METASPLOITABLE-2 Penetration Testing Report

1. Introduction

This penetration test targets Metasploitable-2, a vulnerable Linux-based virtual machine designed for security testing. The objective is to simulate real-world attacks to identify vulnerabilities within the system. Tools like Nmap and Metasploit will be used for scanning and exploitation, with the goal of uncovering weaknesses and providing recommendations for improving system security..

2. Test Scope

The scope of this penetration test focuses exclusively on the Metasploitable-2 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 within 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

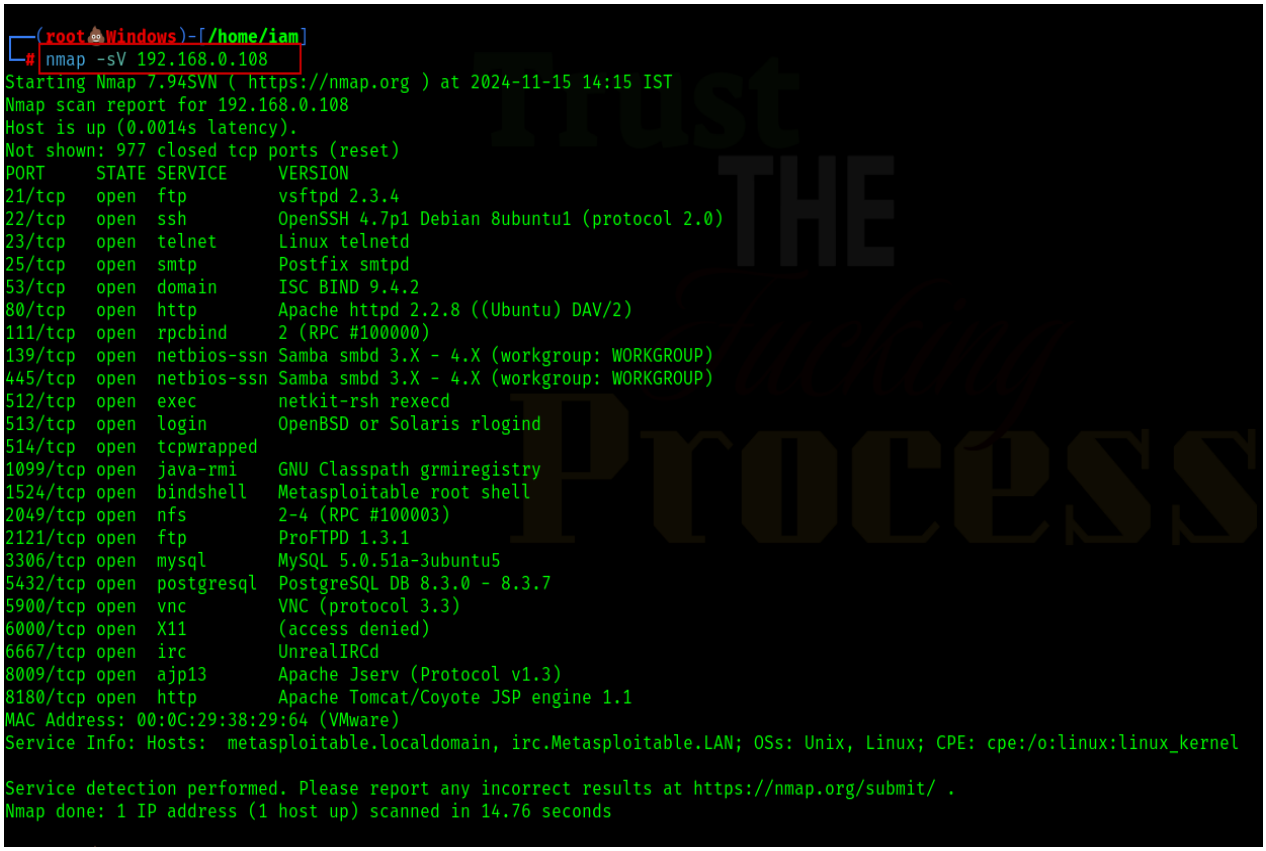
Description: I ran the Nmap scan with the following command to identify open ports and services on the Metasploitable-2 machine:

- nmap -sV 192.168.0.108

The scan revealed all open ports, their respective states, and the associated services and versions running on the target system. Upon identifying the PostgreSQL service, I started it using the following command:

- service postgresql start

Screenshot:



4.2 : Exploiting vsftpd Service

Description: During the Nmap scan, I identified the vsFTPd service running on the target system. The Nmap scan revealed the FTP service version, which indicated the potential for a known vulnerability in vsFTPd. Specifically, I found that the system was running vsFTPd version 2.3.4, which is vulnerable to the backdoor attack:

To exploit this, I used the Metasploit framework with the following exploit:

- **msf6 exploit(unix/ftp/vsftpd_234_backdoor)**

After setting the **RHOST** , I ran the exploit. This successfully opened a reverse shell, providing access to the target system. This backdoor vulnerability in vsFTPd allowed for remote command execution without requiring authentication.

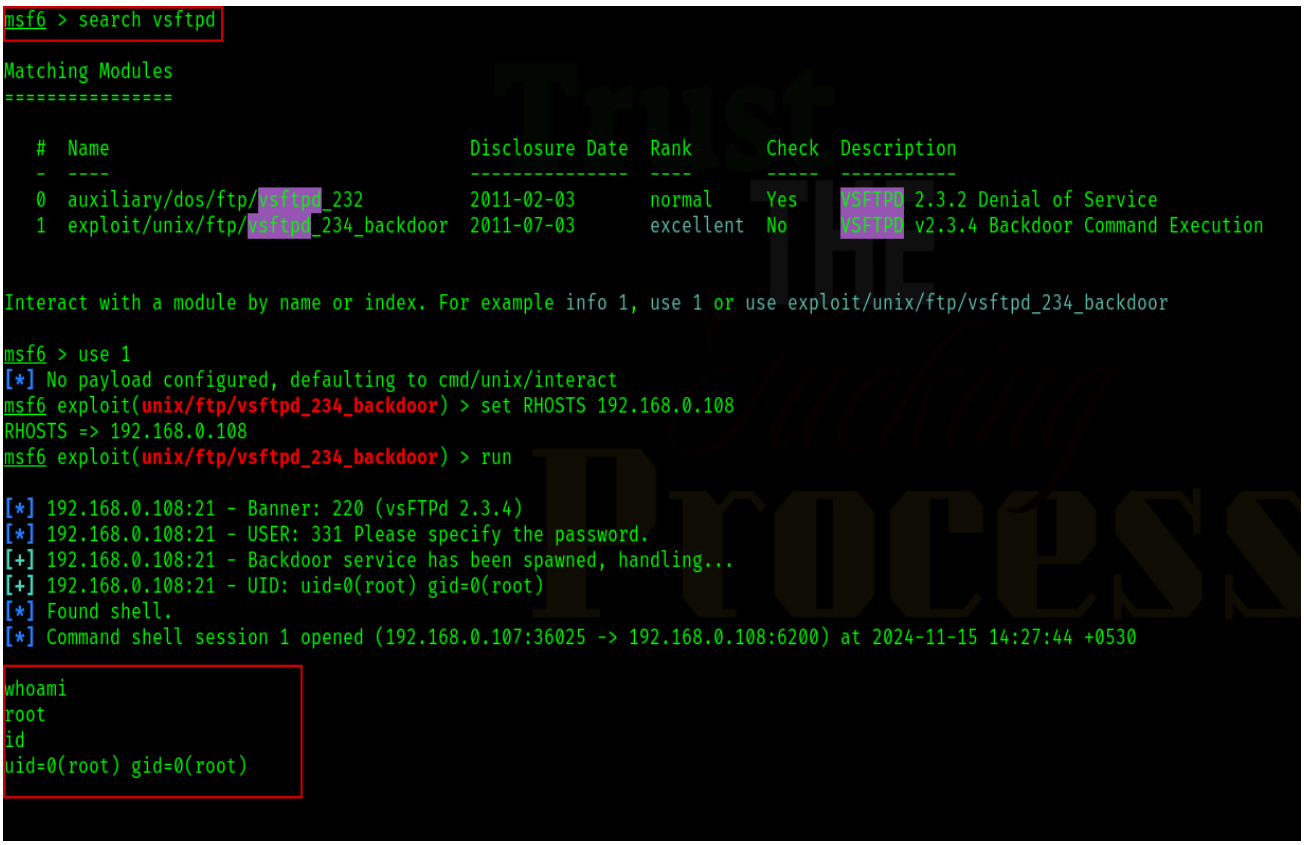
Steps:

