MANiC: Multi-step Assessment for Crypto-miners

Jonah Burgess*,Domhnall Carlin†,Philip O'Kane‡ and Sakir Sezer§
Centre for Secure Information Technologies, Queen's University, Belfast, Northern Ireland
Email: *jburgess03@qub.ac.uk, †dcarlin05@qub.ac.uk, ‡p.okane@qub.ac.uk, §s.sezer@qub.ac.uk

***Abstract*—Modern Browsers have become sophisticated applications, providing a portal to the web. Browsers host a complex mix of interpreters such as HTML and JavaScript, allowing not only useful functionality but also malicious activities, known as browser-hijacking. These attacks can be particularly difficult to detect, as they usually operate within the scope of normal browser behaviour. CryptoJacking is a form of browser-hijacking that has emerged as a result of the increased popularity and profitability of cryptocurrencies, and the introduction of new cryptocurrencies that promote CPU-based mining.**

This paper proposes MANiC (Multi-step AssessmeNt for Crypto-miners), a system to detect CryptoJacking websites. It uses regular expressions that are compiled in accordance with the API structure of different miner families. This allows the detection of crypto-mining scripts and the extraction of parameters that could be used to detect suspicious behaviour associated with CryptoJacking. When MANiC was used to analyse the Alexa top 1m websites, it detected 887 malicious URLs containing miners from 11 different families and demonstrated favourable results when compared to related CryptoJacking research. We demonstrate that MANiC can be used to provide insights into this new threat, to identify new potential features of interest and to establish a ground-truth dataset, assisting future research.

***Index Terms*—CryptoJacking, Drive-by Mining, Crypto-mining, Malware, Browser Security, Web-based Threats**

I. Introduction

In 2008, an entity operating under the pseudonym Satoshi Nakamoto presented a decentralised peer-to-peer (P2P) cryptocurrency called Bitcoin [1]. These pseudonymous virtual coins are generated by a mining process which involves solving a proof of work for a transaction block. If a user successfully mines a block, they are rewarded with Bitcoins. The number of Bitcoins they receive is determined by the number of blocks that have already been mined; every time 210,000 blocks are mined, the reward halves. This process will continue until all 21 million Bitcoins have been mined which is expected to be within the next 120 years [2].

Bitcoin mining becomes more computationally expensive over time, requiring higher computational resources to compete for a continuously decreasing reward. Dedicated hardware, such as ASICs (Application-Specific Integrated Circuits), are now the equipment of choice, designed and deployed specifically for mining purposes. Some users form mining pools, where they share their computational resources and any rewards they receive, increasing the chances of obtaining a reward, but lowering the value [3].

While Bitcoin was the first cryptocurrency, it is certainly not the last. There are currently 1633 cryptocurrencies operating in a market worth over $350 billion [4]. One of these currencies is Monero, known for its privacy-preserving features. The Monero mining process is similar to that of Bitcoin, in that it relies on proof-of-work and rewards miners. However, the computational process is memory-hard, requires a large amount of memory and is better suited to CPU-based mining than the ASIC-based mining that has taken over the Bitcoin mining market [5].

Miners found that the increasing resource costs associated with Bitcoin mining were making the potential reward less enticing, but cyber-criminals found a way to take advantage of Bitcoin mining and the profits it can generate through the production and distribution of crypto-mining malware. This malware infects a victim's computer and configures cryptocurrency mining software, diverting all revenue directly to the attacker, without incurring any of the resource costs. With a large botnet of infected computers [6] or an abundance of computational power [7], this can be a very lucrative attack.

The CPU-friendly design of the Monero mining process enabled a new phenomenon of browser-based mining. In September 2017, CoinHive released a JavaScript file to enable in-browser mining [8] and was quickly followed by other providers offering a similar service, e.g. CoinHave, JSECoin and CryptoLoot. If one of these scripts is integrated into a website, it will hijack the CPU of any visitors to mine Monero.

Ad-blockers and anti-malware vendors quickly started blocking these scripts, but CoinHive argued that their script offers a legitimate alternative to ads [9] and site owners should be responsible for informing their users that the mining is taking place. To combat the blocking of their initial script, CoinHive released a new script called AuthedMine, which will never start mining without explicitly asking for user consent. AuthedMine is currently not blocked by any ad-blocker or anti-virus product [10].

We define CryptoJacking as the unauthorised use of a victim's browser to perform cryptocurrency mining. The remainder of this paper is organised as follows. Section II provides a background into CryptoJacking, while Section III surveys related works. In Section IV we describe our experimental methodology, present our results with an evaluation and discuss system limitations. We conclude in Section V and suggest areas of future work.

II. Background

CryptoJacking is growing as quickly as it emerged, with researchers at AdGuard [11] finding that the presence of mining scripts is increasing by over 30% per month, the most popular being CoinHive with a 95% share of the market. The authors also found that over 33,000 websites were running

miners, earning a combined total of $150,000 per month from over one billion site visits. The data indicate that the same people run the vast majority of mining scripts, with the top three CryptoJackers covering more than 8500 websites.

