How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World

Geng Hong
Fudan University
ghong17@fudan.edu.cn

Zhemin Yang
Fudan University
yangzhemin@fudan.edu.cn

Sen Yang
Fudan University
syang15@fudan.edu.cn

Lei Zhang
Fudan University
lei_zhang14@fudan.edu.cn

Yuhong Nan
Fudan University
nanyuhong@fudan.edu.cn

Zhibo Zhang
Fudan University
zbzhang15@fudan.edu.cn

Min Yang
Fudan University
m_yang@fudan.edu.cn

Yuan Zhang
Fudan University
yuanxzhang@fudan.edu.cn

Zhiyun Qian
University of California Riverside
zhiyunq@cs.ucr.edu

Haixin Duan
Tsinghua University
duanhx@tsinghua.edu.cn

ABSTRACT

As a new mechanism to monetize web content, cryptocurrency mining is becoming increasingly popular. The idea is simple: a webpage delivers extra workload (JavaScript) that consumes computational resources on the client machine to solve cryptographic puzzles, typically without notifying users or having explicit user consent. This new mechanism, often heavily abused and thus considered a threat termed "cryptojacking", is estimated to affect over 10 million web users every month; however, only a few anecdotal reports exist so far and little is known about its severeness, infrastructure, and technical characteristics behind the scene. This is likely due to the lack of effective approaches to detect cryptojacking at a large-scale (e.g., VirusTotal).

In this paper, we take a first step towards an in-depth study over cryptojacking. By leveraging a set of inherent characteristics of cryptojacking scripts, we build CMTracker, a behavior-based detector with two runtime profilers for automatically tracking **C**ryptocurrency **M**ining scripts and their related domains. Surprisingly, our approach successfully discovered 2,770 unique cryptojacking samples from 853,936 popular web pages, including 868 among top 100K in Alexa list. Leveraging these samples, we gain a more comprehensive picture of the cryptojacking attacks, including their impact, distribution mechanisms, obfuscation, and attempts to evade detection. For instance, a diverse set of organizations benefit from cryptojacking based on the unique wallet ids. In addition, to stay under the radar, they frequently update their attack domains (fastflux) on the order of days. Many attackers also apply evasion techniques, including limiting the CPU usage, obfuscating the code, etc.

KEYWORDS

Cryptojacking, Cryptocurrency Mining, Malicious JavaScript

1 INTRODUCTION

The web has long been fueled by online advertising to power the "free" content. Yet users are often dissatisfied with the experience, evident by the popularity of adblockers. As an alternative monetization mechanism, cryptocurrency mining has started to gain traction. Due to the ease of monetization and relative non-intrusiveness of the nature of cryptocurrency mining (not visible), it has also been heavily abused. In particular, websites often conduct cryptocurrency mining without explicit user content. Several anecdotal reports cover this type of abuse, named cryptojacking [25]. However, little is known about this ongoing threat: How prevalent is it? How does the abuse occur? Who distributes the malicious payload? How stealthy are mining scripts, e.g., attempts to evade detection? Are they mitigated by existing solutions? To the best of our knowledge, this paper is the first systematic study on web cryptojacking.

To answer the above questions, the first task is to identify such cryptojacking webpages at scale. This task has never been fully accomplished, due to the limitation of existing identification approaches. Existing approaches to collect cryptojacking samples rely on the assumptions that a cryptojacking web page either exhausts the victim's CPU resources [17], or contains explicit keywords as malicious payload signatures [14]. However, we observed many counterexamples.

Our Approach. In this paper, we take a deeper look into the nature of cryptojacking and target the least common denominator of their workloads: regular, repeated, and hash-based computation. We propose a cryptojacking detector, named CMTracker, armed by two runtime behavior-based profilers. The first profiler, *hash*

CCS '18, October 15–19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5693-0/18/10…$15.00
https://doi.org/10.1145/3243734.3243840

based profiler, leverages the nature of a proof-of-work system that requires heavy workload to compute hashes. We therefore monitor the statistics of the script behaviors related to hash activities. Our second profiler, *stack structure based profiler*, is based on the observation that cryptocurrency mining is repeated and regular and therefore the call stacks of the mining scripts must be periodic. Thus, we record the runtime stacks and identify the mining pages by looking for the hotspot of calling contexts.

In summary, we are able to collect 2,770 cryptojacking samples from 853,936 popular web pages, including 868 among top 100K in Alexa list. In addition, 53.9% of these identified samples would have not been identified with current widely used detectors based on blacklists (See Section 6.1). Besides, our approach discovers over 260% more cryptojacking, a significant increase compared with the latest available public reports only after 2 months (See Section 4.2). Our manual verification over a subset of the identified result shows that all of them are true positives, indicating that CMTracker achieves a good performance of cryptojacking scripts in both coverage and precision.

Measurement and Findings. Assisted by CMTracker, we first measure the severeness of this threat (See Section 4.1). From the 2,770 cryptojacking websites detected by CMTracker, we estimate that they affect 10 million web users per month (See Section 4.2). By conducting a further analysis of this result, we show that cryptojacking workloads cost more than 278K kWh extra power daily, equivalent of the energy consumption of a small town with 9.3K people. Meanwhile, attackers are estimated to earn over 59K US dollars daily. Second, we study the infrastructure of the cryptojacking attacks. Our findings show that various malicious domains are leveraged in collaboration with each other. We empirically classify these domains, and analyze their functionality, distribution and life cycle. Furthermore, many domains are exclusively used in only one cryptojacking sample, and the payloads of different samples rarely contribute to a same beneficial attacker wallet. This indicates a significant number of real-world attackers. In addition, the cryptojacking pages rapidly change their domains, rendering the existing blacklist-based solutions ineffective. Specifically, our experiments show that more than 20% of the cryptojacking domains last less than nine days, while the blacklists are updated every 10 to 20 days on average [21, 35].

Next, we study the evasion techniques applied by these cryptojacking pages. Our findings reveal that at least 3 types of techniques are applied to evade currently available detection approaches, including limiting CPU usage, code obfuscation for mining scripts and payload hiding. These techniques raise the bar for detecting such stealthy behaviors from different perspectives.

To the best of our knowledge, this work is the first of its kind and provides valuable insights into what can be learned about cryptojacking and its damage, infrastructure, and evasion techniques. Our work complements past measurement studies on malicious web pages [37], and provides useful recommendations to browser-based security enhancements [6].

Contributions. In summary, we make the following contributions in this paper:

- We design and implement a detector, named CMTracker, to detect cryptojacking with high precision and coverage (much higher coverage than prior work as discussed in Section 6.2).
- By crawling and visiting the Alexa top 100K websites (and following several links per website), CMTracker locates 2,770 cryptojacking pages. We estimate the damage of cryptojacking, showing that it costs more than 278K kWh extra power daily, and attackers are earning at least 59K US dollars daily.
- We systematically study the infrastructure of cryptojacking attacks. For instance, we analyze various aspects of the attack domains and the behaviors of scripts. Our results provide valuable insights into the innerworkings and ecosystem of cryptojacking.

Roadmap. The rest of the paper is organized as follows: we first provide background and motivation in Section 2. Then we describe our approach to identify cryptojacking websites in Section 3. Section 4 reveals the landscape and impact of cryptojacking, and Section 5 describes the infrastructure of malicious miners. In Section 6 we study the evasion techniques used in cryptojacking, and we further give four case studies in the real world in Section 7. We provide the mitigation approaches and discussion in Section 8, and summarize related work in Section 9. Section 10 concludes our work.

2 BACKGROUND AND MOTIVATION

Cryptocurrrency Mining. Cryptocurrency is a type of digital assets designed to work as a medium of exchange that uses cryptography to secure its transactions [34]. With the growing popularity of cryptocurrency like Bitcoin [4], they become an important type of digital assets, and can be easily exchanged to hard cash in the real world. To make money, hundreds of millions of people exert their efforts to cryptocurrency mining. Successful miners obtain new cryptocurrency as a reward through the mining process. The nature of mining is to solve complicated mathematical problems which requires processing a large number of hash-like computation workload. The efficiency of mining for individuals can be improved by employing either more powerful computing platforms (e.g., powerful servers with high frequency CPU/GPU), or introducing more computing resources (e.g., distribute the workload with multiple PCs).

