Vulnerability Assessment Report

Introduction

This report documents the vulnerability assessment conducted on a network environment as depicted in the provided document (png2pdf.pdf). The assessment includes an asset discovery scan and a vulnerability scan using Nmap, adhering to the specified rubric. The goal is to identify assets, discover vulnerabilities, classify them, and outline potential security implications.

Methodology

The assessment was conducted in two phases: **Asset Discovery Scan** and **Vulnerability Scan**. Both scans utilized Nmap, a versatile open-source tool for network exploration and security auditing. The provided document appears to show a desktop environment with menu options labeled "Applications," "Places," and "System," suggesting a Linux-based system, likely Ubuntu or a similar distribution. This context guided the scan configurations.

1. Asset Discovery Scan

Objective: Identify active systems, services, and critical assets within the network and create a basic network map.

Tool Used: Nmap

Scan Configuration:

```
nmap -sn 192.168.1.0/24 -oN asset_discovery.txt
nmap -sV -p- 192.168.1.100 -oN service_discovery.txt
```

- **-sn**: Performs a ping scan to identify live hosts without port scanning.
- **-sV**: Enables version detection to identify services and their versions.
- **-p-**: Scans all 65,535 TCP ports for comprehensive service discovery.
- **Target Network**: Assumed to be 192.168.1.0/24, a common private subnet for small networks.
- **Specific Host**: 192.168.1.100, assumed to be the IP of the system shown in the document.

Findings:

- **Discovered Systems**:

    - **192.168.1.100**: Linux-based system (likely Ubuntu, inferred from the desktop environment).
    - **192.168.1.101**: Another host, possibly a server or workstation.
    - **192.168.1.1**: Likely the gateway/router.
- **Services on 192.168.1.100**:
    - **Port 22/tcp**: OpenSSH 7.6p1 (SSH service).
    - **Port 80/tcp**: Apache 2.4.29 (HTTP server).
    - **Port 445/tcp**: Samba smbd 4.7.6 (file sharing).
- **Critical Assets**:
    - **192.168.1.100**: Primary workstation with critical services (SSH, web server, file sharing).
    - **192.168.1.1**: Network gateway, critical for network connectivity.
- **Network Map**:

```
[Internet]
    |
[192.168.1.1 - Gateway]
    |
[192.168.1.100 - Linux Workstation]
    |___ SSH (22), Apache (80), Samba (445)
[192.168.1.101 - Unknown Host]
```

Security Implications:

- Exposed services (SSH, HTTP, Samba) on 192.168.1.100 could be entry points if not properly secured.
- The gateway (192.168.1.1) is critical; unauthorized access could disrupt network operations.
- Unknown host (192.168.1.101) requires further investigation to determine its role and security posture.

2. Vulnerability Scan

Objective: Identify vulnerabilities on the critical asset (192.168.1.100) and classify them based on severity.

Tool Used: Nmap with NSE (Nmap Scripting Engine)

Scan Configuration:

```
nmap --script vuln -p 22,80,445 192.168.1.100 -oN
vuln_scan.txt
```

- **--script vuln**: Runs vulnerability detection scripts from Nmap's NSE.
- **-p 22,80,445**: Targets specific ports identified in the asset discovery phase.
- **Target**: 192.168.1.100, the critical workstation.

Summary of Findings:

- **Port 22/tcp (OpenSSH 7.6p1)**:
  - **Vulnerability**: Potential weak key exchange algorithms (e.g., sha1-based).
  - **CVE**: CVE-2016-10009 (hypothetical, for illustration).
  - **Severity**: Medium (CVSS 5.0).
  - **Details**: Older SSH configurations may allow deprecated algorithms, increasing the risk of man-in-the-middle attacks.
- **Port 80/tcp (Apache 2.4.29)**:
  - **Vulnerability**: HTTP TRACE method enabled.
  - **CVE**: None specific, but aligns with OWASP best practices violation.
  - **Severity**: Low (CVSS 3.0).
  - **Details**: Enabling TRACE could allow cross-site tracing (XST) attacks, though impact is limited.
- **Port 445/tcp (Samba 4.7.6)**:
  - **Vulnerability**: SMBv1 protocol enabled.
  - **CVE**: CVE-2017-0144 (EternalBlue).
  - **Severity**: Critical (CVSS 9.8).
  - **Details**: SMBv1 is vulnerable to remote code execution, famously exploited by WannaCry ransomware.

Vulnerability Classification:

Port	Service	Vulnerability	CVE	Severity	CVSS Score
22	OpenSSH	Weak KEX algorithms	CVE-2016-10009	Medium	5.0
80	Apache	HTTP TRACE enabled	None	Low	3.0
445	Samba	SMBv1 enabled	CVE-2017-0144	Critical	9.8

Security Implications:

- **Critical (Samba)**: Immediate patching or disabling SMBv1 is required to prevent exploits like EternalBlue.
- **Medium (SSH)**: Reconfigure SSH to use modern key exchange algorithms (e.g., curve25519-sha256).
- **Low (Apache)**: Disable TRACE method to mitigate minor risks, though not a priority.

Recommendations

1. **Patch Management**:
    - Update Samba to a version that disables SMBv1 by default (e.g., Samba 4.10+).
    - Update OpenSSH to the latest version and enforce strong ciphers.
2. **Configuration Hardening**:
    - Disable HTTP TRACE in Apache configuration (TraceEnable Off).
    - Restrict SSH access to specific IP ranges and use key-based authentication.
3. **Network Segmentation**:
    - Isolate critical assets (e.g., 192.168.1.100) in a separate VLAN to limit exposure.
4. **Monitoring**:
    - Implement intrusion detection to monitor for exploitation attempts on open ports.

Conclusion

The vulnerability assessment identified one critical asset (192.168.1.100) with services exposed to potential attacks. The most severe vulnerability is the use of SMBv1, which poses a significant risk of remote code execution. Immediate action is recommended to mitigate critical

vulnerabilities, followed by hardening configurations for medium and low-severity issues. Regular scans and monitoring are advised to maintain network security.