2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS) | 978-1-6654-3485-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/DAPPS52256.2021.00008

In-Browser Cryptomining for Good: An Untold Story

Ege Tekiner*§, Abbas Acar*§, A. Selcuk Uluagac*, Engin Kirda†, and Ali Aydin Selcuk‡
*Florida International University, Email:{etekiner, aacar001, suluagac}@fiu.edu
†Northeastern University, Email: {ek}@ccs.neu.edu
‡TOBB University of Economics and Technology, Email: {aselcuk}@etu.edu.tr

***Abstract*—In-browser cryptomining uses the computational power of a website's visitors to mine cryptocurrency, i.e., to create new coins. With the rise of ready-to-use mining scripts distributed by service providers (e.g., Coinhive), it has become trivial to turn a website into a cryptominer by copying and pasting the mining script. Both legitimate webpage owners who want to raise an extra revenue under users' explicit consent and malicious actors who wish to exploit the computational power of the users' computers without their consent have started to utilize this emerging paradigm of cryptocurrency operations. In-browser cryptomining, though mostly abused by malicious actors in practice, is indeed a promising funding model that can be utilized by website owners, publishers, or non-profit organizations for legitimate business purposes, such as to collect revenue or donations for humanitarian projects, inter alia. However, our analysis in this paper shows that in practice, regardless of their being legitimate or not, all in-browser mining scripts are treated the same as malicious cryptomining samples (aka *cryptojacking*) and blacklisted by browser extensions or antivirus programs. Indeed, there is a need for a better understanding of the in-browser cryptomining ecosystem. Hence, in this paper, we present an in-depth empirical analysis of in-browser cryptomining processes, focusing on the samples explicitly asking for user consent, which we call *permissioned cryptomining*. To the best of our knowledge, this is the first study focusing on the permissioned cryptomining samples. For this, we created a dataset of 6269 unique websites containing cryptomining scripts in their source codes to characterize the in-browser cryptomining ecosystem by differentiating permissioned and permissionless cryptomining samples. We believe that (1) this paper is the first attempt showing that permissioned in-browser cryptomining could be a legitimate and viable monetization tool if implemented responsibly and without interrupting the user, and (2) this paper will catalyze the widespread adoption of legitimate cryptomining with user consent and awareness.**

***Keywords*—Cryptojacking, cryptomining, cryptocurrency, bitcoin, monero, coinhive**

I. Introduction

Blockchain technologies gained enormous popularity in the last decade. People started looking for cryptocurrency developments and different implementation ideas for various business types supported by Decentralized Applications (DApps). Some systems purposed to put up some extra money by merging existing services with blockchain-based technologies. One of them is in-browser cryptocurrency mining (cryptomining) technology. In-browser cryptomining allows websites to use their visitors' (i.e., clients') computational resources to mine cryptocurrency and to make revenue on behalf of the owner of a webpage. On the client side, in-browser mining is originally proposed as an alternative revenue mechanism to advertisements by the website owners, which in return, offer premium content or add-free surfing to its users. However, with the profitability of bitcoin and alternative cryptocurrencies, the attackers have started hijacking some popular websites [1] to embed cryptomining scripts (aka *cryptojacking*) and start mining without the knowledge and explicit consent of the users [2]. As of this writing, 32.3 million total cryptojacking attacks have been registered during the first half of 2020 [3]. Such malicious cryptomining scripts were even found on some government websites around the world [4]. Although in-browser cryptomining is instrumental for legitimate business purposes, the malicious or illegitimate usage is also gaining traction and is not unknown. There exist some mitigation techniques that can be used by the users in practice such as browser extensions [5], [6] or antivirus programs [7], [8]. Moreover, there have been a number of detection studies proposed in the literature [9]–[18] using the behavioral features such as CPU usage, WebAssembly instructions, or network traffic.

Unfortunately, in the ecosystem of in-browser cryptomining, even though some website owners ask for explicit user consent before starting mining the clients' resources, none of the browser extensions, antivirus programs, and the detection studies [19]–[23] in the literature differentiates the ones asking for explicit user consent (i.e., *permissioned*) from the ones starting mining without the knowledge and consent (i.e., *permissionless*) of the user. All in-browser cryptomining scripts are blacklisted and blocked by the prevention mechanisms. Google Chrome [24] and Opera [25] has recently announced that they would remove the cryptomining browser extensions from their web store and block the websites containing cryptomining scripts to protect their users as they are mostly being abused in practice.

Motivation. Indeed, the legitimate adoption of this emerging technology is instrumental for several reasons: First of all, today, a substantial portion of the revenue on the web is currently generated through online advertisements. However, the advertisement ecosystem is abused by the attackers, who redirect the users to malicious websites to spread the malware [26] (i.e., malvertising [27]) or ransomware [28]. In this case, permissioned in-browser cryptomining would have been very beneficial by allowing the website owners to monetize their content by charging their users with their processing power instead of making without advertisements. This would reduce the risks posed by malicious advertisements. Second, permissioned in-browser cryptomining would have been a great mechanism to reach a large number of users and provide an easy payment method for nonprofit organizations and

§Both authors contributed equally.

978-1-6654-3485-0/21/$31.00 ©2021 IEEE
DOI 10.1109/DAPPS52256.2021.00008

publishers. In this regard, there have already been a few attempts, such as the Hope project of UNICEF [29] and the media outlet Salon [30]. Third, in-browser cryptomining would offer convenience to end-users with its ad-free and customized content offered by the websites in exchange for uninterrupted use of the users' processing power.

Despite these potential benefits, the legitimate side of in-browser cryptomining has never been analyzed by the community due to its bad fame. This paper is the first attempt to differentiate the permissioned from permissionless cryptomining (i.e., cryptojacking). We present a detailed empirical analysis of permissioned and permissionless in-browser cryptomining operations with real data collected from the web. Specifically, we created a dataset of **6269** unique websites containing cryptomining scripts in their source codes. Then, we performed a detailed cross-correlation analysis between the permissioned and permissionless cryptomining samples to reveal the differences (if any) between them and explore the characteristics of the permissioned cryptomining. In addition, we identified five different user consent methods used by the samples after a further analysis on the permissioned cryptomining samples. Finally, based on the samples we analyzed, our findings, and service provider documentations, we revisit the permissioned cryptomining services with the following questions: 1) Can they be an alternative to advertisement? 2) Do they interrupt the users? 3) Do they satisfy consent requirements? We found that an affirmative answer is possible for each question if implemented properly by the service provider and the website owner, which led us to believe that the potential of permissioned in-browser cryptomining as a legitimate and viable monetization tool.

Contributions. We summarize the main contributions of this paper as follows:

- We, for the first time in the literature, categorized in-browser cryptomining into two categories 1) Permissioned and 2) Permissionless (i.e., cryptojacking).
- We performed an empirical analysis with recent cryptomining samples[1] focusing on the permissioned cryptomining. For this, we collected a large number (i.e., **6269**) of unique cryptomining samples from **14** different service providers. We identified **24** unique keywords that can be used to detect the samples with those service providers. Moreover, we also identified **9** keywords for the consent detection and **4** obfuscated scripts.
- We perform profit, usability, and user consent analysis on the existing cryptomining scripts provided by the service providers found in our dataset.
- We proposed a novel consent evaluation framework for the service providers and presented our benchmarking results for the 14 service providers we detected in the dataset.

