INTRODUCTION TO PENETRATION TESTING

1. What is Penetration Testing?

Penetration Testing (or Pentesting) is a structured and legal attempt to break into a system the same way a hacker would—**but with permission and documentation**.

The goal is simple:

> Find vulnerabilities before an attacker does, explain the risk, and guide the organization on how to fix them.

Pentesters don't just run tools. They think like attackers, understand systems deeply, and use a mix of manual and automated techniques.

Real-world objectives of pentesting

- Evaluate how secure networks, apps, and devices actually are
- Identify misconfigurations and weak design choices
- Test employee awareness and incident response
- Help businesses achieve compliance (ISO, PCI-DSS, HIPAA)
- Reduce financial, reputational, and legal risk

Pentesting is not hacking "for fun"—it's a **controlled, documented, business-driven security assessment**.

2. Types of Penetration Testing

a. Network Penetration Testing

Focus: Internal & external corporate networks

Examples:

- Open ports exposing sensitive services

- Outdated software running on servers
- Weak password policies
- Unsegmented networks

Tools often used: **Nmap, Masscan, Nessus, Metasploit**

Outcome:

A clear view of how an attacker could move laterally through a network.

---

b. Web Application Penetration Testing

Focus: Websites, portals, APIs, dashboards

Looks for:

- Input validation flaws (SQLi, XSS)
- Broken authorization (IDOR, privilege escalation)
- Business logic bypasses
- Insecure API endpoints
- Misconfigured server headers

Tools: **Burp Suite Pro, OWASP ZAP, SQLmap**

Outcome:

Proof of how a hacker could abuse the application to access data or accounts.

---

c. Mobile ApplicationPentesting

Focus: Android/iOS apps

Tests for:

- Insecure data storage
- API communication flaws
- Reverse engineering
- Insecure local databases
- Hardcoded credentials

Tools: **MobSF, JADX, Frida**

Outcome:

Ensures mobile users are safe and cannot be impersonated.

d. Wireless Pentesting

Focus: WiFi networks (Enterprise & Home)

Common findings:

- Weak encryption (WEP, WPA1)
- Default router passwords
- Rogue access points
- Weak enterprise authentication

Tools: **Aircrack-ng, Kismet**

e. Cloud Security Pentesting

Focus: AWS, Azure, GCP environments

What we check:

- Misconfigured S3 buckets
- Over-permissive IAM policies
- Exposed endpoints
- Credential leaks

Tools: **ScoutSuite, Pacu, Prowler**

Outcome:

Protection against cloud-specific attack vectors.

f. Social Engineering

Tests the human element.

Includes:

- Phishing emails
- Malicious attachments

- Fake login pages
- Phone-based exploitation

Social engineering is often the **entry point** of most real attacks.

---

g. Physical Pentesting

Tests physical security layers:

- Server rooms
- Badge access
- CCTV blind-spots
- Lock bypassing

Outcome:

Shows exploit paths that bypass technology entirely.

---

3. Pentesting Methodology (XSploit Premium Version)

This is the backbone of every real-world assessment.

A good pentest report is built on **methodology**, not shortcuts.

Step 1: Reconnaissance (Information Gathering)

Objective: Understand the target without touching it (passive) and then interact safely (active).

Passive Recon

- WHOIS records
- Company open-source information
- Employee LinkedIn profiles
- Shodan device search
- Subdomain enumeration

Why it matters:

Passive data often reveals forgotten assets—the easiest starting point for attackers.

Active Recon

- Ping sweeps
- Port scanning
- Service enumeration

Tools used: **Nmap, Amass, Subfinder, Shodan**

Step 2: Scanning & Enumeration

Objective: Map the system like a blueprint.

Scanning

- Identify open ports
- Detect running services
- Fingerprint OS versions

Enumeration

This is where pentesters extract:

- Users
- Emails
- Directories
- SMB shares
- Database schemas

Tools: **Nmap NSE, Enum4linux, SMBMap, Nikto**

Step 3: Vulnerability Assessment

You compare discovered services with known weaknesses.

What you analyze:

- CVEs

- Patch history
- Misconfigurations
- Framework vulnerabilities

Tools:

- **Nessus** – Industrial-grade scanning
- **OpenVAS** – Open-source
- **Burp Suite Scanner** – Web-specific

A good pentester always manually verifies scanner results.

Step 4: Exploitation

This is the practical attack phase.

Goal: Prove a vulnerability is real, **without causing damage**.

Examples:

- Gaining shell access through an exposed service
- Exploiting SQL injection to extract database
- Bypassing authentication

Tools: **Metasploit, SQLmap, Hydra**

Important rule:

> Exploit only what is in scope. Never exceed permission.

Step 5: Post-Exploitation

Now you behave like a real attacker who got inside.

Activities:

- Privilege escalation
- Lateral movement
- Credential harvesting
- Persistence (for simulation only)
- Data exfiltration (controlled & documented)

Tools:

- **Mimikatz** (Windows credentials)
- **LinPEAS / WinPEAS** (privesc detection)
- **BloodHound** (Active Directory paths)

Outcome:

Understanding the depth of damage a real attacker could do.

Step 6: Reporting (Most Valuable Skill)

This is what clients pay for.

A perfect report includes:

1. Executive Summary

One-page non-technical explanation for management.

2. Methodology

Clearly list the steps you followed.

3. Findings List (Table)

| Severity | Vulnerability | Impact | Proof | Fix |

4. Detailed Findings

- Description
- Technical impact
- Evidence/screenshots
- Exploitation path
- Remediation & reference links

5. Final Recommendations

Security best practices & hardening guide.

4. Critical Pentesting Concepts (Explained Simply but Professionally)

Attack Surface

Total number of entry points attackers could hit.

OWASP Top 10 (Web Security Standard)

A1–A10 most common web vulnerabilities.

CVSS Score

Industry scoring system to measure severity.

Threat Modeling

Predicting how an attacker will think.

Zero Trust Approach

Never trust; verify every action.

---

5. Pentesting Tools With Clean Explanations

Nmap

Scans networks to discover systems, ports, and services.

Burp Suite

Intercepts and manipulates web traffic for testing applications.

Metasploit Framework

Collection of exploits and payloads for controlled exploitation.

Wireshark

Captures and analyzes network packets.

Hydra

Brute-forces login credentials for services.

SQLmap

Fully automates SQL exploitation.

Dirsearch/Gobuster

Find hidden directories and files on web servers.

---

6. Recommended Learning Resources

Beginner Level

- TryHackMe — Complete Learning Path
- PortSwigger Academy — Web security
- OWASP Cheat Sheet Series

Intermediate

- HackTheBox Machines
- PentesterLab

Advanced

- Bug Bounty platforms (HackerOne, BugCrowd)
- Cloud security labs (AWS Goat, Azure Labs)

---

8. Bonus: Internal Pentest Checklist (For Your Students)

A ready-to-use checklist for practical use:

🟦 Pre-Engagement

- Scope confirmed
- Permissions documented

- NDA signed
- Emergency contact shared

🟦 Technical Checklist

- Identify exposed ports
- Scan for vulnerable services
- Check for outdated software
- Test authentication & session security
- Validate API endpoints
- Perform business logic tests
- Attempt privilege escalation
- Check logging & monitoring

🟦 Reporting Checklist

- Screenshots
- PoC
- Risk rating
- Fix steps