Measuring Ad-Blocker Efficiency and Privacy

Swadhin Routray
Software and Societal Systems Department
Carnegie Mellon University
sroutray@andrew.cmu.edu

Nicholas Park
Information Networking Institute
Carnegie Mellon University
hyunsoo2@andrew.cmu.edu

Jatan Loya
Software and Societal Systems Department
Carnegie Mellon University
jloya@andrew.cmu.edu

Zili Zhou
Software and Societal Systems Department
Carnegie Mellon University
ziliz@andrew.cmu.edu

Abstract

This research study aims to conduct and provide insights into Ad-Blocker efficiency and privacy. It is no surprise that with the advent of the internet, digital advertisements have become commonplace since most people spend a significant time online. These advertisements, while useful sometimes, tend to be intrusive and invade user privacy. We aim to find the best possible ad blocker in terms of efficiency and privacy protection, by running experiments with the most popular ad blocker options on Fortune 250 websites. We aim to derive a comparative result based on our experiments.

1 Introduction

In today's society, it is not uncommon to see third-party advertisements on most websites we browse. This is because to support mostly free websites companies incorporate "third-party iframes and Javascript to show ads and track users' activities on websites" [5]. Furthermore, to make these ads more effective, companies allow what is called "online profiling", which is the method of tracking users' personal information such as their browsing history, product search history, and product purchase history, etc. [4] While this can be beneficial to a company's profit, there have been concerns that these methods of targeted advertising and tracking bring upon severe privacy concerns as they dabble upon the users' sensitive data.

To combat the pervasive infiltration that happens to users' privacy numerous developers have created ad and tracker blockers to prevent companies from gathering user data in the first place. Some popular options include Ghostery and AdBlockPlus which are known to crack down on irrelevant web requests that the browser sends to third-party trackers and advertisers [4]. Other approaches include popup blocking, third-party cookie blocking, and utilizing the Do Not Track header which can mitigate the effects of cookie-based tracking without severely impacting the browser's functionality [8]. Radical solutions include disabling Javascript but this has been widely known not to be a wise solution as most websites rely on this language.

Fortunately, there has been an increasing trend in research regarding the diverse aspects of online tracking. Regarding why we should care about ads and trackers, Shiller, Waldfogel, and Ryan have observed that websites that host multiple users with ad blockers enabled also experience less overall traffic [9] which consequentially provides a correlation between website performance and ad blocker usage. Concerning research solely about ad and tracker blockers, Garimella et al. have compared the performances of the 5 most popular ad-blocking extensions by analyzing the HAR data with each extension usage [3]. This is particularly a significant reference point to us as we build upon their research and we try to include and test additional extensions from different browsers as well.

Following the previous statement, we have decided to test the effectiveness of some known ad and tracker blockers. The significance of this experiment lies in the fact that a good majority of people rely on blockers to prevent privacy intrusion. If those blockers do not function properly, or in a worse case log and sell the users' data to a third-party vendor, the blocker becomes nothing better than a placebo to people who believe their online privacy is being secured. Therefore, it is crucial to users' privacy that researchers correctly verify that ad and tracker blockers are working as intended.

For our research, we have chosen four popular ad blockers as our test subjects. Our controlled environment includes testing the extensions of three different browsers (Chrome, Firefox, Edge) and running the tests on the websites belonging to the Fortune 250 list. The first step of the experiment involves gathering the HTTP Archive format (HAR) files from each user of our script. We were able to complete this by building a script using both the Selenium and Browsermob Proxy libraries. What comes next is analyzing the HAR files using a custom script, which generates aggregate statistics from our HAR data. We then refine these results by certain metrics such as "Average Distinct Requests" which could aid us in comparing the effectiveness of the 4 ad blockers.

In the remainder of the paper, the outline of the upcoming sections is as follows. Section 2 will cover related works that are grouped by popular themes about this topic. In section 3, we will detail our testing methodology and organize the results in section 4. Section 5 marks a discussion chapter in which we discuss the conclusions from our results and what impact that could have on the perspective regarding specific ad and tracker blockers. Finally, in the last section, we summarize our work and provide a conclusion to our report.

