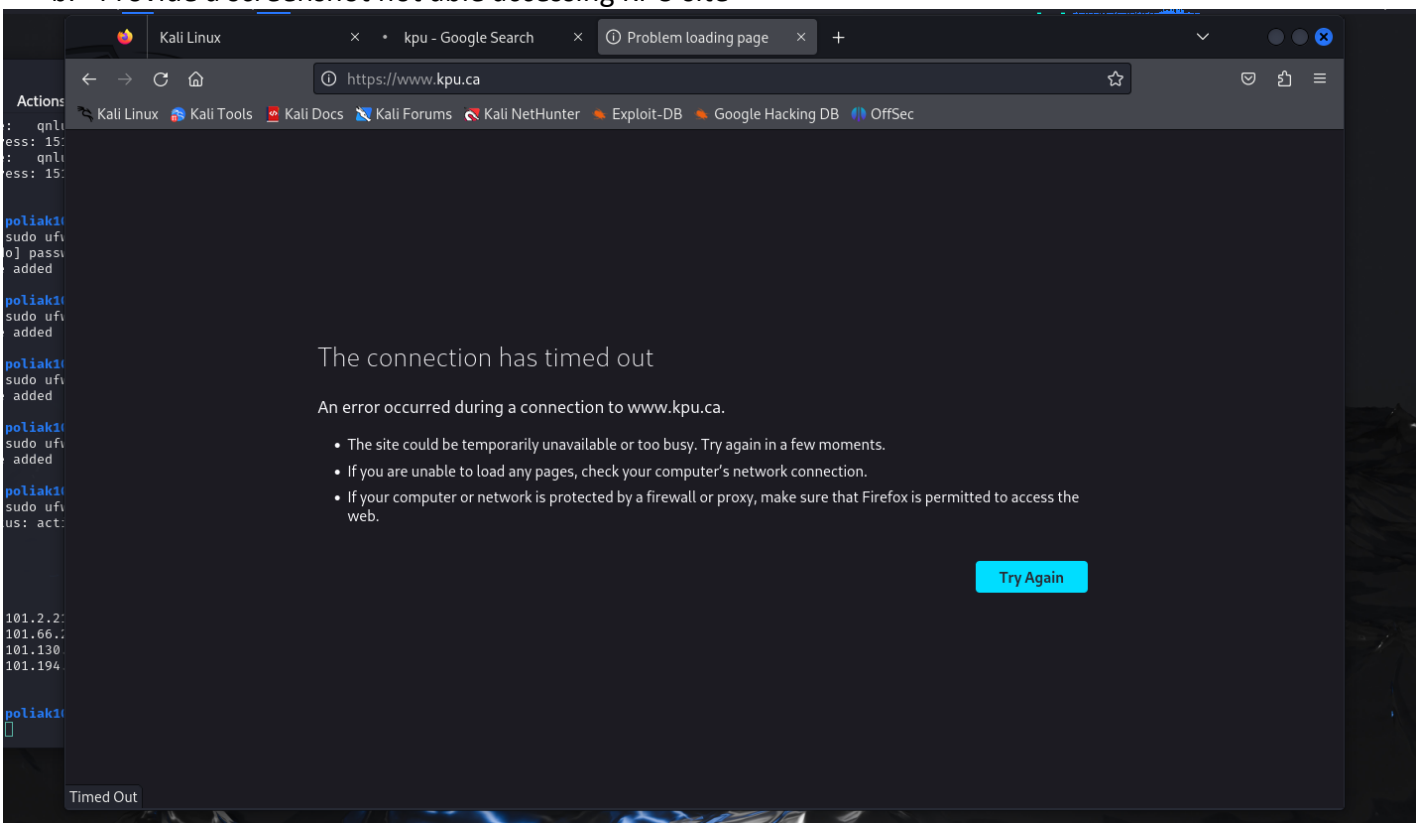


9. [10] Using UFW, prevent M2 from accessing KPU site using HTTPS protocol port
    a. Write the rule you used?

