International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

Extension Auditing: Privacy-Preserving Extension

Chetan Pathade[1], Paras Saxena[2], Aditya Sudhansu[3]

[1,2,3]*Information Networking Institute*
College of Engineering
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract - *In the current stage of the internet, browser extensions are everywhere, offering better functionality and user experience. However, this convenience often comes at the cost of the user's security and privacy. In the following research, we are tackling the world of browser extensions to audit their privacy criteria. It turns out, that many of these handy tools are not just adding functionality; they're also introducing significant security risks. Through a methodical and detailed approach that includes policy review, source code analysis, and comprehensive documentation, we tried to uncover the hidden threats posed by these extensions along with the claims by these extensions. Our methodology comprises installing extensions in isolated environments, utilizing tools like ExtAnalysis[1], Chrome DevTools[2], Sonarqube[12], and Bearer[10] for in-depth analysis, and testing for privacy policy offered by the extension and auditing it. This research not only seeks to answer critical questions regarding privacy violations but also the potential for privilege escalation. With the ultimate goal of enhancing digital safety, this paper highlights the imperative for improved security oversight in the development and deployment of browser extensions.*

Key Words: Extension, Privacy, Auditing, SonarQube[12], CRXcavator[16], Bearer [9], Extension Analysis, Policy.

1. INTRODUCTION

The growing number of browser extensions has unquestionably enhanced the online experience by providing users with a wide range of features easily accessible. Nevertheless, this expansion brings unique difficulties, mainly related to the safety and confidentiality of users. As the number of data breaches and privacy violations increases, the security of browser extensions has become a key topic in discussions about digital safety. This research starts by critically evaluating the security position of these extensions, with the goal of uncovering hidden weaknesses. We investigate browser extensions for potential threats by combining penetration testing, examining source code, and utilizing tools such as ExtAnalysis [1], Chrome DevTools, SonarQube[12], Bearer[9] and CRXcavator[14]. This study is driven by two main goals: emphasizing the urgent demand for improved security measures in the development and deployment of browser extensions and presenting a structured framework for evaluating these extensions. Our main objective is to promote a more secure digital environment where the benefits of browser extensions can be utilized without sacrificing user privacy and security.

2. Background

In the digital world, we observe exponential growth in browser extensions. They are transforming the way users browse the web by adding new and customizing features and offering us a personalized experience. From utility extensions such as ad blockers that keep ads away to productivity extension tools that autofill online forms, these add-ons have become a key part of the browser's user experience. They adapt to what different individuals require. Whereas, with all these extensions growing so fast, there's a downside too. We're facing a lot of security issues and weak spots that can disrupt our data privacy, cybersecurity, and even the economy.

Although their utility is great, there is a dark side to it as well, browser extensions to work properly require elevated privileges, access to sensitive user data (PII), and interacting closely with web pages(scripts). This inherent access and integration with web content make extensions gullible to exploitation by malicious actors seeking to compromise user privacy, conduct cyber attacks, and engage in illicit activities. As a result, the security stance of browser extensions has emerged as a critical concern, making necessary requirements for comprehensive assessments along with enhanced security measures, and robust oversight mechanisms.

3. Methodology

A. Scope:

We picked privacy-preserving extensions for our study because keeping data safe online is getting more important due to more frequent privacy issues. We chose extensions that are popular and downloaded a lot because they are important to lots of people. This helps us look into how well these tools are doing their job in protecting users' privacy.

Our selection of scope is as follows, from the broad spectrum of browser extension categories, our methodology was to strategically select privacy-preserving extensions for in-depth

analysis. This specific selection of extensions was based on criteria, extension's popularity, number of downloads, and presence in "recommended" / "popular" directories, akin to methods employed in precedent studies where extensions were chosen for their relevance and user engagement.
We selected UBlock Origin[5], NoScript[4], and Malwarebytes Browser Guard[6] for our study due to their unique approaches to privacy and security. UBlock Origin[5] is renowned for its efficient ad blocking and minimal resource usage, helping prevent tracking through ads. NoScript[4] offers robust protection by allowing JavaScript, Java, and other executable content to run only from trusted domains, directly combating cross-site scripting attacks. Malwarebytes Browser Guard[6] is chosen for its comprehensive threat-blocking capabilities, including malware, scams, and PUPs (Potentially Unwanted Programs), ensuring a safer browsing experience. These choices reflect a spectrum of privacy-preserving strategies, from content filtering to script control and direct threat mitigation.