2 Background and Related Work

In this section, we focus on a few papers and articles that discuss ad blockers, their impact, and current methods and strategies to mitigate threats and maximize the ad blocker's impact on a user's experience.

2.1 Why care about ads and trackers?

As online advertising becomes prevailing and highly personalized, Shiller et al. [9] interpreted the observation that "sites with more users who block targeted ads experience reductions in traffic" in a way that with ads being blocked, websites experience reductions in revenues which undermine investment and therefore website quality, and that makes consumers less likely to visit them. That provides us an insight into the performance of ad blockers from the perspective of websites. Wielki and Grabara [11] explored the development of the ad-blocking phenomenon and its impact on the digital advertising ecosystem and showed its destructive effect on the functioning of the electronic commerce system. Butler [2] comprehensively examined the "technical functionality of ad-blocking software and the ways internet content providers have responded" and raised legal and ethical arguments around the use of ad-blocking software. The paper elaborated and inspired us on the privacy-related concerns of ad-blocking, including its effect on do-not-track cookies, opt-in basis, and restrictive terms of service. All of the articles show the necessity to study ad blockers and evaluate their effectiveness.

2.2 Ad & Tracker Related Work

Mathur et al. [6] outlined an approach to measuring user awareness and thereby using it as a mechanism to test browser failure because of ad blockers. The paper helps us keep a special focus on these blocking methods and thereby, look into whether they still employ the method and draw a comparison to the previous study.

Nithyanand et al. [7] detailed methodology to detect the existence of anti-ad blockers and mentioned as further defense, some popular blockers deploy counter-block anti-ad blocking scripts. We consider these as possible factors affecting the performance of ad blockers. Brinson et al. [1] found users' negative perceptions about the use of third-party data by advertisers but not for first-party data, which inspires us that some ads/tracking might not be blocked because they are from first-party partners.

Garimella et al. [3] chose the top 5 most popular ad-blocking extensions from the google chrome extension store and compared their performances by capturing data in the form of HAR (HTTP Archive File) which includes network requests, types and sizes of objects, and load times. Their paper serves as a very good foundation for our project as we build on top of it to include more popular browser extensions and try them on Google Chrome and Mozilla Firefox, and also compare these extensions with Brave Browser's Shield. Singh and Potdar [10] laid out a model which entailed how advertisers purchase ads from publishers, detailed the methodology undertaken to evaluate the different ad blockers, and elaborated upon different methods to bypass an ad blocker as well. These bypass strategies help us build edge cases into our test suite for evaluation.

3 Methodology

In this paper, we analyze the performance of four modern ad blockers: AdBlock, AdBlock Plus, UBlock Origin, and Ghostery, in three different browser settings: Chrome, Firefox, and Edge. To draw statistics, we use the Fortune 250 list and run our experiments on these websites.

3.1 Data Generation using Selenium and Browsermob Proxy

We use HTTP Archive format (HAR) files to collect different analytics. HAR files are similar to Javascript Object Notation (JSON) files and contain a complete log of all the interaction that takes place between a website and other domains in a browser session while keeping a track of the time taken to retrieve data from third-party websites and the different load times and the list of different domains called, among other information that is generated during a browser session.

In order to generate the HAR files, we use the Selenium Python library and the Browsermob Proxy Library to generate the HAR files. We first establish a proxy using the Browsermob proxy library. This is essential, Browsermob proxy allows us to modify HTTP requests and answers, capture HTTP content, and export performance statistics as a HAR file. We embed this proxy in our Selenium script and allow it to collect the browser log when a session is initialized.

Once the proxy is set up, we set up the browser engine in that we are going to run the different ad blockers. Selenium provides the flexibility of running different browser engine binaries for different purposes. Once the binary is initialized, we add an ad blocker extension to the driver and pass the proxy created using Browsermob Broxy as an additional option, in order to track the session and generate the HAR files.

Once the initial setup of the browser engine and extension is complete, we load the file which contains the list of Fortune 250 websites and iterate through the list. Each iteration initializes an instance of the browser and makes an HTTP GET request to the website associated with the iteration. Each iteration ends with the generation of a HAR file that retains a complete log of the interaction between the website and other domains and third-party applications for each ad blocker.