Organization. The remainder of this paper is organized as follows: In Section II, we give background information for in-browser cryptomining, and we present our data collection process in Section III. After that, Section IV-A presents the initial analysis of the entire dataset to show the distribution of the service providers and other features of the dataset. Then, in Section IV-B, we delve into our dataset and present the cross-correlation between the different features of the dataset. Section IV-C reports the different consent types we found in permissioned cryptomining samples. Section V analyzes the existing service provider scripts' suitability as a monetization tool and presents the consent evaluation framework we propose. Finally, Section VII concludes the paper.

[1]In order to accelerate the research in this area, we also release our dataset and the list of keywords in the following link: https://bit.ly/3sVj2cp

II. Background: Browser-based Cryptojacking

A. How does it work?

Blockchain technologies ensure the immutability of the chain with several consensus algorithms. The most well-known consensus algorithm is Proof of Work (PoW), and it is used by several leading blockchain technologies such as Bitcoin [31], Ethereum [32], Monero [33], and some other famous cryptocurrencies [34]. The PoW consensus algorithm depends on the processors' computational power such as CPU, GPU, and embedded chipsets (e.g., ASIC miners). In-browser mining aims to use webpage visitors' (i.e., clients) computational power as long as the related tab on the browser remains open on the user side.

Fig. 1: Creation and injection of in-browser mining script

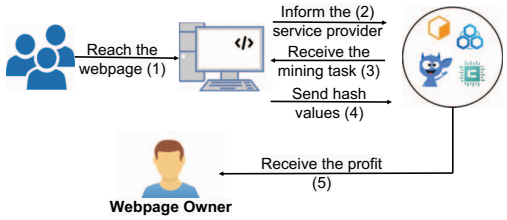

Fig. 2: Lifecycle of in-browser mining

Service providers generally manage in-browser cryptomining source codes and operations, as shown in Figure 1. The webpage owner creates an account on the service provider's website and receives the needed script and credentials for the in-browser mining. The webpage owner embeds this code into the HTML source code or adds it as a plugin for some service providers. After this process, the in-browser mining operation starts as shown in Figure 2, and all the visitors become ad-hoc miners for these webpages and solve mining tasks for webpage owners. Mining tasks are assigned to the users by the service providers, or they may be directly coming from the mining pool. At the end of the pre-defined period, the webpage owners receive the mining share after the service provider cuts the service commission. In this process, the users do not receive any profit.

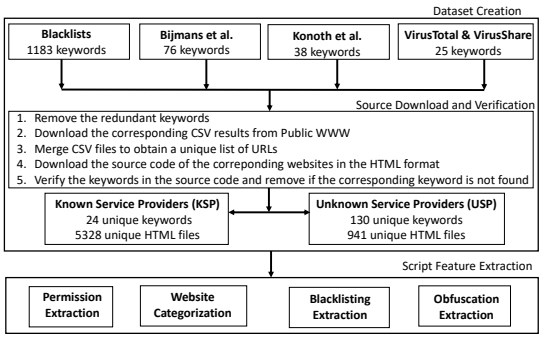

Fig. 3: Data collection process.

B. Permissioned vs. Permissionless Cryptomining

We categorize the cryptomining scripts into two categories: 1) *permissioned* 2) *permissionless*. Permissioned cryptomining samples contain a code snippet for explicit user consent. In contrast, permissionless cryptomining samples do not ask for user consent, i.e., automatically starts mining without the visitor's knowledge or consent. However, while some service providers have methods to implement these options in their script, some of them have different ways of implementing such a user interface.

Listing 1: A sample permissioned in-browser cryptominer script.

```
1 <script src="<path-to-script>"></script>
2 <script>
3     var _client = new Client.Anonymous("<site-key>", {throttle: 0, c:
          'w' });
4     _client.start();
5     _client.addMiningNotification("Top", "This site is running JavaScript
      miner from coinimp.com","#cccccc", 40, "#3d3d3d");
6 </script>
```

Listing 1 shows a sample in-browser cryptominer script provided by the Coinimp service provider [35], which is a currently active service provider. Line 5 in the code snippet contains a method for the user notification. The method can be used by adjusting the parameters by the website owners. The decision to include the notification or not is in the control of the website owner. If line 5 is not included, the miner will start automatically in the background without notifying the user beforehand. Moreover, this method does not give the user the option to opt-out easily. We present the notification method provided by the Coinimp service provider here as a representative example, but there are other user consent methods provided by other service providers as well. We will analyze the service providers in Section IV-A and different user consent types in Section IV-C in further detail.

III. Dataset Creation & Methodology

In this section, we explain the methodology and tools we used for the dataset creation. The entire process is illustrated in Figure 3.

Keyword Collection: In order to find the cryptojacking samples, we used static keyword detection methods. Our primary source is the keyword blacklists released by browser extensions NoCoin [36] and MinerBlock [5]. We obtained a total of 1183 keywords from the merged blacklist of these two lists. Our second source for the keywords is the keyword lists released by other important studies Bijmans et al. [37] and Konoth et al. [38], in which we obtained 76 and 36 keywords, respectively. Our final keyword list is the keyword lists we found manually from the publicly known service providers. With this method, we extracted 25 keywords that could be used to detect cryptomining samples. After this process, we obtained 1322 a list of keywords.

TABLE I: Service provider keywords.

Service Provider	Keyword
Authedmine	authedmine.min.js authedmine.eu/lib/1.js simple-ui.min.js
Browsermine	bmst.pw
Coinhave	cdn.minescripts.info coin-have.com
Coinhive	coinhive.min.js wp-monero-miner-using-coin-hive wp-monero-miner-pro
Coinimp	_client.start
CoinNebula	CoinNebula
Crypto-Loot	CRLT.Anonymous(cryptoloot.pro
DeepMiner	deepMiner.Anonymous deepMiner.Init
JSEcoin	load.jsecoin.com
Monerise	monerise_payment_address
Nerohut	nerohut.com/srv
Webmine	webmine.cz
Webminepool	WMP.Anonymous(wmp-site-key WMP.User
WebMinerPool	webmr.js webminer

Source Download and Verification: The collected list of 1322 keywords also includes duplicates and multiple keywords for the same service providers. Therefore, we removed the duplicates from the list and decided unique keywords for each known service provider. Then, we used PublicWWW [39] to find the corresponding websites containing those keywords in their HTML source code. PublicWWW is a web search engine tool, allowing us to search the HTML source of the websites, as of this writing, including over 500M websites. We downloaded the query result of all of the keywords. Some of the different keywords belong to the same URL; therefore, we again removed the duplicates. After removing the duplicates, we downloaded the corresponding source codes of the websites using a crawler. Our crawler checks the URL's HTTP response and connects to the webpage to fetch the HTML source code. After downloading the HTML source code, our crawler checks the related keywords from the keyword list and saves the HTML document to the related file. To avoid any discrepancies, we searched the keywords in the downloaded source code and removed samples that do not contain the keywords from the dataset. At the end of this process, we obtained a total of 6269 unique samples. We also obtained 24 unique keywords that can be used to identify 14 different service providers and corresponding 5328 HTML files as well as 130 unique keywords and 941 HTML source codes with unknown service providers. We labelled the samples with a known service provider as Known Service Provider (KSP) samples, and we used these samples for the rest of the analysis. In other words, we used 5328 unique samples with known

TABLE II: Consent keywords.

