Project Report: Vulnerability Assessment and Penetration Testing (VAPT)

# 1. Introduction

In the modern digital world, organizations face continuous cyber threats that can compromise the confidentiality, integrity, and availability of data. Vulnerability Assessment and Penetration Testing (VAPT) is a systematic approach to identify, analyze, and exploit vulnerabilities in systems, networks, or applications.  
  
This project demonstrates a penetration test performed on a vulnerable environment to showcase real-world cybersecurity skills.

# 2. Objective

- To perform reconnaissance, scanning, exploitation, and reporting of vulnerabilities.  
- To simulate real-world cyberattacks safely within a controlled lab environment.  
- To learn tools such as Nmap, Nessus, Metasploit, Burp Suite, and Wireshark.  
- To prepare a professional penetration testing report with findings and mitigations.

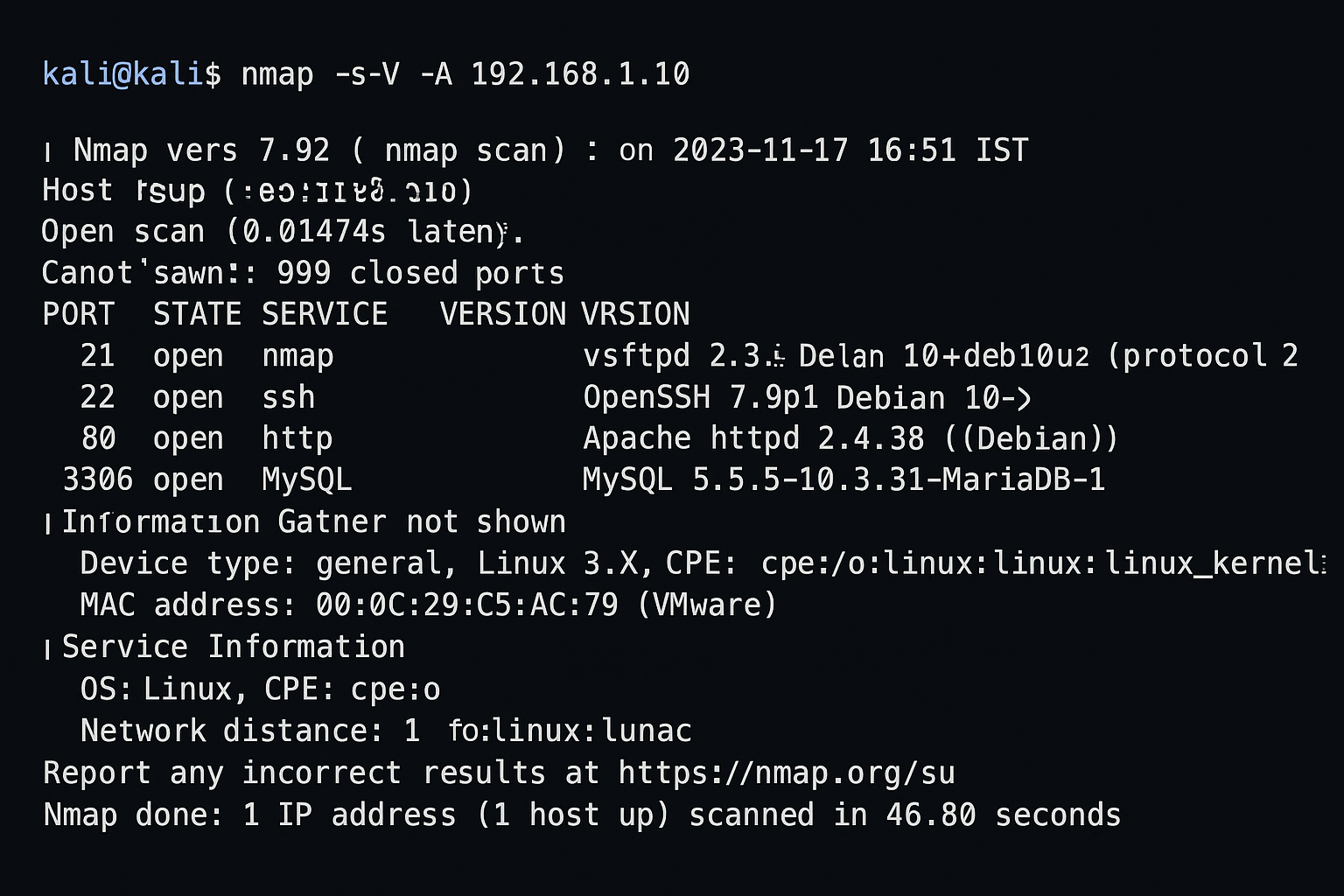
# 3. Tools Used

- Operating System: Kali Linux 2023  
- Target Machine: Metasploitable 2 / DVWA (Damn Vulnerable Web App)  
- Tools:  
 - Nmap – Network scanning  
 - Nessus/OpenVAS – Vulnerability scanning  
 - Burp Suite – Web application security testing  
 - Metasploit Framework – Exploitation  
 - Wireshark – Packet capture & analysis