We run this experiment of four ad blockers and three browser engines, for each of the Fortune 250 websites. This results in the generation of 750 HAR files for each ad blocker, and a total of 3750 HAR Files, including our baseline HAR files. This provides a large dataset for the efficiency measurement of the different ad blockers.

It is important to note that the packaged ad blockers have different formats for different browser engines and therefore the ad blocker used has the same version but a different binary that is loaded. For Example, in order to load AdBlock to the Chrome engine, we require *.crx* binary of the extension. Whereas for AdBlock to run on Firefox engine, we require the *.xpi* binary. Thus, this paper provides a look into the differences in efficiency between binaries for different browser engines, for the same ad blocker.

3.2 Analyzing HAR files

We built a custom script to analyze the HAR files and to generate aggregate statistics for the different sets of HAR files generated by our script. To draw comparisons and conclusions between the efficiency of different ad blockers and browser combinations, we define a few metrics for measurement.

- **Average Distinct Requests**: Average Number of distinct HTTP domain requests made during page load.

- **Average Median Time**: Average Median time taken for page loads. This is obtained by summing up the duration of all the different domain requests while the page is loading.

- **Average Maximum Calls to Top URL**: Average of all the URL loads to the top URL from each website.

- **Average Longest URL Call**: Average of the longest page load across different websites.

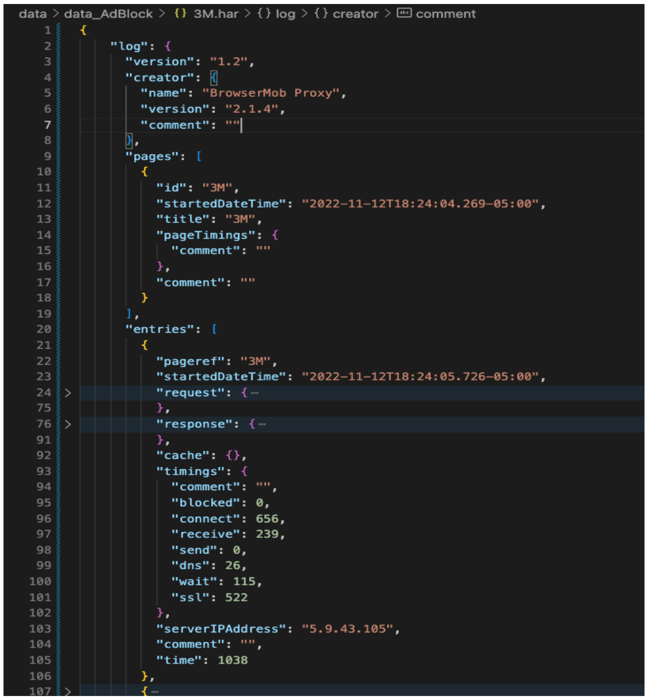
- **Average Ninety-Fifth Time**: Average Time taken for 95% of the page loads. We drop 5% of the slowest times in a website load.

We conduct our experiment using this setup for the different ad blockers and website configurations and draw a comparison between the performance and efficiency of these ad blockers.

4 Results

For the results, we would mainly be looking at the statistical analysis of the HAR files collected by our mechanism mentioned previously.

Ideally, when we use an Ad-Blocker, it usually works by blocking the network requests for websites that serve the ads in the first place.


data > data_AdBlock > {} 3M.har > {} log > {} creator > comment

```
{
    "log": {
      "version": "1.2",
      "creator": {
        "name": "BrowserMob Proxy",
        "version": "2.1.4",
        "comment": ""
      },
      "pages": [
        {
          "id": "3M",
          "startedDateTime": "2022-11-12T18:24:04.269-05:00",
          "title": "3M",
          "pageTimings": {
            "comment": ""
          },
          "comment": ""
        }
      ],
      "entries": [
        {
          "pageref": "3M",
          "startedDateTime": "2022-11-12T18:24:05.726-05:00",
          "request": {⋯
          },
          "response": {⋯
          },
          "cache": {},
          "timings": {
            "comment": "",
            "blocked": 0,
            "connect": 656,
            "receive": 239,
            "send": 0,
            "dns": 26,
            "wait": 115,
            "ssl": 522
          },
          "serverIPAddress": "5.9.43.105",
          "comment": "",
          "time": 1038
        },
        {⋯
```

