Understanding abusive web resources: characteristics and counter-measures of malicious web resources and cryptocurrency mining

Ha Dao
University of Information Technology, VNU-HCM

Johan Mazel
The National Cybersecurity Agency of France

Kensuke Fukuda
NII / Sokendai

ABSTRACT

Web security is a big concern in the current Internet; users may visit websites that automatically download malicious codes for leaking user's privacy information, or even mildly their web browser may help for someone's cryptomining.

In this paper, we analyze *abusive* web resources (i.e. malicious resources and cryptomining) crawled from the Alexa Top 150,000 sites.

We highlight the abusive web resources on Alexa ranking, TLD usage, website geolocation, and domain lifetime. Our results show that abusive resources are spread in the Alexa ranking, websites particularly generic Top Level Domain (TLD) and their recently registered domains. In addition, websites with malicious resources are mainly located in China while cryptomining is located in USA. We further evaluate possible counter-measures against abusive web resources. We observe that ad or privacy block lists are ineffective to block against malicious resources while coin-blocking lists are powerful enough to mitigate in-browser cryptomining. Our observations shed light on a little studied, yet important, aspect of abusive resources, and can help increase user awareness about the malicious resources and drive-by mining on web browsers.

CCS CONCEPTS

• **Security and privacy** → Security requirements; Security protocols;

KEYWORDS

Web security, in-browser mining, privacy

ACM Reference Format:
Ha Dao, Johan Mazel, and Kensuke Fukuda. 2018. Understanding abusive web resources: characteristics and counter-measures of malicious web resources and cryptocurrency mining. In *AINTEC '18: ASIAN INTERNET ENGINEERING CONFERENCE (AINTEC '18), November 12–14, 2018, Bangkok, Thailand.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3289166.3289174

AINTEC '18, November 12–14, 2018, Bangkok, Thailand

ACM ISBN 978-1-4503-6131-6/18/11…$15.00
https://doi.org/10.1145/3289166.3289174

1 INTRODUCTION

In an era of increasingly popular and fast-paced technology, the Internet has become an effective means of connecting with people from all over the world. On one hand, users have become accustomed to using the Internet as a substitute for traditional channels of information and communication. On the other hand, malicious actors leverage Internet pervasiveness by incorporating malicious resources on websites and inside advertisements [36]. The New York Times recently fell victim to this phenomenon by displaying a malicious pop-up on its homepage [1]. Moreover, browser-based cryptomining is an increasingly popular technique where websites embed JavaScript files that mine cryptocurrency without users' permission or knowledge [15, 24]. We hereafter name malicious resources and cryptomining-related resources as *abusive web resources*. Cryptomining may exist in the form of malicious software installed on victim's computers, a browser-based javascript miner or a pop-under window that can continue to run even if the browser is closed [41].

Some work previously analyzed malicious resources and in-browser cryptocurrency [15, 26, 36, 50]. Provos et al. [36] provide a detailed study of drive-by downloads on the Internet, including advertisement-related ones. Li et al. [26] focus on advertisement-related malicious resources (also called malvertisement) and propose a detection method that improves the previous results [36]. Zarras et al. [50] present an in-depth analysis of malvertisement. Eskandari et al. [15] provide the first analysis of the cryptomining deployment. Papadopoulos et al.[32] compare the profitability of in-browser cryptomining and advertising. Rüth et al.[37] analyze the popularity of in-browser cryptomining. Konoth et al. [21]

measure the existence of in-browser cryptomining and propose a detection method.

Our work provides new insights on abusive web resources, and evaluates previously unaddressed available counter-measures.

In this paper, we study the ecosystem of abusive resources (malicious resources and in-browser cryptomining) in Alexa Top 150,000 websites with Google SafeBrowsing API to assess their maliciousness and a custom list to detect cryptomining. We make the following contributions:

- We detect two types of abusive web resources (malicious resources and in-browser cryptocurrency mining) in the Alexa Top 150,000 websites, and compare with past results [26, 36, 50]. We further highlight similarity and difference of the two types in terms of TLD usage, geolocation and position of abusive resources in the Alexa Top 150k ranking.
- We evaluate blocking list-based countermeasures as a mean to shield users from abusive resources. We analyze protection ability from both malicious resources and cryptomining.

2 BACKGROUND

In the following sections we briefly describe abusive resources (malicious resources and in-browser cryptomining).

