

Cybersecurity Home Lab Project

Niña Jorene L. Montecer

November 2024

Project Description

This project aims to establish a foundational cybersecurity home lab with network setup and tool utilization using virtual machines with Linux operating systems allowing for simple experimentation with network security concepts, vulnerability assessment, and penetration testing.

Project Objectives

1. Setup a Virtual Environment
 - a. Configure VMware Workstation to host Ubuntu and Kali Linux virtual machines
2. Install Essential Tools
 - a. Install Nmap and Wireshark as part of the network security experimentation
3. Simulate Network Attacks and Defenses
4. Analyze Network traffic using Wireshark

Project Methodology

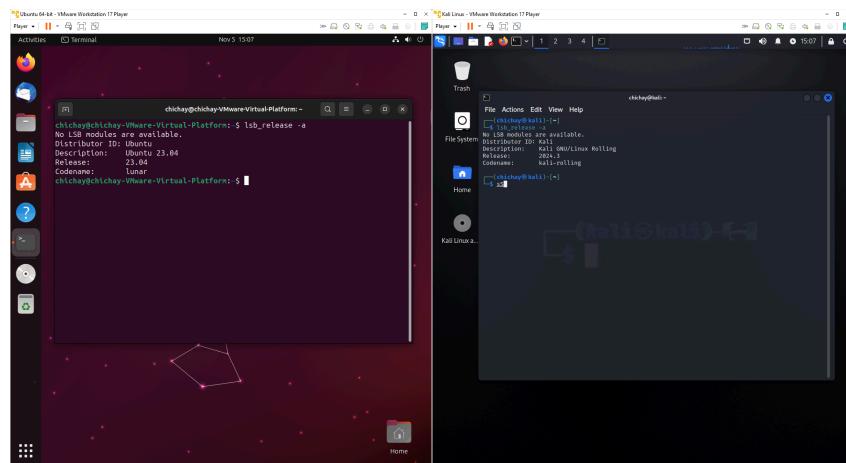
1. Hardware and Software Requirements
 - a. A personal computer or laptop
 - b. VirtualBox or VMware Workstation Player
 - c. Ubuntu and Kali Linux ISO images

2. Virtual Machine Setup

- a. Operating System

Ubuntu 64-bit (Ubuntu)

Debian 10.x 64-bit (Kali Linux)



Nmap: discover devices and vulnerabilities on a network

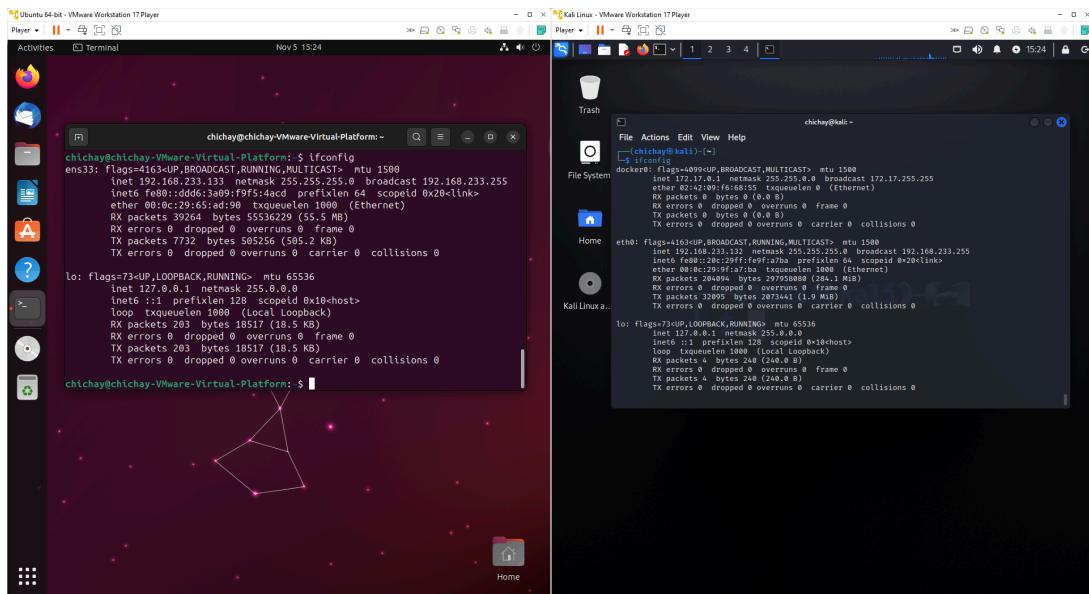
Wireshark: see and analyze network data

UFW (Uncomplicated Firewall): manage network access and improve security on system