CryptoJacking rates increased by 8500% in Q4 of 2017 [12], leading anti-malware vendors such as MalwareBytes to list it as the top threat for 2018 [13]. The research in [14] has shown that three of the top four malware threats are browser-based crypto-miners. Other work has suggested that nearly 90% of Remote Code Execution (RCE) attacks on web servers are being used to set up drive-by mining [15], demonstrating a shift in typical RCE payloads from botnets to CryptoJackers. As the value of cryptocurrencies soars, crypto-mining malware becomes more lucrative [16], and the trend of CryptoJacking is likely to continue.

A. Breaches

Crypto-mining scripts may be used legitimately by website owners to increase overall revenue or as a replacement for advertising. Whether this should be considered as malicious or not largely depends on whether the user provides consent. However, the introduction of crypto-miners due to a breach is clearly malicious and can affect websites, third-party services and browser extensions. Due to the potential to serve mining scripts to vast numbers of victims, legitimate high-traffic services are extremely valuable targets for hackers (i.e. watering-hole attacks).

a) Websites: Notable website breaches include the LA Times [17] and Tesla's public cloud service [18], which were compromised due to a poorly configured Amazon Web Service (AWS). Two other prominent sites caught running CoinHive were Showtime and UFC. Showtime declined to make a statement on the matter, and some researchers pointed out that it was likely due to a third-party service called New Relic [19]. However, New Relic denied the accusations and suggested that Showtime developers added the miner. UFC has not responded to questions, so the origin of the miner is unknown [20].

b) Third-party Services: Many websites utilise third-party services such as tracking and analytic tools, JavaScript libraries and advertisements. This presents an opportunity for the injection of coin-mining scripts into the third-party services and consequently, the introduction of the scripts into all websites using the service. Researchers at Trend Micro discovered one example of this threat, observing a 285% rise in the number of CoinHive miners in a single day and determined that the traffic came from DoubleClick advertisements [21].

A popular plugin called Browsealoud was compromised and used to serve the CoinHive miner to over 4200 websites [22] including the UK's Information Commissioner's Office and NHS [23]. Another plugin called LiveHelpNow was compromised and used to perform mining on around 1500 sites [24]. PolitiFact, a website designed to verify the factual accuracy of statements made by US politicians, was compromised via a third-party JavaScript library. This led to CoinHive being executed at 100% CPU usage for an unknown portion of the site's 3.2 million monthly visitors [25].

c) Browser Extensions: Browser extensions present a similar threat to third-party services. As an additional bonus for attackers, the reach of the crypto-mining script is not limited to specific websites or services. Archive Poster, a popular Chrome extension, was running a crypto-miner on an unknown portion of their user base for several days, allegedly due to the credentials of a developer being stolen [11]. A Firefox add-on called Image Previewer was also caught running a crypto-mining script. The code was base64 encoded to bypass detection and the throttle threshold was set at 50% [26].

B. Implications

There are some dire threats posed by the desire for increased computational resources to mine cryptocurrencies. A critical infrastructure security firm reported that it discovered crypto-mining malware in the operational technology network of a water utility in Europe [27]. This is the first known instance of crypto-mining malware being used against an Industrial Control System (ICS) and the malware used sophisticated techniques to prevent detection. The company said that the attack had a significant impact on the system and could have caused it to hang or crash.

The threat posed to ICS is not the only way that crypto-mining malware could lead to real physical harm. Recently, a piece of Android malware called Loapi was discovered on mobile phones. It was being used to mine Monero so aggressively that it caused visible physical damage [28]. Researchers at Kaspersky tested the malware in a lab and found that after two days, the mining caused the battery to bulge so severely that it warped the case of the phone.

C. Mitigations

Traditional security products such as anti-virus applications may block CryptoJacking scripts by default. Specific browser extensions such as NoCoin, minerBlock and NoScripts will also provide protection, as will many off-the-shelf ad-blockers [29]. These solutions typically rely on signature-based scanning or URL blacklisting. Websites may implement Subresource Integrity (SRI) and Content Security Policy (CSP) to help protect against CryptoJacking [30].

SRI helps to reduce threats by verifying third-party code integrity using a SHA-256 hash. Any mismatch indicates the code has been modified and will be blocked until the webmaster verifies the new hash. CSP provides a whitelist-based approach to web security and can be used to ensure scripts are only loaded from trusted providers. When used in conjunction, SRI and CSP can provide a high level of protection against CryptoJacking and general web-based malware [31].

D. Evasion

CryptoJacking sites can make use of existing evasion and anti-analysis techniques employed by web malware. One such technique is obfuscation, which may be applied to bypass signature-based detection. In the case of JavaScript, this can include whitespace, comments, string manipulation, number

substitution, encoding, variable and function name randomisation, encryption, code logic modification, script division and DOM-based obfuscation [32].

Sites can employ URL randomisation to bypass blacklists, this was observed with Minr, a crypto-mining script that provides automatic code obfuscation and periodically checks blacklists, modifying URLs accordingly [33]. In a similar effort to avoid blacklists, websites can load crypto-mining scripts via proxies [34]. Researchers at MalwareBytes found that CryptoJackers were also able to achieve persistence. They accomplished this by hiding a pop-up window beneath the taskbar to ensure the mining continues long after the browser is closed [35].

