

# Summary of *How Effective Are Commonly-Used Ad-Blockers?*

MATTHEW OROS, SONNY SMITH, MICHAEL TEREKHOV, DENNIS ULICHNEY

## ABSTRACT

With the development of the internet web pages, online advertisements have also developed. Online advertisements can help find products that a user has been looking for, but they often ruin user experiences. Online advertisements might take up the whole space of a web page, and for mobile users, online advertisements might even make the usage of the page impossible. This research focuses on the software that removes such advertisements from web pages. The research paper includes an introduction, breakdown/related work, experiment design, results, and conclusion/future work.

## INTRODUCTION

The services offered online are mostly free and the most common way to monetize them is through online advertising. Online ads can be helpful, but most users find them to be one of the most annoying things on the internet. Most internet users want to minimize the number of ads they see and various ad-blocking software can be used to achieve this. The easiest and most popular way to block ads is to use browser extensions. An advanced ad-blocking method is to use specialized software that blocks incoming ad domains. Advanced software is considered more reliable, which means it blocks more ads. The purpose of this study is to find out how effective ad-blocking extensions are. This study focuses on the most popular browser ad blockers and their effectiveness compared to Pi-Hole software (the advanced software used for this study). Ad-blocking extensions that were utilized are Adblock Plus, Adblock, uBlock Origin, Privacy Badger, and Ghostery. Internet browsers that were utilized to test the software are Chrome, Microsoft Edge, and Firefox. Also, an integral part of this research work was the MITM Proxy (Man-in-the-Middle Proxy), which was used to capture all needed data for testing the effectiveness of the above-specified software. Man-in-the-Middle Proxy was used to collect all the domains that passed the tested web pages while the ad-blocker software was active. In order to parse the collected data for a better analysis the programming language Python was utilized, further specifications can be found in the section Experiment Design.

## BACKGROUND/RELATED WORK

With the widespread use of ad-blockers, there have been numerous studies on their usage and effectiveness. One such study, for example, combined information from two million users from across the world with data on close to two trillion web requests to assess the usage and impact of ad-blockers across the Internet. The study found that, among other things, ad-blockers reduced the number of ads seen by about half, with the other half making it through [3]. Unlike our study, however, this one was more focused on the demographics of ad-blocker usage as well as some of its financial effects on website publishers. Because of this, it only looked at ad-blockers in a more general sense, instead of evaluating the performance of specific ones.

Another study, on the other hand, conducted a more specific analysis of the performance of different ad-blockers, many of which were also tested in our study. These ad-blockers were evaluated based both on how they affected

---

Author's address: Matthew Oros, Sonny Smith, Michael Terekhov, Dennis Ulichney.

---

2022. Manuscript submitted to ACM

the performance of the browser and how many user-tracking requests they blocked, with uBlock having the best performance. This study primarily differs from ours in its construction by looking at how ad-blockers affect browser performance, which we did not consider. Additionally, the study also only used one internet browser, Google Chrome, and also only made use of news websites, limiting the variety of the data collected [1].

It is also worth mentioning that ad-blockers do not go unchallenged in their work, with an ongoing arms race between them[them] and advertisers. This, naturally, has resulted in a variety of methods for getting past ad-blockers. One such method, according to another study, is the use of replica ad domains. Replica ad domains are new advertising domains that advertisers will create in hopes of getting past the list of blocked domains that powers most ad-blockers. This method also seems to be highly effective at overcoming ad-blockers as well, with the researchers finding that some replica ad domains would, on average, extend the lifetime of both ads and trackers by 784.7 days before being blocked [2]. Furthermore, it is entirely possible, if not, likely that some of the advertisers on the websites looked at in our study have made use of this technique, which could have resulted in some ad domains flying under the radar of the ad-blockers that we used. Now, after looking at some of the past research on ad-blockers, we will detail our own methods and findings next.

## EXPERIMENT DESIGN

Our testing methodology involves visiting a list of 30 websites with various ad-blocking extensions. We first recorded their effectiveness using the number of ads blocked reported by the extension. This information is self-reported by the ad-blocker so we then compared the results of this data to our recorded results using a network proxy.

