

Experiment 1: Network Reconnaissance & Footprinting

Scenario:

An organization, "TechSecure Corp," suspects that its internal LAN might contain devices with unpatched services. As an external consultant with limited initial knowledge, your first step is to gain intelligence about the network. You have been given a subnet range and must map out devices and open ports.

Tasks:

- Use Nmap for host discovery, port scanning, and service enumeration.
- Employ Recon-ng or Amass for passive reconnaissance to discover hostnames, subdomains, or metadata.
- Document identified hosts, operating systems, and running services.

Deliverable:

A network inventory report listing IP addresses, OS guesses, and active services.

Reconnaissance: Active vs Passive

- Active Reconnaissance involves directly interacting with the target system by sending packets or requests (e.g., Nmap scans). This helps discover live hosts, open ports, services, and OS but can alert the target system of the scan.
- Passive Reconnaissance involves gathering information without directly interacting with the target system (e.g., using tools like Amass or Recon-ng). This method relies on publicly available data like DNS records and websites and is less likely to be detected.

TASK:1

Active Reconnaissance with Nmap

Nmap (Network Mapper) is a free and beginner-friendly tool used to scan networks. It helps you find devices (hosts), check which ports are open, what services are running (like web or FTP servers), and even guess the operating system. It's widely used in cybersecurity for network discovery and vulnerability checks.

Metasploitable 2

In this experiment, we are using Metasploitable 2 as the target for active reconnaissance. Metasploitable 2 is a purposely vulnerable machine that provides a safe environment for practicing penetration testing and vulnerability scanning. Active reconnaissance helps us discover the live devices, open ports, services, and operating systems on the Metasploitable 2 machine.

Steps for Active Reconnaissance:

1.Host Discovery:

Command:

```
nmap -sn <metasploitable_IP>
```

-sn:

- Stands for Ping Scan (does not scan ports).
- It checks if a device is online without scanning ports.

Output:

```
(kali㉿kali)-[~]
$ nmap -sn 192.168.140.12
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-06 11:13 EDT
Nmap scan report for 192.168.140.12
Host is up (0.011s latency).
MAC Address: 08:00:27:C8:28:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
```

2. Port Scanning:

Command:

```
nmap -sS <metasploitable_IP>
```

-sS:

- Stands for SYN scan, also known as a half-open scan.
- It sends a request to check if a port is open without fully connecting, making it faster and harder to detect.

Output:

```
(kali㉿kali)-[~]
$ nmap -sS 192.168.140.12
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-06 11:16 EDT
Nmap scan report for 192.168.140.12
Host is up (0.0064s 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: 08:00:27:C8:28:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

3. Service Enumeration:

Command:

```
nmap -sV <metasploitable_IP>
```

-sV:

- Used for service version detection.
- Detects the versions of services running on open ports.

Output:

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.140.12
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-06 11:17 EDT
Nmap scan report for 192.168.140.12
Host is up (0.0022s 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        login
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: 08:00:27:C8:28:ED (PC Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.76 seconds
```

4. OS Detection:

Command:

```
nmap -O <metasploitable_IP>
```

-O:

- Used to detect the operating system of the target machine.

Output:

```
(kali㉿kali)-[~]
└─$ nmap -O 192.168.140.12
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-06 11:18 EDT
Nmap scan report for 192.168.140.12
Host is up (0.0072s 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: 08:00:27:C8:28:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
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 2.00 seconds
```

TASK:2

Passive Reconnaissance with Amass

Amass is an open-source reconnaissance tool widely used in cybersecurity to discover subdomains, DNS records, and related metadata about a target domain. It performs passive information gathering by querying public sources such as search engines, SSL certificates, APIs, and DNS databases. Since it does not directly interact with the target system, it is stealthy and helps avoid detection. Amass is especially useful in the early stages of penetration testing, allowing security professionals and beginners to map the external structure of an organization's network safely.

In this task, we use Amass to discover:

- Hostnames
- Subdomains
- Metadata (like WHOIS, DNS records)

Command:

```
amass enum -passive -d demo.testfire.net
```

or

```
amass enum -d example.com
```

enum:

- Short for “enumerate” — it tells Amass to start finding subdomains.

-d example.com:

- The -d option means "domain", so you're telling Amass to search for subdomains of example.com.