sudo ufw deny out to 151.101.2.216 port 443
sudo ufw deny out to 151.101.66.216 port 443

sudo ufw deny out to 151.101.130.216 port 443
sudo ufw deny out to 151.101.194.216 port 443

b. Provide a screenshot not able accessing KPU site



10. [10] Using UFW, prevent M1 from pinging M2. Your rule should use M1 IP address.
    a. Write the rule you used?

sudo vi /etc/ufw/before.rules

-A ufw-before-input -p icmp --icmp-type echo-request -s 192.168.1.67 -j DROP

    b. Provide screenshot

```
dmytropoliak@Dmytros-MacBook-Pro ~ % ping 192.168.1.65
PING 192.168.1.65 (192.168.1.65): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
Request timeout for icmp_seq 8
Request timeout for icmp_seq 9
^C
--- 192.168.1.65 ping statistics ---
11 packets transmitted, 0 packets received, 100.0% packet loss
dmytropoliak@Dmytros-MacBook-Pro ~ %
```

```
# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines


# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT

# quickly process packets for which we already have a connection
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

-A ufw-before-input -p icmp --icmp-type echo-request -s 192.168.1.67 -j DROP

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT

-A ufw-before-input -p icmp --icmp-type echo-request -s 192.168.1.67 -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -s 192.168.64.1 -j DROP

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
"/etc/ufw/before.rules" 82L, 2850B
```

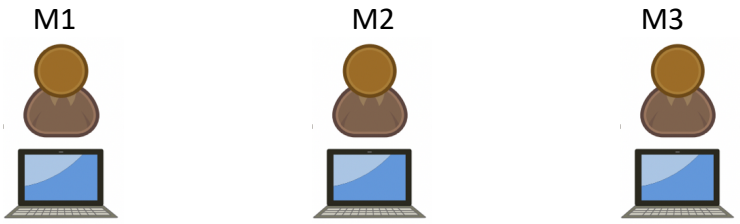
11. [5] Display the status of UFW
    a. Write the command you used?

Sudo ufw status verbose

    b. Provide screenshot

```
┌──(poliak100443432㉿kali)-[~]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    192.168.1.67

151.101.2.216 443          DENY OUT    Anywhere
151.101.66.216 443         DENY OUT    Anywhere
151.101.130.216 443        DENY OUT    Anywhere
151.101.194.216 443        DENY OUT    Anywhere


┌──(poliak100443432㉿kali)-[~]
└─$
```

12. Install a second Kali Linux as a Virtual Guest (M3)



Windows/MacOS. KALI VM1 KALI VM2

13. [10] Configure the firewall on M2 to allow SSH session between M3 and M2 but not M1 and M2
    - Provide 2 screenshots: M3/M2 and M1/M2

M1/M2:

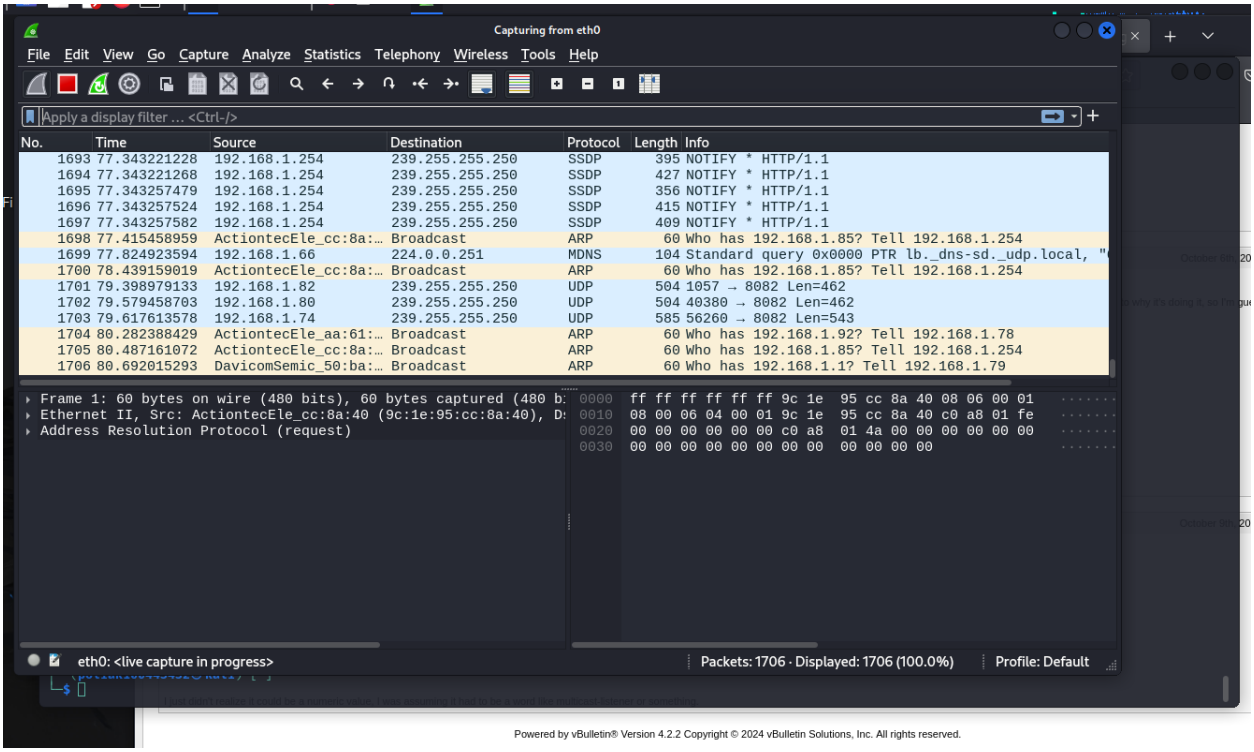
```
dmytropoliak@Dmytros-MacBook-Pro ~ % ssh poliak100443432@192.168.1.65
ssh: connect to host 192.168.1.65 port 22: Operation timed out
dmytropoliak@Dmytros-MacBook-Pro ~ %
```

M3/M2:

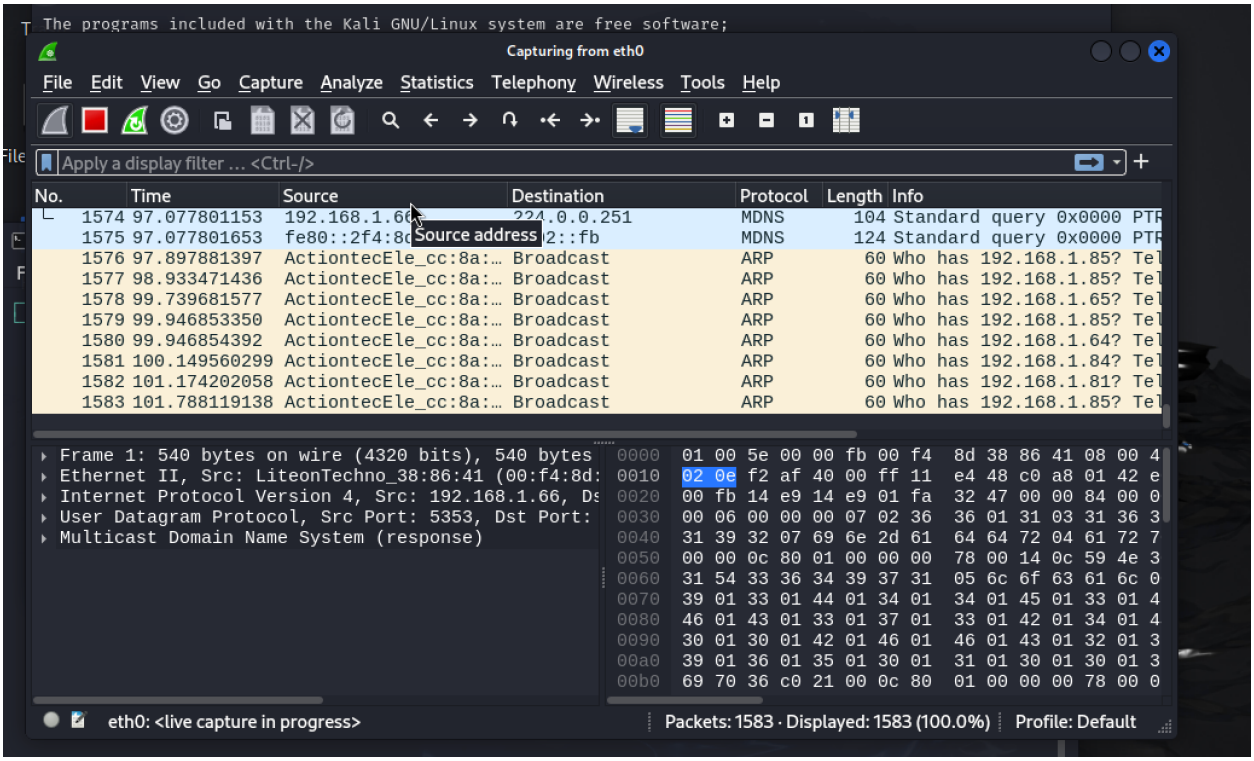
```
┌──(dmytropoliak㉿kali)-[~]
└─$ ssh poliak100443432@192.168.1.65
The authenticity of host '192.168.1.65 (192.168.1.65)' can't be established.
ED25519 key fingerprint is SHA256:tIs14XLtJv2Ry29rcesjID7bVRr5PFXm24m61hNRr8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.65' (ED25519) to the list of known hosts.
poliak100443432@192.168.1.65's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 24 19:37:02 2024 from 192.168.1.64
┌──(poliak100443432㉿kali)-[~]
└─$
```