Cryptojacking. Recent reports [2, 27] showed that malicious adversaries were utilizing web users' CPU resources to mining cryptocurrency by injecting malicious payloads into the compromised websites. This new phenomenon, named cryptojacking, is on the rise likely due to the boosting market value of cryptocurrency. For example, as reported by Adguard [2] from November 2017, 220 cryptojacking websites have already been discovered from Alexa top 100K list. Meanwhile, Whorunscoinhive [33] reported that the number of websites using Coinhive (most popular cryptocurrency mining scripts) has increased by 31.7% within only one month. Although the above mentioned reports highlight the emerging threats about cryptojacking, little has been discussed about its severeness, infrastructure, and technical characteristics.

Although an in-depth understanding to answer such questions in cryptojacking is very necessary, achieving this goal is by no

means trivial. To get the ground truth of cryptojacking in real world, the first challenge is how to precisely identify those cryptojacking scripts in websites in a fully-automatic way. Most existing approaches apply naive strategies to detect cryptojacking, which can be easily evaded by cryptojackers. For example, a keyword based search of cryptocurrency mining scripts [28] can be evaded by simple code obfuscation techniques. As another example, although cryptocurrency mining may take high CPU usage of its hosted platform, attackers can easily configure their mining tasks to stay under certain limit. In fact, in our research, we have already discovered many such cases, as further discussed in Section 6 with a set of case studies (Section 7).

Giving the above mentioned challenges in our study, we first propose a novel approach to detect cryptojacking scripts in a fully-automatic way, by leveraging a set of characteristics resistant to current evasion techniques. Our approach achieves high precision in identifying cryptojacking scripts in real world. Further, using this approach, we are able to discover 2,770 cryptojacking scripts from top 100K Alexa lists and their direct external links. We discovered 260% more scripts, as compared with the latest reports in 360.cn [1] in February 2018. Additionally, the amount of cryptojacking scripts detected by our approach is significant higher than blacklist based approaches (See Section 6.1).

3 CRYPTOJACKING IDENTIFICATION

In this section, we depict our methodology to detect cryptojacking websites. First, we introduce the dataset used for conducting our large-scale study. Then we illustrate the two types of behavior-based approaches for dynamically discovering cryptocurrency mining pages. Those automatically identified websites go through a further verification to determine whether they are indeed cryptojacking websites.

3.1 Sample Collection

To conduct our measurement analysis over a large-scale and real-world dataset, we focus on most popular websites and their direct external links (first level hyperlinks in these webpages). Specifically, we first crawl Alexa top 100K websites and their sub-domains. The intuition here is that malicious mining pages tend to maximize the profit gain, by injecting malicious payload into those frequently accessed pages. As a result, to speed up our sample collection, we follow only the *internal links* (including sub-domains) from the homepage of each website. In addition, we also follow the direct external links from each homepage, since some malicious miners tend not to directly inject malicious payload to the top ranked pages, instead, they "advertise" the malicious pages on the top websites, and bait the users to visit a malicious page. Because of time and resource limitation, we randomly accessed 20% of the internal/external links (subject to 3 links maximum per subdomain, e.g., maps.google.com and news.google.com). In general, following the above-mentioned process, we collected a total of 853,936 webpages as our dataset for locating cryptojacking webpages.

3.2 CMTracker Design

Figure 1 depicts our overall architecture to locate pages involved in cryptojacking. First, we collect the above mentioned 853,936 webpage samples as the dataset for our measurement study. We leverage Chrome Remote Interface [11] to record and profile the visited pages by stack sampling. Second, we locate the cryptocurrency mining pages by leveraging two behavior-based profilers. Then, an additional verification step is processed to exclude those minor benign cryptocurrency mining pages (that explicitly inform the user about the mining activities).

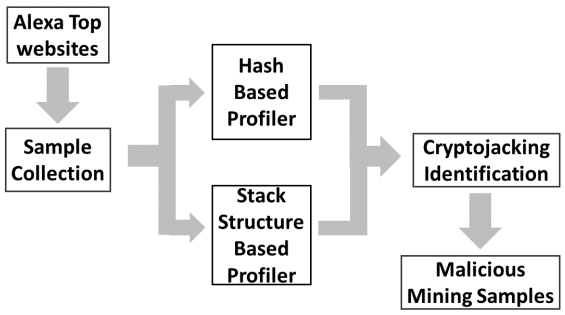

Figure 1: Work-flow for identifying cryptojacking webpages

Note that the design of CMTracker is to efficiently detect existing cryptojacking webpages, and to provide important ground truth for our further measurement study. We do not guarantee to defeat all existing or future evasion techniques. Towards this goal, our approach successfully identified nearly three times more cryptojacking domains, compared with latest reports [2].

3.3 Automated Mining Scripts Discovery

Hash Based Profiler. The core functionality of cryptocurrency miners is a proof-of-work system. Normally, most of their computing workloads are hashing. For example, Bitcoin-liked applies the double SHA-256 to verify the transactions, and other cryptocurrency miners widely use Scrypt, which is a hash algorithm used in cryptocurrency [5, 31]. Based on this observation, our hash-based profiler focuses on the low-level hash functions. We annotate nine common accessible hash library interfaces, which are identified by a set of fixed signatures (e.g., "cryptonight_hash","sha256","crypto") from multiple open-sourced cryptocurrency or commercial mining services. Then, we calculate the cumulative time of the websites they spent on hashing to identify whether a webpage is mining. As normal websites usually spend very little time on processing hashing functions (e.g., as shown in Figure 2, 99% of top 100 Alexa websites take less than 0.47% of the whole execution time for hashing). On the other hand, cryptocurrency mining scripts spent most of their time on hashing. In practice, if a webpage uses more than 10% of its execution time on hashing, our profiler reports it as a cryptocurrency miner.

Stack Structure Based Profiler Although the hash-based profiler is straightforward and precise in identify most mining scripts, further discovery showed that they can also be easily evaded by cryptojacking attackers with code obfuscation techniques (Section 6.2). As a result, we propose the stack structure based profiler as a complementary detector.

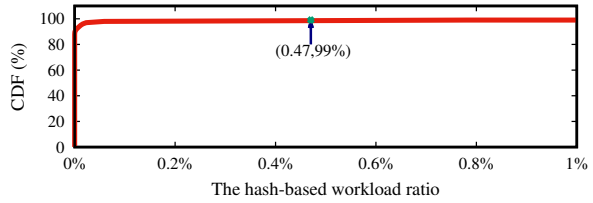

Figure 2: The hash-based workload ratio of Alexa top 100 websites. The ratio is calculated by dividing the total execution time by the time spent on hash-based computation

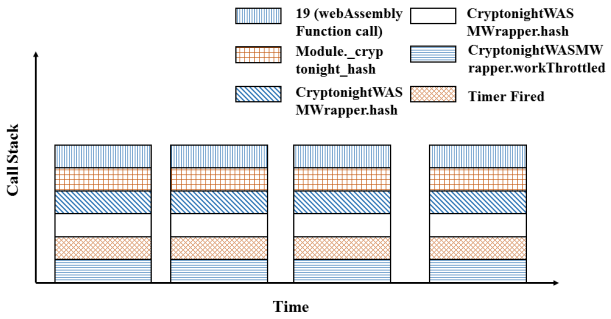

Figure 3: Execution stack of a cryptocurrency mining sample over a period of time. The x-axis is the timeline during execution and y-axis indicates the running threads in the stack

The key observation is that cryptocurrency miners run heavy workloads with repeated behavioral patterns revealed by their execution stack, which can be utilized as an important tip for identifying the existence of cryptocurrency mining scripts at runtime. Figure 3 illustrates the call stack transition of a sample mining script. As can be seen from this figure, both the stack depth and call chains of the mining scripts are repeated and regular. On the contrary, Figure 4 shows that a normal webpage rarely repeats the same calling stack for more than 5.60% of the execution time. Thus, we profile and record the calling stack of evaluated webpages. Since cryptocurrency mining is heavy, to avoid anything noticeable by the user, most mining tasks won't take place at the main thread when loading the webpage. Instead, they prefer to create one or more dedicated threads. Thus, if a dedicated thread repeats its call chain periodically (in a fixed time internal), and the call chain occupies more than 30% of the whole execution time in this thread, we report it as a cryptocurrency miner.