Output:

```
└─(kali㉿kali)-[~]
$ amass enum -passive -d demo.testfire.net
testfire.net (FQDN) → a_record → 65.61.137.117 (IPAddress)
testfire.net (FQDN) → node → demo.testfire.net (FQDN)
demo.testfire.net (FQDN) → a_record → 65.61.137.117 (IPAddress)
65.61.136.0/22 (Netblock) → contains → 65.61.137.117 (IPAddress)
33070 (ASN) → managed_by → RMH-14 - Rackspace Hosting (RIROrganization)
33070 (ASN) → announces → 65.61.136.0/22 (Netblock)

The enumeration has finished
```

```
└─(kali㉿kali)-[~]
$ amass enum -d example.com
example.com (FQDN) → ns_record → a.iana-servers.net (FQDN)
example.com (FQDN) → ns_record → b.iana-servers.net (FQDN)
example.com (FQDN) → a_record → 23.215.0.136 (IPAddress)
example.com (FQDN) → a_record → 96.7.128.175 (IPAddress)
example.com (FQDN) → a_record → 96.7.128.198 (IPAddress)
example.com (FQDN) → a_record → 23.192.228.80 (IPAddress)
example.com (FQDN) → a_record → 23.192.228.84 (IPAddress)
example.com (FQDN) → a_record → 23.215.0.138 (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1406:bc00:53::b81e:94ce (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1408:ec00:36::1736:7f31 (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1406:3a00:21::173e:2e65 (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1406:3a00:21::173e:2e66 (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1406:bc00:53::b81e:94c8 (IPAddress)
example.com (FQDN) → aaaa_record → 2600:1408:ec00:36::1736:7f24 (IPAddress)
b.iana-servers.net (FQDN) → a_record → 199.43.133.53 (IPAddress)
b.iana-servers.net (FQDN) → aaaa_record → 2001:500:8d::53 (IPAddress)
23.215.0.0/22 (Netblock) → contains → 23.215.0.136 (IPAddress)
23.215.0.0/22 (Netblock) → contains → 23.215.0.138 (IPAddress)
96.7.128.0/23 (Netblock) → contains → 96.7.128.175 (IPAddress)
96.7.128.0/23 (Netblock) → contains → 96.7.128.198 (IPAddress)
23.192.228.0/22 (Netblock) → contains → 23.192.228.80 (IPAddress)
23.192.228.0/22 (Netblock) → contains → 23.192.228.84 (IPAddress)
199.43.133.0/24 (Netblock) → contains → 199.43.133.53 (IPAddress)
2001:500:8d::/48 (Netblock) → contains → 2001:500:8d::53 (IPAddress)
20940 (ASN) → managed_by → AKAMAI-ASN1 (RIROrganization)
20940 (ASN) → announces → 23.215.0.0/22 (Netblock)
20940 (ASN) → announces → 96.7.128.0/23 (Netblock)
20940 (ASN) → announces → 23.192.228.0/22 (Netblock)
26710 (ASN) → managed_by → ICANN-ANYCASTED-SERVICES - ICANN (RIROrganization)
26710 (ASN) → announces → 199.43.133.0/24 (Netblock)
26710 (ASN) → announces → 2001:500:8d::/48 (Netblock)
example.com (FQDN) → node → www.example.com (FQDN)
www.example.com (FQDN) → cname_record → www.example.com-v4.edgesuite.net (FQDN)
2600:1406:bc00::/48 (Netblock) → contains → 2600:1406:bc00:53::b81e:94ce (IPAddress)
2600:1406:bc00::/48 (Netblock) → contains → 2600:1406:bc00:53::b81e:94c8 (IPAddress)
2600:1408:ec00::/48 (Netblock) → contains → 2600:1408:ec00:36::1736:7f31 (IPAddress)
2600:1408:ec00::/48 (Netblock) → contains → 2600:1408:ec00:36::1736:7f24 (IPAddress)
2600:1406:3a00::/48 (Netblock) → contains → 2600:1406:3a00:21::173e:2e66 (IPAddress)
2600:1406:3a00::/48 (Netblock) → contains → 2600:1406:3a00:21::173e:2e65 (IPAddress)
20940 (ASN) → announces → 2600:1406:bc00::/48 (Netblock)
20940 (ASN) → announces → 2600:1408:ec00::/48 (Netblock)
20940 (ASN) → managed_by → Akamai International B.V. (RIROrganization)
20940 (ASN) → announces → 2600:1406:3a00::/48 (Netblock)

The enumeration has finished
```

Network Inventory: Open Ports & Services:

Port	Service	Description
21/tcp	FTP	File Transfer Protocol
22/tcp	SSH	Secure Shell
25/tcp	SMTP	Simple Mail Transfer Protocol
53/tcp	DNS	Domain Name System
80/tcp	HTTP	HyperText Transfer Protocol
139/tcp	NetBIOS-SSN	NetBIOS Session Service
445/tcp	Microsoft-DS	Microsoft Directory Services
3306/tcp	MySQL	MySQL Database Server
5432/tcp	PostgreSQL	PostgreSQL Database Server
5900/tcp	VNC	Virtual Network Computing

Conclusion:

This experiment involved performing active and passive reconnaissance to map out the network. Nmap was used for host discovery, port scanning, and service enumeration, while Amass was employed for passive reconnaissance to uncover subdomains and metadata. The results, including IP addresses, operating systems, and active services, were compiled into a network inventory report, aiding in the identification of potential security risks posed by unpatched services.
