SoK: Cryptojacking Malware

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

2021 IEEE European Symposium on Security and Privacy (EuroS&P) | 978-1-6654-1491-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/EUROSP51992.2021.00019

Abstract—**Emerging blockchain and cryptocurrency-based technologies are redefining the way we conduct business in cyberspace. Today, a myriad of blockchain and cryptocurrency systems, applications, and technologies are widely available to companies, end-users, and even malicious actors who want to exploit the computational resources of regular users through *cryptojacking* malware. Especially with ready-to-use mining scripts easily provided by service providers (e.g., Coinhive) and untraceable cryptocurrencies (e.g., Monero), cryptojacking malware has become an indispensable tool for attackers. Indeed, the banking industry, major commercial websites, government and military servers (e.g., US Dept. of Defense), online video sharing platforms (e.g., Youtube), gaming platforms (e.g., Nintendo), critical infrastructure resources (e.g., routers), and even recently widely popular remote video conferencing/meeting programs (e.g., Zoom during the Covid-19 pandemic) have all been the victims of powerful cryptojacking malware campaigns. Nonetheless, existing detection methods such as browser extensions that protect users with blacklist methods or antivirus programs with different analysis methods can only provide a partial panacea to this emerging cryptojacking issue as the attackers can easily bypass them by using obfuscation techniques or changing their domains or scripts frequently. Therefore, many studies in the literature proposed cryptojacking malware detection methods using various dynamic/behavioral features. However, the literature lacks a systemic study with a deep understanding of the emerging cryptojacking malware and a comprehensive review of studies in the literature. To fill this gap in the literature, in this SoK paper, we present a systematic overview of cryptojacking malware based on the information obtained from the combination of academic research papers, two large cryptojacking datasets of samples, and 45 major attack instances. Finally, we also present lessons learned and new research directions to help the research community in this emerging area.**

Index Terms—**cryptojacking, cryptomining, malware, bitcoin, blockchain, in-browser, host-based, detection**

1. Introduction

Since the day Bitcoin was released in 2009, blockchain-based cryptocurrencies have seen an increasing interest beyond specific communities such as banking and commercial entities. It has become so trivial and ubiquitous to conduct business with cryptocurrencies for any end-user as most financial institutions have already started to support them as a valid monetary system. Today, there are more than 2000 cryptocurrencies in existence. Especially in 2017, the interest for cryptocurrencies peaked with a total market value close to $1 trillion [1]. According to a recent Kaspersky report [2], 19% of the world's population have bought some cryptocurrency before. However, buying cryptocurrency is not the only way of investing. Investors can also build mining pools to generate new coins to make a profit. Profitability in mining operations also attracted attackers to this swiftly-emerging ecosystem.

Cryptojacking is the act of using the victim's computational power without consent to mine cryptocurrency. This unauthorized mining operation costs extra electricity consumption and decreases the victim host's computational efficiency dramatically. As a result, the attacker transforms that unauthorized computational power into cryptocurrency. In the literature, the malware used for this purpose is known as *cryptojacking*. Especially after the emergence of service providers (e.g., Coinhive [3], CryptoLoot [4]) offering ready-to-use implementations of in-browser mining scripts, attackers can easily reach a large number of users through popular websites.

In-browser cryptojacking examples. In a major attack, cryptojacking malware was merged with Google's advertisement packages on Youtube [5]. The infected ads package compiled by victims' host performed unauthorized mining as long as victims stayed at the related page. Youtube and similar media content providers are ideal for the attackers because of their relative trustworthiness, popularity, and average time spent on those webpages by the users. In another incident, cryptojacking malware was found in a plugin provided by the UK government [6]. At the time, this plugin was in use by several thousands of governmental and non-governmental webpages.

Cryptojacking examples found on critical servers. In addition to cryptojacking malware embedded into webpages, cryptojacking malware has also been found in *well-protected governmental and military servers*. The USA Department of Defense discovered cryptojacking malware in their servers during a bug-bounty challenge [7]. The cryptojacking malware found in the DOD servers was created by the famous service provider Coinhive [8] and mined 35.4 Monero coin during its existence. Similarly, another governmental case came up from the Russian Nuclear Weapon Research Center [9]. Several scientists working at this institution were arrested for uploading cryptocurrency miners into the facility servers. Moreover, attackers do not only use the scripts provided by the service providers but also modified the non-malicious, legitimate, open-source cryptominers. For example, a cybersecurity company detected an irregular data transmission to a well-known European-based botnet from the corporate network of an Italian bank [10]. Further investigation identified that this malware was, in fact, a Bitcoin miner.

Cryptojacking examples utilizing advanced techniques. There have also been many incidents where the attackers used *advanced techniques* to spread cryptojacking

© 2021, Ege Tekiner. Under license to IEEE.
DOI 10.1109/EuroSP51992.2021.00019

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

malware. For example, in an incident, a known botnet, Vollgar, attacked all MySQL servers in the world [11] to take over the admin accounts and inject cryptocurrency miners into those servers. Another recent incident was reported for the Zoom video conferencing program [12] during the peak of the Covid-19 pandemic, in which the attacker(s) merged the main Zoom application and cryptojacking malware and published it via different file-sharing platforms. In other similar incidents, attackers used gaming platforms such as Steam [13] and game consoles such as Nintendo Switch [14] to embed and distribute cryptojacking malware. Last but not least, in a recent study [15], researchers discovered a firmware exploit in Mikrotik routers that were used to embed cryptomining code into every outgoing web connection, where 1.4 million MikroTik routers were exploited.

Challenges of cryptojacking detection. Given the prevalent emerging nature of the cryptojacking malware, it is vital to detect and prevent unauthorized mining operations from abusing any computing platform's computational resources without the users' consent or permission. However, though it is critical, detecting cryptojacking is challenging because it is different from traditional malware in several ways. First, they abuse their victims' computational power instead of harming or controlling them as in the case of traditional malware. Traditional malware detection and prevention systems are optimized for detecting the harmful behaviors of the malware, but cryptojacking malware only uses computing resources and sends back the calculated hash values to the attacker; so the malware detection systems commonly consider cryptojacking malware as a heavy application that needs high-performance usage. Second, they can also be used or embedded in legitimate websites, which makes them harder to notice because those websites are often trustworthy, and users do not expect any nonconsensual mining on their computers. Third, while in traditional malware attacks, the attacker may ultimately target to exfiltrate sensitive information (i.e., Advanced Persistent Threat (APT)), to make the machine unavailable (i.e., Distributed Denial of Service (DDoS)) or to take control of the victim's machine (i.e., Botnet), in cryptojacking malware attacks, the attacker's goal is to stay stealthy on the system as long as possible since the attack's revenue is directly proportional to the time a cryptojacking malware goes undetected. Therefore, attackers use filtering and obfuscation techniques that make their malware harder for detection systems and harder to be noticed by the users.

Our contributions. Due to the seriousness of this emerging threat and the challenges presented above, many cryptojacking studies have been published before. However, these studies are either proposing a detection or prevention mechanism against cryptojacking malware or analyzing the cryptojacking threat landscape. And, the literature lacks a systemic study covering both different cryptojacking malware types, techniques used by the cryptojacking malware, and a review of the cryptojacking studies in the literature. In this paper, to fill this gap in the literature, we present a systematic overview of cryptojacking malware based on the information obtained from the combination of 42 cryptojacking research papers, $\approx 26K$ cryptojacking samples with two unique datasets, and 45 major attacks instances. Given the widespread usage of cryptojacking, it is important to systematize the cryptojacking malware knowledge for the security community to accelerate further practical defense solutions against this ever-evolving threat.

Key takeaways. In addition to the systematization of cryptojacking knowledge and review of the literature, some of the key takeaways from this study are as follows:

- Recently, security reports [16]–[18] spotted some increase in the number of cryptojacking attacks targeting more powerful platforms such as cloud servers [7], [19], Docker engines [20], IoT devices on large-scale Kubernetes clusters [18]. To hijack and gain initial access [21], [22] to spread the cryptojacking malware, the attackers utilize:
  1) hardware vulnerabilities [23],
  2) recent CVEs [22],
  3) poorly configured IoT devices [24],
  4) Docker engines and Kubernetes clusters [20] with poor security,
  5) popular DDoS botnets for the side-profit [24].

  This new trend of cryptojacking malware has not been investigated in detail by researchers.
- We identified several issues in the studies proposing cryptojacking detection mechanisms in literature. First, we found that as the websites containing cryptomining scripts are updated frequently, it is important for the proposed detection studies to report the dataset dates, which is not very common in the studies in the literature. Second, it is important to report if the proposed detection is online or offline, which is missing in most studies. Third, we also note that the studies in the literature do not measure the overhead on the user side of the proposed solutions, which is critical, especially for browser-based solutions.
- We see that although cryptomining could be an alternative funding mechanism for legitimate website owners such as publishers or non-profit organizations, this usage with the in-browser cryptomining has diminished due to the keyword-based automatic detection systems.

Other surveys. In the literature, a number of blockchain or Bitcoin-related surveys have been published. However, these surveys only focus on consensus protocols and mining strategies in blockchain [25]–[28], challenges, security and privacy issues of Bitcoin and blockchain technology [29]–[34], and the implementation of blockchain in different industries [35] such as IoT [36], [37]. The closest work to ours is Jayasinghe et al. [38], where the authors only present a survey of attack instances of cryptojacking targeting cloud infrastructure. Hence, this SoK paper is the most comprehensive work focusing on cryptojacking malware made with the observations and analysis of two large datasets.

Organization. The rest of this systematization paper is organized as follows: In Section 2, we provide the necessary background information on blockchain and cryptocurrency mining. Then, in Section 3, we explain the methodology we used in this paper. After that, in Section 4, we categorize cryptojacking malware types and give their lifecycles. In Section 5, we give broad information about the source of cryptojacking malware, infection methods, victim platform types, target cryptocurrencies, detection and prevention methods, and finally, evasion and obfuscation techniques used by the cryptojacking malware. Section 6 presents an overview of the cryptojacking-related studies and their salient features in the literature. Finally, in Sec-

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

tion 7, we summarize the lessons learned and present some potential research directions in the domain and conclude the paper in Section 8.

2. Background

In this section, we briefly explain the blockchain concept and cryptocurrency mining process in blockchain networks. Note that cryptocurrency mining is a legitimate operation, and it can be used for profit. To see how cryptojacking malware exploits this process, we first explain how this process works.

2.1. Blockchain

Blockchain is a distributed digital ledger technology storing the peer-to-peer (P2P) transactions conducted by the parties in the network in an immutable way. Blockchain structure consists of a chain of blocks. As an example, in Bitcoin [39], each block has two parts: block header and transactions. A block header consists of the following information: 1) Hash of the previous block, 2) Version, 3) Timestamp, 4) Difficulty target, 5) Nonce, and 6) The root of a Merkle tree. By inclusion of the hash of the previous block, every block is mathematically bound to the previous one. This binding makes it impossible to change data from any block in the chain. On the other hand, the second part of each block includes a set of individually confirmed transactions.

2.2. Cryptocurrency Mining

The immutability of a blockchain is provided by a consensus mechanism, which is commonly realized by a ”Proof of Work” (PoW) protocol. The immutability of each block and the immutability of the entire blockchain are preserved thanks to the chain of block structure. In PoW, some nodes in the network solve a hash puzzle to find a unique hash value and broadcast it to all other nodes in the network. The first node broadcasting the valid hash value is rewarded with a block reward and collects transaction fees. A valid hash value is verified according to a difficulty target, i.e., if it satisfies the difficulty target, it is accepted by all other nodes, and the node that found the valid hash value is rewarded. Different PoW implementations usually have different methods for the difficulty target.

The miners try to find a valid hash value by trial-and-error by incrementing the nonce value for every trial. Once a valid hash value is found, the entire block is broadcast to the network, and the block is added to the end of the last block. This process is known as *cryptocurrency mining* (i.e., *cryptomining*), and it is the only way to create new cryptocurrencies. The chance of finding of valid hash value by a miner is directly proportional to the miner's hash power, which is related to the computational power of the underlying hardware. However, more hardware also increases electricity consumption. Therefore, attackers have an incentive to find new ways of increasing computational power without increasing their own electricity consumption.

Following the invention of Bitcoin, many other alternative cryptocurrencies (i.e., altcoins) have emerged and are still emerging. These new cryptocurrencies either claim to address some issues in Bitcoin (i.e., scalability [40], privacy) or offer new applications (i.e., smart contracts [41]).

In the early days of Bitcoin, the mining was performed with the ordinary Central Processing Unit (CPU), and the users could easily utilize their regular CPUs for Bitcoin mining. Over time, Graphical Processing Unit (GPU)-based miners gained significant advantages over CPU miners as GPUs were specifically designed for high computational performance for heavy applications. Later, Field Programmable Gate Array (FPGA) have changed the cryptocurrency mining landscape as they were customizable hardware and provided significantly more profit than the CPU or GPU-based mining. Finally, the use of the Application-Specific Integrated Circuit (ASIC) based mining has recently dominated the mining industry as they are specially manufactured and configured for cryptocurrency mining.

The alternative cryptocurrencies also used different hash functions in their blockchain structure, which led to variances in the mining process. For example, Monero [42] uses the CryptoNight algorithm as the hash function. CryptoNight is specifically designed for CPU and GPU mining. It uses L3 caches to prevent ASIC miners. With the use of RandomX [42] algorithm, Monero blockchain fully eliminated the ASIC miners and increased the advantage of the CPUs significantly. This feature makes Monero the only major cryptocurrency platform that was designed specifically to favor CPU mining to increase its spread. Moreover, Monero is also known as a private cryptocurrency, and it provides untraceability and unlinkability features through mixers and ring signatures. Monero's both ASIC-miner preventing characteristics and privacy features make it desirable for attackers.

