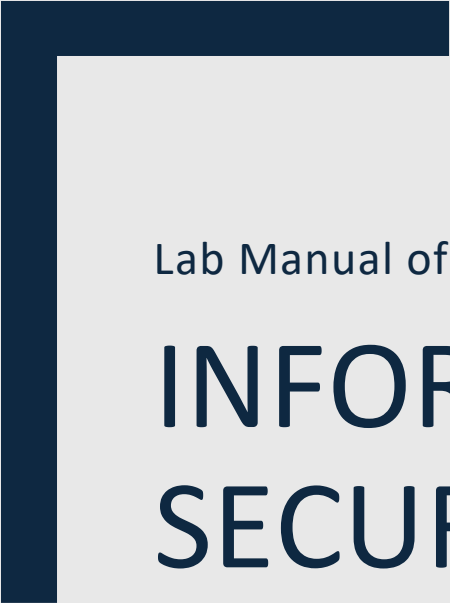




Lab Manual of

INFORMATION SECURITY

Dr. Umar Khokhar
Dr. Binh Tran



Table of Contents

Lab 1: Introduction to Kali Linux & Security Fundamentals	5
Course Alignment (Chapter 1 – Theory).....	5
Learning Objectives.....	6
Lab Environment Requirements.....	6
Part 1: Installing Kali Linux.....	6
Task 1.1: Kali Linux Installation	6
Part 2: Basic Linux Commands for Security.....	7
Task 2.1: Identify Network Information	7
Task 2.2: System Updates and Patch Management.....	8
Task 2.3: User and Account Information	8
Task 2.4: File and Directory Handling	9
Ethical Awareness Note	9
Task 2.5: Understanding Attack Tools (Demonstration Only)	10
Task 2.6: Local Web Services Awareness	10
Task 2.7: Python-Based HTTP Service.....	11
Lab 1 – Solutions	11
Solution 2.1	11
Solution 2.2	11
Solution 2.3	11
Solution 2.4	11
Solution 2.5	12
Solution 2.6	12
Solution 2.7	12
Self-Assessment Tasks.....	12
Lab 2: Network Fundamentals, Information Gathering & Honeypots	13
Overview	13
Learning Objectives.....	13
Lab Environment Requirements.....	13
Part 1: Information Gathering (Reconnaissance).....	13
Task 1.1: Website Information Gathering.....	13
Part 2: Honeypot Deployment.....	14
Task 2.1: Setting Up a Honeypot.....	14
Part 3: Port Scanning with Nmap	15
Task 3.1: Scanning Your Own System	15
Task 3.2: External Target Scanning.....	15
Task 3.3: Vulnerability Scanning	16

Lab 2 – Solutions	16
Solution 1.1 (Examples)	16
Solution 2.1 (Example Honeypot)	16
Solution 3.1 & 3.2	16
Solution 3.3	17
Self-Assessment Tasks.....	17
<i>Lab 3: Malware Concepts, Social Engineering & Web Application Attacks.....</i>	<i>18</i>
Overview	18
Learning Objectives.....	18
Lab Environment Requirements.....	18
Ethical Use Notice.....	18
Part 1: Social Engineering Demonstration	19
Task 1.1: Website Cloning (Demonstration Only)	19
Part 2: Phishing Page (Conceptual Demonstration)	19
Task 2.1: Phishing Page Design.....	19
Part 3: Web Application Vulnerability Assessment	20
Task 3.1: Vulnerability Identification	20
Part 4: DMZ Reconnaissance (Self-Assessment)	21
Scenario.....	21
Lab 3 – Solutions (High-Level)	21
Solution 1.1	21
Solution 2.1	21
Solution 3.1	22
Solution 4.1	22
Self-Assessment Questions	22
<i>Lab 4: Authentication, Password Attacks & Access Control Mechanisms</i>	<i>23</i>
Overview	23
Learning Objectives.....	23
Lab Environment Requirements.....	23
Ethical & Legal Notice.....	23
Part 1: Password Attacks (Controlled Demonstration)	24
Task 1.1: Identify Authentication Factors.....	24
Part 2: Password Attacks (Controlled Demonstration)	24
Task 2.1: Password Attack Awareness	24
Part 3: Credential Storage & Protection	24

Task 3.1: Credential Storage Observation (Read-Only)	25
Part 4: Centralized Authentication Systems.....	25
Task 4.1: Authentication Protocol Comparison	25
Part 4.2: Mini Multi-Factor Authentication (MFA) Demonstration.....	25
Part 5: Secure Scripting for Authentication Monitoring	26
Task 5.1: User Activity Logging (Safe Simulation)	26
Task 5.2: Malware Detection by Signature (Simulation)	26
Task 5.3: Master Script.....	26
Lab 4 – Sample Answers	27
Sample Answer 1.1: Authentication Factors	27
Sample Answer 2.1: Password Attacks	27
Sample Answer 3.1: Credential Storage.....	27
Sample Answer 4.1: Centralized Authentication	27
Sample Answer 4.2: MFA	27
Sample Answer 5.x: Secure Scripting	28
Self-Assessment Questions	28
<i>Lab 5: Network Sniffing & Session Hijacking.....</i>	<i>29</i>
Overview	29
Learning Objectives.....	29
Lab Environment Requirements.....	29
Ethical & Legal Notice.....	29
Part 1: Packet Sniffing and Traffic Analysis	30
Task 1.1: Analyze ICMP (Ping) Traffic	30
Task 1.2: Analyze Ping to Host System.....	30
Task 1.3: Detect Network Scanning Activity	31
Task 1.4: HTTP Traffic Observation (Unencrypted)	31
Part 2: Session Hijacking (Conceptual Demonstration)	32
Task 2.1: Burp Suite Configuration.....	32
Task 2.2: Session Hijacking Explanation	32
Part 3: Web Application Vulnerability Assessment (Controlled)	33
Task 3.1: Vulnerability Identification	33
Lab 5 – Sample Answers	33
Sample Answer 1.1	33
Sample Answer 1.4	33
Sample Answer 2.2	33
Sample Answer 3.1	34
Self-Assessment Questions	34