E. Ethics and Legality

The legality of CryptoJacking is unclear. Without explicitly obtaining user consent or at least informing users, the practice could be considered a theft of computing resources. Due to its recent emergence, there are few tested legal cases. A browser-based mining company called Tidbit faced a legal challenge in 2015 and eventually settled, agreeing that they would cease operations entirely. The Attorney General at the time stated that "No website should tap into a person's computer processing power without clearly notifying the person and giving them the chance to opt out." [36]. This verdict may be an indication of how future trials are likely to conclude.

Ethically speaking, CryptoJacking is a grey area. It is obvious that CryptoJacking resulting from a breach is both illegal and unethical. There is also a consensus that without user consent or awareness, the practice is unethical. Some researchers suggest that even with user consent, the ethics are blurred because users may not technically understand what they are agreeing to and what they are receiving in return [37]. This argument could be applied to other areas such as tracking and advertisements, which are now regulated and accepted as the norm.

III. Related Work

Eskandari et al. [5] provided a first look at browser-based CryptoJacking. To evaluate the impact of the threat, they used a Censys.io BigQuery dataset to determine how many of the top one million websites indexed by Zmap contained the coinhive.min.js library over a 3-month period. They observed the usage of CoinHive miner scripts rise from zero to over 1200 before stabilising at around 800. They verified their results using the PublicWWW search engine, which indexes the source code of public websites. Using this method, they found 30,611 websites running CoinHive and 2671 sites running other miner families.

Later, the authors used the same method of searching PublicWWW to identify whether the blocking of CoinHive had led to the increased usage of different miner families, but found that 92% of websites running miners were still using CoinHive. They broke down the 8% of non-CoinHive miners and found that JSEcoin and Crypto-Loot were the most popular, with a 43% and 26.4% market share respectively. The remaining 30.6% share was split between four smaller crypto-miner families.

This paper was published as we were in process of analysing the data collected from our experiments and served as a useful benchmark for our results. The work differs from ours in that it focuses on detecting the presence of mining scripts on web pages. Our method goes further by extracting and storing data to gain a greater understanding of CryptoJacking and develop new detection techniques. Furthermore, simply checking for a string such as 'coinhive.min.js' anywhere within the HTML body produces false positives (FPs) as many sites mention the script in a non-malicious context. It also produces false negatives (FNs) as some sites load mining scripts via a proxy. Our technique is less prone to these incorrect classifications.

Liu et al. [38] presented BMDetector, a framework that detects in-browser crypto-miners by hooking JavaScript in the kernel source of Chrome Webkit, and analysing the data structure features obtained from the browser heap snapshot and stack data. Capturing this data at the parser level of the browser ensures that any obfuscation or encryption is removed before performing feature extraction. BMDetector uses these features to perform automated detection based on RNN (Recurrent Neural Networks) and passes its findings to a cloud analysis module for verification. When applied to 1159 samples, experimental results demonstrated a 98% detection rate for original samples and 92% rate for encrypted and obfuscated samples. This work differs from ours in that it takes a dynamic approach to detection as opposed to our static, crawler-based approach.

Wang et al. [39] introduced SEISMIC (Secure In-lined Script Monitors for Interrupting CryptoJacks), a dynamic method of identifying in-browser crypto-miners that focuses on semantic features, which are more difficult to obfuscate than syntactic features. CryptoJacking scripts typically use WebAssembly (Wasm), a binary format that allows C, C++ and Rust code to execute in the browser with a similar performance to native code. The authors utilise this knowledge in their approach, which monitors Wasm scripts as they execute to derive a statistical model of known mining and non-mining behaviour. They used Intel Processor Tracing (PT) to record native instruction counts for different types of Wasm web apps (random, video, game/graphics and mining). Next, they manually identify the top 5 Wasm bytecode instructions and use the normalised count of their occurrences as feature vectors. Finally, they use Support Vector Machine (SVM) with linear kernel function and evaluate their approach using stratified 10-fold cross validation 1900 samples, 500 of which are miners. The results show that all mining scripts are identified correctly and, the overall accuracy is 98% or above in all cases. The authors point out that, although their technique is robust against syntactic obfuscation, semantic obfuscation could be applied to crypto-mining scripts to bypass detection, though this would incur a performance cost. They suggest that future work should investigate this area when such attacks have been demonstrated, and samples are available. This work differs from ours in that it uses a dynamic detection method with the

goal of identifying obfuscated scripts whereas ours focuses on a purely static approach to detection, with the primary objective of building a reliable, ground-truth dataset that will enable future research.

Rüth et al. [40] built a database of around 160 Wasm code signatures that performed mining activities. They compared this signature base with Wasm code found on the Alexa 1M and the .org TLD and determined that most instances of Wasm code contained mining functionality, the majority of which was CoinHive (75%). They compared their results with the NoCoin blocklist and found that some sites blacklisted by NoCoin did not integrate Wasm code. Random manual inspections of these sites confirmed that they were false positives. Similarly, NoCoin failed to identify many sites that contained malicious Wasm mining code. Up to 82% of the sites detected using Wasm signatures are not detected by block lists.

The authors go on to analyse the short link forwarding service provided by CoinHive. A third of all links are created by a single user and around 85% of links are contributed by 10 users. The majority of these short links can be resolved in less than 51 seconds (1024 hashes). Analysis of these URLs shows that the most common link destinations are entertainment & music and filesharing. Finally, the authors attempt to verify the CoinHive network size and based on their 4-week observation data, estimate that it contributes around 1.18% of the mining power of the Monero network. This equates to about $250,000 a month based on the current price of Monero.

