Assessing the Security and Privacy Implications of Ad Blockers in Web Browsers

Lanting Hou
Information Networking Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
lantingh@andrew.cmu.edu

Wenyi Qian
Information Networking Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
wenyiq@andrew.cmu.edu

Vicky Sun
Human-Computer Interaction
Carnegie Mellon University
Pittsburgh, Pennsylvania
vcsun@andrew.cmu.edu

***Abstract*—In the contemporary web landscape, browser extensions significantly enhance user experiences, with ad blockers being particularly prevalent due to their utility in filtering unwanted content. Despite their widespread adoption, these extensions often require broad permissions that, if misused or exploited, could pose significant privacy and security risks to users. This project endeavors to illuminate the potential hazards associated with ad blockers. Through a systematic assessment of various ad blockers, we aim to uncover and detail the vulnerabilities that may not be apparent to the average user. Our methodology includes an analysis of the permissions required by these extensions, an examination of their code bases for potential security flaws, and a review of the external lists they rely on for ad filtering. The goal is to foster a more informed public discourse on the balance between the benefits of ad blockers and the risks they entail. By identifying specific threats and presenting our findings, we hope to encourage the development of safer ad-blocking technologies and help users make more informed decisions regarding their web security.**

***Index Terms*—Browser Extensions, Permission Analysis, Privacy Risks, Ad Blockers**

I. Introduction

The proliferation of digital advertising has led to the widespread adoption of ad-blocking technologies, a development that, while enhancing user experience, raises significant concerns about security and privacy. Ad blockers, as browser extensions, have become essential tools for users seeking to navigate the web without the intrusion of unwanted advertisements [1]–[3]. However, the permissions required by these extensions to operate effectively can introduce vulnerabilities, potentially exposing users to risks that undermine their digital safety [4]–[7].

This project embarks on critically examining ad blockers to uncover and analyze the latent vulnerabilities that may be exploited maliciously. By scrutinizing the permissions, code injections, and reliance on third-party lists that characterize these extensions, our research seeks to illuminate the complex interplay between user convenience and security. Our inquiry is situated within a broader discourse on digital privacy and security, engaging with existing literature and contemporary studies to anchor our findings in the current academic and practical understanding of browser extension vulnerabilities.

In light of the evolving landscape of digital advertising and the continuous arms race between ad-blocking technologies and ad delivery mechanisms, our work contributes a timely perspective on the risks associated with ad blockers. By comprehensively assessing these tools, we aim to identify specific vulnerabilities and foster a more nuanced understanding of the trade-offs involved in using ad blockers. This understanding is crucial for informing users, developers, and policymakers about ad-blocking technologies' implications and advancing strategies that safeguard user privacy and security without compromising the integrity of the web's economic models.

II. Related Work

Adblocker Extension is a browser extension that can block advertising content on web pages to increase browsing speed, improve user experience, and enhance personal privacy protection. These extensions are installed into the user's web browser and automatically run when the user visits a web page [8]. Ad-blocking extensions use complex lists of rules and filters to detect and block ads. These rules identify specific ad servers, ad tags, or network elements. When the browser attempts to load these elements, ad-blocking extensions step in and prevent this content from being loaded, preventing ads from being displayed [9]. Common filter lists like EasyList contain extensive information about known ad sources and tracking scripts [10].

Ad blockers mainly improve web page loading speed and reduce data usage by reducing ads on the page, thereby optimizing the user's web browsing experience [11]. They also enhance user privacy and reduce user behavior monitoring by blocking trackers and ad networks [7].

Although ad blockers offer many benefits such as enhanced browsing experience and improved privacy, research has also revealed their flaws and security risks. For instance, the widely used filter list EasyList has suffered severe traffic overload due to frequent access by certain Android browsing apps in India, leading to prolonged download times and triggering CloudFlare's DDoS mitigation protocols, which in turn affects the functionality of ad blockers that rely on EasyList [12]. Additionally, some ad blockers may mistakenly block non-ad content, affecting website functionality [13]; others might serve as potential carriers for malware, especially if the ad blocker extensions are from unknown sources. This includes vulnerabilities like filter list manipulation that may allow

malicious ads through and risks introduced by ad blockers, such as code injection or unauthorized data access. One notable instance reported by Lee Mathews involves a dangerous flaw in popular ad blockers like Adblock Plus, AdBlock, and uBlock, which put over 100 million users at risk by allowing cybercriminals to push malicious code through these extensions [14]. Furthermore, some ad blockers collect user data to improve their filtering algorithms, raising concerns about how this data is used and stored, especially when it is monetized, which could pose privacy risks if mishandled [15].

III. Threat Model

Our threat model considers a remote attacker exploiting the permission or vulnerabilities in adblock extensions within a victim's browser. Such exploitation can happen in multiple scenarios:

A. Browser Extension Permissions Exploitation

Ad blockers generally require extensive permissions in order to be able to block advertisements, which can be exploited by the developer. These include reading and modifying all data on the websites a user visits and other critical user privacy, which can lead to significant security breaches if misused. An attack scenario here could involve an ad blocker that has been modified to track user behavior across all websites under the guise of blocking ads, effectively functioning as spyware.

B. Insider Threats

The risk of insider threats looms large where employees or developers could potentially insert malicious code into the ad blocker if they are sufficiently motivated, possibly by financial gain. The impact is heightened by the vast number of users reliant on these extensions. As of the second quarter of 2023, it was estimated that there were approximately 912 million users of ad-blocking extension globally, and the number showed an increasing trend [16], illustrating the potential reach of such an attack. An example might be an insider who inserts a malicious script that redirects all ad requests to a server controlled by the attacker, capturing user data in the process.