3. SoK Methodology

In this section, we explain the sources of information and the methodology used throughout this SoK paper. Particularly, we benefit from the papers, recent cryptojacking samples we collected, and publicly known major attack instances.

3.1. Papers

Cryptomining and cryptojacking have recently become popular topics among researchers after the price surge of cryptocurrencies and the release of Coinhive cryptomining script in 2017. For our work, we scanned the top computer security conferences (e.g., USENIX) and journals (e.g., IEEE TIFS) given in [43] as well as the digital libraries (e.g., IEEEXplore, ACM DL) with the keywords such as cryptojacking, bitcoin, blockchain, etc. and a variety of combinations of these keywords. In total, we found 43 cryptojacking-related papers in the literature. While one of the papers [38] is a survey paper, the rest are focusing on two separate topics: 1) Cryptojacking detection papers, 2) Cryptojacking analysis papers. We found that there are 15 cryptojacking analysis papers, while there are 27 cryptojacking detection papers in the literature. We further present a literature review of these 42 studies in Section 6. Figure 5 in Appendix A shows the distribution of research cryptojacking-related research papers per year. As seen in the figure, there is an increasing effort in the academia in the last three years with many research papers. Therefore, there is a need for a systematization of this knowledge for cultivating better solutions.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

3.2. Samples

Each research paper in the literature focuses only on one aspect of the cryptojacking malware. For a comprehensive understanding of the cryptojacking malware, we also benefited from the real cryptojacking malware samples. For this purpose, we collected two datasets: 1) VirusTotal (VT) Dataset and 2) PublicWWW Dataset. The VT dataset consists of 20200 cryptocurrency ”miner” samples uploaded to the VT [44] and their VT scan reports. On the other hand, we created the PublicWWW dataset using the website source search engine PublicWWW [45]. We found 6269 unique domain containing cryptomining script in their source code. We used these two datasets for the following purposes and in the noted sections:

- To understand the lifecycle of in-browser and host-based cryptojacking (Section 4.1 & 4.2)
- To verify the service provider list given in other studies and as a source of cryptojacking malware (Section 5.1.1)
- To verify the use of mobile filtering methods used in the in-browser cryptojacking malware. (Section 5.3.6)
- To verify that Monero is the main target currency used by cryptojackings (Section 5.4.1)
- To find the other cryptocurrencies used by cryptojacking malware (Section 5.4)
- To verify that the existence of CPU limiting technique for the obfuscation (Section 5.6.1)
- To verify and understand the use of code encoding for the obfuscation (Section 5.6.3)

We note that even though these two datasets were useful for these purposes, they also have some limitations that may affect the findings in this paper. For the VT dataset, we are given $\approx 437K$ unique samples (cryptojacking and non-cryptojacking) by the VT [44] using our Academic API privileges and their scan reports. Therefore, the VT dataset is not an exhaustive list of cryptojacking samples on the VT because API accesses with more extended privileges exist. For example, we observed that 85% of all cryptojacking samples in the VT dataset are from 2018, showing that 2018 samples are over-represented in the VT dataset. Figure 6 in Appendix B.1 shows the distribution of all samples by the submission date. Therefore, any conclusion in terms of the representation of the samples in real-life may have a bias. However, we also note that such a large-scale measurement study is outside the scope of this work. For example, the study in [46] presents such a study with 1.2M malicious cryptocurrency miners collected over a period of twelve years. In our paper, we focused on understanding the behavioral characteristics of cryptojacking malware and reviewing the studies in the literature. e give a more detailed explanation of the datasets, their limitations, and perform distribution analysis of these datasets in Appendix. Finally, we also published our datasets[1]to further accelerate the research in this field.

3.3. Major Attack Instances

Our third source of information is the major attack instances that appeared on the security reports released by the security companies such as Kaspersky, Trend Micro, Palo Alto Network, IBM, and others, as well as the major security instances that appeared on the news. The major attack instances that appeared on the news may be used to identify unique and interesting cases, while the security reports may shed light on the trends due to the real-time and large-scale reach of the security companies. Particularly, we used these instances in Section 1 for motivational purposes; in Section 5 to find out different techniques used by cryptojacking malware; in Section 7 in order to find out potential new trends in the cryptojacking malware attacks. Since the collection of these resources may be valuable to other researchers and due to the space limitation here, we also release them in a detailed and organized way together with the blacklists and service providers’ documentation links in our dataset link[1]. Table 10 in Appendix A shows the yearly distribution of the attack instances we used in this paper.

4. Cryptojacking Malware Types

Cryptojacking malware, also known as cryptocurrency mining malware, compromises the computational resources of the victim’s device (i.e., computers, mobile devices) without the authorization of its user to mine cryptocurrencies and receive rewards. A cryptojacking malware’s lifecycle consists of three main phases: *1) script preparation, 2) script injection,* and *3) the attack*. The script preparation and attack phases are the same for all cryptojacking malware types. In contrast, the script injection phase is conducted either by injecting the malicious script into the websites or locally embedding the malware into other applications. Based on this, we classify the cryptojacking malware into two categories: *1) In-browser cryptojacking* and *2) Host-based cryptojacking*. In the following sub-sections, we explain the lifecycle of both in-browser and host-based cryptojacking malware.

4.1. Type-I: In-browser Cryptojacking

The development of web technologies such as JavaScript (JS) and WebAssembly (Wasm) enabled interactive web content, which can access the several computational resources (e.g., CPU) of the victim’s device (e.g., computer or mobile device). In-browser cryptojacking malware uses these web technologies to create unauthorized access to the victim’s system for cryptocurrency mining via web page interactions on the victim’s CPU.

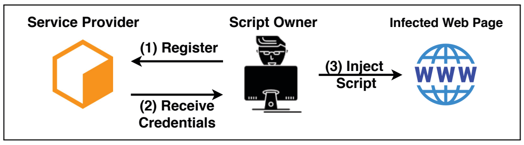

Figure 1. Script preparation and injection phases of a in-browser cryptojacking malware.

Figure 1 shows the script preparation and injection phases of in-browser cryptojacking malware. The script owner[2] first registers (Step 1) and receives its service credentials and ready-to-use mining scripts from the service provider (Step 2). The service provider separates the mining tasks among its users and collects all the revenue from the mining pool later to be shared among

[1] https://github.com/sokcryptojacking/SoK

[2] We call it script owner rather than an attacker because the script can also be used for legitimate purposes.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

its users. After receiving the service credentials, the script owner injects the malicious cryptojacking script into the website's HTML source code (Step 3). We explain this and other cryptojacking infection methods in Section 5.2 in detail.

In the attack phase, as shown in Figure 2, victims first reach the website source code from their devices (Step 1,2). The web browser loads the website and automatically calls the cryptojacking mining script (Step 3). Once the script is executed, it requests a mining task from the service provider (Step 4). The service provider transfers the task request to the mining pool (Step 5). Then, the mining pool assigns the mining task (Step 6). The service provider returns the task to the mining script (Step 7). The mining script returns this new mining assignment to the victim's computer (Step 8), and the victim's device starts the mining process (Step 9). As long as the mining script and service provider remain online, the script continues the mining process on the victim's computer (Step 9) and then returns the mining results to the service provider (Step 10) directly. The service provider collects all the data from different sources and sends the results to the mining pool (Step 11). Finally, the mining pool sends the reward back to the service provider in the form of a mined currency (Step 12). The script owner receives its share from the service providers using its service credentials after the service provider cuts its service fee. In this ecosystem, the attackers use the CPU power of their victims, and the victims do not receive any payment nor benefit from any other entity.

4.2. Type-II: Host-based Cryptojacking

Host-based cryptojacking is a silent malware that attackers employ to access the victim host's resources and to make it a zombie computer for the malware owner. Compared to in-browser cryptojacking malware, host-based malware does not access the victim's computation power through a web script; instead, they need to be installed on the host system. Therefore, they are generally delivered to the host system through methods such as embedded into third-party applications [12], [47], using vulnerabilities [22], or social engineering techniques [48], or as a payload in the drive-by-download technique [49]. We explain these methods in more detail in Section 5.2.

Figure 3 shows the lifecycle of a host-based cryptojacking malware. The script preparation phase starts with the creation of unauthorized cryptocurrency mining malware (1). Then, the attacker merges this malware with a legitimate application to trick the victim (2). After the malware preparation, the malware injection process starts with uploading this malicious application to online data-sharing platforms (e.g., torrent, public clouds) (3). When the victim downloads any of the infected applications and installs them on their host machines (e.g., Personal Computer, IoT device, Server)(4), the malware injection phase of the lifecycle is completed.

In the attack phase, the host-based cryptojacking malware is connected to the mining pool via web socket or API and receives the hash puzzle tasks to calculate hash values (5). The calculated hash values are sent back to the mining pool (6). Finally, the attacker receives all of the revenue without any energy consumption (7) and not sharing anything with the victim.

After receiving all its revenue in the form of cryptocurrency from the service provider, the attacker has three options to use its revenue: 1) Converting to fiat currency via exchanges or p2p transactions, 2) Using it as a cryptocurrency for a service [50], or 3) Using cryptocurrency mixing services [51], [52] to cover its traces. Further end-to-end analysis of the cryptojacking economy/payments is out of this study's scope, and similar studies can be found in the ransomware domain [53]–[56].

5. Cryptojacking Malware Techniques

In this section, we explain the techniques used by cryptojacking malware. Particularly, we articulate on the following:

- Source of cryptojacking malware
- Infection methods
- Victim platform types
- Target cryptocurrencies
- Evasion and obfuscation techniques

5.1. Source of Cryptojacking Malware

This sub-section explains whom the scripts are created by and how they are distributed to attackers.

5.1.1. Service Providers. The service providers are the leading creators and distributors of cryptojacking scripts. The service providers give every user a unique ID to distinguish them in terms of the hash power. The service provider generates the script for the user regardless of the user is malicious or not. All the user needs to do is copy and paste the script to create a malicious sample for the attack.

Coinhive [8] was the first service provider to offer a ready-to-use in-browser mining script in 2017 to create an alternative income for web site and content owners. Even though the initial idea of Coinhive was to provide an alternative revenue to webpage owners, it rapidly became popular among attackers. During the operation of Coinhive, they were holding a significant share of the total hash rate of Monero. After the sharp decrease in Monero's price [57], Coinhive was shut down by their owners in March 2019 due to the business' being no longer profitable.

Some of the alternative service providers which had continued/continuing their operations are Authedmine [3], Browsermine [58], Coinhave [59], Coinimp [60], Coin nebula [61], Cryptoloot [4], DeepMiner [62], JSECoin [63], Monerise [64], Nerohut [65], Webmine [66], WebminerPool [67], and Webminepool [68]. Some of these service providers also came up with several new functionalities, such as offering a user notification method or a GUI for the user to adjust the cryptomining parameters. Note that, we also verified these service providers using the samples in the PublicWWW dataset. In order to find the corresponding service providers of each sample, we performed a keyword search on the HTML source code of all samples. We found that 5328 samples use one of these 14 aforementioned service providers, while 941 samples with unknown service providers. Moreover, we also found out that 144 samples are using scripts from multiple service providers in their source codes. More details on the PublicWWW dataset can be found in Appendix.

5.1.2. Cryptominer Software. Blockchain networks rely on several network protocols and cryptographic authentication methods. Miners must be part of these protocols

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

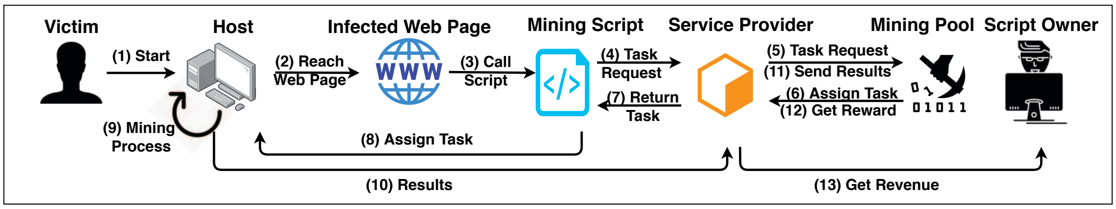

Figure 2. The lifecycle of a in-browser cryptojacking malware.

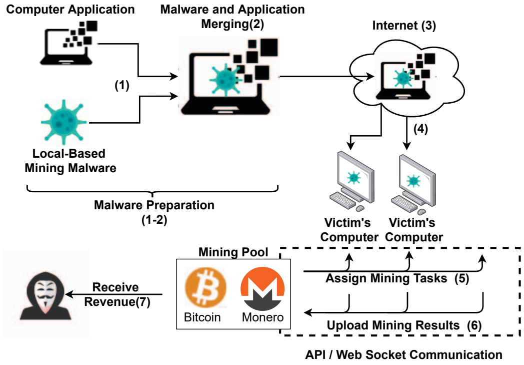
Computer Application
Malware and Application Merging(2)
Internet (3)
(1)
(4)
Local-Based Mining Malware
Malware Preparation (1-2)
Victim's Computer
Victim's Computer
Mining Pool
Receive Revenue(7)
Bitcoin
Monero
Assign Mining Tasks (5)
Upload Mining Results (6)
API / Web Socket Communication

Figure 3. The lifecycle of a host-based cryptojacking malware.

and follow the rules provided and developed by the communities. PoW-based cryptocurrencies also have specific rules for their blockchain networks. Due to blockchain technology's public and open nature, the source code of these miners are published by the communities via code sharing and communal development platforms. Attackers can easily obtain and modify these miners and adopt them to perform mining inside their victims' host machines. Moreover, there are also several plug-and-play style mining applications provided by several mining pools. Attackers are also modifying these applications for cryptojacking. For example, XMRig [69] is a legitimate high-performance Monero miner implementation, and it is open-source. Its signature is found in several highly impactful attacks affecting millions of end devices around the world [70], [71], which are also reported by Palo Alto Networks and IBM. Moreover, we also found 139 unique samples that are labeled with the signature of ”xmrig” in our VT dataset.