5. Simulating Attacks and Defenses

- Determine IP address of both virtual machines and ping both machines to test connectivity and is reachable over the network

Command: `ifconfig`



The image shows two side-by-side Linux desktop environments. On the left is an Ubuntu 64-bit desktop with a terminal window open showing the output of the 'ifconfig' command. The output lists network interfaces: ens33 (ethernet), ens3 (loopback), and lo (loopback). On the right is a Kali Linux desktop with a terminal window showing the same 'ifconfig' command output. Both terminals show similar interface details, including MAC addresses, broadcast addresses, and link layer statistics.

```
chichay@chichay-Virtual-Platform:~$ ifconfig
ens33: flags=4163 mtu 1500
inet 192.168.233.132 brd 192.168.233.255 netmask 255.255.255.0 broadcast 192.168.233.255
inet6 fe80::4c29:65ff%ens33 brd fe80::ff:fe29:65ff scopeid 0x20<link>
      ether 0c:29:65:ad:9f:0a txqueuelen 64  queueing discipline 0x20<llink>
RX packets 39264 bytes 55536229 (55.5 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7732 bytes 505256 (505.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
      ether 00:0c:29:ff:fe:01 txqueuelen 0  queueing discipline 0x20<link>
RX packets 203 bytes 18517 (18.5 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 203 bytes 18517 (18.5 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

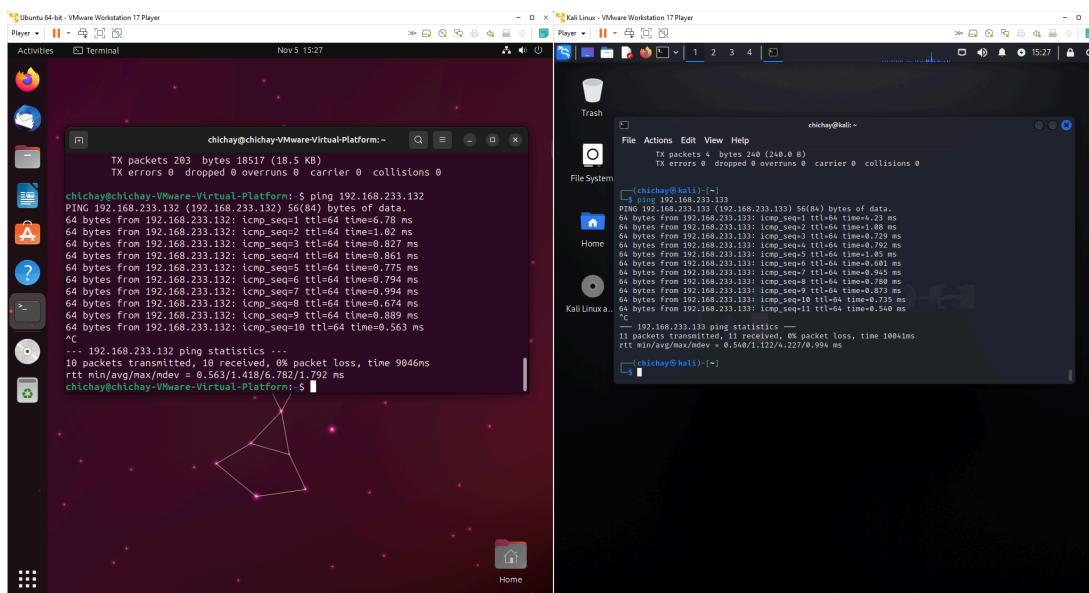
chichay@chichay-Virtual-Platform:~$ 

chichay@chichay-Virtual-Platform:~$ ifconfig
eth0: flags=4163 mtu 1500
inet 192.168.233.132 brd 192.168.233.255 netmask 255.255.255.0 broadcast 192.168.233.255
inet6 fe80::4c29:65ff%eth0 brd fe80::ff:fe29:65ff scopeid 0x20<link>
      ether 00:0c:29:19:f7:a7 txqueuelen 1000  queueing discipline 0x20<link>
RX packets 32095 bytes 2073441 (1.9 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 32095 bytes 2073441 (1.9 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
      ether 00:0c:29:ff:fe:01 txqueuelen 0  queueing discipline 0x20<link>
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Kali Linux a...
```

