PROJECT REPORT

CS Project 1- February 24, 2023

Group: Techies

Vooha sree Chitta (A05306488)

Charishma Maganti (A05317920)

D. Dhanush Narasimha Reddy (A05307038)

Section 1: Build Sandbox:

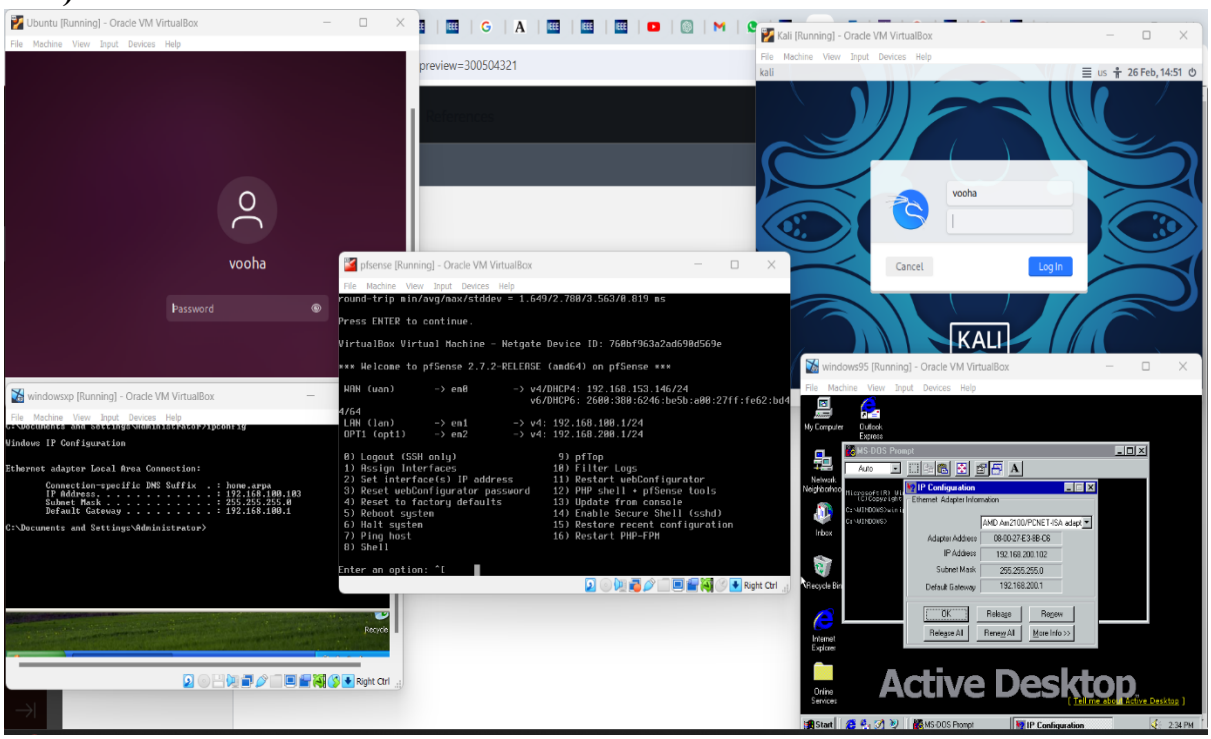
Introduction:

The primary objective of this project is to create a sandbox network using virtualization technologies and implement security policies to enforce a secure environment. Through this project, we aim to:

- Build a sandbox network using virtual machines and a virtual router. Download necessary software (VirtualBox, VMware Workstation, pfSense Router, Server ISO files). Configure a network with a router and four machines, divided into two internal networks (Network A and Network B).
- Conduct network checks and diagnosis to ensure proper configuration and functionality.
- Implement a comprehensive security policy on the network router to enforce security measures.
- Test the implementation of the security policy and analyze the results.
- Provide a detailed report summarizing the setup, implementation, testing, and findings.

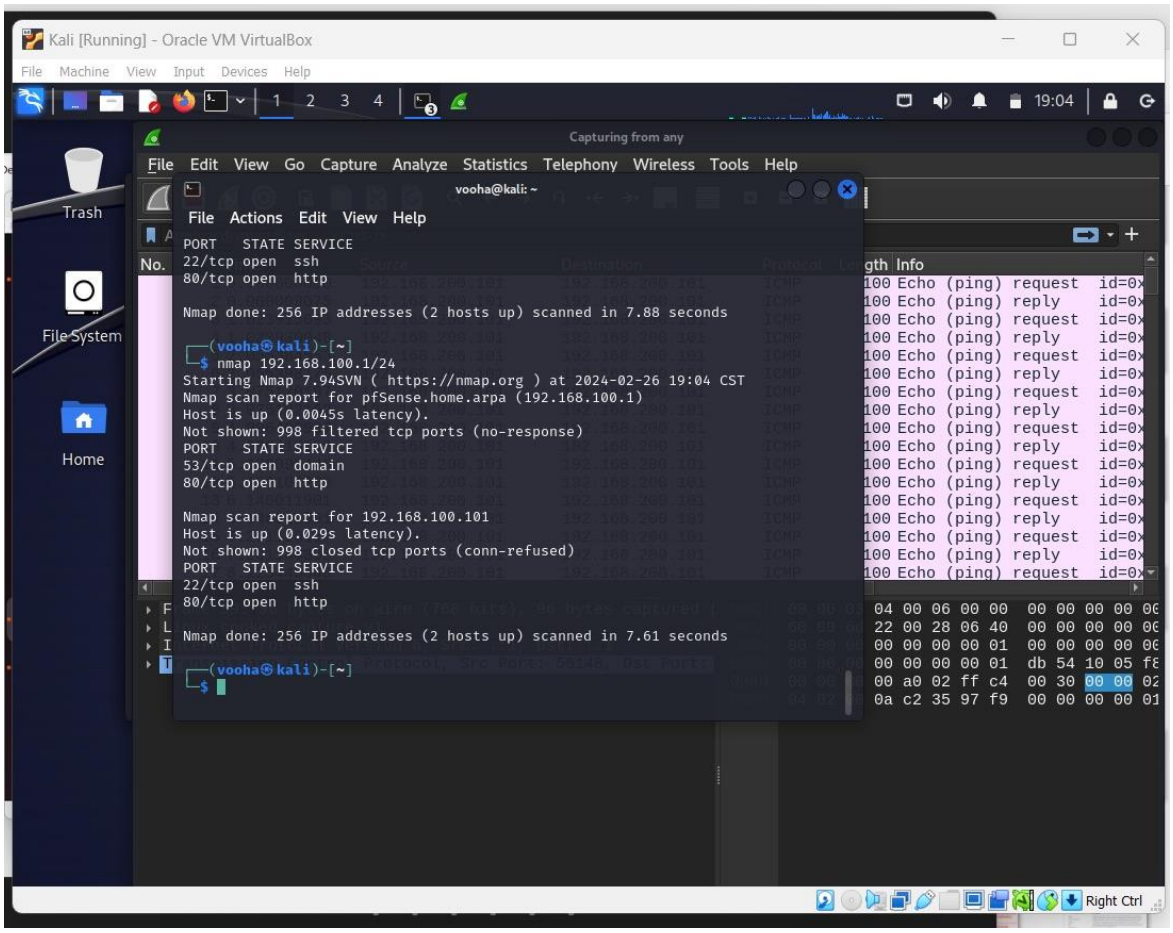
Section 2: (Task 2 and 3):

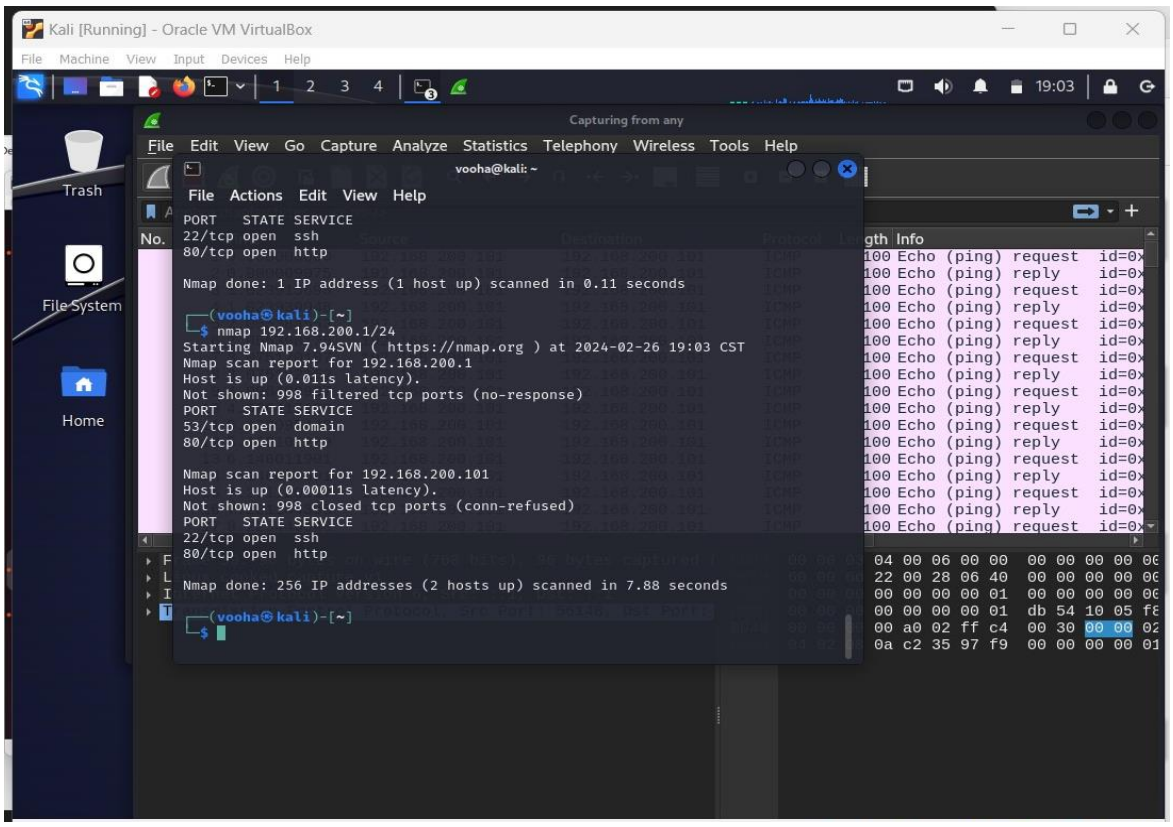
i) Screenshots of the VM's:



ii) Nmap Commands used:

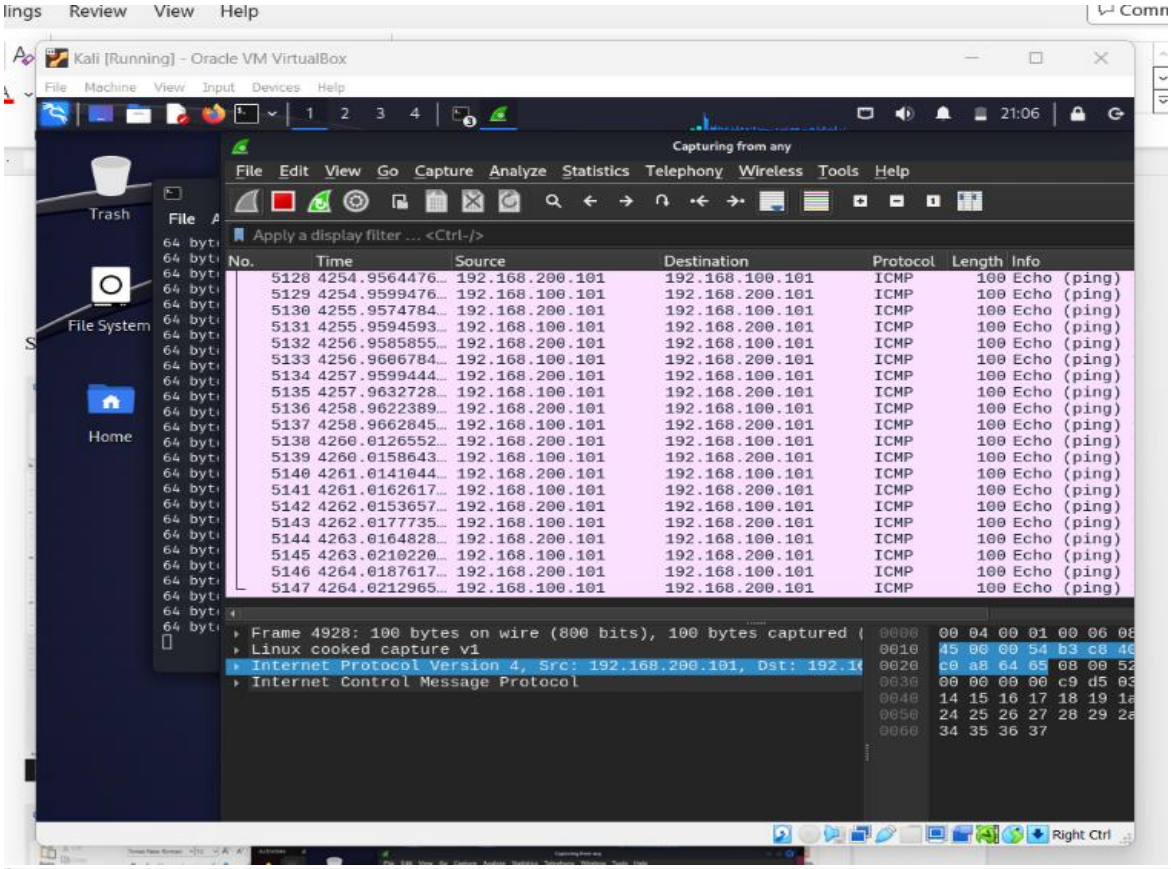
nmap 192.168.100.1/24
nmap 192.168.200.1/24

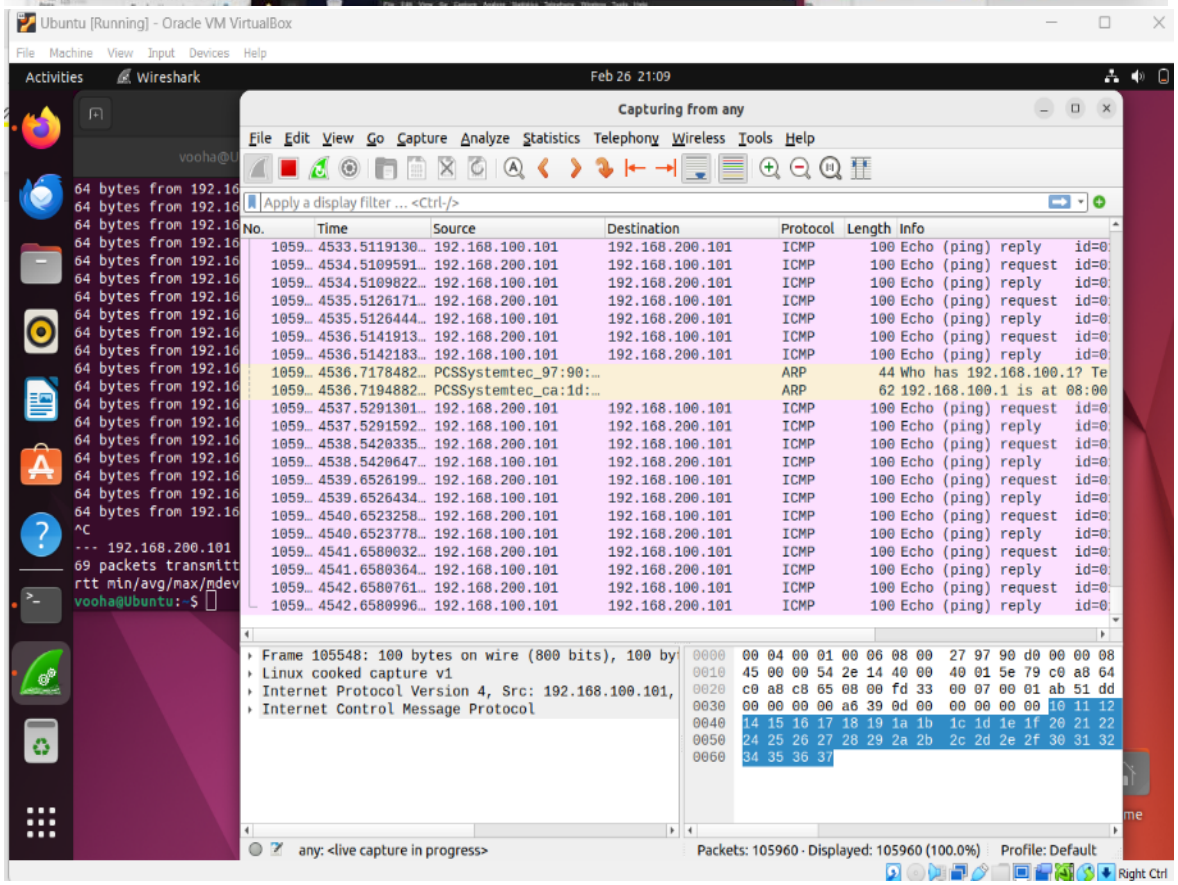

Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
19:04
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
vooha@kali: ~
File Actions Edit View Help
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.88 seconds
(vooha㉿kali)-[~]
$ nmap 192.168.100.1/24
Starting Nmap 7.94SVN (https://nmap.org) at 2024-02-26 19:04 CST
Nmap scan report for pfSense.home.arpa (192.168.100.1)
Host is up (0.0045s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT STATE SERVICE
53/tcp open domain
80/tcp open http
Nmap scan report for 192.168.100.101
Host is up (0.029s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.61 seconds
(vooha㉿kali)-[~]
$
Trash
File System
Home
Right Ctrl


Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
19:03
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
vooha@kali: ~
File Actions Edit View Help
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
(vooha㉿kali)-[~]
$ nmap 192.168.200.1/24
Starting Nmap 7.94SVN (https://nmap.org) at 2024-02-26 19:03 CST
Nmap scan report for 192.168.200.1
Host is up (0.011s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT STATE SERVICE
53/tcp open domain
80/tcp open http
Nmap scan report for 192.168.200.101
Host is up (0.00011s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.88 seconds
(vooha㉿kali)-[~]
$
Trash
File System
Home

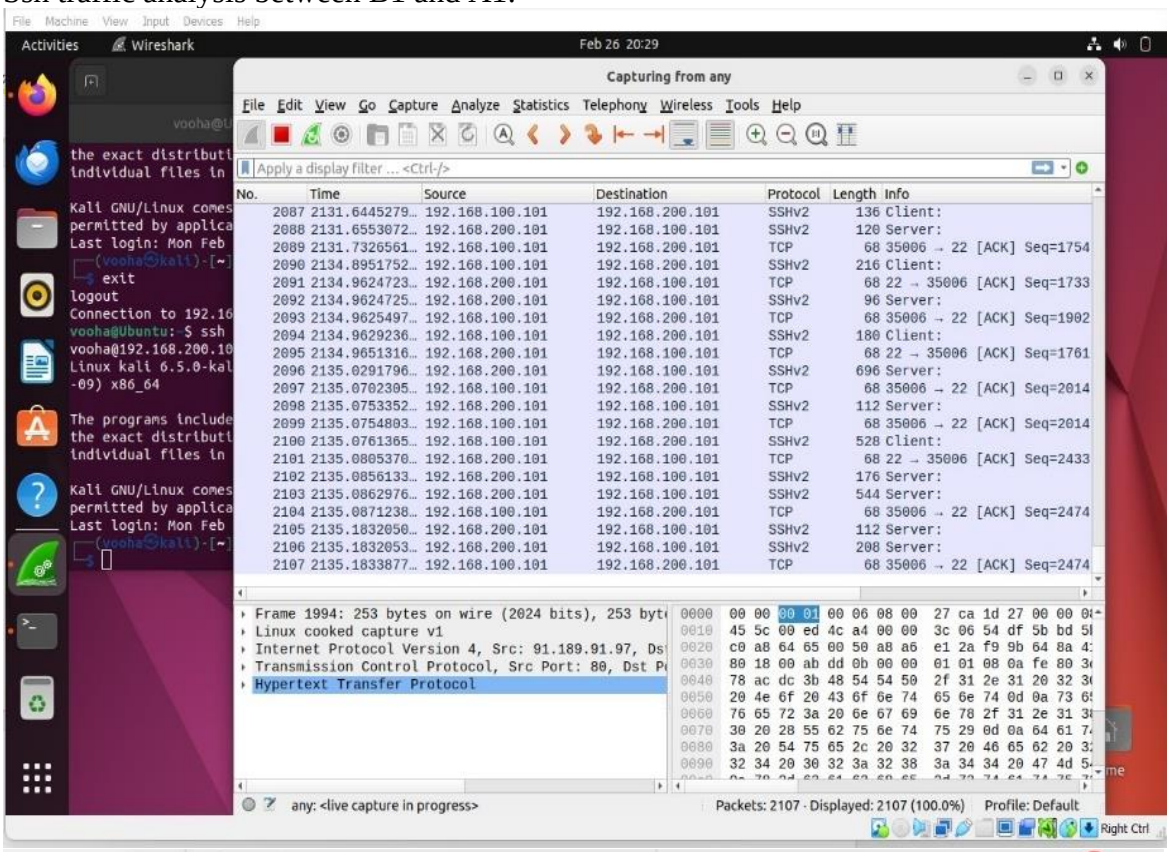
iii. Show screen shots of Wireshark results from/to systems checking the web the services between Task-III.2 before Task-IV, implementing security.

