Vulnerability Scanning Report

1. Objective

The main objective of this practical task was to perform vulnerability scanning on a target system in a controlled laboratory environment. The goal was to identify open network ports, active services, and possible security weaknesses using automated vulnerability scanning tools. Based on the results, the identified vulnerabilities were analyzed and prioritized according to their severity and potential impact.

2. Scope of Assessment

- **Target System:** Metasploitable Virtual Machine
- **IP Address:** 192.168.75.131
- **Environment:** Controlled lab setup
- **Assessment Type:** Non-intrusive vulnerability scanning

The assessment was limited to scanning and analysis only. No exploitation or service disruption activities were performed.

3. Tools Used

The following tools were used during the assessment:

- **Nmap** – To scan open ports and identify running services
- **OpenVAS** – To perform automated vulnerability assessment
- **Nikto** – To scan the web server for common vulnerabilities and misconfigurations

4. Methodology

A step-by-step vulnerability scanning methodology was followed:

1. **Network scanning** was performed using Nmap to detect open ports and services running on the target system.
2. **Vulnerability assessment** was carried out using OpenVAS to identify known vulnerabilities, outdated software, and insecure configurations.

3. **Web server scanning** was performed using Nikto to identify web-related vulnerabilities and misconfigurations.
4. All identified issues were reviewed and categorized based on their severity and possible security impact.

5. Key Findings

5.1 Network-Level Findings (Nmap)

The Nmap scan revealed multiple open ports and services such as FTP, SSH, Telnet, SMTP, HTTP, MySQL, PostgreSQL, and Apache Tomcat. Having many exposed services increases the attack surface and makes the system more vulnerable to attacks.

5.2 Vulnerability Assessment Findings (OpenVAS)

The OpenVAS scan detected several vulnerabilities with different severity levels. Some vulnerabilities were classified as **high** and **medium risk**, mainly due to outdated services and insecure system configurations. These weaknesses could potentially be exploited by attackers if left unpatched.

5.3 Web Server Findings (Nikto)

Nikto identified multiple web server issues, including:

- Missing security-related HTTP headers
- Directory listing enabled
- Public exposure of phpMyAdmin and phpinfo files
- Use of an outdated Apache web server version

These issues increase the risk of information disclosure and unauthorized access.

6. Risk Analysis

The presence of outdated software, exposed administrative interfaces, and multiple open services creates a high security risk. If an attacker exploits these vulnerabilities, it could result in unauthorized system access, sensitive data leakage, or complete system compromise.

7. Remediation Recommendations

To reduce security risks, the following actions are recommended:

- Update and patch the Apache web server to a supported version
- Restrict access to phpMyAdmin and remove unnecessary test files
- Disable directory indexing on the web server
- Close unused ports and restrict services using firewall rules
- Follow secure configuration practices and conduct regular vulnerability scans

8. Conclusion

This vulnerability scanning exercise successfully identified several critical security weaknesses in the target system. The results clearly show the importance of regular vulnerability assessments and proper system hardening. By implementing the recommended remediation steps, the system’s attack surface can be significantly reduced, improving its overall security posture.