Week 1 – Foundations of Ethical Hacking

Concepts: Ethics, Linux CLI, Networking basics, Lab Setup

Day 1: Introduction to Ethical Hacking

- **Definition:** Ethical hacking is the practice of legally breaking into computers and devices to test an organization's defenses.

Day 2: Linux Command Line Basics

Commands & Steps:

- **List files:** ls -la
- **Change directory:** cd /path/to/folder
- **Change file permission (make executable):** chmod +x script.sh
- **Check processes:** ps aux | grep firefox
- **Kill a process by PID:** kill 1234

Day 3: Networking Fundamentals

- **Definition:** Understanding the OSI model and TCP/IP protocols which form the foundation of network communications.

Day 4: Setting Up Lab Environment

- **Steps:**
- Download and install VirtualBox or VMware.
- Download Kali Linux ISO.
- Create a new VM and mount the Kali ISO.
- Follow Kali installation wizard.

Day 5: Linux Networking Commands Commands & Steps

:

- Check IP address: ifconfig
- Show network connections and listening ports: netstat -tulnp
- Ping example.com to check connectivity: ping example.com
- Trace route to google.com: traceroute google.com

Day 6: TCP/IP Protocols Deep Dive

- Use Wireshark (GUI tool) to capture packets.

Steps:

1. Open Wireshark.
2. Select network interface.
3. Click Start Capture.
4. Filter by protocol: type tcp or icmp in filter box.

Day 7: Review & Practice

- Practice above commands on Kali Linux VM.

Week - 2 Network Scanning and Enumeration

- **Concepts:** Nmap, Wireshark, Banner Grabbin

Day 1: Introduction to Nmap Commands:

Basic nmap commands

Basic nmap commands	Discription
nmap 192.168.1.1	Scan a single IP address.
nmap 192.168.1.1 192.168.1.2	Scan multiple IPs.
nmap 192.168.1.0/24	Scan a subnet (CIDR notation).
nmap -iL targets.txt	Scan a list of IPs from a file.
nmap -sP 192.168.1.0/24	Ping scan (check which hosts are up).
nmap -sn 192.168.1.0/24	Same as above (modern syntax).
nmap -p 80 192.168.1.1	Scan specific port.
nmap -p 1-1000 192.168.1.1	Scan a range of ports.
nmap -v 192.168.1.1	Increase verbosity (show more info).

Advanced nmap commands

command	decription
nmap -sS 192.168.1.1	SYN scan (stealth scan).
nmap -sT 192.168.1.1	TCP connect scan.
nmap -sU 192.168.1.1	UDP scan.
nmap -sV 192.168.1.1	Detect service versions.
nmap -O 192.168.1.1	OS detection.
nmap -A 192.168.1.1	Aggressive scan (includes -O, -sV, traceroute, and more).

Day 2: Wireshark Packet Capture

Wireshark is a **network protocol analyzer** used to capture and inspect packets in real time. It's widely used by ethical hackers, network engineers, and cybersecurity professionals.

Common filters to use

Filter	Purpose
ip.addr == 192.168.1.1	Filter packets to/from IP
tcp.port == 80	Show only HTTP traffic
http	Filter HTTP packets
dns	Filter DNS requests/responses
icmp	View ping/echo packets
tcp / udp	View only TCP or UDP

Day 3: Banner Grabbing

Banner Grabbing is a technique used to gather information about a service running on an open port of a system. It helps ethical hackers identify

- The **software name & version**
- Potential **vulnerabilities**
- Configuration details

Basic banner grabbing commands:

Tool	Purpose	Command Style
nc (Netcat)	Manual banner grabbing	Simple & fast
telnet	Older tool, still works	Text-based
Nmap	Scriptable banner grabbing	Powerful & flexible
curl	Web banners & headers	Lightweight

Day 4: Masscan High-Speed Scan

Masscan is a high-performance TCP port scanner, similar to Nmap but **much faster**. It uses its own TCP/IP stack and is designed for **massive-scale scanning**.

