Ad-Blocking with AdGuard

Network wide Ad-Blocking with Raspberry Pi

Akshay Kadav, Sankalp Madhavi, Tejas Gorivale and Dr. Vrajesh Maheta
Department of Information Technology Engineering
Terna Engineering College, Nerul, Navi Mumbai, Maharashtra, India

Abstract—

The internet is a scary place. Businesses track users through advertisements (Ads) that they push. Cyber criminals try to scam and steal data through Ads within minutes. Hence it is necessary to overcome the problem of Ads. Ads are annoying and pose a threat to security and also compromise performance and efficiency. It quickly becomes scary as the products that were recently searched for appear as Ads to the user. In the digital area, Privacy is considered as a Myth and Ads sum up to be a big part of that reason. Hence to preserve privacy, the solution is proposed in this paper as Adguard which filters the Ads and blocks them at a network level with DNS filtering. The Adblocker blacklists Ad agency domains, in turn, the request doesn't reach the Ad agency and thereby Ads are not fetched and displayed. Adguard is a collection of software packages that provide filtering and advertisement blocking for all internet services on the network. It alleviates what might traditionally be done via Ad blockers or per-device software because it blocks things at the DNS level. It uses DNS masq, Flask , FTL and some other packages to tie all this together into a nice, easy-to-use web interface. Adguard applications can be installed on any existing Raspberry Pi device or a Virtual Machine. Adguard utilizes several well-known and trusted internet blacklists and keeps them up-to-date. At the DNS level it can then thwart unwanted Ads, malware domains and other unsavory internet denizens from appearing on any of the internet devices or computers. The main motive of the system is to make the internet a safe place and preserve the user’s privacy using Ad-blocking application Ad-Guard.

Keywords—Raspberry Pi, DNS, IPV4, IPV6, Internet Advertisement, DNS filtering, AdBlocker.

I. INTRODUCTION

Online advertising exists in a variety of forms, including web banners, pictures, animations, embedded audio and video, text, or pop-up windows, and can employ audio and video autoplay. For an average user, a typical day on the Web involves exposure to Ads. There has been substantial research into user perception of online Ads and the steps taken to avoid them. Ads are often seen as annoying, or lead to a negative Web browsing experience [1, 2, 3], and the prevalence of behavioral or retargeted Ads raises concern about privacy [4, 5, 6]. There are a number of ways that employing Ad blockers to improve users’ experience on the Web, beyond avoiding the annoyance of Ads.The amount of data loaded when visiting pages with Ads is significantly reduced with them [7, 8], leads to savings in both load time and data cost on mobile. In fact, among a selection of popular news sites, over half the data loaded (in aggregate) was found to be Ad-related [9]. In addition to boosting site performance, blocking Ads reduces exposure to privacy and security threats associated with Ads such as behavioral tracking and malvertising [10, 11, 12].

The proposed Adguard uses home-based techniques, like the inexpensive and energy-efficient Raspberry Pi, to protect all of the devices on the home network from unwanted content. It is a network-level advertisement and internet tracker blocking application which acts as a DNS sink-hole which is intended for use on a private network. When the webpage is being loaded it makes requests to fetch the required data and Ads together from a separate server. The website has only control over the server that sends the data but has no control over the Ads being displayed.

The system works on the network level and hence it doesn't require any client software or special setup for devices in the network. It makes it possible to block Ads on any device,such as smart TV’s that do not allow any modifications. It comes with a Web interface that offers a central place to view and monitor statistics.The web interface can be accessed with any device with a browser within the network. But for network admins, AdGuard can also be used as a network monitoring tool as it can record all DNS queries sent to it and hence it is possible to analyze and review traffic. This can be particularly helpful during any network investigation and it is also possible for AdGuard to increase network speed. This paper mainly highlights the use of Adguard and the procedure to use it effectively as a DNS on the network.

II.LITERATURE REVIEW

Adblock Plus [13] is an extension that allows the user to improve the web experience. It is available for almost all popular browsers. The system is mostly used by masses who don't understand technology. It is a great tool for public networks although it fails to offer network wide protection in a private network. The system doesn’t work at the network level and therefore the Ads are fetched by the system in turn consuming system resources and network bandwidth. Although the popularity of the Adblock plus system grew due to its ease of use. However, a major concern is its lack of protecting the privacy of the user as the Ads are fetched but just not displayed. Furthermore, such ad blockers top the list of most popular Firefox extensions, with at least 18M installs [14] that fail in providing privacy.