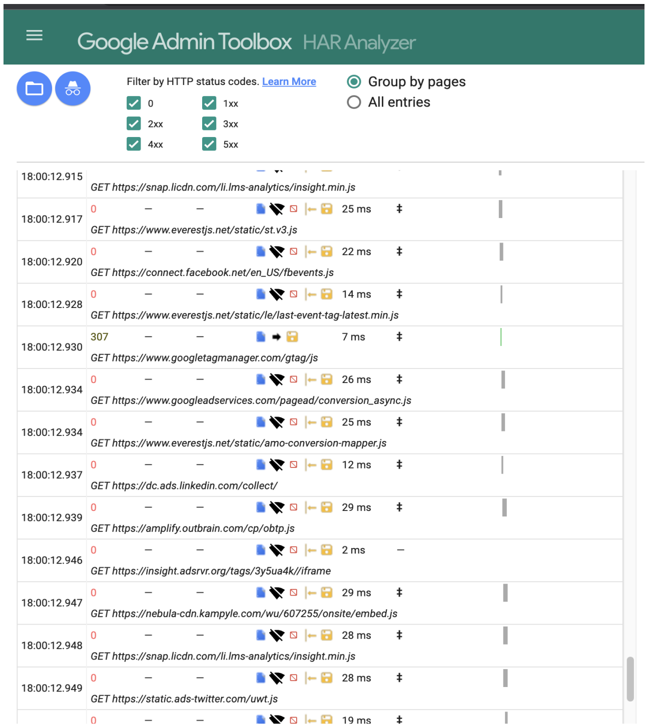
Figure 1: Sample HAR data for www.3M.com

Figure 3 to 7 are the average distinct requests, average median time, average ninety-fifth time, average median calls, and average longest URL call of the three browsers - Chrome, Edge, Firefox - in vanilla mode, with each of the four ad blockers - Adblock, AdblockPlus, Ublock, Ghostery - and an average value of these features with ad blockers, calculated by experiments.

To be more clear, the last group of bars, marked as "Average With Blockers", is the average of the values for Adblock, AdblockPlus, Ublock, and Ghostery, serving as a reference to show the general performance of ad blockers compared to the vanilla mode. The blue bars are the data for Chrome, the orange is for Edge, and the grey is for Firefox.

5 Discussion

By treating online advertisements as URL calls, we could quantify the performance of ad blockers by the proportion of distinct HTTP domain requests they filtered. From Figure 3, we see that on average, with an ad blocker,



Filter by HTTP status codes. Learn More

☑ 0 ☑ 1xx
☑ 2xx ☑ 3xx
☑ 4xx ☑ 5xx

● Group by pages
○ All entries

```
18:00:12.915
GET https://snap.licdn.com/li.lms-analytics/insight.min.js
18:00:12.917  0    –    –    25 ms   ‡
GET https://www.everestjs.net/static/st.v3.js
18:00:12.920  0    –    –    22 ms   ‡
GET https://connect.facebook.net/en_US/fbevents.js
18:00:12.928  0    –    –    14 ms   ‡
GET https://www.everestjs.net/static/le/last-event-tag-latest.min.js
18:00:12.930  307  –    –    7 ms    ‡
GET https://www.googletagmanager.com/gtag/js
18:00:12.934  0    –    –    26 ms   ‡
GET https://www.googleadservices.com/pagead/conversion_async.js
18:00:12.934  0    –    –    25 ms   ‡
GET https://www.everestjs.net/static/amo-conversion-mapper.js
18:00:12.937  0    –    –    12 ms   ‡
GET https://dc.ads.linkedin.com/collect/
18:00:12.939  0    –    –    29 ms   ‡
GET https://amplify.outbrain.com/cp/obtp.js
18:00:12.946  0    –    –    2 ms    –
GET https://insight.adsrvr.org/tags/3y5ua4k//iframe
18:00:12.947  0    –    –    29 ms   ‡
GET https://nebula-cdn.kampyle.com/wu/607255/onsite/embed.js
18:00:12.948  0    –    –    28 ms   ‡
GET https://snap.licdn.com/li.lms-analytics/insight.min.js
18:00:12.949  0    –    –    28 ms   ‡
GET https://static.ads-twitter.com/uwt.js
              0    –    –    19 ms   ‡
```