Hong et al. [41] proposed MineSweeper, a system that leverages several intrinsic characteristics of crypto-mining code to detect CryptoJacking, even when obfuscation techniques have been applied. The authors crawled the Alexa top 1m websites, visiting three random internal pages of each site to maximise the chance of detection. Each webpage is stored alongside any embedded JavaScript and associated network requests and responses. Initially, an offline parser is applied to the collected data to filter out known mining families using string-based pattern matching. The parser then identifies the Wasm-based mining payload using one of two methods. If the mining payload is compiled at runtime, this is achieved by dumping all JavaScript code and searching for keywords relating to the CryptoNight hashing library. If the payload is retrieved from an external server (raw or pre-compiled), this is accomplished by analysing the network requests and responses to and from the browser's web worker. Since keyword-based detection is vulnerable to obfuscation techniques, MineSweeper also checks for other indicators related to crypto-mining. The WebSocket frames are logged and analysed to detect any communication with mining pools by searching for keywords related to the Stratum protocol which is commonly used. The CPU usage is recorded although this is used to assist analysis and gain insights rather than for detection as the system runs in docker which has various processes which may contribute to the overall CPU usage. Also, each site is only loaded for 4 seconds during which time the CPU usage is likely to be high due to the browser performing its initial loading of any required resources.

During the experiment, the authors detected 1735 websites in the Alexa top 1m performing crypto-mining with an estimated profit of $188,878.84 with the most profitable website earning an estimated $17,166.97. They found that 42.88% of the detected websites only applied the crypto-mining scripts to internal web pages and 82.14% of drive-by mining services used one or more obfuscation techniques such as packed code, charCode, renaming, dead code injection and URL randomisation.

Konoth et al. [42] developed a CMTracker, a behaviour-based detection system which initially leverages the Chrome Devtools Protocol (CDP) to crawl websites and perform stack sampling. The JSON files produced during this stage are used as the input for two behaviour-based profilers, hash based and stack structure based. The hash based profiler identifies low-level hashing functions associated with crypto-mining and calculates the cumulative time that the website spends using these functions. The reasoning behind this is that 99% of the top 100 Alexa websites spend less than 0.47% of their total execution time on these functions, whereas crypto-mining sites spend most of their time hashing. Based on this analysis, they use a threshold of 10% to classify sites as malicious.

The authors recognise that the hash based profiler relies on text-based pattern matching to identify the hashing functions, which can easily be obfuscated. To combat this weakness, they also apply the stack structure-based profiler. This technique is based on the observation that crypto-mining websites run heavy workloads with repeated behavioural patterns, but typical websites rarely repeat the same calling stack for more than 5.6% of the execution time. They focus this profiler on dedicated threads which are commonly used by crypto-mining websites and use a threshold of 30% to classify sites as malicious.

The authors state that CMTracker detected 2770 unique crypto-mining samples from 853,936 popular web pages, including 868 among the Alexa top 100k. They analyse the collected data in order to provide numerous insights such as the distribution of CryptoJacking domains by website category, the profitability and energy costs of CryptoJacking, the distribution of wallet IDs, the life-cycle of malicious miners, the effectiveness of blacklists, the use of evasion techniques (methods and prevalence) and the infrastructure of mining campaigns.

Vierthaler et al. [43] presented WebEye, a system which automates the collection of malicious HTTP traffic. Their goal was to produce realistic datasets of correctly classified malicious web traffic which could assist researchers in future work, particularly machine learning (ML). They sourced URLs from the Alexa top 1m websites, MalwareDomainList and Openphish and fed them into their Selenium-based web crawler. WebEye extracts 58 features from websites and aggregates the data with metadata obtained from external sources (GeoIP and Whois).

The authors used Google Safebrowsing, Virustotal and ClamAV to separate the 500gb of data collected by WebEye into malicious and benign samples. Out of the 43 million

samples, about 20,000 (0.5%) were labelled as malicious. To demonstrate how the WebEye dataset can be used for ML, the authors applied a Random Forest classifier and achieved a 99% true positive (TP) rate. They indicated that further optimisations would be needed to improve the FP rate which was recorded as 17.32%.

This work is similar to ours in that it is a web-crawler and primarily focuses on building a dataset for future work. However, WebEye collects all web traffic where we focus specifically on CryptoJacking sites. WebEye also extracts 58 features from websites to assist in future ML research. While MANiC does not currently extract these features, we plan to adapt it in future work to extract a subset of these features in conjunction with JavaScript obfuscation-based features and new features which are specific to CryptoJacking, e.g. CPU usage.