Other systems that work on the network level like Alternate DNS [15] that is a DNS server which is capable of achieving the same results to the proposed system but as the DNS server is owned by a third party it allows for monitoring of the DNS requests being made. Ad-away[16] is another such system that works on the network level on a smartphone and uses a private DNS server but lacks ease of use and installation. It requires special root permissions from the system that a typical user won't be able to provide. It provides Ad blocking that can be considered private but it only blocks Ads on the device on which it is installed.

III. METHODOLOGY

A. *Adguard*

Adguard is a little process that runs on the Raspberry PI and blocks advertisements at the DNS level. When an ad is blocked it is actually prevented from being downloaded in the first place because the DNS query is intercepted. Since these Ads, images, videos, and sounds are not being downloaded, the network will perform in a better manner. Adguard blacklists such Ad server domains and redirects them to a blank IP (0.0.0.0), in turn the request doesn't reach the Ad servers and Ads are not fetched and thus not displayed. It puts itself between the device and an upstream DNS server and blocks out any requests to known Ad and tracking servers.

For a better understanding, domains trying to spread malware through false ‘Adobe Flash’ updates can be blocked at Network level. For devices such as Smart TVs which don’t use a browser but still feature Ads can also be blocked using Adguard. Most networks these days support IPv4 and IPv6. If an advertisement fails to deliver over IPv4, it can still be delivered via IPv6. Adguard can already use both protocols to block advertisements.

B. *Outcomes by Using Ad Blockers*

Ad Blockers are popular solutions in order to increase web privacy. By enabling Ad blocking software, not only advertisements, but also many tracking scripts for tracking user activities are also blocked. Results show significant differences among Ad-blockers regarding their filtering performance as affected by the applied configurations of Ad blockers and filtering lists [20]. Ad Blockers are often considered as web privacy tools that block third-party advertising. They are very effective at reducing third-party tracking [21]. Third party tracking scripts are classified into such categories as: ad trackers, analytics, beacons, social, and widgets [22]. Some propositions suggest that users should be charged for using Ad-blocking software to maintain the balance between Advertisers and users. because this kind of software acts as an intermediary platform [23]. Some researchers have designed systems for smart advertisement blocking. For instance, such systems can protect users’ privacy and preserve online advertising business [24]. Ad-blocking has a positive impact on user engagement with the Web. Firefox has revealed that two groups were tested: one using ad blocker, the other one not using them. For the first group, there was improvement in both active time spent in the browser and the number of pages viewed, while seeing no change in the number of searches for another group [25].

C. *Objectives of Adguard*

Adguard has the same objectives as most of the other popular Ad-blockers that work at the DNS level but with added security and more user control.

- Protect the users privacy by reducing the browser http cookies that track the user.
- Faster Page Load times as Ads are being blocked and also it comes with it’s own private DNS server.
- Added Security as DNS requests are resolved locally. The ISP cannot spy or track internet activity.
- Protect users from malvertising as any intrusive actions from the Ads, invisible links, auto-redirects, etc. are being blocked.
- Blocks Ads on any devices that do not allow for any modifications provided they are connected to the same network.
- Admin Control Panel allows the user to check the statistic of Adguard and the request made from the entire network
- Responsive and seamlessly speeds up everyday browsing by caching DNS queries

Unlike some of the Ad-blockers, Adguard is open source which ensures that the user is in complete control of his privacy and is lightweight and runs smoothly with minimal hardware and software requirements.