B. *Source Code Review:*

Our way of looking at code is carefully made to understand how each privacy extension works in detail. We looked at many types of extensions, like ones that block ads or prevent the execution of various scripts on a web page, to see how they protect privacy in different situations. We studied how the code is organized, what other programs it needs to work, and the files it uses to get a full picture of how it's built. This was a manual process that we followed for the review of the permissions and capabilities given to the extension in the manifest.json file along with the flow of code during the execution of the JavaScript files like background.js and content.js. Furthermore, we conducted checks for various secure coding practices within the source code. This included examining the presence of potentially risky functions like eval() in JavaScript files, inspecting HTTP request headers in JavaScript, and evaluating other security-related aspects to ensure the integrity and safety of the code base.

C. *Expanding Attack Surface:*

We are conducting an audit of browser extensions using the tools ExtAnalysis[1], a crucial aspect to consider is how these extensions may contribute to expanding the attack surface of the browser environment. This expansion occurs due to various factors, including the extension's file system structure, URLs it interacts with, external IP addresses it connects to, and the permissions it requests. By examining these elements, we gain insights into how the extension operates within the browser ecosystem and potential avenues for exploitation by malicious actors.

One key aspect of expanding the attack surface is the extension's interaction with external resources. This interaction can include fetching data from external servers, loading content from third-party websites, or making network requests to external services. Each interaction introduces additional points of vulnerability, as it opens the door for potential attacks such as data exfiltration, injection of malicious code, or exploitation of vulnerabilities in external systems.

Moreover, analyzing the permissions requested by the extension provides valuable information about the level of access it has within the browser environment. Extensions that request extensive permissions beyond their intended functionality may inadvertently increase the attack surface by granting attackers broader capabilities if they manage to compromise the extension. Understanding these permissions and their implications is crucial for evaluating the security posture of the browser and mitigating potential risks associated with extension usage. Therefore, by comprehensively assessing the file system structure, URLs, external IP addresses, graphs, manifest.json file, and permissions of browser extensions, we can effectively identify and address vulnerabilities that contribute to expanding the attack surface, thus enhancing overall browser security.

4. Findings and Analysis

Assessment Methods: We adopted the extension testing methodology highlighted in the paper[9] and detailed in the blog post[11], Our comprehensive approach began with a secure code review, followed by automated testing using static code analysis tools such as Sonarqube[12], then with vulnerability assessment through Bearer[9], and CRXcavator[16] followed by policy review with EXTanalysis[1] and concluded the audit by manual testing of extension on different browser platform. Our consolidated observations are presented in Table 1. The results from automated analysis tools are depicted in the bar graph shown in Figure 2. Comparative security observations for the three extensions are illustrated in Figure 1.

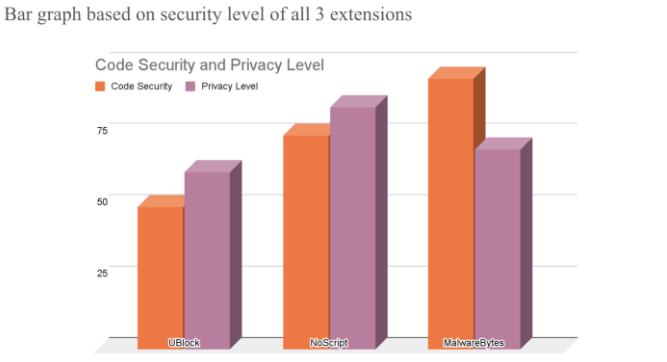

Fig.1 Security Level

A. *Common Observation:*

1. ***Code Review:*** For any injection attack to be successful, there is a need for a source and a sink in such a way that the source is user-controlled. To identify possibilities of successful injection attacks, we performed a reverse search for vulnerable sinks in the code and backtracked the results to identify vulnerable user-controlled sources. Some of the common sources that we identified were: eval(), constructor(), execCommand(), prompt(), confirm(), innerHTML(), document.write(). We went through the callback tree to figure out the source of the invocation of these vulnerable functions and it was observed that none of these functions were controlled by any user input. Also, we performed a search for secrets hard-coded in the code base for this we utilized regex for a general API key.