Figure 2: We can see that domains serving advertisements are blocked by the extension

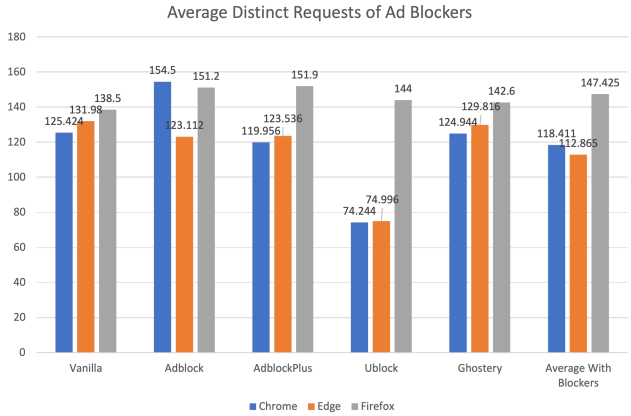


Figure 3: Average Distinct Requests of Ad Blockers

118.411/125.424 = 94.4% of distinct requests made through ad blockers compared to the vanilla mode, and among all blockers, the average number of distinct requests after being blocked by Ublock were only 74.244/125.424 = 59.2% of the original. The figures are calculated by Chrome data, but based on Edge data, there were only 112.865/131.98 = 85.5% of the distinct requests on average made through ad blockers and 74.996/131.98 = 56.8% made through Ublock. That shows ad blockers are effectively blocking the ads to some extent, of which the best performance (Ublock) could be blocking over 40% of the distinct HTTP domain requests.

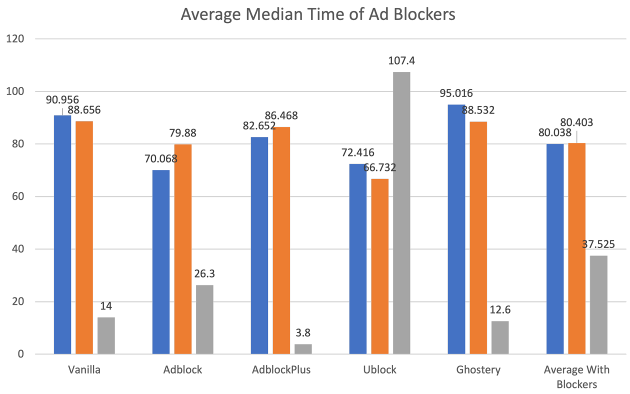


Figure 4: Average Median Time of Ad Blockers

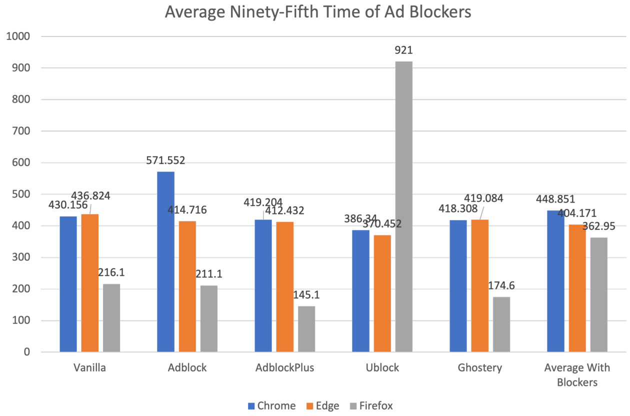


Figure 5: Average Ninety-Fifth Time of Ad Blockers

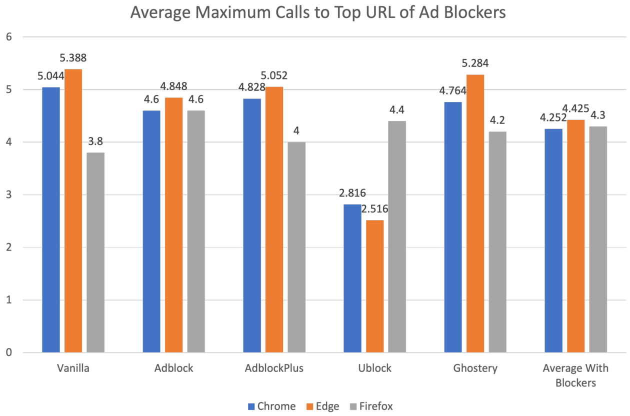


Figure 6: Average Maximum Calls to Top URL of Ad Blockers

To further evaluate the performance of ad blockers, we tracked the average median loading time and the average