Service Provider	Consent Type	Keyword
Authedmine	Permission	authedmine.min.js
Authedmine	Dashboard	simple-ui.min.js—
Coinimp	Notification	_client.addMiningNotification
Coinimp	Mandatory Mining	messageDiv
Crypto-loot	Dashboard	minui.js
JSEcoin	Permission	load.jsecoin.com
Webminepool	Dashboard	wmp-site-key
Webmine	UI	authedminer.js
WP Monero Miner[1]	Dashboard	wp-monero-miner

[1] WP Monero Miner is indeed not a service provider, however, it provides a plugin which can be used by multiple service provider and it has a dashboard style consent type defined in this paper.

TABLE III: Obfuscation keywords.

Service Provider	Keyword
Coinhive	authedmine.min.js
Crypto-loot	minui.js — crypta.js
Webmine Pool	base.js

service providers for all of the analysis in the paper.

Table I shows the keywords we used for detecting the service providers. While deciding keywords, we tried to use the keywords that can not be changed easily. Instead of the source path, we used variables that are uniquely identifying the service providers.

Script Feature Extraction: In this step, our goal is to obtain more details about the usage of the cryptomining samples. Specifically, we are interested in the following features of the samples:

- *User Consent Extraction:* In this part, we wanted to identify if the samples contain any user consent. For this purpose, we checked the documentation provided by the service providers. We obtained nine different keywords that could be used for user consent detection as well as the type of user consent. The list of keywords are given in Table II. More details about different consent types are explained in Section IV-C.
- *Website Categorization:* In this part, our goal is to see if there is any correlation between the user consent and websites using the cryptomining script. For the website categorization, we used Webshrinker [40], which provides a public web categorization service. However, it only returned a category for the half of the dataset, where we manually labelled the rest of them following the same taxonomy of Webshrinker.
- *Blacklisting Extraction:* In addition, we wanted to identify if a website is blacklisted. For these, we used the public keywords lists released by the browser extensions by NoCoin [36] and MinerBlock [5], and if any keyword from the blacklists is detected in the source code of the sample, we labelled the sample as blacklisted.
- *Obfuscation Extraction:* Similar to the user consent extraction, we also observed that some scripts were obfuscated to avoid being blacklisted. We found 4 scripts given in Table III as obfuscated and labelled the samples utilizing these scripts as obfuscated.

TABLE IV: The list of service providers we identified in our sample set and their other related features.

Service Provider	Activeness	Permission Type	Currency
Authedmine [42]	No	Permissioned	Monero
Browsermine [43]	Yes	Permissionless	BrowserMineCoin
Coinhave [44]	No	Permissionless	Monero
Coinhive [42]	No	Permissionless[1]	Monero
Coinimp [35]	Yes	Optional	Monero Mintme
CoinNebula [45]	No	Permissionless	No info
Crypto-Loot [46]	Yes	Optional	Monero uPlexa
DeepMiner [47]	No	Permissionless	Monero Electroneum Sumokoin [48]
JSEcoin [49]	No	Permissioned	JSEcoin
Monerise [50]	No	Permissionless	Monero
Nerohut [51]	No	Permissionless	Cryptonight coins
Webmine [52]	Yes	Optional	Monero
Webminepool [53]	Yes	Optional	Monero
WebMinerPool [54]	Yes	Optional	Monero

[1] Coinhive is not providing a method to make the sample permissioned; however, it can be integrated to third party extensions to be made permissioned, which we marked them as permissioned sample.

IV. Analysing the In-browser Cryptomining Ecosystem

A. Overall Analysis

In this section, we provide the overall distribution results of our dataset. Table IV shows the list of 14 service providers we identified in our dataset and their related features while Table V shows the sample counts of those service providers.

The List of Service Providers. Using the service providers in [37], [38] and extending them with publicly known service providers, we identified 14 service providers used by the samples in our dataset. We presented the list of service providers in our dataset in Table IV. We found that some of these service providers have discontinued their service. We used the Wayback Machine digital archive [41] to access their documentation for those that are not active. We found that only 6 of 14 service providers are active as of this writing.

We also marked the service providers according to their permission type. For the permission type, we marked the ones who are enforcing the consent method in the source script as ”permissioned” because it does not give an option to remove the user consent to the website owners. On the other hand, we marked the ones which are not providing a method for user consent as ”permissionless”. Finally, we marked as both the service providers, which are providing a method for the user consent optionally. We found that among these 14 service providers, 7 of them are permissionless, while 5 of them are classified as ”optional”. Only Authedmine and JSEcoin embed the user consent in the script so that the website owner cannot remove it unless s/he modifies the original script.

We also extracted the cryptocurrency that can be mined based on the service provider documentation. As can be seen from Table IV, the majority of the service providers prefer Monero due to its anonymity and cpu-oriented mining algorithm features while some of them utilize their own cryptocurrencies such as BrowserMineCoin or JSEcoin.

Service Provider Distribution: Among all cryptomining service providers, the first and most popular one was Coinhive; however, Coinhive discontinued its operations as of March 2019. Since then, many other service providers have surfaced,

TABLE V: Sample counts of 14 different service providers in our dataset.

Service Provider	Total	Permissioned	Permissionless	Blacklisted	Obfuscated
Coinhive	2380	152 (6.4%)	2228 93.6%)	2373 (99.7%)	34 (1.4%)
Coinimp	1123	56 (5%)	1067 (95%)	108 (9.6%)	13 (1.15%)
DeepMiner	492	0 (0%)	492 (100%)	16 (3.2%)	8 (1.6%)
JSEcoin	448	448 (100%)	0 (0%)	444 (99.1%)	53 (11.83%)
Authedmine	378	378 (100%)	0 (0%)	78 (20.63%)	224 (59.2%)
Crypto-Loot	210	6 (2.8%)	204 (97.1%)	181 (86.2%)	136 (64.7%)
Browsermine	155	0 (0%)	155 (100%)	8 (5.2%)	0 (0%)
Webmine Pool	84	5 (5.9%)	79 (94.0%)	7 (8.3%)	77 (91.7%)
WebMinerPool	83	45 (54.2%)	38 (45.8%)	83 (100%)	1 (1.2%)
Coinhave	58	0 (0%)	58 (100%)	35 (60.3%)	0 (0%)
Monerise	31	1[1](3.2%)	30 (96.8%)	0 (0%)	1 (3.2%)
Webmine	23	1 (4.3%)	22 (95.6%)	9 (39.1%)	0 (0%)
Nerohut	6	0 (0%)	6 (100%)	2 (33.3%)	0 (0%)
CoinNebula	1	0 (0%)	1 (100%)	1 (100%)	0 (0%)
Total	**5472**	**1092 (19.9%)**	**4380 (80%)**	**3345 (61.1%)**	**547 (10%)**

