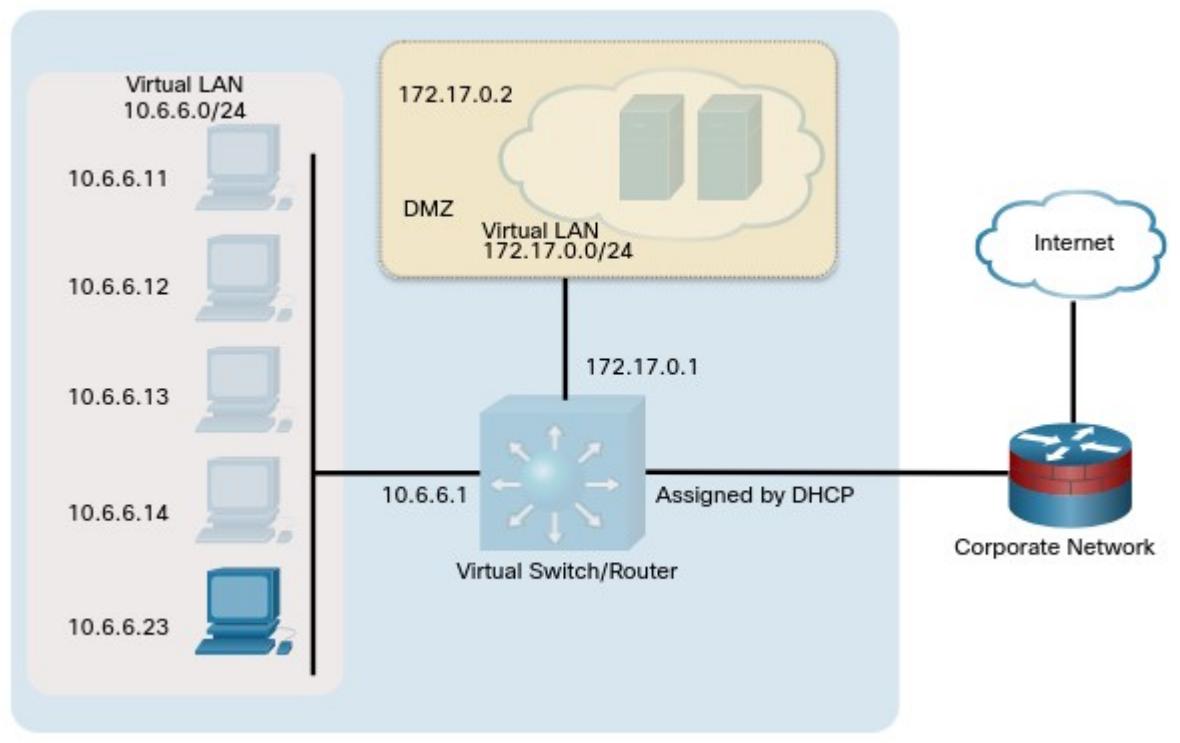


Lab - Enumeration with Nmap

Topology



Objectives

Nmap is a powerful open-source tool for network mapping and discovery. In this lab, you will use Nmap as part of your active reconnaissance strategy.

- Investigate Nmap
- Perform Basic Nmap Scans

Background / Scenario

A Wireshark capture shows unusual activity on a machine on the 10.6.6.0 DMZ network. You've been asked to do some active recon on the machine to determine what services it may be offering and if there are vulnerable applications that could present security issues. The IP address of the suspicious computer is 10.6.6.23. You have access to a Kali Linux system on the 10.6.6.0 network.

Required Resources

- Kali VM customized for Ethical Hacker course

Instructions

Part 1: Investigate Nmap

Step 1: Log into Kali Linux and verify the environment.