Command:

```
masscan 192.168.1.0/24 -p0-65535 --rate=1000
```

Day 5 Vulnerability scanning

- **Vulnerability scanning** is the process of **automatically identifying weaknesses** in a system, network, or application that could be exploited by attackers.

Tools	Level	Purpose
Nmap	Beginner	Open ports & service discovery
Nikto	Beginner	Web server vulnerability scanner
OpenVAS	Advanced	Full vulnerability assessment
Nessus	Advanced	Commercial vulnerability scanner
SQLmap	Web	SQL injection vulnerabilities
LinEnum	PrivEsc	Local privilege escalation checks

Day 6: Review & Practice

- Combine scanning commands to map network.

Week-3 Web Applcation security

Concepts: OWASP, SQLi, XSS, CSRF, Burp Suite

Day -1 Open Web Application Security Project

- **OWASP (Open Web Application Security Project)** is a nonprofit organization that provides free tools, resources, and documentation to help improve the security of web applications.
- 🛠️ **Tools to Test OWASP Vulnerabilities**
- **Burp Suite** – Intercept and manipulate requests
- **OWASP ZAP** – Free security scanner
- **SQLmap** – SQL Injection
- **Nikto** – Server misconfigurations
- **DVWA** – Practice platform for OWASP Top 10
- **OWASP Juice Shop** – Gamified OWASP training app

Continued…

📌 Examples of OWASP Vulnerabilities

Vulnerability	Real-life Example
SQL Injection	admin' OR 1=1 -- lets attacker bypass login
XSS	<script>alert('Hacked!')</script> injected into input
Broken Auth	Accessing /admin page without login
Misconfiguration	Admin panel exposed at example.com/admin with default creds

Continued…

The **OWASP Top 10** is a list of the **10 most common and critical security vulnerabilities** found in web apps, updated regularly.

OWASP Top 10 – 2021 Edition (Latest)

Rank	Vulnerability	Description
1	**Broken Access Control**	Improper restriction of user permissions (e.g., accessing admin functions without being admin).
2	**Cryptographic Failures**	Weak or missing encryption for sensitive data.
3	**Injection**	Attacker injects code (like SQL, OS, or LDAP) into a vulnerable input field.
4	**Insecure Design**	Flaws in how the app is architected, allowing potential abuse.
5	**Security Misconfiguration**	Default passwords, error messages revealing too much info, misconfigured headers, etc.
6	**Vulnerable and Outdated Components**	Using outdated libraries or plugins that have known vulnerabilities.
7	**Identification and Authentication Failures**	Issues with login systems (e.g., broken multi-factor auth, brute-forceable logins).

Day – 2 SQL Injection

- **What is SQL Injection?**
- **SQL Injection** is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It often lets attackers:
- View sensitive data
- Bypass authentication
- Modify or delete data
- Execute administrative operations on the database

Continued…

Types of SQL Injection

Type	Description
In-band	Use the same communication channel (e.g., classic UNION and OR 1=1).
Inferential (Blind)	No data shown directly, but attacker deduces information from behavior.
Out-of-band	Uses different channels to exfiltrate data (e.g., DNS, HTTP requests).

continued…

C Tools for SQL Injection

Tool	Command Example
sqlmap	Automates SQLi exploitation
Burp Suite	Intercepts and modifies requests
Havij (GUI)	Deprecated, but used in older cases

SQL USING

Step 1: Basic ScanbashCopyEditsqlmap –u
"http://example.com/page.php?id=1" --batch📌
Step 2: List Available DatabasesbashCopyEditsqlmap -u
"http://example.com/page.php?id=1" --dbs📌
Step 3: Dump Tables from a DatabasebashCopyEditsqlmap -u
"http://example.com/page.php?id=1" -D users --tables
Step 4: Dump Data from a TablebashCopyEditsqlmap -u
"http://example.com/page.php?id=1" -D users -T accounts --dump

Continued…

- 🔒 **How to Prevent SQL Injection**
- Use **Prepared Statements (Parameterized Queries)**.
- Validate and sanitize user input.
- Use **ORMs** (Object Relational Mappers).
- Enable Web Application Firewalls (WAF).
- Avoid dynamic SQL queries.
- 🧪💻 **Practice Platforms**
- DVWA
- bWAPP
- OWASP Juice Shop

Day 3: Cross-Site Scripting (XSS)

📘 What is XSS?

Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applications. It allows attackers to inject **malicious scripts** into webpages viewed by other users.

⚠️ These scripts are usually JavaScript, and they can steal cookies, session tokens, or perform actions on behalf of the user.

Types of XSS

Type	Description
Reflected XSS	Script is reflected in the URL and executed immediately.
Stored XSS	Script is permanently stored (e.g., in database) and runs every time the page loads.
DOM-based XSS	The vulnerability exists in client-side JavaScript logic (DOM manipulation).

Continued…

tooles for XSS testing

Tool	Use
Burp Suite	Intercept and modify HTTP requests.
XSStrike	Automated XSS scanner.
OWASP ZAP	XSS scanner and proxy.

Prevention method

Prevention Method	Description
Input Validation	Never trust user input. Validate strictly.
Output Encoding	Encode special characters (e.g., &, <, >, ', ").
Content Security Policy (CSP)	Helps block inline scripts.
Sanitization Libraries	Use tools like DOMPurify to sanitize input.

.

Day 4: CSRF Testing

What is CSRF ?

Cross-Site Request Forgery (CSRF) is an attack that forces a logged-in user to perform unwanted actions on a web. application, without their consent.

🔒 CSRF tricks the browser into sending a **legitimate request** to a site where the user is authenticated, like changing a password or transferring money.

How CSRF Works (Concept)

2.Attacker sends a malicious request (e.g., via a hidden form) to bank.com.

3.The browser automatically includes session cookies in the request.

The bank thinks the request is valid. □ **How to Test for CSRF (Manual Steps)**

1.User logs into bank.com and the session is stored in cookies.

🔑 Using Burp Suite:

1.Open Burp Suite → Turn **Proxy ON**.

2.Login to the target site and perform an action (like changing email or password).

3.Burp intercepts the POST request.

4.Right-click the request → Select **"Engagement tools" > "Generate CSRF PoC"**.

5.Copy the HTML form generated.

6.Paste it into a local .html file and open it in the browser.

7.If the action happens without any login prompt, it's **vulnerable**.

Continued…

Example: CSRF Attack via HTML Form

```
<form action="http://bank.com/transfer" method="POST">
  <input type="hidden" name="amount" value="1000">
  <input type="hidden" name="to" value="attacker">
  <input type="submit" value="Click me">
</form>
```

Usefull tooles forCSRF

Tool	Description
Burp Suite	Auto-generate PoC forms for testing.
OWASP ZAP	Scans for missing CSRF tokens.
Postman	For manually testing request behavior.

Day 5: Insecure Direct Object Reference (IDOR)

- **consept**
- Insecure Direct Object Reference (IDOR) is a type of **access control vulnerability**. It occurs when an application exposes internal objects (like database records, files, or user IDs) directly in a URL or request **without proper authorization checks**.

Definition

IDOR allows attackers to manipulate object references (like id=1) to gain
unauthorized access to data or functions meant for other users.

Day 6: Burp Suite Tools

Concept

Burp Suite is a powerful all-in-one tool for web application penetration testing. It intercepts, analyzes, modifies, and automates HTTP requests and responses.

Definition

Burp Suite acts as a **man-in-the-middle proxy** between the browser and web server. It allows hackers to **intercept**, **inspect**, and **manipulate** requests/responses to discover vulnerabilities.

Basic Setup Steps

1. **Start Burp Suite** → choose Temporary or Project file.
2. Set **browser proxy** to 127.0.0.1:8080.
3. Install **Burp CA certificate** in your browser.
4. Enable **Intercept** to view and modify requests.