5.2. Infection Methods

In this section, we explain the infection methods used by cryptojacking malware in detail.

5.2.1. Website Owners. Website owners, who have admin access to the website's servers, may employ in-browser mining scripts to gain extra revenue or provide in exchange of an alternative option to premium content they provide. Only with this method, webpage owners may receive the revenue of the script in their webpage. While some website owners inform their visitors about the cryptomining script they employ, some others do not inform their visitors, and this behavior can be considered as crime [72] in several countries.

5.2.2. Compromised Websites. Attackers may inject their cryptojacking malware into random web pages that have several vulnerabilities. Indeed the name cryptojacking itself is the combination of ”cryptomining” and ”hijacking.” Ruth et al. [73] state that ten different users created 85% of all Coinhive scripts they found. The owners of these webpages do not have any information about these scripts; additionally, they do not profit from them.

Several works claim that the attackers generally use the same ID for all the infected web pages, making them more traceable. For example, the authors in [74] reveals the cryptojacking campaigns through this method and discover that most of these campaigns utilize the vulnerabilities such as remote code execution vulnerabilities. When we investigated the common instances related to this domain, one security company found cryptojacking malware inside of the Indian government webpages [75], which affect all *ap.gov.in* domains and sub-domains.

5.2.3. Malicious Ads. Some attackers embed their cryptojacking malware into JavaScript-based ads and distribute them via mining scripts. With this method, the attackers can reach random users without any extra effort. To make this attack, they do not need to infect any webpage or application. YouTube [5] and Google ad [76] services were also infected and the users of these websites and their services became the victims of the cryptojacking attacks. The attackers successfully mined Monero with their visitors. The attackers successfully mined Monero with their visitors. The advantage of this method is that it allows attackers to reach a large number of visitors when it is embedded into popular websites without getting access to the website's servers.

5.2.4. Malicious Browser Extensions. Browser extensions can also reach the computer's CPU sources and act like cryptojacking malware located into a webpage. These extensions have a major distinctive difference; they can stay online and perform mining as long as the infected browser remains open independent from the websites accessed by the victim. However, major browser operators like Google announced that they would ban all the cryptomining extensions on their platform regardless of their intention as it is mostly being abused in practice [77].

5.2.5. Third-party Software. Merging malware with any market application and publishing it via several sharing platforms is a well-known method among the attackers to spread the malware. Attackers modify the cryptominer software to run cryptojacking in the background and merge it with legitimate applications. The attackers tend to use computation-intensive applications (e.g., animation applications, games with high hardware needs, engineering programs) because the use of those applications means that the victims' system has computationally powerful

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

hardware and the application that host-based cryptojacking malware embedded, have access permission to the needed hardware components of the victim's host system.

Several major instances have already happened, such as, one attacker merged Zoom [12] video calling application with a regular bitcoin miner and distributed it via several sharing platforms. In another attack, the attackers used a popular video game Fortnite to spread the virus [78] to mine Bitcoin. Unlike the in-browser mining, which became popular in 2017, we found the attack instances using this method even in 2013, where the Bitcoin mining script found as part of the game's code itself [79].

5.2.6. Exploited Vulnerabilities. In several cases, attackers exploit several zero-day vulnerabilities that they found in hardware and software. Attackers inject their mining malware into several devices and make them mine cryptocurrency. There are several important instances happened in the last several years. The most remarkable example directly affects 1.4 Million Mikrotik [15] routers globally, and a vulnerability in the hardware operating system causes this instance. The researchers claim that a major percentage of Remote Code Execution (RCE) attacks [80] aims to locate mining scripts inside the host systems.

5.2.7. Social Engineering Techniques. Social engineering is a commonly used technique among malware attackers to bypass security practices. Similarly, attackers also use social engineering attacks to manipulate human psychology and navigate the victims' access or install malicious software on their computers. The researchers have observed that attackers are still using old techniques such as social engineering to install cryptojacking malware on their victims' computers [21].

5.2.8. Drive-by Download. A drive-by download is another technique used by malware attackers to deliver and install malicious files to victims' devices without their knowledge. Victims may face this attack while visiting a web page, opening a pop-up window, or checking an email attachment. In one case [49], the attackers used this method to inject their cryptojacking malware into their victims' devices. They exploited shell execution vulnerability to download their cryptojacking malware to victims' computers directly.

5.3. Victim Platform Types

5.3.1. Browser. Browsers are the most commonly used victim platforms as the attackers do not need to deliver any malicious payload to the victim to use the computational resources of the victim. In other words, when the victim reaches the infected webpage, the malware automatically starts mining and do not leave any data behind. The second significant advantage of the browser environment is, thanks to service providers, ready-to-use mining scripts can be applied to any webpage very easily and quickly. The studies in the literature that we also present in Section 6 mostly focus on in-browser cryptojacking. However, the attackers can access only the CPUs of the victims through the browsers, which makes them infeasible for the currencies allowing ASIC miners such as Bitcoin. Therefore, cryptojacking malware samples utilizing browsers mostly mine Monero or other cryptocurrencies, which allow cryptomining by personal computers on non-ASIC CPU architectures.

5.3.2. Personal Computers. Personal computers are generally designed to allow end-users to perform their daily tasks. Personal computers are recently modified to overcome high-level computations to allow their users to use heavy-computation applications (e.g., video-games, video rendering applications). Attackers targeting personal computers aim to reach many victims because a limited number of victims would not be profitable. In-browser cryptojacking embedded into popular websites is ideal for this type of cryptojacking attack. In addition, they can also instantiate such an attack through large-scale campaigns. For example, in [81], Cisco researchers document their findings of a two-year campaign delivering XMRig in their payload. They also observed that the malware "makes a minimal effort to hide their actions" and posting the malware "on online forms and social media" to increase the victim pool.

5.3.3. On-premise Server. On-premise (i.e., in-house) servers are the servers where the data is stored and protected on-site. It is preferred by highly critical organizations such as governmental organizations as it offers greater security and full control over the hardware and data. However, on-premise servers are also another victim platform type attacked by the host-based cryptojacking malware samples. Compared to personal computers, on-premise servers are more computationally powerful and host numerous services accessed by many connections. This allows attackers to the broader attack surface. Still, the attackers have to find a way to deliver and install the cryptomining script on the on-premise server to access this platform. In several instances, the attackers used system vulnerabilities [10], third party infected software [6], and several social engineering methods [21] to install cryptojacking malware to the victims' on-premise server.

5.3.4. Cloud Server. Cryptojacking malware also exploits cloud resources to mine cryptocurrencies. Cloud-based cryptojacking attack is a fast-spreading problem in the last two years, where it became popular, especially after the shutdown of the Coinhive when the attackers were looking for new platforms to infect. Attackers target several vulnerabilities to hijack victims' cloud servers and locate cryptocurrency miners into their systems. Clouds servers, especially Infrastructure-as-a-service platforms such as Amazon Web Services (AWS), are being targeted by the attackers because of their:

- Virtually infinite resources,
- Large attack surface due to server structure,
- Malware spreading capabilities,
- Reliable Internet connection,
- Longer mining/profit period due to host-based capabilities

Several instances of this type of cryptojacking malware have been found on cloud servers [19], [20], [22], [82]–[84]. In these attacks, attackers used different techniques to hijack the cloud server to inject cryptojacking malware. For example, in their 2020 annual report, Check Point Research [82] observed that attackers integrate the cryptominer to the popular DDoS botnets such as KingMiner targeting Linux and Windows servers for side-profits. In another attacker instance [20], the researchers

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

found an open directory containing malicious files. Further analysis revealed that the file contains a DDoS bot targeting open Docker daemon ports of Docket servers and ultimately installing and running the cryptojacking malware after the execution of its infection chain. In a similar attack instance [22], the researchers noted a cryptojacking malware delivered using a CVE exploitation targeting WebLogic servers. Tesla-owned Amazon [19] and the clients of Azure Kubernetes clusters [84] were exposed to cryptojacking attacks due to poorly configured cloud servers. Indeed, Jayasinghe et al. [38] showed that the count of cryptojacking malware targeting cloud-based infrastructure is increasing every year and affects more prominent domains such as enterprises.

5.3.5. IoT Botnet. IoT devices generally have small processing powers to perform basic tasks. It is being expected that there will be more than 21.5 billion IoT devices connected to the internet [85] by 2025. Attackers aim to create botnets with the collaboration of thousands of these IoT devices and perform several attacks such as DDoS due to their small processor, limited hardware, low-level security, and weak credentials, which was also exploited in the example of Mirai botnet's DDoS attack [86]. Later, IBM researchers also found that the modified version of the Mirai Botnet also started to mine Bitcoin [24]. Bartino et al. [87] states that there are several worms in IoT devices that hijacked them for mining purposes, and Ahmad et al. [88] proposes a lightweight IoT cryptojacking detection system to detect any cryptojacking attack that focuses on IoT devices.

5.3.6. Mobile. Cryptojacking malware samples targeting mobile devices inject cryptojacking script into their application and list the application in the application markets. Like every other type of cryptojacking attack, the mobile-based cryptojacking samples also have seen a great increase in the number of attacks. Because of this, both Google [89] and Apple [90] removed the cryptomining applications from their platforms. However, they still exist in alternative markets [91]. The study by Dashevskyi et al. [91] focuses on Android-based cryptojacking malware.

Moreover, mobile devices are generally not considered powerful enough for cryptocurrency mining because they generally use more restricted hardware and optimized operating systems (e.g., iOS and Android). Besides, the cryptocurrency mining process consumes extra battery and processing power, which may cause hardware problems such as overheating and apps to freeze or crash on mobile devices. Due to these reasons, cryptojacking attacks on mobile devices are not preferred by attackers, and they generally apply a mobile filtering method to opt-out mobile devices.

```
1 <script>
2     var miner=new CoinHive.Anonymous('Key', {
3        Threads:4, autoThreads:false, throttle:0.8);
4     if (!miner.isMobile()) &&!miner.didOptOut(14400)
5        { miner.start(); } }
6 </script>
```

Listing 1: The mobile device filtering method used in a cryptojacking sample.

Listing 1 is a recent cryptojacking sample with the mobile device filtering method found in a sample in our dataset. In line 4, the script automatically calls a mobile device detection function and starts the cryptocurrency mining process only if it is not a mobile device.

5.4. Target Cryptocurrencies

In this section, we give brief information about the most preferred cryptocurrencies by the attackers.

5.4.1. Monero. Monero has several advantages over other cryptocurrencies, making it favorable to attackers. First of all, Monero successfully implements and modifies the RandomX mining algorithm and CryptoNight hashing algorithm to prevent ASIC miners and give a competitive advantage to the CPU miners over GPUs via L3 caches [92]. The Monero community aims to keep their network decentralized and allows even small miners to mine Monero. As in-browser cryptojacking malware can only access the CPUs of the personal computers through the browsers, Monero is ideal as a target cryptocurrency instead of other cryptocurrencies that are mined dominantly by other computationally more powerful ASIC and GPU miners. Second, Monero provides anonymity features through cryptographic ring signatures [93], [94], which makes the attackers untraceable. Thanks to these features of the Monero, attackers tend to mine Monero with their in-browser cryptojacking malware.

When we analyze the samples' cryptomining scripts in the PublicWWW dataset and their service providers' documentation, we found that all eleven service providers except Browsermine, CoinNebula, JSEcoin either use Monero or have the option to choose Monero in their scripts as a target cryptocurrency. This shows to 91% of the samples in the PublicWWW dataset use Monero to mine.

5.4.2. Bitcoin. In recent years, Bitcoin mining has seen enormous attention, which led to a dramatic increase in the difficulty target. ASIC and FPGA miners are the main reason behind this dramatic increase because the mining structure of the Bitcoin allows to build and use of specified mining hardware which is much more powerful and profitable than the CPUs and GPUs. The increase in difficulty target and disadvantages of CPU made the CPU mining infeasible and not profitable. Therefore, attackers who perform in-browser cryptojacking attack donot prefer Bitcoin mining. We also see that none of the service providers of the in-browser cryptojacking samples in our PublicWWW dataset supports Bitcoin mining. However, host-based cryptojacking malware can reach all the components of the victims' computer system and make Bitcoin mining on GPU and other high-performance computational resources of the computers. We also observe this in our VT dataset. We performed a keyword search for ”bitcoin” on the AV labels of 20200 samples of both in-browser and host-based cryptojacking malware. We found that 7111 of 20200 samples are marked with a label containing the keyword ”bitcoin”. Even though this does not show that those samples are absolutely using bitcoin as a target cryptocurrency, but it is a potential indicator for the host-based cryptojacking samples mining Bitcoin based on the assumption of AV vendors are labeling the correct currency for the AV labels.

5.4.3. Other Cryptocurrencies. Cryptojacking is attractive for attackers as cryptomining can be parallelized

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

among many victims. Therefore, it is possible for cryptocurrencies to allow distributed cryptomining. Both Monero and Bitcoin use PoW as a consensus method. However, instead of PoW, other cryptocurrencies utilize different consensus models such as Proof of Stake [95], and Proof of Masternode [96]. Most of these new consensus models do not depend on distributed power-based mining algorithms; therefore, cryptojacking is not an option for those currencies. For the cryptocurrencies that can be mined distributively [97], the mining pools provide collective mining services to their participants. Other cryptocurrencies that are preferred by attackers are Bitcoin Cash [98], Litecoin [99], and Ethereum [100].

