1. RECONNAISSANCE (Information Gathering)

Reconnaissance is the **first and most critical phase** in cybersecurity. It involves collecting information about a target before attempting any attack or security assessment. It prepares you to understand the target's structure, weaknesses, and potential attack surfaces.

Reconnaissance is divided into two types:

1.1 Passive Reconnaissance

What is Passive Recon?

Passive reconnaissance means *collecting information about the target without interacting with it directly*. You do not send any packets to the target. This avoids detection and keeps your presence hidden.

Why is it Important?

- Helps understand the organization's structure.
- Reveals public information attackers can misuse.
- Cannot be easily detected by the target.

Techniques Used

---

1. WHOIS Lookup

WHOIS gives publicly registered details about a domain.

You can find:

- Domain owner
- Organization name
- Country
- Registrar
- IP ranges
- Technical contact emails

Command:

```
whois example.com
```

---

2. Nslookup

Used to fetch DNS records of a domain.

You learn:

- IP address
- Mail servers (MX)
- Name servers (NS)
- CNAME records

Command:

```
nslookup example.com
```

---

3. Google Dorking

Advanced search queries to extract sensitive information.

Examples:

```
site:example.com filetype:pdf
intitle:"index of" confidential
```

What you may find:

- Public PDFs
- Config files
- Exposed data
- Login pages

---

4. Shodan Search

Shodan is called *"Google for hackers."*
It scans all internet-connected devices.

You can discover:

- Open ports
- Cameras
- Servers
- Firewalls
- Weak/old software versions

Example query:

```
apache country:IN
```

What You Learned From Passive Recon

- How big companies expose information unintentionally.
- Why DNS, registrar data, and public files matter.
- How attackers identify targets without touching them.

1.2 Active Reconnaissance

What is Active Recon?

Active reconnaissance means *direct interaction with the target.*
You send packets to the target, which means you can be detected.

1. Ping Sweep

Used to find **which hosts are alive** in a network.

Example tools:

```
fping -a -g 192.168.1.0/24
```

You learn:

- Which machines are online
- Response times
- Network topology

---

2. Banner Grabbing

This extracts service information such as:

- Software name
- Version number
- Operating system

Attackers use this to target vulnerabilities.

Methods:

```
nc -v target_ip 80
curl -I http://target_ip
```

What You Learned From Active Recon

- How real attackers identify live systems
- How service banners leak sensitive details
- How reconnaissance enables future attacks

2. PORT & SERVICE SCANNING (Nmap)

Port scanning identifies *what services a target is running* and *which ports are open.*
Nmap is the most powerful tool for this.

Why Port Scanning Is Critical

- Shows entry points into a system
- Reveals running services
- Helps detect vulnerabilities
- Helps in OS detection

Nmap Scans You Performed

---

1. TCP SYN Scan (Stealth Scan)

This is the most commonly used scan.
It sends only the SYN packet and avoids full connection.

Command:

```
nmap -sS target_ip
```

What you learn:

- Open ports
- Closed ports
- Filtered ports

This scan is fast and harder to detect.

---

2. UDP Scan

UDP services do not respond like TCP; these scans are slower.

Command:

```
nmap -sU target_ip
```

Identifies:

- DNS (53)
- DHCP (67/68)
- SNMP (161)
- TFTP (69)

---

3. Service Version Detection

Finds the exact version of each service.

Command:

```
nmap -sV target_ip
```

Example output:

- Apache 2.4.41
- OpenSSH 7.9
- MySQL 5.7

This is crucial for vulnerability exploitation.

---

4. OS Detection

Nmap compares packet responses to a huge database.

Command:

```
nmap -O target_ip
```

It attempts to identify:

- Linux or Windows
- Kernel version
- Device type

---

5. Aggressive Scan

Runs multiple scans in one command.

Command:

```
nmap -A target_ip
```

Includes:

- OS detection
- Version detection
- Script scanning
- Traceroute

What You Learned From Nmap

- How attackers identify service versions
- How open ports reveal attack surfaces
- Why outdated software is dangerous
- How Nmap scripts automate vulnerability checks

3. VULNERABILITY SCANNING (OpenVAS / Nessus)

Vulnerability scanners automate the process of identifying weaknesses.

You scanned Metasploitable2, a purposely vulnerable system used for learning.

What the Scanner Does

- Checks for outdated software
- Detects misconfigurations
- Finds weak passwords
- Flags dangerous services
- Assigns severity using CVSS scores

Vulnerability Levels

Severity	Meaning
Critical	Can fully compromise the system
High	Major security risk
Medium	Needs attention
Low	Minor issues

Process You Followed

1. Setup Scanner

- Updated vulnerability feeds
- Started services
- Added target IP

---

2. Initiated Full Scan

The scanner automatically:

- Sent probes
- Checked services
- Matched vulnerabilities

---

3. Analyzed Report

You exported and studied the PDF.

Learned:

- Vulnerability descriptions
- Impact
- Exploitation methods
- Recommended fixes

What You Learned

- How automated scanners work
- Why vulnerability assessment is essential
- Real-world examples of critical flaws
- Importance of patching software

4. PACKET ANALYSIS WITH WIRESHARK

Wireshark is the most powerful packet capturing tool, used by cybersecurity professionals.

What You Did

1. Capture HTTP Traffic

HTTP is unencrypted.

You observed:

- URL requests
- Cookies

- Form data
- Server responses

This shows why HTTP is insecure.

---

2. FTP Packet Analysis

FTP sends credentials in plain text.

Using filter:

```
ftp
```

You captured:

- USER command
- PASS command

This demonstrated how attackers steal credentials.

---

3. DNS Analysis

DNS resolves domain names.

Filter:

```
dns
```

You observed:

- Queries
- Responses
- TTL values
- IP resolutions

Understanding DNS traffic helps detect:

- DNS spoofing
- DNS tunneling

---

4. SYN Flood Attack Analysis

Used `hping3` to simulate an attack.

Command:

```
hping3 -S --flood -p 80 target_ip
```

In Wireshark:

```
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

You saw:

- Huge number of SYN packets
- No ACK responses

This imitates how attackers overload servers.

What You Learned

- How insecure protocols leak sensitive information
- How network attacks look in real time
- How packet filtering reveals abnormal behavior

---

5. FIREWALL BASICS (iptables)

A firewall controls incoming and outgoing packets based on rules.

`iptables` is the Linux firewall.

What You Did

1. Allow SSH Traffic

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Allows only SSH connections.

---

2. Block HTTP Traffic

```
sudo iptables -A INPUT -p tcp --dport 80 -j DROP
```

Blocks anyone trying to access HTTP.

3. Block ICMP (Ping)

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Helps protect from network discovery.

4. Block NULL Scans

```
sudo iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

NULL scans are stealthy reconnaissance methods.

5. Check Rules

```
sudo iptables -L -n -v
```

Allows verification.

What You Learned

- How packet filtering protects systems
- How attackers use scanning techniques

- How firewall rules prevent attacks
- Why rule order matters