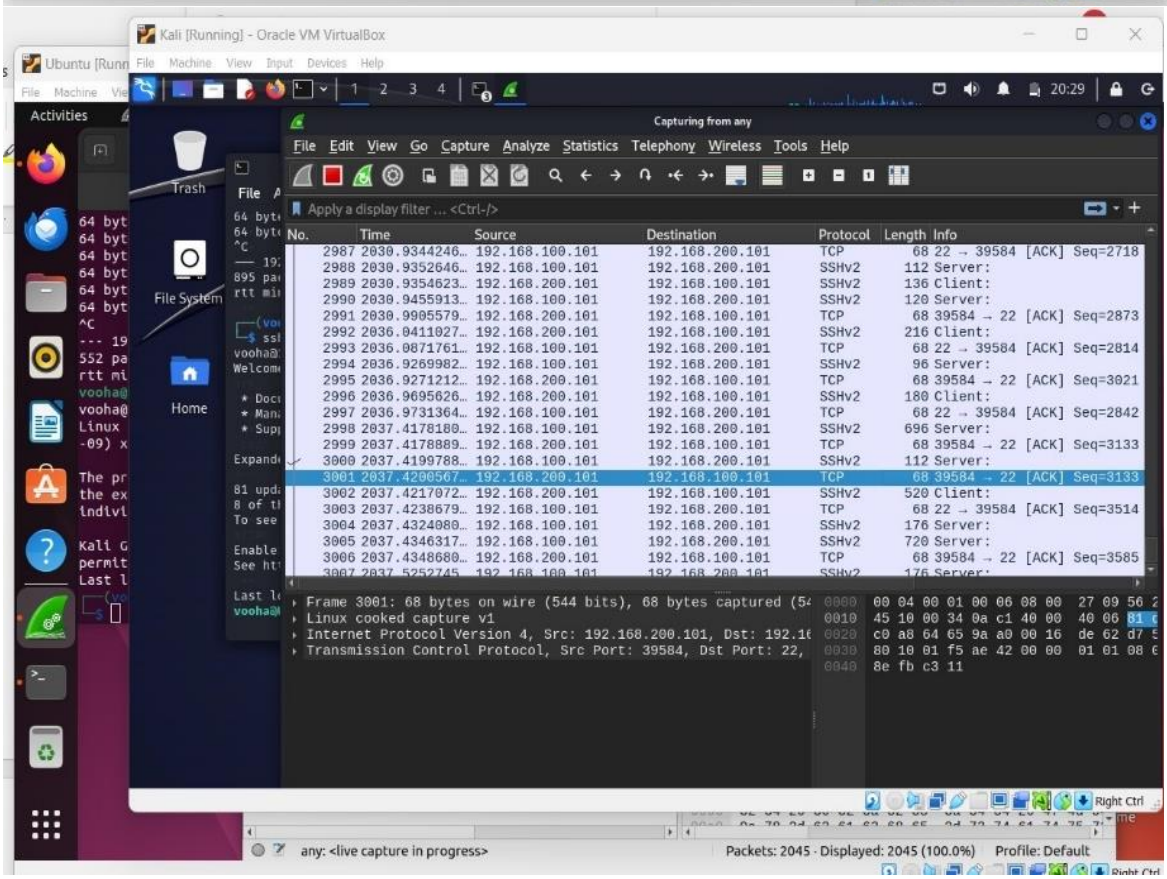
a) Ping traffic analysis between B1 and A1:



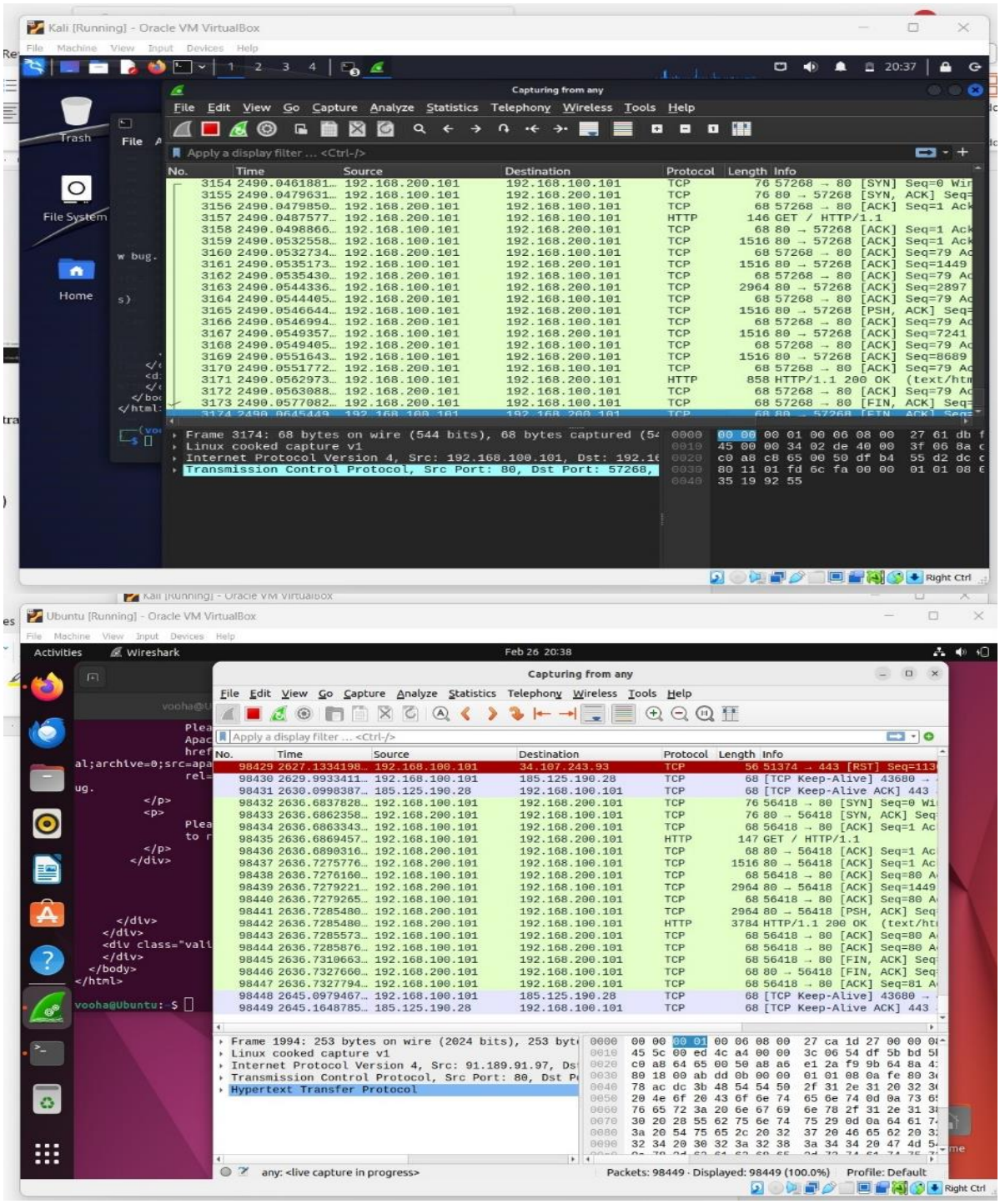


Ssh traffic analysis between B1 and A1:

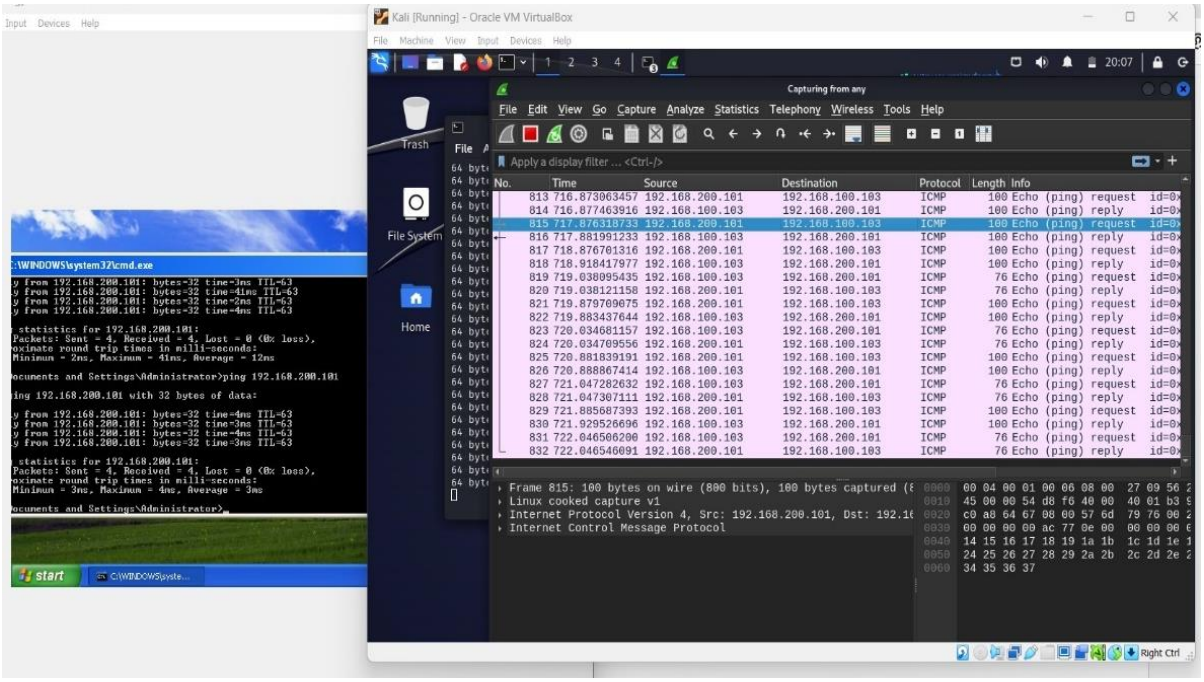




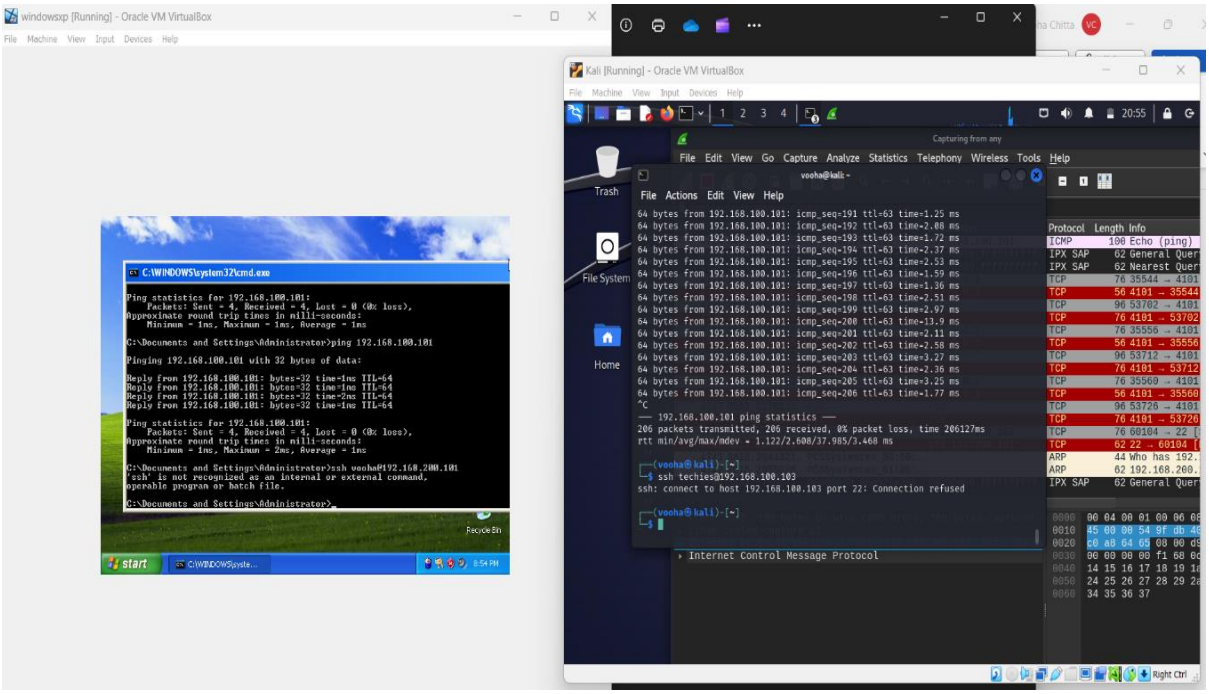
Curl traffic analysis between B1 and A1:



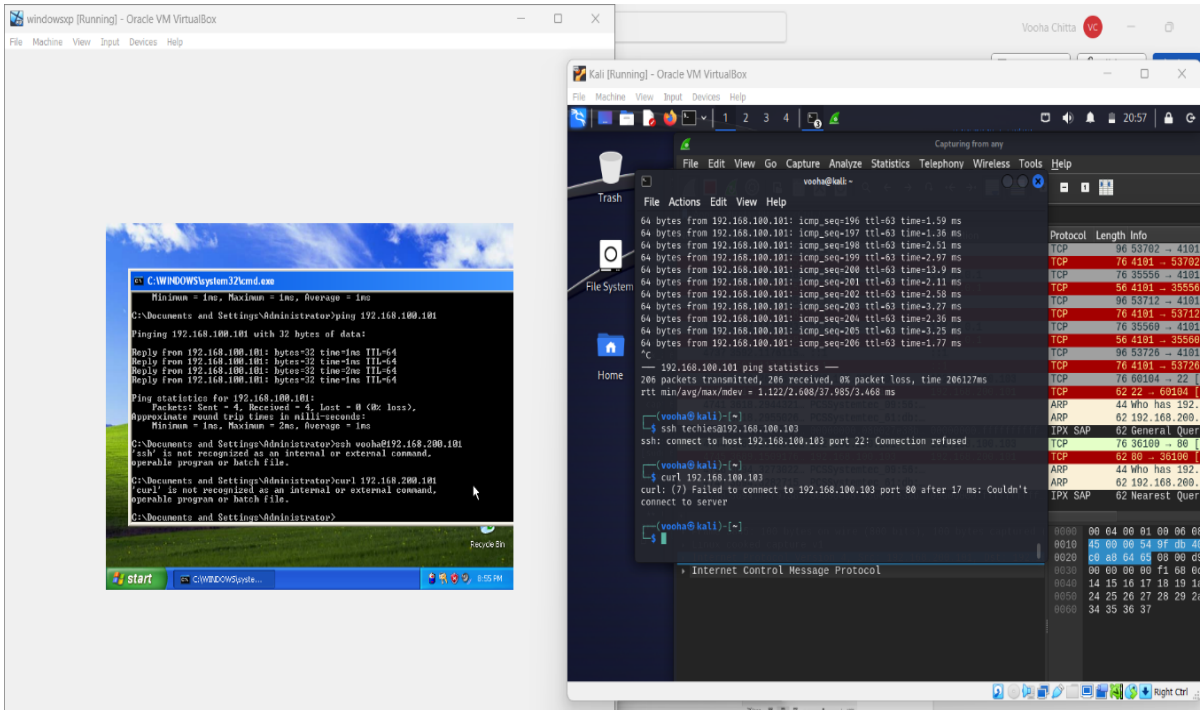
b) Ping traffic analysis between B1 and A2:



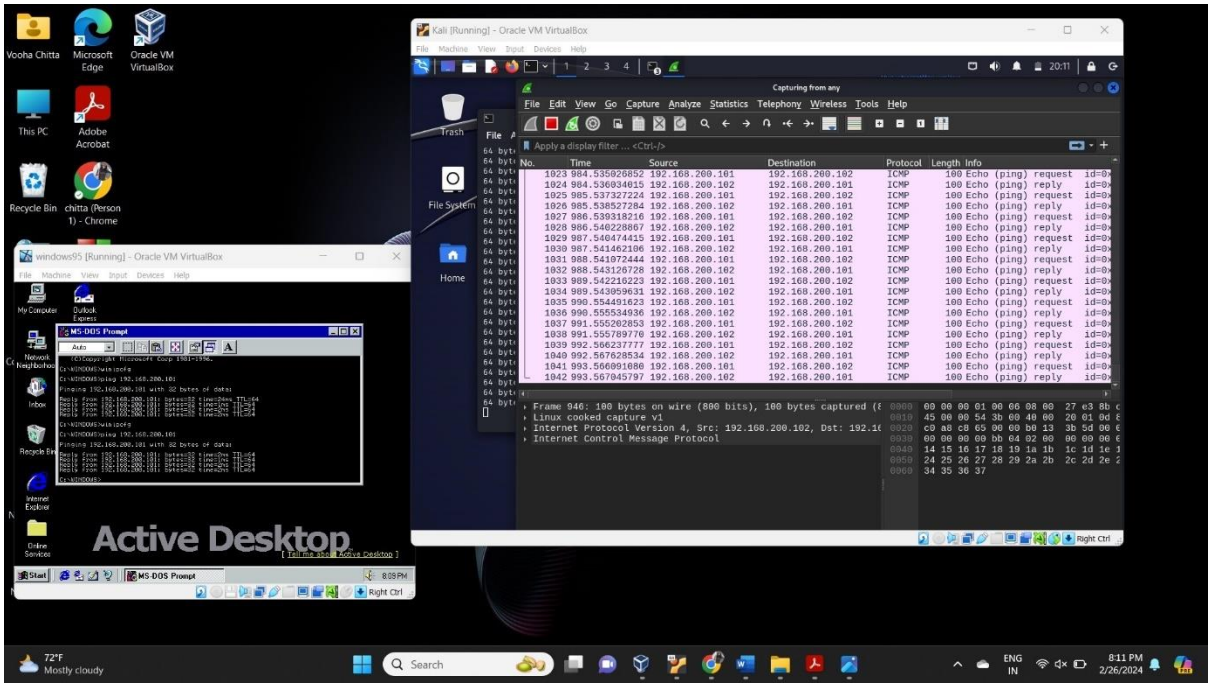
Ssh between B1 and A2:



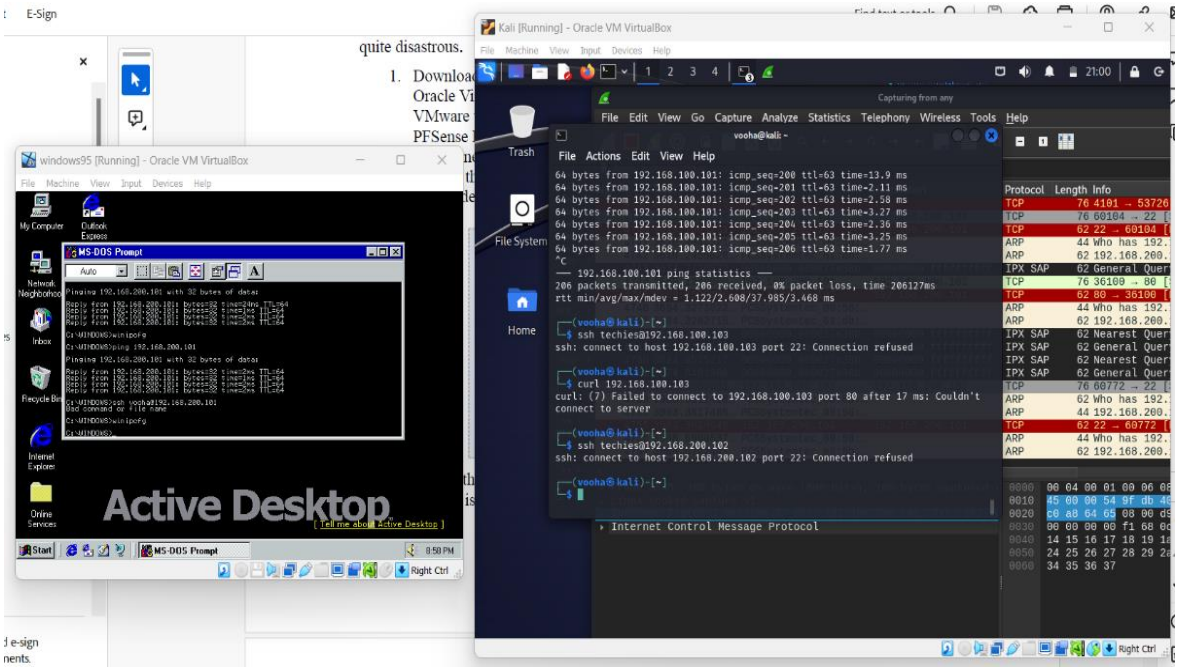
Curl between B1 and A2:



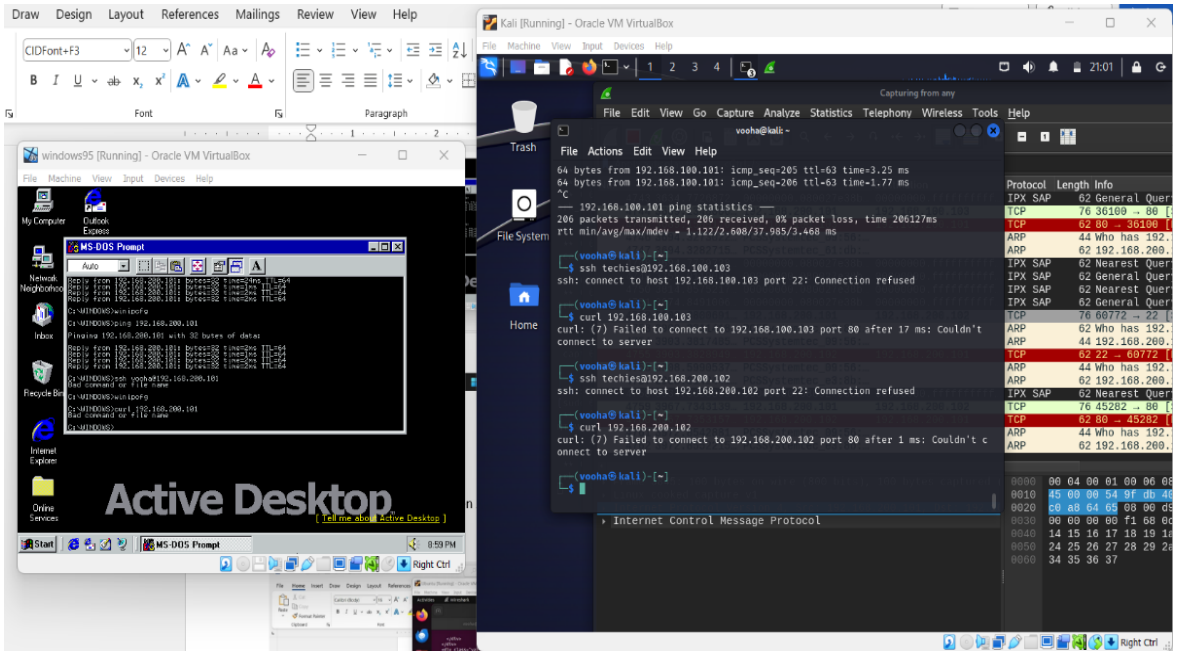
c) Ping traffic analysis between B1 and B2:



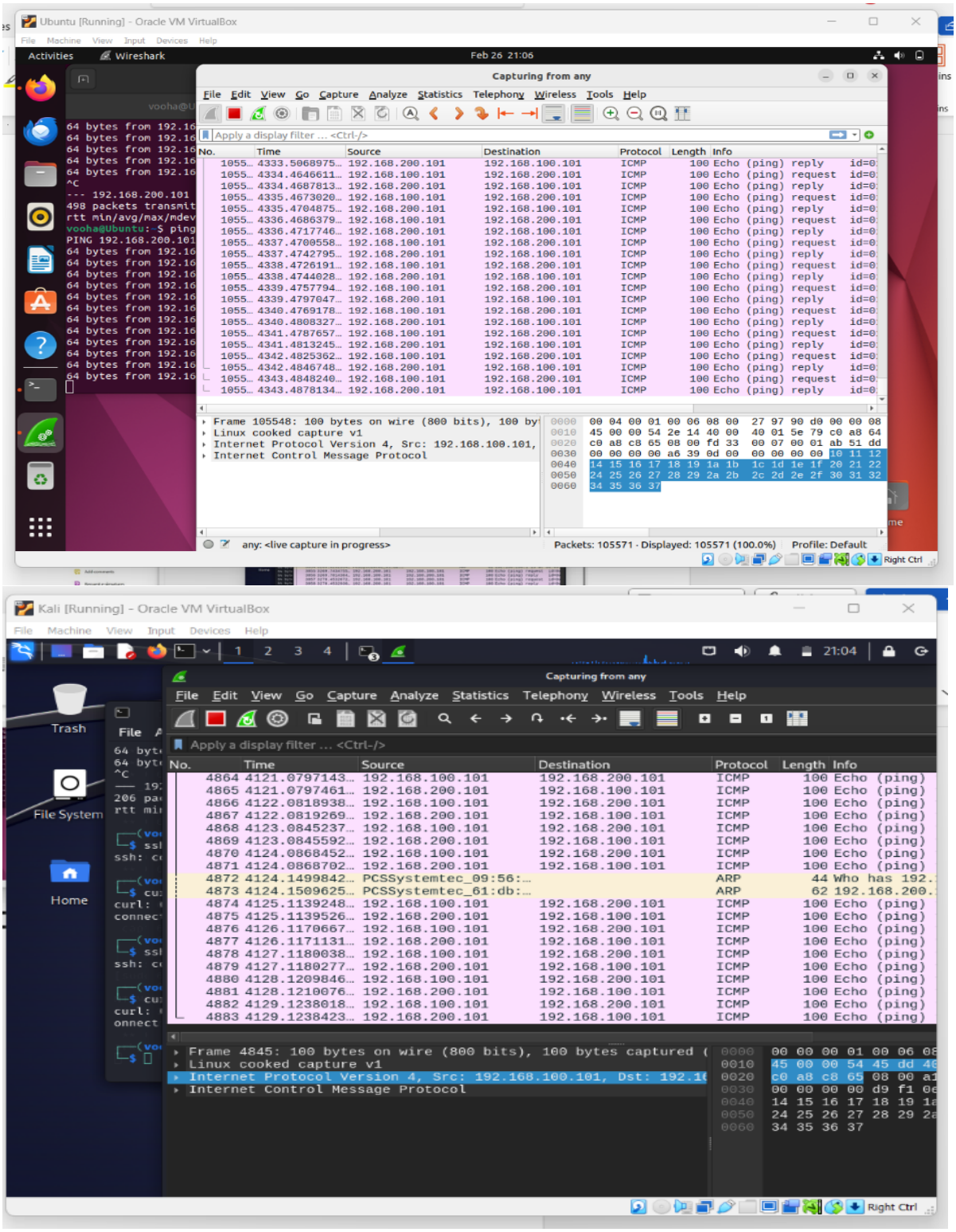
Ssh between B1 and B2:



Curl between B1 and B2:



d) Ping traffic analysis between A1 to B1:



iv. Web services are available are as follows:

The company server (A.1) provides only web service to external computers in Network B.
The company server (A.1) provides both SSH and web services to the workstations (A.2) in Network A.
Workstations (A.2) do not provide any services.
Workstations (A.2) can access services hosted by the server (A.1).

Workstations (A.2) can only access the web service provided by external computers in Network B.

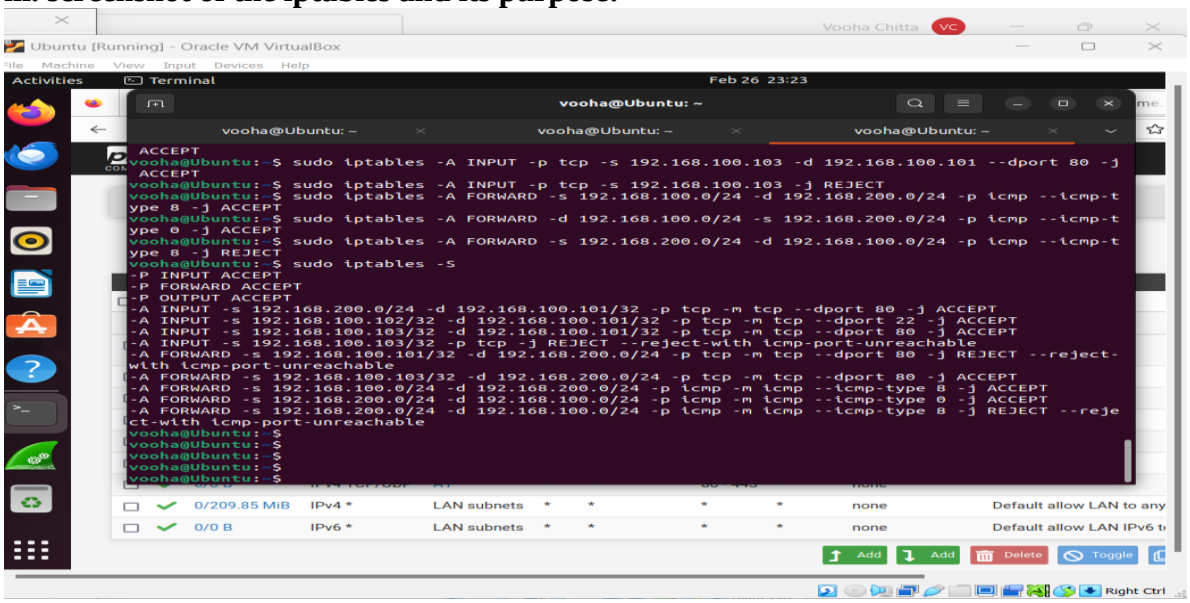
d)Section 3 (Task 4 and 5):

i. ACM:

Access Control Matrix			
Legend: * - anything to anywhere; I - Input; O - Output			
Sources/ Destinations	**Server (A.1)**	**Work Stations (A.2)**	**External Computers (B.1, B.2)**
Server (A.1)	*(*,*)	A(22,O) A(80,O)	A(80,O) D(*,I) A(ICMP,O)
Work Stations (A.2)	A(80,I) A(22,I) D(*,O)	*(*,*)	A(80,I) D(*,O) A(ICMP,O)
External Computers (B.1, B.2)	D(ICMP,O)	D(ICMP,O)	*(*,*)

ii. The policy 'b' between the servers and the workstations cannot be completely enforced by the iptables of R, because the router can't interfere with in an internal network.

iii. screenshot of the iptables and its purpose:



#Accepting the incoming http requests to the server
sudo iptables -A INPUT -p tcp -s 192.168.200.0/24 -d 192.168.100.101 --dport 80 -j ACCEPT

#The Server can't access external subnet http services
sudo iptables -A FORWARD -p tcp -s 192.168.100.101 -d 192.168.200.0/24 --dport 80 -j REJECT

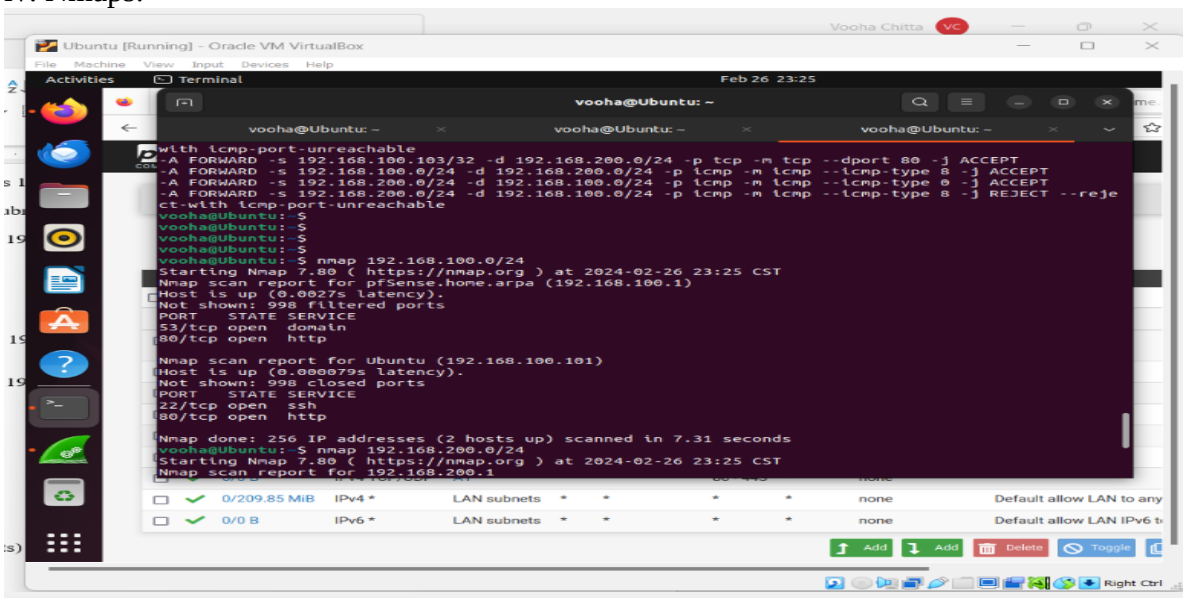
# Workstations can access external subnet services
sudo iptables -A FORWARD -p tcp -s 192.168.100.103 -d 192.168.200.0/24 --dport 80 -j ACCEPT

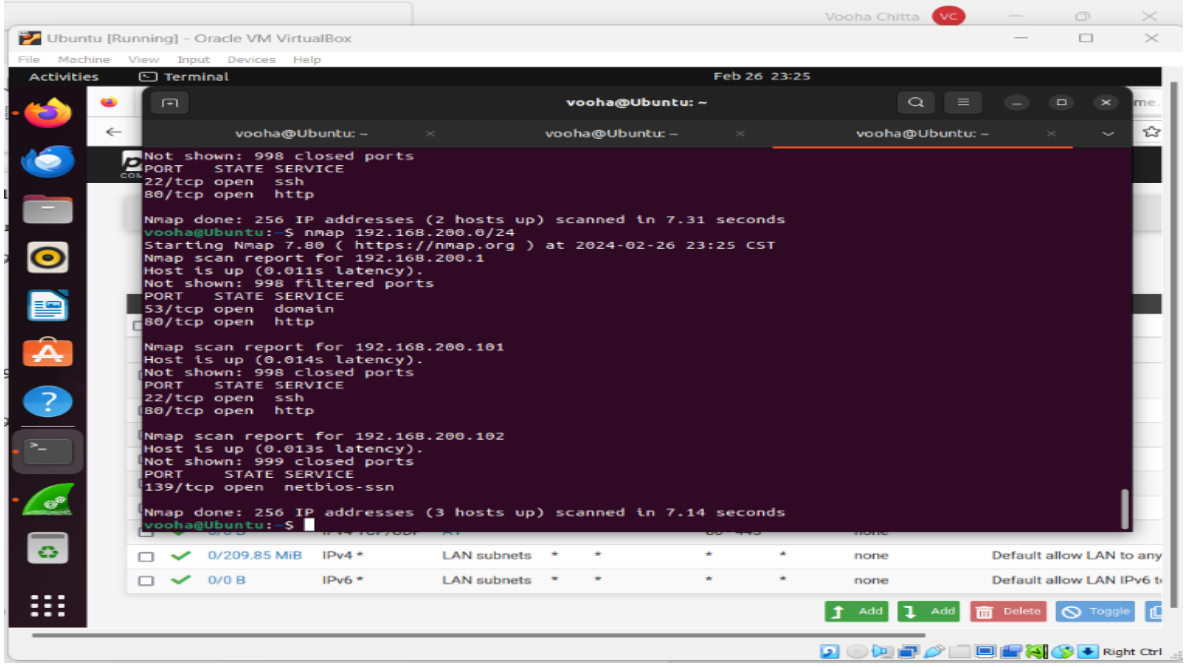
# Allowing workstations to access the server's services
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -d 192.168.100.101 --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -d 192.168.100.101 --dport 80 -j ACCEPT