Continued...

Use Repeater to Test Requests

Steps:

1.Right-click a request → “Send to Repeater”.
2.Modify parameters (e.g., id=1 → id=2).
3.Click “Send” → View and analyze response.

Use Intruder for Fuzzing

Steps:

1.Right-click → "Send to Intruder".
2.Set attack positions (e.g., login fields).
3.Load a wordlist (e.g., rockyou.txt).

4.Start the attack → Analyze successful attempts.

Continued…

Use Case Example

- Testing login forms with **Intruder**
- Checking XSS or SQLi payloads with **Repeater**
- Capturing session hijack attempts using **Proxy**

🔐 Tip: Use With DVWA or Juice Shop

Practice Burp Suite tools on **vulnerable apps** like:

- DVWA (Damn Vulnerable Web App)
- OWASP Juice Shop

Day 7: Review & Practice

Practice on DVWA or OWASP Juice Shop.

Week 4 – Exploitation and Post-Exploitation

- **Concepts**: Metasploit, Privilege Escalation

Day 1: Metasploit Basics

- **Concept:**
- Metasploit Framework is a powerful tool for developing, testing, and executing exploits against target machines.

Setup Requirements:

- **Kali Linux** (comes with Metasploit)
- **Target VM** like **Metasploitable2**

Continued…

Basic Metasploit commands

Command	Purpose
msfconsole	Starts the Metasploit Framework console
help	Lists available commands and their usage
search <name>	Searches for exploits, payloads, etc.
use <module>	Selects a specific exploit module
info	Shows detailed info about the selected module
show options	Lists options required for the module
set <OPTION> <value>	Sets a value for a module option (e.g., set RHOST 192.168.1.10)
unset <OPTION>	Unsets a previously set option
exploit or run	Launches the exploit
back	Deselects the current module
exit or quit	Exits the Metasploit console

Continued…

Intermediate Metasploit Commands

Command	Purpose
show exploits	Lists all available exploits
show payloads	Lists available payloads
show auxiliary	Shows auxiliary (scanner, fuzzing, etc.) modules
show post	Shows post-exploitation modules
set PAYLOAD <payload>	Sets a specific payload for the exploit
set LHOST <IP>	Sets the local host IP (for reverse shells)
set LPORT <port>	Sets the local port for connections
check	Tests if the target is vulnerable before exploitation
sessions	Lists active sessions (after successful exploit)
sessions -i <id>	Interacts with a specific session

Continued...

Advanced Metasploit Commands

Command	Purpose
use auxiliary/scanner/portscan/tcp	Uses a TCP port scanning module
use exploit/multi/handler	Used to handle payloads like reverse shells
route add <subnet> <session>	Adds a route through a pivot session
db_nmap <options>	Scans target using Nmap and imports into database
creds	Lists gathered credentials from sessions
loot	Displays collected loot (e.g., hashes, tokens)
exploit -j	Runs exploit as a background job
sessions -k <id>	Kills a specific session
resource <file.rc>	Runs a script of MSF commands (automation)
setg <OPTION> <value>	Sets a global option (applies to all modules)
save	Saves current environment and settings

Day 2: Exploiting Metasploitable and Privilege Escalation

- Use Metasploit to exploit known vulnerabilities on Metasploitable VM.
- And privilege escalation

Day 3: Writing Exploit Scripts Basic Python reverse shell

- Use reverse shell already you get it

Persistence and Cleanup and

> Create a cron job for persistence (Linux):

Bash

Review & Practice

- Practice exploits and escalation on test environments.

Must know in this topic

What is Bettercap?

Bettercap is a **Man-in-the-Middle (MITM)** framework that allows you to:

•Intercept and monitor network traffic.
•Spoof ARP, DNS, and HTTPS.
•Sniff passwords and cookies (if not encrypted).
•Run wireless attacks (like WiFi jamming or fake APs).
•Scan and monitor all devices on the network.

✅ It's like a modern replacement for older tools like Ettercap — but much more powerful.

