

BUILD YOUR PERSONAL CYBERSECURITY LAB

Task–2

**Internship Program: Cyber Security &
Ethical Hacking**

Prepared by: Aditya Kamble

I. OBJECTIVE :

The objective of this task is to build a safe and isolated cybersecurity lab environment on a personal computer. This lab will serve as a foundation for performing vulnerability assessment, web application testing, and network validation in a controlled environment.

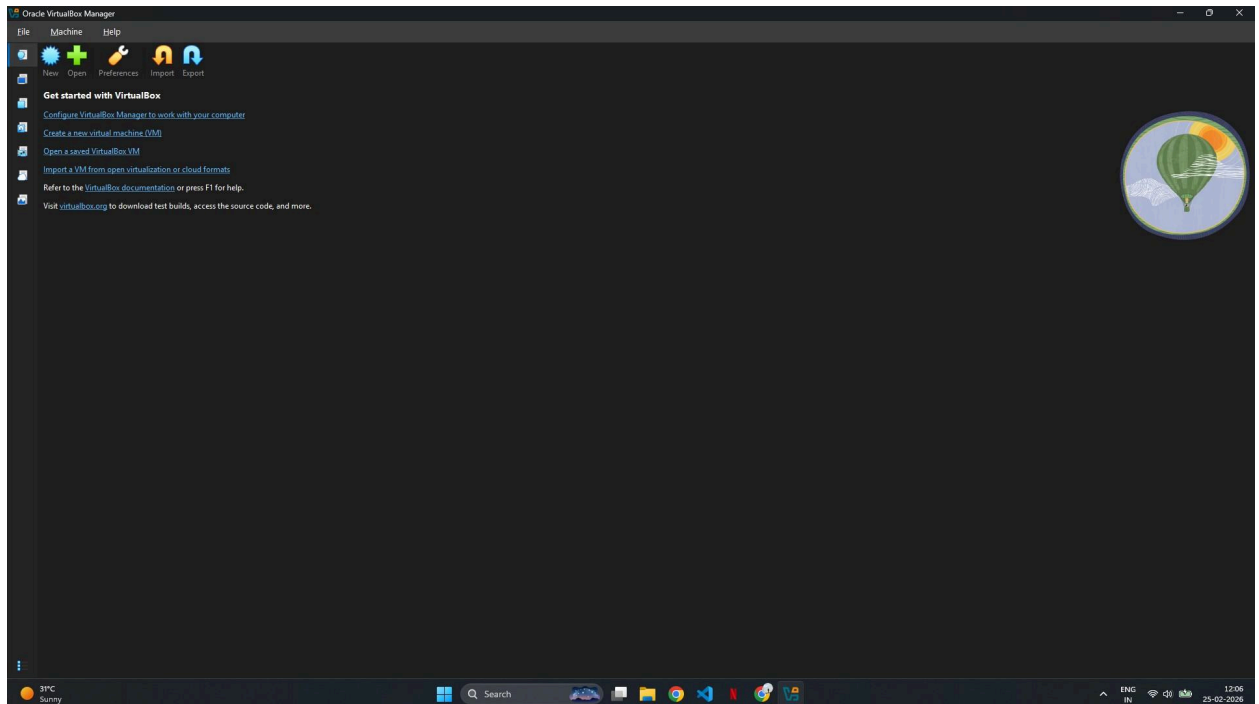
II. SYSTEM REQUIREMENTS :

The lab was built on a system with 16GB RAM and CPU virtualization enabled. Oracle VirtualBox was used as the virtualization platform to create isolated virtual machines.