3.4 Cryptojacking Identification

The above steps help to automatically identify scripts which contain cryptocurrency mining payloads, however, they are not necessarily malicious. Normally, benign webpages mine the cryptocurrency under a certain type of user agreement. As a result, for each identified webpage by one of the two profilers, we extract all textual content and check if there exists any type of related user agreement, based on a set of pre-defined keywords (e.g., "mining agreement"). Although this setting is a little bit coarse, it is effective enough for us to filter out most benign webpages. Actually, our further analysis showed that only 35 of identified webpages are benign and the rest are indeed malicious.

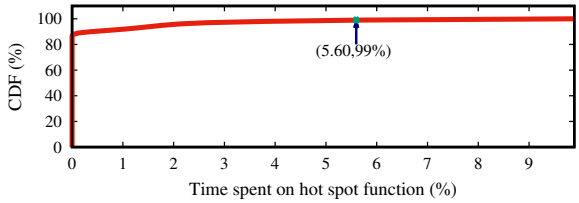

Figure 4: Proportion of time spent on hot spot function from top 100 Alexa websites

3.5 Effectiveness

CMTracker identifies 2,770 domains in total that contain cryptocurrency mining scripts from the pre-mentioned dataset. Among them, 868 belong to Alexa top 100K websites, and the rest of 1,902 webpages belong to external links other than top 100K websites. We further give a detailed analysis of this identification results, as shown in Section 4.

To evaluate the effectiveness of CMTracker, we randomly select a subset of identified cryptojacking samples and conduct a further manual verification process. Specifically, we inspect 200 webpages, to check whether they are indeed running cryptojacking scripts without user consent. The reason why we do not evaluate CMTracker with blacklists is that they are both incomplete (FNs) and inaccurate (FPs), and cannot be used as ground truth. Our manual verification showed that none of the identified webpages are false positives. We believe such a high precision is a result of our conservative detection thresholds (e.g., limited hash signatures and 30% time-consuming in stack threads). This result can be considered a lower bound of cryptojacking in the real world.

Note that we acknowledge there may be cryptojacking pages that CMTracker misses. However, to the best of our knowledge, neither we nor other publicly available reports showed any evidence of real cases that can escape CMTracker's two behavior-based detectors.

4 BREAKDOWN OF CRYPTOJACKING SCRIPTS

In this section, we report the breakdown of 2,770 identified cryptojacking samples. We first show their over distribution among all websites we detect, including a comparable analysis with other prior reports, as well as their distribution in different webiste categories. Then we give a coarse-grained analysis about the impact of these cryptojacking scripts regarding the profit and energy consumption aspects.

Domain Category	# Identified Domains	# Visited domains
Top 100K	868	99,964
External Link	1,902	448,660
Total	2,770	548,624

Table 1: Cryptojacking landscape detected by CMTracker

4.1 Overall Distribution

Landscape. As can be seen from Table 1, among all top 100K Alexa websites, CMTracker identifies 868 unique domains that contain cryptojacking. Additionally, it also detects another 1,902 cryptojacking domains by traversing all the available external links, including 448,660 distinct domains from top 100K Alexa websites. In total, our CMTracker successfully identified 2,770 cryptojacking cases from 548,264 distinct domains. The highest ranked Alexa site hosting cryptojacking is *thepiratebay.org* a rank of 125.

It is also worth noting that the number of cryptojacking cases detected by CMTracker achieves a much higher scale, compared with the latest available prior public reports. As shown in Table 2, the number of identified cryptojacking domains is over 260% more than the last public report from *360.cn* [1] which was published only two months earlier (Feb. 2018). In comparison, the report from *360.cn* only detected 10% more cryptojacking websites compared with another report from AdGuard[2], which was published four months prior to it. As a result, we can conclude that the number of cryptojacking cases are rapidly boosting.

# Report Name	Top 100K	Report Date	Incremental
AdGuard	220	Nov. 2017	-
360.cn	241	Feb. 2018	10%
CMTracker	868	Apr. 2018	260%

Table 2: Comparison of latest available public reports

Websites Category	# Websites with Scripts	Percentage (%)
Art and Entertainment	752	27.1
Adult	360	13.0
Internet and Telecom	323	11.7
Business	182	6.6
Game	180	6.5
Others	973	35.1
Total	2,770	100

Table 3: The distribution of cryptojacking domains based on the category of their host websites

Categories. As shown in Table 3, among all cryptojacking domains we discovered, nearly half of malicious samples (49%) are *Art and Entertainment* and *Adult* websites. Among them, most provide pirate resources (i.e. free movies or cracked games). Such types of web contents are attractive to users who may stay for an extended period of time while searching for resources. Compared with other normal websites (e.g., a landing page of a company), websites in such categories can obviously bring more profits for cryptocurrency mining attackers.

4.2 Impact of Cryptojacking Scripts

We also measure the severeness of existing cryptojacking scripts on the web by raising the following two research questions: First, how many profits do these cryptojacking scripts help malicious adversaries gain? Second, how much extra power do they cost to earn the profits?

Domain	Visitors per Month (#)	Duration (s)
www.thepiratebay.org	211.47M	326
www.cinecalidad.to	34.28M	272
www.primewire.is	11.86M	449

Table 4: Sample statistics of visitors and duration from similarweb [32]

Profits gain. The motivation of cryptocurrency miners is to gain profits. Thus it is important to get an estimate on the scale of the cryptojacking activities. Here, we use the following formula to measure the profit approximately and conservatively. It is directly borrowed from Coinhive [8], the largest web-based cryptocurrency mining provider.

$$Profit = \sum \frac{\#Visitors \times Duration \times HashSpeed}{Difficulty} \times Reward$$

Variable	Value	Description and Source
Visitors	211.5	Num. of visitors millions/month [32]
Duration	326s	Avg. stay per visit in seconds [32]
HashSpeed	50 hash/sec	Based on avg. CPU power (32.5w) [8]
Difficulty	56.2G hash	Current diff. for mining Monero [8]
Reward	$1,095	Based on the block reward of 4.74 [8] and $221 Monero price [7]

Table 5: Values and their corresponding sources for calculating revenue of a single cryptojacking website *www.thepiratebay.org*. All statistics are collected as of May 9th, 2018.

Our revenue estimation is based on *Monero*, a most popular cryptocurrency widely mined by cryptojacking attackers. The statistics of these variables are acquired from different reliable sources, as shown in Figure 5. $HashSpeed$ is the average hashing speed of users' processors, $Difficulty$ is the difficulty to mine a block of cryptocurrency, $Reward$ is the profit gain of each mined cryptocurrency block. For each malicious mining page we detect, we obtain its number of visitors per month and their average stay duration from *similarweb.com* [32], as a set of samples shown in Table 4. In total, our basic approximation shows that malicious miners can gain more than 1.7 million US Dollars, from more than 10 million users per month.

Energy Consumption. We also estimate the extra energy consumption that these websites cause to the web users and their machines. Specifically, we use the following method:

$$Energy = \sum \#Visitors \times Duration \times Power$$

In here, $\#Visitors$ is the number of visitors of each identified domains, $Duration$ is the average duration that a visitor stays. $Power$ is the CPU Power available for mining of the browsers.

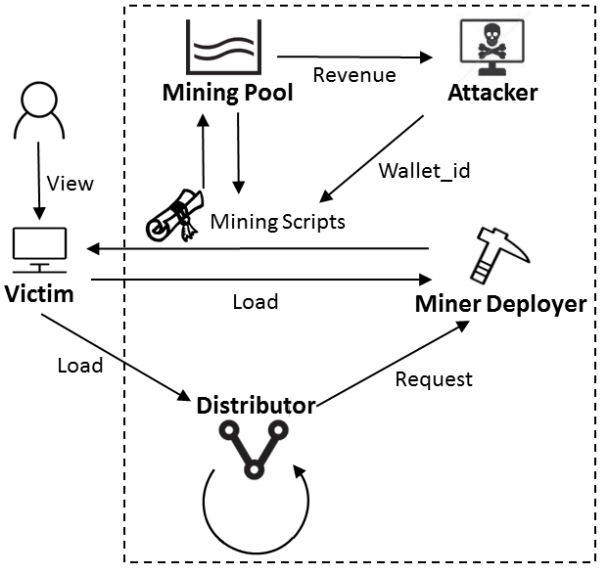

Figure 5: Mining participants involved in cryptojacking

Similarly, for each malicious miner, we crawl its number of visitors per month and their average stay duration from similarweb.com. The average computing power of a browser is based on 50% capacity of the mainstream CPU (Intel i5, 65w [19]) in desktop PC, which is 32.5W. In total, our experiment shows that the malicious mining pages consume at least 278K kWh electricity energy per day, which is equal to the electrical energy consumption of 9.3K residential customers in the United States [3].