Continued…

- what you can do with bettercup

Use Case	Example
MITM attacks	Intercept a target’s traffic
ARP spoofing	Trick network into sending traffic to you
DNS spoofing	Redirect a domain to fake IP
Sniff credentials	Catch usernames/passwords over HTTP
Wi-Fi recon & jamming	Scan Wi-Fi devices and jam or clone
Device discovery	Find hosts in the network

Continued…

1. Install Bettercap:
bash
sudo apt update sudo apt install bettercap
2. Run Bettercap:
bash
sudo bettercap -iface wlan0 # or eth0 depending on your connection
3. Inside Bettercap CLI, use commands:
bash
net.probe on # Discover live hosts set arp.spoof.targets 192.168.1.23 # Replace with victim IP
arp.spoof on # Start MITM attack net.sniff on # Start sniffing packets
You'll now see DNS queries, visited domains, and possibly some unencrypted data.

Continued

Extra modules you can use

Module	What it Does
net.sniff	**Packet sniffing**
dns.spoof	**DNS hijacking (redirect domains)**
http.proxy	**HTTP interception proxy**
wifi.recon	**Wi-Fi scanning nearby networks**
wifi.ap	**Create fake access points**

Week 5 – Wireless & Password Attacks

- **Concepts:** Wireless security, password cracking

Day 1: Wireless Security Protocols

What are Wireless Security Protocols?

- These are encryption methods and standards used to **secure Wi-Fi** communication between devices and routers.

Main protocol you should to know

Protocol	Full Name	Status	Description
WEP	Wired Equivalent Privacy	✕ Obsolete	Weak, easily cracked
WPA	Wi-Fi Protected Access	⚠ Weak	Improved from WEP but still vulnerable
WPA2	WPA version 2	✓ Secure (until WPA3)	Uses AES encryption, widely used

Continued…

Key differences(simplified)

Feature	WEP	WPA	WPA2	WPA3
Encryption	RC4	TKIP	AES	AES-GCMP
Key Strength	Weak	Medium	Strong	Very Strong
Cracking Time	Seconds	Minutes	Hours (if weak password)	Nearly impossible (unless misconfigured)
Status	Retired	Legacy	Active	Modern

Continued…

- Security issues:

- **WEP:** Can be cracked in seconds with tools like aircrack-ng.
- **WPA:** Vulnerable to dictionary and replay attacks.
- **WPA2 (with PSK):** Weak passwords can still be cracked.
- **WPA3:** Strong but newer, and not supported on all routers/devices.

Day 2: WiFi Recon with Airodump-ng

Goal: Put your WiFi card into monitor mode and scan nearby wireless networks and clients.

Step 1: Start monitor mode

bash

airmon-ng start wlan0

- wlan0 is your wireless adapter's default interface.
- Monitor mode allows you to listen to all wireless traffic, not just your own.
- After this command, your interface will change to something like wlan0mon.

Step 2: Scan for networks and clients

bash

airodump-ng wlan0mon

- wlan0mon is the monitor-mode interface.
- This command shows a live list of wireless Access Points (APs) and connected clients.
- Information displayed includes:
    - **BSSID**: MAC address of AP.
    - **ESSID**: Network name.
    - **Channel**: Wireless channel AP is on.
    - **Encryption**: e.g., WPA2.
    - **Clients**: Devices connected to each AP.

Day 3: Capturing Handshake & Cracking

Goal: Capture a WPA handshake, then crack the password offline.

Step 1: Focus on the target AP

bash

airodump-ng --bssid <BSSID> -c <CHANNEL> -w capture wlan0mon

- Replace <BSSID> with the AP's MAC address from the previous scan.
- Replace <CHANNEL> with the AP's wireless channel.
- -w capture saves the traffic to files starting with "capture".
- This filters to only listen to your target AP and its clients.

Continued…

Step 2: Capture the handshake

•When a client connects/reconnects to the AP, a 4-way handshake is captured.