III. STEPS :

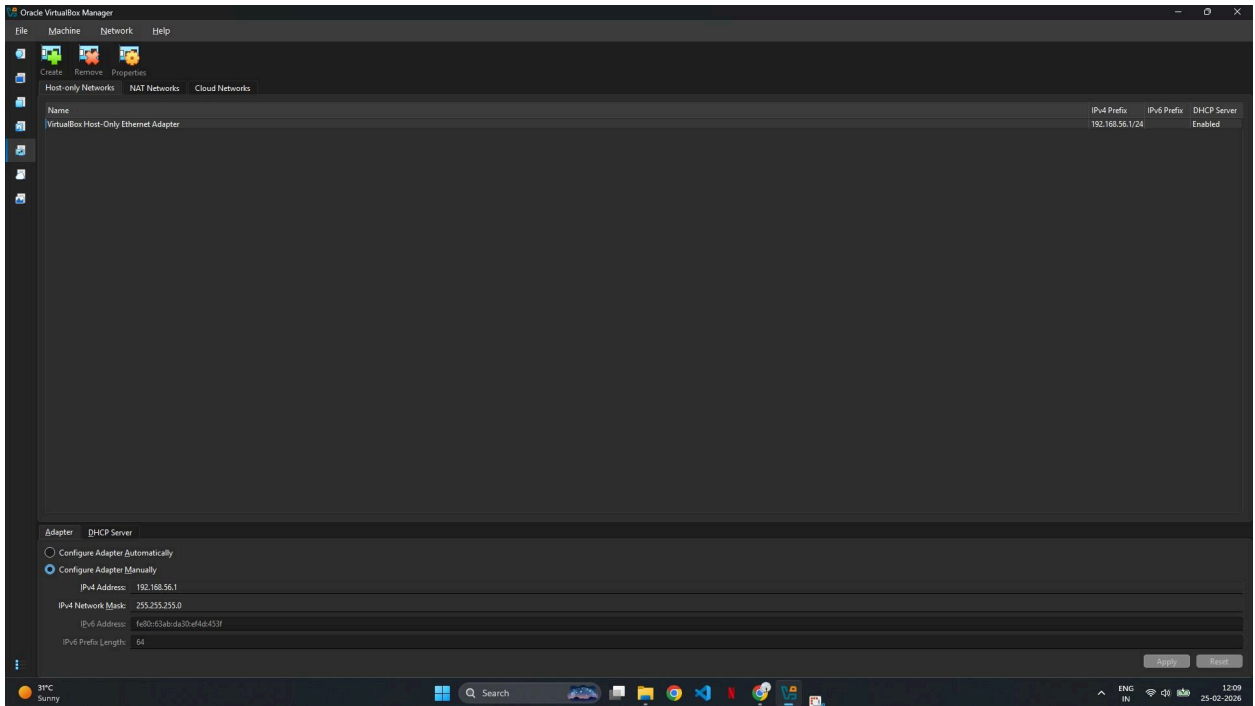
1. Installation of VirtualBox

VirtualBox was installed to create and manage virtual machines for the cybersecurity lab environment. Virtualization allows multiple isolated operating systems to run on a single physical machine, ensuring safe security testing without affecting the host system.



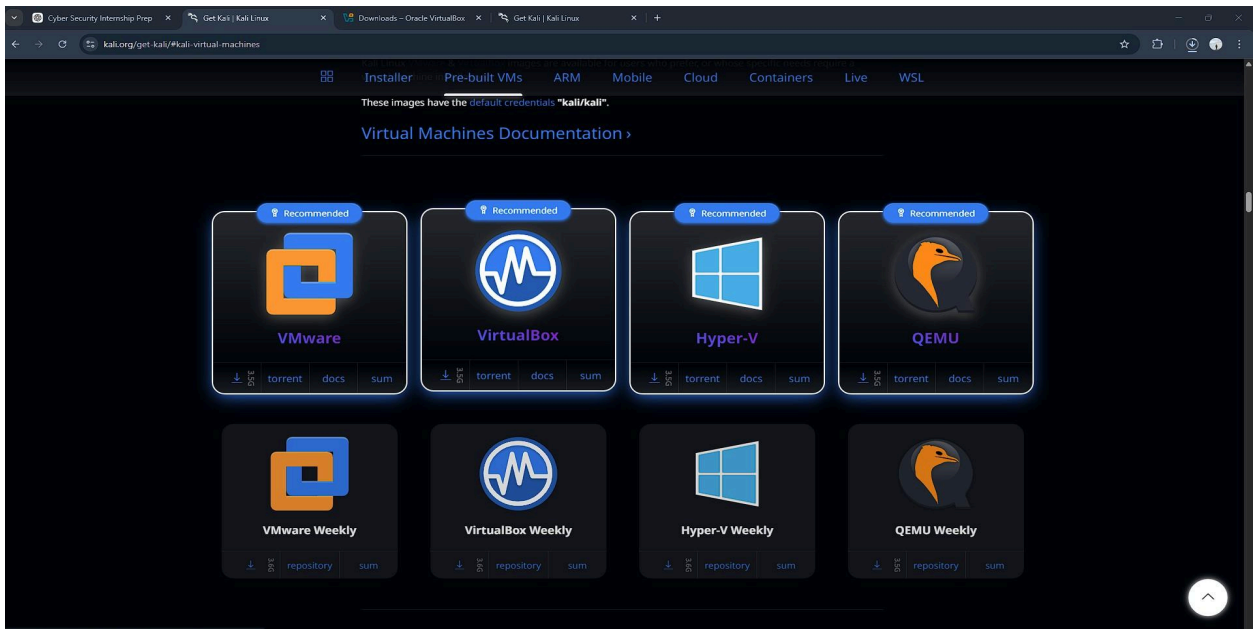
2. Network Configuration (NAT + Host-Only)