Command: `ping [target IP]`



The image shows two side-by-side Linux desktop environments. On the left is an Ubuntu 64-bit desktop with a terminal window open showing the output of the 'ping' command to 192.168.233.132. The output shows several ICMP echo requests being sent with TTL values of 64 and 128. On the right is a Kali Linux desktop with a terminal window showing the same 'ping' command output. Both terminals show the same ping statistics, indicating successful connectivity between the two hosts.

```
chichay@chichay-Virtual-Platform:~$ ping 192.168.233.132
PING 192.168.233.132 (192.168.233.132) 56(84) bytes of data.
64 bytes from 192.168.233.132: icmp_seq=1 ttl=64 time=0.78 ms
64 bytes from 192.168.233.132: icmp_seq=2 ttl=64 time=0.22 ms
64 bytes from 192.168.233.132: icmp_seq=3 ttl=64 time=0.87 ms
64 bytes from 192.168.233.132: icmp_seq=4 ttl=64 time=0.86 ms
64 bytes from 192.168.233.132: icmp_seq=5 ttl=64 time=0.77 ms
64 bytes from 192.168.233.132: icmp_seq=6 ttl=64 time=0.94 ms
64 bytes from 192.168.233.132: icmp_seq=7 ttl=64 time=0.79 ms
64 bytes from 192.168.233.132: icmp_seq=8 ttl=64 time=0.67 ms
64 bytes from 192.168.233.132: icmp_seq=9 ttl=64 time=0.89 ms
64 bytes from 192.168.233.132: icmp_seq=10 ttl=64 time=0.56 ms
...
...
--- 192.168.233.132 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9046ms
rtt min/avg/max/mdev = 0.563/1.418/6.782/1.792 ms
chichay@chichay-Virtual-Platform:~$ 

chichay@chichay-Kali:~$ ping 192.168.233.132
PING 192.168.233.132 (192.168.233.132) 56(84) bytes of data.
64 bytes from 192.168.233.132: icmp_seq=1 ttl=64 time=0.23 ms
64 bytes from 192.168.233.132: icmp_seq=2 ttl=64 time=0.79 ms
64 bytes from 192.168.233.132: icmp_seq=3 ttl=64 time=0.729 ms
64 bytes from 192.168.233.132: icmp_seq=4 ttl=64 time=0.45 ms
64 bytes from 192.168.233.132: icmp_seq=5 ttl=64 time=0.601 ms
64 bytes from 192.168.233.132: icmp_seq=6 ttl=64 time=0.45 ms
64 bytes from 192.168.233.132: icmp_seq=7 ttl=64 time=0.788 ms
64 bytes from 192.168.233.132: icmp_seq=8 ttl=64 time=0.873 ms
64 bytes from 192.168.233.132: icmp_seq=9 ttl=64 time=0.735 ms
64 bytes from 192.168.233.132: icmp_seq=10 ttl=64 time=0.516 ms
64 bytes from 192.168.233.132: icmp_seq=11 ttl=64 time=0.516 ms
...
...
102.168.233.132 ping statistics
11 packets transmitted, 11 received, 0% packet loss, time 10841ms
rtt min/avg/max/mdev = 0.540/1.122/4.227/0.994 ms
chichay@chichay-Kali:~$ 
```

- b. Use Nmap to scan the Ubuntu VM for open ports and services (Simulated Attack)

Command: `nmap -A 192.168.233.133`

This command performs an aggressive scan on the target address providing detailed information on open ports and services.