Ad-blocking extensions can work in various ways from analyzing HTML elements on a webpage and comparing them to a known database to DNS to blocking DNS queries or a combination of multiple methods. Because of this flexibility, it can be difficult to fully assess the capabilities of a number of different ad-blocking extensions. In this study, we focused on the area of DNS query based blocking.

DNS stands for Domain Name System and is the system that maps human memorable domain names, such as google.com, to IP addresses that computers use to connect to a web server. DNS based blocking looks for requests the browser makes to certain domain names such as ads.google.com and instead of forwarding the request, blocks it by routing the domain to an invalid IP such as 0.0.0.0.

To test the effectiveness of the ad-blocker we use a proxy called MITM-Proxy to record all DNS (domain) requests when we visit each website. This was done using a Python script addon for MITM-Proxy. We were able to store all the unique domain requests into a text file for later processing. Using crowd-sourced ad-lists, we compared and counted how many of the domains requested are considered ads despite the ad-blocker being installed. This process was done using Python. To parse the known ad-lists, a hosts file parser library was used which then stored the domains into a list that can be easily compared to the list of requests produced by the proxy. The script then output the number of total requests and then the number of those requests that matched the ad list. This data was stored into a spreadsheet and later graphed. Python was also used to create many of the visualizations used for our results.

Pi-hole is a DNS based ad-blocker and by using the same ad-lists that Pi-hole uses by default to test the effectiveness of the traditional ad-blocking extensions, we are able to predict the effectiveness of Pi-hole without having to actually run a Pi-hole instance. This allows us to compare the effectiveness of a DNS based ad-blocking solution to/with traditional ad-blocking extensions. The ad-list Pi-hole uses by default is called the Steven Black Hosts which are is a crowd-sourced list of ads. This list is updated daily with domains. There are multiple versions of these ad-lists which provide greater

coverage beyond ads such as known malicious websites, fake news sites, and pornography. The ones chosen for this study contains both known ad domains and malicious domains which is the most popular option.

The outcome of the experiments include data about self-reported ads blocked by browsers extensions, number of ads per website tested, total requested domains per ad-blocker, and ratio of domains that are considered ads based on that total. The ratio is determined by which requested domains are also found in the default Pi-hole ad-list.

## RESULTS

The number of domains and ads blocked was first gathered from the blockers while in use and then gathered from proxy logs for comparison. The results of the first test found that uBlockOrigin detected and blocked the most domains on average followed by Ghostery and PrivacyBadger. uBlockOrigin blocked over 50% more domains on average than the other blockers as shown in figure 1a. Although PrivacyBadger is not marketed as an adblocker, it blocked more domains on average than Adblock and Adblock Plus. Adblock and Adblock Plus blocked the lowest number of domains on average with minimal difference between the two blockers.

The second test found that overall there were more domains and ads blocked by each blocker on average than reported by their icons as shown in figure 1b. Similarly to the first test, uBlockOrigin was found to block the most domains due to the fewest number of domains and ads being detected while it was active. With this test there was an inverse relationship with the blocker's performance and the number of domains detected and blocked. The fewer the amount of total domains and ads detected is, the better the blocker. uBlockOrigin had less than 200 total domains to block while the other blockers had more than 400 total domains which indicates that uBlockOrigin performed 50% better than the other blockers in this test as well. Adblock Plus had the most total domains active of nearly 600. In comparison, Adblock, Ghostery, and PrivacyBadger all had around 450-475 total domains active. Adblock and Ghostery had similar total domain counts while Ghostery had a greater proportion of ad-domains. PrivacyBadger fell somewhere inbetween Ghostery and Adblock in terms of total domain and ad-domain counts.