<i>Lab 6: Cryptography & Public Key Infrastructure (PKI)</i>	35
Overview	35
Learning Objectives.....	35
Lab Environment Requirements.....	35
Ethical Use Notice.....	35
Part 1: Hashing and Data Integrity.....	36
Task 1.1: Generate Cryptographic Hashes	36
Part 2: Symmetric Encryption	36
Task 2.1: File Encryption Using a Shared Key	36
Part 3: Asymmetric Encryption (Public Key Cryptography)	36
Task 3.1: Key Pair Generation	36
Part 4: Digital Certificates & PKI.....	37
Task 4.1: Certificate Inspection	37
Lab 6 – Sample Answers	37
Sample Answer 1.1	37
Sample Answer 2.1	38
Sample Answer 3.1	38
Sample Answer 4.1	38
Self-Assessment Questions	38
<i>Lab 7: Firewalls and Intrusion Detection Systems (IDS)</i>	39
Learning Objectives.....	39
Lab Environment Requirements.....	39
Part 1: Firewall Concepts in Practice.....	39
Task 1.1: Packet Filtering Firewall (Conceptual Demonstration).....	39
Task 1.2: Circuit-Level Firewall (Conceptual Understanding)	40
Task 1.3: Application-Level Firewall (Conceptual Mapping)	40
Part 2: Intrusion Detection Systems (IDS)	40
Task 2.1: IDS Types Review	40
Part 3: SNORT Installation and Configuration	41
Task 3.1: Verify SNORT Installation	41
Task 3.2: Configure SNORT Rule for ICMP (Ping Detection)	41
Task 3.3: Detect Port Scanning Activity	42
Part 4: Analysis and Defense Perspective.....	42
Task 4.1: Alert Interpretation	42
Lab 7 – Solutions (Sample)	43
Solution 1.1	43

Solution 3.1	43
Solution 3.2 (Sample Rule)	43
Solution 3.3 (Sample Rule)	43
Self-Assessment Tasks.....	43
<i>Lab 8: Bash Scripting for Security Automation.....</i>	44
Learning Objectives.....	44
Lab Environment Requirements.....	44
Part 1: Bash-Based Ping Scanner.....	44
Task 1.1: Interactive Ping Scanner	44
Part 2: DNS & WHOIS Reconnaissance Tool.....	45
Task 2.1: DNS and WHOIS Information Gathering	45
Part 3: File-Based Ping Scanner	45
Task 3.1: Batch Domain Scanning	46
Part 4: Bash-Based Port Scanning (Conceptual).....	46
Task 4.1: Simple Port Scanner	46
Part 5: Security Awareness – Social Engineering & Vulnerability Scanning	47
Task 5.1: Automation Risks Discussion	47
Lab 8 – Sample Solutions (Partial).....	48
Sample: Ping Scanner	48
Sample: File-Based Ping Scanner	48
Sample: Port Check Logic.....	48
Self-Assessment Tasks.....	48

Lab 1: Introduction to Kali Linux & Security Fundamentals

Course Alignment (Chapter 1 – Theory)

This lab aligns with **Chapter 1** of the course, where we cover:

- Introduction to Information Security & why it is important
- CIA Triad (Confidentiality, Integrity, Availability)
- Seven domains of IT infrastructure and associated threats
- Types of attackers (hackers)
- Cyber Kill Chain

This lab focuses on **building a secure lab environment**, **basic system awareness**, and **mapping practical tasks to security concepts** discussed in theory.

Learning Objectives

By the end of this lab, students will be able to:

- Install and access Kali Linux securely
 - Use basic Linux commands relevant to security operations
 - Understand how system information supports security monitoring
 - Relate practical tasks to CIA, attacker behavior, and the Cyber Kill Chain
-

Lab Environment Requirements

- Host OS: Windows / macOS / Linux
 - Virtualization Software: VirtualBox or VMware
 - Guest OS: Kali Linux (latest version)
 - Internet connection
-

Part 1: Installing Kali Linux

Task 1.1: Kali Linux Installation

Objective: Create a controlled and isolated environment for security testing.

Instructions:

1. Download Kali Linux from the official website.
2. Install Kali Linux using VirtualBox or VMware.
3. Log in successfully and ensure the desktop loads correctly.

Deliverable:

- Paste **one screenshot of the Kali Linux desktop** showing date and username.

Security Mapping:

- **Seven Domains:** Workstation Domain, System/Application Domain
 - **CIA:** Integrity (trusted OS), Availability (usable lab environment)
-

Part 2: Basic Linux Commands for Security

Task 2.1: Identify Network Information

Objective: Understand how attackers and defenders identify systems on a network.

1. Find the IP address of your Kali Linux machine.
2. Check whether the Host OS can communicate with the Guest OS using `ping`.