The screenshot shows a terminal window titled 'chichay@kali:~'. The output of the Nmap scan is displayed:

```
File Actions Edit View Help
-- 192.168.233.133 ping statistics --
11 packets transmitted, 11 received, 0% packet loss, time 10041ms
mrtt min/avg/max/mdev = 0.540/1.122/4.227/0.994 ms
m (chichay@kali) [~]
$ nmap -A 192.168.233.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-05 15:39 PST
Nmap scan report for 192.168.233.133
Host is up (0.000835 latency).
All 1000 scanned ports on 192.168.233.133 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 00:0C:29:65:AD:90 (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  0.83 ms  192.168.233.133

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit
/
Nmap done: 1 IP address (1 host up) scanned in 4.35 seconds
(chichay@kali) [~]
$
```

- c. Configure UFW on the Ubuntu VM to implement basic firewall rules and protect against attacks (Simulated Defense)

Commands:

<code>sudo ufw enable</code>	activates firewall
<code>sudo ufw allow ssh</code>	allow connections on port 22
<code>sudo ufw allow from [specific IP]</code>	allow specific traffic or grant access to trusted source

`sudo ufw status` display current status and other rules

The screenshot shows a terminal window titled 'chichay@chichay-VMware-Virtual-Platform:~'. The user runs several commands to configure UFW:

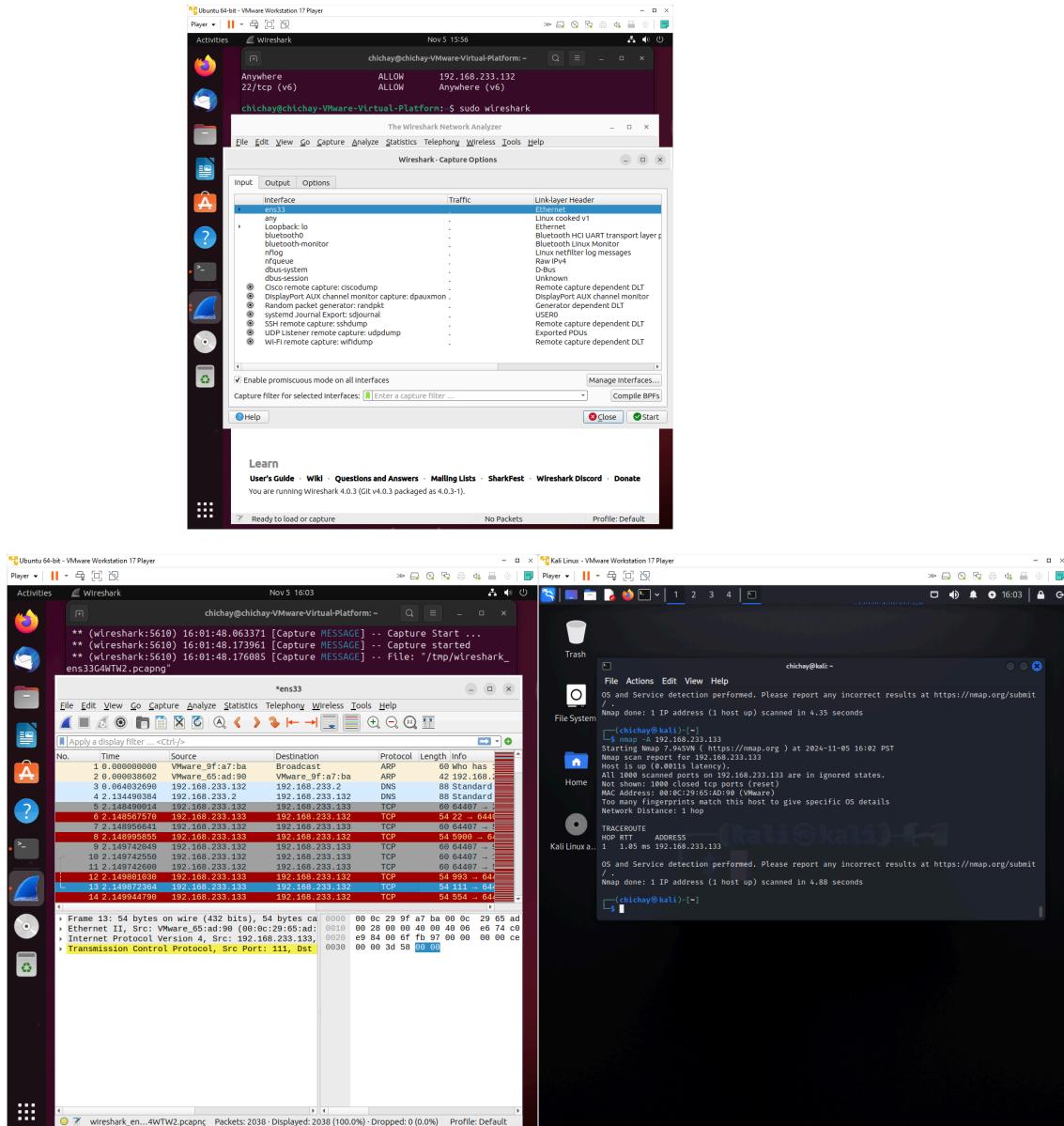
```
chichay@chichay-VMware-Virtual-Platform:~$ sudo ufw enable
Firewall is active and enabled on system startup
chichay@chichay-VMware-Virtual-Platform:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
chichay@chichay-VMware-Virtual-Platform:~$ sudo ufw allow from 192.168.233.132
Rule added
chichay@chichay-VMware-Virtual-Platform:~$ sudo ufw status
Status: active

To           Action    From
--           ----     ---
22/tcp        ALLOW     Anywhere
Anywhere      ALLOW     192.168.233.132
22/tcp (v6)   ALLOW     Anywhere (v6)

chichay@chichay-VMware-Virtual-Platform:~$
```

