# Lab 1: Personal System Vulnerability Assessment & Patch Management

**Purpose & Objectives**

This lab introduces you to foundational aspects of personal system security. You will learn to inventory installed software, verify its patch status, and identify potential unpatched vulnerabilities on your own computer. This exercise emphasizes the importance of proactive patch management and understanding your system's security posture, laying the groundwork for addressing common security issues.

### Relevant Course Outcomes Addressed:

- **Knowledge:**
  - Demonstrate proficiency in using both Linux and Windows operating systems, including the command-line interface (CLI).
  - Identify and describe common security tools used for security testing.
- **Skills:**
  - Utilize shell and Python scripting to automate tasks and conduct security testing (encouraged for advanced students).
- **Competence:**
  - Identify simple security vulnerabilities.

---

## Assignment Task: Patch Scanning Your Personal System

For this lab, you will perform a basic vulnerability assessment and patch status check on *one* of your personal computers. The objective is to understand the software installed, determine if it's up-to-date, and identify any known unpatched vulnerabilities.

### Instructions:

1. **Choose Your System:** Select *one* personal computer running either **Linux** or **Windows** for which you have administrative access. You will perform all tasks on this chosen system. If you work in a group, I encourage every member of the group to do all the steps on their own computer. When writing the report make separate sub-sections with the results for each machine you worked on.
2. **Software Inventory (Installed Applications & Versions):**
   - **Goal:** Compile a comprehensive list of all installed applications and their specific versions.
   - **Windows:**
     - Use PowerShell (`Get-WmiObject -Class Win32_Product | Select-Object Name,Version`) or Command Prompt (`wmic product get name,version`).

  - Alternatively, manually inspect "Apps & features" in the Control Panel.
  - Consider using `winget list` for applications installed via Windows Package Manager.
- **Linux:**
  - Utilize your distribution's package manager commands (e.g., `dpkg -l` for Debian/Ubuntu, `rpm -qa` for Red Hat/Fedora, `pacman -Q` for Arch Linux).
  - List major applications (browsers, office suites, virtualization software, development tools, etc.) and their exact versions.
- Don't forget to list applications that you manually installed (without using a package manager)!

3. **Patch Status Verification (Are Updates Installed?):**
   - **Goal:** For the software identified in step 2, determine if the latest stable updates or patches are installed.
   - **Operating System Updates:**
     - **Windows:** Check "Windows Update" settings. Note any pending updates.
     - **Linux:** Run your system's update commands (e.g., `sudo apt update && sudo apt upgrade`, `sudo yum update`, `sudo dnf upgrade`, `sudo pacman -Syu`). Note any packages awaiting updates, especially security patches.
   - **Application Updates:** For major third-party applications (web browsers, media players, PDF readers, etc.), check their built-in update mechanisms or visit the vendor's official website to compare your installed version with the latest stable release.
4. **Vulnerability Identification (Unpatched Vulnerabilities):**
   - **Goal:** Research and identify known vulnerabilities for any *outdated* software versions you found, or even for current software if public exploits are known to exist without immediate patches. This involves **research**, not active exploitation.
   - **Methodology:**
     - For each piece of software identified as outdated (from step 3) and any critical applications you want to investigate further, use public vulnerability databases to search for associated security flaws.
     - **Recommended Resources:**
       - **NVD (National Vulnerability Database):** https://nvd.nist.gov
       - **CVE Details:** https://www.cvedetails.com
       - **Exploit-DB:** https://www.exploit-db.com
       - Vendor security advisories (e.g., Microsoft Security Response Center, Mozilla Security Advisories, etc.).
       - Feel free to use and mention other resources in your report.
     - Document any critical or high-severity vulnerabilities found, focusing on those for which your system is currently susceptible due to an unpatched state or an existing known flaw in your version. This includes writing down the CVE number and a short description of the vulnerability.

## Deliverables:

Submit a report in **PDF format** (2-4 pages, excluding appendices) structured as follows:

1. **Title Page:**
   - Course Name, Lab 1: Personal System Vulnerability Assessment & Patch Management
   - Your Full Name, Student ID
   - Date
2. **Introduction:**
   - Briefly state the purpose of the lab.
   - Provide basic specifications of the analyzed system (e.g., OS version, CPU, RAM, primary storage type).
3. **Methodology:**
   - Clearly describe the steps you took to gather information for each task.
   - Include specific commands used (e.g., `Get-WmiObject -Class Win32_Product`), menu navigation paths, and any tools or websites consulted.
   - Explain how you determined patch status and researched vulnerabilities.
4. **Findings:**
   - **Software Inventory Summary:** Present a concise list of major installed applications and their versions.
   - **Patch Status:**
     - List any operating system updates that were pending or not yet installed at the time of your assessment.
     - List any major third-party applications that you identified as *not* running the latest stable version.
   - **Identified Vulnerabilities:** For each significant vulnerability found (e.g., high/critical severity, easily exploitable):
     - State the affected Software Name and Version.
     - Provide a brief description of the vulnerability (e.g., "Remote Code Execution via [specific attack vector]").
     - Include the CVE ID (e.g., CVE-2023-XXXXX), if applicable.
     - Specify the Severity (e.g., Critical, High, Medium).
     - Provide a direct link to the source of the vulnerability information (e.g., NVD entry, vendor advisory).
5. **Conclusions & Recommendations:**
   - Summarize the overall security posture of your system based on your findings (e.g., "My system appears relatively well-patched but has X critical unaddressed vulnerabilities").
   - Provide specific, actionable recommendations for improving your system's security based on your discoveries (e.g., "Update [Application Name] to version X.X," "Enable automatic updates for all core applications," "Remove unused software [Application Name]").

- Reflect on the importance of regular patch management and how it contributes to overall cybersecurity.

6. **References/Appendices:**
   - List any external resources (websites, articles) cited.
   - Optionally include screenshots of command outputs, "Apps & features" lists, or relevant vulnerability database entries to support your findings.

---

## Grading Criteria:

- **Completeness of Inventory (20%):** Thoroughness in listing installed software and their versions.
- **Accuracy of Patch Status (25%):** Correct identification of outdated software and pending OS updates.
- **Vulnerability Identification & Detail (30%):** Accuracy in finding relevant CVEs/vulnerabilities, providing clear descriptions, and linking to credible sources.
- **Report Quality (25%):** Clarity, organization, professional presentation, detailed methodology, and insightful conclusions/recommendations.

---

**Important Notes:**

- **Safety First:** Only run commands and use tools that you understand and trust. Do not install untrusted software or attempt to exploit vulnerabilities during this lab. Focus on *identification* and *research*.
- **Ethical Considerations:** You are assessing *your own* personal computer. Do not perform these activities on systems you do not own or for which you lack explicit permission.
- **Troubleshooting:** If you encounter difficulties with commands or research, refer to your OS documentation, online resources, or seek assistance during dedicated lab hours.
- **Remediation (Optional but Recommended):** While this lab focuses on identifying issues, you are strongly encouraged to apply patches and implement your recommendations *after* submitting the assignment to enhance your system's security.