- a. Log into the Kali system with the username **kali** and the password **kali**. You are presented with the Kali desktop.
- b. Open a terminal window.
- c. Verify that Kali has an interface in the 10.6.6.0/24 network using the **ifconfig** command.

```
kali@Kali: ~
File Actions Edit View Help
[(kali㉿Kali)-[~]]$ ifconfig
br-339414195aeb: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.5.5.1 netmask 255.255.255.0 broadcast 10.5.5.255
        inet6 fe80::42:56ff:fe48:ed98 prefixlen 64 scopeid 0x20<link>
            ether 02:42:56:48:ed:98 txqueuelen 0 (Ethernet)
            RX packets 35 bytes 1996 (1.9 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 18 bytes 2690 (2.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-355ee7945a88: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::42:aeff:fe6e:1ad9 prefixlen 64 scopeid 0x20<link>
            ether 02:42:ae:6e:1a:d9 txqueuelen 0 (Ethernet)
            RX packets 21 bytes 2192 (2.1 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 15 bytes 2496 (2.4 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-internal: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.6.6.1 netmask 255.255.255.0 broadcast 10.6.6.255
        inet6 fe80::42:bcff:fe9d:f0e5 prefixlen 64 scopeid 0x20<link>
            ether 02:42:bc:9d:f0:e5 txqueuelen 0 (Ethernet)
            RX packets 35 bytes 1996 (1.9 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 18 bytes 2690 (2.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        inet6 fe80::42:2dff:fe21:3ef1 prefixlen 64 scopeid 0x20<link>
            ether 02:42:2d:21:3e:f1 txqueuelen 0 (Ethernet)
            RX packets 9 bytes 900 (900.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 15 bytes 2467 (2.4 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fd17:625c:f037:2:a00:27ff:fe4a:f36e prefixlen 64 scopeid 0x0<global>
        inet6 fe80::a00:27ff:fe4a:f36e prefixlen 64 scopeid 0x20<link>
        inet6 fd17:625c:f037:2:8a35:ff78:2ff9:15ba prefixlen 64 scopeid 0x0<global>
            ether 08:00:27:4a:f3:6e txqueuelen 1000 (Ethernet)
            RX packets 11 bytes 3912 (3.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 30 bytes 4982 (4.8 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 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
```

- d. Use the **nmap -V** command to verify that Nmap is installed and to display the Nmap version.
The output will be similar to what is shown below.

```
(kali㉿Kali)-[~]
$ nmap -V
Nmap version 7.94 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.4.4 openssl-3.0.9 libssh2-1.10.0 libz-1.2.13 libpcre-8.39 libpcap-1.10.4 nmap-libdn
et-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select

(kali㉿Kali)-[~]
$
```

Step 2: Investigate Nmap Options and Features

- Using the command **nmap** without specifying any options or targets returns a list of commonly used Nmap options.

```
(kali㉿Kali)-[~]
nmap
Nmap version 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}

TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org; microsoft.com/24, 192.168.0.1; 10.0.0.255,1-254
  -sn: Scan for hosts online
  -PR: Scan for hosts online using ICMP ping
  -IR: Scan for hosts online using TCP/UDP/SCTP
  -Iw: Scan for hosts online using ICMP
  -O: Scan for hosts online using OS detection
  -sT: Scan for hosts online using TCP SYN/Connect()
  -sU: Scan for hosts online using UDP
  -sN: Scan for hosts online using Null, FIN, and Xmas scans
  --script-args <Flags>: Customize TCP scan flags
  -sI: zombie host;probeport>; Idle scan
  -sV: Scan for hosts online using TCP SYN/Connect()
  -sO: IP protocol scan
  -b <IP>: Perform a TCP Bounce scan
  PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
  -Ex: -p22; -p1-65535; -p U53,111,137,T:21-25,80,139,8080,S:9
  -exclude-ports <port ranges>: Exclude the specified ports from scanning
  -P<mode>: Scan mode (use random ports by default)
  -r: Scan ports sequentially; don't randomize
  --top-ports <n>: Scan <n> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
  SERVICE/VERSION DETECTION:
  -SV: Scan open ports to determine service/version info
  --version-intensity <level>: Set From 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
  SCRIPT SCAN:
  -Sc: equivalent to --script=default
  --script=<Lua script>: <Lua script> is a comma separated list of
  --script-directories: script-files or script-categories
  --script-args=<args>: arguments to pass to scripts
  --script-file=<filename>: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
```