IV. Experiment

A. Methodology

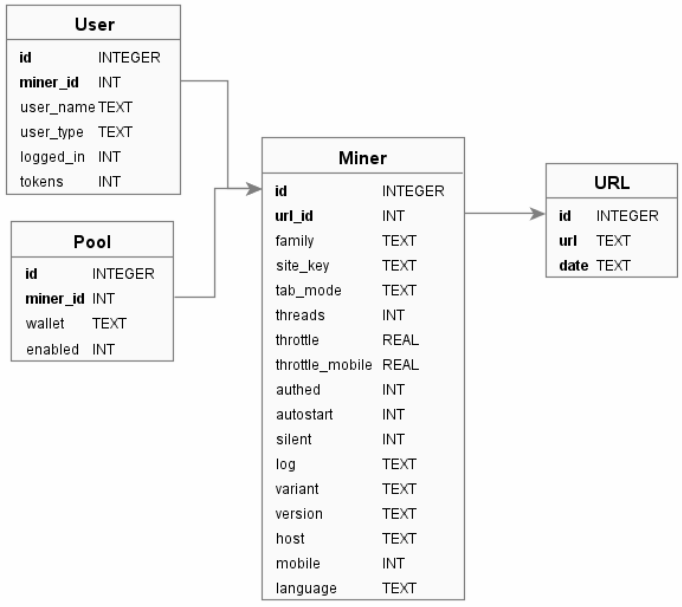

Fig. 1. Miner object structure and content (extracted parameters)

We developed MANiC, a python-based crawler implemented using Scrapy [44], a fast and powerful scraping and web crawling framework. When provided with a list of URLs, MANiC opens each in succession, logging any that fail to load. If the webpage opens without error, the contents of all script tags on the page are extracted and parsed, discarding any duplicates in the process. To scan for miners, MANiC uses a collection of regular expressions which are based on the API syntax structure used to implement and configure various miner families.

If MANiC detects a match, it attempts to extract 21 parameters and store them in a miner object as shown in Figure 1. Any miner objects that do not contain a family and site key are discarded and the page is flagged as suspicious for manual analysis. MANiC will continue this process until all scripts have been tested for each miner family, irrespective of whether the URL has already been classified as malicious. This is important because some websites host multiple mining scripts which are from different families or use different configurations.

If MANiC successfully detects and extracts a miner, it logs the URL as malicious and exports the objects to both JSON and SQL format. The malicious JavaScript and entire HTML body are also stored for future work. If MANiC fails to detect any miners, it searches the entire HTML body for suspicious keywords that indicate a miner may be present. If it finds any of these keywords on the page, it logs the URL as suspicious and records the detected keywords to assist manual analysis later. If MANiC fails to extract any miner objects or detect any suspicious keywords, it logs the URL as benign.

We tested MANiC against a collection of 5441 URLs that had previously been identified as CryptoJacking sites, but the dataset was problematic for a few reasons. Firstly, the dataset was almost six months old, and due to the recent emergence and publicity of crypto-mining, some sites are likely to have removed or deactivated the scripts after a trial period. Similarly, sites which had been injected with crypto-mining scripts resulting from a breach may have since detected and removed the scripts. Secondly, the technique used to detect the crypto-mining scripts was not explained, and therefore the accuracy cannot be verified.

Although we could not rely on the data for our experimental results, the data proved useful as we were able to manually analyse MANiC's results and use the insights to fine-tune regular expressions and improve the detection process. This was an iterative process and involved manually analysing URLs detected as benign, suspicious and malicious and updating MANiC accordingly. To evaluate the state of CryptoJacking on the web, we ran MANiC against the Alexa top 1 million websites (March-June 2018). We performed this experiment on a desktop machine using Ubuntu 17.10, an Intel Core i7-6700 3.4GHz CPU, 16GB RAM and a 250MB internet connection.

B. Results

TABLE I
Detection results

URLs	Alexa 1m
Total	988,399
Benign	913,550
Malicious	887
Suspicious	415
Failed	63,002

The results from Table 1 show that miner objects were successfully extracted from 887 URLs, and suspicious keywords were detected on a further 415 URLs. Due to the recent emergence of CryptoJacking and ease in which the scripts can be deactivated and removed, there is currently a lack of reliable datasets to formally verify the accuracy of results. Initially, to determine the false positive (FP) rate, we tested MANiC

against the top 10,000 domains on the OpenDNS list [45] which are considered to be benign. MANiC classified three of these URLs as malicious but manual analysis confirmed that these sites did, in fact, contain mining scripts.

Later, we manually analysed the 1302 URLs classified as malicious or suspicious. We determined that 99.21% of the malicious URLs were, in fact, malicious and had been correctly classified by MANiC. However, 14.7% of the suspicious URLs were malicious but were not properly classified by MANiC, primarily due to obfuscation. The remaining 85.3% were labelled as suspicious for various reasons; the script was loaded but not referenced, the mining code was commented out or, more commonly, the site simply mentioned a CryptoJacking related word in a non-malicious context, e.g. a news site or blog.

We recognise that the false negative (FN) rate is likely to be higher than these figures indicate because heavily obfuscated scripts wouldn't be detected as malicious or suspicious. Similarly, we appreciate that there may be unobserved exceptions which have not been identified due to the absence of a large, ground-truth dataset and the impracticality of manually analysing every URL in the Alexa top 1m.

TABLE II
DISTRIBUTION OF MINER FAMILIES

Family	MANiC	[5]
CoinHive	685	443
CryptoLoot	107	46
JSECoin	95	58
WebMinePool	16	N/A
DeepMiner	10	N/A
Papoto	2	2
Coinerra	1	N/A
ProjectPoi	1	1
MinerWorker	1	N/A
CryptoNoter	1	N/A
RandomSatoshiMiner	1	N/A
CoinImp	0	4
AFMiner	0	1
Minr	0	1

The results from Table 2 show the distribution of miner families found in the Alexa top 1m dataset using MANiC and the technique described in [5]. CoinHive remains the most popular with a 74% share in the market, and the remaining 26% is divided among alternative families. The results presented in [5] showed that CoinHive had a 92% market share so this could be an indication that the popularity of CoinHive is dropping in favour of alternative families.