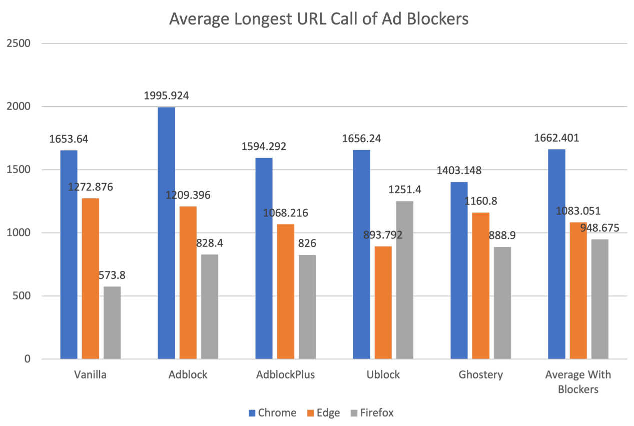


Figure 7: Average Longest URL Call of Ad Blockers

ninety-fifth loading time of ad blockers, as Figures 4 and 5 show. While the average median time shows how in general ad blockers are capable of blocking ads and saving time for users, the average ninety-fifth time emphasizes more on blockers' stable functioning and capability of blocking or shutting down extreme advertisement URL calls that take a long time to load. We find these figures great measurements of an ad blocker's performance and usability, especially from the perspective of users. Similar to figures 4 and 5, the smaller the values are, the better the performance is.

More specifically, from Figure 4, we can see that the average median time with and without blockers is 80.038 vs 90.956 for Chrome and 80.403 vs 88.656 for Edge. As for the best performance, Adblock shortened the median loading time to 70.068 for Chrome and Ublock shortened the median loading time to 66.732 for Edge. From Figure 5, we can see that although the average ninety-fifth time with blockers was extended from 430.156 to 448.851 based on Chrome data, it was actually caused by the anomaly of Adblock while the other three blockers all shortened the ninety-fifth loading time. Given that Adblock significantly lowered the median loading time, our best guess is that Adblock did not prioritize blocking ads with long loading times and that contributed to a long ninety-fifth loading time, or Adblock did not perfectly compile with Chrome. Edge data shows that on average, blockers shortened the ninety-fifth loading time from 436.824 to 404.171 with Ublock lowering that to 370.452. To conclude, blockers are effectively blocking ads and saving the web page loading time for the users, and are generally capable of functioning stably and blocking or shutting down ads with long loading times.

We also included the average maximum number of calls to the Top URL and the average longest URL call in our measurements, because URL calls mean data transferred and consumed, and we hope these figures could serve as an indication of ad blockers' capabilities of saving data. Similar to the loading time, while average maximum calls to the top URL act as a general measurement of blockers' ability to block ads and save data, the average longest URL call emphasizes more blockers' abilities to handle extreme cases and block or shut down a large number of calls.

From Figure 6, we see that on average blockers reduced the maximum calls to the top URL from 5.044 to 4.252, of which Ublock reduced that to 2.816 based on Chrome data, and on average blockers reduced the maximum calls to the top URL from 5.388 to 4.425, of which Ublock reduced that to 2.516 based on Edge data. From Figure 7, we see that on average blockers actually increased the longest URL call from 1653.64 times to 1662.401 times, of which Ghostery reduced that to 1403.138 based on Chrome data. Again, we don't see a reduction on average because of the anomaly of Adblock, which might confirm our guess on how Adblock did not prioritize blocking time-consuming and data-consuming ads, or it did not perfectly compile with Chrome. Based on Edge data, on average blockers reduced the longest URL call from 1272.876 times to 1083.051 times, of which Ublock reduced that to 893.792. To conclude, blockers are effectively blocking ads and saving data for the users, and are generally capable of blocking or shutting down ads with a large number of calls.