IV. System Design

A. *System Block Diagram*

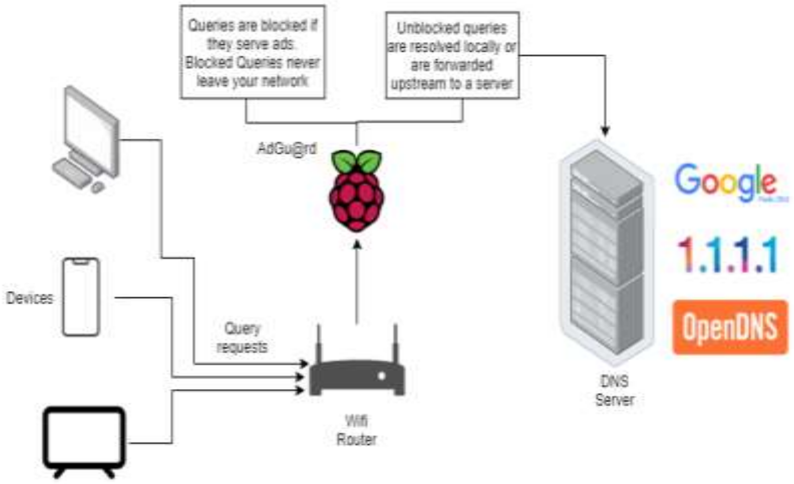

Queries are blocked if they serve ads. Blocked Queries never leave your network
Unblocked queries are resolved locally or are forwarded upstream to a server
AdGu@rd
Devices
Query requests
Wifi Router
DNS Server
Google
1.1.1.1
OpenDNS

Fig 1: System Block Diagram

All the devices connected to the router send requests to the router so that devices can display web pages from the Internet.The router uses a DNS (Domain Name System) to translate human readable domain names (for example, www.amazon.com) to machine readable IP addresses (for example 192.0.2.44). When using Adguard, devices that want to find out where a server is, the query is first sent to Adguard that acts as a DNS.

If the domain is not an Ad-serving domain (for example, google.com), the associated web page or data is displayed. If Adguard is unable to resolve the domain then a request is sent to an upstream (public) DNS server (OpenDNS, Google DNS). It passes through the router and out to the Internet and the webpage is displayed.

If the domain is an ad-serving domain (for example ads.google.com), Adguard responds to your device request and points to an address (0.0.0.0) that has a blank webpage. So in place of the Ad the blank web page is displayed.The request never leaves the network and remains private.

B. *Algorithm Based Flow Chart*

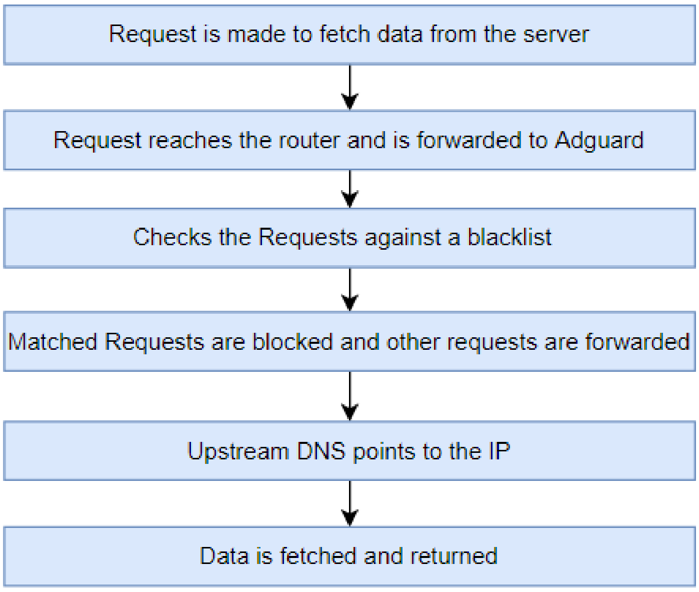

Request is made to fetch data from the server
Request reaches the router and is forwarded to Adguard
Checks the Requests against a blacklist
Matched Requests are blocked and other requests are forwarded
Upstream DNS points to the IP
Data is fetched and returned

Fig 2: FlowChart

When the user visits a website the device makes a request to fetch data from the server. The request is then forwarded to the router. Usually, the router resolves the request by itself or forwards it to an upstream DNS to resolve. Instead of resolving the request the router forwards the request to Adguard.

Adguard checks the requested domain name against a blacklist of domains where the blacklist contains domains that serve ads. The domains matched in the blacklist returns a non-existent IP address that results in ads not being fetched and inturn not displayed. Other domains are either resolved by the inbuilt DNS within Adguard or are forwarded to a configured upstream DNS server.

The request is resolved and the website is displayed free of advertisements. The system blocks Ads from being resolved which in-turn prevents tracking and unnecessary network usage.

V. Software Configuration

A. *Upgrading the Rpi*

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

With this done, reboot the pi using;

```
sudo reboot
```

B. *Setup Adguard*

Clone the repository on to the Raspberry Pi

```
git clone https://github.com/root-Akshay/AdGuard.git
```

If, git is not installed,

```
sudo apt install git
```

After the cloning completes,
Open the AdGuard Directory:

```
cd AdGuard
```

Convert the install file to an executable file:

```
sudo chmod +x install.sh
```

Run the file, it will install Adguard and other Required Dependencies.