Deliverables:

- Command used
- Output screenshot

Screenshot:

Security Mapping:

- **Cyber Kill Chain:** Reconnaissance
 - **Threat Perspective:** Network discovery by attackers
-

Task 2.2: System Updates and Patch Management

Objective: Understand the importance of patching in reducing vulnerabilities.

1. Update all system files and software packages.

Security Mapping:

- **CIA:** Integrity, Availability
- **Threat Mitigation:** Reduces known vulnerabilities

Screenshot:

Task 2.3: User and Account Information

Objective: Understand user accounts as part of access control.

Find the following:

- a. Current username
- b. Recursive details of the user
- c. List all users on the system

Screenshot:

Security Mapping:

- **CIA:** Confidentiality
- **Attackers:** Insider threats, privilege escalation

Screenshot:

Task 2.4: File and Directory Handling

Objective: Learn file operations commonly used in security investigations.

Perform the following tasks:

- Find the current directory
- Go back one directory
- Return to the previous directory
- Access the Desktop directory
- Create a file on the Desktop
- Delete the file
- Create a folder
- Remove the folder
- Display command history
- Copy a file
- View file permissions
- Change file permissions

Security Mapping:

- **CIA:** Confidentiality & Integrity
- **Seven Domains:** System/Application Domain

Screenshot:

Ethical Awareness Note

The following tasks are **for educational purposes only** and must be performed **only in a controlled lab environment**.

Task 2.5: Understanding Attack Tools (Demonstration Only)

1. Download a phishing-related tool from GitHub (do not deploy it against real targets).
2. Briefly explain what stage of the **Cyber Kill Chain** this tool fits into.

Deliverable:

- Command used to clone the repository (Demo of the tool with screenshots)
- Short written explanation (2–3 lines)

Screenshot:

Task 2.6: Local Web Services Awareness

Objective: Understand how services can be misconfigured and exploited.

1. Configure the local Apache web server on Kali Linux.
2. Verify that the service is running locally.

Screenshot:

Task 2.7: Python-Based HTTP Service

Objective: Learn how attackers and defenders quickly deploy services.

1. Navigate to the Desktop directory.
2. Use Python to configure a simple HTTP service.
3. Access the service through a browser.

Security Mapping:

- **Cyber Kill Chain:** Delivery / Command & Control (conceptual)

Screenshot:

Lab 1 – Solutions

Solution 2.1

```
ip a
ping <host-ip-address>
```

Solution 2.2

```
sudo apt update && sudo apt upgrade -y
```

Solution 2.3

```
whoami
id
cat /etc/passwd
```

Solution 2.4

```
pwd
cd ..
cd -
cd ~/Desktop
touch test.txt
rm test.txt
mkdir testfolder
rmdir testfolder
history
cp source.txt destination.txt
ls -l
chmod 644 filename
```

Solution 2.5

```
git clone <repository-url>
```

Cyber Kill Chain Stage: **Reconnaissance / Weaponization**

Solution 2.6

```
sudo service apache2 start  
systemctl status apache2
```

Solution 2.7

```
cd ~/Desktop  
python3 -m http.server 8080
```

Self-Assessment Tasks

1. Explain how finding an IP address relates to the **Reconnaissance** phase of the Cyber Kill Chain.
2. Which CIA principle is most affected if system updates are ignored? Why?
3. Identify which **IT infrastructure domain** is most at risk if file permissions are misconfigured.
4. Why is phishing considered a high-risk attack vector?
5. What ethical responsibilities do cybersecurity professionals have when using tools like Kali Linux?

Lab 2: Network Fundamentals, Information Gathering & Honeypots

Overview

This lab builds on networking concepts discussed in class, including the OSI model, TCP/IP, IP addressing, and how networking functions from a security perspective. Students will explore how attackers gather information, how defenders detect malicious activity, and how these actions fit into the cyber kill chain.

Learning Objectives

By the end of this lab, students will be able to:

- Apply OSI and TCP/IP concepts to real-world security tasks
 - Perform basic information gathering using ethical tools
 - Understand how reconnaissance supports cyber attacks
 - Configure a simple honeypot to observe attacker behavior
 - Use Nmap to analyze network exposure and services
-

Lab Environment Requirements

- Kali Linux (Virtual Machine)
 - Internet connection
 - Nmap (preinstalled on Kali)
 - Administrative privileges
-

Part 1: Information Gathering (Reconnaissance)

Task 1.1: Website Information Gathering

Objective: Learn how publicly available information can be collected during the reconnaissance phase of an attack.

Target Website: `testphp.vulnweb.com`
(Use only for educational purposes.)

Use any ethical information-gathering tools to collect the following details. You may provide **typed answers or screenshots**.

1. Hosting company
2. Location of the web server
3. IP address(es) of the domain and any subdomains
4. Website availability status (up/down)
5. Name and email address of a key IT contact (*if not found, state "Not Publicly Available"*)
6. Social media presence of the IT contact (*you may use your own name for demonstration*)
7. Directories or paths not indexed or disallowed by search engines

Screenshot / Answer Space:

Part 2: Honeypot Deployment

Task 2.1: Setting Up a Honeypot

Objective: Understand how defenders monitor and detect malicious activity on a network.

1. Install a honeypot capable of logging connection attempts and generating alerts.
2. Configure the honeypot to listen on at least one open port.
3. Simulate or observe connection attempts to the honeypot.