# Blocking all webservice requests to workstations
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -j REJECT

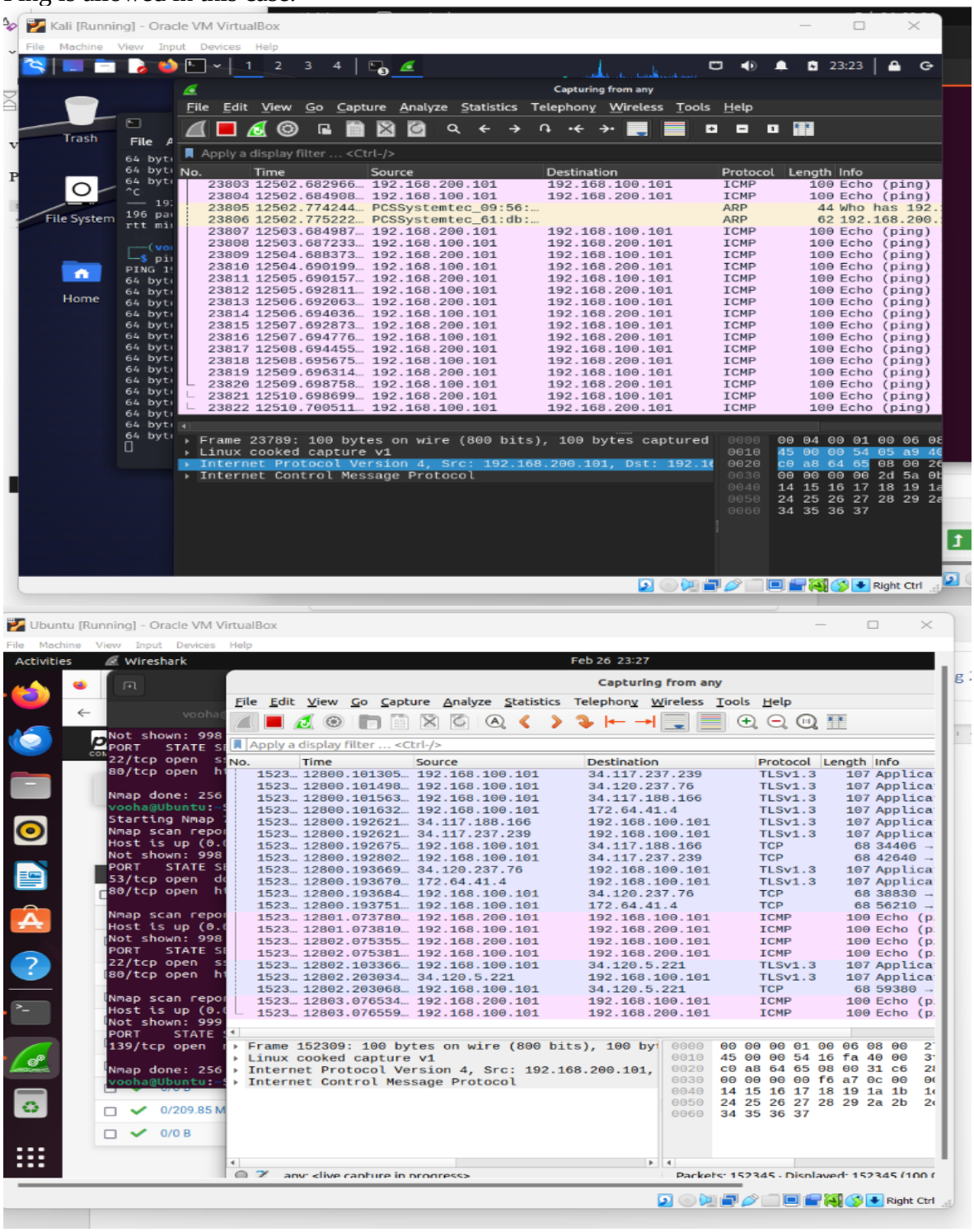
# Accept a one way ping from subnet A to subnet B
sudo iptables -A FORWARD -s 192.168.100.0/24 -d 192.168.200.0/24 -p icmp --icmp-type 8 -j ACCEPT
sudo iptables -A FORWARD -d 192.168.100.0/24 -s 192.168.200.0/24 -p icmp --icmp-type 0 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.200.0/24 -d 192.168.100.0/24 -p icmp --icmp-type 8 -j REJECT

iv. Nmaps:

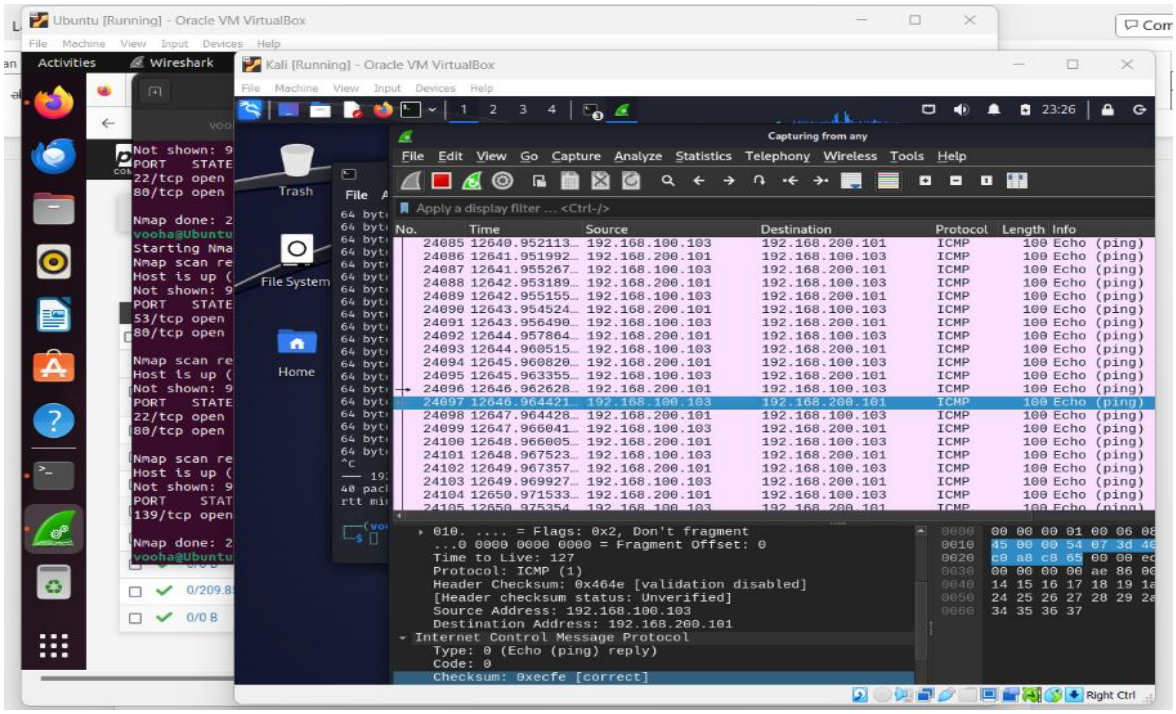




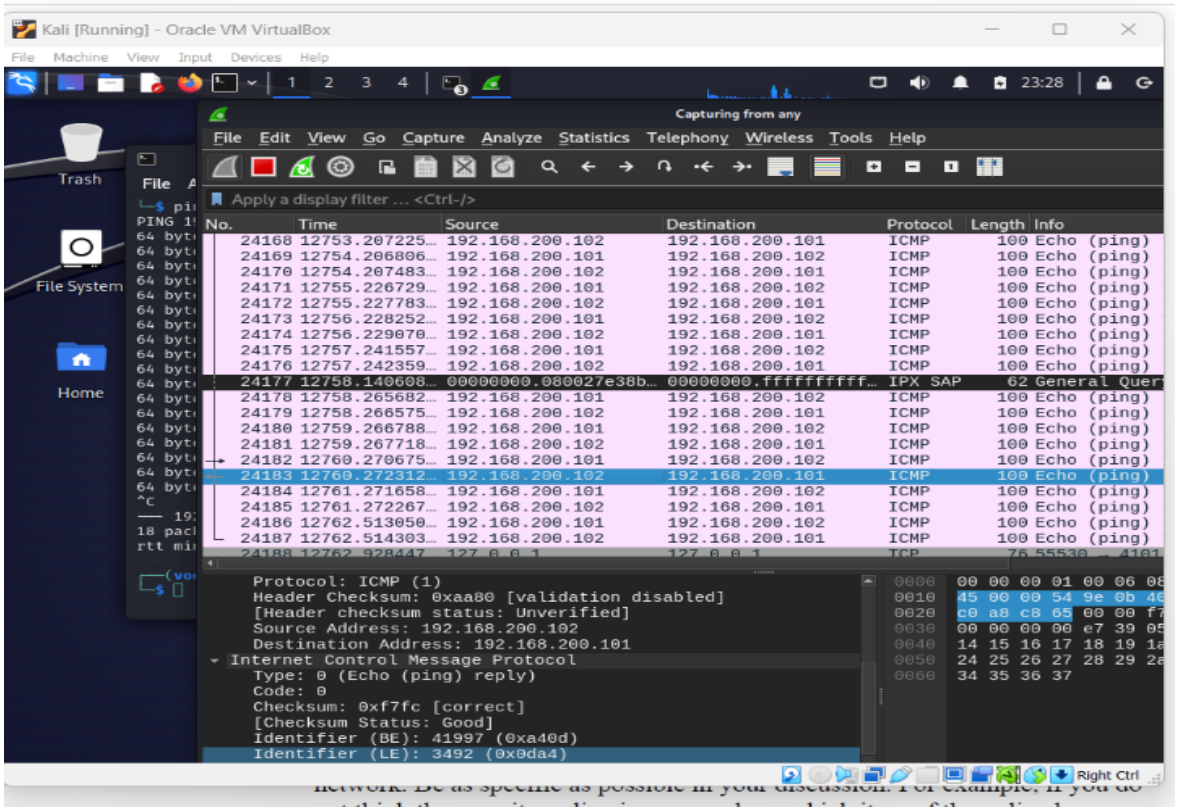
v. Wireshark results (screen shots) of routines checking the web service between B.1 and A.1 Ping is allowed in this case.