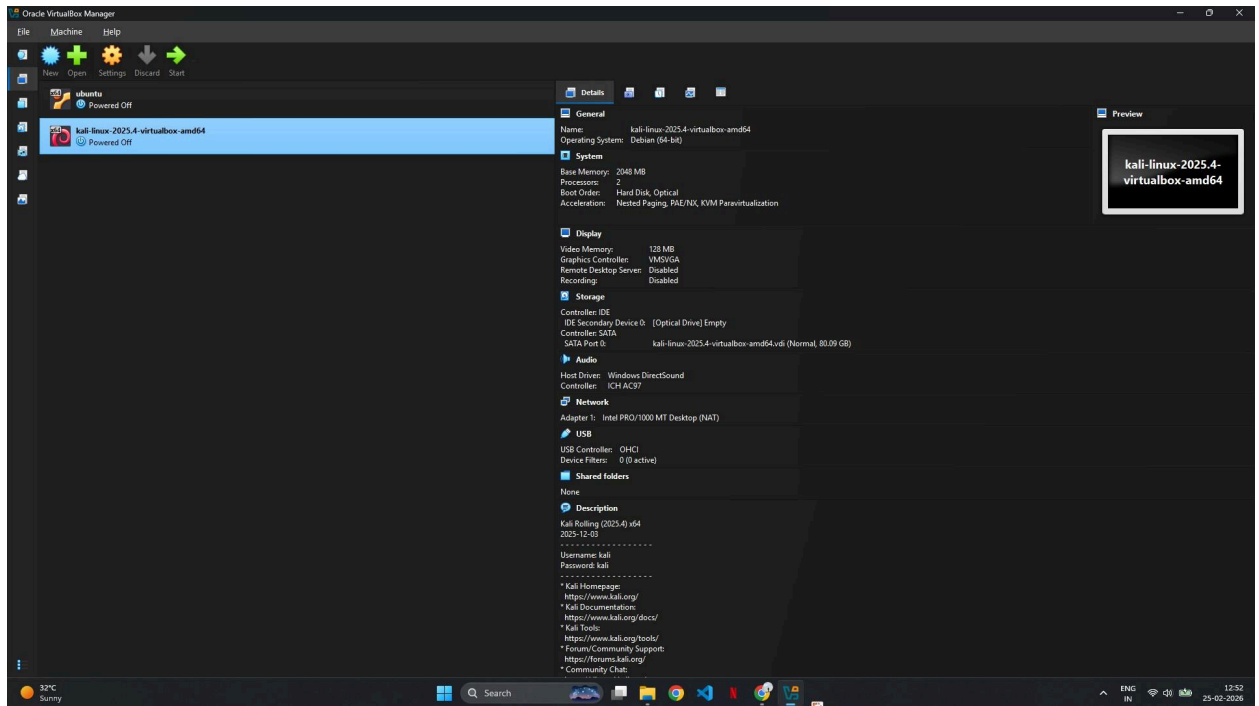
A Host-Only network was created to allow secure communication between the attacker and target machines within an isolated environment. This ensures that lab traffic does not interact with external networks. NAT is used to provide internet access for updates, while Host-Only enables internal lab communication.



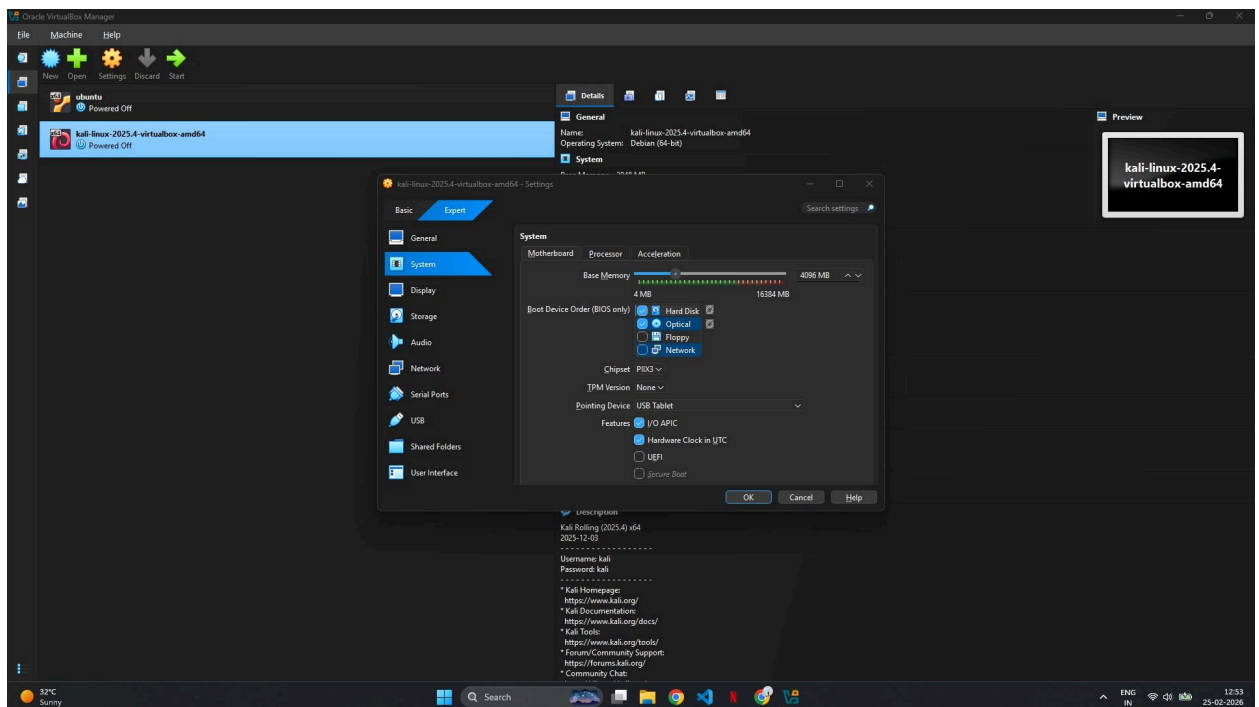
3. Deployment of Attacker Machine (Kali Linux)

i) Kali Linux was deployed as the attacker machine using a prebuilt VirtualBox image. The virtual machine was allocated 2 CPUs and 4GB RAM. Two network adapters were configured: NAT for internet connectivity and Host-Only for internal lab communication.