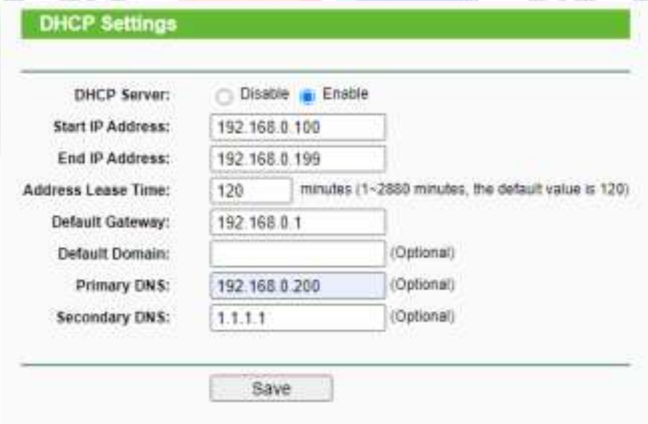
```
./install.sh
```

Reboot system after installation completes

C. *Configure Router*

Pi-hole needs to be the only DNS server because it intercepts queries and decides whether or not they should be blocked. If other DNS servers are configured, requests may be sent to the correct domain, bypassing Adguard completely.

- Log into the router's configuration page and find the DHCP/DNS settings.



DHCP Settings

DHCP Server:	○ Disable ● Enable
Start IP Address:	192.168.0.100
End IP Address:	192.168.0.199
Address Lease Time:	120 minutes (1~2880 minutes, the default value is 120)
Default Gateway:	192.168.0.1
Default Domain:	(Optional)
Primary DNS:	192.168.0.200 (Optional)
Secondary DNS:	1.1.1.1 (Optional)

Note: Make sure to adjust this setting under the LAN settings.

D. *Web Control Panel*

The install.sh script gives the RPi a static IP 192.168.0.200. To access the web Panel. Open any web browser and type the address in the search bar.

```
192.168.0.200:5000/login
```

Enter the Login Id and Password::

```
Id: Adguard
Pass::Adguard123
```

Login Page

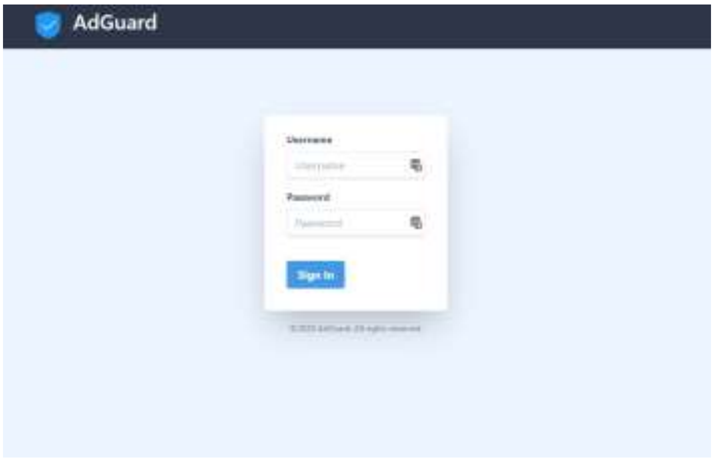
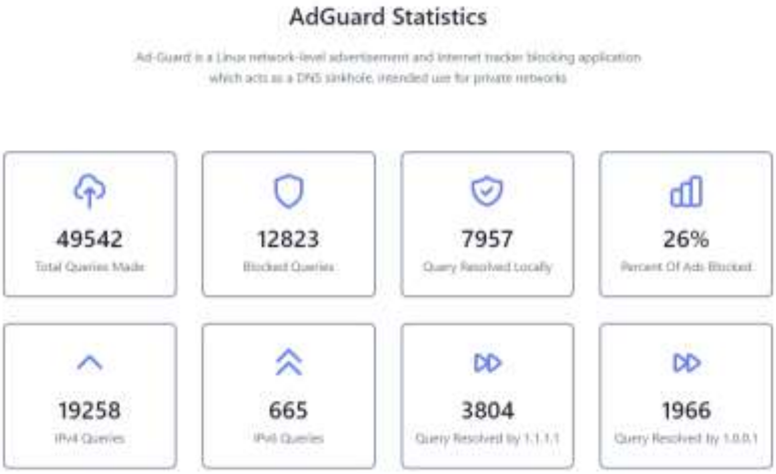


Fig 3 Login Page

The login page is a safeguard for the network admin/user to block unwanted access to the network details and the network activity of the user.