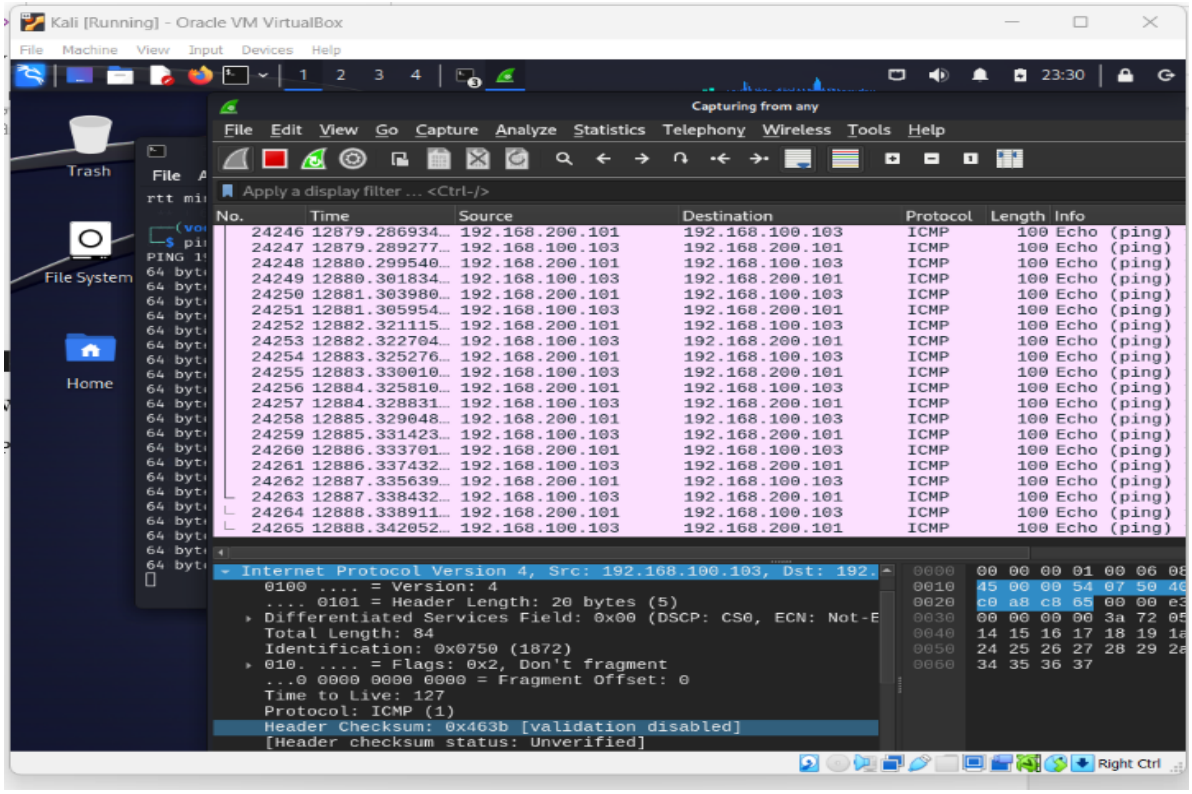
Wireshark results (screen shots) of routines checking the web service between B.1 and A.2: Ping is allowed between B1 and A2:



Wireshark results (screen shots) of routines checking the web service between B.1 and B.2: Ping is allowed between B1 and B2:

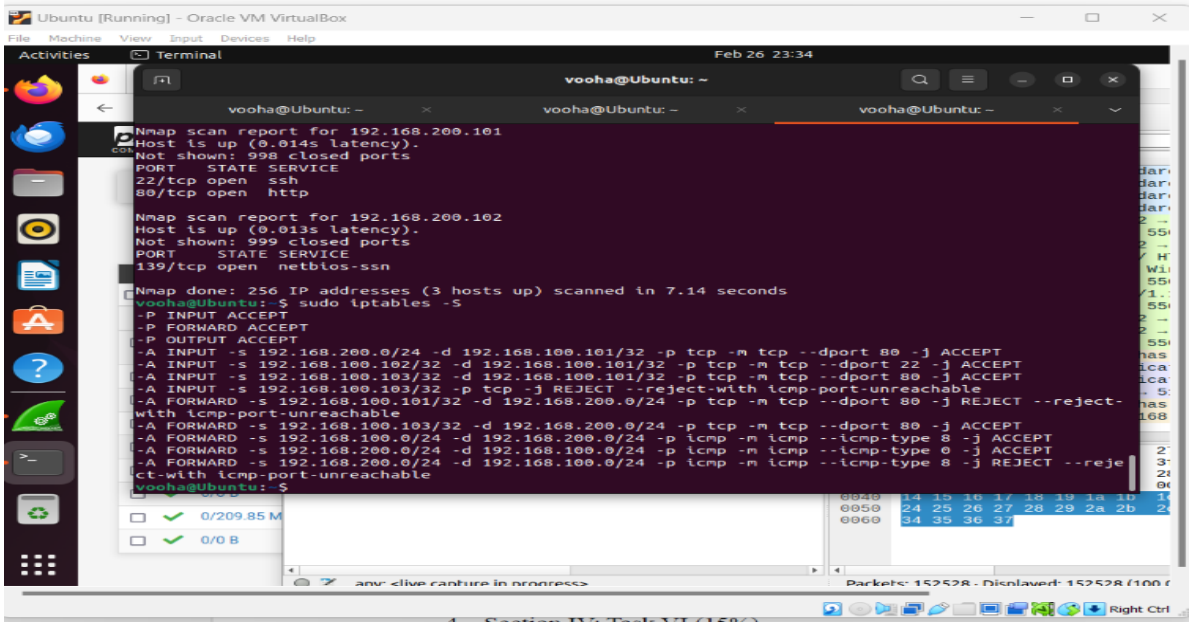


Wireshark results (screen shots) of routines checking the web service between B.1 and A.2: Ping is allowed between B1 and A2:



e) Section 4: (Task 6):

i. screen shot of the local A.1 router configuration rules and its the purpose:



#Delete the rules of the chain
Sudo iptables -F

#Deletes the chain
Sudo iptables -X

#To set default chain policies to incoming, outgoing and forward requests
Sudo iptables -P INPUT ACCEPT

Sudo iptables -P FORWARD ACCEPT
Sudo iptables -P OUTPUT ACCEPT

#Accepting the incoming http requests to the server
sudo iptables -A INPUT -p tcp -s 192.168.200.0/24 -d 192.168.100.101 --dport 80 -j ACCEPT

#The Server can't access external subnet http services
sudo iptables -A FORWARD -p tcp -s 192.168.100.101 -d 192.168.200.0/24 --dport 80 -j REJECT

# Workstations can access external subnet services
sudo iptables -A FORWARD -p tcp -s 192.168.100.103 -d 192.168.200.0/24 --dport 80 -j ACCEPT

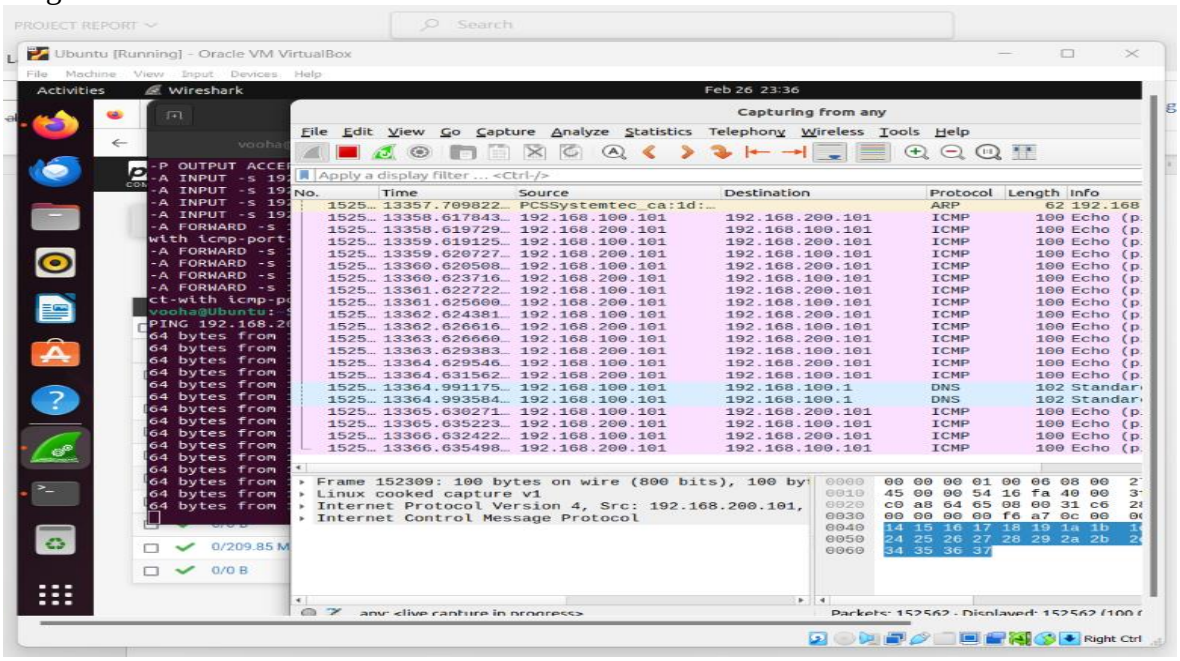
# Allowing workstations to access the server's services
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -d 192.168.100.101 --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -d 192.168.100.101 --dport 80 -j ACCEPT

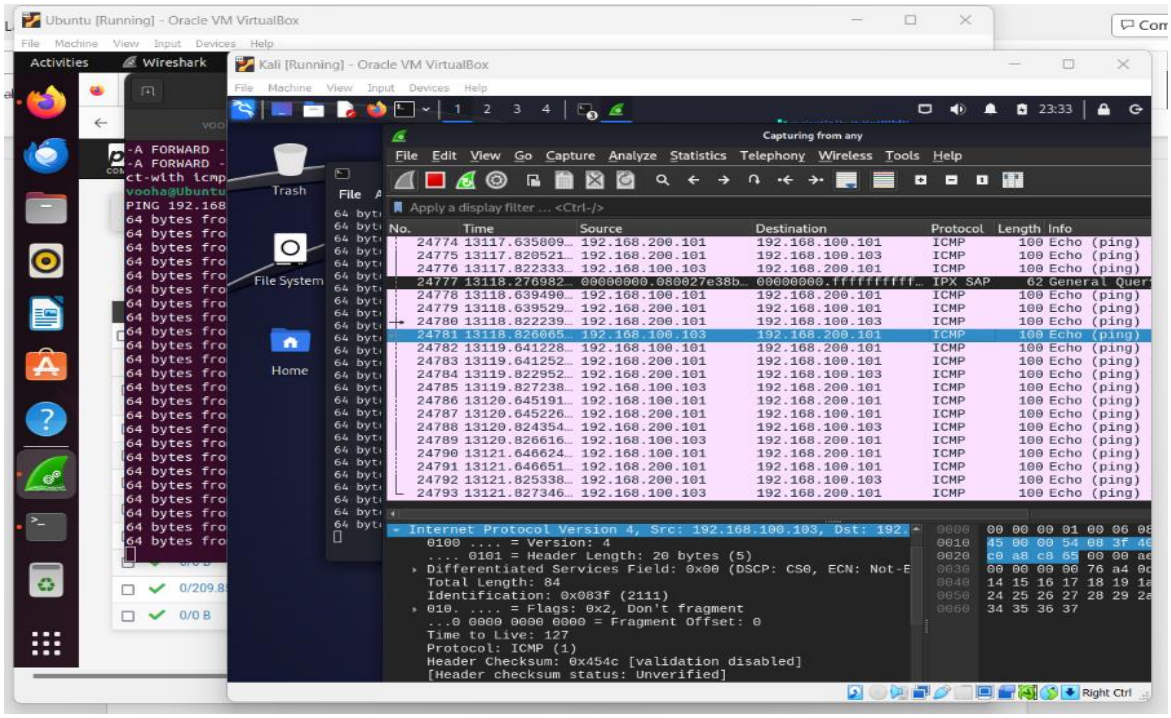
# Blocking all webservice requests to workstations
sudo iptables -A INPUT -p tcp -s 192.168.100.103 -j REJECT

# Accept a one way ping from subnet A to subnet B
sudo iptables -A FORWARD -s 192.168.100.0/24 -d 192.168.200.0/24 -p icmp --icmp-type 8 -j ACCEPT
sudo iptables -A FORWARD -d 192.168.100.0/24 -s 192.168.200.0/24 -p icmp --icmp-type 0 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.200.0/24 -d 192.168.100.0/24 -p icmp --icmp-type 8 -j REJECT