[1] This sample is using Coinimp and Monerise together and utilizing Coinimp's notification method.

and some of them still continue their operations. In our dataset, we noted that 43.5% of the samples (2380/5472) still have Coinhive script on their websites. If the script is not deployed locally or in a proxy, basically, these website owners are not making any money. Among the top five service providers with most samples, the only active service provider is Coinimp, with 1123 (20.5%) samples. Using the activeness information of the service providers, we found that in our dataset, a total of 1677 websites (30.6%) are using one of the active service providers. Moreover, we note that 144 samples[2] are using two service providers at the same time. Among these, in 139 samples, either of the service providers is Coinhive or Authedmine, while others contain Coinimp' scripts. This shows the popularity of these service providers as they are either used standalone or along with others.

User consent Distribution: As we showed in Table IV, seven service providers do not provide a method for the user consent while five of them provide an optional method, and Authedmine and JSEcoin are enforcing the website owner to ask for the user consent explicitly by embedding the user consent code snippet in the source of the script. Using the permission extraction method we explained in the previous section, we identified 1092 permissioned cryptomining scripts from all service providers. Among all service providers, the three service providers with the most permissioned samples are JSEcoin, Authedmine, and Coinhive, where none of them are active as of now. We found 56 samples utilizing a permission method provided by Coinimp, which is active as of now. From its documentation, we found that the most basic script provided by Coinimp does not indeed utilize a permission method. Among service providers supporting both permissioned and permissionless cryptomining, a similar ratio is also observed for Crypto-Loot and Webmine Pool. For these, we conclude that most website owners are using the most basic script provided by the service provider.

[2]Therefore, we further want to note that we have 5328 unique websites in our dataset, 144 of them are counted twice as they included two cryptomining scripts. We found no websites using more than two cryptomining scripts at the same time in our dataset.

Are all samples in the dataset blacklisted? As we explained in Section III, using the keyword list publicly released by the browser extensions, we can decide if a given website is going to be blacklisted by one of these blockers. We found that these browser extensions blacklist 61.1% (3345/5472) of the websites in our dataset. Moreover, we also noticed that most of the samples utilizing Coinhive (99.7%) are blacklisted as it is the pioneer in the cryptomining business. Similarly, JSEcoin's blacklisting ratio is also very high as its parameters are embedded in the script, and the script is called in one line. In this method, when the source for the script is blacklisted, all of the samples using that same URL will be blocked. On the other hand, there are also service providers with very low blacklisting ratios such as Coninimp (9.6%), DeepMiner (3.2%), Browsermine (5.2%), Webmine Pool (8.3%), Monerise (0%).

Are the samples deploying obfuscated scripts? We noticed that some of the scripts are utilizing obfuscated scripts. We identified four such scripts, which is given in Table III. In our dataset, we observed that 547 (10%) cryptomining scripts are utilizing one of these scripts, which may be suspicious. We also noticed that even though these scripts are obfuscated, they can be detected by the blacklists by the code snippet calling the source of the script. We basically used this method in order to identify these samples. We also note that this method cannot detect the scripts located on a proxy server or locally with a different filename.

B. Comparison of the Permissioned and Permissionless Cryptomining

As shown in Section IV-A, we found that 1092 (19.9 %) samples in our dataset are utilizing a way to notify the user about the cryptomining operation that will be performed using the user's resources. In this section, our purpose is to compare the permissioned with permissionless samples to reveal the differences (if any) between them and explore the characteristics of the permissioned cryptomining. For this purpose, we perform a cross-correlation analysis among the features of the samples in our dataset. Particularly, we analyze the correlations between the following pairwise features: 1) activeness vs permission type, 2) blacklisting vs permission type, and 3) web category vs permission type. We present our results in the following subsections.

1) Activeness: As we noted in Section IV-A, not all of the service providers are active and continue their operations. For example, the most common permissioned cryptojacking service providers Coinhive/Authedmine, JSEcoin, WP Monero Miner do not continue their operations. Figure 4a and 4b show the activeness ratio of permissioned and permissionless cryptomining samples, respectively. We can see that the activeness ratio of the permissioned samples is 11.5%, while it is 35.9% for the permissionless cryptomining samples.

2) Blacklisting: Figure 4a and Figure 4b show the distribution of blacklisted websites for permissioned and permissionless cryptomining samples. We found that blacklisted websites' ratios are very close to each other (37.0% and 40.5%) for permissioned and permissionless. The reason for this is that blacklists do not really differentiate between the permissioned cryptojacking samples. Some service providers

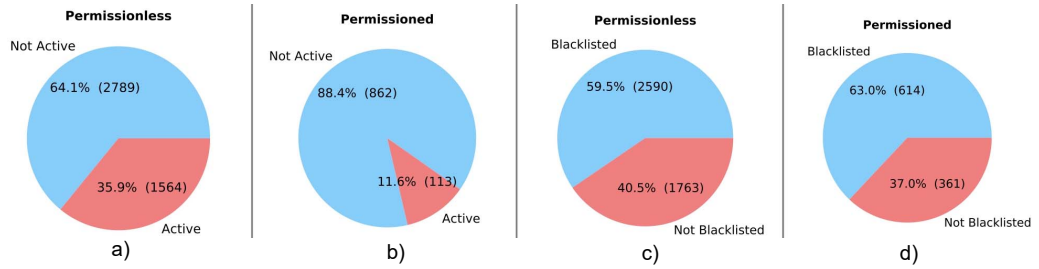

Fig. 4: Activeness distribution of a) permissioned (b) permissionless cryptomining samples. Blacklisted distribution of (c) permissioned (d) permissionless cryptomining samples.

such as Authedmine or Webmine enforce the website owners to use a permissioned version of the scripts by embedding and sometimes making it impossible to modify by obfuscating to avoid being blacklisted. However, as we can see from the distribution, the blacklists do not consider if a script provides consent type to the website owner while choosing keywords for the blacklists. While this is to avoid the malicious cryptojacking samples, it also kills the responsible, legitimate website owners' functionality.

3) Web Category: Figure 5 shows the corresponding web categories used by the permissioned and permissionless cryptomining samples for those that can be categorized by Webshrinker. From Figure 5, adult content is the highest web category for permissionless samples, while the permissioned cryptomining samples are used on technology websites. On the other hand, overall distribution also shows the more normal distribution for permissioned samples, while the adult content dominates the permissionless samples.