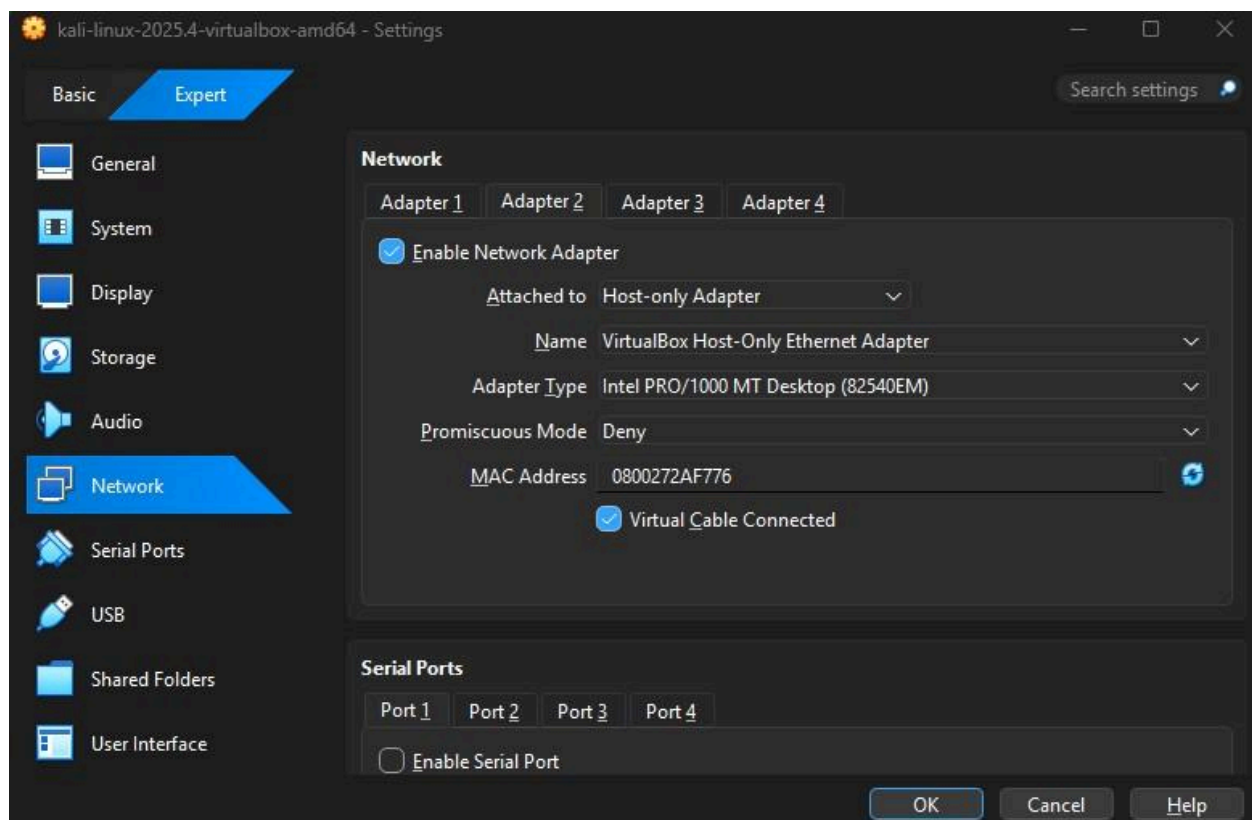
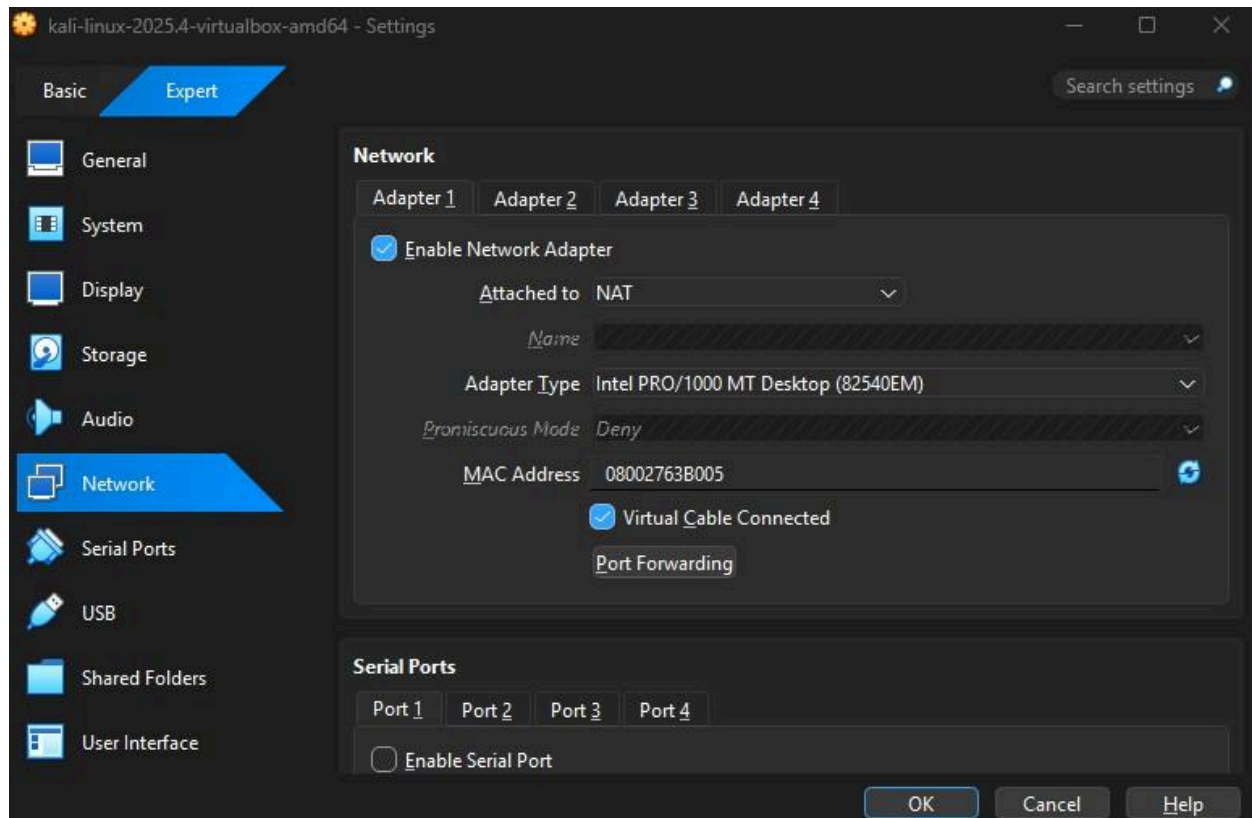