Deliverables:

- Screenshot showing the honeypot running
- Screenshot or log output of recorded activity

Screenshot Space:

Concept Note: Honeypots help defenders study attacker techniques and behavior without risking production systems.

Part 3: Port Scanning with Nmap

Task 3.1: Scanning Your Own System

Objective: Understand how open ports and services reveal system exposure.

Using Nmap, perform the following scans on **your own public IP address**:

1. Simple scan
2. Operating system detection
3. Ping scan
4. Port scan on the following ports:
 - 20, 21, 22, 23, 80, 139, 443

Screenshot Space:

Task 3.2: External Target Scanning

1. Scan the **top 20 ports** of `gwinnetttech.edu`.
2. Create a text file containing:
 - Your IP address
 - `gwinnetttech.edu`
3. Use Nmap to read from the file and perform scanning.

Screenshot Space:

Task 3.3: Vulnerability Scanning

Objective: Identify known vulnerabilities using automated scripts.

1. Perform an Nmap vulnerability scan using built-in CVE scripts.
2. Target: testphp.vulnweb.com

Screenshot Space:

Lab 2 – Solutions

Solution 1.1 (Examples)

```
whois testphp.vulnweb.com
nslookup testphp.vulnweb.com
host testphp.vulnweb.com
```

Solution 2.1 (Example Honeypot)

```
sudo apt install cowrie
sudo systemctl start cowrie
```

Solution 3.1 & 3.2

```
nmap <your-public-ip>
nmap -O <your-public-ip>
nmap -sn <your-public-ip>
nmap -p 20,21,22,23,80,139,443 <your-public-ip>
nmap --top-ports 20 gwinnetttech.edu
nmap -iL targets.txt
```

Solution 3.3

```
nmap --script vuln testphp.vulnweb.com
```

Self-Assessment Tasks

1. Which OSI layers are primarily involved during port scanning?
 2. Why is reconnaissance considered a critical stage of the cyber kill chain?
 3. How does a honeypot improve network security visibility?
 4. What risks exist if port scanning is performed on unauthorized systems?
 5. How can network defenders use scan results to reduce attack surfaces?
-

Lab 3: Malware Concepts, Social Engineering & Web Application Attacks

Overview

This lab supports the concepts covered in Chapter 3, including malware types, common attack techniques, and web application vulnerabilities. The focus is on understanding **how attacks work**, **how they are identified**, and **why defenses are required**, rather than uncontrolled exploitation.

All activities are conducted for **educational purposes only** using intentionally vulnerable targets.

Learning Objectives

By the end of this lab, students will be able to:

- Explain common social engineering techniques and their impact
 - Recognize phishing and malicious web behavior
 - Identify web application vulnerabilities such as SQL injection and XSS
 - Perform controlled reconnaissance and vulnerability assessment
 - Relate attacks to real-world risks such as data breaches and service disruption
-

Lab Environment Requirements

- Kali Linux (Virtual Machine)
 - Internet connection
 - Web browser (Firefox)
 - Security tools available in Kali Linux
-

Ethical Use Notice

This lab involves **demonstrations of attack techniques**. These activities must only be performed:

- In a controlled lab environment

- Against intentionally vulnerable or authorized targets
- For learning and defensive understanding only

Unauthorized testing against real systems is illegal and unethical.

Part 1: Social Engineering Demonstration

Task 1.1: Website Cloning (Demonstration Only)

Objective: Understand how social engineering is used to trick users into revealing sensitive information.

1. Demonstrate a social engineering tool that can clone a legitimate website.
2. Explain how attackers use cloned websites to capture confidential information.
3. Do **not** target real users.

Deliverables:

- Screenshot of the cloned website running locally
- Short explanation (3–4 lines) describing the attack scenario

Screenshot Space:

Part 2: Phishing Page (Conceptual Demonstration)

Task 2.1: Phishing Page Design

Objective: Understand how phishing pages deliver malicious content to victims.

1. Design a basic phishing page for demonstration purposes.
2. Explain how malicious scripts can be injected or executed in a victim's browser.
3. Clearly label the page as a **simulation**.

Deliverables:

- Screenshot of the phishing page
- Brief explanation of how the attack works

Screenshot Space:

Part 3: Web Application Vulnerability Assessment

Task 3.1: Vulnerability Identification

Objective: Identify common web application vulnerabilities in an intentionally vulnerable environment.

Target: testphp.vulnweb.com

1. Perform a vulnerability assessment of the target website.
2. Identify SQL injection points.
3. Demonstrate how SQL injection can expose backend data **without modifying the system.**

Note: Do not alter, delete, or damage data.

Deliverables:

- Screenshot showing vulnerability detection
- Screenshot showing database enumeration results

Screenshot Space:

Part 4: DMZ Reconnaissance (Self-Assessment)

Scenario

Assume `testphp.vulnweb.com` is your client's web server hosted in a DMZ. You are tasked with performing **initial reconnaissance** only.

Complete the following table based on your findings:

Item	Findings
Subdomain(s) with IP addresses	
Hosting Company	
Location of Hosting Company	
Focal Person	
Email Address	
Open Ports	
Closed Ports	
Filtered Ports	
Operating System	
Traceroute	
Domain Expiry Date	

Lab 3 – Solutions (High-Level)

Exact exploitation commands may vary based on tools used. Answers should demonstrate **understanding**, not automation.

Solution 1.1

- Website cloning is typically performed using social engineering frameworks.
- This attack relies on **user trust**, not technical vulnerabilities.

