Penetration Testing Report on Ubuntu Machine (Black Box Approach)

Date: September 01, 2025

Author: Sebin Mathew

1. Introduction

This report summarizes a black box penetration test performed on an Ubuntu 14.04 LTS machine. In a black box test, the tester begins without any prior knowledge of the system—no usernames, credentials, or network structure are provided—simulating a real-world attack scenario.

The primary goal of this assessment was to identify potential vulnerabilities, gain unauthorized access, and locate CTF flags hidden within the machine. The system is Star Wars-themed, with multiple user accounts, some of which were intentionally vulnerable.

2. Methodology

The testing process followed a structured approach to uncover security weaknesses:

- **Host Discovery:** The machine's IP address was identified using `arp -a`, revealing `192.168.56.102`.

- **Port Scanning:** Using Nmap (`nmap -sV 192.168.56.102`), open ports were detected, including SSH (22) and FTP (version ~7.5).

- **FTP Exploitation for Username Discovery:**
  An FTP vulnerability in version 7.5 allowed us to enumerate valid usernames without authentication. A carefully crafted payload, executed via Nmap, revealed `luke_skywalker` as a valid user.

- **SSH Access:** Using the discovered username and password `like_my_father_beforeme`, we successfully logged in via SSH.

- **Privilege Escalation:** The same password enabled escalation to root via `sudo su`.

- **File Analysis:** Flags and sensitive information were located using `find`, `grep`, and `strings`. Hidden files in user directories (`.*`) were also examined to uncover overlooked data.

3. Findings

3.1 Network & Access

- The machine was fully exposed on the local network at `192.168.56.102`.
- Weak SSH credentials and misconfigured sudo privileges made root access straightforward.
- Multiple user accounts existed, some possibly unused, increasing the attack surface.

3.2 Hidden Files & Configurations

- Standard hidden files, such as `.profile`, `.bashrc`, and `.bash_logout`, were present.
- The `vagrant` user's `.npm` directory contained outdated Node packages, introducing potential vulnerabilities.
- Root's `.m2` directory was empty, and `.bash_history` had been removed, indicating a lack of audit trails.

3.3 CTF Flags

- **The main flag was discovered: `VolgaCTF{15de975cfd8a8b36ff14c9ec9d1c72ef}`.**
- **Additional flags included `CTF{0811c36c88a44e0640328eca1d063df8}` and `CTF{4c4988d8c8a86b3bc5143eba1098c2f3}`.**
- **Flags were primarily located in hidden caches and directories, such as `/home/vagrant/.bundle/cache/...`.**

3.4 Suspicious Files

- **Files like `/home/kylo_ren/.secret_files/my_recordings_do_not_open.iso` may contain hidden data; attempts to mount failed.**

- **Media files (.wav, .png) were present and may conceal information via steganography.**

3.5 Other Vulnerabilities

- Ubuntu 14.04 LTS is end-of-life, exposing the system to unpatched exploits.

- The FTP service allowed username enumeration via Nmap payloads without authentication.

- Root access lacked proper auditing, leaving no trace of administrative activity.

4. Recommendations

- Enforce strong passwords and enable multi-factor authentication.

- Remove or disable unused user accounts to reduce exposure.

- Upgrade the operating system to a supported version to mitigate unpatched vulnerabilities.

- Restrict network access to SSH and FTP services using firewalls.

- Audit all hidden and media files for malware or concealed information.

- Patch or restrict vulnerable FTP services to prevent unauthorized username enumeration.

5. Conclusion

The black box penetration test successfully demonstrated how a malicious actor could exploit weak credentials and misconfigurations to compromise the system:

- Usernames were enumerated using an FTP vulnerability and Nmap payloads.
- SSH login was achieved using weak credentials.
- Privilege escalation allowed full root access.
- Multiple CTF flags were discovered, highlighting overlooked sensitive data.

Overall, the Ubuntu 14.04 LTS machine showed significant security weaknesses, including outdated software, weak passwords, exposed services, and poor auditing practices. Addressing these issues is essential to safeguard sensitive information and improve the system’s security posture.