C. Third-Party Filter Vulnerabilities

Ad blockers often allow users to subscribe to third-party filters, which are usually maintained independently of the ad blocker developers. These third-party lists can be a vector for attacks if compromised. Since the developers typically do not govern these lists, a convincing but malicious filter could be widely adopted. For instance, in the release of Adblock Plus 3.2 on July 17, 2018, a new feature was introduced: the rewrite filter action [17]. This feature enabled users to redirect requests to a new URL, with the restriction that the destination must share the same origin as the original request. Despite this precaution, the introduction of this feature inadvertently opened up new security vulnerabilities. These vulnerabilities could potentially lead to arbitrary code execution on user systems if they applied malicious filter rules to their extension [18]. It might also redirect the victim to phishing sites whenever they attempt to access common services like email or banking, capturing sensitive information.

IV. Methodology

Our method of selecting and analyzing ad blockers involves conducting a preliminary scan with CRXcavator [19] and then performing code analysis on extensions with high-risk vulnerabilities with SonarQube [20]. We adopt a systematic and rigorous approach to ensure our results are comprehensive and reliable.

First, we employed CRXcavator, a powerful tool designed to review Chrome extensions. Using CRXcavator's analysis of adblockers, we conducted an in-depth analysis of each extension's permissions model, audited the access levels requested by ad blockers, and identified permissions that could pose potential risks to security or privacy. In cases where CRXcavator identified a JavaScript vulnerability in an extension, we used SonarQube for further code scanning. SonarQube is a useful static analysis tool that allows us to pinpoint ad blocker code modules vulnerable to potential injection risks. This step is critical for isolating specific areas of the code that require deeper inspection.

We conducted additional experimental testing and exploitation trials for modules identified as potential risks. These tests are designed to verify the actual impact of vulnerabilities and confirm whether they can be exploited in real-world scenarios. Our criteria for assessing risks and vulnerabilities include the severity of potential exploitation, likelihood of occurrence, and impact on user privacy and data security.

By combining the analytical capabilities of CRXcavator and SonarQube with targeted experiments, our approach identifies theoretical risks and assesses the exploitability of vulnerabilities in ad blockers. This comprehensive approach allows us to provide users and developers with substantive advice about the security and privacy of ad-blocking extensions.

In addition to conducting scans for vulnerabilities, we delved into the intricacies of understanding how ad blockers are implemented and how they interact with people and web content. Our approach involved analyzing the current landscape and researching past events to inform our understanding of the potential threats. By evaluating these interactions, we aimed to gain a comprehensive grasp of ad blockers' effectiveness in enhancing user privacy and browsing experiences, as well as the challenges posed by websites that attempt to circumvent ad blocking measures for their benefits.

A. Ethical Considerations

In our research, identifying vulnerabilities demands a methodical approach to disclosure that emphasizes user safety and system security. Upon discovering vulnerabilities in browser extensions or underlying design flaws, we commit to a responsible disclosure protocol. This protocol is guided by the best practices established by the IEEE Symposium on Security and Privacy [21], which recommend allowing a 45 to 90-day window for vendors and other stakeholders to rectify

the vulnerabilities prior to public disclosure. This period is crucial for preventing potential harm to users and ensuring that corrective measures are effectively implemented.

Furthermore, to safeguard external parties from unintended consequences, all our experiments are strictly confined to the internal team using our own testing devices within a controlled development environment. This ensures that our simulations of potential exploits do not impact any external users or systems.

V. Analysis of Ad Blockers

A. Heightened Permissions

In our comprehensive analysis of eight popular ad blockers—AdBlock, Adblock Plus, AdGuard AdBlocker, Adblocker Ultimate, uBlock Origin, AdLock, Adblocker for YouTube™, and Ghostery [22]–[29], —we meticulously examined the permissions each ad blocker requires to function effectively within a user's browser. In Figure 1, areas marked in red by CRXcavator are labeled as critical risk, orange as high risk, and yellow as medium risk. Ad blockers generally require extensive permissions like access to all URLs, network request interception, and cookie management, which are essential for functionality [30]. Our further analysis emphasizes the breadth and depth of permissions these extensions require, underscoring the potential impacts on user privacy and security.

1. Universal Access to All URLs(<all_urls>): Almost all ad blockers, except for Ghostery, request access to all URLs. This permission is crucial as it allows ad blockers to filter out ads on any website a user visits. While essential for functionality, it grants extensive access to users' browsing data, posing privacy concerns.

2. Network Request Interception: Every ad blocker in our study utilizes the webRequest API, enabling them to modify or block network requests. This capability is vital for stopping unwanted ads but also brings risks, as it could be used to alter or intercept sensitive information transmitted over the network.

3. Cookies: AdGuard, AdBlocker, and Ghostery have requested permission to manage website cookies. These cookies can be used to remove tracking cookies or store ad preferences, thus enhancing user privacy. However, if mishandled or leaked, this data could become a conduit for privacy breaches.

4. Tabs: Ad blockers also commonly grant permission to access browser tabs. Tab access is necessary for managing pop-up ads and redirects. However, if not properly safeguarded, these permissions could also be used to track user activity across sites.

5. Enhanced Privacy and Content Settings: Most ad blockers request permissions that affect privacy and content settings, allowing them to enforce privacy-enhancing rules against tracking scripts and cookies. While these permissions support privacy protection, they also allow ad blockers significant control over how content is displayed and user data is managed.

6. Storage and Web Navigation: All eight ad blockers we investigated have these two permissions. Permissions for storage and web navigation are crucial for maintaining user settings and managing updates to filter lists. These permissions support the storage of configuration data and effectively manage the blocking of dynamic content during online navigation.

Ad blocker ---------- - Permission	AdBlock	Adblock Plus	AdGuard AdBlocker	Adblocker Ultimate	uBlock Origin	AdLock	Adblocker for YouTube™	Ghostery
<all_urls>	v	v	v	v	v	v	v	
webRequest	v	v	v	v	v	v	v	v
cookies			v					v
tabs	v	v	v	v	v	v	v	v
privacy			v		v			
contentSettings		v						
storage	v	v	v	v	v	v	v	v
webNavigation	v	v	v	v	v	v	v	v
management	v	v						

Fig. 1. Permissions for top ad blockers on Chrome Store

7. Management of Other Extensions (Management): Only a few ad blockers, such as Adblock Plus, request access to the **chrome.management** API, which allows them to interact with or manage other extensions. This can be beneficial for resolving conflicts between extensions but introduces potential security vulnerabilities if the ad blocker or other managed extensions are compromised.