14. Run Wireshark on both M2 & M3 to capture the network packets
    a. Provide screenshot showing the packets capture

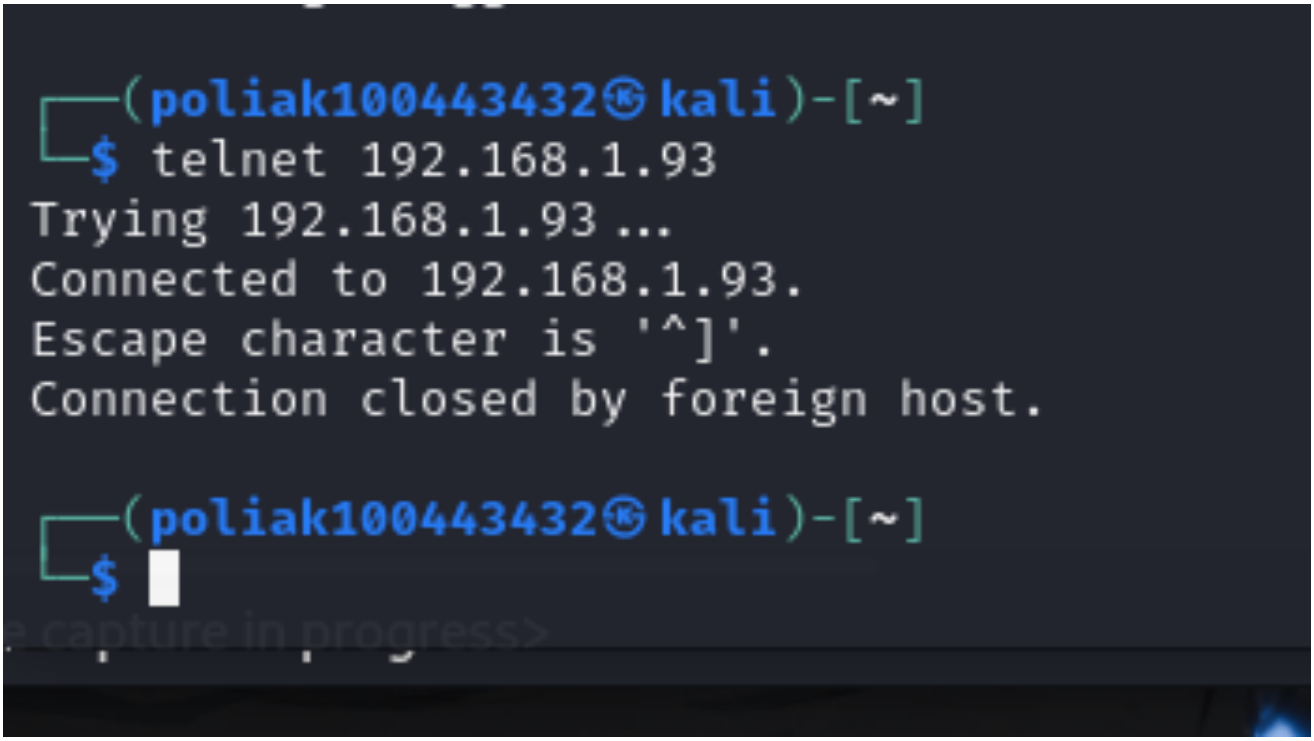
M2:



M3:



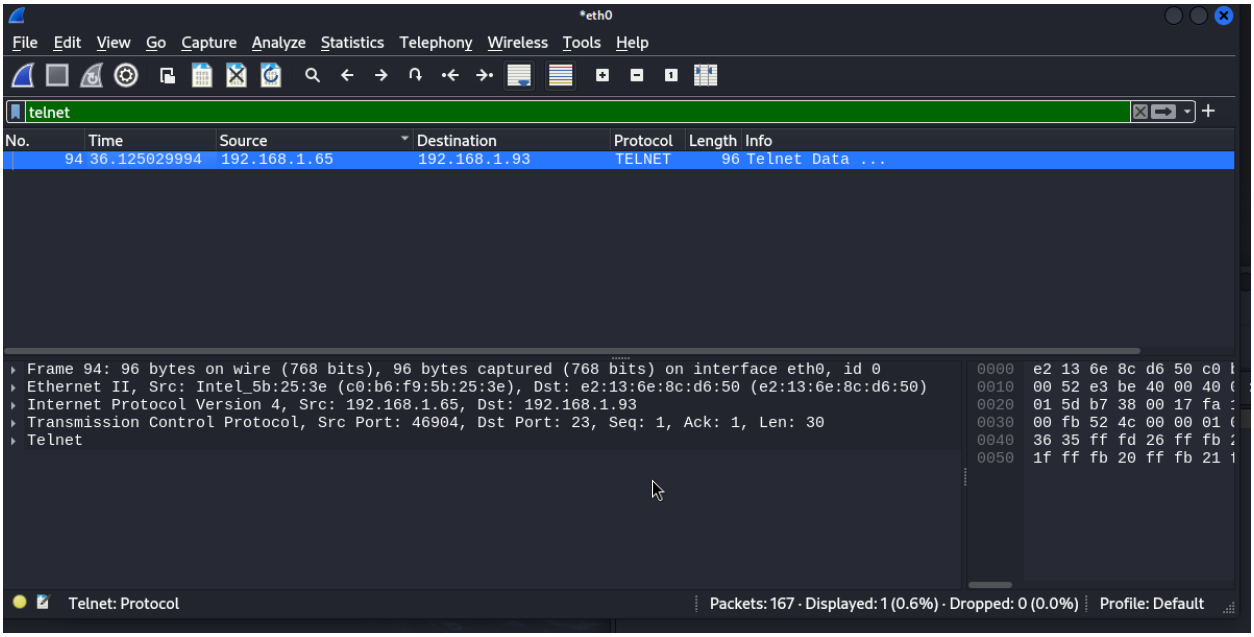
15. [10] While in M2, use telnet (not stelnet) command to access M3
    b. Provide a screenshot of successful connection



16. Stop Wireshark

17. [20] Trace the telnet session captured packets by hand:
    c. Copy all the packets in sequence for the telnet session only (discard other packets)
        i. Paste the telnet session packets only here

2. 0000 c8 89 f3 c5 88 76 08 00 27 15 fe 9e 08 00 45 00
3. 0010 00 52 e3 be 40 00 40 06 d2 f8 c0 a8 01 41 c0 a8
4. 0020 01 5d b7 38 00 17 fa 12 85 e0 e5 45 b5 0e 80 18
5. 0030 00 fb 84 33 00 00 01 01 08 0a cf 07 04 d3 45 44
6. 0040 36 35 ff fd 26 ff fb 26 ff fd 03 ff fb 18 ff fb
7. 0050 1f ff fb 20 ff fb 21 ff fb 22 ff fb 27 ff fd 05



    a. Show the plain text password you used to access M3
        i. Provide a screenshot of the packet that show the password