2.1 Term definitions

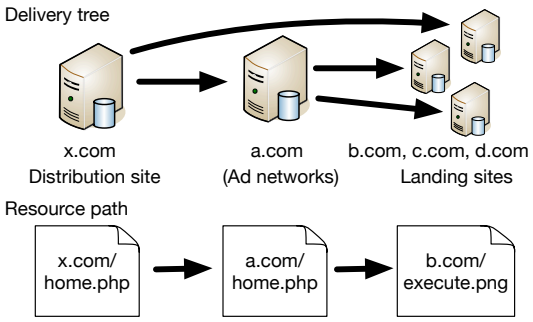

Figure 1: Delivery tree and resource path

We first define some terms we use throughout this paper (see also Figure 1):

- *Abusive web resources* include both malicious web resources and in-browser cryptocurrency mining.
- *Landing website* refers to a crawled website. In our data, we have 150,000 landing websites (see Section 4.1).
- *Delivery tree* is composed of the landing website and all URLs that the browser visits until it contacts the distribution website (website distributes abusive resources). We call this particular website the root of the delivery tree [36].
- *Abusive resource path* is a path in the delivery tree from the landing website to a contacted URL (including redirections).

2.2 Malicious web resources

Malicious web resources can execute various kinds of actions that impact users. These resources are located on distribution websites. Users may be infected by such malicious web resources, when browsing websites. The goal may be to help performing distributed denial-of-service (DDoS) attacks [33], exploitation through drive-by download attacks [36] or user's privacy leakage by phishing.

2.3 Cryptomining

In-browser cryptomining is a method using scripting language such as JavaScript or WebAssembly to mine cryptocurrencies, in user's browser.

We are aware of two cases of transparent in-browser cryptomining: the Salon website [44], and the Australian branch of UNICEF [27] have both launched pages that let users donate through in-browser cryptomining.

Additionally, Coinhive offers an alternative API for advertisements to explicitly asks for consent from users before any mining activity. However, the opt-in version of the Coinhive API usage remains marginal, only 1.3% of total websites hosting in-browser cryptomining [40].

The exploitation of cryptocurrency on the browser is increasingly popular [15, 24, 37]. Websites can utilize the power of machine's CPU of users to mine coins without permission.

The proportion of average power consumption that these miners account for is between from 10% to 40% of the CPU ability [24]. Some users reported that the CPU usage reached 100% when surfing websites containing these mining scripts[24]. We thus consider these scripts as abusive resources.

3 RELATED WORK

Provos et al. [36] provide the seminal contribution in the field of drive-by downloads on the Internet. They developed a large scale data collection infrastructure that continuously detects and monitors the behavior of websites that perpetrate drive-by downloads. Google released that work as SafeBrowsing [17]. SafeBrowsing is used inside web browsers such as Chrome and Firefox to identify malicious websites across the web and notify users of potential harm.

Online marketing and advertising can be used for illegal purposes such as malware distribution, phishing, or click fraud [26]. Li et al. [26] focused on ad serving and detection of malvertisement activity. They used Google Safe-Browsing API [17] and Microsoft Forefront 2010 to detect malvertising. Zarras et al. [50] addressed ecosystems around malicious ads and their status, to analyze malvertisement and how

these reach end users. They used Wepawet [10], malware and phishing blacklists [22], and VirusTotal [45] to label malicious behavior. This study extracted advertisements from a variety of websites and used a number of oracles to classify the advertisements as malicious or legitimate, then analyzed them. Stone et al.[43] addressed click fraud in online advertising. Furthermore, Xing et al. [49] showed that browser extensions that use advertisements as their monetization strategy often facilitate the deployment of malware advertisement.

Eskandari et al.[15] inspected the recent trend of in-browser cryptocurrency mining. Rüth et al.[37] analyzed the prevalence of browser-based mining and presented a new approach to identify mining websites. Konoth et al. [21] measured the occurrence of in-browser cryptomining and proposed a detection method based on the characteristics of cryptomining code. Papadopoulos et al. [32] compared the profitability of in-browser cryptomining and advertising by analyzing user's system resources when user access websites hosting cryptomining. Our work is close to [15, 26, 36, 50] but improves the state of the art along three axes: target websites, target resources (both malicious resources and in-browser cryptomining), and evaluation of counter-measures. *First*, we crawl the 150,000 highest-ranked websites by Alexa [4] while Li et al. [26], Eskandari et al. [15] and Zaras et al. [50] respectively use the Alexa's Top 90,000 Web sites, over 30,000 websites. *Second*, we address websites hosting malicious resources and in-browser cryptomining, compare their characteristics, and provide new insights about both of them while [26] and [50] just focus on malicious advertisement, [36] analyze drive-by-download and [15] address the recent trend of in-browser mining of cryptocurrencies. *Finally*, we here evaluate blocking list-based countermeasures as a mean to shield users from both malicious resources and cryptomining.