B. Untrusted Inputs

We explored two primary areas of vulnerability related to untrusted inputs in ad blockers: 1) potential **Code Injection** and **Denial of Service** in source code functions, and 2) potential **Arbitrary Code Execution** through manipulation of custom filter lists. We utilized the tool SonarQube to deliver surface level findings of potential vulnerabilities in the code, which discovered multiple locations that could be susceptible to Code Injection and Denial of Service (caused by Regular Expression), among which included using the URL or user inputs in the extension for further filtering. To deepen our investigation, we conducted a manual inspection of these risky functions, examining the specific contexts in which they operate and the nature of the risks they pose.

1) From Source Code: The scanning results of SonarQube on selected adblocker extensions warned of certain Code Injection vulnerabilities. For example, the function `constructJavascriptFunction(data)` detailed in Listing (1) from our initial static code analysis on AdGuard Adblocker illustrates as a case where dynamic code execution could potentially take place. This function constructs JavaScript functions dynamically from strings, which can be manipulated to inject malicious code if the input validation is inadequate.

In this example, if the `data` parameter is not properly sanitized, it could lead to arbitrary JavaScript execution. For instance, an attacker could pass a string that closes the function and adds additional malicious scripts. This type of vulnerability underscores the need for stringent input validation and the use of safer alternatives to `new Function()`.

2) From Custom Filters: Ad blockers operate by employing a set of rules known as filter lists to determine which content should be blocked or hidden during web browsing (blacklisting). These filters are crucial in managing the visibility

```
function constructJavascriptFunction(data) {
  /*...*/

  if (ast.body[0].expression.body.type === 'BlockStatement') {
    /*eslint-disable no-new-func*/
    return new Function(params, source.slice(body[0] + 1, body[1] - 1));
  } // ES6 arrow functions can omit the BlockStatement. In that case, just return
  // the body.

  /*eslint-disable no-new-func*/

  return new Function(params, 'return ' + source.slice(body[0], body[1]));
}
```

Listing 1: Code with vulnerability in redirect.js

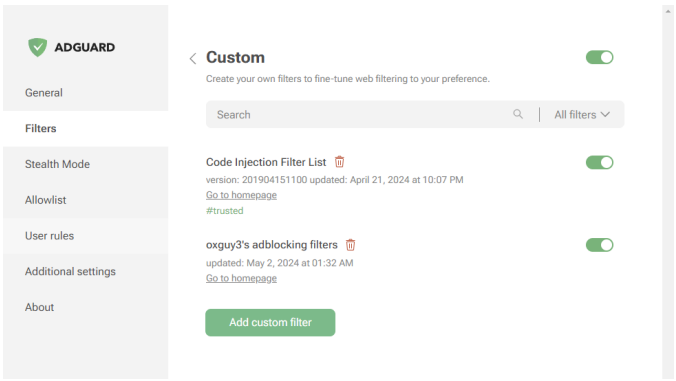

Fig. 2. Custom Filters in Adguard Adblocker, supporting a "#trusted" option

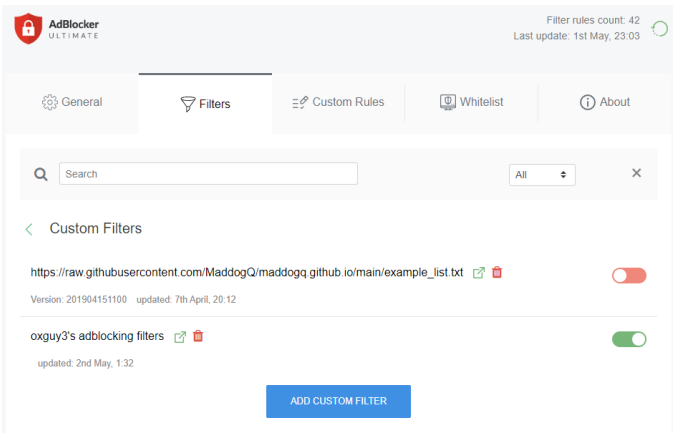

Fig. 3. Custom Filters in Adblocker Ultimate

of ads and potentially malicious content on websites. Filter lists are typically organized into collections maintained by volunteers and communities, which could inadvertently expand the attack surface due to varying levels of oversight and security awareness among the maintainers [31].

Furthermore, most ad blockers allow the importation of custom filters to add specific filtering rules, as shown in Figure (2) and Figure (3). These user-generated filters specify URLs or URL patterns and designate actions like blocking or allowing these links. This feature is designed to enhance user flexibility by allowing them to block additional content as per their preferences, which is not included in the standard filters. However, it also further increases the attack surface, considering it adds complexity by involving third-party elements in ad blocking, which could increase the potential for security vulnerabilities. The syntax used in these filters can be complex and not readily understandable to most users, allowing attackers to disguise malicious actions within filters that appear to offer additional functionalities.

Given the critical role of these filters, we conducted a thorough security analysis using the Chromium browser [32], specifically examining how AdGuard AdBlocker (version 4.3.46) [24] and AdBlocker Ultimate (version 3.8.21) [25] manage and apply these lists. The version of Chromium used for this analysis was 124.0.6367.61, chosen for its contemporary relevance and support for modern web technologies.

Our investigation revealed potential vulnerabilities, particularly in how filter lists process untrusted inputs. This can occur when filter rules, designed to intercept or modify HTTP requests or responses, are manipulated to inject or execute arbitrary code. The security implications are significant, as malicious entities could craft specially designed filter rules that execute arbitrary code when processed by an ad blocker, leading to unauthorized actions within the user's browser environment.