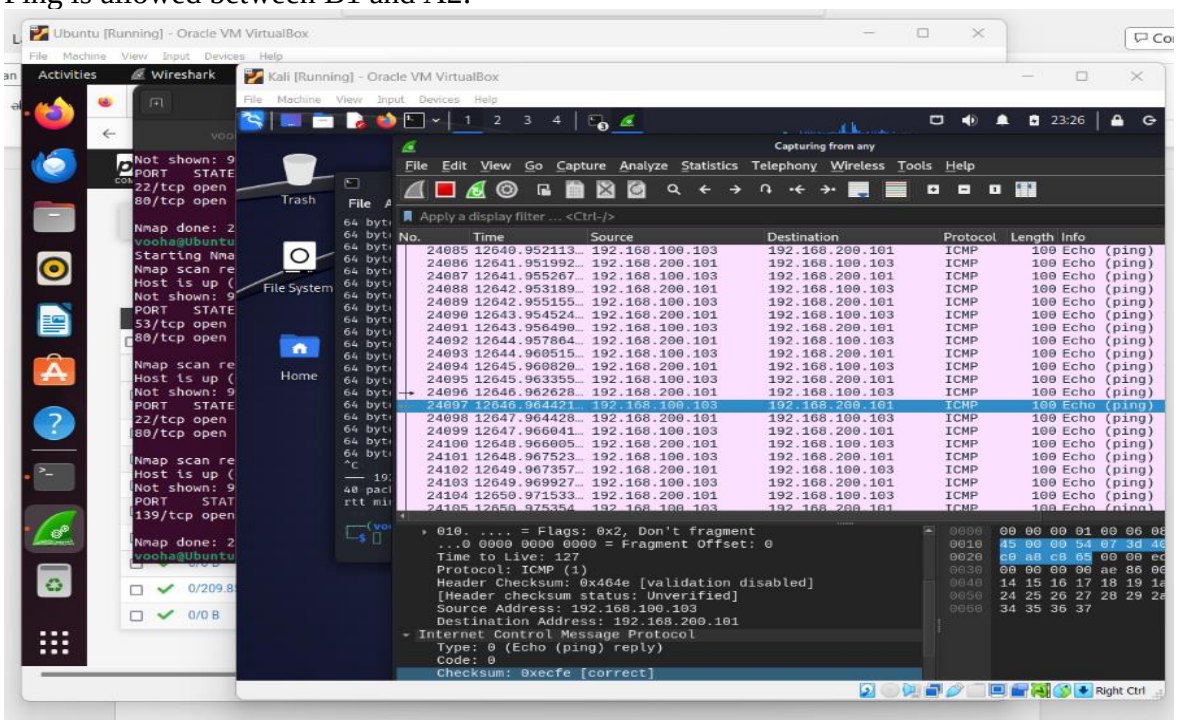
ii. **Wireshark results (screen shots) of routines checking the web service** between B.1 and A.1

Ping is allowed in this case.

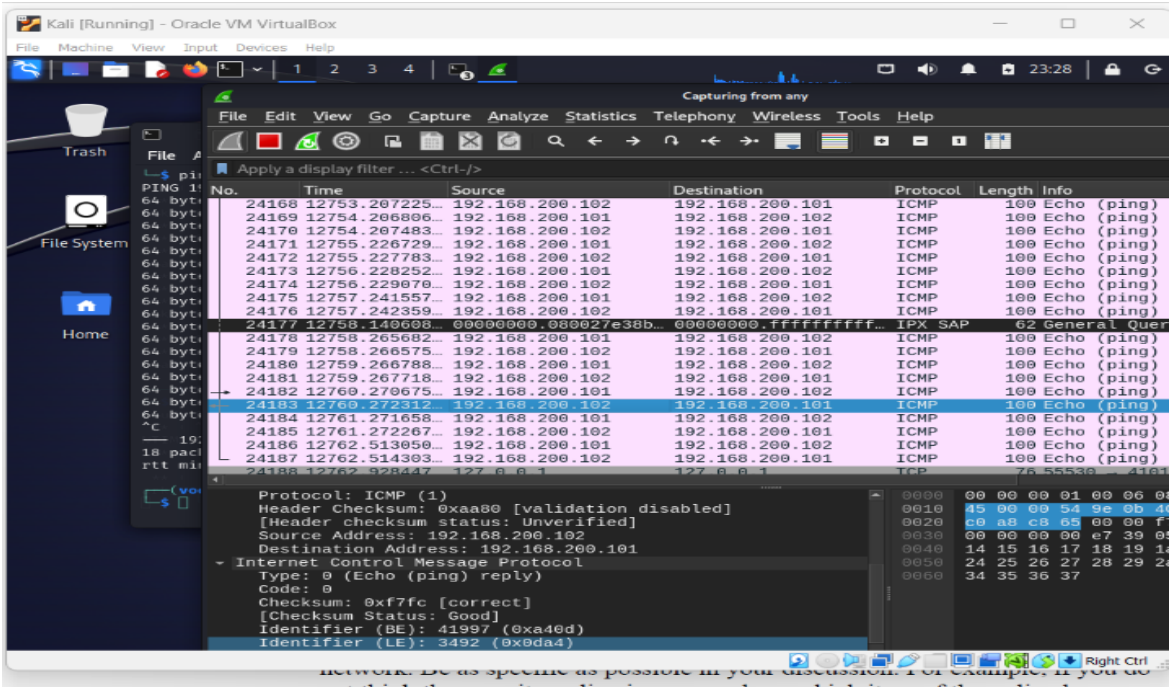




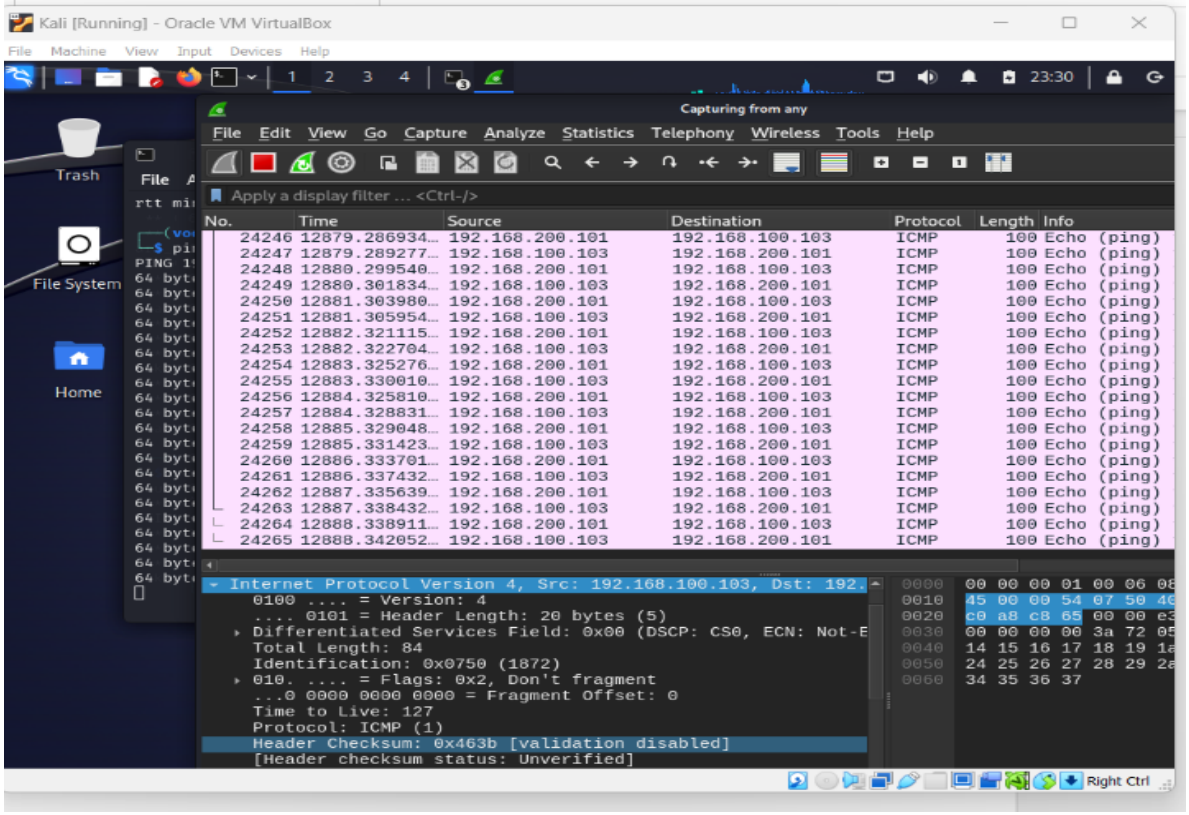
Wireshark results (screen shots) of routines checking the web service between B.1 and A.2:
Ping is allowed between B1 and A2:



Wireshark results (screen shots) of routines checking the web service between B.1 and B.2:
Ping is allowed between B1 and B2:



Wireshark results (screen shots) of routines checking the web service between B.1 and A.2:
Ping is allowed between B1 and A2:



iii. Web services are available all across all the tasks:

The company server (A.1) provides only web service to external computers in Network B.
The company server (A.1) provides both SSH and web services to the workstations (A.2) in Network A.
Workstations (A.2) do not provide any services.
Workstations (A.2) can access services hosted by the server (A.1).

Workstations (A.2) can only access the web service provided by external computers in Network B.

v. Restricting the users in transferring the data through external devices from computer A.1 helps in avoiding data leakage. But this security policy does not prevent any unauthorized access to the computer through the network. There are very high chances for network attacks. By implementing Network security controls, the security policy is improved on the computer A.1.

f) Conclusion:

In this project, we undertook a comprehensive exploration of building, testing, and implementing a security policy within a network environment. The project was divided into several tasks, each focusing on a specific aspect of network setup, security configuration, testing, and analysis.

Task-I involved setting up a sandbox network using virtualization technology to provide a secure environment for conducting security experiments. We utilized VirtualBox, VMware Workstation, and pfSense Router to build a network infrastructure consisting of four machines divided into two internal networks (Network A and Network B).

Task-II focused on checking the network configurations, ensuring all machines were started, NICs were configured correctly, and essential services such as web services and SSH were operational. Additionally, we verified connectivity within the network and to external destinations.

Task-III revolved around network diagnosis, including network discovery using NMap scans and traffic analysis using tools like Wireshark. This task aimed to identify active hosts and services within both internal networks and analyze traffic patterns between different hosts.

Task-IV introduced the implementation of a security policy within the network router (R) to enforce specific access control rules. An access control matrix was created based on the provided security policy, and traffic rules were configured accordingly on the router to enforce the policy.

Task-V involved testing the implementation of the security policy by performing NMap scans to identify exposed services and IPs of Network A. Traffic analysis was conducted to observe any deviations from the expected traffic patterns compared to Task-III.

Task-VI extended the security implementation to the local router of the company server (A.1) to enforce additional security measures. NMap scans and traffic analysis were repeated to assess the impact of these additional security measures.

Overall, this project provided hands-on experience in setting up, securing, testing, and analyzing network environments, equipping participants with valuable skills in network administration and cybersecurity. Through systematic tasks and analyses, participants gained insights into the complexities of network security and the importance of implementing robust security measures to protect against potential threats and vulnerabilities.

Problems faced: Obstacles during creation of VM’s and their connections.

- In the first attempt we were not able to create windows systems properly since we are trying to give high storage to the systems. Later from the YouTube videos we got to

know that each system can support a range of memory. Thus, we resolved the issue in VM creation.

- During the VM's connection at first, we were not able to get the proper IP address to the machines according to the range set in the router. We resolved it by recreating the machine and setting it again.
- For the Ubuntu system setting root user permissions while installing the external software became a challenging task since the user does not have access to "sudo". After a long search, we found that we can add a username under /etc/sudoers file to get root permissions to the user.