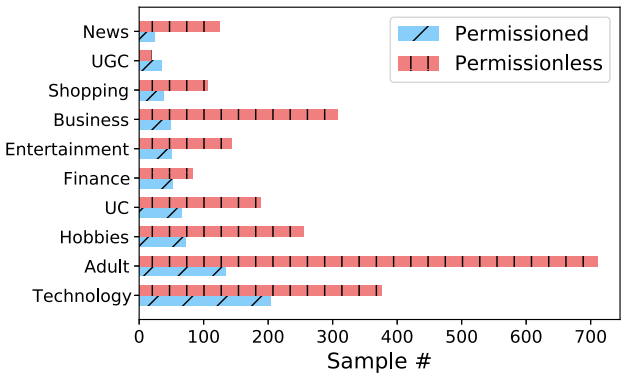

Fig. 5: Website categorization distribution of permissioned permissionless cryptomining samples. (UC: Under Construction, UGC: User-generated Content)

C. Extracting User Consent Methods

In Section IV-A and IV-B, we analyzed the entire dataset and showed that the permissioned and permissionless samples mostly show similar behaviors, and they are mostly treated the same by the blacklists although the cryptomining business is shifting from the permissionless to permissioned cryptomining. In this section, we will focus on the permissioned cryptomining samples. Using the permissioned cryptomining samples, we identified five different types of consent types used by the samples: 1) permission, 2) dashboard, 3) notification, 4) mandatory mining, and 5) user interface (UI).

TABLE VI: The permission types provided by the service providers and their sample counts found in our dataset.

Permission Type	Sample #
Permission	666
Dashboard	255
Notification	51
Mandatory Mining	2
UI	2

Table VI shows the permission types provided by the service providers and their sample counts found in our dataset. The results show that the permission, which we explain in the next subsection, is the most common consent type among the permissioned cryptomining samples in our dataset. The list of keywords, permission type, service providers offering these methods, and the keyword decision process are given in Table II.

Figure 6 shows an example for each of the five consent methods, and in the next subsections, we explain these five consent methods in greater detail.

1) Notification: In the notification consent type, the user is notified by showing a pop-up screen. It only notifies the user without giving a selection to opt out. The Coinimp provides an example of the notification consent type, and it is shown in Figure 6a. The visibility of the notification can be adjusted by the website owner. For example, the text itself, text color, background color, as well as the size and position of the notification, can be configured. We note most of the samples in our dataset were using default settings; however, the notification can be even hidden from the user.

Moreover, this method is not mandatory and does not have to be placed in the code by the website owner. In this case, the mining will be starting automatically when the website is accessed by the user, which we call permissionless cryptomining. We found that only 51 (4.5%) of 1123 Coinimp samples are using this consent type in our dataset.

2) Permission: Permission consent type is similar to the notification, but it also gives the user an option to opt out from cryptomining. From the service providers in our dataset, we found that Authedmine and JSEcoin's scripts have a method to provide the permission consent type. Authedmine's script intentionally enforces this option in the source of the cryptomining script, and it does not provide an option to

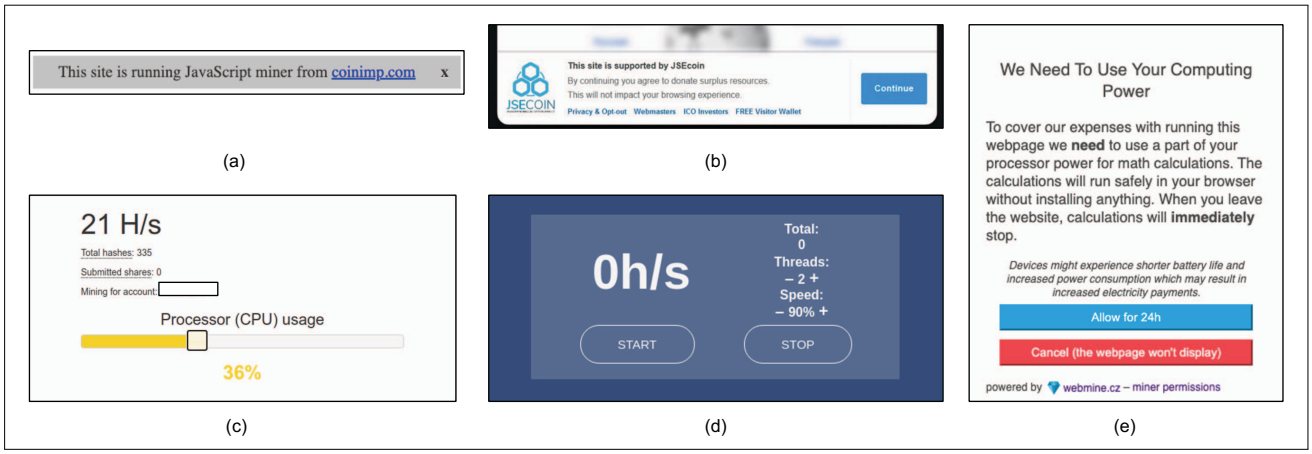

Fig. 6: (a) In-browser notification provided by CoinImp (b) The user view of JSECoin's permission consent type. (c) The user view of Webmine's UI consent type. (d) The user view of Webminepool's dashboard consent type. (e) The user view of Webmine's Mandatory mining consent type.

remove that part from the script to the website owner. In this way, the service provider's purpose is to avoid being blocked and to continue their service. However, as we have noted in Section IV-B, the blacklist does not consider being permissioned. On the other hand, as JSEcoin is originally proposed for transparency and accountability, similar to Authedmine's script, this part of the script is embedded into the source, and the website owner can not remove it. JSEcoin's permission example is shown in Figure 6b. As can be seen from the figure, the user can easily click the opt-out option and prefer not to allow the website to use his/her computational power.

3) Dashboard: Dashboard is another consent type where the user has the ability to start and stop the mining as well as configure the parameters such as the number of threads or CPU percentage. Configurable parameters may be essential for the websites who want their user to decide the use of his/her resources depending on the convenience. Webminepool provides an example of a dashboard consent types shown in Figure 6d. Some of the service providers implement dashboards so that mining starts automatically, but the users can set the parameters or stop the mining using the dashboard.

4) User Interface (UI): UI is an improved version of the dashboard, where the user can set the parameters in an increased scaling as well as a better experience through the use of elements such as sliders. It requires more effort by the service provider, but it gives easy control to the users. We show an example of Webmine's UI consent method in Figure 6c. As can be seen from the figure, the user can see its resources used by the service provider and can also set the Processor CPU usage through the slider element.

5) Mandatory Mining: Mandatory mining is another consent type, in which if the user does not accept the mining, s/he will not be allowed to access to the website. The Coinimp service provider gives a method for this consent type. This method is also the rarest mining method in our dataset, and is found in only two samples. For example, Figure 6e shows the user view of Webmine's mandatory mining consent message.

Service Provider	Throttle	Time (min)	Currency	Profit (in currency)	Average Price (6 Months)	Profit (USD)
Coinimp	100	1	Mintme	0.03199108	0.0024 USD	0.000074
Browsermine	100	1	BMC	0.000000009101	0.047 USD	0.0000000004277
Webminepool	100	1	Monero	0.00001172	141.29 USD	**0.001655**
Webmine	100	1	Monero	0.000001312308	141.29 USD	0.000185
Crypto-Loot	100	1	Uplexa	0.17	0.00019 USD	0.0000323

TABLE VII: Real-time profitability analysis of active service providers.