5 INFRASTRUCTURE OF MALICIOUS MINERS

In this section, we first introduce our methodology to annotate various parties in the process of cryptojacking. Then, we study the infrastructure of the malicious miners from several perspectives: first, how many attackers distributed these malicious payloads? It could be powerful hacker groups that compromise a large number of websites with their injected script, or it could be individual hackers or website owners themselves who deployed the malicious payload. Is is unclear which case it is. Interestingly, our study shows that there is no large hacker group involved in cryptojacking based on our domain and wallet ID analysis. Second, what is the life cycle of the malicious mining domains? Do the attackers rapidly change their domains?

5.1 Mining Participants Definition

Illustrated by Figure 5, to annotate various participants of malicious mining, at least four parties are involved to distribute the malicious miners. Here, we first give a detailed definition about these parties, as listed below:

- **Attacker** are malicious adversaries who utilize the client machines as mining infrastructure for profit. Their scripts running on the clients are configured with their unique wallet IDs, indicating who will receive the cryptocurrency rewards.

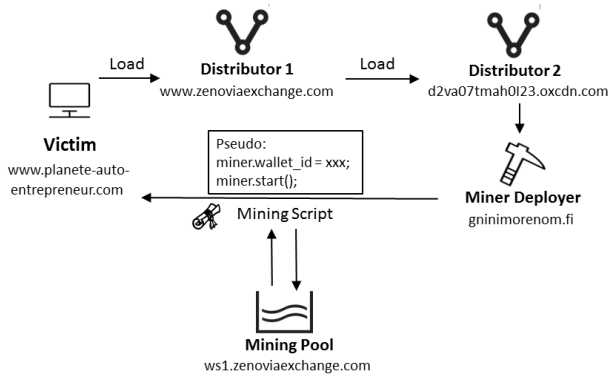

Figure 6: Mining flowchart of an real-world example

- **Miner Deployer** are domains or servers that host mining scripts for cryptojacking. These scripts are either crafted by attacker themselves, or copy-pasted from other public available sources (e.g., Coinhive). The Miner Deployers can be easily replaced by the attacker to evade detection, for example, using an alternative mining scripts (changing from Coinhive to Coin-have).
- **Mining Pool** are domains or servers that distribute mining tasks (confirming the correct hash results), and return revenue (cryptocurrency) to miners. From our analysis, we find that mining pool and miner deployers can often be the same party (e.g., Coinhive).
- **Distributor** are the intermediate domains that act as redirectors to reach the final destination of mining scripts (this is optional for cryptojacking). Like many redirectors such as proxy servers, attackers usually change such domains frequently to make sure they are not on any domain or URL blacklist.

Given the definition of these different parties in cryptojacking, once a victim is browsing a malicious mining page, the malicious mining scripts should be fetched from the **Miner Deployers** assigned by **Attackers**. In addition, some malicious miners deploy several levels of **Distributors** through domain redirection, so that they can easily change the URL of **Miner Deployers** to evade detection. Lastly, the mining tasks are assigned by the **Mining Pools**, which then generate revenue to **Attackers** when completed. Figure 6 depicts a real-world example in an automobile business website *http://www.planete-auto-entrepreneur.com/*. In this website, malicious mining scripts that resides in **Miner Deployer** *gninimorenom.fi* finally reached a victim's browser through 2 redirected domains by the **Distributor**: *ad2va07tmah0l23.oxcdn.com* and *www.zenoviaexchange.com*. Its mining script employs a **Mining Pool** from *ws1.zenoviaexchange.com*. In addition, we find that sometimes miner deployers and mining pools mining are offered together as a one-stop shop which we call **Mining Services**. They provide easy-to-use cryptocurrency mining interfaces, but charge a commission fee from 20% to 30% of the mined cryptocurrency; therefore, some attackers will choose to build their own private mining services instead.

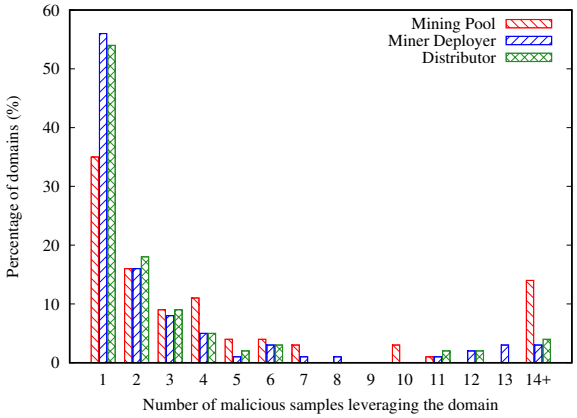

Figure 7: Domain distribution of Mining Pools, Miner Deployers and Distributors

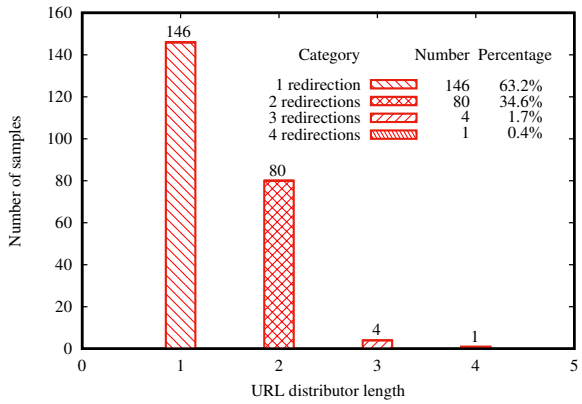

Figure 8: Distributor length distribution. The x-axis is the length of Distributor and the y-axis is the number of websites

Participants Identification. In our study, these participants are finally represented by a set of domains. To identify them, we first record the requests of all visited webpages. The domains of **Miner Deployers** can be directly identified from the request URLs which load the detected cryptojacking scripts. If these cryptojacking scripts are fetched by multiple requests, we consider the **Attacker** employs **Distributors**. Thus, the domains of **Distributors** can be identified from the URLs of the requests prior to the final one. Besides, after monitoring some real-world cryptojacking samples, we find that WebSockets are widely used in the data transformation between cryptojacking scripts and mining pool[12]. Specifically, since a **Mining Pool** uses standard WebSocket requests to communicate to the deployer, we identify the domains of **Mining Pools** by domains that belong to WebSocket requests. Note that WebSocket could be used by other services and thus cause false positives. Fortunately, from our manual analysis, they are in fact used only occasionally (mostly due to online chatting). Lastly, **Attackers** are identified by the parameter value of wallet_id from cryptojacking scripts, a constant string that indicates the cryptocurrency owner of the mining process. For example, the cryptojacking script from *Coinhive*, the most popular mining service provider, sends its wallet id through WebSocket to the **Mining Pool** in the first packet.

Experimental Setup. To understand the infrastructure of malicious miners, we randomly select some samples from our results, and conduct further investments to study the distribution of the participants and the life cycle of different miners. Actually, because the limitation of our computing server, we can only continuously monitor 1,000 samples. Besides, we re-visit these samples every 3 days and study the life cycle of each sample. We detail the findings of this part in the following section.

5.2 Mining Participants Distribution

To understand who is responsible for the cryptojacking in websites, we study the distributions of different mining participants. Actually, we count the occurrence of **Miner Deployer**, **Distributor** and **Mining Pool** on the first day of our experiment. As a result, we obtain two findings.

Observation 1: Most malicious miners are not centrally controlled.

Figure 7 depicts the distribution of either **Miner Deployers**, **Distributors**, or **Mining Pools**. Surprisingly, most participant domains (60% **Mining Pools**, 81% **Distributors** and 80% **Miner Deployers**) occur in no more than three malicious samples. In contrast, only a small proportion (15% **Mining Pools**, 8% **Distributors** and 9% **Miner Deployers**) are spotted in over 10 webpages. This result invalidates the hypothesis that malicious miners are controlled centrally, thus limiting the effectiveness of blacklists.

Observation 2: Mining services and advertisers facilitate most cryptojacking websites.