1. Identify vsFTPd service through Nmap scan.
2. Search for exploit: vsftpd 2.3.4 backdoor.

3. Use Metasploit's **vsftpd_234_backdoor** exploit.
4. Set **RHOST** to the target IP.
5. Run the exploit and gain reverse shell access.

This step demonstrates how vulnerable versions of FTP services can be exploited to gain unauthorized access to a system.

Screenshot:



4.3 : Exploiting SSH Login

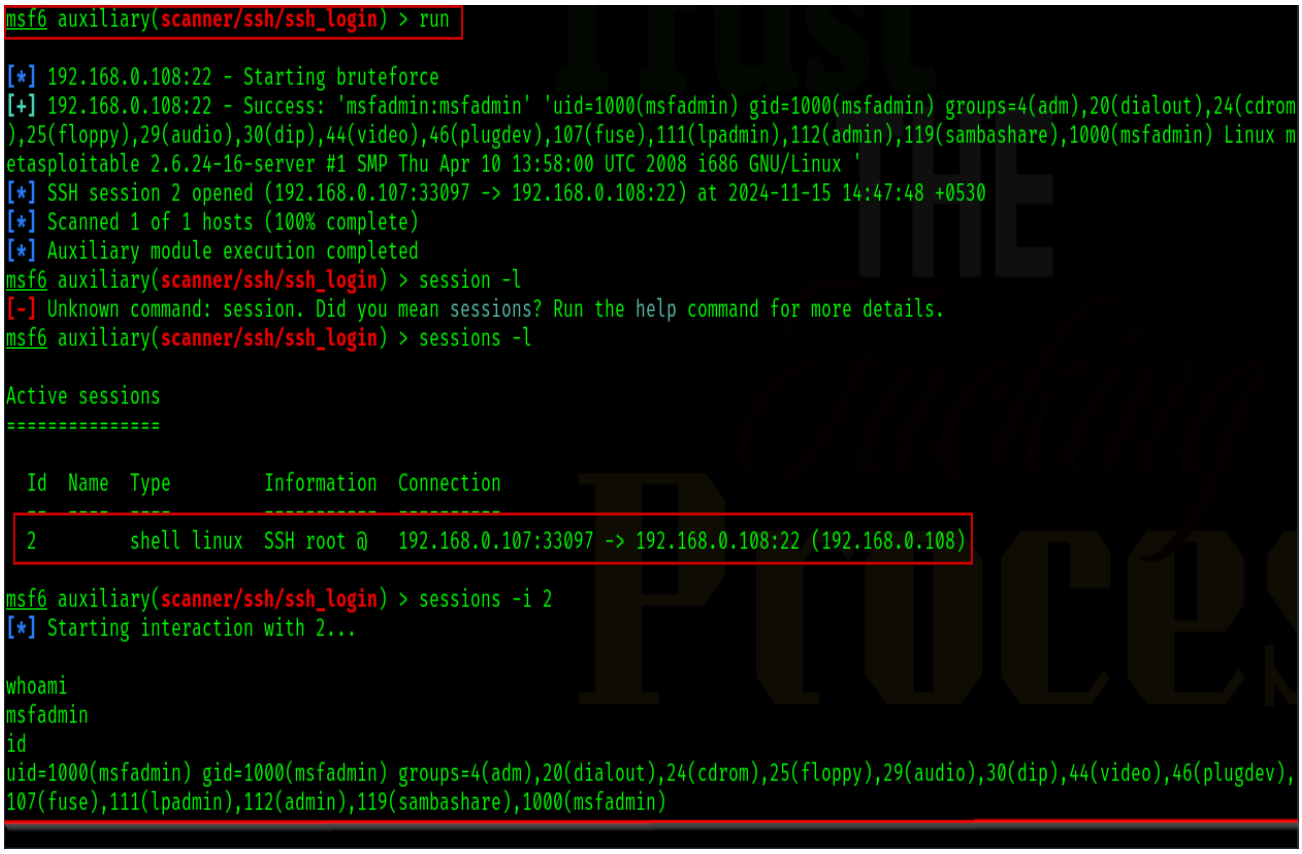
Description: In this phase, I focused on the SSH service running on the Metasploitable-2 machine. Using the **auxiliary(scanner/ssh/ssh_login)** module in Metasploit, I attempted to brute-force the SSH login by providing a list of potential usernames and passwords.