Solution 2.1

- Phishing pages exploit lack of input validation and user awareness.
- Script injection often targets browser-side execution.
- BeEF (Tool)

Solution 3.1

- SQL injection occurs when user input is not sanitized.
- Vulnerable queries allow attackers to retrieve database information.
- Sqlmap (tool)

Solution 4.1

- Reconnaissance uses publicly available tools such as WHOIS, DNS lookups, traceroute, and port scanning.

Self-Assessment Questions

1. Why is social engineering considered one of the most effective attack methods?
 2. How does SQL injection compromise confidentiality and integrity?
 3. Which malware types could be delivered through phishing attacks?
 4. Why is a DMZ used to host web servers?
 5. How can secure coding practices prevent common web attacks?
-

Lab 4: Authentication, Password Attacks & Access Control Mechanisms

Overview

This lab strengthens understanding of **authentication concepts** by connecting theory with practical system behavior. Students will explore authentication factors, observe how credentials are stored and attacked, and understand how centralized authentication systems protect enterprise environments.

Rather than focusing on unsafe scripting practices, this lab emphasizes **defensive awareness**, **controlled demonstrations**, and **secure authentication workflows**.

Learning Objectives

By the end of this lab, students will be able to:

- Explain the five authentication factors with real system examples
 - Understand common password attack techniques and why they succeed
 - Examine how credentials are stored and protected by operating systems
 - Compare centralized authentication mechanisms (RADIUS, TACACS+, Kerberos, LDAP, OAuth)
 - Demonstrate basic scripting for monitoring and security enforcement (ethically)
-

Lab Environment Requirements

- Kali Linux (VM)
 - Windows system or Windows VM (for observation tasks)
 - Internet connection
 - Administrative privileges (where required)
-

Ethical & Legal Notice

This lab discusses **credential handling and attack techniques** strictly for educational purposes. Students must not:

- Steal credentials from real users
- Deploy keyloggers or malware
- Bypass authentication on unauthorized systems

All demonstrations must remain **local, simulated, and instructor-approved**.

Part 1: Password Attacks (Controlled Demonstration)

Task 1.1: Identify Authentication Factors

Objective: Map real-world authentication methods to the five authentication factors.

Complete the following table:

Authentication Example	Factor Type
Password / PIN	
Fingerprint	
Smart Card / Token	
Face Recognition	
Location-based Login	

Part 2: Password Attacks (Controlled Demonstration)

Task 2.1: Password Attack Awareness

Objective: Understand how weak passwords are compromised.

1. Research and explain the following attack types:
 - Dictionary attack
 - Brute-force attack
 - Rainbow table attack
 - Probabilistic attack
2. Explain **which authentication factor** is targeted by these attacks.

Deliverable: Short written explanation (2–3 lines per attack).

Part 3: Credential Storage & Protection

Task 3.1: Credential Storage Observation (Read-Only)

Objective: Understand where and how systems store credentials.

1. Identify where Linux stores password hashes.
2. Identify where Windows stores authentication credentials (conceptually).
3. Explain why direct access to these files is restricted.

Answer Space:

Part 4: Centralized Authentication Systems

Task 4.1: Authentication Protocol Comparison

Objective: Compare enterprise authentication mechanisms.

Complete the table:

Protocol	Used For	Authentication Type	Common Use Case
RADIUS			
TACACS+			
Kerberos			
LDAP			
OAuth			

Part 4.2: Mini Multi-Factor Authentication (MFA) Demonstration

Objective: Observe how MFA strengthens authentication beyond passwords.

1. Research and demonstrate (conceptually or practically) one MFA implementation, such as:
 - Google Authenticator (TOTP)
 - Microsoft Authenticator
 - Linux PAM-based MFA (overview only)
2. Identify:
 - First authentication factor used
 - Second authentication factor used
3. Explain how MFA reduces the risk of password-based attacks.

Screenshot / Explanation Space:

Part 5: Secure Scripting for Authentication Monitoring

Task 5.1: User Activity Logging (Safe Simulation)

Objective: Learn how logging supports authentication security without keylogging.

Write a simple script (Batch or Bash) that:

- Records username and login time
- Stores logs in a file
- Displays a warning banner about authorized use

Script / Output Space:

Task 5.2: Malware Detection by Signature (Simulation)

Objective: Understand basic malware detection logic.

1. Create a script that scans a directory for a **known test filename** (e.g., `malware_test.txt`).
2. If found, log the event and delete the file.

Script / Output Space:

Task 5.3: Master Script

Objective: Understand automation in security operations.

Create a master script that:

- Displays a menu
- Allows the user to run Task 5.1 or Task 5.2
- Logs which option was selected

Script / Output Space:

Lab 4 – Sample Answers

Sample Answer 1.1: Authentication Factors

- Password / PIN → Something you know
 - Fingerprint → Something you are
 - Smart Card / Token → Something you have
 - Face Recognition → Something you are
 - Location-based Login → Somewhere you are
-

Sample Answer 2.1: Password Attacks

- **Dictionary attack:** Uses common password lists to guess weak passwords.
- **Brute-force attack:** Tries all possible combinations.
- **Rainbow table attack:** Uses precomputed hashes to crack passwords.
- **Probabilistic attack:** Uses patterns and probability to guess passwords.

All target **single-factor (knowledge-based) authentication**.