To access the Nmap help system use the command **nmap -h**. The help output is divided into sections based on the type of detection that the option supports.

```

File Machine View Input Devices Help
File Actions Edit View Help
[~]# nmap -h
Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
Targets:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL <inputfilename>: Input from list of hosts/networks
-iR <range>: Choose random targets
--exclude <host1[,host2],host3...>: Exclude hosts/networks
--excludefile <excludefile>: Exclude list from file
HOST DISCOVERY:
-sL: List Scan - simply list targets to scan
-sA: Ping Scan - ping + port scan
-sP: Ping all hosts at once - skip host discovery
-PS/PA/PV/[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
-PO[protocol list]: IP protocol Ping
-N/-AR: DNS resolution/Always resolve [default: sometimes]
--dns-servers <server1[,server2,...>: Specify custom DNS servers
--system-dns: Use OS's DNS resolver
--traceroute: Trace hop path to each host
SCAN TECHNIQUES:
-S<script>: Scripted scans: TCP SYN/Connect() / ACK/Window/Maimon scans
-SU: UDP Scan
-SN/SF/SX: TCP Null, FIN, and Xmas scans
--scanflags <flags>: Customize TCP scan flags
-SI <zoneinfo>: Timezone report: Idle scan
-SV/A: TCP INIT/Connect/Echo scans
-SO: IP protocol scan
-b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
-p<ports>: Scan the specified ports
-Ex:<p22>,-p1-65335: Scan U-53,111,111,137,T-21-25,80,139,8080,S:9
--exclude-ports <port ranges>: Scan the specified ports from scanning
-F: Fast mode - Scan fewer ports than the default scan
-R: Scan ports sequentially - don't randomize
--top-ports <n>: Scan <n> most common ports
--port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
-sV: Probe open ports to determine service/version info
--version-intensity <level>: Scan intensity (try 9 (try all probes)
--version-all: Limit to most likely probes (intensity 2)
--version-trace: Try every single probe (Intensity 9)
--version-script: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
--scf <script>: equivalent to --script-default
--script [<script>]: clua script is a comma separated list of
--script-directories, script-files or script-categories
--script-args=<n>[,<n>2,...]: provide arguments to scripts
--script-args-file=<filename>: provide NSE script args in a file
--script-trace: Show all data sent and received

```

- b. The man page for Nmap provides additional information. To access the man page, enter the command **man nmap**. To exit the man pages, press **q** to quit and return to the terminal prompt.

```

File Machine View Input Devices Help
File Actions Edit View Help
NMAP(1)                                         Nmap Reference Guide                                         NMAP(1)
NAME                                              nmap - Network exploration tool and security / port scanner
SYNOPSIS
nmap [Scan Type ...] [Options] {target specification}
DESCRIPTION
Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the "interesting ports table". That table lists the port number and protocol, service name, and state. The state is either open, filtered, closed, or unfiltered. Open means that an application on the target machine is listening for connections/packets on that port. Filtered means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it is open or closed. Closed ports have no application listening on them, though they could open up at any time. Ports are classified as filtered if they are responsive to Nmap's probes, but Nmap cannot determine whether they are open or closed. Nmap reports the state combinations open|filtered and closed|filtered when it cannot determine which of the two states describes them. The port table may also include software version details when version detection has been requested. When an IP protocol scan is requested (-sO), Nmap provides information on supported IP protocols rather than listening ports.

In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.

A typical Nmap scan is shown in Example 1. The only Nmap arguments used in this example are -A, to enable OS and version detection, script scanning, and traceroute; -T4 for faster execution; and then the hostname.

Example 1. A representative Nmap scan
# nmap -A -T4 scanme.nmap.org

Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
Script scan report for 74.207.244.221: lib6-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh        OpenSSH_5.3p1 Debian 7 (protocol 2.0)
| ssh-hostkey: 1024:8d:60:f1:7c:a5:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldp
3720/tcp  filtered H.323/Q.931
975/tcp   filtered ping-echo
Nmap done: 1 IP address (1 host up) scanned in 11.39 seconds
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel2.6.39
OS details: Linux 2.6.39
Network Distance: 11 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

TRACEROUTE (using port 53/tcp)
HOP RTT     ADDRESS
Manual page nmap(1) line 3 (press h for help or q to quit)

```

Use the man page for Nmap to complete the table.

Option	Description
-A	Aggressive scan that enables OS detection, version detection, script scanning and traceroute
-O	Enables OS detection
-p <port ranges>	Allows for specific ports or port ranges to be scanned
-sF	Performs TCP FIN scan
-sn	Performs host discovery scan
-sS	Performs TCP SYN scan
-sT	Performs TCP Connect scan
-sV	Probes open ports to determine service/version info
-T<0-5>	Sets the timing of the scan. Higher numbers produce results faster. Slower scans elude detection better.
-v	Increases the verbosity of the output
--open	Only reports open (or possibly open) ports

Part 2: Perform Basic Nmap Scans

Step 1: Initiate a basic Nmap scan of the target computer.