V. Revisiting the Permissioned Cryptomining

In this section, our goal is to see if the existing service provider scripts can serve as a monetization tool, rather than an attack tool. For this purpose, we test them to see

- whether they can be an alternative revenue mechanism,
- whether they interrupt the end-users,
- whether they satisfy the consent requirements.

In the following subsections, we present our analysis results for each of them.

A. Can they be an alternative to advertisement?

The idea of using permissioned in-browser cryptomining as a monetization tool for the websites brings the question of how much profit they offer for the website visits. For this purpose, we deployed the scripts provided the active service providers found in our dataset on a sample website and calculated the total profit per user per minute. We run each experiment five times and calculated the average. Table VII shows the real-time calculation of the monetary value for one regular user in one minute. As service providers may use different currencies, we converted them into the corresponding USD value as of this writing[3]. As can be seen from the table, Webminerpool offers significantly more profit compared to the other service providers due to it's reward/price balance.

Moreover, even though an advertisement is assumed to pay a constant, the profit of a cryptominer increases linearly as the visit duration increases. Figure 7 shows the profit per user value for varying visit duration. An average ads revenue is calculated per thousand impressions and on averages it varies

[3]Nov 30, 2020.

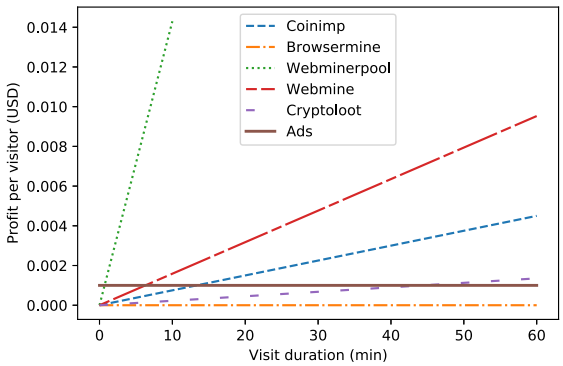

Fig. 7: Profit per user value for different visit duration.

Conf 1	Chrome 2 tab static page
Conf 2	Conf 1 + 1080P Youtube Video
Conf 3	Conf 2 + Spotify
Conf 4	Conf 3 + Whatsapp App Video Download process
Conf 5	Conf 3 + Software uptader
Conf 6	Conf 4 + 4K video
Conf 8	Conf 4 + Virtual machine

TABLE VIII: Configurations used for evaluating the usability of in-browser cryptomining.

between 0.5-2 USD [55]. We assume 1 USD in our calculations. As we can see from the figure, while Webminepool can reach to the same amount of profit offered by the ads in less than a minute, Browsermine can not make the same amount of profit even for the 60 minutes of visit duration.

B. Do they interrupt the users?

In this section, our purpose is to perform a usability test on the end users of in-browser cryptomining as greedy mining operations can be computationally challenging process for daily user computers. For the experiments, we used computes with four different CPUs: 1) Intel Core i5, 2) Intel Core i7, 3) Intel Xeon E5, and 4) Intel Xeon Gold for the throttle values 0.2, 0.5, and 0.8. All of the computers used in our experiments use the same RAM (16 gb, 1333MHz), similar SSDs and the same Linux distribution (Ubuntu 18.04 LTS). We run all experiments 5 times with the exactly the same applications to obtain more accurate results.

We present the configurations in Table VIII. As a result of the analysis, we observed that it is possible to efficiently use in-browser mining scripts if the web page owner will not act greedy and keep the throttle rate under the 50%. Above 50% will dramatically affect the user's experience and user might tend to close the related tab due to the interruptions and overheating.

C. Do they satisfy consent requirements?

If implemented properly and responsibly, in-browser cryptomining can be used for good such as collecting donations as in UNICEF's case example [29]. However, the lack of an evaluation framework for proper and responsible implementation of the permissioned cryptomining makes this technology's widespread adoption impractical, if not impossible, for the website owners. Therefore, here we first present a consent evaluation framework and then use it to rank the current service providers in our dataset.

1) A Consent Evaluation Framework: Herein, we suggest ten requirements, encompassing two categories: User Requirement (UR) and Webpage owner Requirement (WR).

User-related Requirements:

- *UR1: A method to notify the user to the website owner:* The webpage owners should at least inform the users about mining activity on the webpage. Some service providers offer built-in functions to inform users. This notification should be visible and explain in-browser mining employed by the webpage. The webpage owner may also place an informative link at the end of the notification for more information.
- *UR2: An option or link to the user to learn more info about the mining service:* With this method, webpage owners allow their users to learn more about the meaning of the in-browser mining and what they actually do during their visit to the related webpage. This option is also used by fundraising projects/webpages. These webpages aim to provide additional income for several civil society initiatives and humanitarian projects.
- *UR3: An opt-out screen to the user:* Webpages that operate in-browser mining should give options to their users. If the user does not want to give permission, the webpage should not force the user to allow the mining script. JSE coin, Webmine, and Authedmine let their users make this decision. Besides, webpage owners may not want users to visit their webpage if they do not wish to opt out. This is a decent option for both users and webpage owners.
- *UR4: An option to stop the mining after starting during the session:* Some service providers have built-in functionality to stop and start mining operations manually. This feature allows the user to control the mining process and give the option to stop it if desired/needed. This method is very advantageous for both users and the webpage owners because if the user experiences interruption, they can stop the miner and continue surfing on the webpage.
- *UR5: An option to adjust the parameters (e.g., threads, CPU):* Most of the scripts in our dataset have standard CPU usage permissions (also known as throttle and threads variables) set by the website owners. Some websites and service providers allow their users to set how much they want to contribute to mining. Currently, this is the highest point of consent. If the user does not wish to contribute via mining, they can easily set the parameter to zero and continue surfing. Or if the website gets too greedy, which could interrupt the user, the user can limit the parameters according to his/her convenience.

Webpage Owner Requirements:

- *WR1: An option to change the notification message and its properties:* While some service providers let the website owner choose their own messages and change the properties (e.g., text and background color, location) of the notification, some others have mandatory features that directly come from the main library. This feature has several pros and cons for both users and the web-

TABLE IX: Summary of the service providers' consensual cryptomining features.