Sample Answer 3.1: Credential Storage

- Linux stores password hashes in `/etc/shadow`.
 - Windows stores credentials in the SAM database.
 - These locations are protected to prevent credential theft.
-

Sample Answer 4.1: Centralized Authentication

- RADIUS: Network access authentication
 - TACACS+: Device administration authentication
 - Kerberos: Ticket-based authentication
 - LDAP: Directory-based authentication
 - OAuth: Token-based authorization
-

Sample Answer 4.2: MFA

- MFA combines at least two authentication factors.
- Even if a password is compromised, attackers cannot log in without the second factor.

Sample Answer 5.x: Secure Scripting

- Logging records user activity for accountability.
- Signature-based detection identifies known threats.
- Automation ensures consistent security controls.

Self-Assessment Questions

1. Why is multi-factor authentication more secure than passwords alone?
 2. Which authentication factor is most difficult for attackers to compromise?
 3. How do Kerberos and OAuth differ in purpose?
 4. Why should credential storage locations be protected?
 5. How does logging help detect authentication attacks?
-

Lab 5: Network Sniffing & Session Hijacking

Overview

This lab examines how network traffic can be observed and manipulated, and why modern networks use switching, encryption, and secure session management to defend against these attacks. Students will use packet analysis tools to **observe traffic**, understand **promiscuous mode**, and conceptually study **session hijacking techniques** in a controlled environment.

All activities focus on **monitoring, analysis, and defensive understanding** rather than uncontrolled exploitation.

Learning Objectives

By the end of this lab, students will be able to:

- Explain how packet sniffing works on wired networks
 - Use Wireshark to analyze network traffic
 - Understand why switched networks reduce passive sniffing
 - Describe attacks against switched networks (ARP poisoning, MAC flooding, spoofing)
 - Explain how session hijacking occurs and how it is prevented
-

Lab Environment Requirements

- Kali Linux (Virtual Machine)
 - Host OS (Windows / macOS / Linux)
 - Wireshark
 - Nmap
 - Burp Suite Community Edition
 - Internet connection
-

Ethical & Legal Notice

Packet sniffing and session interception can expose sensitive information. These activities must only be performed:

- On systems you own or are authorized to test
 - Within a lab or classroom environment
 - For learning and defensive purposes only
-

Part 1: Packet Sniffing and Traffic Analysis

Task 1.1: Analyze ICMP (Ping) Traffic

Objective: Observe packet structure and protocol layers using Wireshark.

1. Start Wireshark and select the active network interface.
2. Send a ping request to `google.com`.
3. Capture and analyze the ICMP packets.

Deliverables:

- Screenshot showing ICMP request and reply packets
- Identify source IP, destination IP, and protocol

Screenshot Space:

Task 1.2: Analyze Ping to Host System

Objective: Compare local vs external traffic.

1. Send a ping from Kali to your Host OS.
2. Capture and analyze the packets in Wireshark.

Deliverables:

- Screenshot of captured packets
- Note any differences from Task 1.1

Screenshot Space:

Task 1.3: Detect Network Scanning Activity

Objective: Identify reconnaissance traffic.

1. Use Nmap on Kali to scan your Host OS.
2. Observe and capture the scan traffic using Wireshark.

Deliverables:

- Screenshot showing Nmap scan packets
- Identify TCP SYN packets

Screenshot Space:

Task 1.4: HTTP Traffic Observation (Unencrypted)

Objective: Understand why plaintext protocols are insecure.

1. Capture HTTP traffic (non-HTTPS) in Wireshark.
2. Identify request parameters and session identifiers (if visible).

Note: Do not capture real credentials.

Deliverables:

- Screenshot highlighting HTTP headers or parameters
- Short explanation of the security risk

Screenshot Space:

Part 2: Session Hijacking (Conceptual Demonstration)

Task 2.1: Burp Suite Configuration

Objective: Learn how proxy tools observe web sessions.

1. Configure Burp Suite as a proxy in the browser.
2. Intercept HTTP/HTTPS requests to a test application.

Deliverables:

- Screenshot showing intercepted request

Screenshot Space:

Task 2.2: Session Hijacking Explanation

Objective: Understand session hijacking techniques.

1. Identify session cookies or tokens in intercepted traffic.
2. Explain how attackers could exploit weak session management.

Simulation only — no unauthorized access.

Deliverables:

- Screenshot of session token (masked)
- Written explanation (3–4 lines)

Screenshot / Answer Space:

Part 3: Web Application Vulnerability Assessment (Controlled)

Task 3.1: Vulnerability Identification

Objective: Understand how web vulnerabilities expose session data.

Target:

testphp.vulnweb.com/listproducts.php?cat=1

1. Perform a vulnerability assessment of the target page.
2. Identify input parameters vulnerable to attacks.
3. Explain the potential impact on user sessions and data.

Deliverables:

- Screenshot of vulnerability findings
- Short explanation of risk

Screenshot Space:

Lab 5 – Sample Answers

Sample Answer 1.1

- ICMP packets contain Echo Request and Echo Reply messages.
- These operate at the Network layer.

Sample Answer 1.4

- HTTP traffic is transmitted in plaintext.
- Attackers can view session IDs and parameters using sniffing tools.

Sample Answer 2.2

- Session hijacking occurs when attackers steal or guess valid session tokens.
- Secure cookies and HTTPS reduce this risk.

Sample Answer 3.1

- Vulnerable input parameters can expose backend data.
- Poor session handling increases hijacking risk.