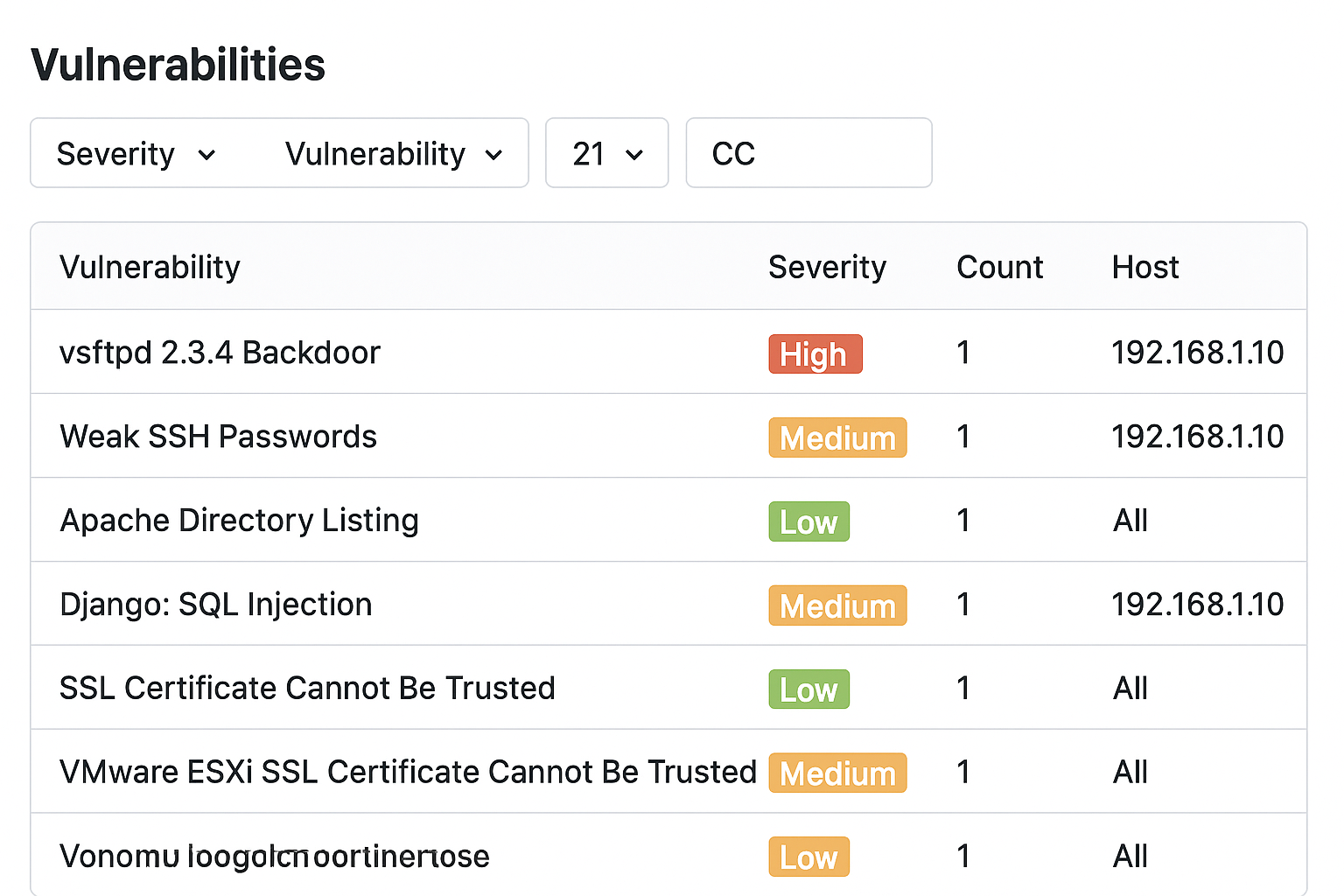
# 4. Methodology (VAPT Lifecycle)