The steps I followed are:

1. **Search for SSH login exploit**: I used the Metasploit module auxiliary(scanner/ssh/ssh_login) to attempt brute-forcing the SSH service.

2. **Set parameters**: I configured the module by specifying the following options:
    - **RHOST**: The target IP address of the Metasploitable-2 system.
    - **USERFILE**: A file containing a list of potential usernames.
    - **PASSFILE**: A file containing a list of common passwords.
3. **Run the exploit**: After configuring the module, I ran it to attempt multiple login combinations.
4. **Successful login and reverse shell**: The brute-force attempt succeeded in logging into the SSH service, resulting in a reverse shell. This gave me remote access to the target system via SSH.

Screenshot:



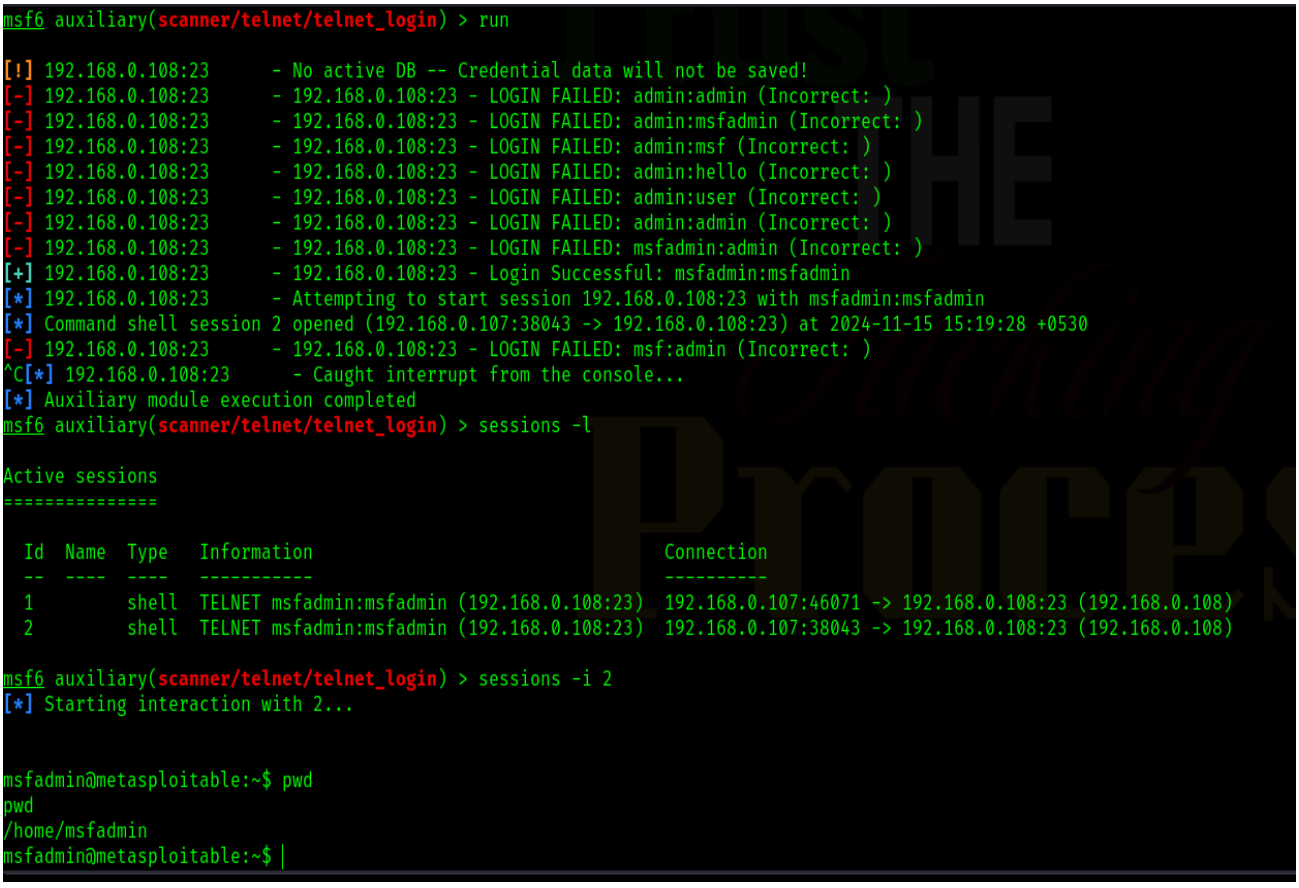
4.4 : Exploiting Telnet Login

Description: For this step, I targeted the Telnet service on Metasploitable-2 using Metasploit's **auxiliary(scanner/telnet/telnet_login)** module. This allowed me to attempt a brute-force login on Telnet, using a list of potential usernames and passwords.

The process followed:

1. **Search for Telnet login exploit**: In Metasploit, I found the module auxiliary(scanner/telnet/telnet_login) to brute-force the Telnet login credentials.
2. **Configure the module**: I set the following parameters:
    - **RHOST**: The IP address of the Metasploitable-2 target.
    - **USERFILE**: A file containing possible usernames.
    - **PASSFILE**: A file containing common passwords.
3. **Run the brute-force attempt**: Upon running the module, it attempted various username-password combinations until it successfully authenticated with the target's Telnet service.
4. **Gaining a session**: After successfully brute-forcing the login, a session was created. By executing session -i 2, I was able to interact with the session, gaining command-line access as the msfadmin user on the Metasploitable-2 machine.

Screenshot:



4.5 : Exploiting VNC Login

Description: In this step, I targeted the **VNC (Virtual Network Computing)** service running on Metasploitable-2 using the **auxiliary(scanner/vnc/vnc_login)** module in Metasploit to identify weak or default VNC credentials. Upon successful authentication, I accessed the VNC session.