A further analysis is applied to the frequently used domains. Table 6 shows the most common domains for **Miner Deployers** and **Mining Pools** respectively. Among them, Coinhive, together with Netflare, and Directprimal, are cryptocurrency mining services, According to their descriptions, they are designed to provide alternatives to micro payments, artificial wait time in online games, intrusive ads and dubious marketing tactics. By investigating these platforms, we find their terms of service quite vague, and almost none of them claims that miners should request users' agreement. Note that even these mining services often come with a hefty commission fee (from 20% to 30%), more than 57.2% of the websites choose these services for convenience.

Interestingly, *Zenoviaexchange* used to be an advertising service provider. We also notice that many other cryptojacking attacks are also initiated by advertising services. For example, adstour.com, freecontent.loan, and popads.net mine cryptocurrency without any user notification. Compared to regular ads, malicious mining is more stealthy and apparently profitable, thus attracting the attention of traditional advertising services. To get a sense of how many advertising services conduct cryptojacking, we check easylist [13] (a filter list that contains advertisement domains and URLs) against our results and find that 20% of the cryptojacking domains are marked as advertisements. We further investigate the overlapping

		Miner Deployers			Mining Pools			Overlap		
	Domain		Number	%		Number	%		Number	%
1	coinhive.com	√	442	44.2	√	518	51.8	√	442	44.2
2	advisorstat.space	√	75	7.5	√	75	7.5	√	75	7.5
3	minescripts.info*	√	33	3.3		-	-		-	-
4	netflare.info*		-	-	√	33	3.3		-	-
5	directprimal.com†		-	-	√	21	2.1		-	-
6	zenoviaexchange.com	√	13	1.3	√	13	1.3	√	13	1.3
7	cryptoloot.pro†	√	12	1.2		-	-		-	-
8	Others		425	42.5		340	34.0		-	-

Table 6: Top Miner Deployer and Mining Pool domains and their intersection. Domains marked by * or † belong to a same service provider

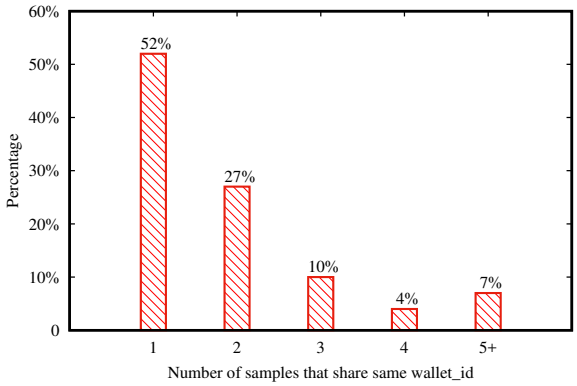

Figure 9: Distribution of the Coinhive wallet IDs

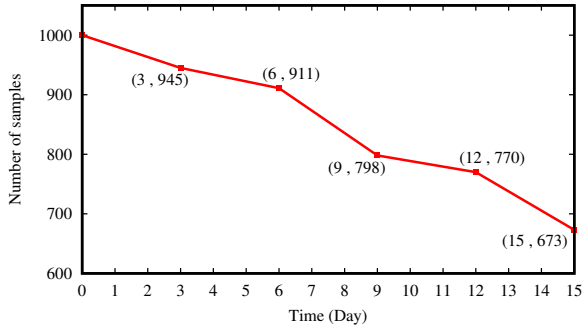

Figure 10: Life cycle of samples. About 1/3 samples vanish in 15 days

domains between **Miner Deployers** and **Mining Pools**. Surprisingly, once a cryptojacking website utilizes a popular mining service (i.e. *coinhive.com*, *minescriptsinfo*, and *cryptoloot.pro*) as the **Miner Deployer**, they, without exception, also adopt the same service as their **Mining Pool**. This observation also applies to the advertising services (*zenoviaexchange.com*). We further analyze these **Miner Deployers** and notice that they all provide a default **Mining Pool** service.

To distribute the URLs of the **Miner Deployers**, 231 malicious miners apply at least one or more levels of **Distributors**. Figure 6 shows a real-world example. In this case, the cryptojacking webpage fetches a distributor, and cascadingly fetches another. Then, the URL of the **Miner Deployer** is loaded in the second distributor. We summarize the length of distributor chains in Figure 8. More than 80 samples use multiple distributors to fetch the address of **Miner Deployers**, and an extreme case adopts a chain of four domains.

5.3 Distribution of Wallet IDs

Our results above reveal that the distribution of various mining participants (**Miner Deployers**, **Distributors**, and **Mining Pools**) is sparse, i.e., only a few domains are heavily used in many malicious samples. These heavily utilized domains (e.g., Coinhive), as depicted in Section 5.2, are mostly cryptocurrency mining services. From this information alone though, it is difficult to know who are abusing these services. Thus, we further study the beneficiaries of the malicious miners.

Observation 3: A significant number of attackers benefit from abusing cryptocurrency mining services.

To identify the beneficiaries, we utilize a programming pattern of Coinhive where the malicious payload should explicitly specify the beneficial wallet id. Specifically, we analyze the malicious scripts and extract the wallet ids automatically. As depicted in Figure 9, a wallet ID is commonly associated with less than three malicious webpages. In addition, Coinhive discourages collecting profits from multiple wallets, and prohibits the reward claim of wallets with small value. Thus, it is likely that many malicious samples are deployed by a single party. Only a few wallet IDs are associated with a large number of cryptojacking pages. An extreme case is Piratebay, a famous illegal online index of digital content (i.e. pirate entertainment media and software). Although it was already reported to abuse user browsers since September 2017 [18], we still witness 43 samples that contribute to its wallets.

5.4 Life Cycle of the Malicious Miners

Observation 4: The malicious samples disappear or update frequently.

Figure 10 shows the life cycles of the cryptojacking pages. Among our evaluated samples, about 20% vanish in less than 9 days. This

	Miner Deployer			Distributor				Mining Pool		
Duration	Unchanged	Migrated	Vanish	Unchanged	Migrated	Vanish	Added	Unchanged	Migrated	Vanish
Day 0 - Day 3	868	77	55	209	4	18	10	920	25	55
Day 3 - Day 6	823	88	34	195	8	20	9	889	22	34
Day 6 - Day 9	752	46	113	129	7	76	3	773	25	113
Day 9 - Day 12	697	73	28	113	8	18	4	742	28	28
Day 12 - Day 15	604	69	97	74	6	45	9	652	21	97

Table 7: Life cycle of various participant domains. For each types of domains, we calculate: 1)how many domains keep unchanged; 2) how many domains migrate to a new URL; 3) how many domains vanish; 4) how many domains are newly added

result illustrates that the malicious miners typically have short life cycles. In addition, to better understand their update frequency, we look into the domains of **Miner Deployers**, **Mining Pools**, or **Distributors**. As shown in Table 7, despite those disappeared samples, the live samples frequently change their domains. Specifically, over 21% **Miner Deployers** migrated to new domains in only nine days. Even though at a lower frequency, we find that **Distributors** also migrate. Interestingly, blacklists do not target distributor domains even though they rarely change (See Section 6.1).

6 DETECTION AND EVASION TECHNIQUES

Some browser extensions claim to block the cryptojacking scripts (e.g., through a dynamically updated blacklist of malicious domains) [2, 22, 26, 36], and a few anti-virus engines detect malicious javascripts. The question is whether they can protect users from this threat effectively? It is especially questionable given that cryptojacking participants can also apply evasion techniques. Understanding the evasion patterns of the attackers is important to guide an effective detection/protection solution. This section analyzes the cryptojacking attacks from two perspectives: first, we evaluate how effective the state-of-the-art detectors are. Then, we analyze the evasion techniques and show their effect on anti-virus engines.

6.1 Effectiveness of Blacklists

Observation 5: State-of-the-art mitigations, for example, blacklists, are insufficiently to locate cryptojacking in time.

To the best of our knowledge, the most popular mitigations currently deployed are based on blacklists. In this section, we collect two popular blacklists, and evaluate their effectiveness. One is NoCoin [22], the most popular cryptojacking blocker in Github.com (with 1,469 user favorites). Another widely used blacklist is MinerBlock [36], which has 369 user favorites. As illustrated by Table 8, less than 51% malicious attacks are detected. Since multiple cryptojacking participant domains need to collaborate to complete a malicious mining progress, as long as one of them on a chain is detected, the entire operation is thwarted. Surprisingly, as shown in Table 8, all blacklists focus on **Miner Deployers** or **Mining Pools**, while none of them detects the **Distributors**. However, our experiments above show that malicious attackers rarely change their **Distributors**. Thus, it could be an effective solution to add **Distributors** to the blacklists. Interestingly, we notice that NoCoin is actually less effective than MinerBlock, which does not match their respective popularity. In addition, the Github logs show that these blacklists are typically updated every 10 to 20 days [21, 35], which is slower than the cryptojacking domains. Figure 11 illustrates the impact of blacklist update. The results show that although the update improves the coverage of the MinerBlock blacklist, the overall detection rate is still unsatisfactory because of the high churn of cryptojacking domains as described previously (with many domains vanishing or migrating).