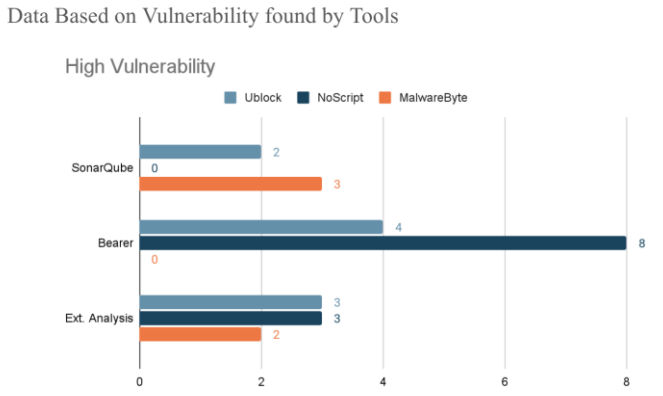

Fig.2 Vulnerabilities

2. ***Automated Tool Scanning:***
   a. *SonarQube[12]:* Sonarqube, is a freely available platform used for ongoing evaluation of code quality, enabling automated reviews through static code analysis. For the analysis of all the extensions (UBlock Origin[5], NoScript[4], Malwarebytes Browser Guard[6]), the extension was uploaded to the Sonarqube[12] web application and with the help of the sonar scanner, following which the automated analysis was conducted. Based on the results of the analysis, vulnerabilities related to the code base were detected which were later investigated to be found to be false positives as the detected functions were internal API calls made by functions and could not be controlled by the user.
   b. *CRXcavator[16]:* This is a tool that scans all the browser extensions and produces risk scores based on several factors such as permissions, vulnerable third-party libraries, weak CSP, and much more. In this tool, we can provide an extension ID or search for the extension to find the risk score. For all three extensions, we gave an extension ID to this tool to calculate the risk score. Based on our analysis of the risk score and the report, we didn't find any critical or high issues in all of the three extensions (UBlock Origin CRXcavator[13], NoScript CRXcavator[14], or Malwarebytes Browser Guard CRXcavator[15]).
   c. *Bearer[10]:* Bearer[9] is a comprehensive security tool designed to automatically detect and mitigate risks associated with API. For Ublock Bearer[9] dashboard indicated 4 high-level issues on further analysis of these issues it was discovered that 2 were associated with third-party libraries and the other 2 were false positives since the code base there was no source for the vulnerable sinks. The NoScript tool indicated 8 High-level issues. Similar observation to Ublock Origin[5]the vulnerable sinks are

not connected to any user-controlled source and thus all the alerts were false positive. In the case of Malwarebytes Browser Guard[6], the tool does not alert any high/critical issues.

3. ***Privacy Policy Review:*** In this section, we analyze the policies for all three extensions.

For UBlock Origin[5], in the privacy section of the settings there were three sections such as disabling pre-fetching (This is for preventing any connection for the blocked requests), disabling hyperlink auditing (This is for preventing informing one or more browsers about the link which is clicked and the timing) and block CSP reports (This is for preventing from the fingerprinting). When we analyze all these privacy options in UBlock Origin[5] then we find that all the policies followed what they are stating.

For the NoScript[4], it states that it will block all the JS files which is not trusted. So to analyze this privacy policy, we tried to visit several websites and we got to know that it actually blocks all the JS files which is not trusted.

Finally, for Malwarebytes Browser Guard, it states primary two things. The first one is related to blocking malicious sites. We tried to visit a malicious site to find out if the extension is able to detect that this is a malicious website and if is it blocking that site or not. We found that the extension detects and blocks malicious websites. Another one is blocking Ads/Trackers. Along with the malicious content, it also states that it blocks Ads and Trackers. As finding we found that it blocks all kinds of Ads and Trackers that are there on the website.