•You can speed this up by sending a deauthentication packet to force reconnect:

bash

aireplay-ng --deauth 10 -a <BSSID> wlan0mon

•--deauth 10: sends 10 deauth packets.

•-a <BSSID>: target AP.

Step 3: Crack the handshake with Aircrack-ng

bash

aircrack-ng -w wordlist.txt capture-01.cap

•-w wordlist.txt: file with possible passwords (dictionary attack).

•capture-01.cap: captured handshake file.

•Aircrack-ng tries each password in the wordlist against the handshake until it finds

•a match.

Day 4: Password Cracking Basics (John the Ripper)

goal: Crack password hashes offline using a dictionary attack.

Step 1: Prepare hash file

•Extract password hashes from system files or dumps into hash.txt.

Step 2: Run John the Ripper

bash

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

•Uses the famous rockyou.txt password list.

•Tries each password against the hashes until it finds matches.

•John saves cracked passwords automatically.

Step 3: View cracked passwords

bash

john --show hash.txt

Day 5: Brute Force with Hydra

Goal: Use Hydra to brute force login credentials on various protocols.

Example: FTP brute force

bash

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt ftp://192.168.1.10
```

- -l admin: username to try.
- -P rockyou.txt: password list.
- ftp://192.168.1.10: target FTP server.

Additional tips:

- For SSH: ssh://192.168.1.10
- For HTTP form login, you need to specify additional options.
- Add -V for verbose output.
- Use -f to stop after the first valid password is found.

Social Enginering toolkit (SET)

- SET is a powerful framework used to perform social engineering attacks, such as phishing, credential harvesting, and more.

Launch SET

bash

sudo setoolkit

•You must use sudo because SET needs elevated privileges.

•After launching, you'll see a menu-based interface.

SET Main Menu Options (Explained)

Option	Description
1	Social-Engineering Attacks (main focus)
2	Penetration Testing (payloads)
3	Third-Party Modules
4	Update the SET
5	About

Continued…

example:**Website Credential Harvester**

Step-by-Step

Step-by-Step

1.Launch SET:

bash

sudo setoolkit

2.Choose:

1) Social-Engineering Attacks

3.Then:

mathematica

2) Website Attack Vectors

4.Choose:

sql

3) Credential Harvester Attack Method

Continued…

1.Choose Site Cloner:
2) Site Cloner
2.Enter your **local IP address** (e.g., 192.168.1.x) as the listener IP.
3.Enter the website to clone, like:
arduino
https://www.facebook.com
4.SET clones the page and hosts it on your machine.
- When the victim visits the fake page and enters credentials, you will receive them.

Captured Credentials Location

Look at:
css
/var/www/html/
And check the terminal output for logs.

Day 6: Review & Practice

Week 6 – Capture The Flag (CTF) & Automation

- **Concepts**: Practical hacking, automation, reporting

Day – 1 Introduction to CTFs

- **What is a CTF (Capture The Flag)?**
- A **CTF** is a cybersecurity competition where participants solve challenges to **"capture flags"** — small pieces of text or codes placed in vulnerable systems.
- **Why CTFs Are Important**
- CTFs are great for:
- Practicing ethical hacking legally
- Improving real-world skills (web, crypto, forensics, reversing)
- Building a strong hacker mindset
- Gaining experience for jobs in cybersecurity

Continued…

- ## Types of CTF

Type	Description
Jeopardy-style	Solve individual tasks (web, crypto, forensics, etc.) for points
Attack-Defense	Compete against other teams by attacking and defending systems
Mixed	Combination of both types above

Common CTF catagories

Category	Example
Web Exploitation	SQLi, XSS, CSRF challenges
Cryptography	Break encryption, decode messages
Forensics	Analyze files, images, network traffic
Reverse Engineering	Decompile programs and find hidden logic
Binary Exploitation	Exploit memory (buffer overflow, format strings)
OSINT	Find public info using Google, social

Continued…