4 METHODOLOGY

In this section, we present the methodology we used to generate and evaluate a large corpus of abusive resources. Our process includes three steps. First, we crawl data from Alexa top websites (Section 4.1). Second, we describe an approach to reconstruct malicious resource path (Section 4.2). Third, we classify the resources as malicious or in-browser cryptomining (Section 4.3).

4.1 Data collection

We crawl the home pages of Alexa [4] Top 150,000 websites in April 2018, using IP addresses located in Japan. Figure 2 shows the process flow. We use OpenWPM [14], a open-source web privacy measurement framework written in Python that relies on Selenium for browser automation.

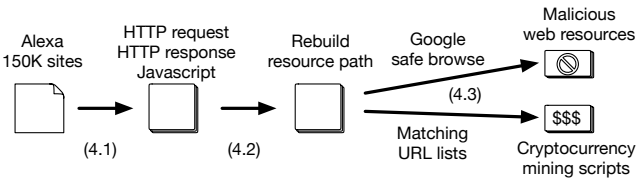

Figure 2: Processing flow

OpenWPM supports Firefox and some of its extensions out-of-the-box. We here use Firefox configuration without any extension to obtain 150,000 websites and 12,796,779 unique URLs during the data collection.

4.2 Abusive resource path reconstruction

Here we present the construction of abusive resource path in our data. We extract the edges connecting URLs by inspecting the HTTP referer field from successive HTTP requests the browser makes after visiting the landing page.

However, in many cases, the referer headers are not sufficient to extract the exact path. For example, when the browser redirection results from an HTTP 3XX Redirection (301 Moved Permanently, 302 Found, 303 See Other, 307 Temporary Redirect) or an external script, we cannot extract the referer from HTTP requests. In these cases, the referrer points to the landing page and one cannot identify the external HTTP redirection or script files.

For HTTP redirections, we get all URLs on location field because the location response header field contains the URL one is redirected to. After obtaining location URL, we set the referer of that location URL as the URL of the previous HTTP request.

Regarding script files, to identify sources of each URL, we extract JavaScript domains and extract URLs inside the these JavaScript code. Then we process the edit distances between a requested URL and extracted URLs in the JavaScript code and the script domains. If all distances are larger than the size difference between the URL length and the smallest script domain length, it means that the URL request does not completely contain any script domain and other URL found inside the scripts. In that case, the landing website will be considered as the referer. If the condition above is not verified, URLs found among script domains and URLs found in scripts with the smallest distance will become the referer of this URL request. This approach extends previous work by Li et al. [26] and Provos et al. [36].

4.3 Abusive resources detection

We use the Google Safe-Browsing API [17, 36] to detect malicious resources among crawled URLs. Google Safe browsing

runs on more than three billion devices and lets client applications check URLs against Google's constantly updated lists of unsafe web resources. If any URL is flagged by Safe-Browsing, we assume that it is a malicious URL (contains malware, unwanted software and potentially harmful application threat types). Our findings are presented in Table 1. Overall, we detected more than three thousand malicious URLs hosted on 123 crawled websites.

	Websites	URLs
Crawled	150,000	12,796,779
Malicious	123	3,300

Table 1: Summary of collected data

We identify browser-based mining by searching for mining scripts such as *coinhive.min.js* within JavaScript calls in the website page. Target scripts are listed in Table 2. Coinhive is the dominant website offering in-browser mining (80.7%).

Script	String in URL	Num
Coinhive [19]	'coinhive.min.js', 'cnhv.co'	180
JSEcoin [28]	'load.jsecoin.com'	23
Coinpot [9]	'coinpot.co'	5
CoinIMP [8]	'hashing.win', 'Coinimp'	3
Webmine [46]	'Webmine'	1
ProjectPoi (PPoi) [13]	'projectpoi.min'	0
AFMiner [3]	'afminer.com/code/miner.php'	0
Papoto	'papoto.com/lib/papoto.js'	0
CryptoNoter	'minercry.pt/processor.js '	0
Crypto-Loot [11]	'CryptoLoot.Anonymous'	0
Adless [2]	'adless.io'	0
Ppoi [35]	'ppoi.org'	0
Coin-have [7]	'coin-have.com'	0
Monerise [30]	'apin.monerise.com' 'monerise_builder'	0
deepMiner [12]	'deepMiner.js" 'deepMiner.Anonymous'	0
NFWebMiner [31]	'nfwebminer.com/lib/','NFMiner'	0
Mineralt [29]	'mineralt.io'	0
Minr [29]	'minr.pw', 'cdn.static-cnt.bid' 'abc.pema.cl','metrika.ron.si', 'cdn.rove.cl','host.d-ns.ga', 'static.hk.rs','hallaert.online', 'cnt.statistic.date', 'st.kjli.fi'	0