CVE Analysis: We reference specific vulnerabilities such as CVE-2019-11593 (Adblock Plus before 3.5.2), CVE-2019-11594 (AdBlock before 3.45.0), and CVE-2019-11595 (uBlock before 0.9.5.15) [33]–[35], which illustrate past exploits where ad blockers were manipulated through the $rewrite option in their filter lists. These CVEs demonstrate scenarios where malicious filter rules led to arbitrary code execution, by injecting a malicious script within filter rules, which was then executed by the ad blocker, bypassing the expected security model.

C. Reliance on Existing Lists

Ad blockers use filter lists to determine which sources needs to be blocked on a site, including frames, images and other elements. These lists often contain numerous segments of URLs that are periodically updated. EasyList, a community-

run filter list, is one of the most prominent filter lists used in the ad blocking industry [10]. It has an extensive collection of ad sources from up to 19 different languages and regions maintained by multiple communities.

Among the top 6 Chrome extensions we investigated, we discovered all of them incorporate EasyList. While the majority of extensions allow users to adjust which list to use or input custom filters, EasyList remains the default filter for these extensions. Many other smaller extensions also like to adopt this list, as it's one of the most wide-spread open source filters.

Any issues with EasyList could severely impact users with ad blockers installed. It is reported that over 30% of the population in countries with internet accessibility, such as the US, China, Germany and Egypt etc. have ad blockers installed on their browsers [36]. Hence, it is imperative we understand the risks that ad blockers face aside from their code bases. We hypothesized characteristics such as being installed by a large portion of the population, or being run by a small community, may come with issues such as maintainers going rogue to exploit the list for self-interest, or that updates may be slow or incomplete to match the fast pace of the web environment. As we did not have access to the lists themselves, we opted to look into past cases to validate the basis of our hypothesis, and theorized on other potential threats.

D. Violation of Privacy

Ad blockers pose a significant privacy risk if not managed properly due to the broad permissions required to operate. Permission cookies such as <all_urls>, webRequest, and allow ad blockers to access virtually all data on websites visited by users, intercept and modify web requests, and manage cookies.This broad level of access is necessary for effective blocking of ads and tracking scripts; however, it also means that these tools can potentially collect sensitive information, including browsing history, shopping habits, and even login data. Without strict data handling and privacy policies, this data may be used inappropriately by the ad blocker providers themselves, or it may be exposed to third parties through data breaches or misconfiguration [7]. Users must be vigilant and choose ad blockers from reputable providers that transparently disclose how they handle and protect user data.

E. Exploiting User Behavior

In the modern era, users are used to consuming free content on the internet, and many websites turn to advertisements as an additional source of income, and some may even have to rely on ads to keep their sites running. However, the introduction of ad blockers prevents sites from acquiring their needed funding or revenue. In response to this, websites started launching counter ad-blocking strategies, most notably the Wall strategy and enrolling in the acceptable ads exchange [37]. The former includes using a popup to request the web visitor to disable their ad blockers, while the latter is a program where ad blockers white list ads that are considered non intrusive to the browsing experience, which is often the cause of users installing ad blockers [36]. We assessed how these methods are implemented, and researched past events to solidify our understanding.

VI. Findings

A. Permission risk

In the world of ad blocking extensions, the scope and meaning of the requested permissions are critical to understanding the trade-off between functionality and user privacy/security. Our analysis covers several popular ad blockers, specifically AdGuard AdBlocker, Adblock Plus, Video Adblocker for YouTube™ Extension, and AdBlocker Ultimate. Each of these tools has unique permission requirements that affect its operation and potential privacy risks.

AdGuard AdBlocker and Adblock Plus Both AdGuard AdBlocker and Adblock Plus require granting full access to all URLs and usage of the "chrome.webRequest" API. This allows these extensions to modify network requests, which is critical to the core functionality of blocking or changing web content before it is loaded. However, these permissions also allow the extension to read and potentially modify any data sent to or received from any website, posing significant privacy risks.

Additionally, **Adblock Plus** requires access to the "chrome.management" API, which allows it to manage or interact with other extensions and applications installed on the user's browser. While this feature provides functional benefits such as compatibility checking and coordinated behavior between extensions, it also introduces moderate risks. It may be used to change the behavior of other extensions or exploit combined permissions in a way that may not be transparent to the user.

Video ad blocker for YouTube™ extension and AdBlocker Ultimate The Video Adblocker for YouTube™ extension is designed to enhance the user experience on YouTube by blocking ads, but it requires permissions that may severely impact user privacy. It requires access to comprehensive data across websites, including observing user browsing behavior and tracking navigation. While necessary for extensions to function effectively on video content, without strong protections, these permissions can be abused to collect sensitive user data.

AdBlocker Ultimate, on the other hand, asks for permission that allows it to freely access all URLs and monitor browser tabs. It can also edit and track web traffic, which is essential for blocking ads but can cause privacy issues. The extension's ability to store and monitor user data adds another layer of risk. However, it specifically does not request permissions for context menus and unlimited storage, limiting its data accumulation potential.

Enhancement of user functions Some permissions, such as those enabled by AdGuard AdBlocker to manage cookies, are intended to enhance user functionality. This includes allowing users to manage the expiration dates of first- and third-party cookies, which can be seen as a positive feature from a privacy perspective.

The broad permissions required by these ad blockers highlight significant privacy concerns. While these permissions are critical to the extension's functionality, they can also open avenues for potential privacy violations if not managed carefully. Users should understand these permissions and manage them wisely to protect their privacy while enjoying the benefits of ad blocking. Developers of these tools must work hard to balance functionality with privacy, ensuring that only necessary permissions are requested to minimize potential risks.