## CONCLUSION/FUTURE WORK

In this study, we made use of the Stephen Black Hosts advertising domain list and a MITM Proxy server to compare the effectiveness of several popular ad-blocking extensions with Pi-Hole. These ad-blocking extensions consisted of Adblock, Adblock Plus, uBlock Origin, Ghostery, and Privacy Badger. To determine their effectiveness, we considered the number of domains that they claimed to block as well as the number of domains that still got through according to the MITM proxy. Using the Stephen Black Hosts list, the default filter list used by Pi-Hole, to identify ad domains that Pi-Hole would have blocked, we found that all of the ad-blockers tested blocked fewer ad domains from the list than Pi-Hole, with uBlock Origin blocking the most. Considering this, we conclude that using Pi-Hole will result in more ads being blocked than the ad-blocking extensions that we studied in this paper. [For future work, we acknowledge that that the use of multiple ad-blocking extensions and the combination of these extensions with Pi-hole could be more effective at blocking ads and other unwanted domains. We also believe that performance degradation should be further investigated to provide insight on the potential trade-offs of using ad-blocking extensions and Pi-hole. ]

## REFERENCES

- [1] GARIMELLA, K., KOSTAKIS, O., AND MATHIOUDAKIS, M. Ad-blocking: A study on performance, privacy and counter-measures. In *Proceedings of the 2017 ACM on Web Science Conference* (New York, NY, USA, 2017), WebSci '17, Association for Computing Machinery, p. 259–262.
- [2] LIN, S.-C., CHOU, K.-H., CHEN, Y., HSIAO, H.-C., CASSEL, D., BAUER, L., AND JIA, L. Investigating advertisers' domain-changing behaviors and their impacts on ad-blocker filter lists. In *Proceedings of the ACM Web Conference 2022* (New York, NY, USA, 2022), WWW '22, Association for Computing

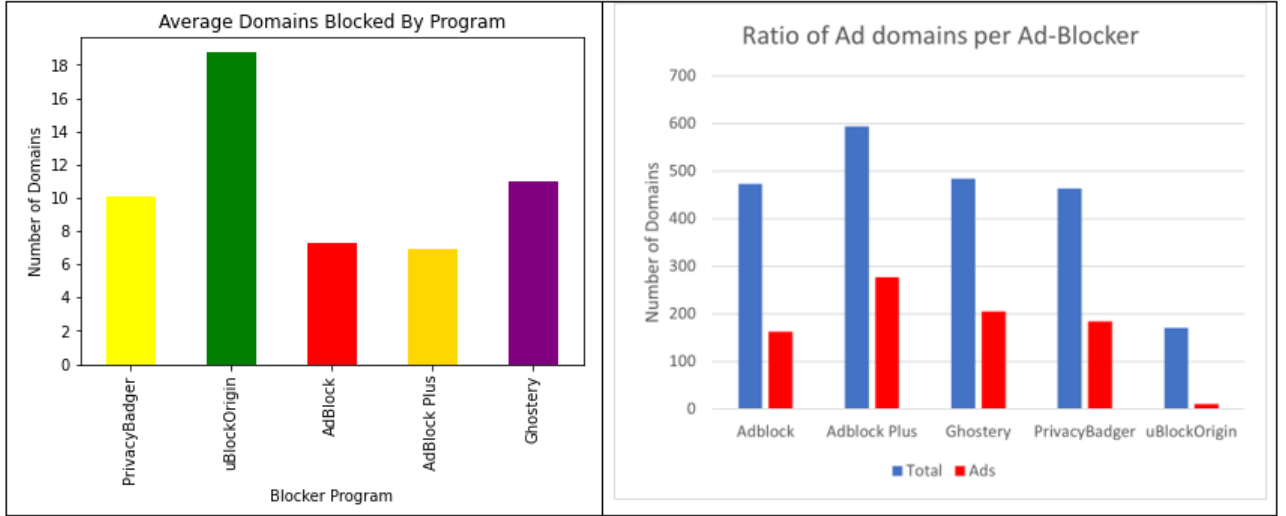


Fig. 1. The leftmost graph(a) shows the average number of ad-domains reported by each adblocker. The rightmost graph(b) compares the total number of domains detected with a proxy to the number of those domains that are ads while each blocker was active.

Machinery, p. 576–587.

- [3] MALLOY, M., McNAMARA, M., CAHN, A., AND BARFORD, P. Ad blockers: Global prevalence and impact. In *Proceedings of the 2016 Internet Measurement Conference* (New York, NY, USA, 2016), IMC '16, Association for Computing Machinery, p. 119–125.