Table 2: Cryptomining detection results

5 RESULTS

This section presents the findings of our study based upon the methodology presented in section 4.

5.1 Malicious resources path

We examine abusive resource paths to understand their characteristics. We provide a breakdown of the total number of unique infected malicious URLs we discovered in Table 3. We find that the vast majority of abusive resources are served by own landing websites by 91%, and third-party domains to less than 9%. One can consider that websites containing malicious resources were added to the Internet by publishers' website objective.

Origin of malicious URL	Numbers
Total	3,300
First-party (crawled website)	3,016
Third-party	284

Table 3: Summary of malicious resource paths

5.2 Landing websites containing abusive resources

In this section, we focus on the characteristics of websites containing malicious resource.

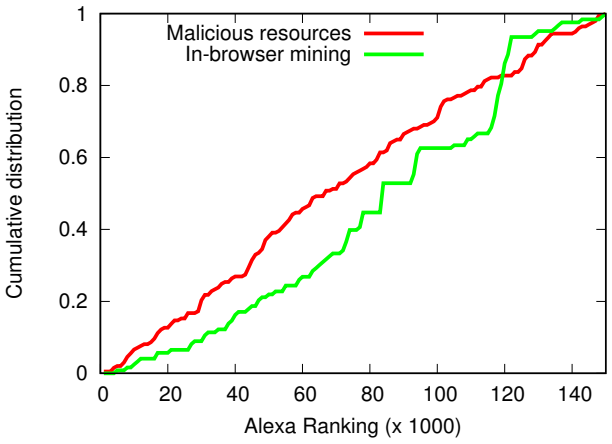

Figure 3: ECDF of the Alexa ranking of websites containing malicious resources and in-browser cryptomining.

5.2.1 Domain ranking. Figure 3 presents an ECDF of the Alexa ranking of websites containing malicious resources and in-browser cryptomining. Websites containing malicious resources and using cryptomining are spread in the Alexa ranking.

The red plot (malicious resources) is close to a straight line, meaning that a probability to contain malicious resources is almost similar among ranks. In contrast, the green plot (in-browser cryptomining) has a sudden increase around

120K, demonstrating specific characteristics. Indeed, most websites here belong to Information Technology category (see also Sec.5.2.2). Overall, popular websites still pose risks to end users.

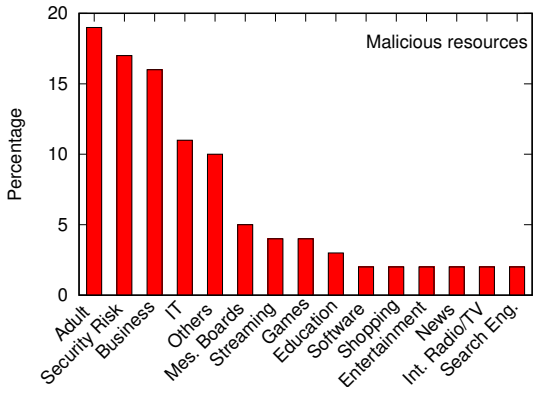

Figure 4: Breakdown of website containing malicious resources by website categories.

5.2.2 Domain category. We then illustrate a breakdown of websites containing abusive resources by website category in Figure 4 and Figure 5. We used FortiGuard Web Filtering [23] in July 2018 for the classification. Figure 4 illustrates that malicious resources are mainly located on adult websites (19% in total). This fact is consistent with a previous study [47]. Besides, 17% of malicious websites are defined as Security risk category (classified as malicious domains by FortiGuard). The percentages for Business and Information Technology are 16% and 11%, respectively.