B. Static Code Analysis

From the scanning results obtained via SonarQube, we identified primarily two function types susceptible to code injection vulnerabilities: dynamic code execution and variations of eval(). Upon closer scrutiny, we determined that the security risks associated with functions capable of code injection, such as the functions in (1) and in (2), are mitigated by their restricted access within the extension's architecture. Such functions are designed to be invoked only through secure, internal function calls, thereby limiting their exposure to external threats. Consequently, for an attacker to exploit this function, they would need to bypass the extension's inherent security barriers and gain elevated privileges within the extension itself. This scenario implies a significant breach of the extension's security protocols, suggesting that the function's security, while robust under normal operating conditions, depends critically on the overall security posture of the extension (known as the Bucket Effect). To strengthen this posture, we recommend implementing additional layers of security checks at the points where internal functions are called, ensuring that even if the extension's primary defenses are compromised, unauthorized code execution is prevented.

Another prevalent form of Denial of Service vulnerability in ad blockers is often triggered by regular expressions used within blocking rules to efficiently filter out ads. These regular expressions can become overly complex and inadvertently cause performance issues due to its backtracking algorithm [38]. During our comprehensive review of all potential hotspots, we found no instances of problematic or 'evil' regular expressions that could lead to such vulnerabilities.

C. Custom Filter List Exploit

We attempted to recreate the exploit in our chosen ad blockers, referencing the arbitrary code execution vulnerabilities reported in older versions of some well-known ad blockers [18]. Our approach involved importing a custom filter list which contained the exploit as detailed in Listing (3). The page containing the payload was modified to our controlled web page. The specific rule would redirect requests on the Google Maps page to Google's "I'm Feeling Lucky" search service, which then redirect to a page designed to execute "alert(document.domain)". As we anticipated, all ad blockers we tested have effectively incorporated preventative measures based on past vulnerabilities; consequently, none of the exploits were successful. The inability of these exploits to succeed provided additional insights into the robustness and efficacy of the ad blockers' filter rules, demonstrating their enhanced security measures.

Sebastian, the researcher who initially discovered the vulnerability, suggested completely eliminating the use of the $rewrite filter as a definitive solution, citing its potential for misuse under various circumstances [18]. In response, Adblock Plus implemented a fix to confine the $rewrite option's redirections solely to internal resources. These resources are static and embedded directly within the extension's source code, significantly reducing the risk of abuse while addressing the security concerns previously identified with this feature. For example, the directive "$rewrite=abp-resource:blank-js" redirects to an empty JavaScript file. Additionally, other internal resources are made available, such as an empty text file ("abp-resource:blank-text"), a blank CSS file ("abp-resource:blank-css"), and an empty HTML document ("abp-resource:blank-html") for convenience [39]. The $rewrite rule continues to be operational but is significantly more restricted in its application to prevent abusing. This strategic adjustment not only mitigates the security risks previously associated with the rewrite feature but also prevents errors when blocking URLs, thereby improving the extension's overall safety and effectiveness.

D. Community Filter List Exploits

As community-run filter lists, they encounter similar security issues open source projects may face. In particular, the wide-spread adoption of EasyList can influence many, if not most, of the ad blockers available.

1) Lack of oversight in filter list maintenance: As a community-run project, filters may lack centralized oversight, leading to potential issues about the validity and accuracy of updated URL segments. The lack of such oversight can lead to mistakes or false positives going unchecked, inadvertently obstructing users' access to normal web pages. While we are led to believe such occurrences are often accidental mistakes, there have been instances where it was the result of a deliberate malicious manipulation.

A notable case involved a Finnish addition to Easylist, where union websites were intentionally added to the list by one of the maintainers to make a political statement regarding ongoing union strikes against the government [40]. Fortunately, users were able to quickly discover this and request a fix. However, during the period before the issue was rectified, users experienced a denial of service of these websites, highlighting the potential impact of unchecked updates to filter lists.

Beyond political motives, we theorize there exists the potential for malicious exploitation of filter lists for various other reasons:

- **Denial of other services:** Similar to the case of the Finnish addition of EasyList, maintainers with malicious intent could exploit the lack of oversight in filter list maintenance to selectively block access to specific services or web pages by adding these sources to the

```
...
if (navigator.userAgent.match(/msie/i)) {
  iframe.src =
    "javascript:'<script>window.onload=function(){document.write(\\'<script>document.domain=
    ↪  \\\"".concat(
      document.domain,
      "\\\";<\\\\/script>\\');document.close();};</script>'"
    );
...
}
```

Listing 2: Example form of `eval()`, "`javascript:`"

```
[Adblock Plus 3.2]
  ! Version: 201904151100
  ! Title: Code Injection Filter List
  ! Last modified: 15 Apr 2019 11:00 UTC
  ! Expires: 1 hours (update frequency)
  /^https://www.google.com/maps/_/js/k=.*/m=pw/.*/rs=.*/$rewrite=/search?hl=en-US&source=
  ↪  hp&biw=&bih=&q=maddogq.github.io&btnI=I%27m+Feeling+Lucky&gbv=1
```

Listing 3: Example filter rule exploiting the $rewrite filter option

list. This could be motivated by personal or competitive interests, or ideological biases, resulting in the denial of access to specific online resources.

- **Allowing of malicious ads:** On top of hostile additions to the filter lists, malicious actors may also selectively remove sources from the list to permit harmful advertisements to reach users' screens. These ads could be disguised in less conspicuous forms, delaying the time users detect and report these to the ad blockers or other maintainers. While users are also subject to these risks without the installation of ad blocking extensions, users' trust with the integrity of ad blockers could decrease their awareness and alertness about the types of content they come across.
- **Profiting from affiliate ads:** Unscrupulous individuals may also exploit filter lists by blocking all ads but theirs, enabling them to profit while others cannot.

2) Lack of frequent updates: Filter lists require a lot of manpower to maintain it with up-to-date content. Our investigation of EasyList revealed that the English list is able to receive frequent updates on the filter [41]. However, this level of maintenance may not be consistent across all the languages under EasyList. For instance, the Dutch filter for EasyList, EasyDutch, as of this time of writing, has not been updated in the past 4 months [42]. This prolonged period without updates raises concerns regarding the integrity of the filter list.

