**Security Vulnerability Assessment Report**

**Project Overview**

* **Software Assessed:** [Google Chrome]
* **Assessment Date:** [22/07/2024]
* **Assessor:** [Muhammad Saad Shaikh]

**Table of Contents**

1. Introduction

2. Identified Vulnerabilities

3. Exploitation Methods

4. Mitigation Strategies

5. Conclusion

**1. Introduction**

* **Overview of the Software:**

(Brief description of the software and its main functions)

**Google Chrome** is a popular web browser developed by Google. Here are its main functions:

1. **Web Browsing:** Allows you visit websites and use web apps.

2. **Omnibox:** Google combines the search and address bar for quick searches and other tasks.

3. **Extensions:** Supports add-ons like ad blockers and password managers.

4. **Sync Across Devices:** Syncs bookmarks, history, and settings across devices when you sign in.

5. **Security Features:** Includes Safe Browsing to warn about dangerous sites and automatic updates for security.

6. **Performance:** Known for its speed and efficiency.

* **Importance of Security Assessments:**
  + Explain why conducting security assessments is crucial for this software.
* **Protect User Data**: Keeps your personal info safe.
* **Maintain Trust**: It ensures Chrome stays reliable.
* **Follow Rules**: Meets industry standards.
* **Prevent Breaches**: It Stops data leaks.
* **Improve Security**: Makes Chrome stronger.
* **Reduce Risks**: Limits potential attack points.

**2. Identified Vulnerabilities**

* **Vulnerability 1:**
  + **CVE ID:** CVE-2024-5274
  + **Description:** this is a flaw that allows attackers to execute arbitrary code within chrome and this was a Zero Day Exploit
  + **Impact:** Could have led to data breaches and unauthorised access

**• Vulnerability 2: the CVE-2024-4761**

**CVE ID: CVE-2024-4761**

* **Description:** An out-of-bounds write bug in Chrome’s V8 JavaScript and WebAssembly engine.
* **Impact:** Allows malicious actors to corrupt data, induce crashes, or execute arbitrary code on compromised systems. [Specifically, it enables a remote attacker to escape Chrome’s sandbox and perform unauthorised actions beyond the browser’s confines1](https://cyberfraudcentre.com/new-chrome-zero-day-vulnerability-cve-2024-4761).

**3. Exploitation Methods**

* **Vulnerability 1:** CVE-2024-5274

**Exploitation Method:** It showed that there was a critical security gap in Chrome’s codebase. The vulnerability was found in a part of Chrome that handles specific web requests, making it a prime target for attackers looking to compromise user data and browser functionality.

**Example Scenario:** **Exploitation of CVE-2024-5274**

An attacker creates a malicious website designed to exploit the CVE-2024-5274 vulnerability in Chrome. This website contains specially crafted web requests that trigger the vulnerability.

**Steps:**

1. **User Visits Malicious Website**: A user unknowingly visits the malicious website using Chrome.

2. **Triggering the Vulnerability**: The website sends a specially formatted web request to Chrome.

3. **Exploitation**: Due to the vulnerability, Chrome mishandles the request, allowing the attacker to execute arbitrary code on the user’s device.

4. **Impact**: The attacker can now-

o Steal sensitive information such as login credentials.

o Install malware on the user’s device.

o Compromise browser functionality, leading to further security breaches.

**Outcome:**

The user’s personal data and device security are compromised, potentially leading to identity theft, financial loss, and further malware infections.

* **Vulnerability 2:** **CVE-2024-4761**
* **Exploitation Method:** An attacker could exploit the bug by creating a crafted HTML page.
* **Example Scenario:   
  Target:** An e-commerce website using the vulnerable web application framework**.**
* **Attack Vector:** The file upload feature on the user profile page**.**
* **Steps to Exploit:**
* **Step 1:** The attacker registers an account on the e-commerce website.
* **Step 2:** The attacker navigates to the profile page and selects the option to upload a profile picture**.**
* **Step 3:** Instead of uploading a legitimate image file, the attacker uploads a specially crafted file containing malicious code.
* **Step 4:** Due to improper input validation, the web application framework processes the malicious file, allowing the attacker to execute arbitrary code on the server**.**
* **Impact:** The attacker gains unauthorized access to the server, potentially leading to data theft, server compromise, or further attacks on other systems**.**
* **This scenario highlights the importance of proper input validation and secure coding practices to prevent such vulnerabilities.**

**4. Mitigation Strategies**

* **Vulnerability 1: CVE-2024-5274**

Google has released updates to mitigate this vulnerability, emphasizing the need for users to update their browsers immediately to prevent exploitation.

* + **Mitigation Steps:**

1)**Update Your Browser**:

-Open your Google Chrome browser. -Click on the three vertical dots in the upper right corner -Go to **Help** > **About Google Chrome**. -Chrome will automatically check for updates and install them. -Click **Relaunch** in your browser to complete the update.

2)Enable Automatic Updates: Turn on automatic updates so you are up to date with Google’s latest updates

3) Use Enhanced protection: Go to **Settings** > **Privacy and security** > **Security**.

* + **Tools Required:**

Internet connection,Google chrome

**Vulnerability 2: CVE-2024-4761**

**Description**: This vulnerability affects a popular **Content Management System (CMS)** allowing attackers to perform SQL injection due to poor input sanitization in the search feature. Real life examples include attackers exploiting Joomla CMS using CVE-2023-23752

**Mitigation Steps:**

1. **Update the CMS:** Install the latest version to patch known vulnerabilities.

**2.** **Validate Inputs:** Ensure all user inputs are properly sanitized.

3. **Use Prepared Statements:** Implement parameterized queries to prevent SQL injection.

4. **Deploy WAF:** Use a Web Application Firewall to block malicious requests.

5. **Regular Audits:** Conduct frequent security audits and code reviews.

**Tools Required:**

* **CMS Update Package**: Latest update from the CMS provider.
* **Static Code Analysis Tools**: To identify insecure code.
* **WAF Solution**: Such as ModSecurity.
* **Database Security Tools**: For monitoring and securing database access.

**5. Conclusion**

* **Summary of Findings:**
  + Google suffered from a lot Zero Day Exploits
  + We have 2 specific examples
  + CVE-2024-5274 was a Zero Day exploit in which attackers would execute arbitrary code in order to cause a data breach
  + CVE-2024-4761 was a Zero Day exploit that allows malicious actors to corrupt data, induce crashes, or execute arbitrary code on compromised systems
* **Recommendations:**
  + Since they are Zero Day exploits there are no guaranteed ways to prevent it as they are unknown exploits. But there are some general security practices you can implement:

· **Patch Management:** Install software updates and security patches ASAP.

· **Application Whitelisting:** Only allow trusted programs to run on your device.

· **Security software:** Use antivirus and security tools to catch suspicious activity.

· **Browse carefully:** Be cautious of unknown websites and avoid suspicious links/downloads.

· **Least privelige:** Give users only the permissions they need (not full access).

These steps can help reduce your risk, even against unknown threats.

**Appendices**

* **Appendix A:** NA
* **Appendix B:**1) [Specifically, it enables a remote attacker to escape Chrome’s sandbox and perform unauthorized actions beyond the browser’s confines1](https://cyberfraudcentre.com/new-chrome-zero-day-vulnerability-cve-2024-4761). – www.cyberfraudcentre.com