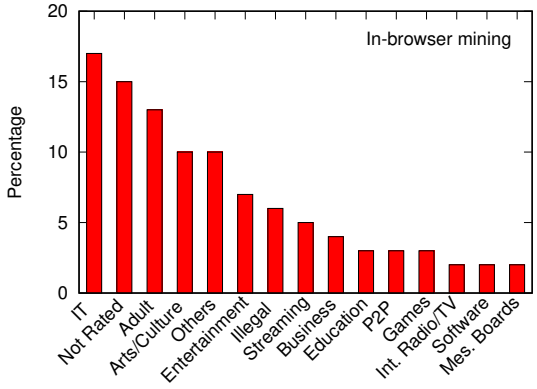

Figure 5: Breakdown of websites containing in-browser cryptomining by website categories.

As for in-browser mining (Figure 5), in-browser mining is mainly observed on Information technology websites. Beyond the IT category, 13%, 10%, and 7% of malicious websites belong to Adult, Arts and Culture and Entertainment category, respectively. Interestingly, 15% of the sites belong to security risk category. According to FortiGuard classification, security risk are domains that are newly configured or newly active, but not necessarily newly registered or domains that were very recently registered. It means that these 15% of domain names hosting in-browser cryptomining recently appeared on the Internet.

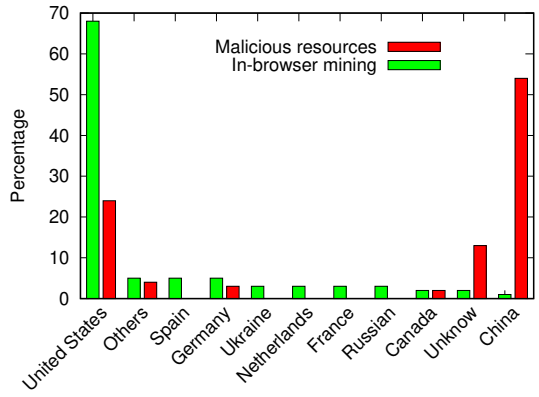

Figure 6: Breakdown of websites containing malicious resource and in-browser cryptomining by Country.

5.2.3 Domain country. Figure 6 represents the breakdown of websites containing malicious resource and in-browser cryptomining by country. We use Geoip to determine website locations. Observed websites with malicious resources are mainly located in China. These websites accounts for just over 54% of the total, in contrast the other countries have significantly lower percentages. The percentage for United States is nearly 24%. As for browser-based cryptomining, most observed websites with cryptomining are located in USA. The percentage for this nation is nearly 65%. Other noticeable countries such as Spain and Germany contain nearly or just over 5%.

5.2.4 Domain lifetime and registration year: An ECDF of the domain lifetime of websites containing malicious resources and in-browser cryptomining is shown in Figure 7. The registration times of domains with malicious resources differ significantly from the remaining ones. We can see that more than 26% websites contain malicious resources (red curve) and 47% websites contain in-browser cryptomining (green curve) expire within one year or two years of registration. In addition, their registration times are much later than the benign one (blue curve). Since malicious domains usually get blacklisted quickly, attackers may have no incentives to register long-living domains. In contrast, normal websites

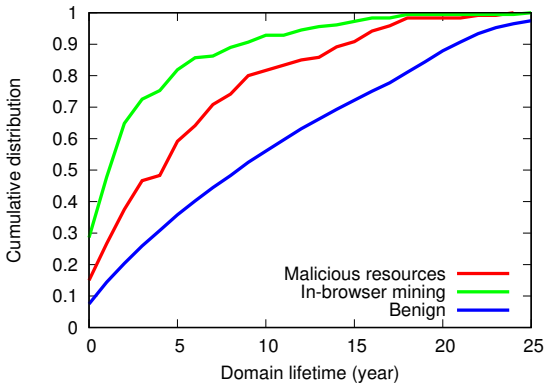

Figure 7: ECDF of the domain lifetime of websites containing malicious resources and in-browser cryptomining.

have much longer expiration dates as their business is expected to operate for years. This is consistent with Li et al. [26] for malicious resources.

These findings also indicate that websites hosting in-browser cryptomining have similar characteristic regarding domain lifetime, but the websites with in-browser cryptomining are further younger than websites containing malicious resources.