6. Network Traffic Analysis

- Use Wireshark to capture network traffic coming into the Ubuntu VM in the Ethernet. Simulate the Nmap attack again to record on Wireshark

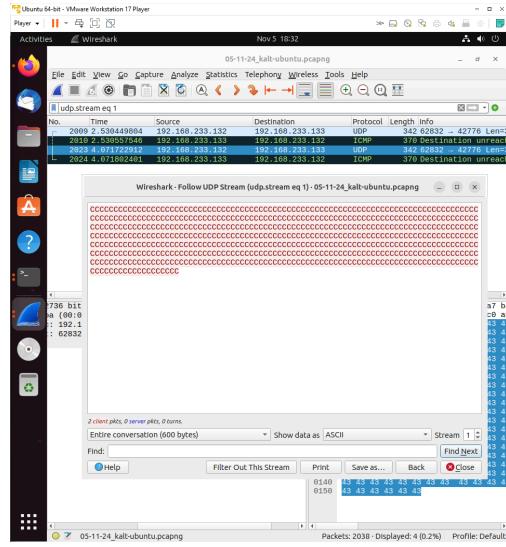


- Analyze the captured packets to identify attack patterns, vulnerabilities, and successful defense mechanisms.

Wireshark Captured Packets Analysis

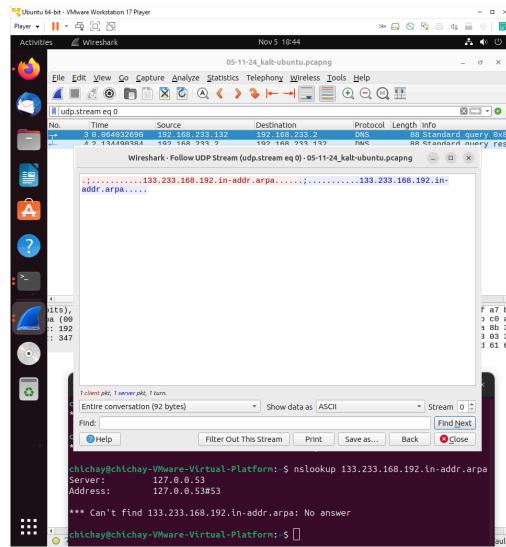
1. UDP Stream

- During the analysis of a captured UDP stream, a significant amount of repetitive character data was identified. The sequence consists of a long repetition of the letter “C”.