ii) The Kali Linux virtual machine memory was increased to 4GB (4096 MB) to ensure smooth performance during lab operations.

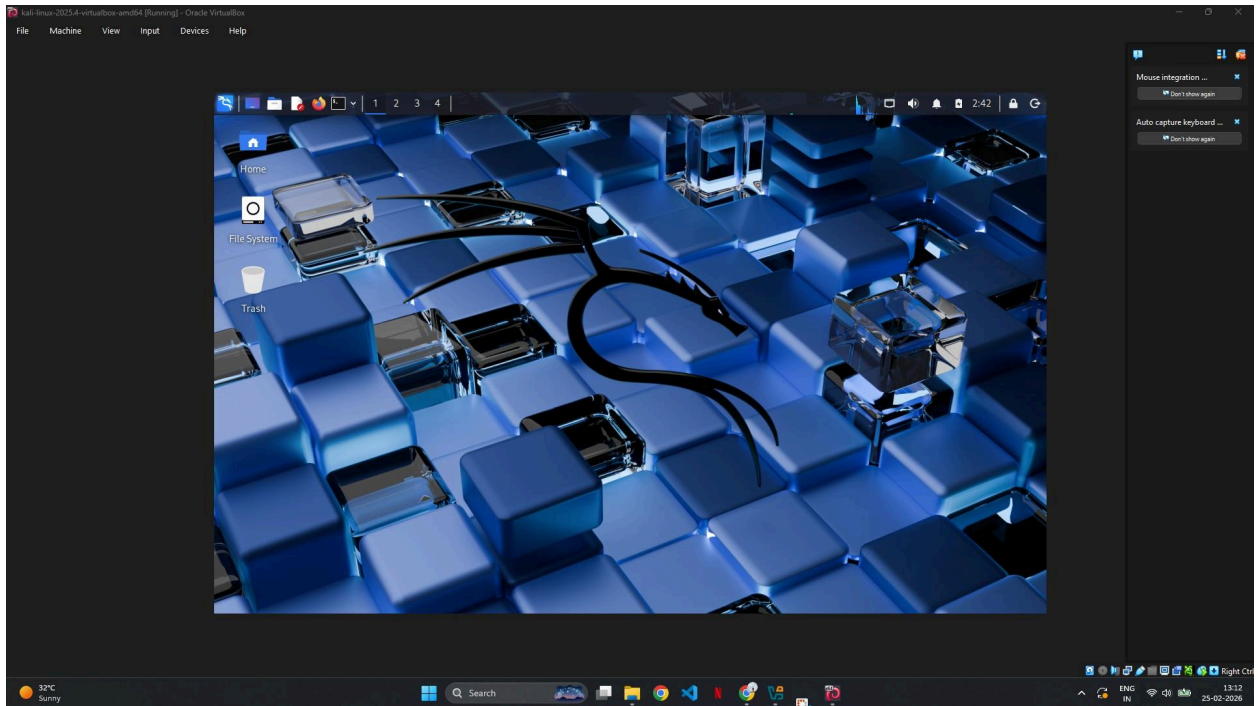


iii) Two network adapters were configured for Kali Linux. Adapter 1 was set to NAT to provide internet connectivity for updates, and Adapter 2 was set to Host-Only to enable secure internal communication within the lab environment.