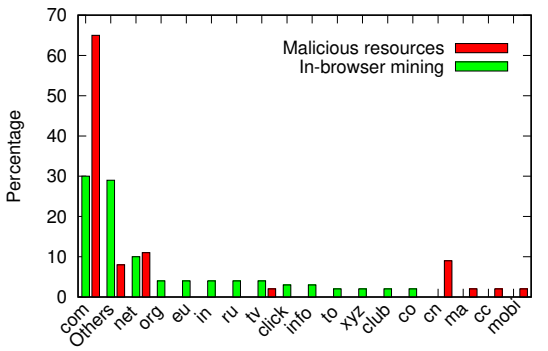

Figure 8: Breakdown of websites containing malicious resources and in-browser cryptomining by Top Level Domain(TLD)

5.2.5 Top Level Domain. Figure 8 represents the breakdown of top-level domains that serve malicious resources and in-browser cryptomining. Websites names ending in generic Top level Domain (TLD) like .com and .net host the majority of abusive resources on the Internet (76% of malicious resources and 40% of in-browser cryptomining).

5.3 Evaluate counter-measures

Some malicious resources are delivered by advertisements [26, 36, 50]. In our data (Table 3), we saw that third parties distributed malicious resources. Existing blocking list target advertisement [18, 25, 34], privacy, [5], and cryptocurrency mining [38, 48]. These lists may thus be able to block abusive resources provided by third parties. Therefore, we intend to assess their efficiency as counter-measures. We used adblockparser [39], a python library to directly match blocking list with collected data.

In this section, we present the evaluation of malicious resources counter-measures and in-browser cryptomining counter-measures.

5.3.1 Malicious resources counter-measures. Here, we evaluate the effectiveness of state-of-the-art block lists against malicious resources.

We evaluate following blocking lists: EasyList [34], EasyPrivacylist [5], Japanese ad-blocking list [18] and a filter set similar to ublock Origin [25] (EasyList, EasyPrivacy, Peter Lowe's Adservers, Malware domains). EasyList is a filter list that removes ads from international webpages. It is also used by default in the popular browser plugin Adblock plus [16] to block ads. EasyPrivacy [5] is an optional supplementary filter list that removes trackers from websites. As our crawl is performed from Japan (see Section 4.1), Japanese advertisers may use our location to provide customized ads. We thus also use the Japanese ad-blocking list suggested to Internet users in Japan by AdBlock Plus: ABPJapanese [18]. UBlock Origin is a general purpose blocker. It can also understand the syntax used by the ad-blocker AdBlock Plus.

Filters	Number	Percentage
Total	3,300	100%
Easylist	31	0.93%
EasyPrivacy list	3	0.09%
ABP Japanese filters	26	0.79%
uBlock Origin	34	1.03%
All (combined)	53	1.61%

Table 4: Ad-related URL filtering

We inspect individual URLs using these well-known filter lists in August 2018. The result of this experiment is shown in Table 4. Because of the vantage point (i.e., from Japan), we discovered 26 malicious URLs are flagged as ads directly from Japan with ABP Japanese Filters. Furthermore, we found that 51 malicious URLs have been flagged by all of these filters. This represents 1.55% of all malicious URLs. It is not surprising results because studied filters are used for ad-blocking or privacy protection.

Filters	Number	Percentage
Total	212	100%
NoCoin list	203	95.75%
Filter list of MinerBlock	207	97.64%

Table 5: No coin filtering

5.3.2 Cryptomining counter-measures. There are several common coin-blocking blocklists, such as NoCoin extension filter list[20], CoinBlock extension filter list [6], NoMiners extension filter list [42], Nocoin filter list[38] or MinerBlock extension filter list[48]. However, NoCoin, CoinBlock, and NoMiners extension filter lists have not been updated in the last few months. We thus analyze the Nocoin filter list and MinerBlock extension filter list. Nocoin filter list [38] contains patterns to block in-browser mining using common ad blockers. MinerBlock [48] is a browser extension to block in-browser cryptomining. It is available for Google Chrome, Mozilla Firefox, and Opera. We used both filters published in April 2018.

The percentage of mining script URL blocked by Nocoin was 95.75% and by MinerBlock was 97.64%. NoCoin missed 5 Coinpot, 3 CoinImp, and 1 Webmine scripts. NoCoin version 72 was updated twice: on June 5th 2018 to block CoinImp and on June 7th 2018 to block Webmine. NoCoin version 87 was then updated on August 4th 2018 to block Coinpot. While MinerBlock missed 5 Coinpot and it still has not been updated at the time of submission (September 19th 2018).