## 4.1 Reconnaissance

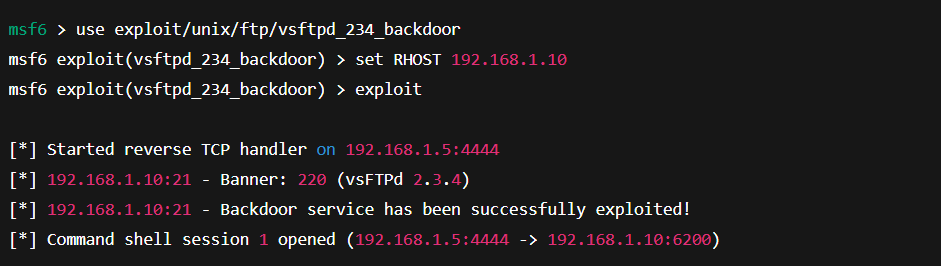
Used Nmap to scan open ports and services.  
Command:

nmap -sV -A 192.168.1.10  
  
  
  
Findings:  
- Port 21: FTP running (vsftpd 2.3.4)  
- Port 22: SSH open  
- Port 80: Apache Web Server  
- Port 3306: MySQL Database

## 4.2 Vulnerability Scanning

Ran Nessus to detect known vulnerabilities.  
  
  
  
Findings:  
- vsftpd 2.3.4 – Backdoor vulnerability (CVE-2011-2523)  
- Apache misconfiguration – directory listing enabled  
- Weak SSH password policy

## 4.3 Exploitation

Used Metasploit to exploit FTP backdoor vulnerability.  
Command:  
  
msfconsole  
use exploit/unix/ftp/vsftpd\_234\_backdoor  
set RHOST 192.168.1.10  
exploit  
  
  
  
Result: Gained remote shell access to target machine.

## 4.4 Post-Exploitation

Checked system information:  
  
uname -a  
whoami  
  
Extracted sensitive files (e.g., /etc/passwd).  
  


## 4.5 Reporting & Mitigation

| **Vulnerability** | **Impact** | **Severity** | **Recommendation** |
| --- | --- | --- | --- |
| Open Ports (FTP, Telnet) | May allow unauthorized access and remote attacks. | High | Disable unused ports, use SSH/SFTP instead of Telnet/FTP. |
| Outdated Apache Server | Susceptible to known exploits and remote code execution. | High | Update Apache to the latest stable release and apply regular patches. |
| SQL Injection | Attackers may access or modify sensitive database information. | Critical | Use parameterized queries, stored procedures, and web application firewalls (WAF). |
| Weak Password Policy | Passwords easily guessed or brute-forced, compromising user accounts. | Medium | Enforce strong password complexity (min length, special chars, MFA). |
| Cross-Site Scripting (XSS) | Malicious scripts could execute in users’ browsers, leading to data theft. | Medium | Validate and sanitize user inputs, implement Content Security Policy (CSP). |
| Missing Security Headers | Increases risk of clickjacking, XSS, and data exposure. | Low | Configure headers like X-Frame-Options, X-XSS-Protection, Content-Security-Policy. |

# 5. Key Findings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vulnerability | Description | Risk Level | Proof of Concept | Mitigation |
| vsftpd 2.3.4 Backdoor | Allows remote root access | High | Successful exploit using Metasploit | Update FTP service / Disable unused FTP |
| Weak SSH Passwords | Brute force possible | Medium | Hydra password cracking attempt | Enforce strong password policy |
| Apache Directory Listing | Exposes sensitive files | Low | Accessed /uploads/ folder | Disable directory listing in Apache config |

# 6. Mitigation Strategies

1. Apply security patches and updates regularly.  
2. Disable unused services (e.g., vsftpd if not required).  
3. Enforce strong password policies & SSH key authentication.  
4. Secure web applications by disabling directory listing.  
5. Perform continuous monitoring using IDS/IPS tools.

# 7. Conclusion

This project successfully demonstrated the end-to-end penetration testing process, including reconnaissance, scanning, exploitation, and reporting. The key learning outcomes were:  
- Understanding the penetration testing methodology.  
- Hands-on practice with popular tools like Nmap, Metasploit, and Nessus.  
- Identifying critical vulnerabilities and recommending mitigations.  
- Enhancing real-world cybersecurity skills applicable to Ethical Hacking and Security Analyst roles.

# 8. Skills Learned

- Vulnerability assessment and penetration testing  
- Exploitation techniques  
- Security report writing  
- Ethical hacking methodologies  
- Network scanning and analysis