Service Provider	UR1	UR2	UR3	UR4	UR5	WR1	WR2	WR3	WR4	WR5	Total (/10)
Authedmine	◐	◐	●	◐	○	◐	◐	○	○	◐	4
Browsermine	○	○	◐	◐	◐	○	○	○	○	○	1.5
Coinhave	○	○	◐	◐	○	○	◐	○	○	◐	2
Coinhive	○	○	○	○	○	○	○	○	○	●	1
Coinimp	◐	◐	◐	○	○	●	●	○	○	●	4.5
CoinNebula	○	○	○	○	○	○	○	○	○	○	0
Crypto-Loot	○	○	●	●	●	○	○	○	○	●	4
DeepMiner	◐	○	○	◐	●	○	○	◐	●	○	3.5
JSEcoin	●	●	●	●	○	○	●	○	○	●	6
Monerise	○	○	○	○	○	○	○	○	○	○	0
Nerohut	○	○	○	○	○	○	○	○	○	○	0
Webmine	◐	○	●	●	●	●	●	○	●	●	**7.5**
Webminepool	○	○	◐	●	●	○	○	●	●	●	5.5
WebMinerPool	◐	◐	●	●	●	○	◐	◐	●	●	**7**

○: Does not satisfy ◐: Partially satisfies ●: Fully satisfies

page owner. Firstly, when service providers let webpage owners arrange the parameters themselves, the webpage owner makes the notification message out of sight or very hard to notice. On the other hand, when service providers embed a mandatory mining script to the main library, the webpage owner must permit some third party to add content to the webpage.

- *WR2: An option to make the script mandatory:* Webpage owners may want to block users who do not exchange their computational power for webpage content. This option is beneficial for the webpage owner, and service providers should provide this option to the webpage owners.
- *WR3: An option to enable/disable the start/stop mining buttons:* Service providers generally make this option mandatory for the webpage owners, but this decision should be left to the webpage owner. As mentioned before, the webpage owner should choose what is going to be displayed on their webpage.
- *WR4: An option to change and modify dashboard location and design:* The dashboard and GUI-based control panels are user-friendly, but their design is generally not changeable by the webpage owners. Webpage owners should have the right to change the design of the dashboard under several restrictions.
- *WR5: A good documentation and guideline for website owners:* Service providers offer several functionalities to the webpage owners. While several service providers publish very poor documentation, some others provide high-quality documentation.

Table IX shows how much the top service providers in our list fit into this framework. Some service providers fit into our framework with an impressive score (e.g., Webmine and Webminerpool). These service providers offer various options for both users and service providers; however, their user options are controlled by the webpage owners. If the webpage owner does not prefer to use any of them, service providers do not force them to do that. Besides, service providers like JSEcoin, Cryptoloot, and Authedmine force consent method usage, and this makes these service providers more user friendly.

VI. Related Work

Cryptojacking Detection. In recent years, there has been an increasing body of literature investigating cryptojackings. Most of these studies focus on the detection of cryptojacking malware [9]–[18], [38], [56]. These studies investigate different behavioral features to propose a more accurate detection mechanism.

Empirical Cryptojacking Analysis. There are also other studies focusing on the empirical analysis of the cryptojacking samples from different perspectives. Some of the topics that have been focused in these studies are the user experience analysis [57], [58], economic analysis [59], and campaign analysis [37] in-browser cryptomining. The closest work to ours is Carlin et al. [60] discussing the legality and ethical side of the cryptojackings. However, there is no study differentiating the permissioned and permissionless cryptomining samples and focusing specifically on the permissioned in-browser mining. To the best of our knowledge, this is the first study that focuses on analyzing the behaviour of permissioned in-browser cryptomining in greater detail.

VII. Conclusion

In this paper, we analyzed the characteristics of the in-browser mining ecosystem and the service providers' consent methods using a large dataset, which consists of 6269 unique websites containing cryptomining script in their source codes. We created the first consent focused in-browser cryptomining dataset in the literature and classified it under different consent methods. After the classification process, we analyzed our results and shared our findings. In light of the new classification process, we categorized consent types under different sections. We used the samples we found in the wild during these classifications. Another contribution of this research is a new evaluation framework for service providers and developers who want to implement a user consent-based in-browser cryptomining. This framework is adaptable and extensible for both academic research and service provider implementations. We believe this paper will further increase the widespread adoption of legitimate cryptomining with user consent and knowledge and will increase the awareness on in-browser cryptomining.

Acknowledgements

We would like to thank the anonymous reviewers for their feedback and time. This work was partially supported by the U.S. National Science Foundation (NSF) (Awards: NSF-CAREER CNS-1453647, NSF-1663051, NSF-CNS-1718116, NSF-CNS-1703454), and ONR under the "In Situ Malware" project. The views expressed are those of the authors only.

References

[1] D. Goodin, "Miners in youtube ads," https://arstechnica.com/information-technology/2018/01/now-even-youtube-serves-ads-with-cpu-draining-cryptocurrency-miners/, accessed: April 13, 2020.

[2] E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda, and A. A. Selcuk, "Sok: Cryptojacking malware," in *6th IEEE European Symposium on Security and Privacy*, Virtual, September 2021.

[3] "Mid-year update — july 2020, 2020 sonicwall cyber threat report," https://www.sonicwall.com/resources/2020-cyber-threat-report-mid-year-update-pdf/, July 2020, accessed: October 19, 2020.

[4] K. Parrish, "Uk government plugin based mining," https://www.digitaltrends.com/computing/government-websites-plugin-coinhive-monero-miner/, accessed: April 13, 2020.

[5] "Minerblock: An efficient browser extension to block browser-based cryptocurrency miners all over the web." https://github.com/xd4rker/MinerBlock/blob/master/assets/filters.txt, accessed: April 8, 2020.

[6] "Easy-redirect broser extension against in-browser mining," https://chrome.google.com/webstore/detail/easy-redirect-prevent-cry/kceciaijnoaceceljkggfocngjleimem?hl=en, accessed: October 15, 2020.

[7] "Mcafee cryptojacking blocker." https://www.mcafee.com/blogs/consumer/webadvisor-cryptojacking-blocker/, accessed: October 15, 2020.

[8] "Avast cryptojacking blocker," https://www.avast.com/c-protect-yourself-from-cryptojacking, Jan 2018, accessed: October 15, 2020.

[9] J. D. P. Rodriguez and J. Posegga, "Rapid: Resource and api-based detection against in-browser miners," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 313–326.

[10] A. Gangwal, S. G. Piazzetta, G. Lain, and M. Conti, "Detecting covert cryptomining using hpc," in *International Conference on Cryptology and Network Security*. Springer, 2020, pp. 344–364.

[11] J. Z. i Muñoz, J. Suárez-Varela, and P. Barlet-Ros, "Detecting cryptocurrency miners with netflow/ipfix network measurements," in *2019 IEEE International Symposium on Measurements & Networking (M&N)*. IEEE, 2019, pp. 1–6.

[12] R. Ning, C. Wang, C. Xin, J. Li, L. Zhu, and H. Wu, "Capjack: Capture in-browser crypto-jacking by deep capsule network through behavioral analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1873–1881.

[13] C. Kelton, A. Balasubramanian, R. Raghavendra, and M. Srivatsa, "Browser-based deep behavioral detection of web cryptomining with coinspy," in *Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb) 2020*, 2020, pp. 1–12.

[14] H. N. C. Neto, M. A. Lopez, N. C. Fernandes, and D. M. Mattos, "Minecap: super incremental learning for detecting and blocking cryptocurrency mining on software-defined networking," *Annals of Telecommunications*, pp. 1–11, 2020.