We can tell from the analysis that, ad blockers perform differently on different browsers. We did not mention the Firefox data, because compared to the Chrome data and Edge data, there were cases in which we detected significantly higher or lower values on Firefox, as shown by the figures. This is primarily due to the fact that Chrome and Edge share the same browser engine - V8 engine whereas Firefox runs on the Gecko driver. We suspect that this overhead is due to Gecko using Brotli compression and the V8 engine using Gzip compression. Our selenium and the browsermob-proxy tool were originally designed for supporting Chromium-based browsers and are not readily compatible with Firefox from our experience. Hence this mismatch is what we suspect leads to outliers in the results. Apart from that, while there were some differences in the values of Chrome and Edge data, they both shared a similar trend and pattern.

Lastly, we hope to determine the ad blocker with the best overall performance. We chose the Edge data for such analysis, because of the anomaly of Adblock on Chrome. Based on Edge data, we ranked each feature and built a spider web graph for each of the blockers, with the outer layer showing a higher ranking among all blockers, and the inner layer showing a lower ranking. We ranked the blockers in such a way that the smaller the feature value is, the better performance indicated, and the higher the ranking should be. In another word, the blocker with the largest web area would be the one with the best overall performance. Based on Edge data, we can easily tell from Figures 8 to 11 that the performance web of UBlock has the largest area and is ranked first for all of the measurements.