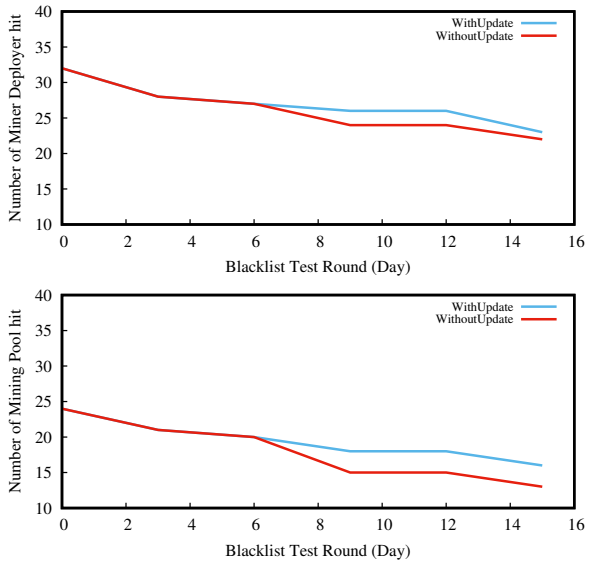

Figure 11: Impact of a blacklist update that occurs on MinerBlock. The hits of the old blacklist and the updated one are presented separately. The blacklist updates at the ninth day of our experiment

6.2 Evasion Techniques

Among the 1,000 tracked samples in Section 5, we manually analyzed 100 cryptojacking websites, and list the overall results in Table 9. We find three evasion techniques commonly applied: First, most mining pages avoid taking 100% of the CPU resources. Second, many malicious pages obfuscate the malicious payload. Finally, some pages hide malicious code into a popular 3rd party library. In the following, we go through each of these evasion techniques.

Limiting CPU Usage. To maximize the profit gain, some cryptojacking webpages exhaust the CPU resources. However, users

	NoCoin				MinerBlock			
Day	Deployer Hit	Distributor Hit	Pool Hit	Overall Hit	Deployer Hit	Distributor Hit	Pool Hit	Overall Hit
0	6.3% (10/159)	0.0% (0/64)	6.1% (6/98)	26.0% (61/235)	20.1% (32/159)	0.0% (0/64)	24.5% (24/98)	46.0% (108/235)
3	4.8% (8/168)	0.0% (0/56)	6.5% (6/93)	29.8% (75/252)	16.7% (28/168)	0.0% (0/56)	22.6% (21/93)	44.9% (113/252)
6	4.5% (6/132)	0.0% (0/52)	6.4% (6/94)	25.4% (50/197)	20.5% (27/132)	0.0% (0/52)	21.3% (20/94)	43.2% (85/197)
9	5.1% (6/117)	0.0% (0/43)	5.6% (5/90)	26.3% (46/175)	20.5% (24/117)	0.0% (0/43)	16.7% (15/90)	43.4% (76/175)
12	6.8% (7/103)	0.0% (0/46)	5.4% (5/92)	30.5% (47/154)	23.3% (24/103)	0.0% (0/46)	16.3% (15/92)	48.7% (75/154)
15	7.2% (6/83)	0.0% (0/35)	6.0% (5/84)	35.9% (46/128)	26.5% (22/83)	0.0% (0/35)	15.5% (13/84)	50.8% (65/128)
Average	5.8%	0.0%	6.0%	29.0%	21.3%	0.0%	19.5%	46.1%

Table 8: Effectiveness of blacklists. The results of two popular blacklists are presented in two separate blocks. For each blacklist, the first three columns show its detection rates on Miner Deployers, Distributors, and Mining Pools separately, and the fourth column shows the detection rates of end-to-end cryptojacking operations (we consider an operation to be blocked as long as one of the domains in the operation is blocked)

Evasion Techniques Category	Used in Samples #
Limiting CPU Usage	56
Code Obfuscation	26
Payload Hiding	43
Total	100

Table 9: Statistics of evasion techniques

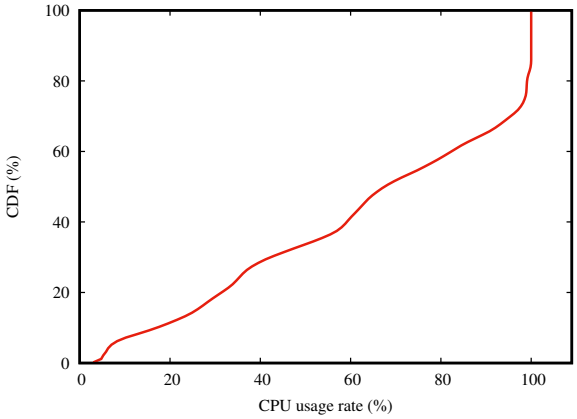

Figure 12: The distribution of CPU usage

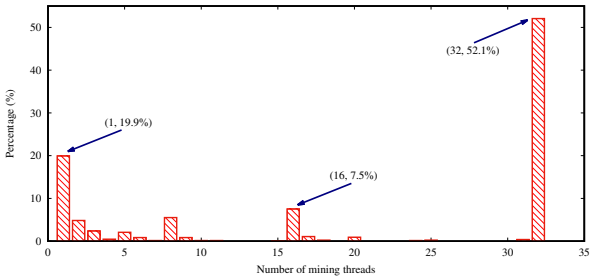

Figure 13: The distribution of mining threads. The results are collected on a Ubuntu 16.04 server, with four 8-core 2.0 GHz CPU and 32 GB memory

may experience obvious lags when visiting such pages. Also, such a behavior is extremely easy to detect by automated methods.

Perhaps unsurprisingly, most cryptojacking scripts do not go full speed. As depicted in Figure 12, most of the cryptojacking samples (70%) set such an attribute, named "throttle", and the throttle values vary from 90% to an extremely low number, 3%. Considering that the average CPU usage of a webpage is 5% [24], it is quite difficult to detect such a sample by only its CPU utilization. Since state-of-the-art processors have multiple cores, it is also common for cryptojacking scripts to take advantage of them with multiple threads. However, like exhausting the usage of a single processor, the creation of too many threads may also degrade the performance of a victim's browser. Thus, many samples not only set the throttle value for a single thread workload, but also limit the number of threads. Figure 13 illustrates the distribution of core utilization in cryptojacking samples. Only less than 20% of the samples use a single thread, indicating that most attacks execute the workload in parallel. Among them, only about 52% malicious pages exhaustedly create as many mining threads as the number of processors, that is, 48% malicious pages restrict the number of mining threads.

Code Obfuscation for Mining Scripts. 26 of the 100 analyzed samples obfuscate their code to hide their malicious intent. Figure 14 illustrates a real-world example. The malicious payload is decoded at Line 3, and executed at Line 10. Such an evasion technique hinders manual analysts or static analysis from understanding the malicious payload. Interestingly, only 16 samples obfuscate the entire mining payload. There are 10 other samples that simply obfuscate the mining pool domain (not the logic itself). As a result, code obfuscation is frequently applied to the payload that distributes the malicious domains.

Payload Hiding. Instead of injecting malicious payload directly to the cryptojacking webpages, some attacks choose to hide their malicious code in 3rd party libraries. For example, attackers frequently inject their attack code into their own version of *jQuery.js*, which is a widely used JavaScript library [20]. As illustrated in Figure 15, the malicious code is appended to the original jQuery code, which gets triggered automatically when the jQuery is loaded.