Self-Assessment Questions

1. Why does HTTPS prevent simple packet sniffing?
 2. What role does promiscuous mode play in sniffing?
 3. How do switched networks limit packet visibility?
 4. Explain ARP poisoning in simple terms.
 5. How can secure session management prevent hijacking?
-

Lab 6: Cryptography & Public Key Infrastructure (PKI)

Overview

This lab introduces core cryptographic concepts and Public Key Infrastructure (PKI) through simple, controlled exercises. The goal is to help students understand **how encryption, hashing, digital signatures, and certificates protect data**, rather than focusing on complex implementations.

Learning Objectives

By the end of this lab, students will be able to:

- Differentiate between encryption and hashing
 - Explain symmetric and asymmetric cryptography
 - Generate and compare cryptographic hashes
 - Understand digital certificates and PKI concepts
 - Observe how HTTPS uses certificates for secure communication
-

Lab Environment Requirements

- Kali Linux (Virtual Machine)
 - OpenSSL (preinstalled on Kali)
 - Web browser
 - Internet connection
-

Ethical Use Notice

All cryptographic tools in this lab are used for **learning and defensive understanding only**. No attempt should be made to decrypt or access unauthorized data.

Part 1: Hashing and Data Integrity

Task 1.1: Generate Cryptographic Hashes

Objective: Understand how hashing ensures data integrity.

1. Create a simple text file (e.g., `message.txt`).
2. Generate the following hashes for the file:
 - MD5
 - SHA-256
3. Modify the file slightly and generate the hashes again.

Deliverables:

- Hash values before and after modification
- Brief observation on the change

Output Space:

Part 2: Symmetric Encryption

Task 2.1: File Encryption Using a Shared Key

Objective: Observe how symmetric encryption protects confidentiality.

1. Encrypt a file using a symmetric encryption method.
2. Decrypt the file using the same password or key.

Deliverables:

- Screenshot showing encryption and decryption commands
- Short explanation of where symmetric encryption is commonly used

Screenshot Space:

Part 3: Asymmetric Encryption (Public Key Cryptography)

Task 3.1: Key Pair Generation

Objective: Understand public and private keys.

1. Generate a public/private key pair.
2. Identify which key is kept secret and which is shared.

Deliverables:

- Screenshot of key generation
- Short explanation of key usage

Screenshot Space:

Part 4: Digital Certificates & PKI

Task 4.1: Certificate Inspection

Objective: Understand how PKI enables trust.

1. Visit a secure website (HTTPS).
2. View and inspect the website's digital certificate.
3. Identify:
 - Issuer
 - Validity period
 - Public key algorithm

Deliverables:

- Screenshot of certificate details
- Short explanation of the certificate's role

Screenshot Space:

Lab 6 – Sample Answers

Sample Answer 1.1

- Hash values change completely even with small file modifications.
- This property ensures data integrity.

Sample Answer 2.1

- Symmetric encryption uses a single shared key.
- It is commonly used for disk encryption and secure file storage.

Sample Answer 3.1

- The private key must remain secret.
- The public key can be safely shared.

Sample Answer 4.1

- Digital certificates bind identities to public keys.
- PKI allows users to trust secure websites.

Self-Assessment Questions

1. Why is hashing not used for data confidentiality?
2. Compare symmetric and asymmetric encryption.
3. What role does a Certificate Authority

Lab 7: Firewalls and Intrusion Detection Systems (IDS)

Learning Objectives

By the end of this lab, students will be able to:

- Understand how different firewall types filter network traffic
 - Observe firewall behavior from a packet and application perspective
 - Configure and test basic IDS rules using SNORT
 - Correlate alerts with attacker activities from the Cyber Kill Chain
-

Lab Environment Requirements

- Kali Linux (Guest OS)
 - Host machine (any OS)
 - Internet connection
 - SNORT installed on Kali Linux
-

Part 1: Firewall Concepts in Practice

Task 1.1: Packet Filtering Firewall (Conceptual Demonstration)

Objective: Observe how packet filtering firewalls control traffic based on IPs, ports, and protocols.

1. Identify which protocol and port are used when you:
 - Ping a remote host
 - Access a website using a browser
2. Use system commands to list active connections and listening ports.

Deliverable:

- Command output
- Brief explanation (2–3 lines)

Security Focus:

- OSI Layer: Network & Transport
 - Attack Mapping: Reconnaissance
-

Task 1.2: Circuit-Level Firewall (Conceptual Understanding)

Objective: Understand how circuit-level firewalls monitor sessions instead of packets.

1. Explain how a TCP handshake works.
2. Describe how a circuit-level firewall decides whether to allow or block a session.

Deliverable:

- Written explanation (3–4 lines)
-

Task 1.3: Application-Level Firewall (Conceptual Mapping)

Objective: Understand how application-layer firewalls inspect payloads.

1. Identify examples of application-layer filtering (e.g., HTTP method blocking).
2. Explain how a Web Application Firewall (WAF) protects against attacks like SQL injection or XSS.

Deliverable:

- Short explanation
-

Part 2: Intrusion Detection Systems (IDS)**Task 2.1: IDS Types Review**

Objective: Reinforce understanding of IDS deployment models.

1. Briefly describe:
 - Host-Based IDS (HIDS)

- Network-Based IDS (NIDS)
- Hybrid IDS

Deliverable:

- Short definitions
-

Part 3: SNORT Installation and Configuration

Task 3.1: Verify SNORT Installation

Objective: Ensure SNORT is installed and operational.