AdGuard Statistics

Ad-Guard is a Linux network-level advertisement and internet tracker blocking application which acts as a DNS sinkhole, intended use for private networks

49542 Total Queries Made

12823 Blocked Queries

7957 Query Resolved Locally

26% Percent Of Ads Blocked

19258 IPv4 Queries

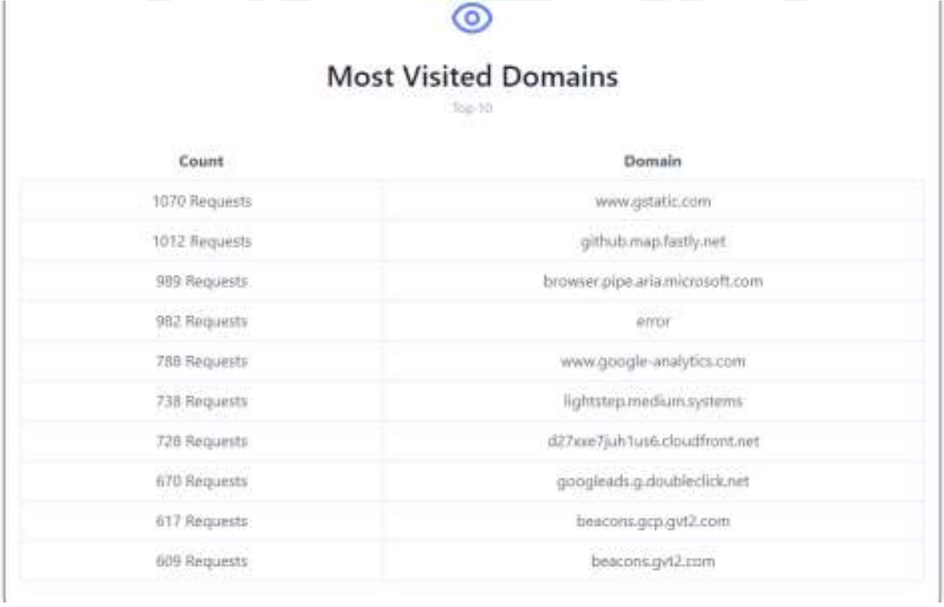
665 IPv6 Queries

3804 Query Resolved by 1.1.1.1

1966 Query Resolved by 1.0.0.1

Fig 4: Statistics page

After the login page the statistics related to the network will be accessible such as total number of queries made, the total number of queries blocked, queries handled by adguard itself, blocked Queries (Ads), IPv4 and IPv6 Queries and queries forwarded to upstream DNS as shown in Fig. 4.