There are also several cryptocurrencies developed specifically for in-browser cryptomining activities. JSEcoin [63] is an example of them and offers also transparency. Other cryptocurrencies created for this purpose are BrowsermineCoin [58], Uplexa [101], Sumocoin [102], and Electroneum [103].

5.5. Detection and Prevention Methods

In the traditional malware detection literature, there are two main analysis methods: 1) static [104] and 2) dynamic [105]. Both analysis methods have several pros and cons in terms of accuracy and usability.

- *Static Analysis:* Static analysis is a widely used method to examine the application without executing it. Static analysis tools generally seek specific keywords, malware signatures, and hash values. In the cryptojacking domain, mining-blocking browser extensions [106], [107] workin this way, i.e., any domain given in the pre-determined blacklist is blocked. However, due to the fix, pre-configured nature of the static detection methods, these implementations are usually easy to circumvent.
- *Dynamic Analysis:* In dynamic analysis, the malware sample is executed in a controlled environment, and its behavioral features are recorded for further analysis and detection. Malware analyzers generally use automated [108] or non-automated sandboxes [105] to run the code and observe the malware's behavior. In the literature, 24 machine learning-based proposed detection mechanisms use dynamic analysis to detect cryptojacking malware. These studies use various datasets, features, classification algorithms and some of them works for both in-browser and host-based cryptojacking malware. We explain these studies in Section 6.1.

As the execution of in-browser cryptojacking malware depends on running the JavaScript code, another way to stop it is to disable the use of JavaScript, but this would also decrease the usability of the browser significantly. Finally, there are antivirus programs with the cryptojacking detection capability [109], [110]. However, their detection algorithms are proprietary.

5.6. Evasion and Obfuscation Techniques

The purpose of the cryptojacking malware is to exploit the resources of the victim as long as possible; therefore, staying on the system without being detected is of paramount importance. For this purpose, they utilize several obfuscation methods.

5.6.1. CPU Limiting. High CPU utilization is still the most important common point of all kinds of cryptojacking malware because CPU usage is the main requirement of the cryptocurrency mining process. Therefore, CPU limiting is a highly preferred method by the attackers to obfuscate the mining script. With this method, the script owners can bypass the high CPU usage-based detection systems and avoid being put on the blacklist. Moreover, the CPU limiting is also used by legitimate website owners performing cryptocurrency mining as an alternative revenue because it provides a better user experience. Line 3 in Listing 1 shows an example of a CPU limiting method used by a cryptojacking malware, where the attacker sets throttle to 0.8, e.g., the attacker wants to use only 20% of the CPU load for cryptocurrency mining. In our PublicWWW dataset, we searched for the keyword ”throttle: 0.9” and we found that 1384 samples out of all 6269 cryptomining scripts set the throttle to 90%, which shows that CPU is limiting is a very common practice among the in-browser cryptomining scripts.

5.6.2. Hidden Library Calls. Library calling [111] is a well-known technique used by programmers to make the code more efficient, systematic, and readable. However, it can also be used by the attackers to obfuscate their scripts. Particularly, in order to hide the mining code from the detection methods, the attackers create new scripts that do not have specific keywords. The malicious part of the script is moved to an external library, which is called during the script's execution, and only the code snippet to call this library is included in the main code.

5.6.3. Code Encoding. Encoding the malware source code with several encoding algorithms provides invisibility against keyword-based static analysis detection methods such as blacklists. This method transforms the text data into another form, such as Base64, and after this process, the data can only be read by the computers. Some examples of this we found in our PublicWWW dataset are the cryptomining scripts provided by the service providers Authedmine [8] and Cryptoloot [4].

5.6.4. Binary Obfuscation. Similar to code encoding technique, binary obfuscation is a practice among malware authors to hide malicious code from standard string matching algorithms and make it harder to recover by the sandboxes and other dynamic malware detection methods. However, they differ in the cryptojacking type that is used to hiding, i.e., binary obfuscation is used by the host-based cryptojacking malware while code encoding is used by the in-browser cryptojacking malware. For binary obfuscation, attackers generally use well-known packers such as UPX. The authors of [46] observe that 30% of 1.2M binary cryptojacking malware samples are obfuscated, which shows that it is a common practice among the cryptojacking malware attackers, too.

6. Literature Review

The surge of cryptojacking malware, especially after 2017, also drew the attention of academia and resulted in many publications. We found these studies focus on three topics: 1) Cryptojacking detection studies, 2) Cryptojacking prevention studies, and 3) Cryptojacking analysis studies. Among 42 academic research papers, we found

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

that 15 of them focus on the experimental analysis of the cryptojacking dataset. At the same time, 3 of them proposes a method for the detection and prevention of cryptojacking malware together, and 24 of them proposes only a method for the detection of the cryptojacking malware. In the next sub-sections, we give a review of these studies.

6.1. Cryptojacking Detection Studies

In this section, we survey the cryptojacking malware detection studies. Table 1 shows the list of the proposed cryptojacking detection mechanisms in the literature. The following sub-section gives a detailed overview of the dataset, platform, analysis method, features, and classifiers used in these detection mechanisms.

6.1.1. Dataset. A dataset is generally used to evaluate the effectiveness of the proposed detection method. Several datasets are commonly used in the cryptojacking malware detection literature. The most common one is Alexa top webpages [113], [116], [117], [119], [120], [122]. Alexa sorts the most visited websites on the Internet; however, it does not provide the source code for these websites. Therefore, these studies also used Chrome Debugging Protocol to instrument the browser and collect the necessary information from the websites, except the study [122], which works with a limited number (500) of websites. Moreover, the study in [116] also used known and frequently updated blacklists [106], [107], [136] to build a ground truth for their training dataset, and then they performed an analysis using Alexa top 1M websites. In addition to the Alexa top websites, the study in [97] used a cryptojacking dataset obtained from VirusTotal. They collected 1500 active Windows Portable Executable (PE32) cryptocurrency mining malware samples registered in 2018 and used the Cuckoo Sandbox [137] to obtain detailed behavioral reports on those samples. Furthermore, the studies in [114], [115], [118], [132] performed their analysis by installing the legitimate mining scripts, and the studies in [117], [121] manually injected miners to the websites to test their detection mechanisms.

6.1.2. Platform. Most of the cryptojacking detection mechanisms in the literature [112], [113], [115]–[119], [121], [122], [132], [138] are proposed for the detection of in-browser cryptojacking malware. There are only a few studies [97], [128] proposed for host-based cryptojacking malware. In addition, Conti et al. [132], propose a hardware-level detection mechanism, which can be used to detect both host-based and in-browser cryptojacking malware.

6.1.3. Analysis Features. As can be seen from Table 1, in the cryptojacking domain, the majority of the proposed detection methods are using dynamic analysis. The main reason for this is that mining scripts use a set of known instructions, and they follow and repeat predefined mining steps. For example, miners use cryptographic hash libraries and increment the value of a static variable (i.e., nonce) repeatedly or connect to some known service providers to continue to upload some output results and receive new tasks. These typical behaviors of the cryptojacking malware create a pattern and make them detectable by dynamic analysis. In the literature, only a few studies use static features such as opcodes [97] and WebAssembly (Wasm) instructions [112]. WebAssembly [139] is a low-level instruction format that allows programs to run closer to the machine-level language and provide higher performance via stack-based virtual machines [140]. This low-level instruction model lets the WebAssembly run the codes more efficiently, and this feature provides more profit because the cryptojacking script eliminates most of the delay caused by the code execution process. All major browsers in the market currently support WebAssembly.

Opcodes are machine language instructions that specify the operations to be performed and are used by system calls. The proposed detection system in [97] uses opcodes for static analysis, where opcodes are extracted using IDA Pro. In the cryptojacking example, opcodes focus on requests between mining scripts and the operating system's kernel. With this method, they achieve 95% accuracy with the Random Forest classifier.

On the other hand, many detection mechanisms have been proposed [113], [115]–[119], [121], [132], [138] using dynamics features. The most commonly used dynamic features in these studies are as follows:

- *CPU Events [113], [115]–[117], [123], [124], [130], [133]–[135], [138]:* CPU events are the most commonly used features among the dynamic analysis-based detection mechanisms because in-browser cryptojacking scripts have to fetch the CPU instructions to perform the mining, independent of the used hardware. If an in-browser operation uses cryptographic libraries too frequently, which is abnormal for regular websites, it can be directly detected by CPU instructions. Even though CPU is the most crucial feature of cryptocurrency mining, using only CPU events as features may cause high false-positive rates (FPR) because flash gaming or online rendering websites also use the CPU of the system heavily for their operations. To keep FPR as low as possible, most detection methods use more than one features simultaneously [112], [113], [115], [117], [123], [124], [134], [135].
- *Memory activities [113], [115], [117], [132]:* Memory activity is another commonly used feature among the dynamic detection methods listed in Table 1.
- *Network package [113]–[115], [117], [118], [128]:* Network packages are also a handy and useful method to detect cryptojacking activity because of the massive network traffic generated while uploading the calculated hash values to the service provider. The studies [113]–[115], [117] utilized network traffic rate as an additional feature along with other features such as memory and CPU-related features. On the other hand, the studies in [118], [128] used only network packages for cryptojacking malware detection. Particularly, Neto et al. [118] use the network flow as a feature, while Caprolu et al. [128] use interarrival times and packet sizes as features in their detection algorithm.
- *JavaScript (JS) compilation and execution time [116], [124]:* In [116], [124], it has been shown that JS engine execution time and JS compilation time is significantly affected by cryptojacking malware. However, online games and other online rendering platforms can also cause the same behavior causing false positives in the detection mechanism. Therefore, the study in [116] also uses CPU usage, garbage collector, and iframe resource loads as secondary features to obtain more accurate results and decrease false positives. The garbage col-

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

TABLE 1. Cryptojacking malware detection mechanisms in the literature.

Ref	Dataset	Type	Method	Features	Classifier	Performance
Rüth et al. [73]	Three largest TLDs Alexa: 1M	In-browser	Static	Wasm signatures	SRSE	N/A
Minesweeper [112]	Alexa 1 Million	In-browser	Static	Wasm code CPU cache events	Matching	N/A
RAPID [113]	Alexa: 330.500	In-browser	Dynamic	Resource consumption (memory, network, and processor) and JavaScript API Events	SVM	Benign (best): Precision: 99.99% Recall: 99.99% F1: up to 99.99% Mining (best): Precision: 96.54% Recall: 95.48% F1: up to 96.0%
Muñoz et al. [114]	Network traffic of six cryptocurrencies using Stratum protocol	In-browser	Dynamic	Metadata of inbound and outbound network traffic	DT	Best: Accuracy: 99.9% Precision: 98.2% Recall: 90.7%
CapJack [115]	Five user applications and a Coinhive miner	In-browser	Dynamic	CPU utilization, Memmory, Disk read/write rate, Network interface)	CNN	87% Instant 99% After 11 seconds 98% Mobile Single 86% Mobile Cross 97% AWS single 89% AWS Cross
OUTGUARD [116]	Alexa 1M and 600K	In-browser	Dynamic	Js Execution Time, JS compilation Time, Garbage Collector, Iframe resource loads, CPU usage	SVM, RF	SVM (best): TPR: 97.9% FPR: 1.1%
CoinSpy [117]	100k websites from Alexa 1M and 50 manipulated cryptojacking websites	In-browser	Dynamic	CPU, Memory, Network behaviors	CNN	Accuracy:97%
MineCap [118]	The network traffic captured from two mining and streaming applications	In-browser	Dynamic	Network packages	IL	Accuracy: 98%, Precision: 99%, Recall: 97% Specifity: 99.9%
CMTracker [119]	Alexa 100k	In-browser	Dynamic	Hash and Stuck based profilers	Thr-based	100% TPR
Musch et al. [120]	Alexa 1M	In-browser	Dynamic	CPU usage	MA	N/A
Tahir et al. [121]	Manually created 320 non-mining and 100 mining websites	In-browser	Dynamic	HPC values	RF	Accuracy: 99.35%, Precision: 100%, Recall: 98%, AUC: 99%
SEISMIC [122]	500 webpages randomly selected from Alexa top 50K	In-browser	Dynamic	Wasm instructions	Matching	F1: 98%
MineThrottle [123]	Alexa 1M	In-browser	Dynamic	Block-level features CPU usage	Matching	FNR: 1.83%
Coinpolice [124]	47k samples	In-browser	Dynamic	CPU usage, HPC, JS/WASM execution time and features, Throttling-independent timeseries	CNN	TPR:97.8 % FPR: 0.74%
Carlin et al. [125]	Captured Opcode trace packets Virusshare (296 Samples)	In-browser	Dynamic	Opcodes	RF	TPR: 99.2 % FPR: 0.9, Precision: 99.2 Recall: 99.2
Liu et al. [126]	1159 samples collected from browsers' memory snapshot	In-browser	Dynamic	Heap snapshots Stack Features	RNN	Precision: 95, Recall: 93
Rauchberger et al. [127]	Alexa: 1M	In-browser	Dynamic	Web socket traffic	Matching	N/A
Caprolu et al. [128]	N/A	In-browser	Dynamic	Network traffic	RF,KFCV	TPR=92% ,FPR=0.8%
MINOS [129]	WASM Samples collected via PublicWWW	In-browser	Dynamic	Image frames of malicious samples	CNN	Accuracy: 98.97%
Yulianto et al. [130]	PublicWWW and Blacklists	In-browser	Static and Dynamic	CPU usage	Matching	TPR:100%
CMBlock [131]	In-browser cryptojacking samples	In-browser	Static and Dynamic	Blacklists Behaviour	N/A	N/A
Gangwal et al. [132]	Combination daily user tasks and miners[1]	Host-based and In-browser	Dynamic	hardware events (e.g., branch-misses), software events (e.g., page-faults) hardware cache events (e.g., cache-misses)	RF, SVM	Recall: 97.84% Precision: 99.7% Accuracy:98.7%
Lachtar et al. [133]	N/A	Host-based and In-browser	Dynamic	CPU instructions	Matching	TPR:100 % FPR: ¡ 2%
Tanana et al. [134]	40 In-browser and 10 executable-type cryptojacking	Host-based and In-browser	Dynamic	CPU utilization share RAM usage	N/A	TPR: 81%
Ahmad et al. [88]	Mixture of Benign and Malicious Network Packages	Host-based and In-browser	Dynamic	Network traffic	DCA	N/A
DeCrypto Pro [135]	Ĩ200 samples	Host-based and In-browser	Dynamic	HPC, CPU usage	k-NN, RF, LSTM	FPR: 2.5, Precision: 96, Recall: 97
Darabian et al. [97]	1500 active cryptomining collected from Virustotal in 2018	Host-based	Static and Dynamic	System calls, opcode sequences	RNN, CNN	System calls (best): LSTM: Accuracy:99% F1: 98% MCC: 98% FPR:0.6%
Crypto-Aegis [128]	Network traffic of 3 legitimate mining scripts and 3 daily user applications	Host-based	Dynamic	Packet sizes Interarrival times	RF	TPR:80-84% FPR: 0.9 - 1.2%

[1] The dataset was not available as of writing this paper (November 1, 2020).