6 CONCLUSION

In this work, we characterized malicious resources and in-browser cryptomining for the Top Alexa 150,000 websites. We highlighted various aspects of websites containing abusive resources. We discovered that websites containing abusive resources are spread in the Alexa rankings. This means that users may be exposed to abusive resources on major websites. Furthermore, we highlighted that main players using two types of abusive resources are different. Finally, we evaluated the effectiveness of blocking lists regarding blocking of malicious resources and cryptomining scripts. We demonstrated that blocking lists do not effectively deal with malicious resources, but coin-blocking filter lists hold a great promise to more effectively mitigate in-browser cryptomining. However, missing domains from the NoCoin list proves that blocking lists need to be updated continuously to protect users from in-browser cryptomining.

As future work, we intend to add longitudinal measurements to improve our data collection process. We also want to provide detailed statistics on resource paths, and use additional oracles for abusive resources. Base on our analysis in Section 5.3, we intend to develop a counter-measure against abusive resources on the Internet.

ACKNOWLEDGEMENTS

Ha Dao thanks the NII international internship program.

REFERENCES

[1] 2018. New York Times pushes Fake AV malvertisement. Retrieved July 22, 2018 from http://countermeasures.trendmicro.eu/new-york-times-pushes-fake-av-malvertisement/
[2] adless.io. [n. d.]. adless.io. Retrieved August 22, 2018 from https://adless.io
[3] AFMiner. [n. d.]. AFMiner. Retrieved August 22, 2018 from https://afminer.com/
[4] Inc. Alexa Internet. 1996. The top 500 sites on the web. Retrieved July 22, 2018 from https://www.alexa.com/topsites
[5] The EasyList authors. 2006. EasyPrivacy. Retrieved July 22, 2018 from https://easylist.to/easylist/easyprivacy.txt
[6] Bechsen. [n. d.]. CoinBlock. Retrieved August 22, 2018 from https://addons.mozilla.org/en-US/firefox/addon/coinblock/?src=recommended
[7] coin have.com. [n. d.]. coin-have.com. Retrieved August 22, 2018 from https://coin-have.com/
[8] CoinIMP.com. [n. d.]. CoinIMP. Retrieved August 22, 2018 from https://www.coinimp.com/
[9] CoinPot. [n. d.]. CoinPot | Cryptocurrency microwallet. Retrieved August 22, 2018 from http://coinpot.co/
[10] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2010. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In *WWW 2010*. 281–290.
[11] Crypto-Loot. [n. d.]. Crypto-Loot. Retrieved August 22, 2018 from https://crypto-loot.com/
[12] deepwn. [n. d.]. deepMiner miner proxy. Retrieved August 22, 2018 from https://github.com/deepwn/deepMiner
[13] EGOIST. [n. d.]. ProjectPoi. Retrieved August 22, 2018 from https://poi.js.org/
[14] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *CCS 2016*.
[15] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. 2018. A first look at browser-based Cryptojacking. *arXiv preprint arXiv:1803.02887*.
[16] eyeo GmbH. 2005. Adblock plus. Retrieved July 22, 2018 from https://adblockplus.org/
[17] Google. 2007. Safe Browsing APIs. Retrieved July 22, 2018 from https://developers.google.com/safe-browsing/v4/
[18] k2japan. 2014. ABP Japanese filters. Retrieved July 22, 2018 from https://raw.githubusercontent.com/k2jp/abp-japanese-filters/master/abpjf.txt
[19] Keraf. [n. d.]. Coinhive – Monero JavaScript Mining. Retrieved July 22, 2018 from https://coinhive.com/
[20] Keraf. 2018. No Coin - Block miners on the web! Retrieved July 22, 2018 from https://chrome.google.com/webstore/detail/no-coin-block-miners-on-t/gojamcfopckidlocpkbelmpjcgmbgjcl
[21] Radhesh Krishnan Konoth, Emanuele Vineti, Martina Lindorfer Veelasha Moonsamy, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. 2018. MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense. In *CCS 2018*.
[22] Marc Kührer and Thorsten Holz. 2012. An empirical analysis of malware blacklists. *PIK-Praxis der Informationsverarbeitung und Kommunikation* 35, 1 (2012), 11–16.

[23] FortiGuard Labs. 2005. FortiGuard Web Filtering. Retrieved July 22, 2018 from https://fortiguard.com/webfilter

[24] Hon Lau. 2017. Browser-Based Cryptocurrency Mining Makes Unexpected Return from the Dead. Retrieved July 22, 2018 from https://www.symantec.com/blogs/threat-intelligence/browser-mining-cryptocurrency

[25] Hon Lau. 2017. uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. Retrieved July 22, 2018 from https://github.com/gorhill/uBlock

[26] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing your enemy: understanding and detecting malicious web advertising. In *CCS 2012*. 674–686.