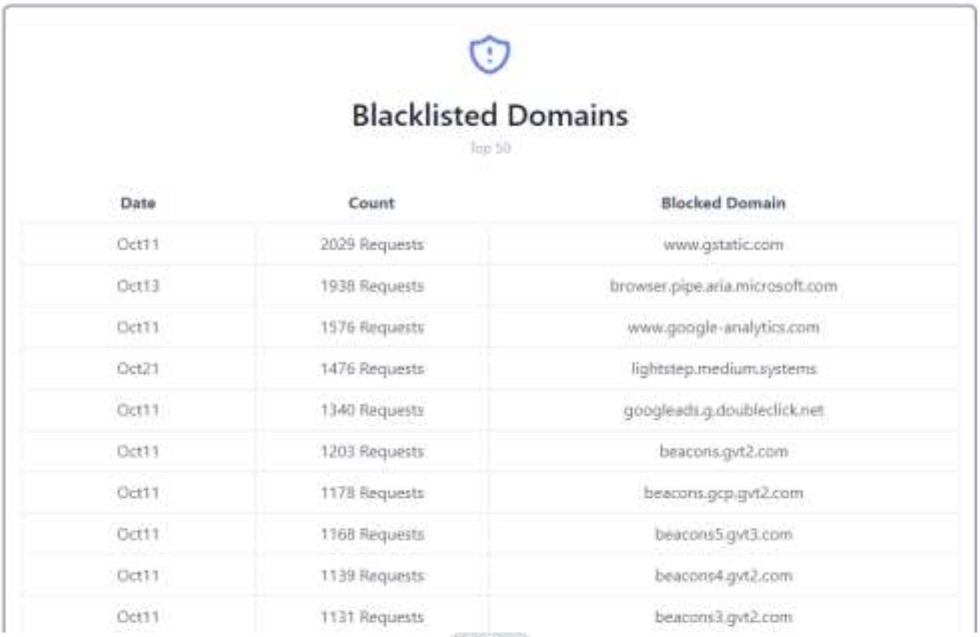
Most Visited Domains

Top 10

Count	Domain
1070 Requests	www.gstatic.com
1012 Requests	github.map.fastly.net
989 Requests	browser.pipe.aria.microsoft.com
982 Requests	error
788 Requests	www.google-analytics.com
738 Requests	lightstep.medium.systems
728 Requests	d27xxe7juh1us6.cloudfront.net
670 Requests	googleads.g.doubleclick.net
617 Requests	beacons.gcp.gvt2.com
609 Requests	beacons.gvt2.com

Fig 5: Statistics page 2

Fig 5 is an element of the statistics page showing the ten most visited domains or websites on the network. It ranks them from the most visited to the least. It displays the number of requests and the domain itself.



Blacklisted Domains

Top 50

Date	Count	Blocked Domain
Oct11	2029 Requests	www.gstatic.com
Oct13	1938 Requests	browser.pipe.aria.microsoft.com
Oct11	1576 Requests	www.google-analytics.com
Oct21	1476 Requests	lightstep.medium.systems
Oct11	1340 Requests	googleads.g.doubleclick.net
Oct11	1203 Requests	beacons.gvt2.com
Oct11	1178 Requests	beacons.gcp.gvt2.com
Oct11	1168 Requests	beacons5.gvt3.com
Oct11	1139 Requests	beacons4.gvt2.com
Oct11	1131 Requests	beacons3.gvt2.com

Fig 6: Statistics page 3
Note: The statistics page is only accessible after login

Fig 6 is another element of the statistics page which shows ten blocked domains or Ads that were requested on the network it ranks them from the most visited to the least. It displays the dates, requests and the Ad domains itself.

VI. CONCLUSIONS AND RECOMMENDATIONS

Ads are annoying and cause security issues and reduce network performance as well. The system developed and proposed in this paper tries to solve these issues, while being feasible, user friendly and open source. It is a simple yet effective way of filtering any DNS request on the network, but keep in mind that the user might need to tweak the black lists a bit to suit the personal browsing habits for any individual user.

There are some limitations to Adguard. One of the most important limitations is Ads that have the same domain as the legit traffic won't be blocked and hence the users are stuck with YouTube Ads even on AdGuard. However that can be improved by generating blacklists that are related to Ads on that domain. Furthermore, parental controls can be added so that parents can block certain devices in the network from accessing adult domains. Another point of recommendation is to allow the user to update blacklists and reboot the system from the web panel itself instead of using the console.

REFERENCES

[1] Giorgio Brajnik and Silvia Gabrielli. 2010. A review of online advertising effects on the user experience. International Journal of Human-Computer Interaction 26, 10 (2010), 971–997

[2] Chang-Hoan Cho and Hongsik John Cheon. 2004. Why do people avoid advertising on the internet? Journal of Advertising 33, 4 (2004), 89–97.

[3] Steven M Edwards, Hairong Li, and Joo-Hyun Lee. 2002. Forced exposure and psychological reactance: Antecedents and consequences of the perceived intrusiveness of pop-up Ads. Journal of Advertising 31, 3 (2002), 83–95.

[4] Avi Goldfarb and Catherine Tucker. 2011. Online display advertising: Targeting and obtrusiveness. Marketing Science 30, 3 (2011), 389–404

[5] Wen Li and Ziying Huang. 2016. The Research of Influence Factors of Online Behavioral Advertising Avoidance. American Journal of Industrial and Business Management 6, 09 (2016), 947.

[6] Catherine E. Tucker. 2014. Social Networks, Personalized Advertising, and Privacy Controls. Journal of Marketing Research 51, 5 (2014), 546–562.

