Lab 1: Personal System Vulnerability Assessment & Patch Management Report

Introduction to Cyber Security

Your Name(s): Alexander Joseph Emilsson, Alfa Reynisdóttir, Gísli Hrafn Halldórsson, Kim Anna Hudson
Student ID(s): 050904-2450 alfar24, 080403-2750 gislih24, 241297-4709 kim24

Date: November 24, 2025

Analyzed System Operating System(s):
Debian GNU/Linux 12 x86_64, MacOS 13.7.8 (22H730), Microsoft Windows 11 Pro 10.0.26200 64-bit

1 Introduction

Analyzed System Specifications:

Alexander:

- **Operating System:** MacOS 26.1 (25B78)
- **Processor:** Apple M1
- **RAM:** 8 GB
- **Storage:** 245,11 GB
- **Network Interfaces:**
- **System Uptime at Assessment:** 5 days 14 hours

Alfa:

- **Operating System:** Debian 12 (bookworm)
- **Processor:** 12th Gen Intel i7-1260P (16) @ 4.700GHz
- **RAM:** 16 GB DDR4
- **Storage:** 1 TB SSD
- **Network Interfaces:** PCH HECI Controller
- **System Uptime at Assessment:** 2 hours and 14 minutes

Gísli:

- **Operating System:** Microsoft Windows 11 Pro 10.0.26200 64-bit
- **Processor:** Intel(R) Core(TM) Ultra 5 235U 4.9 Hz
- **RAM:** 32 GB DDR5
- **Storage:** 512 GB SSD
- **Network Interfaces:** Intel(R) Wi-Fi 7 BE201 320 MHz 122 Mbps
- **System Uptime at Assessment:** 4 hours and 17 minutes

Kim:

- **Operating System:** MacOS 13.7.8 (22H730)
- **Processor:** 2.3 GHz Dual-Core Intel Core i5
- **RAM:** 8 GB 2133 MHz LPDDR3
- **Storage:** 120 GB SSD
- **Network Interfaces:** Thunderbolt Bridge, Wi-fi, Thunderbolt 1, Thunderbolt 2
- **System Uptime at Assessment:** 39 mins

2 Methodology

2.1 Software Inventory

To compile a list of installed applications and their versions, the following methods were employed:
Method/Commands/Tools Used:

- **Alfa:** I used the command `dpkg -l` to list all packages I have.
  I also tried to run `sudo apt update && upgrade`, but I did that this morning, so nothing was out of date. I have a habit of updating and upgrading every Monday.
- **Kim:** I used `brew list -versions` but it only showed what packages had been installed with homebrew. I redirected system_profiler SPApplicationsDataType into a text file and that gave me a comprehensive list. I also found a comprehensive list under System Information > Applications.
- **Alexander:** I used `brew list -versions` to see the packages and dependencies I have installed. Then under System Information > Applications I found all the applications I´m using.
- **Gísli:** I used `Get-WmiObject -Class Win32_Product | Select-Object Name,Version` to list the currently installed packages.
- **Additional Steps:**

2.2 Patch Status Verification

Method/Commands/Tools Used:

- **Kim:** To find any pending MacOS updates: `softwareupdate -list` . For apps I installed normally (not through Homebrew), I needed to check each one manually. There is a Third-party tool which checks popular apps for updates automatically called MacUpdater.
- **Gísli:** I used `winget update` to discover packages that need updating.
- **Alexander:** I used `brew upgrade` to automatically update all packages that need updating (all were up to date). My mac updates periodically but every 3-4 months I wipe everything from my computer and reinstall the newest macOs version. (I had done so a week prior to this exercise)

2.3 Vulnerability Identification

- **All:** Using www.cvedetails.com and searching the current version of our currently installed programs that needed updating.

3 Findings

3.1 Software Inventory Summary

Below is a summary of major applications, their versions and patch status.

Table 1: Summary of Major Installed Software

Software Name	Person	Installed Version	Latest Stable Version	Status (Up-to
Google Chrome	Kim	117.0.5938.92	142.0.7444.134	Outdat
Zoom	Kim	6.5.12	6.3.10	Outdat
Microsoft Visual C++ 2015–2022 Redistributable (x86)	Gísli	14.32.31332.0	14.44.35211.0	Outdat
Git	Alfa	2.39.5	2.52.0	Outdat

*This table provides a concise summary. A full list is in Appendix A.

3.2 Identified Vulnerabilities

Through research of public vulnerability databases for the identified outdated software and other critical components, the following significant vulnerabilities were found to be potentially affecting the system:

Table 2: Identified Vulnerabilities

Software	CVE ID	Description	Severity	Source Link
Zoom	CVE-2025-49457	Untrusted search path in certain Zoom Clients for Windows may allow an unauthenticated user to conduct an escalation of privilege via network access	Critical (9.6)	NVD Link
Chrome	CVE-2025-13223	Type Confusion in V8 in Google Chrome prior to 142.0.7444.175 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.	High (8.8)	NVD Link
Git	CVE-2025-48384	Broken Config files allowed Arbitrary Code Execution	High (8.0)	NVD Link
Microsoft Visual C++ 2015–2022 Redistributable (x86)	CVE-2024-43590	Visual C++ Redistributable Installer Elevation of Privilege Vulnerability	High (7.8)	NVD Link, cvedetails link

*Only critical and high-severity vulnerabilities affecting installed versions are listed.

- **Example Explanation:** CVE-2023-XXXXX, affecting Microsoft Office 365, is a critical remote code execution vulnerability. As my installed version of Office 365 (`16.0.16924.20180`) is outdated, it is susceptible to this flaw. An attacker could exploit this by sending a malicious email, potentially gaining control of the system upon opening.

4 Conclusions & Recommendations

4.1 Actionable Recommendations

To enhance the security of the analyzed system, the following recommendations are crucial:

- **Update Git** to Version 2.52.0
- **Update Google Chrome** to version 142+
- **Update Zoom** to version 6.3.10 or later
- **Update Microsoft Visual C++ 2015–2022 Redistributable (x86)** to version 14.40.x or newer
- Other:
  - Remember to run `apt update && apt upgrade` every now and then.
  - Enable automatic updates for all applications.
  - Remove unused and unnecessary applications.
  - @Kim: Consider upgrading to higher macOS :')
  - @Alexander: You on the other hand may be doing too much

4.2 Reflection on Patch Management

Even a single unpatched application can compromise an entire system. Key Lessons Learned:

1. Vulnerabilities are constantly discovered.

   Firefox for example had 155 vulnerabilities discovered in 2025 alone.

2. The attack surface matters.

   More apps = more potential vulnerabilities. Unused software also pose risks.

3. Security requires vigilance and maintenance.

   Automatic updates are not always sufficient - manual verification is necessary.

References

- Add links to specific vendor advisories or articles cited