[27] Shannon Liao. 2018. UNICEF wants you to mine cryptocurrency for charity. Retrieved July 22, 2018 from https://www.theverge.com/2018/4/30/17303624/unicef-mining-cryptocurrency-charity-monero/

[28] JSEcoin Ltd. [n. d.]. JSEcoin: Digital Currency - Designed for the web. Retrieved July 22, 2018 from https://jsecoin.com/

[29] Mineralt. [n. d.]. Mineralt. Retrieved August 22, 2018 from https://mineralt.io/

[30] Monerise. [n. d.]. Monerise: Web-miner (Content Monetisation Program) as featured on CoinReviews.IO. Retrieved August 22, 2018 from https://bitcointalk.org/index.php?topic=2720304.0

[31] NFWebMiner. [n. d.]. NFWebMiner. Retrieved August 22, 2018 from http://nfwebminer.com/

[32] Panagiotis Papadopoulos, Panagiotis Ilia, and Evangelos P Markatos. 2018. Truth in Web Mining: Measuring the Profitability and Cost of Cryptominers as a Web Monetization Model. *arXiv preprint arXiv:1806.01994* (2018).

[33] Giancarlo Pellegrino, Christian Rossow, Fabrice J Ryba, Thomas C Schmidt, and Matthias Wählisch. 2015. Cashing Out the Great Cannon? On Browser-Based DDoS Attacks and Economics.. In *WOOT 2015*.

[34] R. Petnel. 2005. The official easylist web site. Retrieved July 22, 2018 from https://easylist.to/

[35] ppoi.org. [n. d.]. ppoi.org. Retrieved August 22, 2018 from https://ppoi.org

[36] Niels Provos, Panayiotis Mavrommatis, Moheeb Abu Rajab, and Fabian Monrose. 2008. All your iframes point to us. In *USENIX Security Symposium. USENIX*. 1–16.

[37] Jan Rüth, Torsten Zimmermann, Konrad Wolsing, and Oliver Hohlfeld. 2018. Digging into Browser-based Crypto Mining. In *IMC 2018*.

[38] Hosh Sadiq. 2017. Block lists to prevent JavaScript miners. Retrieved July 22, 2018 from https://github.com/hoshsadiq/adblock-nocoin-list

[39] Scrapinghub. 2017. Python parser for Adblock Plus filters. Retrieved August 22, 2018 from https://github.com/scrapinghub/adblockparser

[40] Jérôme Segura. [n. d.]. The state of malicious cryptomining. Retrieved July 22, 2018 from https://blog.malwarebytes.com/cybercrime/2018/02/state-malicious-cryptomining/

[41] Jérôme Segura. 2017. Persistent drive-by cryptomining coming to a browser near your. Retrieved July 22, 2018 from https://blog.malwarebytes.com/cybercrime/2017/11/persistent-drive-by-cryptomining-coming-to-a-browser-near-you/

[42] Shaa3. [n. d.]. nominers. Retrieved August 22, 2018 from https://github.com/Shaa3/nominers

[43] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. 2011. Understanding fraudulent activities in online ad exchanges. In *IMC 2011*. 279–294.

[44] Salon TV. 2018. FAQ: What happens when I choose to "Suppress AdsâĂİ on Salon? Retrieved July 22, 2018 from https://www.salon.com/about/faq-what-happens-when-i-choose-to-suppress-ads-on-salon/

[45] virustotal.com. [n. d.]. virustotal.com. Retrieved August 22, 2018 from https://www.virustotal.com

[46] Webmine.cz. [n. d.]. Webmine. Retrieved August 22, 2018 from https://webmine.cz/

[47] Gilbert Wondracek, Thorsten Holz, Christian Platzer, Engin Kirda, and Christopher Kruegel. 2010. Is the Internet for Porn? An Insight Into the Online Adult Industry. In *WEIS 2010*.

[48] xd4rker. [n. d.]. MinerBlock. Retrieved August 22, 2018 from https://github.com/xd4rker/MinerBlock

[49] Xinyu Xing, Wei Meng, Byoungyoung Lee, Udi Weinsberg, Anmol Sheth, Roberto Perdisci, and Wenke Lee. 2015. Understanding malvertising through ad-injecting browser extensions. In *WWW 2015*. 1286–1295.

[50] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The dark alleys of madison avenue: Understanding malicious advertisements. In *IMC 2014*. 373–380.