Along with this we also incorporate the ExtAnalysis[1] tool for analysing the permissions for the extensions. We were able to find some of the critical and high permissions that can lead to some privacy issues. With the help of these permissions, an attacker can perform some sort of attack based on the permissions. In the end, we analyzed that all the extensions have proper permission in place and that whatever results we got from the ExtAnalysis tool were false positives.

	Ublock Origin	NoScript	Malwarebytes Browser Guard
Common Observation	- All JavaScript files have been blocked. - No privacy breaches were detected. - The extension appears to be functioning correctly. - No critical or high-level vulnerabilities were identified.	- Blocking all the JS files - No privacy violation observed. - The extension seems to work as expected. - No critical/high-level vulnerability found.	- All JavaScript files are obstructed. - No invasion of privacy noted. - The extension operates as anticipated. - No major or severe vulnerabilities detected.
Chrome	It blocks CSP reports to mitigate fingerprinting attempts	Blocking all the JS files	Blocking Ads/trackers and blocking the site if there is any malware
Firefox	It blocks CSP reports to mitigate fingerprinting attempts	Blocking all the JS files	Blocking Ads/trackers and blocking the site if there is any malware

Table.1 Observations on Browser Extensions

B. Web Browser (Chrome and Firefox):

1. ***Ublock Origin [5]:***

A thorough examination was carried out to evaluate its security, permissions, and overall behavior. The aim was to identify any potential vulnerabilities, assess the extension's permission requirements, and monitor for any abnormal behaviors that could impact user privacy, security, or system performance. The findings of this audit are summarized below.

Key Feature:

- No Vulnerabilities Found: The audit did not uncover any security vulnerabilities within the uBlock Origin[5] extension. This suggests that the extension is well-maintained and developed with security best practices in mind, minimizing risks to users.
- Permission Usage: The examination of the extension's permissions revealed that all requested permissions are necessary for its intended functionality, such as blocking ads, filtering content, and enhancing privacy. There were no unnecessary or overly broad permissions that could suggest privacy invasions or other risks.
- Normal Behavior Observed: Throughout the duration of the audit, uBlock Origin[5] consistently exhibited normal behavior consistent with its design and purpose. There were no instances of abnormal behavior, such as unauthorized data transmission, unexpected alterations to web content beyond its ad-blocking

capabilities, or any actions that would degrade user experience or privacy.

- Performance Impact: The audit also paid special attention to the extension's impact on system performance. Findings confirm that uBlock Origin[5] remains one of the most resource-efficient ad blockers available, affirming its developer's claims regarding its lightweight nature and minimal impact on system resources

2. ***NoScript [4]:***
   - Appropriate Permission Use: Analysis of NoScript's[4] permissions revealed that the extension requests only what is strictly necessary to perform its core functions. This careful and minimal approach to permissions minimizes potential privacy risks and ensures that users retain control over their browsing experience.
   - Consistent and Expected Behavior: In the course of monitoring, NoScript[4] consistently behaved as intended, effectively blocking unauthorized scripts and allowing users to manually whitelist trustworthy sources. There were no instances of abnormal behavior, such as unapproved data transmission or interference with web page functionality beyond its security scope, which could compromise the user experience or privacy.
   - Impact on Performance: The audit also assessed NoScript's[4] impact on browser and system performance. Findings confirm that NoScript[4] is designed to be lightweight, with a negligible impact on browsing speed and system resources, emphasizing efficiency alongside security.

3. ***Malwarebytes Browser Guard [6]:***
   - No Security Vulnerabilities Found: The audit process did not uncover any security vulnerabilities within the Malwarebytes Browser Guard[6] extension. This result underscores the extension's strong security posture and the developers' commitment to safeguarding users from a wide array of web threats, including malware, phishing, and scams.
   - Permissions Are Justified: Upon reviewing the permissions requested by the Malwarebytes Browser Guard[6] extension, it was found that all permissions are justifiable and necessary for its operational objectives. The extension seeks permissions that align with its features, such as blocking malicious content and ensuring users' privacy, without overreaching or infringing on users' rights.
   - Behavior Aligns with Claims: Throughout the audit, Malwarebytes Browser Guard[6] exhibited behavior that aligns perfectly with its stated aims. It effectively blocked known malicious sites, ads, and trackers without any detected abnormal behavior that could potentially compromise the user experience or privacy. This consistency is indicative of the extension's reliability and effectiveness.
   - Minimal Impact on Performance: An important aspect of the audit was to assess the impact of Malwarebytes Browser Guard[6] on browser performance. The findings indicate that the extension has a minimal impact on browsing speed and system resources, which is a significant advantage for users seeking both protection and efficiency.