1. Check the installed version of SNORT.

Deliverable:

- Command used and output
-

Task 3.2: Configure SNORT Rule for ICMP (Ping Detection)

Objective: Detect ICMP ping attempts using SNORT.

1. Create a custom SNORT rule to detect ICMP echo requests.
2. Run SNORT in IDS mode.
3. Generate ICMP traffic by pinging the Kali machine.

Deliverable:

- SNORT rule
- Alert output

Security Mapping:

- Cyber Kill Chain: Reconnaissance
-

Task 3.3: Detect Port Scanning Activity

Objective: Detect suspicious scanning behavior.

1. Create or enable a SNORT rule to detect TCP port scans.
2. Perform a port scan on Kali Linux from another machine.
3. Observe and document SNORT alerts.

Deliverable:

- Alert screenshots
 - Short explanation
-

Part 4: Analysis and Defense Perspective

Task 4.1: Alert Interpretation

Objective: Understand how alerts help defenders respond.

1. Identify:
 - Source IP
 - Destination IP
 - Type of attack detected

Deliverable:

- Short written analysis
-

Lab 7 – Solutions (Sample)

Solution 1.1

```
netstat -tuln  
ss -tuln
```

Ping uses ICMP; web browsing uses TCP ports 80/443.

Solution 3.1

```
snort -V
```

Solution 3.2 (Sample Rule)

```
alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)
```

Solution 3.3 (Sample Rule)

```
alert tcp any any -> any any (flags:S; msg:"Possible TCP Port Scan";  
sid:1000002; rev:1;)
```

Self-Assessment Tasks

1. Which firewall type operates at the application layer and why is it more secure?
 2. Why are IDS alerts considered detection and not prevention?
 3. How does SNORT help identify reconnaissance activities?
 4. Which IDS type is most suitable for detecting insider threats and why?
-

Lab 8: Bash Scripting for Security Automation

Learning Objectives

By the end of this lab, students will be able to:

- Use Bash scripting to automate basic security tasks
 - Understand how scripting supports reconnaissance and monitoring
 - Safely design simple scanners for learning purposes
 - Map automation tasks to the Cyber Kill Chain from a defensive perspective
-

Lab Environment Requirements

- Kali Linux
 - Basic knowledge of Linux commands
 - Internet connectivity (for testing domains)
-

Part 1: Bash-Based Ping Scanner

Task 1.1: Interactive Ping Scanner

Objective: Automate basic network reachability checks.

Requirements:

- Prompt the user to enter:
 - A domain or IP address
 - Number of ICMP packets
- Display whether the target is reachable

Deliverables:

- Bash script
- Output screenshot

Security Perspective:

- Cyber Kill Chain: Reconnaissance
 - Used by defenders to verify host availability
-

Part 2: DNS & WHOIS Reconnaissance Tool**Task 2.1: DNS and WHOIS Information Gathering**

Objective: Understand how attackers and defenders gather domain intelligence.

Requirements:

- Prompt the user for a domain name
- Perform:
 - DNS lookup
 - WHOIS lookup
- Detect and display if a null or empty response is returned

Deliverables:

- Bash script
- Sample output

Security Perspective:

- OSI Layer: Application
 - Information disclosure risks
-

Part 3: File-Based Ping Scanner

Task 3.1: Batch Domain Scanning

Objective: Learn how automation scales reconnaissance.

Requirements:

- Read domain/IP addresses from a text file
- Ping each entry and display reachability
- Clearly separate reachable vs unreachable hosts

Deliverables:

- Bash script
- Text file used for testing
- Output screenshot

Part 4: Bash-Based Port Scanning (Conceptual)

Task 4.1: Simple Port Scanner

Objective: Understand port scanning logic without advanced tools.

Requirements:

- Allow the user to enter a target IP/domain
- Check the status of:
 - Common ports (Top 20)
- Display open or closed status

Optional (Bonus):

- Detect the operating system using banner or TTL observation (conceptual)

Deliverables:

- Bash script
- Output screenshot

Security Perspective:

- Cyber Kill Chain: Reconnaissance
 - Defensive use: attack surface identification
-

Part 5: Security Awareness – Social Engineering & Vulnerability Scanning**Task 5.1: Automation Risks Discussion**

Objective: Understand the dual-use nature of scripts.

Answer the following:

1. How can simple scripts be misused in social engineering attacks?
2. Why should vulnerability scanning be authorized?
3. How do defenders use automation responsibly?

Deliverable:

- Written response (5–6 lines)
-

Lab 8 – Sample Solutions (Partial)

Sample: Ping Scanner

```
read -p "Enter domain or IP: " target
read -p "Enter number of packets: " count
ping -c $count $target
```

Sample: File-Based Ping Scanner

```
for host in $(cat targets.txt); do
    ping -c 1 $host > /dev/null && echo "$host is reachable" || echo "$host is
unreachable"
done
```

Sample: Port Check Logic

```
for port in 20 21 22 23 25 53 80 110 139 143 443; do
    timeout 1 bash -c "</dev/tcp/$target/$port" && echo "Port $port open" ||
echo "Port $port closed"
done
```

Self-Assessment Tasks

1. Why is scripting considered a force multiplier in cybersecurity?
 2. Which phase of the Cyber Kill Chain benefits most from automation?
 3. What ethical rules must be followed when running scanners?
 4. Why are tools like Nmap preferred over custom scripts in production?
-