4. Starting the Attacker Machine

The Kali Linux virtual machine was successfully started after resolving the network configuration issue. The system booted correctly and loaded the Kali desktop environment, confirming proper virtual machine setup.



5. IP Configuration Verification

The `ip a` command was executed to verify network interfaces. The Kali Linux machine successfully obtained two IP addresses:

- 10.0.2.15 via NAT adapter (internet connectivity)
- 192.168.56.101 via Host-Only adapter (internal lab communication)

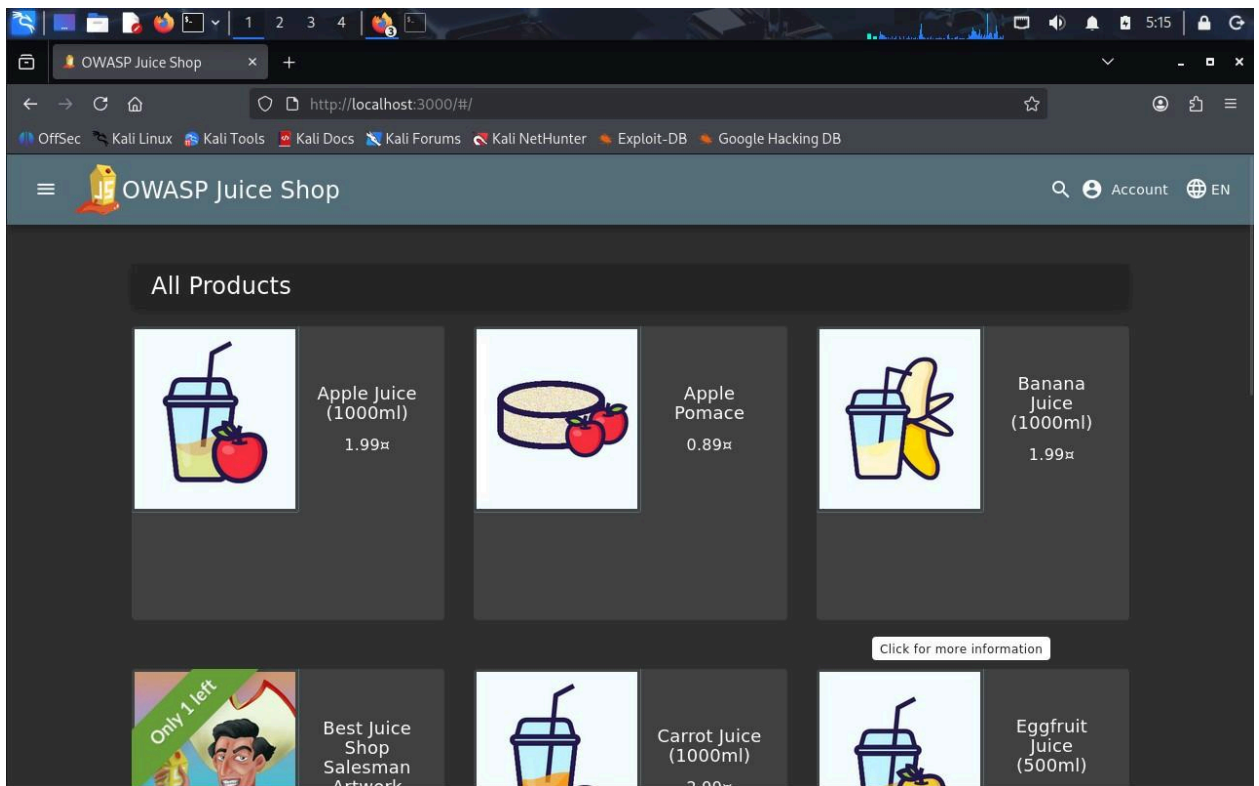
This confirms correct dual-network configuration for secure lab setup.

```
kali@kali: ~  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:63:b0:05 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0  
        valid_lft 86370sec preferred_lft 86370sec  
    inet6 fd17:625c:f937:2:0804:cca:f919:5761/64 scope global dynamic noprefixroute  
        valid_lft 86371sec preferred_lft 14371sec  
    inet6 fe80::6782:cd67:e720:1ebb/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:2a:f7:76 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.56.181/24 brd 192.168.56.255 scope global dynamic noprefixroute eth1  
        valid_lft 570sec preferred_lft 570sec  
    inet6 fe80::4000:f367:c851:3fa0/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
kali@kali: ~
```