[2] Support Vector Machine: SVM, Random Forest: RF, Decision Tree: DT, Convolutional Neural Network: CNN, Recurrent Neural Network: RNN, Incremental Learning: IL, Threshold-based: Thr-based, Manual Analysis: MA, Dendritic Cell Algorithm: DCA, k-Nearest Neighbors: k-NN, Light-weight machine learning models: LSTM, Symantec RuleSpace Engine:SRSE, k-Fold Cross Validation:KFCV

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

TABLE 2. The list of publicly available blacklists.

Ref	Link
Nocoin [106]	https://github.com/keraf/NoCoin
CoinBlocker [136]	https://zerodot1.gitlab.io/CoinBlockerListsWeb/index.html
Minerblock [107]	https://github.com/xd4rker/MinerBlock/blob/master/assets/filters.txt
Coinhive Blocker [141]	https://raw.githubusercontent.com/Marfjeh/coinhive-block/master/domains
Andreas CH Blocker [142]	https://raw.githubusercontent.com/andreas0607/CoinHive-blocker/master/blacklist.json

TABLE 3. The list of open-source cryptojacking malware detection implementations.

Ref	Implementation Link	Description	Last Update
CMTracker [119]	https://github.com/deluser8/cmtracker	code	Sep 21, 2018
Minesweeper [112]	https://github.com/vusec/minesweeper	data and code	Mar 17, 2020
OUTGUARD [116]	https://github.com/teamnsrg/outguard	data and code	Sep 6, 2019
SEISMIC [122]	https://github.com/wenhao1006/SEISMIC	code	Sep 10, 2019
Retro Blacklist [143]	https://github.com/retrocryptomining/	data and code	Jul 16, 2020

lector is a feature of the JS programming language to optimize memory usage, and it deletes unnecessary data from memory and prevents memory overloading. The memory and CPU continuously interact with each other during the mining operation, and the CPU sends calculated data to the memory. The garbage collector deletes all calculated hash values one by one after being sent to the service provider; therefore, the mining process causes irregular usage of the garbage collector. Due to this behavior, the garbage collector can be used as a feature for the dynamic detection mechanism. Iframes are the HTML tags used for embedding another program/function to an HTML source code. Mining scripts are inserted into those tags and work under HTML codes. Similar to previous features, cryptojacking scripts cause irregular usage in iframe resource loads. This feature cannot be used as a primary feature because too many modern web applications use iframe resources irregularly, and it may cause a high false-positive rate.

- *Hardware Performance Counter (HPC) [121], [124], [135]:* HPC values [144] are used on modern computers' CPUs and keepthe record of internal CPU events (e.g., Cycles, Cache misses). The values of the registers with CPU clock cycles and executed instructions provide unique information about the behaviors of a running application. Several studies check the hardware activities and the related applications with HPC values to detect the cryptocurrency mining operations on the system.
- *System calls [97]:* System calls are the API structures that enable the connection between applications and the running system's kernel. System calls run with level 0 privileges to invoke calls and request services from the OS's kernel. The proposed detection system in [97] uses the system calls for dynamic analysis, and system calls are recorded using the Cuckoo Sandbox. Then, the system calls are used to train deep learning models, and they achieve 99% accuracy.

6.1.4. Classifier and Performance. The collected features are mostly used to train different machine learning classifiers such as Support Vector Machine (SVM) [113], [116], [132], Random Forest [121], [132], Neural Network [97], [117], Decision Tree [114]. Moreover, Neto et al. [118] proposed the use of incremental learning, which takes the classification probabilities of an ensemble of classifiers as a feature for an incremental learning process. Moreover, Hong et al. [119], proposed a threshold-based detection, and the studies in [112], [122] used a static matching method to detect certain functions in the script. Musch et al. [120] only report the number of detected websites in the Top 1M Alexa websites. As can be seen from Table 1, all classifiers achieve a near-perfect (∼100%) detection results.

6.1.5. Open Source Implementations. Finally, some of the studies [112], [116], [119], [122], [145] published their code to help the research community.Table 3 presents the list of open-source cryptojacking malware implementations.

6.2. Cryptojacking Prevention Studies

A majority of the detection mechanisms do not focus on preventing or interrupting of cryptojacking malware; however, there are still several studies [123], [130], [131] focusing on both the detection and prevention of cryptojacking malware. Using dynamic features to detect ongoing cryptojacking is like other dynamic analysis studies, but their prevention methods vary. While Yulianto et al. [130] only raises a notification, Bian et al. [123] sleep the mining process, and Razali et al. [131] directly kill the related process.

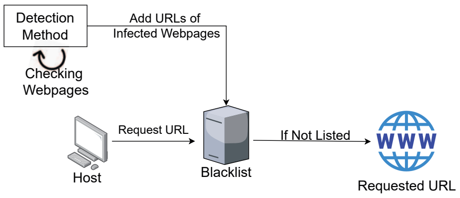

Figure 4. Blacklisting method.

For cryptojacking prevention, there are also several tools in the market. Against host-based cryptojacking malware, proprietary antivirus programs [110], [157][3] are commonly preferred. Against in-browser cryptojacking malware, open-source browser extensions such as NoCoin [106] and MinerBlock [107] are widely used. These open-source browser extensions are based on blacklisting, where the lists are updated as new malicious domains are discovered. Table 2 shows the list of publicly available blacklists that we identified during our research. Browser extensions warn the user when the user wants to access a

[3] As these programs are closed-source, their methods are not publicly available.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

TABLE 4. Cryptojacking malware analysis studies in the literature.

Ref	Cryptojacking Dataset	Sample Type	Focus of the Study
[146]	2000 executable	binary	the practice of using compromised PCs to mine Bitcoin
[147]	33282 websites	script	prevalence analysis
[148]	-	-	how cybercriminals are exploiting cryptomining
[74]	5190 websites	script	campaign and domain analysis
[149]	XMR-stak, cpuminer-multi	binary	attack impact on consumer devices and user annoyance
[150]	5700 websites	script	static, dynamics and economic analysis
[151]	CoinHive cryptominer	script	sample characteristics and network traffic analysis
[46]	1.2 million miners	binary	currencies, actors , campaign and earning analysis, underground markets
[152]	107511 websites	script	profitability and the imposed overheads
[15]	3.2 TB historical scan results	script	investigation of a new type of attack that exploits Internet infrastructure for cryptomining
[153]	-	-	business model, threat sources, implications, mitigations, legality and ethics
[154]	53 websites	script	sample characteristics
[155]	2770 websites	script	activeness analysis
[156]	XMRig miner	binary	sample characteristics
[143]	156 domains, 25892 proxies	script	impact on the web users

website on the blacklist. Figure 4 shows the blacklisting process, which is repeated as a continuous loop.

Pure blacklisting-based prevention is not an efficient way for stopping cryptojacking malware because attackers can easily change their domain by domain fluxing or other methods to downshift the effects of blacklists. There are also some new methods [158] proposed by researchers for better and more optimized blacklisting, but even dynamic blacklisting methods are not fully effective nor protective [159] against domain fluxing methods.

6.3. Cryptojacking Analysis Studies

In addition to the cryptojacking malware detection and prevention studies, some researchers also performed empirical measurement studies to understand the cryptojacking threat landscape better. Table 4 shows cryptojacking malware analysis studies in the literature. In these studies, cryptominers are either in the format of binary [46], [146], [149], [156] or script [74], [143], [147], [150]–[152], [154], [155] except [148], [153] where the findings in these studies are based on the other studies and publicly available documents.

Researchers analyzed several different perspectives of cryptojacking. In the first study [146], the authors analyze the binary samples identified as engaged in mining operations to characterize their scope, operations, and revenue. This is the first and only study analyzing Bitcoin miners, where the samples used in other studies are mining Monero. The increase in the cryptojacking malware attack instances in 2017 also drew researchers' attention. [147] is the first study analyzing the Monero cryptojacking samples, where the authors used over 30000 websites utilizing *coinhive.min.js* library for the prevalence analysis of cryptojacking samples. Many follow-up studies are published. For example, the studies [151], [154], [156] also performed an analysis of the cryptojacking samples to identify characteristics of the samples. In addition, the studies in [143], [149] performed the impact analysis. Particularly, [149] analyzed the attack impact on consumer devices and user annoyance, and [143] analyzed the impact of cryptojacking malware on web users, while [152] analyzed the overhead of cryptojacking samples. In an interesting study, the authors in [15] investigated a new type of attack exploiting the Internet infrastructure for cryptomining, which indeed has an impact on 1.4M infected routers.

Moreover, there are also studies performing the economic analysis of cryptojacking samples such as [46], [150], [152]. Other than that, the authors in [46], [74] performed a campaign analysis of the cryptojacking samples and [155] analyzed the activeness of cryptojacking threat after the discontinuation of Coinhive. Finally, while [148] gives an overview of how cybercriminals are exploiting cryptomining, [153] presents a review of the business model, threat sources, implications, mitigations, legality, and ethics of cryptojacking malware.

7. Lessons Learned and Research Directions

This section covers lessons we learned during this research and potential research directions that can further be explored by other researchers:

A recent trend in cryptojacking attacks.

Some security reports published in 2020 [16], [82] noted a trend in the cryptojacking attacks, in which the attackers now target the devices with more processing power rather than the personal computers as in the in-browser cryptojacking attacks. With this, the attackers' goal is to obtain more profit in a lesser time. Some examples of these targeted devices are enterprise cloud infrastructure [19], [160], servers [161], a large number of inadequately protected IoT devices [18] or Docker engines [83]. In these attacks, the attackers did not only use the Coinhive's script but also modified a non-malicious and open-source Monero miner called XMRig to perform the cryptomining in the background [162]. Unlike in-browser cryptominers, the client does not come to the attacker; therefore, the attacker needs to deliver the malicious mining script to the victims. For this purpose, the attackers used the vulnerabilities as in the case of Mikrotik routers [23] or a recent CVE to deliver Monero cryptominer [22], poorly configured IoT devices [18], or poor security [83]. It is also seen that insiders may want to take advantage of the servers [9].

However, despite the decrease in the number of in-browser samples from active service providers and the potential trend shift in the attackers' behavior to host-based cryptojacking malware and techniques used to deliver the malware, host-based cryptojacking malware literature is not as rich as in-browser cryptojacking malware literature. As can be seen from Table 1 and 4, there are only several studies on the detection [88], [97], [128], [132]–[135] and the analysis [46], [146], [149], [156]. Therefore, there is a need for more effort from the security researchers to find better solutions to detect and mitigate this continually evolving threat.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

Monero as a target cryptocurrency. In recent attacks, Monero has become a de-facto cryptocurrency for the cryptojacking attacks. Another pattern we spotted is that in almost all of the attack instances in the previous section [7], [20], [49], [83], [84], [160], [163], the attackers use Monero as a target cryptocurrency instead of Bitcoin or other cryptocurrencies. Even we are not sure about their motive, Monero is the most popular privacy coin hiding the track of the transactions. For example, if the attackers would use Bitcoin, even though the attack has been detected after a long time, it would be possible to track down the Bitcoin transactions.

The evaluation of the proposed solutions. We identified threes issues regarding to the evaluation of the proposed solutions in the literature:

- *Dataset dates.* The effectiveness of an in-browser cryptojacking malware detection mechanism is directly related to the number of websites detected. However, the infected websites modify or move their script to other domains frequently to avoid being blacklisted. Moreover, many websites discontinued mining after the Coinhive shutdown [164]. Therefore, the accuracy of a detection method may significantly vary depending on when the dataset was collected. Only five of the proposed detection mechanism [112], [116], [119], [122], [143] in Table 1 reports the dataset date. Therefore, we do not know most of the studies' dataset collection date; which makes a fair comparison difficult.
- *Online vs. offline detection.* The detection mechanisms proposed in the literature usually focus on accuracy as an evaluation metric, and they mostly claim a near-perfect accuracy in detecting cryptojacking malware. However, most of the time, they do not report how their method was implemented, that is, whether it was offline or online. In offline detection, the sample is detected randomly and added to the database (e.g., signature, blacklist). In the online detection, the sample is detected in a real-time manner. As it has been shown that detection ratio may vary for online and offline detection [165], it is critical for the detection studies to report if the proposed method is implemented in an online or offline environment.
- *Overhead analysis.* Only the authors of the two [112], [116] proposed dynamic analysis tools consider the usability of their detection mechanisms on the end-user side. But, especially for machine learning-based detection methods, using behavioral features may introduce a high overhead on the end-user side. This should be taken into consideration by researchers in future studies.