In consideration regarding the exploits from malicious filter maintainers, the low frequency of maintenance could also entail that in less active communities, these exploits could go unchecked for longer periods, harming regular users and the browsing environment.

E. Behavioral Exploits

As ad blockers impede web owners to profit from advertisements, strategies such as the Wall strategy and acceptable ads started to get implemented. We explored the possible issues

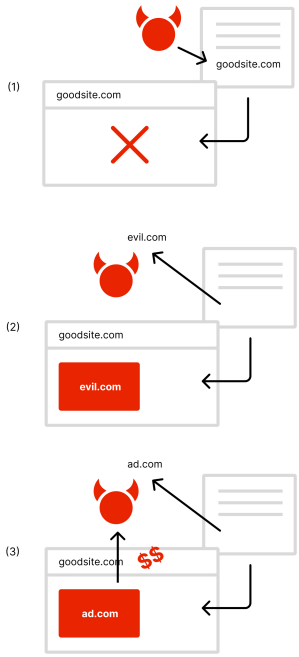

Fig. 4. Malicious use of ad filters. (1) Denial of other services, (2) Allowing of malicious ads, (3) Profiting from affiliate ads

users might encounter when they come across publishers that adopt these strategies.

1) The Wall strategy: When a website implements the Wall strategy, the site shows a popup requesting the user to turn off or pause the ad-blocker once an ad blocker is detected. When a user complies, their actions whitelists the publisher's website. This is mostly implemented in two ways: hard block and soft block [43]. The first method to completely deny access to content for visitors with ad blockers, forcing them to whitelist

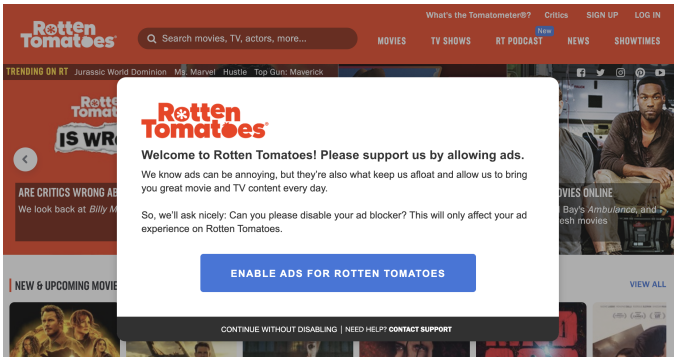

Fig. 5. Rotten Tomatoes' request to disable ad blockers [44]

Fig. 6. Acceptable Ads aren't all considered acceptable [49]

the site. The second method is less forceful, but utilizes design methods to persuade or guide the visitor to think they have to disable the ad blocker, such as placing the option to dismiss the request at the bottom with a small font size, rendering it unnoticeable to the majority of the users. In addition, they use terms such as "support" or "care" to guilt-trip users [44]. Under these influences, users can feel compelled to oblige. By obliging to the request, however, users are opened up to potential malicious ads that were previously blocked by the ad blocker.

In 2016, users who landed on Forbes' website were asked to turn off their ad blockers to gain access to an article they were eager to read, which eventually resulted in the visitors being attacked by malicious pop-up ads that aimed to install malware on their systems [45]. This highlights an issue where publishers are putting their visitors at risk when asking them to disable ad blockers, especially when they cannot control the source of the ads. In addition, malicious affiliated publishers can also use the same tactic to persuade users to whitelist their websites to display malvertising, exploiting human behaviors.

However, we also have discovered some major websites are moving away from utilizing this strategy to profit, and are now adopting a subscription wall instead. We discovered a list of some of the top 10 websites that implement the Wall strategy, including websites like GQ, Telegraph and Forbes etc. in 2016 [46]. Among these websites, 2 of which has shut down since, and the remaining 8 have abandoned the Wall strategy for a subscription model, where the popup to request users to disable ad blockers are replaced with a popup or overlay letting users know they need to subscribe to view the content. This entails that users were overall dissatisfied with the experience of these requests, and may have limited effects for web owners to profit off from this, or that the security risks have prompted publishers to opt for a safer method to gain revenue. Nevertheless, the Wall strategy is still being implemented across the web, and users should be aware of the implications of disabling their ad blockers upon the request of web sites.

2) Acceptable Ads: Acceptable Ads was introduced as a whitelisting mechanism for ad blockers to allow non-intrusive ads to be displayed on websites through specifying acceptable positioning, sizing and distinction from the main content [47]. Out of 6 of the Chrome extensions we reviewed, 3 of them incorporate Acceptable Ads. Although extensions allow users to toggle on and off whether to enable this feature, they're set to on by default. According to research, 95% of users stick to the default settings, and only 5% will make any configurations [48]. This means that 95% of people using these ad blockers will unknowingly always allow Acceptable Ads.

The main issue lies in, despite Acceptable Ads' good intentions, what is considered "acceptable" is controversial. Some users argue that there are no ads that are acceptable [49]. For example, an ad was found to be marked as acceptable despite it's clearly designed as a click bait. This raises the concern where a scam advertisement might still be deemed acceptable if they meet the Acceptable Ads criteria.

While these may not be immediate threats to people brought on by the ad blockers themselves, users need to be aware that ad blocking extensions do not make them immune to malvertising, and should be alert of the possibility of being exposed to malicious ads.

VII. Recommendations

For users, maintaining a secure and private browsing experience is crucial, and choosing the right ad-blocking extensions plays a significant part in this process. It is essential to stay informed about the latest updates and vulnerabilities associated with ad blockers. Users should select reputable ad-blocking extensions that are widely trusted, have a proven track record, and are scrutinized by security professionals. Managing the permissions of these extensions is vital—only necessary permissions should be allowed to minimize potential security risks. Keeping the ad-blocking software up-to-date ensures users benefit from the latest security patches and improvements. Additionally, users should exercise caution with free ad-blocking extensions, which might pose hidden security or privacy risks, and consider supporting trusted extension developers instead [50].