[7] Kiran Garimella, Orestis Kostakis, and Michael Mathioudakis. 2017. Ad-blocking: A Study on Performance, Privacy and Counter-measures. In Proceedings of the 2017 ACM on Web Science Conference (WebSci '17). ACM, New York, NY, USA, 259–262

[8] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. 2015. Annoyed Users: Ads and Ad-Block Usage in the Wild. In Proceedings of the 2015 Internet Measurement Conference (IMC '15). ACM, New York, NY, USA, 93–106. https://doi.org/10.1145/ 2815675.2815705

[9] Gregor Aisch, Wilson Andrews, and Josh Keller. 2015. The Cost of Mobile Ads on 50 News Websites. (October 2015). Retrieved October 26, 2017 from https: //www.nytimes.com/interactive/2015/10/01/business/cost-of-mobile-Ads.html

[10] Catherine Dwyer and Ameet Kanguri. 2017. Malvertising - A Rising Threat To The Online Ecosystem. Journal of Information Systems Applied Research 10, 3 (2017), 29–37.

[11] Arthur Gervais, Alexandros Filios, Vincent Lenders, and Srdjan Capkun. 2017. Quantifying web adblocker privacy. In European Symposium on Research in Computer Security. Springer, 21–42.

[12] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising. In Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS ’12). ACM, New York, NY, USA, 674–686. https://doi.org/10.1145/ 2382196.2382267

[13] Adblock Plus | The world's #1 free ad blocker

[14] Mozilla. 2017. Firefox Add-ons: Most Popular Extensions. (2017). Retrieved October 31, 2017 from https://addons.mozilla.org/en-US/firefox/search/?sort= users&type=extension

[15] Alternate DNS - Ad Blocking DNS Server

[16] AdAway

[17] C. E. Wills and D. C. Uzunoglu, "What Ad Blockers Are (and Are Not) Doing," 2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, 2016, pp. 72-77, doi: 10.1109/HotWeb.2016.21.

[18] I. Jalbă, A. Olteanu and A. Drăghici, "Customized ad blocking," 2016 15th RoEduNet Conference: Networking in Education and Research, Bucharest, 2016, pp.1-5, doi: 10.1109/RoEduNet.2016.7753245.

[19] S. S. Hashmi, M. Ikram and M. A. Kaafar, "A Longitudinal Analysis of Online Ad-Blocking Blacklists," 2019 IEEE 44th LCN Symposium on Emerging Topics inNetworking (LCN Symposium), Osnabrück, Germany, 2019, pp. 158-165, doi:10.1109/LCNSymposium47956.2019.9000671.

[20] Sołtysik-Piorunkiewicz, Anna & Strzelecki, Artur & Abramek, Edyta. (2019). Evaluation of Adblock Software Usage. Complex Systems Informatics and Modeling Quarterly. 51-63. 10.7250/csimq.2019-21.04.

[21] A. Gervais, A. Filios, V. Lenders, and S. Capkun, “Quantifying Web Adblocker Privacy,” Lecture Notes in Computer Science, S. Foley, D. Gollmann, and E. Snekkenes, Eds. vol. 10493, Springer, pp. 21–42, 2017. Available: https://doi.org/10.1007/978-3-319-66399-9_2

[22] D. Ajdari, C. Hoofnagle, T. Stocksdale, and N. Good, “Web Privacy Tools and Their Effect on Tracking and User Experience on the Internet,” 2013. Available: https://www.truststc.org/education/reu/13/Papers/AjdariD_StocksdaleT_Paper.pdf

[23] C. E. Wills and D. C. Uzunoglu, “What Ad Blockers Are (and Are Not) Doing,” Proceedings of the 2016 Fourth IEEE Workshop on Hot Topics

in Web Systems and Technologies (HotWeb), pp. 72–77, 2016. Available: https://doi.org/10.1109/HotWeb.2016.21

[24] A. Ray, H. Ghasemkhani, and K. Kannan, “Ad-Blockers, Advertisers and Internet: The Economic Implications of Ad-Blocker Platforms,” Proceedings of the ICIS 2017: Transforming Society with Digital Innovation, pp. 1–10, 2018. Available:https://aisel.aisnet.org/icis2017/EBusiness/Presentations/15/

[25] D. Sánchez and A. Viejo, “Privacy-preserving and advertising-friendly web surfing,” Comput. Commun., vol. 130, pp. 113–123, 2018. Available. https://doi.org/10.1016/j.comcom.2018.09.002

[26] B. Miroglio, D. Zeber, J. Kaye, and R. Weiss, “The Effect of Ad Blocking on User Engagement with the Web,” Proceedings of the 2018 World Wide Web Conference on World Wide Web – WWW ’18, pp. 813–821, 2018. Available: https://doi.org/10.1145/3178876.3186162