The legitimate use of in-browser cryptocurrency mining. An issue we identified during our research is that the in-browser cryptocurrency mining was initially started to provide an alternative revenue to the legitimate website owners such as new publishers [166] or non-profit organizations like UNICEF [167]. Later, some service providers such as Coinimp [60], WebMinePool [68] even provided methods for explicit user consent in their implementations. However, with the keyword-based automatic detection and prevention methods such as browser extensions [106], [107] or even browsers themselves [77], [168] blocking the websites containing cryptomining script, this use of web-based cryptomining scripts is not possible anymore. A practical solution to this issue would be asking for the user's explicit consent instead of directly blocking a website trying to upload a mining script. Moreover, there is a need for more effort by researchers to work on the usage of legitimate cryptomining with user consent and knowledge as a funding model.

The use of traditional malware attacks on Bitcoin and blockchain infrastructure. There are two types of Bitcoin- and blockchain-related malware seen in the wild: those that use the Bitcoin and blockchain infrastructure to exploit the victim; or those that use the traditional malware attacks such as key stealing, social engineering, or fake application attacks to exploit Bitcoin and blockchain users. Cryptojacking attacks use the Bitcoin and blockchain infrastructure to exploit the victim's computational power; however, Bitcoin and blockchain users are also exposed to many traditional malware attacks. These attacks specifically aim to obtain Bitcoin and blockchain users' private keys through social engineering methods [169]–[171], fake wallets [172], [173], and key-stealing trojan malware [174]–[176]. Although these attacks and their countermeasures [177]–[179] have been studied extensively in the literature [180], their impact in the Bitcoin and blockchain domain has not been investigated yet and can lead to new research directions.

8. Conclusion

The rapid rise of cryptocurrencies incentivized the attackers to the lucrative blockchain and the Bitcoin ecosystem. With ready-to-use mining scripts offered easily by service providers (e.g., Coinhive [8], and CryptoLoot [4]) and untraceable cryptocurrencies (e.g., Monero), cryptojacking malware has become an essential tool for hackers. The lack of mitigation techniques in the market led to many cryptojacking malware detection studies proposed in the literature. In this paper, we first explained the cryptojacking malware types and how they work in a systematic fashion. Then, we presented the techniques used by cryptojacking malware based on the previous research papers, cryptojacking samples, and major attack instances. In particular, we presented sources of cryptojacking malware, infection methods, victim platform types, target cryptocurrencies, evasion, and obfuscation techniques used by cryptojacking malware. Moreover, we gave a detailed review of the existing detection and prevention studies as well as the cryptojacking analysis studies in the literature. Finally, we presented lessons learned, and we noted several promising new research directions. In doing so, this SoK study will facilitate not only a deep understanding of the emerging cryptojacking malware and the pertinent detection and prevention mechanisms but also a substantial additional research work needed to provide adequate mitigations in the community.

Acknowledgment

We would like to thank VirusTotal for sharing the samples. We also would like to thank the anonymous reviewers, and our shepherd Dr. Christian Rossow 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, and CyberFlorida Capacity Building Program. The views expressed are those of the authors only.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

References

[1] A. Marshall, “Combined crypto market capitalization races past $800 bln,” https://cointelegraph.com/news/combined-crypto-market-capitalization-races-past-800-bln, accessed: 2020-02-28.

[2] Kaspersky, “Kaspersky global reports,” https://www.kaspersky.com/about/press-releases/2019_fear-of-the-unknown, accessed: 2020-10-19.

[3] “The official webpage of both coinhive and authedmine,” http://web.archive.org/web/20190130232758/https://coinhive.com/documentation, accessed: 2020-10-19.

[4] “Cryptoloot,” https://crypto-loot.org/, accessed: 2020-06-20.

[5] D. Goodin, “Miners in youtube ads,” https://arstechnica.com/information-technology/2018/01/now-even-youtube-serves-ads-with-cpu-draining-cryptocurrency-miners/, accessed: 2020-04-13.

[6] K. Parrish, “Uk government plugin based mining,” https://www.digitaltrends.com/computing/government-websites-plugin-coinhive-monero-miner/, accessed: 2020-04-13.

[7] C. Cimpanu, “Miner found at usa department of defense,” https://www.zdnet.com/article/bug-hunter-finds-cryptocurrency-mining-botnet-on-dod-network/, accessed: 2020-04-13.

[8] “Coinhive snapshots that taken from webarchive,” https://web.archive.org/web/20181101000000*/coinhive.com, accessed: 2020-05-23.

[9] A. Milano, “Russian scientists arrested crypto mining nuclear lab,” https://www.coindesk.com/russian-scientists-arrested-crypto-mining-nuclear-lab, accessed: 2021-2-23.

[10] J. I. Wong, “An italian bank’s server was hijacked to mine bitcoin,” https://qz.com/1024930/bitcoin-malware-an-italian-banks-server-was-hijacked-to-mine-bitcoin-says-darktrace/, accessed: 2020-02-17.

[11] C. Cimpanu, “A crypto-mining botnet has been hijacking mssql servers for almost two years,” https://www.zdnet.com/article/a-crypto-mining-botnet-has-been-hijacking-mssql-servers-for-almost-two-years/, accessed: 2020-02-17.

[12] D. Olenick, “Miner into third party zoom,” https://www.trendmicro.com/en_us/research/20/d/zoomed-in-a-look-into-a-coinminer-bundled-with-zoom-installer.html, accessed: 2020-04-13.

[13] E. Kent, “Miner found in popular game in steam,” https://www.eurogamer.net/articles/2018-07-30-steam-game-abstractism-turns-pcs-into-cryptocurrency-miners, accessed: 2020-04-19.

[14] T. Smith, “Miner found at nintendo switch console,” https://bitcoinist.com/nintendo-switch-game-pulled-over-cryptojacking-concerns/, accessed: 2020-04-13.

[15] H. L. Bijmans, T. M. Booij, and C. Doerr, “Just the tip of the iceberg: Internet-scale exploitation of routers for cryptojacking,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019, pp. 449–464.

[16] IBM-Security, “X-force threat intelligence index 2020,” https://securityintelligence.com/series/ibm-x-force-threat-intelligence-index-2020, accessed: 2021-02-16.

[17] Check-Point-Research, “Checkpoint 2020 cyber security report,” https://research.checkpoint.com/2020/the-2020-cyber-security-report/, accessed: 2021-2-23.

[18] L. Greenemeier, “Crytojacking can corrupt the iot,” https://www.scientificamerican.com/article/how-cryptojacking-can-corrupt-the-internet-of-things/, accessed: 2021-2-23.

[19] R. Hackett, “Tesla hackers hacked aws cloud services to mine monero,” https://fortune.com/2018/02/20/tesla-hack-amazon-cloud-cryptocurrency-mining/, accessed: 2020-10-19.

[20] J. M. Augusto Remillano II, “Coinminer, ddos bot attack docker daemon ports,” https://www.trendmicro.com/vinfo/hk-en/security/news/virtualization-and-cloud/coinminer-ddos-bot-attack-docker-daemon-ports, accessed: 2021-2-14.

[21] M. J. Schwartz, “Social engineering attacks for cryptojacking,” https://www.bankinfosecurity.com/cryptocurrency-theft-hackers-repurpose-old-tricks-a-10685, accessed: 2021-2-23.

[22] B. G. Mark Vicente, Johnlery Triunfante, “Cve-2019-2725 exploited, used to deliver monero miner,” https://www.trendmicro.com/en_ca/research/19/f/cve-2019-2725-exploited-and-certificate-files-used-for-obfuscation-to-deliver-monero-miner.html, accessed: 2021-2-23.

[23] C. Cimpanu, “Mikrotik router hack affect 200k routers in the world,” https://www.bleepingcomputer.com/news/security/massive-coinhive-cryptojacking-campaign-touches-over-200-000-mikrotik-routers/, accessed: 2021-2-23.

[24] D. McMillen and M. Alvarez, “Mirai iot botnet: Mining for bitcoins?” https://securityintelligence.com/mirai-iot-botnet-mining-for-bitcoins/, accessed: 2021-2-23.

[25] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[26] I.-C. Lin and T.-C. Liao, “A survey of blockchain security issues and challenges.” *IJ Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[27] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, “Survey of consensus protocols on blockchain applications,” in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[28] W. Wang, D. T. Hoang, Z. Xiong, D. Niyato, P. Wang, P. Hu, and Y. Wen, “A survey on consensus mechanisms and mining management in blockchain networks,” *arXiv:1805.02707*, pp. 1–33, 2018.

[29] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, “Sok: Research perspectives and challenges for bitcoin and cryptocurrencies,” in *2015 IEEE symposium on security and privacy (S&P)*. IEEE, 2015, pp. 104–121.

[30] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[31] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[32] E. F. Jesus, V. R. Chicarino, C. V. de Albuquerque, and A. A. d. A. Rocha, “A survey of how to use blockchain to secure internet of things and the stalker attack,” *Security and Communication Networks*, vol. 2018, 2018.

[33] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[34] R. Zhang, R. Xue, and L. Liu, “Security and privacy on blockchain,” *ACM Comput. Surv.*, vol. 52, no. 3, Jul. 2019. [Online]. Available: https://doi.org/10.1145/3316481

[35] J. Al-Jaroodi and N. Mohamed, “Blockchain in industries: A survey,” *IEEE Access*, vol. 7, pp. 36 500–36 515, 2019.

[36] H.-N. Dai, Z. Zheng, and Y. Zhang, “Blockchain for internet of things: A survey,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[37] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, “Blockchain and iot integration: A systematic survey,” *Sensors*, vol. 18, no. 8, p. 2575, 2018.

[38] K. Jayasinghe and G. Poravi, “A survey of attack instances of cryptojacking targeting cloud infrastructure,” in *Proceedings of the 2020 2nd Asia Pacific Information Technology Conference*, 2020, pp. 100–107.

[39] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system.” 2008.

[40] A. Kurt and et al., “Lnbot: A covert hybrid botnet on bitcoin lightning network for fun and profit,” in *Computer Security – ESORICS 2020*. Cham: Springer International Publishing, 2020, pp. 734–755.

[41] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy *et al.*, “Formal verification of smart contracts: Short paper,” in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security (PLAS)*, 2016, pp. 91–96.

[42] N. Van Saberhagen, “Cryptonote v 2.0,” https://bytecoin.org/old/whitepaper.pdf, 2013, accessed: 2021-02-23.

[43] “Top publications,” https://scholar.google.ca/citations?view_op=top_venues&vq=eng_computersecuritycryptography, accessed: 2021-02-12.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

[44] "Virus total payload scanning and ranking platform," https://www.virustotal.com/, accessed: 2020-02-26.

[45] "Source code search engine," https://publicwww.com/, accessed: 2020-10-16.

[46] S. Pastrana and G. Suarez-Tangil, "A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth," in *Proceedings of the Internet Measurement Conference (IMC)*, 2019, pp. 73–86.

[47] M. Santos, "utorrent update smuggles shady cryptocurrency miner into your computer," https://99bitcoins.com/utorrent-update-cryptocurrency-miner/, accessed: 2020-03-31.

[48] C. McDonald, "Cryptojacking malware hid into emails," https://www.mailguard.com.au/blog/brandjacking-malware-hiding, accessed: 2021-02-23.

[49] K. G. Rakesh Sharma, Akhil Reddy, "A vulnerability used to deliver cryptojacking malware," https://www.fireeye.com/blog/threat-research/2018/02/cve-2017-10271-used-to-deliver-cryptominers.html, accessed: 2021-02-23.

[50] S. Lee, C. Yoon, H. Kang, Y. Kim, Y. Kim, D. Han, S. Son, and S. Shin, "Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web." in *Network and Distributed Systems Security (NDSS) Symposium 2019*, 2019.

[51] A. Goriacheva, N. Jakubenko, O. Pogodina, and D. Silnov, "Anonymization technologies of cryptocurrency transactions as money laundering instrument," *KnE Social Sciences*, pp. 46–53, 2018.

[52] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *International Conference on Financial Cryptography and Data Security (ICFCIDS)*. Springer, 2014, pp. 486–504.

[53] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy, "Tracking ransomware end-to-end," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 618–631.

[54] M. Paquet-Clouston, B. Haslhofer, and B. Dupont, "Ransomware payments in the bitcoin ecosystem," *Journal of Cybersecurity*, vol. 5, no. 1, p. tyz003, 2019.

[55] M. Conti, A. Gangwal, and S. Ruj, "On the economic significance of ransomware campaigns: A bitcoin transactions perspective," *Computers & Security*, vol. 79, pp. 162–189, 2018.

[56] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*. Springer, 2015, pp. 3–24.

[57] "Monero price chart," https://coinmarketcap.com/currencies/monero/, accessed: 2020-04-13.

[58] "The official webpage of browsermine," https://browsermine.com/faq, accessed: 2020-10-19.

[59] "The official webpage of coinhave," http://web.archive.org/web/20180102115842/https://coin-have.com/, accessed: 2020-10-19.

[60] "The official webpage of coinimp," https://www.coinimp.com/documentation, accessed: 2020-10-19.

[61] "Coinnebula official webpage," https://web.archive.org/web/20180818144049/https://coinnebula.com/, accessed: 2020-10-19.