For developers, it is imperative to adhere to best practices in designing and maintaining ad-blocking extensions to ensure user security and trust. Developers should minimize

the permissions requested by their extensions to reduce the attack surface [51]. Regular security audits can help identify and mitigate the extension's code base vulnerabilities. Implementing secure coding practices, including rigorous input validation and proper error handling, is essential. Educating users about the security aspects of ad blockers and encouraging safe browsing practices can enhance their overall security posture. Transparency in data collection and operational processes helps build trust with users. Collaborating with security researchers and the broader cybersecurity community can facilitate the early detection of security issues and foster ongoing innovation and improvement in ad-blocking technology.

VIII. Conclusion

In conclusion, our investigation into ad blockers' security and privacy implications presents a complex landscape where benefits and risks coexist. Ad blockers provide users with a streamlined online experience by filtering out unwanted content, yet they pose potential threats to privacy and security due to the extensive permissions they require.

Our research identified several critical findings regarding the risks associated with ad blockers. Specifically, we highlighted the potential for malicious actors to exploit permissions granted to ad blockers, leading to code injection and other security breaches. These potential vulnerabilities emphasize the importance of carefully examining the functionality required by ad-blocking extensions and the need for robust security measures.

Reflecting on the broader implications, we recognize the delicate balance between user convenience and the potential threats posed by ad blockers. While these tools enable users to control their online experience, they also disrupt the traditional web ecosystem, affecting websites that rely on advertising revenue. Moreover, ethical considerations surrounding the development and usage of ad blockers, such as acceptable ads, prompt us to question the responsibilities of developers in safeguarding user data and users' decision-making process when enabling such extensions.

Future efforts should focus on developing secure ad-blocking technologies and understanding user behavior to inform better design and policy decisions. Additionally, studying user behavior concerning ad blocker usage can provide insights into user preferences and concerns.

Finally, we urge developers, users, and vendors to engage in ongoing dialogue and collaboration to ensure that ad blockers evolve in ways that prioritize functionality and user safety. By addressing these challenges collectively, we can foster a more secure and privacy-respecting online environment for all stakeholders.

References

[1] R. Nithyanand, S. Khattak, M. Javed, N. Vallina-Rodriguez, M. Falahrastegar, J. E. Powles, E. De Cristofaro, H. Haddadi, and S. J. Murdoch, "Adblocking and counter blocking: A slice of the arms race," in *6th USENIX Workshop on Free and Open Communications on the Internet (FOCI 16)*, 2016.

[2] E. Pujol, O. Hohlfeld, and A. Feldmann, "Annoyed users: Ads and ad-block usage in the wild," in *Proceedings of the 2015 Internet Measurement Conference*, 2015, pp. 93–106.

[3] M. Malloy, M. McNamara, A. Cahn, and P. Barford, "Ad blockers: Global prevalence and impact," in *Proceedings of the 2016 internet measurement conference*, 2016, pp. 119–125.

[4] G. S. Blog, "Protect and manage browser extensions using chrome browser cloud management," 2024, accessed: 2024-05-02. [Online]. Available: https://security.googleblog.com/

[5] A. N. et al., "Exposing and addressing security vulnerabilities in browser text input fields," *arXiv*, 2023, accessed: 2024-05-02. [Online]. Available: https://dx.doi.org/10.48550/arxiv.2308.16321

[6] I. Arghire, "Password-stealing chrome extension demonstrates new vulnerabilities," 2024, accessed: 2024-05-02. [Online]. Available: https://www.securityweek.com/password-stealing-chrome-extension-demonstrates-new-vulnerabilities

[7] R. Nithyanand, S. Khattak, M. Javed, S. J. Anderson, D. Pöhn, N. Vallina-Rodriguez, and S. Sundaresan, "Ad-blocking: A study on performance, privacy and counter-measures," *arXiv preprint arXiv:1705.03193*, 2017. [Online]. Available: https://ar5iv.labs.arxiv.org/html/1705.03193

[8] B. VanderSloot and S. Sprecher, "Beyond acceptable advertisement: Better understanding blocking extensions," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:211161318

[9] C. Drazner, N. Đuza, H. Jonker, and D. S. Wallach, "Investigating the effectiveness of web adblockers," *arXiv*, 2019. [Online]. Available: https://arxiv.org/abs/1912.06176

[10] "Easylist," 2023, accessed: 2024-05-02. [Online]. Available: https://easylist.to/

[11] R. Srinivasan, "Website ad-blocker research," https://www.cmu.edu/tepper/news/stories/2020/june/website-ad-blocker-research-srinivasan-ravi.html, June 2020, accessed: 2024-05-02.

[12] L. Maddison, "Your adblocker could be facing a potentially big traffic problem," October 2022, accessed: 2024-05-02. [Online]. Available: https://www.techradar.com/news/your-adblocker-could-be-facing-a-potentially-big-traffic-problem

[13] Admiral, "10 ways adblockers break, disrupt, or impact websites," August 2022, accessed: 2024-05-02. [Online]. Available: https://blog.getadmiral.com/10-ways-ad-blockers-break-or-impact-websites

[14] L. Mathews, "A dangerous flaw in popular ad blockers put 100 million users at risk," *Forbes*, April 2019. [Online]. Available: https://www.forbes.com/sites/leemathews/2019/04/17/a-dangerous-flaw-in-popular-ad-blockers-put-100-million-users-at-risk/?sh=696bbfaf61d5

[15] TechCrunch, "Your vpn or ad-blocker app could be collecting your data," *TechCrunch*, March 2020. [Online]. Available: https://techcrunch.com/2020/03/10/your-vpn-or-ad-blocker-app-could-be-collecting-your-data/