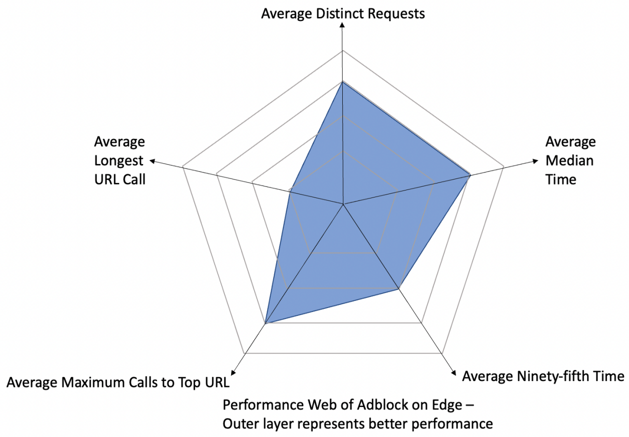


Figure 8: Performance Web of Adblock on Edge

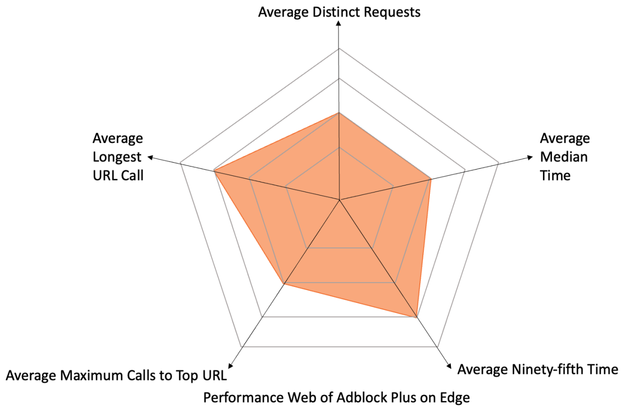


Figure 9: Performance Web of Adblock Plus on Edge

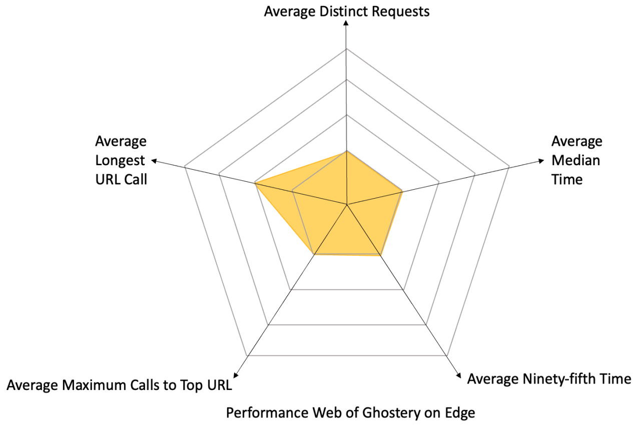


Figure 10: Performance Web of Ghostery on Edge

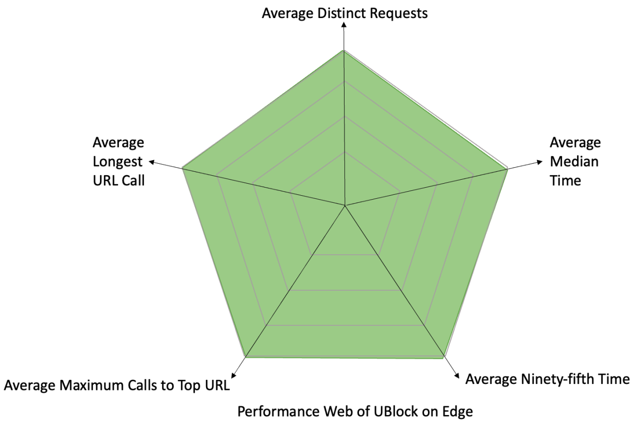


Figure 11: Performance Web of UBlock on Edge

6 Conclusion

In this work, we have successfully measured the range of efficiency of different ad blockers in various browser environments. By aggregating certain properties of the HAR files generated by our Python script, we were able to compare metrics such as the average median time, the average distinct requests, etc. of each ad blocker in their own distinct setting. Our conclusion to this experiment reveals that in general, ad blockers perform differently in each browser setting. However, a close winner with regard to efficiency and speed is Ublock when used in Google Chrome. This result is, of course, after nullifying the Firefox results considering that Firefox did not accept our script very well.

Therefore, future plans to improve our current testing include:

- Creating an optimized script to test properly for the Firefox browser
- Gauging if HAR analysis is the best method to test for ad blocker efficiency and privacy
- Ultimately adjoining the current script and the optimized script such that we have one tool that can run seamlessly on all browsers

These extra steps will allow us to achieve a more uniform and unbiased result on our findings. Presenting our research poster to our peers will also allow us to receive feedback and aid us in contributing more to this field of privacy and security.

References

[1] Nancy Brinson, Matthew Eastin, and Vincent Cicchirillo. Reactance to personalization: Understanding the drivers behind the growth of ad blocking. In *Journal*

of Interactive Advertising, volume 18, pages 136–147, 2018.

[2] Ian C. Butler. The ethical and legal implications of adblocking software note. In *OpenCommons@UConn*, 2016.

[3] Kiran Garimella, Orestis Kostakis, and Michael Mathioudakis. Ad-blocking: A study on performance, privacy and counter-measures. In *CoRR abs/ 1705.03193 (2017)*, 2017.

[4] Arthur Gervais, Alexandros Filios, Vincent Lenders, and Srdjan Capkun. Quantifying web adblocker privacy. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 21–42, Cham, 2017. Springer International Publishing.

[5] Saad Sajid Hashmi, Muhammad Ikram, and Mohamed Ali Kaafar. A longitudinal analysis of online ad-blocking blacklists. In *2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*, pages 158–165, 2019.

[6] Arunesh Mathur, Jessica Vitak, Arvind Narayanan, and Marshini Chetty. Characterizing the use of browser-based blocking exten- sions to prevent online tracking. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, 2018.

[7] Rishab Nithyanand, Sheharbano Khattak, Mobin Javed, Narseo Vallina-Rodriguez, Marjan Falahrastegar, Julia E. Powles, Emiliano De Cristofaro, Hamed Haddadi, and Steven J. Murdoch. Ad-blocking and counter blocking: A slice of the arms race. In *arXiv.org*, 2016.

[8] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against Third-Party tracking on the web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 155–168, San Jose, CA, April 2012. USENIX Association.

[9] Benjamin Shiller, Joel Waldfogel, and Johnny Ryan. The effect of ad blocking on website traffic and quality. In *The RAND Journal of Economics*, volume 49, pages 43–63, 2018.

[10] Ashish Kumar Singh and Vidyasagar Potdar. Blocking online advertising - a state of the art. In *2009 IEEE International Conference on Industrial Technology*, 2009.

[11] Janusz Wielki and Janusz Grabara. The impact of adblocking on the sustainable development of the digital advertising ecosystem. In *Sustainability*, volume 10, 2018.