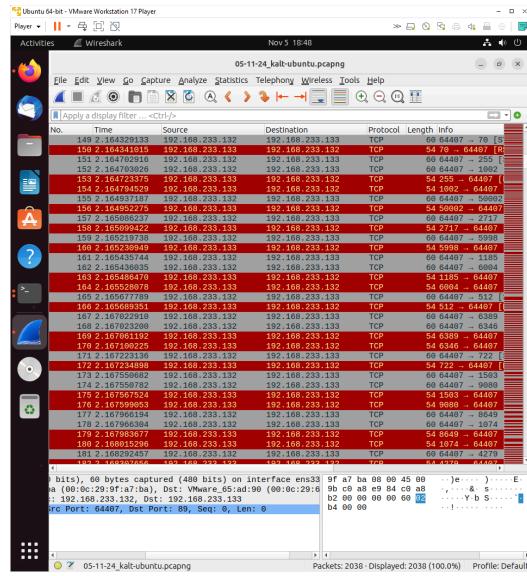
2. DNS Stream

- The following string was observed in the DNS traffic. The repetitive structure suggests a DNS query attempting to resolve the IP address back to a host name. Since the IP address is reversed, a reverse DNS lookup was performed using nslookup. However, it resulted in a “No answer”.



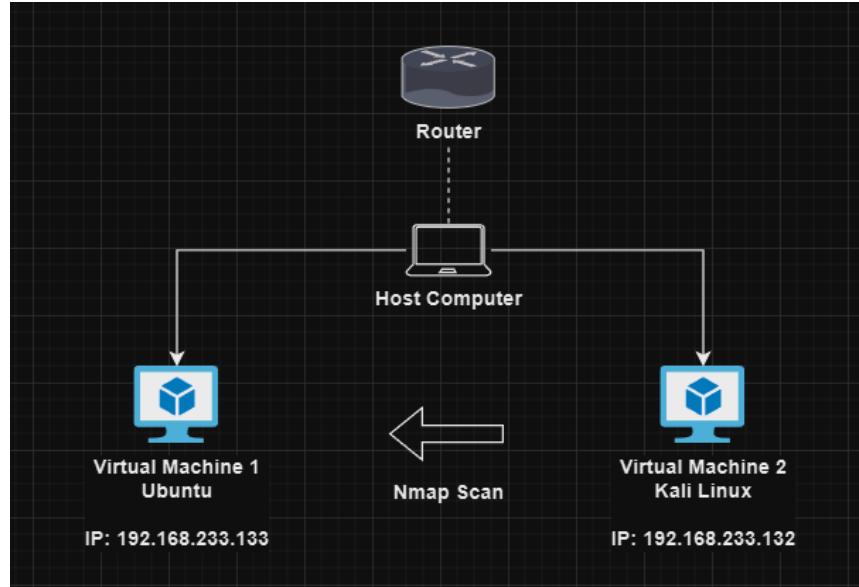
3. TCP Stream

- Most of the packets captured by wireshark were TCP protocols. As per checking the stream, it does not contain any noteworthy contents.



- ### 4. Other protocols captured were ARP (Address Resolution Protocol), which facilitates local network communications through MAC addresses and ICMP (Internet Control Message Protocol), which aids in diagnosing network issues and errors with regards to packets.

Network Diagram



The diagram shows a home lab environment consisting of a router, a host computer, and two virtual machines.

Main Components:

Router

- The central device that connects all other components in the network.
- It facilitates communication between the host computer and the virtual machines.

Host Computer

- The physical computer where the virtual machines are running.
- It acts as the bridge between the physical network and the virtualized environment.

Virtual Machine 1 (Ubuntu)

- A virtual machine running the Ubuntu operating system.
- It serves as the target for network scans and potential attacks.

Virtual Machine 2 (Kali Linux)

- A virtual machine running the Kali Linux operating system.
- It is used to perform network scans, vulnerability assessments, and penetration testing on the Ubuntu VM.

Network Connections:

- The host computer is connected to the router via a wireless connection, as indicated by the dotted line.
- The virtual machines are isolated from the physical network and communicate with the external world through the host computer's network connection, using Network Address Translation (NAT).
- The router assigns IP addresses to all devices in the network, including the virtual machines.

Project Outcomes

1. A functional home lab environment for cybersecurity practice.
2. Basic hands-on experience with network scanning, vulnerability assessment, and penetration testing.
3. Understanding of basic network defense mechanisms and firewall configuration.
4. Ability to analyze network traffic and identify potential threats.

Recommendations for Improvements

- Expand the lab by adding more virtual machines to simulate complex network topologies.
- Provide more In-depth analysis of project outcomes.
- Explore advanced penetration testing techniques and exploit development.
- Explore more security tools already present in Kali Linux.