Category	With evasion techniques		w/o evasion techniques		Total	
#	V.T./CMTracker	V.T. Cov.	V.T./CMTracker	V.T. Cov.	V.T./CMTracker	V.T. Cov.
Redirected/Forwarded Pages (Distributor)	5/19	26.3%	45/81	55.5%	50/100	50.0%
Cryptojacking Scripts Pages (Deployer)	4/16	25.0%	67/84	89.7%	71/100	71.0%
Cryptojacking Website Pages	11/39	28.2%	33/61	54.1%	44/100	44.0%

Table 10: Effectiveness of anti-virus engines. The three blocks depict results on three sets of cryptojacking samples respectively. For each block, the first column shows the the number of detected samples by either VirusTotal or CMTracker, while the second column highlights the detection rate of the anti-virus engines. As a conservative estimation, if any of the anti-virus engines on VirusTotal reports a given sample as either malicious or suspicious, we consider it detected by the anti-virus engines

```
1 # (a) Source code in cryptojacking web-pages
2
3       document.write(unescape('%3c...%63%6f%69%6e
            %68%69%76%65%0d%0a%3c%2f%73%63%72%69%70'));
4
5 # (b) After "unescape()" decoding:
6
7  <script src="https://coin-hive.com/lib/coinhive.min.
        js"></script>
8  <script>
9  var miner = new CoinHive.Anonymous('tByG...0exk');
10 miner.start();$coinhive
11 </script>
```

Figure 14: An example of code obfuscation

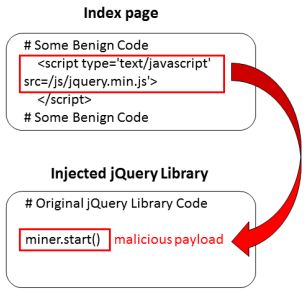

Figure 15: A code hiding example

Observation 6: Evasion techniques are effective against anti-virus engines

Our experiments above characterize several types of evasion techniques from the real-world cryptojacking webpages. Then, our further investigation uploads all the samples to VirusTotal, and observe whether the evasion techniques are effective against the anti-virus engines. Table 10 reveals that the evasion techniques effectively decrease the detection rate of these engines. Specifically, although about 44% cryptojacking websites can be detected by at least one anti-virus engines, the detection rate drops to 28% when considering only the scripts with evasion techniques are applied. The drop also applies to **Miner Deployers** and **Distributors**.

Overall, they are still insufficient in detecting cryptojacking webpages and are comparable to the effectiveness of blacklists (as show in Table 8).

Note that although these evasion techniques help many cryptojacking scripts escape detection, our behavior-based profilers can still catch them. This, in turn, indicates that CMTracker is much more effective than today's best available tools for discovering cryptojacking websites. The results reported by CMTracker serve as an ideal ground truth for in-depth measurement study.

7 CASE STUDIES FOR VARIOUS TYPES OF MINERS

In this section, we show four typical cases about cryptojacking webpages. The first case shows the most aggressive miner that makes full use of available resources. The second case shows a typical example where the malicious miner intentionally hides itself to avoid being noticed. The third case shows a stealthy miner that applies multiple evasion techniques to maximize its profit. The last case shows an adaptive miner which employs a platform-dependent mechanism for distributing its mining tasks. All these case studies demonstrate that although cryptojacking is clearly in the process of applying more and more sophisticated strategies.

```
1 # Many Normal Code
2 # Malicious Payload (Repeat 10 lines)
3   var _0xd1d168=["iframe","setAttribute","https://www.
        jqrcdn.download/lot.html"...;
4   var _0x7e5874=["iframe","setAttribute","https://www.
        jqrcdn.download/lot.html"...;
5   ...
```

Figure 16: Malicious payload in *bookstore.investmentu.com*

Case 1: Most Aggressive Miner. As previously discussed in Section 6, at the current stage, about 30% of our identified cryptojacking scripts exhaust the CPU resources completely to maximize profits. One example is *https://bookstore.investmentu.com*, an online bookstore. When a user visits this website, after a few seconds, the cryptojacking script in *bookstore.js* invoke all its miners using 32 threads on our test machine with 100% CPU usage. It has an even more aggressive behavior to escalate its priority if there are multiple browser tabs (processes) that are mining cryptocurrencies concurrently. Specifically, in Line 3 of Figure 17, the cryptojacking script uses "CoinHive.FORCE_EXCLUSIVE_TAB" to block other tabs from mining. Interestingly, none of the 67 anti-malware engines in VirusTotal reports this website as a suspicious one.

Case 2: Stealthy Miner. Some of our identified cryptojacking samples already take actions to avoid user attention. The most straightforward way is to limit the resource consumption. A good example is *http://filikulamo.to/*, an online video streaming website

which provides pirate video copies. Since the core function of this website is video playing which already takes more than 30% of the CPU usage, it takes only about 10% of the CPU for mining, which is very difficult for users to notice.

Case 3: Robust Miner. As discussed in Section 5, the whole process of cryptocurrency mining relies on the collaboration of multiple participants, including **Miner Deployers**, **Distributors**, as well as **Mining Pools**. If any of its parts becomes invalid (e.g., a URL for loading mining script is blocked by AdBlocker), the operation will fail. To overcome this uncertainty, *http://dlight.ir/* employs two different mining services (Coinhive and Crypto-Loot) to do its mining concurrently. As a result, any single failure does not kill the other mining operation.

```
1 # Cryptojacking Payload
2   if (!miner.isMobile()) {
3       miner.start(CoinHive.FORCE_EXCLUSIVE_TAB);
4 }
5 # Mobile Filter
6   Miner.prototype.isMobile = function() {
7       return /mobile|Android|webOS|iPhone|iPad|iPod|
            IEMobile|Opera Mini/i.test(navigator.
            userAgent)
8   }
```

Figure 17: Malicious payload and mobile filter in *http://www.planetatvonlinehd.com/dark-temporada-1/*

Case 4: Platform-dependent Adaptive Miner. We also observed that some cryptojacking miners employ a platform-dependent adaptive mining strategy to achieve a good trade-off between its revenue and exposure risk. Due to the limited power of CPU on mobile platforms, mining cryptocurrency in mobile browsers is unrealistic. At the same time, this can also create a negative browsing experience and expose suspicious activities to users. As a result, some cryptocurrency miners explicitly disable their mining scripts on mobile platforms. A sample code snippet of *http://www.planetatvonlinehd.com/dark-temporada-1/* is shown in Figure 17.

The above case studies illustrate a set of notable and interesting cases that employ different mechanisms in the process of cryptojacking. These cases indicate that like many previous discovered malicious scripts, cryptojacking is evolving towards more sophisticated techniques and operating infrastructures.

8 MITIGATION AND DISCUSSION

Our work studies cryptojacking, a widespread and serious issue that affects millions of users. To mitigate this new threat, this section provides some recommendations to browser developers and cryptocurrency mining services.

Behavior-based cryptojacking detection. This paper proposes a behavior-based approach to detect cryptojacking. Proven by our experiments, this method can effectively detect cryptojacking webpages. Thus, both browser extensions and anti-virus engines can leverage such an approach to detect and block cryptojacking pages. A possible obstacle to applying this technique is its performance overhead. However, to detect a cryptojacking page, our profilers require only a short time to collect the runtime behavior of a page. During our experiments, CMTracker monitors each webpage for less than three seconds. A crowdsourcing-based solution can amortize the cost.

Cryptocurrency mining services with explicit user notification. Our study reveals that some cryptocurrency mining services, such as Coinhive, are abused to launch cryptojacking in large scale. We observed that cryptocurrency mining services have not paid enough attention to avoid abuses. For example, the cryptocurrency mining scripts are executed without any user notification, and unfortunately users do not have an option to turn off mining when visit cryptojacking webpages. Given that more than half of webpages use popular cryptocurrency mining services such as Coinhive, we argue that the mining services should take more responsibilities regarding the notification to the web users (e.g. a popup window), and disable the mining process if the user chooses to deny its request.

9 RELATED WORK

Cryptocurrrency Security. With the rapid development of cryptocurrency ecosystem, cryptocurrency security has brought more and more attention by researchers. However, most of the existing studies focused on building a more secure ecosystem, by improving the design, architecture of cryptocurrency (e.g., Bitcoin). For example, Eleftherios et al.[23] proposed a novel Byzantine consensus protocol which leverages scalable collective signing to commit Bitcoin transactions irreversibly within seconds. Their design brings more efficiency for Bitcoin transactions without sacrificing its security guarantees. Some work focused on identifying threats and opportunities of mitigations in the cryptocurrency's architecture. Eyal et al.[16] presented a new type of attack for Bitcoin mining. They found that collusion can allow miners obtain more revenue than their fair share, and it can lead the Bitcoin system into a decentralized cryptocurrency. Reid et al.[30] studied the anonymity of Bitcoin system. They found that a mining pool may trigger a costly distributed denial-of-service (DDoS) attack to lower the expected success outlook of a competing mining pool. Further, Johnson et al.[15] explored the trade-off between these strategies with a series of game-theoretical models of competition between two pools of varying sizes. In comparison, our work is more focused on a particular real-world security threat of cryptocurrency. The most related work in this topic is Plohmann et al.[29] where they studied the security incident of a miner botnet. However, their research is not in the web context which is a much larger-scale problem that has not been analyzed.