[62] "The official github page of deep miner," https://github.com/deepwn/deepMiner, accessed: 2020-10-19.

[63] "Jsecoin," https://jsecoin.com/, accessed: 2020-10-19.

[64] "The official webpage of monerise," http://web.archive.org/web/20200813110918/http://monerise.com/, accessed: 2020-10-19.

[65] "The official webpage of nerohut," https://web.archive.org/web/20190131001253/https://nerohut.com/documentation.php, accessed: 2020-10-19.

[66] "Webmine official webpage," webmine.cz/, accessed: 2020-10-19.

[67] "The official webpage of webminerpool," https://github.com/notgiven688/webminerpool, accessed: 2020-10-19.

[68] "The official webpage of webmine pool," https://www.webminepool.com/page/documentation, accessed: 2020-10-19.

[69] "Xmrrig," https://github.com/xmrig/xmrig, accessed: 2021-2-23.

[70] J. Grunzweig, "Large scale monero mining operation," https://unit42.paloaltonetworks.com/unit42-large-scale-monero-cryptocurrency-mining-operation-using-xmrig/, accessed: 2021-2-23.

[71] L. Kassem, "xmrig father zeus of cryptocurrency mining malware," https://securityintelligence.com/xmrig-father-zeus-of-cryptocurrency-mining-malware/, accessed: 2021-2-23.

[72] H. Partz, "Ukrainian man faces up to 6 years in jail for cryptojacking on his own websites," https://cointelegraph.com/news/ukrainian-man-faces-up-to-6-years-in-jail-for-cryptojacking-on-his-own-websites, accessed: 2021-2-23.

[73] J. Rüth, T. Zimmermann, K. Wolsing, and O. Hohlfeld, "Digging into browser-based crypto mining," in *Proceedings of the Internet Measurement Conference (IMC) 2018*, 2018, pp. 70–76.

[74] 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.

[75] N. Christopher, "Hackers mined a fortune from indian websites," https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/hackers-mined-a-fortune-from-indian-websites/articleshow/65836088.cms, accessed: 2021-2-23.

[76] T. Claburn, "Google tag manager exploited," https://www.theregister.com/2017/11/22/cryptojackers_google_tag_manager_coin_hive/, accessed: 2021-02-23.

[77] L. H. Newman, "Google bans all cryptomining extensions from the chrome store," https://www.wired.com/story/google-bans-all-cryptomining-extensions-from-the-chrome-store/, accessed: 2020-10-16.

[78] J. Pearson, "A 'fortnite' cheat maker duped players into downloading a bitcoin miner," https://www.vice.com/en/article/8x598p/a-fortnite-cheat-maker-duped-players-into-downloading-a-bitcoin-miner-epic-games-sued, accessed: 2021-2-23.

[79] S. Sebastián and J. Caballero, "Brilliant but evil: Gaming company fined $1 million for secretly using players' computers to mine bitcoin," https://www.forbes.com/sites/kashmirhill/2013/11/19/brilliant-but-evil-gaming-company-turned-players-computers-into-unwitting-bitcoin-mining-slaves/?sh=11f7e958570b, accessed: 2021-2-23.

[80] N. Avital, "New research: Crypto-mining drives almost 90% of all remote code execution attacks," https://www.imperva.com/blog/new-research-crypto-mining-drives-almost-90-remote-code-execution-attacks/, accessed: 2021-2-23.

[81] A. Windsor, "Breaking down a two-year run of vivin's cryptominers," https://blog.talosintelligence.com/2020/01/vivin-cryptomining-campaigns.html, accessed: 2021-2-20.

[82] C. P. Research, "Cloud-based cryptojacking article," https://research.checkpoint.com/2020/the-2020-cyber-security-report/, accessed: 2020-10-19.

[83] Unit42, "Watchdog: Exposing a cryptojacking campaign that's operated for two years," https://unit42.paloaltonetworks.com/watchdog-cryptojacking/, accessed: 2021-02-23.

[84] "Detect large-scale cryptocurrency mining attack against kubernetes clusters," https://azure.microsoft.com/en-us/blog/detect-largescale-cryptocurrency-mining-attack-against-kubernetes-clusters/, accessed: 2021-2-20.

[85] S. R. Department, "Estimated iot device count by 2025," https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, accessed: 2021-2-23.

[86] D. McMillen, "What is the mirai botnet?" https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/, accessed: 2020-11-02.

[87] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[88] A. Ahmad, W. Shafiuddin, M. N. Kama, and M. M. Saudi, "A new cryptojacking malware classifier model based on dendritic cell algorithm," in *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, 2019, pp. 1–5.

[89] "Google bans crypto-mining apps from play store," https://www.bbc.com/news/technology-44980936, accessed: 2020-10-16.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

[90] C. Osborne, "Apple bans developers from submitting cryptocurrency mining apps for ios devices," https://www.zdnet.com/article/apple-bans-developers-from-creating-ios-cryptocurrency-mining-apps/, 2018, accessed: 2020-10-16.

[91] S. Dashevskyi, Y. Zhauniarovich, O. Gadyatskaya, A. Pilgun, and H. Ouhssain, "Dissecting android cryptocurrency miners," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2020, pp. 191–202.

[92] 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.

[93] A. Bender, J. Katz, and R. Morselli, "Ring signatures: Stronger definitions, and constructions without random oracles," in *Theory of Cryptography Conference*. Springer, 2006, pp. 60–79.

[94] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan *et al.*, "An empirical analysis of traceability in the monero blockchain," *Proceedings on Privacy Enhancing Technologies (PETS)*, vol. 2018, no. 3, pp. 143–163, 2018.

[95] P. Vasin, "Blackcoin's proof-of-stake protocol v2," vol. 71, 2014.

[96] E. Duffield, H. Schinzel, and F. Gutierrez, "Transaction locking and masternode consensus: A mechanism for mitigating double spending attacks," *CryptoPapers. info*, 2014.

[97] 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.

[98] "Bitcoin cash official community page," https://www.bitcoincash.org/, accessed: 2020-04-28.

[99] "Litecoin official webpage," https://litecoin.org/, accessed: 2021-2-23.

[100] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[101] "The official webpage of uplexa coin," https://uplexa.com/, accessed: 2020-10-19.

[102] "The official webpage of sumokoin," https://www.sumokoin.org/, accessed: 2020-10-19.

[103] "The official webpage of electroneum coin," https://electroneum.com/, accessed: 2020-10-19.

[104] H. V. Nath and B. M. Mehtre, "Static malware analysis using machine learning methods," in *International Conference on Security in Computer Networks and Distributed Systems (SNDS 2014)*. Springer, 2014, pp. 440–450.

[105] C. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using cwsandbox," *IEEE Security & Privacy*, vol. 5, no. 2, pp. 32–39, 2007.

[106] "Nocoin: Block lists to prevent javascript miners," https://github.com/hoshsadiq/adblock-nocoin-list, accessed: 2020-04-08.

[107] "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: 2020-04-08.

[108] A. Acar, L. Lu, A. S. Uluagac, and E. Kirda, "An analysis of malware trends in enterprise networks," in *Information Security*, Z. Lin, C. Papamanthou, and M. Polychronakis, Eds. Cham: Springer International Publishing, 2019, pp. 360–380.

[109] "Cryptojacking," https://www.malwarebytes.com/cryptojacking/, accessed: 2020-03-29.

[110] Avast, "Avastantimalware," https://www.avast.com/c-protect-yourself-from-cryptojacking, accessed: 2020-04-09.

[111] "Javascript library callig instructions from the official documents," https://javascript.info/modules-intro, accessed: 2020-05-25.

[112] R. K. Konoth, E. Vineti, V. Moonsamy, M. Lindorfer, C. Kruegel, H. Bos, and G. Vigna, "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.

[113] 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 (ACSAC)*, 2018, pp. 313–326.

[114] 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.

[115] 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 *INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1873–1881.

[116] A. Kharraz, Z. Ma, P. Murley, C. Lever, J. Mason, A. Miller, N. Borisov, M. Antonakakis, and M. Bailey, "Outguard: Detecting in-browser covert cryptocurrency mining in the wild," in *The World Wide Web Conference (WWW)*, 2019, pp. 840–852.

[117] "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.

[118] 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.

[119] 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.

[120] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Thieves in the browser: Web-based cryptojacking in the wild," in *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES)*, 2019, pp. 1–10.

[121] 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.

[122] 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.

[123] W. Bian, W. Meng, and M. Zhang, "Minethrottle: Defending against wasm in-browser cryptojacking," in *Proceedings of The Web Conference (WWW) 2020*, 2020, pp. 3112–3118.

[124] I. Petrov, L. Invernizzi, and E. Bursztein, "Coinpolice: Detecting hidden cryptojacking attacks with neural networks," *arXiv:2006.10861*, 2020.

[125] D. Carlin, P. O'kane, S. Sezer, and J. Burgess, "Detecting cryptomining using dynamic analysis," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2018, pp. 1–6.

[126] J. Liu, Z. Zhao, X. Cui, Z. Wang, and Q. Liu, "A novel approach for detecting browser-based silent miner," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 490–497.

[127] J. Rauchberger, S. Schrittwieser, T. Dam, R. Luh, D. Buhov, G. Pötzelsberger, and H. Kim, "The other side of the coin: A framework for detecting and analyzing web-based cryptocurrency mining campaigns," in *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES)*, 2018, pp. 1–10.

[128] M. Caprolu, S. Raponi, G. Oligeri, and R. Di Pietro, "Crypto mining makes noise," *arXiv:1910.09272*, 2019.

[129] 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.

[130] A. D. Yulianto, P. Sukarno, A. A. Warrdana, and M. Al Makky, "Mitigation of cryptojacking attacks using taint analysis," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 2019, pp. 234–238.

[131] M. A. Razali and S. M. Shariff, "Cmblock: In-browser detection and prevention cryptojacking tool using blacklist and behavior-based detection method," in *International Visual Informatics Conference (IVIC)*. Springer, 2019, pp. 404–414.

[132] 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.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

[133] N. Lachtar, A. A. Elkhail, A. Bacha, and H. Malik, "A cross-stack approach towards defending against cryptojacking," *IEEE Computer Architecture Letters*, vol. 19, no. 2, pp. 126–129, 2020.

[134] D. Tanana, "Behavior-based detection of cryptojacking malware," in *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*. IEEE, 2020, pp. 0543–0545.

[135] G. Mani, V. Pasumarti, B. Bhargava, F. T. Vora, J. MacDonald, J. King, and J. Kobes, "Decrypto pro: Deep learning based cryptomining malware detection using performance counters," in *IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 2020, pp. 109–118.

[136] "Coinblockerlists," https://zerodot1.gitlab.io/CoinBlockerListsWeb/, accessed: 2020-06-17.

[137] C. Guarnieri, A. Tanasi, J. Bremer, and M. Schloesser, "The cuckoo sandbox," 2012, https://www.cuckoosandbox.org.

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

[139] A. Rossberg, B. L. Titzer, A. Haas, D. L. Schuff, D. Gohman, L. Wagner, A. Zakai, J. F. Bastien, and M. Holman, "Bringing the web up to speed with webassembly," *Commun. ACM*, vol. 61, no. 12, p. 107–115, Nov. 2018.

[140] "Webassembly," https://webassembly.org//, accessed: 2021-02-23.

[141] "Coinhive blockerproject github page," https://github.com/Marfjeh/coinhive-block, accessed: 2021-2-23.

[142] "Coinhive blocker project," https://github.com/andreas0607/CoinHive-blocker, accessed: 2021-2-23.

[143] 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:2004.13239*, 2020.

[144] S. Das, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, "Sok: The challenges, pitfalls, and perils of using hardware performance counters for security," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 20–38.

[145] "Retro cryptomining project github page," https://github.com/retrocryptomining/data, accessed: 2021-2-23.

[146] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. McCoy, S. Savage, N. Weaver, A. C. Snoeren, and K. Levchenko, "Botcoin: Monetizing stolen cycles." in *Network and Distributed Systems Security (NDSS) Symposium*. Citeseer, 2014.

[147] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark, "A first look at browser-based cryptojacking," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2018, pp. 58–66.

[148] K. Sigler, "Crypto-jacking: how cyber-criminals are exploiting the crypto-currency boom," *Computer Fraud & Security*, vol. 2018, no. 9, pp. 12–14, 2018.

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

[150] 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.

[151] A. Zimba, Z. Wang, and M. Mulenga, "Cryptojacking injection: A paradigm shift to cryptocurrency-based web-centric internet attacks," *Journal of Organizational Computing and Electronic Commerce*, vol. 29, no. 1, pp. 40–59, 2019.

[152] P. Papadopoulos, P. Ilia, and E. Markatos, "Truth in web mining: Measuring the profitability and the imposed overheads of cryptojacking," in *International Conference on Information Security (ISC)*. Springer, 2019, pp. 277–296.

[153] 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.

[154] A. B. A. Aziz, S. B. Ngah, Y. T. Dun, and T. F. Bee, "Coinhive's monero drive-by crypto-jacking," in *IOP Conference Series: Materials Science and Engineering*, vol. 769, no. 1. IOP Publishing, 2020, p. 012065.

[155] S. Varlioglu, B. Gonen, M. Ozer, and M. Bastug, "Is cryptojacking dead after coinhive shutdown?" in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2020, pp. 385–389.

[156] A. Zimba, Z. Wang, M. Mulenga, and N. H. Odongo, "Crypto mining attacks in information systems: An emerging threat to cyber security," *Journal of Computer Information Systems*, vol. 60, no. 4, pp. 297–308, 2020.

