Kali Linux Penetration Testing Labs

Hands-on vulnerability discovery, enumeration, and exploitation in a controlled environment.

Objective	Conduct vulnerability scans and basic penetration tests using Kali Linux (Nmap, Nikto, Hydra) to simulate an attacker workflow and produce actionable remediation insights.
Outcome	Identified weak credentials, outdated web components, and misconfigurations; documented findings, evidence, and mitigations.
Duration	Approximately 6-10 hours across two lab sessions.
Skills	Reconnaissance, enumeration, web scanning, password attacks, Linux networking, report writing.

Ethics & Scope: All activities executed against lab-only targets I control, for defensive education. No unauthorized testing was performed.

1. Lab Environment

Platform: VMware / VirtualBox
Attacker: Kali Linux with core toolset
Targets: Ubuntu Server (web), legacy VM with SMB/FTP/SSH
Network: Isolated NAT / Host-Only network; no Internet exposure

Node	IP (example)	Purpose	Notes
Kali	192.168.56.10	Attacker / toolkit	Nmap, Nikto, Hydra, curl
Web-01	192.168.56.20	Apache/PHP app	Intentional weak configuration for learning
Legacy-01	192.168.56.30	FTP/SMB/SSH	Default credentials and outdated services (lab only)

2. Methodology & Workflow

2.1 Reconnaissance

• Identify live hosts and map exposed services/versions; build a prioritized target list.

2.2 Enumeration

• Deep-dive per-service checks (HTTP, SMB, FTP, SSH); fingerprint apps and confirm misconfigurations.

2.3 Exploitation (controlled)

• Validate risk using non-destructive proofs-of-concept; password attacks against lab accounts with consent.

2.4 Reporting & Remediation

• Document findings with evidence and map mitigation steps.

Representative Commands

```
# Host discovery & full TCP scan (safe defaults)
nmap -Pn -sS -sV -O -T3 192.168.56.0/24

# Focus scan for top ports + scripts
nmap -sC -sV -p- 192.168.56.20

# Web server probe
nikto -h http://192.168.56.20

# HTTP tech fingerprint (alternate)
whatweb http://192.168.56.20

# SMB enumeration (legacy target)
nmap --script smb-enum-shares,smb-enum-users -p445 192.168.56.30

# FTP anonymous test
nmap --script ftp-anon -p21 192.168.56.30

# Password attack (SSH example, small wordlist for lab)
hydra -l student -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.30 -t 4 -f
```

3. Key Findings

ID	Asset	Issue	Evidence (summary)	Risk
F-01	Web-01 (Apache)	Directory listing enabled	GET /uploads/ reveals files; Nikto confirms.	Medium
F-02	Web-01 (PHP app)	Outdated component (jQuery 1.x)	whatweb and page source show vulnerable version.	High
F-03	Legacy-01 (SSH)	Weak credentials	Hydra cracked 'student:student123'.	High
F-04	Legacy-01 (FTP)	Anonymous login allowed	nmap ftp-anon: read access to /pub.	Medium
F-05	Legacy-01 (SMB)	Over-permissive share	smb-enum-shares shows READ on 'public'.	Medium

Evidence Snippets

```
# Nikto (excerpt)
+ Server: Apache/2.4.41 (Ubuntu)
+ The X-Frame-Options header is not present.
+ Uncommon header 'x-powered-by' found, with contents: PHP/7.4.3
+ OSVDB-3092: /uploads/: Directory indexing found.

# Hydra (excerpt)
[22][ssh] host: 192.168.56.30   login: student   password: student123
1 of 1 target successfully completed, 1 valid password found

# Nmap SMB scripts (excerpt)
| smb-enum-shares:
|   account_used: guest
|   \\192.168.56.30\public: Read
|_  \\192.168.56.30\IPC$: Read
```

4. Remediation Plan

Finding	Recommended Fix	Rationale / Reference
F-01	Disable directory listing in Apache (Options -Indexes); restrict /uploads/.	Reduce unintended data exposure; align with CIS Apache benchmarks.
F-02	Upgrade front-end libraries; pin to supported LTS versions; enable Subresource Integrity (SRI).	Eliminate known client-side vulnerabilities; integrity checking prevents tampering.
F-03	Enforce strong password policy; disable SSH password auth or enable MFA; consider fail2ban.	Prevents brute-force; reduces credential stuffing risk on SSH.
F-04	Disable FTP or require auth over FTPS; restrict anonymous access.	Prevents data leakage; encrypts credentials in transit.
F-05	Harden SMB shares with least privilege; audit guest access; enable signing.	Minimize lateral movement and unauthorized read access.

Validation Steps (Post-Fix)

```
# Verify directory listing disabled
curl -I http://192.168.56.20/uploads/

# Verify SSH hardening (no password auth)
nmap -p22 --script ssh2-enum-algos 192.168.56.30
ssh -o PreferredAuthentications=password -o PubkeyAuthentication=no user@192.168.56.30  # should fail

# Confirm SMB access tightened
smbclient -L //192.168.56.30 -N
```

5. Metrics & Learning Outcomes

Coverage: Two target hosts; approximately 1,000+ ports assessed.
Credentials Tested: Controlled wordlist; one lab account cracked.
Notable Lessons:

- Service fingerprinting improves prioritization and reduces noise.
- Evidence-driven reporting turns scan output into actionable fixes.
- Password attacks must be tightly scoped and rate-limited to avoid service disruption.

Next Steps (Roadmap)

- Add Burp Suite for authenticated web testing and content discovery.
- Integrate OpenVAS or Nessus for differential scans over time.
- Build scheduled scans in a disposable lab to practice triage.
- Explore Ansible hardening playbooks to automate remediations.

Appendix A – Command Cheat Sheet

```
# Nmap quicks
nmap -sC -sV -oN nmap_initial.txt 192.168.56.0/24
nmap -p80,443 --script http-enum,http-headers 192.168.56.20

# Web recon
nikto -h http://TARGET
whatweb http://TARGET

# SMB
nmap --script smb-os-discovery -p445 TARGET
smbclient -L //TARGET -N

# FTP
nmap --script ftp-anon -p21 TARGET
ftp TARGET

# Hydra
hydra -L users.txt -P passwords.txt ssh://TARGET -t 4 -f -o hydra_ssh.txt
```

Appendix B – Glossary

Enumeration: Systematic extraction of service and user details to expand the attack surface.
Proof of Concept (PoC): Minimal, controlled action that demonstrates exploitability without damage.
Least Privilege: Granting only the minimal access necessary to perform a function.

All testing confined to owned lab assets. These notes are for educational and defensive purposes.