The chart in Figure 2 shows the level of throttling that sites apply to their CoinHive scripts to limit the percentage of CPU resources utilised on the victim's machine, where 100% indicates that the site will utilise as much CPU power as possible. The most commonly applied throttling parameter was 70-80%, closely followed by 50-60%. This indicates that most sites are aiming to avoid detection and minimise the impact on user experience. The average configured throttling level detected was 56.49% although this does not include any sites which failed to specify a throttle value, e.g. CoinHive

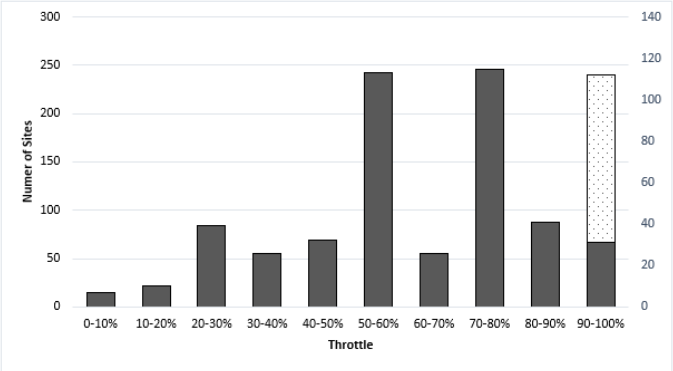

Fig. 2. Throttling level applied to CryptoJacking sites

utilises 100% of the CPU by default. The dotted bar shows the number of CoinHive sites that did not specify a throttle value. This increases the average to 70.26%, indicating that the impact of CryptoJacking on user experience is more severe than may be intended.

35.74% of the 887 malicious URLs were found to be using a site key that was detected on more than one URL. The most highly occurring site key was found on 65 different websites, indicating that the CryptoJacking script is being operated by the same person or group. It may also be an indicator of an attack, e.g. if an attacker compromises multiple websites and injects mining scripts, the profits will all go to the same wallet address.

C. Evaluation

There is an agreement between the results of the MANiC experiment on the Alexa top 1m websites and the results in [5], although MANiC had a higher detection rate and greater accuracy. MANiC successfully extracted 920 miners from the 887 URLs in the Alexa top 1m index and classified a further 415 URLs as suspicious, 61 of which were identified as malicious through manual analysis.

To accurately compare our results, we replicated the experiment from [5] using the same Google BigQuery commands and the Censys.io dataset that was recorded during the same period as the MANiC experiment. The results of this comparison are shown in Table 3. Using their method we detected 515 URLs suspected of hosting mining scripts. MANiC classified 430 of these URLs as malicious and 42 as suspicious. MANiC also identified 457 additional malicious URLs that were not detected using the technique described in [5].

Further analysis of these results showed why the results differed. The technique used by the authors of [5] simply checks for the presence of the string 'coinhive.min.js' anywhere in the HTML body. This is problematic for two reasons. Firstly, it creates FPs because some sites will load the script but not activate it, e.g. not referenced in code or commented out. Similarly, some sites will mention the string in a non-malicious context, e.g. news, blogs and tutorials. These scenarios would

lead to a malicious classification using the technique described in [5] but MANiC correctly labels them as suspicious.

Secondly, it creates FNs because some sites load mining scripts via proxies to bypass ad-blockers and blacklists. These mining sites would not be detected using the method in [5], but MANiC classifies them correctly when they use the same method to build and activate the mining code. We observed such cases repeatedly and validated that MANiC correctly classified the sites while the technique from [5] did not. Any other variations in the statistics between our experiment and [5] can be accounted for by the increased use of obfuscation and evasion techniques and a genuine fluctuation in the number of CryptoJacking sites.

To compare MANiC to a state-of-the-art dynamic analysis system, we installed and configured CMTracker [42]. Although the techniques used by this system are novel and well-suited to obfuscated crypto-mining scripts, side-by-side testing revealed that the CMTracker often failed to detect non-obfuscated mining scripts which were consistently identified by MANiC. We selected a random subset of 100 URLs from the Alexa top 1m which were detected by MANiC and verified that they contained mining scripts. The results of this comparison are shown in Table 3. When we tested these URLs with CMTracker's unaltered source code it was only able to correctly detect 44% of these sites as malicious.

TABLE III
EVALUATION

Paper	Dataset	Our Result	Their Results
[5]	Alexa 1m	887	515
[42]	Subset 100	100	44

The FN rate of MANiC is low because the system checks for various suspicious keywords. This means that if MANiC fails to detect and extract any mining scripts but finds a miner related keyword, the URL will be classified as suspicious. It is then subject to further manual analysis to verify the result and identify any flaws in the detection process. There are some exceptions to this rule, most notably the use of obfuscation, which often makes the miner undetectable and may prevent the detection of suspicious keywords.

The FP rate of MANiC is also low because to be classified as malicious, a miner object containing a sufficient level of data must be extracted. As the requirements are very specific, anything that falls outside of these restrictions will be classified as suspicious.