[15] G. Hong, Z. Yang, S. Yang, L. Zhang, Y. Nan, Z. Zhang, M. Yang, Y. Zhang, Z. Qian, and H. Duan, "How you get shot in the back: A systematical study about cryptojacking in the real world," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 1701–1713.

[16] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Web-based cryptojacking in the wild," *arXiv preprint arXiv:1808.09474*, 2018.

[17] R. Tahir, S. Durrani, F. Ahmed, H. Saeed, F. Zaffar, and S. Ilyas, "The browsers strike back: countering cryptojacking and parasitic miners on the web," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 703–711.

[18] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao, "Seismic: Secure in-lined script monitors for interrupting cryptojacks," in *European Symposium on Research in Computer Security (ESORICS)*. Springer, 2018, pp. 122–142.

[19] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya, "A usable and robust continuous authentication framework using wearables," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[20] A. Acar, W. Liu, R. Bayeh, K. Akkaya, and A. S. Uluagac, "A privacy-preserving multifactor authentication system," *Security and Privacy*, vol. 2, no. 5, p. e88, 2019.

[21] Z. B. Celik, A. Acar, H. Aksu, R. Sheatsley, P. McDaniel, and A. S. Uluagac, "Curie: Policy-based secure data exchange," in *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, 2019, pp. 121–132.

[22] A. Acar, Z. B. Celik, H. Aksu, A. S. Uluagac, and P. McDaniel, "Achieving secure and differentially private computations in multiparty settings," in *Privacy-Aware Computing (PAC), 2017 IEEE Symposium on*. IEEE, 2017, pp. 49–59.

[23] F. Naseem, A. Aris, L. Babun, E. Tekiner, and S. Uluagac, "MINOS: A lightweight real-time cryptojacking detection system," in *28th Annual Network and Distributed System Security Symposium, NDSS, February 21-25, 2021*, 2021.

[24] "Google bans all cryptomining extensions from the chrome store," https://www.wired.com/story/google-bans-all-cryptomining-extensions-from-the-chrome-store/, 2018, accessed: April 7, 2021.

[25] "Opera mini and mobile now block cryptocurrency-mining scripts," https://www.androidpolice.com/2018/01/22/opera-mini-mobile-now-block-cryptocurrency-mining-scripts/, accessed: April 7, 2021.

[26] A. Acar, L. Lu, A. S. Uluagac, and E. Kirda, "An analysis of malware trends in enterprise networks," in *International Conference on Information Security*. Springer, Cham, 2019, pp. 360–380.

[27] A. K. Sood and R. J. Enbody, "Malvertising–exploiting web advertising," *Computer Fraud & Security*, vol. 2011, no. 4, pp. 11–16, 2011.

[28] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *arXiv preprint arXiv:2102.06249*, 2021.

[29] UNICEF, "Give hope, just by being here," https://web.archive.org/web/20200518224428/https://www.thehopepage.org/, accessed: October 16, 2020.

[30] Salon, "Faq: What happens when i choose to "suppress ads" on salon?" https://web.archive.org/web/20200604052723if_/https://www.salon.com/about/faq-what-happens-when-i-choose-to-suppress-ads-on-salon, 2019, accessed: October 16, 2020.

[31] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[32] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[33] H. Takahashi, S. Nakano, and U. Lakhani, "Sha256d hash rate enhancement by l3 cache," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2018, pp. 849–850.

[34] "Full list of pow-based cryptocurrencies," https://cryptoslate.com/cryptos/proof-of-work/, accessed: October 15, 2020.

[35] "The official webpage of coinimp," https://www.coinimp.com/documentation, accessed: April 7, 2021.

[36] "Nocoin: Block lists to prevent javascript miners," https://github.com/hoshsadiq/adblock-nocoin-list, accessed: April 8, 2020.

[37] H. L. Bijmans, T. M. Booij, and C. Doerr, "Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 1627–1644.

[38] R. K. e. a. Konoth, "Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 1714–1730.

[39] "Source code search engine," https://publicwww.com/, accessed: October 16, 2020.

[40] "Domain data & threat intel, powered by artificial intelligence," https://www.webshrinker.com/, accessed: October 16, 2020.

[41] "Internet archive wayback machine," https://web.archive.org/, accessed: October 16, 2020.

[42] "The official webpage of both coinhive and authedmine," http://web.archive.org/web/20190130232758/https://coinhive.com/documentation, accessed: April 7, 2021.

[43] "The official webpage of browsermine," https://browsermine.com/, accessed: April 7, 2021.

[44] "The official webpage of coinhave," http://web.archive.org/web/20180102115842/https://coin-have.com/, accessed: April 7, 2021.

[45] "The official webpage of coinnebula," https://web.archive.org/web/20180818144049/https://coinnebula.com/, accessed: April 7, 2021.

[46] "The official webpage of cryptoloot," https://crypto-loot.org/, accessed: April 7, 2021.

[47] "The official github page of deep miner," https://github.com/deepwn/deepMiner, accessed: April 7, 2021.

[48] "The official webpage of sumokoin," https://www.sumokoin.org/, accessed: April 7, 2021.

[49] "Jse coin official webpage," https://jsecoin.com/, accessed: April 7, 2021.

[50] "The official webpage of monerise," http://web.archive.org/web/20200813110918/http://monerise.com/, accessed: April 7, 2021.

[51] "Nerohut official webpage," https://web.archive.org/web/20190131001253/https://nerohut.com/documentation.php, accessed: April 7, 2021.

[52] "The official webpage of webmine," http://webmine.cz/, accessed: April 7, 2021.

[53] "The official webpage of webmine pool," https://www.webminepool.com/, accessed: April 7, 2021.

[54] "The official webpage of webminerpool," https://github.com/notgiven688/webminerpool, accessed: April 7, 2021.

[55] "Adsense cpm rates in usa: 2020," https://adcpmrates.com/2020/08/16/adsense-cpm-rates-in-usa/, accessed: April 7, 2021.

[56] H. Darabian, S. Homayounoot, A. Dehghantanha, S. Hashemi, H. Karimipour, R. M. Parizi, and K.-K. R. Choo, "Detecting cryptomining malware: a deep learning approach for static and dynamic analysis," *Journal of Grid Computing*, pp. 1–11, 2020.

[57] P. H. Meland, B. H. Johansen, and G. Sindre, "An experimental analysis of cryptojacking attacks," in *Nordic Conference on Secure IT Systems*. Springer, 2019, pp. 155–170.

[58] R. Holz, D. Perino, M. Varvello, J. Amann, A. Continella, N. Evans, I. Leontiadis, C. Natoli, and Q. Scheitle, "A retrospective analysis of user exposure to (illicit) cryptocurrency mining on the web," *arXiv preprint arXiv:2004.13239*, 2020.

[59] M. Saad, A. Khormali, and A. Mohaisen, "Dine and dash: Static, dynamic, and economic analysis of in-browser cryptojacking," in *2019 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2019, pp. 1–12.

[60] D. Carlin, J. Burgess, P. O'Kane, and S. Sezer, "You could be mine (d): the rise of cryptojacking," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 16–22, 2019.