6. Deployment of Vulnerable Web Application

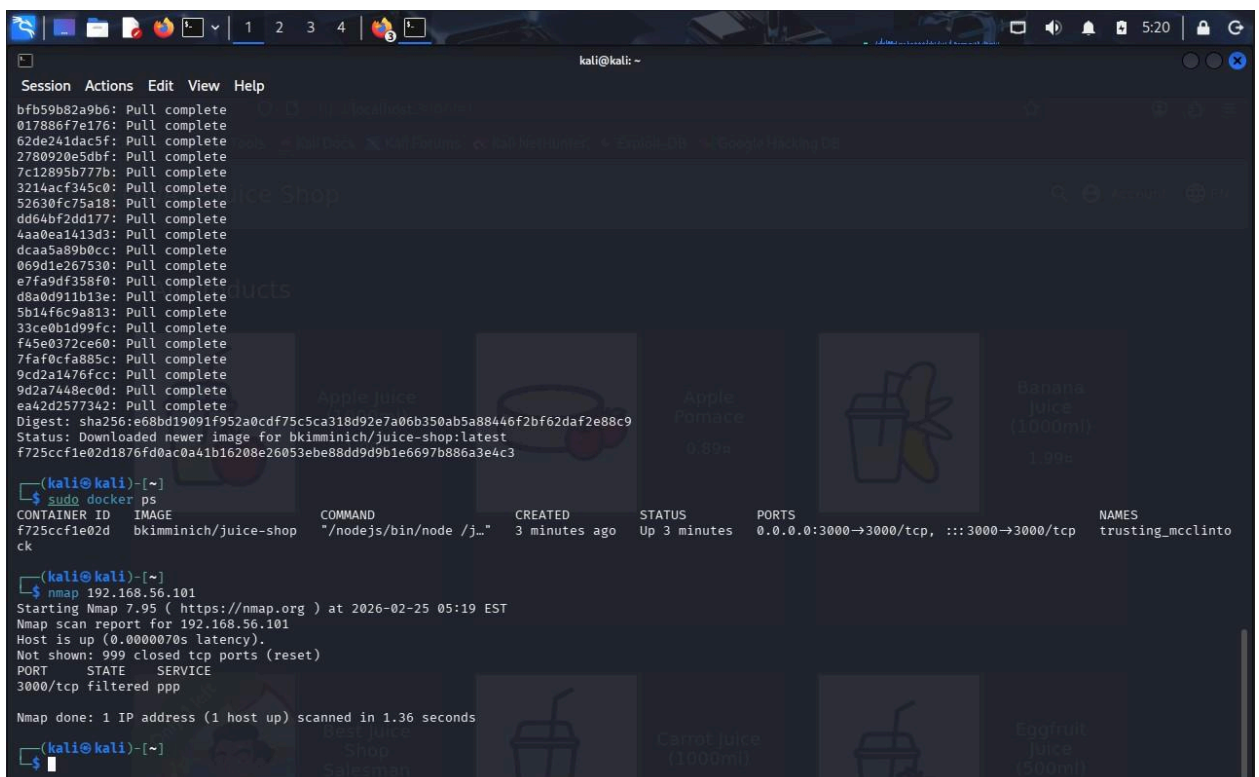
OWASP Juice Shop was deployed using Docker within the Kali Linux environment. The container was launched on port 3000 and verified using the `docker ps` command. The application was successfully accessed via web browser, confirming proper target deployment.

```
kali@kali: ~  
$ sudo docker run -d -p 3000:3000 bkimminich/juice-shop  
[sudo] password for kali:  
Unable to find image 'bkimminich/juice-shop:latest' locally  
latest: Pulling from bkimminich/juice-shop  
fd4aa3667332: Pull complete  
bfb59b82a9b6: Pull complete  
017886f7e176: Pull complete  
62de241dac5f: Pull complete  
2780920e5dbf: Pull complete  
7c12895b777b: Pull complete  
3214acf345c0: Pull complete  
52630fc75a18: Pull complete  
dd64bf2dd177: Pull complete  
4aa0ea1413d3: Pull complete  
dcaa5a89b0cc: Pull complete  
069d1e267530: Pull complete  
e7fa9df358f0: Pull complete  
d8a0d911b13e: Pull complete  
5b14fc9a8121: Pull complete  
33ce0b1d09fc: Pull complete  
fa5e0372ce60: Pull complete  
7faf0cfa885c: Pull complete  
9cd2a1476fcc: Pull complete  
9d2a7448ec0d: Pull complete  
ea42d2577342: Pull complete  
Digest: sha256:e68bd19091f952a0cdf75c5ca318d92e7a06b350ab5a88446f2bf62daf2e88c9  
Status: Downloaded newer image for bkimminich/juice-shop:latest  
f725ccf1e02d1876fd0ac0a41b16208e26053ebe88dd9d9b1e6697b886a3e4c3  
kali@kali: ~  
$ sudo docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES  
f725ccf1e02d        bkimminich/juice-shop  "/nodejs/bin/node /j..."  3 minutes ago      Up 3 minutes       0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  trusting_mcclinto
```

7. Network Validation Using Nmap

A network scan was performed using Nmap against the Host-Only IP address (192.168.56.101). The scan confirmed that the host was active and port 3000 was detected. This validates that the vulnerable web application is reachable within the lab environment.



8. Traffic Interception Using Burp Suite

Burp Suite was configured as a local proxy (127.0.0.1:8080) to intercept HTTP traffic from the browser. When accessing OWASP Juice Shop, HTTP requests to `localhost:3000` were successfully captured. This confirms proper proxy configuration and successful traffic interception within the lab environment.

The screenshot displays the Burp Suite Community Edition v2025.10.6 interface. The top menu bar includes options like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. The 'Proxy' tab is active, showing a table of intercepted HTTP requests.