A limitation of the present system is its inability to process dynamic JavaScript. Initially, Splash was used in conjunction with Scrapy to ensure web pages are fully rendered before analysis. Unfortunately, this introduced a severe performance overhead and was subject to regular, unpredictable crashing which appears to be a common issue with the Splash framework.

Another potential limitation is MANiC's crawl depth. Currently, MANiC only analyses the homepage of each URL rather than every page on the site. This means that if a site only has mining code on specific pages or an injected iFrame, it may not be detected. Similarly, external JavaScript files are not analysed. These limitations can be overcome with relative ease, but the performance cost may outweigh the benefit in some cases. For example, most sites will apply crypto-mining scripts on each webpage to maximise profit. Increasing the crawl depth would severely decrease the speed of MANiC, with little improvement to the accuracy.

V. Conclusions and Future Work

This paper provided a look at a new emerging threat called CryptoJacking. We gave an overview of the history and implications of this threat and presented MANiC, a web-based crawler that uses regular expressions to detect crypto-mining scripts and extract a collection of parameters. When tested against the Alexa top 1m websites, our technique proved to be more effective than current research focused on CryptoJacking and was less prone to FPs and FNs.

The data we extracted from the crypto-mining websites serves three purposes. Firstly, it allows us to compile the first ground-truth dataset of crypto-mining URLs. We are currently in the process of developing a web app with MANiC running on the back-end, compiling a daily list of CryptoJacking websites. This service will be accessible by researchers who require access to our dataset for their experiments and will include functionality to test specific, user-supplied URLs for crypto-mining scripts.

Secondly, the data provides insights into the threat such as the distribution of miner families, the average throttling level applied and the proportion of websites that share the same wallet address. During our experiment, we observed how the use of ad-blockers and blacklists has impacted CryptoJacking, leading some sites to use proxies, obfuscation and other evasion techniques.

Finally, the collection of suspicious URLs allows us to manually analyse sites suspected of CryptoJacking and identify new techniques that are being employed to bypass detection. We can use this information to iteratively improve the accuracy of MANiC and identify new features that may assist ML research.

In future work, we plan to address some of the limitations described earlier and focus on detecting crypto-mining scripts that employ evasion techniques, particularly obfuscation. The detection and analysis of obfuscated JavaScript is an open research problem and can be approached from a static or dynamic perspective. We also plan to adapt MANiC to extract features from the website in addition to the crypto-mining scripts. Initially, the 58 features described in [43] will be taken into consideration.

The CPU-usage of websites alone is not enough to determine if a site is performing crypto-mining as throttling levels can be applied and sites may have genuinely high CPU-usage depending on the content. However, if we consider the CPU-usage alongside a range of other features typical of malicious websites, it could achieve high accuracy. Future work may focus on applying ML techniques to a combination of these

features and any new features that have been identified by MANiC.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[3] A. Laszka, B. Johnson, and J. Grossklags, "When bitcoin mining pools run dry," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 63–77.

[4] CoinMarketCap. (2018) Coinmarketcap. [Online]. Available: https://coinmarketcap.com

[5] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark, "A first look at browser-based cryptojacking," *arXiv preprint arXiv:1803.02887*, 2018.

[6] 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 *NDSS*. Citeseer, 2014.

[7] BBC. (2018) Russian nuclear scientists arrested for 'bitcoin mining plot'. [Online]. Available: http://www.bbc.co.uk/news/world-europe-43003740

[8] J. Segura. (2017) A look into the global drive-by cryptocurrency mining phenomenon. [Online]. Available: https://blog.malwarebytes.com/cybercrime/2017/11/a-look-into-the-global-drive-by-cryptocurrency-mining-phenomenon

[9] CoinHive. (2018) Monetize your business with your users' cpu power. [Online]. Available: https://coinhive.com

[10] I. Thompson. (2017) Stealth web crypto-cash miner coinhive back to the drawing board as blockers move in. [Online]. Available: https://www.theregister.co.uk/2017/10/19/malwarebytes_blocking_coin_hive_browser_cryptocurrency_miner_after_user_revolt

[11] AdGuard. (2017) The state of cryptojacking. [Online]. Available: https://crypto.adguard.com

[12] S. Liao. (2018) Cryptojacking rates increased by 85 times in q4 2017 as bitcoin prices spiked: report. [Online]. Available: https://www.theverge.com/2018/3/22/17147320/cryptojacking-8500-percentage-points-bitcoin-monero-spike-symantec-security-mining

[13] MalwareBytes. (2018) Malwarebytes reveals 2018 security predictions. [Online]. Available: https://press.malwarebytes.com/2017/11/20/malwarebytes-reveals-2018-security-predictions

[14] C. Cimpanu. (2018) Report: Three of top four malware threats are in-browser cryptocurrency miners. [Online]. Available: https://www.bleepingcomputer.com/news/security/report-three-of-top-four-malware-threats-are-in-browser-cryptocurrency-miners

[15] N. A. Gilad Yehudai. (2018) New research: Crypto-mining drives almost 90all remote code execution attacks. [Online]. Available: https://www.imperva.com/blog/2018/02/new-research-crypto-mining-drives-almost-90-remote-code-execution-attacks

[16] BBC. (2017) Bitcoin boom prompts growth of coin-mining malware. [Online]. Available: http://www.bbc.co.uk/news/technology-41693556