[157] Norton, "Official site — norton™ - antivirus, anti-malware software," https://us.norton.com/, accessed: 2020-04-09.

[158] S. Ramanathan, J. Mirkovic, and M. Yu, "Blag: Improving the accuracy of blacklists," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[159] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with dns traffic analysis," *IEEE/Acm Transactions on Networking*, vol. 20, no. 5, pp. 1663–1677, 2012.

[160] Cado-Security, "Aws cloud-based cryptojacking report," https://www.cadosecurity.com/post/team-tnt-the-first-crypto-mining-worm-to-steal-aws-credentials, accessed: 2020-02-16.

[161] W. Foxley, "Coindesk; jacking virus infect 850000 servers," https://www.coindesk.com/crypto-jacking-virus-infects-850000-servers-hackers-on-the-run-with-millions, accessed: 2021-2-23.

[162] T. Spring, "Cryptominer, winstarnssmminer, has made a fortune by brutally hijacking computers," https://blog.360totalsecurity.com/en/cryptominer-winstarnssmminer-made-fortune-brutally-hijacking-computer/, accessed: 2021-2-23.

[163] Palo-Alto-Networks, "Hildegard: New teamtnt cryptojacking malware targeting kubernetes," https://unit42.paloaltonetworks.com/hildegard-malware-teamtnt/, accessed: 2021-02-26.

[164] S. Varlioglu, B. Gonen, M. Ozer, and M. F. Bastug, "Is cryptojacking dead after coinhive shutdown?" *arXiv:2001.02975*, 2020.

[165] AV-Comparatives, "Malware protection test march 2019," https://www.av-comparatives.org/tests/malware-protection-test-march-2019/, accessed: 2020-11-05.

[166] 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: 2020-10-16.

[167] S. Liao, "Unicef mining for charity," https://www.theverge.com/2018/4/30/17303624/unicef-mining-cryptocurrency-charity-monero, accessed: 2020-04-13.

[168] C. Davenport, "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: 2020-10-16.

[169] Redactie, "Analysing a cryptocurrency phishing attack that earns $15k in two hours," https://www.kpn.com/zakelijk/blog/analysing-cryptocurrency-phishing-attack.htm, accessed: 2020-04-13.

[170] "Phishing attack caused 1.7 billion loss," https://cointelegraph.com/news/israeli-citizen-accused-of-stealing-over-17-million-in-crypto, accessed: 2020-04-13.

[171] M. Boddy, "Phishing attack performed on xrp network," https://cointelegraph.com/news/phishing-sites-use-trick-letters-in-domain-names-to-steal-xrp, accessed: 2020-04-13.

[172] Lookout, "Fake bitcoin wallet," https://blog.lookout.com/fake-bitcoin-wallet, accessed: 2020-04-13.

[173] L. Stefanko, "Fake cryptocurrency apps google play bitcoin," https://www.welivesecurity.com/2019/05/23/fake-cryptocurrency-apps-google-play-bitcoin/, accessed: 2020-04-13.

[174] D. Parkin, "Cryptocurrency stealer malware," https://www.express.co.uk/finance/city/1213514/cryptocurrency-fraud-malware-clipper-victims-crv, accessed: 2020-04-13.

[175] C. Cimpanu, "Chrome extension caught stealing crypto-wallet keys," https://www.zdnet.com/article/chrome-extension-caught-stealing-crypto-wallet-private-keys/, accessed: 2020-04-13.

[176] J. Stewart, "Cryptocurrency-stealing malware landscape," https://www.secureworks.com/research/cryptocurrency-stealing-malware-landscape, accessed: 2020-04-13.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

[177] A. Acar and et al., “A usable and robust continuous authentication framework using wearables,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[178] A. Acar and et al., “A privacy-preserving multifactor authentication system,” *Security and Privacy*, vol. 2, no. 5, p. e88, 2019.

[179] Z. B. Celik and et al., “Curie: Policy-based secure data exchange,” in *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY ’19, 2019, p. 121–132. [Online]. Available: https://doi.org/10.1145/3292006.3300042

[180] R. Heartfield and G. Loukas, “A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks,” *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, pp. 1–39, 2015.

[181] S. Sebastián and J. Caballero, “Avclass2: Massive malware tag extraction from av labels,” in *Annual Computer Security Applications Conference*, ser. ACSAC ’20, 2020, p. 42–53. [Online]. Available: https://doi.org/10.1145/3427228.3427261

[182] “Getting started with v2,” https://developers.virustotal.com/reference#file-report, accessed: 2021-2-11.

Appendix A.
Papers Distribution

Figure 5 shows the distribution of cryptojacking-related research papers per year.

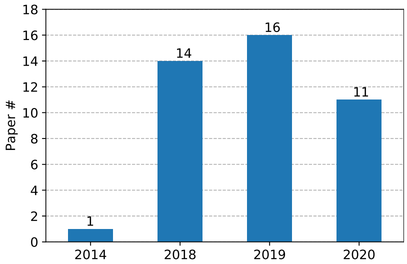

Figure 5. Yearly distribution of 42 cryptojacking-related academic research papers we used in this study.

Appendix B.
Samples Distribution

In this section, our goal is to give more details about the VT and PublicWWW datasets, perform quantitative and longitudinal analysis on our two datasets to confirm some of our findings in the paper, present the limitations of the datasets, and give more insights on the dataset. More details about both VT and PublicWWW datasets can be found on the following website: https://github.com/sokcryptojacking/SoK

B.1. VT Dataset

We used VT Academic API to access the VT dataset consisting of 437279 unique samples (both cryptojacking and non-cryptojacking) and their VT scan reports in the format of JSON. To detect the cryptojacking samples among all samples, we used AV labels in the scan reports of these samples and looked for the keyword ”miner”, i.e., if any of Antivirus (AV) label in the report include the keyword ”miner”, we included in our samples. We picked the keyword ”miner” as we considered it to be the most generic keyword to find all of the cryptojacking samples, and it is also used in recent work as a generic class label for the VT samples [181]. Our scan resulted in the 20200 cryptojacking malware samples. We want to note that this method for selecting cryptojacking samples will not detect the samples that AVs can not label.

B.1.1. More Insights on VT Dataset. In addition to the AV labels that we used to detect the miners, VT scan reports also include other information regarding the samples such as *first seen* date, file type, submission names, the total number of detection by AVs of the samples. We performed more analysis using this information and explain our results in the rest of the section.

Time Distribution: Figure 6 shows the yearly distribution of the samples’ first seen date in the VT academic dataset [182].

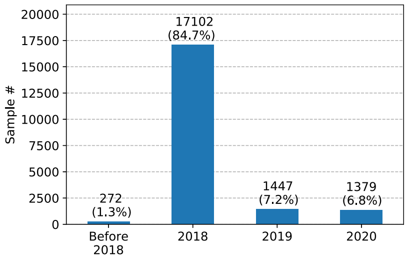

Figure 6. Yearly distribution of VT dataset.

When we continued more analysis on the samples, we found the VT dataset has the following two limitations:

- There are more cryptojacking samples that can be accessed through the VT interface with more privileges. Therefore, this dataset may have bias in terms of the representation of the real-life cryptojacking samples. For example, as can be seen from Figure 6, 85% of all cryptojacking samples in the VT dataset are from 2018; therefore, the samples from 2018 are over-represented in the VT dataset.
- Samples in the VT dataset are uploaded to VT as a batch every six months. Therefore, we concluded that a smaller time frame analysis than the yearly distribution might not be reliable as representing the time distribution of real-life samples seen in the wild.

File Type Distribution: In order to detect the file type, we used the type given by the VT scan reports [182]. Figure 7 shows the top 10 file types of the samples in the VT dataset. According to the figure, HTML is the most common file type in the VT dataset, while the Win32 EXE is the second most common file type.

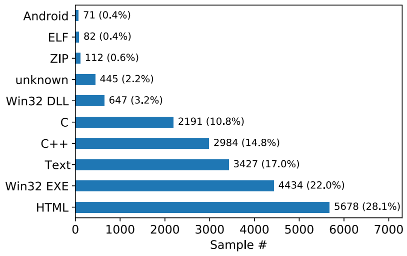

Figure 7. Top-10 file types of the samples in VT dataset.

File type distribution of VT dataset samples is important as they can be used to decide if the sample is an in-browser or host-based type of cryptojacking. Even though some of the file types clearly indicate the type of the cryptojackings, some may require a more in-depth analysis of the sample. In-browser samples only contain the mining script and in the form of text format to embed in the website source code, while the host-based cryptojacking malware samples are in the executable or other formats that can be run on the host machine. For example, we found that all HTML files are in-browser

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.

samples while Win32 EXE and Win32 DLL samples are host-based cryptojacking samples. However, for the file types such as Text, C, C++, ZIP, one needs to check the sample itself and other useful information like submission names to decide whether the sample is in-browser or host-based.

Detection Ratio: VT scan reports includes the detection results of around 60 vendors for each sample [182]. In this part, our goal is to able to see the detection ratio of AVs for the cryptojackings samples in our VT dataset. For this, we plot the histogram of the detection ratio, and the results are given in Figure 8.

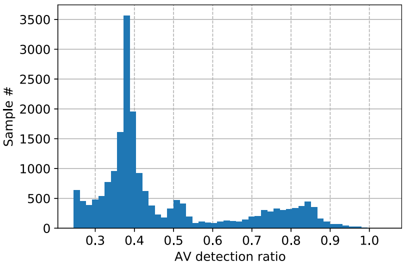

Figure 8. Histogram distribution of detection ratio of the samples in VT dataset.

The results show that the average detection ratio of cryptojacking samples is approximately 40%. We note that every cryptojacking sample in our dataset is detected by at least one AV vendor. This is because of the filtering method we used to find the cryptojacking samples among all samples, i.e., searching for the keyword ”miner” among all AV labels. Therefore, if any AV vendors have not detected a sample, it would not be in our dataset in the first place. This is a limitation of the VT dataset. In order to overcome this limitation and create a recent and more comprehensive dataset, we created another dataset, which we will explain in the next section.

B.2. PublicWWW Dataset

The VT dataset does not include the samples that can bypass the AV detection methods and samples that have never submitted to VT. In order to create a more comprehensive and recent dataset of cryptojacking malware, we used the HTML source keyword search engine PublicWWW [45]. We created the PublicWWW dataset using the following steps:

1) We obtained the keyword lists from the blacklists [106], [107], previous studies [74], [112] and manual analysis of the samples from the VT dataset. Particularly, we used a merged blacklist from NoCoin [106] and MinerBlock [107]; 76 keywords from [74] and 38 keywords from [112]; 25 keywords from the VirusTotal samples.
2) We downloaded the list of URLs for each keyword from PublicWWW.
3) We merged the lists and removed the duplicates to obtain a unique list of URLs.
4) We used a web crawler to download the HTML source code of each URL.
5) We verified the samples by checking the keywords in their source code and removed the samples that do not satisfy this condition.

This process resulted in 6269 unique URLs, their HTML source codes, and their final keyword list with 154 unique keywords used in these samples. From the previous two studies [74], [112] and our findings of publicly known service providers, we identified 14 service providers in total. We manually analyzed their documentation and found that 5328 samples are using the scripts from those 14 service providers. Then, we identified 24 unique keywords to uniquely capture the samples. We released the service provider and keywords lists in our dataset link.

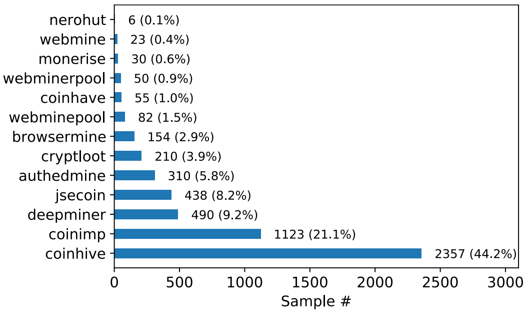

Figure 9. Service provider distribution of the samples in PublicWWW dataset.

Figure 9 shows the service provider distribution of the samples in the PublicWWW dataset. As shown in the figure, even though it is inactive, Coinhive is still the most common service provider among all. On the other hand, Coinimp is the second highest service provider and it is still active as of writing this paper. In addition, we found that 144 samples are using scripts from the multiple samples while we are not able to identify the associated service provider of 941 samples, which we marked as domain lists with an unknown service provider. We also want to note that the samples we have in the PublicWWW are captured during this paper's experiments, but it does not mean that these domains will contain the cryptocurrency mining script any time in the future. Therefore, one may need to re-verify the existence of cryptocurrency mining scripts for their analysis by checking the source code.

Appendix C. Attack Instances Distribution

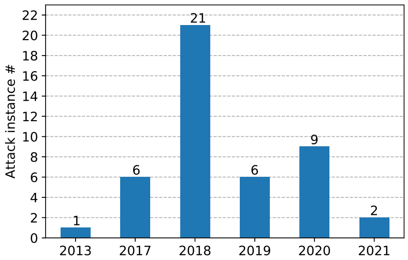

Figure 10. Yearly distribution of 39 cryptojacking attacks instances we used in this paper.

Figure 10 shows the distribution of the attack instances we used in this paper per year. This distribution graph does not show any indicative result regarding the cryptojacking malware's popularity over time in our paper. Only one attack instance from 2013 may seem like an outlier; however, that example shows one of the first instances of cryptojacking malware idea, which is very similar to its usage after 2018. In that attack, a cryptojacking malware attack is instantiated by attaching the sample inside a video game to mine Bitcoin. Finally, in addition to 45 major attack instances, we also added 14 service providers' webpage and 5 blacklists' link and shared in our dataset link.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on June 29,2023 at 17:18:04 UTC from IEEE Xplore. Restrictions apply.