5. CONCLUSION AND KEY FINDINGS:

The audit study investigated the effectiveness of the privacy-preserving nature of the three popular privacy-preserving extensions -- Ublock Origin [5], NoScript [4], and Malwarebytes Browser Guard [6]. Our analysis results confirm that all three extensions are effectively blocking javascript, which is a common vector for web-based threats. Study also support that, none of the extensions were found to breach privacy or exhibit vulnerabilities at critical or high levels.

A. Key Findings:

a. **Privacy Preservation:** No privacy violations were found with any of the extensions in any of the tests. This observation demonstrates how consistently they uphold user privacy and guard against illegal data access or tracking.

b. **Functional Integrity:** Every extension performed well and as planned, showcasing reliable design and alignment with modern web technologies. Its operating efficiency is essential for long-term usefulness and end-user adoption.

c. **Vulnerability Assessment:** Every extension performed well and as planned,

showcasing reliable design and alignment with modern web technologies. Its operating efficiency is essential for long-term usefulness and end-user adoption.

d. **Cross-Browser Consistency:** When extensions were used with different browsers such as Firefox, and Chromium, the functionality of preventing advertisements and trackers and blocking JavaScript remained similar, protecting user security regardless of the browser platform.

e. **Adherence to Privacy Policy:** The extensions don't gather or utilize user data for purposes other than those required to operate, in complete compliance with their own privacy rules. An essential component of preserving integrity and confidence in the usage of browser extensions is preventing data from being shared with third parties without user agreement, which was not demonstrated to have occurred.

REFERENCES

[1] Tuhinshubhra, "GitHub - Tuhinshubhra/ExtAnalysis: Browser Extension Analysis Framework - Scan, Analyze Chrome, firefox and Brave extensions for vulnerabilities and intels," GitHub. https://github.com/Tuhinshubhra/ExtAnalysis

[2] "Chrome DevTools," Chrome for Developers. https://developer.chrome.com/docs/devtools

[3] "Burp Suite - Application Security Testing Software," PortSwigger. https://portswigger.net/burp

[4] "NoScript." https://chromewebstore.google.com/detail/noscript/doojmbjmlfjjnbmnoijecmcbfeoakpjm

[5] "UBlock Origin." https://chromewebstore.google.com/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm

[6] "Malwarebytes Browser Guard." https://chromewebstore.google.com/detail/malwarebytes-browser-guar/ihcjicgdanjaechkgeegckofjjedodee

[7] "CRX Viewer." https://robwu.nl/crxviewer/

[8] "PBL Report", https://help.passbolt.com/assets/files/PBL-02-report.pdf

[9] "Bearer | Developer-first SAST for security and privacy." https://www.bearer.com/

[10] "TestingMethodology", https://book.hacktricks.xyz/pentesting-web/browser-extension-pentesting-methodology

[11] "Cobal Introduction to Extension Testing", https://www.cobalt.io/blog/introduction-to-chrome-browser-extension-security-testing

[12] "SonarQube", https://www.sonarsource.com/products/sonarqube/

[13] "CRXcavator Ublock ", https://crxcavator.io/report/ihcjicgdanjaechkgeegckofjjedodee?platform=Chrome

[14] "CRXcavator NoScript", https://crxcavator.io/report/doojmbjmlfjjnbmnoijecmcbfeoakpjm?platform=Chrome

[15] "CRXcavator Malwarebytes Browser Guard", https://crxcavator.io/report/cjpalhdlnbpafiamejdnhcphjbkeiagm/1.57.0?platform=Chrome

[16] "crxcavator." https://crxcavator.io/