- To quickly scan the DMZ for active hosts, you can perform a discovery scan. In a discovery scan, the scanning host sends an ICMP echo request (ping), a TCP SYN to port 443, a TCP ACK to port 80, and an ICMP timestamp request. A response to any of the requests indicates that the host is up and the IP protocol stack on the host is functioning. Enter the following command to scan the DMZ network:

```
(kali㉿Kali)-[~]
└─$ nmap -sn 10.6.6.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:27 UTC
Nmap scan report for 10.6.6.1
Host is up (0.00039s latency).
Nmap scan report for webgoat.vm (10.6.6.11)
Host is up (0.00021s latency).
Nmap scan report for juice-shop.vm (10.6.6.12)
Host is up (0.00017s latency).
Nmap scan report for dwa.vm (10.6.6.13)
Host is up (0.00014s latency).
Nmap scan report for mutillidae.vm (10.6.6.14)
Host is up (0.000070s latency).
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00050s latency).
Nmap scan report for 10.6.6.100
Host is up (0.00050s latency).
Nmap done: 256 IP addresses (7 hosts up) scanned in 7.83 seconds
└─$
```

The host 10.6.6.23 was identified as suspicious in a Wireshark capture, and it is necessary to perform additional reconnaissance to discover more about the computer and its services. Use the **nmap** command to execute a default scan on the target host.

```
(kali㉿Kali)-[~]
└─$ nmap 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:29 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00017s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
└─$
```

By default, Nmap performs a connect scan of 1000 most common TCP ports. This makes use of the operating system's networking software to establish a full TCP connection. This type of scan creates a lot of networking traffic and increases the probability of detection by intrusion detection services. You can also specify a TCP connect scan using the command option **nmap -sT**. The output of the connect scan includes the status codes shown below:

Status	Response Received	Interpretation
Open	TCP SYN-ACK	There is a service listening on the identified port.
Closed	TCP RST	There is no service listening on the identified port.
Filtered	No response, or an ICMP destination unreachable message received.	The port is being filtered by a firewall.

The **-O** option can be used to further determine information about the operating system running on the target host. Some Nmap options require additional permissions and must be run as **root** or using the **sudo** command. To find operating system information on the target host, use the **nmap -O** command. Enter the password of **kali** when prompted.

```
(kali㉿Kali)-[~]
└─$ sudo nmap -O 10.6.6.23
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:32 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.000040s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 02:42:0A:06:06:17 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
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 1.67 seconds

(kali㉿Kali)-[~]
└─$
```

Step 2: Obtain additional information about the host and services.

- To provide additional information about the target computer, it is possible to combine different options into a single command line. The previous command identified several potentially open ports on the 10.6.6.23 host. You can use **-v**, **-p**, and **-sV** to find additional information about the services running on the open ports. This command provides information about the FTP service running on port 21 on the target in verbose mode, with the timing set to fast (**-T4**):

```
(kali㉿Kali)-[~]
└─$ nmap -v -p21 -sV -T4 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:35 UTC
Nmap wishes you a merry Christmas! Specify -sX for Xmas Scan (https://nmap.org/book/man-port-scanning-techniques.html).
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 04:35
Scanning 10.6.6.23 [2 ports]
Completed Ping Scan at 04:35, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 04:35
Scanning gravemind.vm (10.6.6.23) [1 port]
Discovered open port 21/tcp on 10.6.6.23
Completed Connect Scan at 04:35, 0.00s elapsed (1 total ports)
Initiating Service scan at 04:35
Scanning 1 service on gravemind.vm (10.6.6.23)
Completed Service scan at 04:35, 0.00s elapsed (1 service on 1 host)
NSE: Script scanning 10.6.6.23.
Initiating NSE at 04:35
Completed NSE at 04:35, 0.00s elapsed
Initiating NSE at 04:35
Completed NSE at 04:35, 0.00s elapsed
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00017s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
Service Info: OS: Unix

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

The **-A** option executes OS detection, version detection, script scanning, and traceroute. The **-A** scan can be very intrusive and therefore will be detected by many IDS systems, so ensure that you have permission before attempting this scan outside of the lab environment. To gather more information regarding the FTP service, enter the command **nmap -p21 -sV -A 10.6.6.23**.

The sample detailed output of this command is shown below:

```
(kali㉿Kali)-[~]
└─$ nmap -p21 -sV -A 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:37 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00018s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|_ STAT:
|   FTP server status:
|     Connected to 10.6.6.1
|     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
|     At session startup, client count was 3
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--  1 0        0          16 Aug 13  2021 file1.txt
|_-rw-r--r--  1 0        0          16 Aug 13  2021 file2.txt
|_-rw-r--r--  1 0        0          29 Aug 13  2021 file3.txt
|_-rw-r--r--  1 0        0          26 Aug 13  2021 supersecretfile.txt
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 1.73 seconds
```

Examine the output of the **nmap -A** command. Notice that the output indicates that a connection was made between the Kali Linux system and the target FTP service. How many files on the FTP server are accessible through this connection? **4 text files**

What weakness in the FTP server configuration enabled the Kali Linux system to log into the FTP server? **The FTP server is configured to permit anonymous logins.**

Step 3: Investigate SMB Services with Scripts

The Server Message Block (SMB) protocol is a network file sharing protocol supported on Windows computers and by SAMBA on Linux. SMB enables applications to read and write files or request services over a network. Open public shares or shared devices such as print servers on a network, can be accessed through SMB.

- a. The earlier scan of open ports on the target computer indicates that the SMB ports 139 and 445 are open. Find more information on these ports using the **-A** and **-p** command options. The **-A** option executes several functions including running the default scripts. Specify more than one port to scan by listing them separately with a comma between them.