[17] L. O'Donnell. (2018) Cryptojacking attack found on los angeles times website. [Online]. Available: https://threatpost.com/cryptojacking-attack-found-on-los-angeles-times-website/130041

[18] L. H. Newman. (2018) Hack brief: Hackers enlisted tesla's public cloud to mine cryptocurrency. [Online]. Available: https://www.wired.com/story/cryptojacking-tesla-amazon-cloud

[19] K. McCarthy. (2017) Cbs's showtime caught mining crypto-coins in viewers' web browsers. [Online]. Available: https://www.theregister.co.uk/2017/09/25/showtime_hit_with_coinmining_script

[20] I. Thompson. (2017) Let's get ready to grumble! ufc secretly choke slams browsers with monero miners. [Online]. Available: https://www.theregister.co.uk/2017/11/07/ufc_coin_hive

[21] J. C. Chaoying Liu. (2018) Malvertising campaign abuses google's doubleclick to deliver cryptocurrency miners. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/malvertising-campaign-abuses-googles-doubleclick-to-deliver-cryptocurrency-miners

[22] C. Williams. (2018) Uk ico, uscourts.gov... thousands of websites hijacked by hidden crypto-mining code after popular plugin pwned. [Online]. Available: http://www.theregister.co.uk/2018/02/11/browsealoud_compromised_coinhive

[23] N. Lomas. (2018) Cryptojacking attack hits 4,000 websites, including uk's data watchdog. [Online]. Available: https://techcrunch.com/2018/02/12/ico-snafu

[24] C. Cimpanu. (2017) Cryptojacking script found in live help widget, impacts around 1,500 sites. [Online]. Available: https://www.bleepingcomputer.com/news/security/cryptojacking-script-found-in-live-help-widget-impacts-around-1-500-sites

[25] I. Thompson. (2018) Pulitzer-winning website politifact hacked to mine crypto-coins in browsers. [Online]. Available: https://www.theregister.co.uk/2017/10/13/politifact_mining_cryptocurrency

[26] FossBytes. (2018) Beware! this is the first firefox extension that injects crypto miner in your browser. [Online]. Available: https://fossbytes.com/image-previwer-firefox-extension-in-browser-crypto-miner

[27] L. H. Newman. (2018) Now cryptojacking threatens critical infrastructure, too. [Online]. Available: https://www.wired.com/story/cryptojacking-critical-infrastructure

[28] D. Goodin. (2018) Currency-mining android malware is so aggressive it can physically harm phones. [Online]. Available: https://arstechnica.com/information-technology/2017/12/currency-mining-android-malware-is-so-aggressive-it-can-physically-harm-phones

[29] FossBytes. (2018) 6 easy ways to block cryptocurrency mining in your web browser. [Online]. Available: https://fossbytes.com/block-cryptocurrency-mining-in-browser

[30] J. D. Parra Rodriguez and J. Posegga, "Csp & co. can save us from a rogue cross-origin storage browser network! but for how long?" in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. ACM, 2018, pp. 170–172.

[31] S. Helme. (2018) Protect your site from cryptojacking with csp + sri. [Online]. Available: https://scotthelme.co.uk/protect-site-from-cryptojacking-csp-sri

[32] M. AbdelKhalek and A. Shosha, "Jsdes: An automated de-obfuscation system for malicious javascript," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*. ACM, 2017, p. 80.

[33] B. N. (2018) Wikipedia page linked with 'minr' cryptojacking malware infected 3rd party website. [Online]. Available: https://gbhackers.com/wikipedia-page-linked-minr-malware

[34] C. Cimpanu. (2018) In-browser cryptojacking is getting harder to detect. [Online]. Available: https://scotthelme.co.uk/protect-site-from-cryptojacking-csp-sri

[35] J. Segura. (2017) Persistent drive-by cryptomining coming to a browser near you. [Online]. Available: https://blog.malwarebytes.com/cybercrime/2017/11/persistent-drive-by-cryptomining-coming-to-a-browser-near-you

[36] D. Maciejak. (2017) Cryptojacking: Digging for your own treasure. [Online]. Available: https://www.fortinet.com/blog/threat-research/cryptojacking-digging-for-your-own-treasure.html

[37] M. J. Zuckerman. (2018) The ethics of cryptojacking: Rampant malware or ad-free internet? [Online]. Available: https://cointelegraph.com/news/the-ethics-of-cryptojacking-rampant-malware-or-ad-free-internet

[38] 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.

[39] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao, "Seismic: Secure in-lined script monitors for interrupting cryptojacks."

[40] J. Rüth, T. Zimmermann, K. Wolsing, and O. Hohlfeld, "Digging into browser-based crypto mining," *arXiv preprint arXiv:1808.00811*, 2018.

[41] 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*. ACM, 2018, pp. 1714–1730.

[42] 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*. ACM, 2018, pp. 1701–1713.

[43] J. Vierthaler, R. Kruszelnicki, and J. Schütte, "Webeye-automated collection of malicious http traffic," *arXiv preprint arXiv:1802.06012*, 2018.

[44] Scrapy. (2018) Scrapy framework. [Online]. Available: https://scrapy.org

[45] jedisct1. (2018) Opendns top domain list. [Online]. Available: https://github.com/opendns/public-domain-lists/blob/master/opendns-top-domains.txt