[16] "Ad blocker usage and demographic statistics," Mar 2023, accessed: 2024-05-02. [Online]. Available: https://backlinko.com/ad-blockers-users

[17] "Implement the $rewrite filter option," Mar 2018, accessed: 2024-05-02. [Online]. Available: https://issues.adblockplus.org/ticket/6622.html

[18] A. Sebastian, "Adblock Plus filter lists may execute arbitrary code in web pages," Apr. 2019, accessed: 2024-05-02. [Online]. Available: https://armin.dev/blog/2019/04/adblock-plus-code-injection/

[19] "Crxcavator," 2024, accessed: 2024-05-02. [Online]. Available: https://crxcavator.io/

[20] "Code quality, secure & static analysis tool with sonarqube," 2024, accessed: 2024-05-02. [Online]. Available: https://www.sonarsource.com/products/sonarqube/

[21] "IEEE Symposium on Security and Privacy 2024," accessed: 2024-05-02. [Online]. Available: https://sp2024.ieee-security.org/cfpapers.html

[22] "Adblock," Official Website, 2024. [Online]. Available: https://getadblock.com

[23] "Adblock plus," Official Website, 2024. [Online]. Available: https://adblockplus.org

[24] "Adguard adblocker," accessed: 2024-05-02. [Online]. Available: https://chromewebstore.google.com/detail/adguard-adblocker/bgnkhhnnamicmpeenaelnjfhikgbkllg

[25] "Adblocker ultimate," accessed: 2024-05-02. [Online]. Available: https://chromewebstore.google.com/detail/adblocker-ultimate/ohahllgiabjaoigichmmfljhkcfikeof

[26] “ublock origin,” Official Website, 2024. [Online]. Available: https://ublockorigin.com
[27] “Adlock,” Official Website, 2024. [Online]. Available: https://adlock.com
[28] “Adblocker for youtube™,” Official Website, 2024. [Online]. Available: https://www.adblockerforyoutube.com
[29] “Ghostery,” Official Website, 2024. [Online]. Available: https://www.ghostery.com
[30] “Permissions list,” Google Chrome Developers, 2024. [Online]. Available: https://developer.chrome.com/docs/extensions/reference/permissions-list
[31] A. Meshkov, “EasyList is in trouble and so are many ad blockers,” Oct 2022, accessed: 2024-05-02. [Online]. Available: https://adguard.com/en/blog/easylist-filter-problem-help.html
[32] “Chromium,” 2024, accessed: 2024-05-02. [Online]. Available: https://www.chromium.org/Home/
[33] “CVE-2019-11593,” Apr. 2019, accessed: 2024-05-02. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-11593
[34] “CVE-2019-11594,” Apr. 2019, accessed: 2024-05-02. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-11594
[35] “CVE-2019-11595,” Apr. 2019, accessed: 2024-05-02. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-11595
[36] B. Dean, “Ad blocker usage and demographic statistics,” Feb. 2024. [Online]. Available: https://backlinko.com/ad-blockers-users
[37] S. Zhao, M. K. Chen, C. Borcea, and Y. Chen, “Personalized dynamic counter ad-blocking using deep learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8358–8371, 2023.
[38] 2024, accessed: 2024-05-02. [Online]. Available: https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS
[39] Feb 2021, accessed: 2024-05-02. [Online]. Available: https://help.adblockplus.org/hc/en-us/articles/360062733293-How-to-write-filters#rewrite-notes
[40] “Consider removing "FIN: Finnish Addition to Easylist" -filterlist from uBo assets.json · Issue #285 · uBlockOrigin/uBlock-issues,” Oct 2018, accessed: 2024-05-02. [Online]. Available: https://github.com/uBlockOrigin/uBlock-issues/issues/285
[41] “Easylist github repository,” May 2024, accessed: 2024-05-02. [Online]. Available: https://github.com/easylist/easylist
[42] “Easydutch github repository,” May 2024, accessed: 2024-05-02. [Online]. Available: https://github.com/EasyDutch-uBO/EasyDutch
[43] “The ad block wall: How anti-adblock popups harm ux and ad revenue - ad-shield.” [Online]. Available: https://www.ad-shield.io/blog/the-ad-block-wall-how-anti-adblock-popups-harm-ux-and-ad-revenue
[44] C. Gilman, “Spineless pushovers: This website has a pop-up asking you to disable your ad-blocker while still providing you with an option to continue without disabling,” Jun. 2022, accessed: 2024-05-02. [Online]. Available: https://clickhole.com/spineless-pushovers-this-website-has-a-pop-up-asking-you-to-disable-your-ad-blocker-while-still-providing-you-with-an-option-to-continue-without-disabling/
[45] “Forbes readers served malicious ads after asking them to disable adblocker - wiadomości bezpieczeństwa - trend micro pl,” accessed: 2024-05-02. [Online]. Available: https://www.trendmicro.com/vinfo/pl/security/news/cybercrime-and-digital-threats/forbes-readers-served-malicious-ads-after-asking-them-to-disable-adblocker
[46] B. Davis, “10 publishers that want you to disable your ad blocker,” Feb. 2016. [Online]. Available: https://econsultancy.com/10-publishers-that-want-you-to-disable-your-ad-blocker/
[47] “Allowing acceptable ads in adblock plus,” May 2024, accessed: 2024-05-02. [Online]. Available: https://adblockplus.org/acceptable-ads
[48] “95% of the people stick to the default option,” accessed: 2024-05-02. [Online]. Available: https://service-design.co/95-of-the-people-stick-to-the-default-option-9e16316a64e1
[49] “There are no ’acceptable’ ads,” May 2024, accessed: 2024-05-02. [Online]. Available: https://blockads.fivefilters.org/acceptable.html
[50] J. Caesar, “Is adblock safe? evaluating security and ethical concerns,” Sep 2023. [Online]. Available: https://techreviewadvisor.com/is-adblock-safe/
[51] 2015. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-the-principle-of-least-privilege