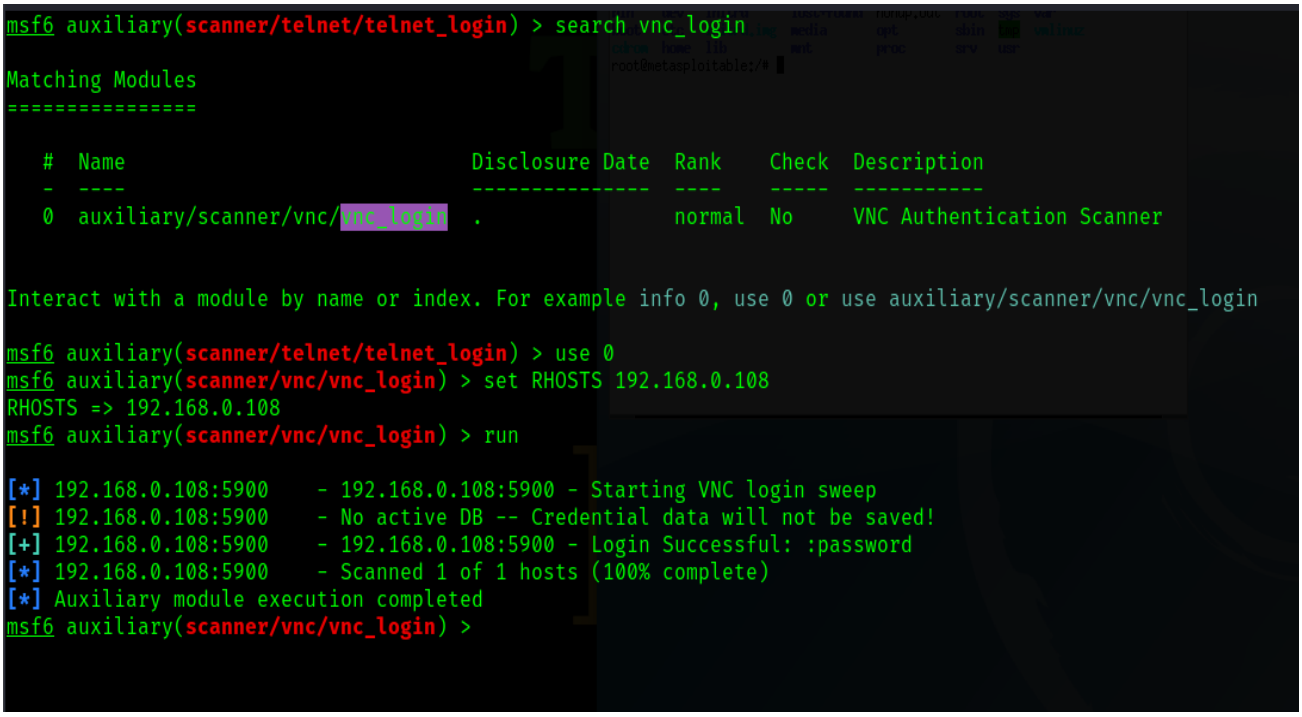
Steps:

1. **Search and configure the VNC login module:**
    - **I searched for the VNC login brute-forcing module in Metasploit: search vnc_login.**
    - **I selected the module: use auxiliary(scanner/vnc/vnc_login).**
2. **Set parameters:**
    - **RHOSTS: Set the target IP address: set RHOSTS 192.168.0.108.**
3. **Run the module:**
    - **Executed the module using the run command.**
    - **The module attempted to log in to the VNC service with default or weak credentials.**
4. **Successful login:**
    - **The module successfully logged in using the password password:**
        - **[+] 192.168.0.108:5900 - Login Successful: :password**
5. **Accessing the VNC session:**
    - After identifying the valid credentials, I used the vncviewer tool to connect**:**
        - **vncviewer 192.168.0.108**
    - Entered the password **password** to authenticate.
    - Successfully accessed the VNC interface, revealing the desktop environment of the target.

Outcome:

Gaining access to the VNC session provided control over the graphical user interface (GUI) of the target system. This demonstrated how weak VNC passwords can compromise system security, highlighting the importance of strong authentication practices.

Screenshot:


msf6 auxiliary(scanner/telnet/telnet_login) > search vnc_login
Matching Modules
Name Disclosure Date Rank Check Description
0 auxiliary/scanner/vnc/vnc_login . normal No VNC Authentication Scanner
Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login
msf6 auxiliary(scanner/telnet/telnet_login) > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.0.108
RHOSTS => 192.168.0.108
msf6 auxiliary(scanner/vnc/vnc_login) > run
[*] 192.168.0.108:5900 - 192.168.0.108:5900 - Starting VNC login sweep
[!] 192.168.0.108:5900 - No active DB -- Credential data will not be saved!
[+] 192.168.0.108:5900 - 192.168.0.108:5900 - Login Successful: :password
[*] 192.168.0.108:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >


(root Windows)-[/home/iam]
vncviewer 192.168.0.108
Connected to RFB server, using protocol version 3.
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shi
Using default colormap which is TrueColor. Pixel
32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shi
TightVNC: root's X desktop (metasploitable:0)
root@metasploitable: /
root@metasploitable:/# ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt sbin tmp vmlinuz
cdrom home lib mnt proc srv usr
root@metasploitable:/# HACKDBYKHP