Typical tools

Burp Suite – Web app analysis
Wireshark – Network traffic capture
Ghidra – Reverse engineering binaries
John the Ripper / **Hashcat** – Cracking hashes
Python / **Bash** – Writing automation scripts
Binwalk / **ExifTool** – File analysis

Flag Example

Web page challenge may ask:
pgsql
Find the admin password hidden in the source code
Check the page:
html
<!-- flag{you_found_the_secret} -->

Day 2: Automate Recon with Python
Example script snippet:

a **recon tool** that does:

- Subdomain scanning
- Port scanning
- Takes screenshots of live subdomains *(optional with Selenium or webscreenshot)*

The script already you have

Screenshot Recon Tool (Optional)

If you want screenshots of the live web pages:

1. Install webscreenshot:

bash

pip install webscreenshot

2. Run:

bash

webscreenshot -i -o screenshots/ -l urls.txt

>Where urls.txt is a list of live URLs (you can write from the first script).

Day 3: Automated Port Scanning in Python

- **Basic port scanner**
- import socket

- target = "192.168.1.1"
- print(f"[🔍] Scanning {target} for open ports...")

- for port in range(1, 1025):
- try:
- sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- sock.settimeout(0.5)
- result = sock.connect_ex((target, port))
- if result == 0:
- print(f"[+] Port {port} is open")
- sock.close()

Day 4: Writing Penetration Testing Reports

- reating a professional **penetration testing report** is just as important as finding vulnerabilities. Here's how to write one with the correct **structure**, **format**, and **tools**.
- **1. Structure of a Pentest Report**
- **A. Executive Summary**

  Audience: Managers, non-technical stakeholders

  Content:
  - Purpose of the test
  - High-level findings
  - Business impact
  - Recommendations summary

Continued…

Example:

During the test on StudentHub Web App, we discovered 3 critical vulnerabilities including SQL Injection and exposed admin endpoints. If exploited, attackers could compromise user data and control the server. Fixing these is high priority.

◆ **2. Methodology**

- **What tools & techniques** were used
- **Testing approach**: black-box, white-box, gray-box
- Include stages:
  - Reconnaissance
  - Scanning
  - Exploitation
  - Post-exploitation
  - Reporting

Example:

Tools used: Nmap, Burp Suite, Nikto, SQLmap, Metasploit
Methodology followed: OWASP Testing Guide v4

3. Findings (Technical Section)

Each vulnerability should be written in this format:

▶ **Vulnerability Name**

- **Description**: What is it?
- **Location**: URL or endpoint
- **Impact**: What could happen if exploited?
- **Proof of Concept (PoC)**: Steps or code used
- **Risk Level**: Low / Medium / High / Critical
- **CVE/CWE Reference** (if applicable)
- **Screenshot** (if possible)

Example:

Title: SQL Injection in Login Page
URL: http://target.com/login
Impact: Bypass login, extract DB contents
PoC:
sql

Continued…

bath

- OR '1'='1 --
- Risk: Critical
- Recommendation: Use parameterized queries, input sanitization

4. Remediation Recommendations

•Give **clear**, **technical** fixes
•Include **code examples** if possible
📌 **Example:**
Use Python’s sqlite3 parameter binding:
python
cursor.execute("SELECT * FROM users WHERE username=? AND password=?", (username, passv

Continued...

5. Appendices

- Tool output
- Nmap scans
- Logs
- Full payloads
- Hashes or screenshots

🛠 Tools for Reporting

- **Word/LibreOffice** – for document writing
- **Markdown** – if you prefer plaintext-style reports
- **CherryTree** – for organizing notes during the test
- **Dradis Framework** – for collaborative pentest reporting
- **LaTeX** – if you want a pro PDF layout

Final Tips

✓ Keep it clear and organized
✓ Avoid jargon in the Executive Summary
✓ Include screenshots
✓ Be honest – report both what you found *and what you didn't*

Day 40 & 41: Practice on TryHackMe & Hack The Box

- Signup and complete beginner rooms.

prepared by. *Yeabsira lamessa*