```
(kali㉿Kali)-[~]
└─$ nmap -A -p139,445 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:42 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00030s latency).

PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp    open  netbios-ssn  Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
Service Info: Host: GRAVEMIND

Host script results:
| smb2-time:
|   date: 2025-12-25T04:42:51
|   start_date: N/A
| smb2-security-mode:
|   3:1::1:
|     Message signing enabled but not required
|   clock-skew: mean: 0s, deviation: 1s, median: 0s
|   smb-security-mode:
|     account_used: guest
|     authentication_level: user
|     challenge_response: supported
|     message_signing: disabled (dangerous, but default)
|   smb-os-discovery:
|     OS: Windows 6.1 (Samba 4.9.5-Debian)
|     Computer name: gravemind
|     NetBIOS computer name: GRAVEMIND\x00
|     Domain name: \x00
|     FQDN: gravemind
|   System time: 2025-12-25T04:42:52+00:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.71 seconds

(kali㉿Kali)-[~]
└─$
```

Examine the information returned by the Nmap scan. From this information, it can be determined that the target computer is a member of the default workgroup, named WORKGROUP, and that SMB is supported on this host through SAMBA on Linux.

What is the NetBIOS computer name assigned to the target host? **GRAVEMIND\x00**

Nmap contains the powerful Nmap Scripting Engine (NSE), which enables the programming of various Nmap options and conditional actions to be taken as a result of the responses. NSE has built-in scripts that enumerate users, groups, and network shares. One of the more commonly used scripts for SMB discovery is the **smb-enum-users.nse** script. Use the Nmap NSE script with the command:

```

└─(kali㉿Kali)-[~]
$ nmap --script smb-enum-users.nse -p139,445 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:46 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00014s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Host script results:
| smb-enum-users:
|_ GRAVEMIND\arbiter (RID: 1001)
   Full name:
   Description:
   Flags:          Password not required, Account disabled, Normal user account
GRAVEMIND\masterchief (RID: 1000)
   Full name:
   Description:
   Flags:          Password not required, Account disabled, Normal user account

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
└─(kali㉿Kali)-[~]
$ 

```

Did the script uncover any SMB usernames on the target host? If so, how many? **Yes, 2 usernames: arbiter and masterchief.**

A serious security concern is the existence of publicly shared directories (folders). You can enumerate the network shares using another NSE script, **smb-enum-shares.nse**. To discover shared directories on the target computer. Use the Nmap share enumeration script with the command:

```

└─(kali㉿Kali)-[~]
$ nmap --script smb-enum-shares.nse -p445 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-25 04:48 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00016s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Host script results:
| smb-enum-shares:
|_ account_used: <blank>
  \\10.6.6.23\IPC$:
    Type: STYPE_IPC_HIDDEN
    Comment: IPC Service (Samba 4.9.5-Debian)
    Users: 1
    Max Users: <unlimited>
    Path: C:\tmp
    Anonymous access: READ/WRITE
  \\10.6.6.23\print$:
    Type: STYPE_DISKTREE
    Comment: Printer Drivers
    Users: 0
    Max Users: <unlimited>
    Path: C:\var\lib\samba\printers
    Anonymous access: READ/WRITE
  \\10.6.6.23\workfiles:
    Type: STYPE_DISKTREE
    Comment: Confidential Workfiles
    Users: 0
    Max Users: <unlimited>
    Path: C:\var\spool\samba
    Anonymous access: READ/WRITE

Nmap done: 1 IP address (1 host up) scanned in 7.71 seconds
└─(kali㉿Kali)-[~]
$ 

```

Examine the output created by the **smb-enum-shares** script. In the output, share names that end with a “\$” character represent hidden shares that include system and administrative shares. How many hidden shares were discovered on the target host? **2**

What serious security risk is uncovered in this script output? **Anonymous access: READ/WRITE.**

N/B

Nmap scans can be used to identify active devices on a network. The basic scans will uncover open ports and services that may need to be secured. Anonymous access to FTP files or network shares can be detected and corrected or limited. Malicious actors can use these same functions to find computers that may be vulnerable to attack.