Time	Type	Direction	Method	URL	Status code	Length
05:33:33.25...	HTTP	→ Request	GET	https://www.google.com/warmup.html		
05:38:19.25...	HTTP	→ Request	GET	http://localhost:3000/		
05:38:35.25...	HTTP	→ Request	GET	https://www.google.com/warmup.html		

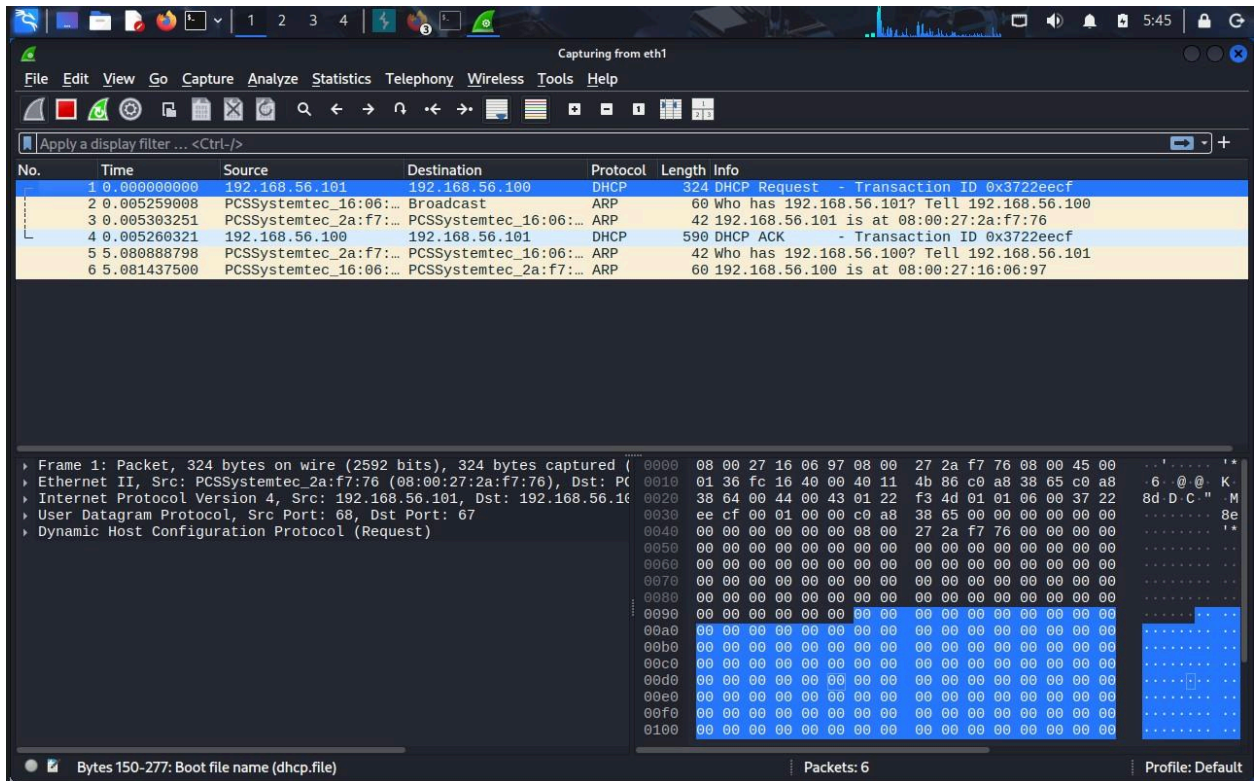
Below the table, the 'Request' tab is selected, showing the raw HTTP request details for the selected request (05:38:19.25...). The request is a GET request to `http://localhost:3000/`. The raw request text is as follows:

```
1 GET /warmup.html HTTP/2
2 Host: www.google.com
3 Sec-Ch-Ua: "Not A Brand";v="99", "Chromium";v="142"
4 Sec-Ch-Ua-Mobile: 70
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 Sec-Purpose: prefetch;prerender
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 X-Client-Data: CNP3ygE=
12 Sec-Fetch-Site: none
13 Sec-Fetch-Mode: navigate
```

The 'Inspector' tab on the right shows the request details, including the method (GET) and the decoded from field (Select). The status bar at the bottom indicates 'Memory: 143.4MB' and 'Disabled'.

9. Packet Capture Using Wireshark

Wireshark was used to monitor traffic on the Host-Only interface (eth1). During lab activity, DHCP and ARP packets within the 192.168.56.x subnet were captured. This confirms successful internal network communication and packet-level monitoring within the isolated cybersecurity lab.



CONCLUSION :

The objective of this task was to design and deploy a secure and isolated cybersecurity lab environment using virtualization technologies. The lab was successfully built using Oracle VirtualBox with Kali Linux as the attacker machine and OWASP Juice Shop as the intentionally vulnerable target application.

A dual-network configuration was implemented using NAT and Host-Only adapters to ensure both internet connectivity and secure internal communication. Network validation was performed using Nmap to confirm service availability, Burp Suite to intercept HTTP traffic, and Wireshark to capture packet-level communication within the isolated lab network.

The successful deployment and validation of all components demonstrate a foundational understanding of virtualization, networking configuration, traffic analysis, and secure lab isolation. This lab environment now serves as a practical platform for future vulnerability assessment, penetration testing, and cybersecurity experimentation in a controlled and safe manner.