Malicious JavaScript Detection. Our measurement study for cryptojacking are mainly based on techniques for JavaScript code analysis. Existing related studies usually adopt either static or dynamic analysis to identify the characteristics of malicious JavaScript. For dynamic analysis, JS. JSAND[9] extracted features of four different aspects (redirection, deobfuscation, environmental context and exploitation). They employed Naïve Bayes based approach to detect JavaScript malware samples that automatically distribute themselves on the victim machines through background downloading. For static analysis, Curtsinger et al. [10] presented ZOZZLE, a tool

that predicates benign or malicious JavaScript code by extracting features associated with the program's abstract syntax tree (AST). Specifically, part of the research focused on detecting malicious advertisement scripts by utilizing some unique characteristics (e.g., advertiser ID). For example, Zarras et al. [37] studied the safety of the advertisements and how users may be exposed to malicious content and their sources. However, the above-mentioned approaches are not directly applicable to our research, due to the unique characteristics of cryptocurrency mining. In comparison, our hash-based and stack structure based profiler (Section 3) are more efficient and precise in detecting cryptocurrency mining scripts dynamically. Meanwhile, we acknowledge that our research can further benefit from existing approaches for improving the coverage or precision for identifying malicious mining scripts at a larger scale.

10 CONCLUSION

This paper conducts the first systematic study on the scale and impact of cryptojacking attacks. To support automatically recognizing malicious behaviors, we design CMTracker which out-performs the state-of-the-art detectors, with two behavior-based runtime profilers. We collect 2,770 malicious samples from 853,936 popular web pages, and our manual verification over a subset shows that they are all true positives. We estimate the real-world damage of this threat to over 10 million web users and 278K kWh extra power daily, equivalent of the energy consumption of a small town with 9.3k people. We measure the organization, life cycle, and technical details of cryptojacking webpages. Our results show that a significant number of attackers benefit from such attacks, and existing mitigation solutions are ineffective in blocking cryptojacking. In addition, three types of common evasion approaches by cryptojacking webpages are discovered. Finally, we believe the study will prompt us to rethink existing mitigation mechanisms and propose new solutions against future threats like this. Besides, to facilitate further research in cryptojacking, we release CMTracker source code and cryptojacking websites list at https://github.com/deluser8/cmtracker.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of the paper. This work was supported in part by the National Natural Science Foundation of China (U1636204, 61602121, U1736208, 61602123) and the National Program on Key Basic Research (NO. 2015CB358800). Yuan Zhang was supported in part by the Shanghai Sailing Program under Grant 16YF1400800. Min Yang is corresponding author of Shanghai Institute of Intelligent Electronics & Systems, and Shanghai Institute for Advanced Communication and Data Science. Zhiyun Qian was supported in part by the NSF 1719147. Haixin Duan was supported by the National Natural Science Foundation of Chain (61472215).

REFERENCES

[1] 360Netlab. 2018. who is stealing my power web mining domains measurement via dnsmon. https://blog.netlab.360.com/who-is-stealing-my-power-web-mining-domains-measurement-via-dnsmon-en/.
[2] ADGuard. 2018. The State of Cryptojacking. https://crypto.adguard.com/.
[3] U.S. Energy Information Administration. 2017. How much electricity does an American home use? https://www.eia.gov/tools/faqs/faq.php?id=97&t=3.
[4] Bitcoin. 2018. bitcoin. https://bitcoin.org/en/.
[5] bitcoinlion. 2018. Cryptocurrency Mining Hash Algorithms. http://www.bitcoinlion.com/cryptocurrency-mining-hash-algorithms/.
[6] Nicholas Carlini, Adrienne Porter Felt, and David Wagner. 2012. An Evaluation of the Google Chrome Extension Security Architecture.. In USENIX Security Symposium (USENIX Security). 97–111.
[7] coingecko.com. 2018. Monero Price Chart US Dollar. https://www.coingecko.com/en/price_charts/monero/usd.
[8] Coinhive. 2018. coinhive. https://coinhive.com/.
[9] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2010. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In Proceedings of the 19th international conference on world wide web (WWW). ACM, 281–290.
[10] Charlie Curtsinger, Benjamin Livshits, Benjamin G Zorn, and Christian Seifert. 2011. ZOZZLE: Fast and Precise In-Browser JavaScript Malware Detection.. In USENIX Security Symposium (USENIX Security). 33–48.
[11] cyrus and. 2018. chrome-remote-interface. https://github.com/cyrus-and/chrome-remote-interface.
[12] deepMiner. 2018. deepMiner. https://github.com/deepwn/deepMiner.
[13] easylist. 2018. EasyList filter subscription. https://github.com/easylist/easylist.
[14] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. 2018. A first look at browser-based Cryptojacking. IEEE Security & Privacy on the Blockchain (IEEE S&B) (2018).
[15] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol.. In USENIX Symposium on Networked Systems Design and Implementation (NSDI). 45–59.
[16] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security. Springer, 436–454.
[17] Dan Goodin. 2017. Cryptojacking craze that drains your CPU now done by 2,500 sites. https://arstechnica.com/information-technology/2017/11/drive-by-cryptomining-that-drains-cpus-picks-up-steam-with-aid-of-2500-sites/.
[18] Alex Hern. 2017. Ads don't work so websites are using your electricity to pay the bills. https://www.theguardian.com/technology/2017/sep/27/pirate-bay-showtime-ads-websites-electricity-pay-bills-cryptocurrency-bitcoin.
[19] intel.com. 2017. Intel Core i5-7400 Processor. https://www.intel.com/content/www/us/en/products/processors/core/i5-processors/i5-7400.html.
[20] Jquery. 2018. jquery. https://jquery.com/.
[21] Keraf. 2017. Blacklist of NoCoin. History for NoCoin/src/blacklist.txt.
[22] Keraf. 2017. NoCoin. https://github.com/keraf/NoCoin.
[23] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. 2016. Enhancing bitcoin security and performance with strong consistency via collective signing. In 25th USENIX Security Symposium (USENIX Security 16). 279–296.
[24] Matt Murray. 2017. Firefox Quantum vs. Chrome: Which Is Faster? https://www.laptopmag.com/articles/firefox-quantum-vs-chrome.
[25] Michael Nadeau. 2018. What is cryptojacking? How to prevent, detect, and recover from it. https://www.csoonline.com/article/3253572/internet/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html.
[26] Notmining. 2017. notmining. http://notmining.org/.
[27] Bad Packets. 2017. Cryptojacking: 2017 Year-End Review. https://badpackets.net/cryptojacking-2017-year-end-review/.
[28] Bad Packets. 2018. How to find cryptojacking malware. https://badpackets.net/how-to-find-cryptojacking-malware/.
[29] Daniel Plohmann and Elmar Gerhards-Padilla. 2012. Case study of the miner botnet. In 4th International Conference on Cyber Conflict (CYCON). IEEE, 1–16.
[30] Fergal Reid and Martin Harrigan. 2013. An analysis of anonymity in the bitcoin system. In IEEE International Conference on Privacy, Security, Risk, and Trust. 197–223.
[31] RFC. 2016. The scrypt Password-Based Key Derivation Function. https://tools.ietf.org/html/rfc7914.
[32] SimilarWeb. 2018. similarWeb. https://www.similarweb.com/.
[33] Whorunscoinhive. 2018. whorunscoinhive. http://whorunscoinhive.com/.
[34] Wikipedia. 2018. Page semi-protected Cryptocurrency. https://en.wikipedia.org/wiki/Cryptocurrency.
[35] xd4rker. 2017. Blacklist of MinerBlock. https://github.com/xd4rker/MinerBlock/commi- ts/master/assets/filters.txt.
[36] xd4rker. 2017. MinerBlock. https://github.com/xd4rker/MinerBlock.
[37] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The dark alleys of madison avenue: Understanding malicious advertisements. In Proceedings of the 2014 Conference on Internet Measurement Conference (IMC). ACM, 373–380.