2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34137

Web-based Cryptomining Detection

Dr. Vijay Anand, University of Missouri, St. Louis
Mr. Dmytro Kudriashov, EPAM Systems

Dmytro Kudriashov is a Software Engineer at EPAM Systems in Seattle, WA. He received his BS (2004) ans MS (2006) in Banking and Financial Support Services from the Kyiv National Economics University, and his MS (2019) in Applied Computer Science from the Southeast Missouri State University. Since 2018 his research efforts have focused on the interconnection areas of cybersecurity and digital financial instruments.

Web-Based Cryptomining Detection

Dima Kudriashov, EPAM Systems, Seattle, Washington, USA

Vijay Anand, Department of Information Systems and Technology, University of Missouri, St. Louis, Missouri, USA

A drastic surge on the cryptocurrency market of late 2017 and early 2018 lead to development and widespread implementation of web-based cryptomining. Initially providing a valid alternative to a regular advertisement-based forms of monetization, cryptomining quickly became a novelty form of malware by silently executing in the background without obtaining explicit consent from a user, an activity later became commonly known as drive-by mining or cryptojacking. To solve the issue of timely detection and prevention of cryptojacking, a number of in-browser solutions were developed, but neither of these achieved an absolute efficacy in eliminating targeted cybersecurity issues. This paper, provides a brief overview for this relatively newform of malware, analyzes technology used for browser cryptomining defined by two evolutionary phases of cryptojacking, and the financial reasoning behind this phenomenon.

By examining the steps and stages of web-based cryptomining and distinguishing potential detectable characteristics, this paper attempts to outline possible preventive anti-malware approach that can be developed as a counter-measure to this online threat.

Cryptocurrency is the latest cyber enabled commerce offering that students in Engineering should understand. Such new technology also brings new types of attacks which requires understanding of how existing technology is exploited by malicious actors and what type of remedies can be taken to prevent such attacks. This paper outlines the different aspects of cryptocurrency operation which is important to understand modern cyber commerce.

1 INTRODUCTION

With the introduction of the first cryptocurrency in 2009, Bitcoin [8] immediately became a niche instrument for anonymous currency transactions. The underpinnings of the cryptocurrency is based on the construct of a blockchain where blocks are added to an existing set of blocks by arriving at a consensus. A block in such a structure will be accepted as part of an existing chain by this network of users when a miner through trial and error discovers a unique number(nonce) which when included the aforementioned block yields a hash to meet the networks proof of work requirements. Once verified the block is added to the chain and the "first" miner to successfully complete this transaction is awarded some cryptocurrency. The block does not contain the user information who started this transaction providing a level of privacy and this feature is misused by cybercriminals. Another important aspect of this cryptocurrency was to be able to mine which required significant computation power due to the property of hashing functions. Mining thereafter became individual computer based

to cluster based to distributed network based. Thereafter many existing techniques were used to maximize mining profits (i.e. botnets), but only after the emerging of alternative coins, or altcoins, that made mining possible on a range of mainstream CPUs, the problem of silent mining manifested itself. One of the mechanisms that cybercriminals used to mine was using web service based mining where the mining computation activity was offloaded to any user who visits the infected site with or without the users permission and thus emerged the problem of cryptojacking emerged: using website visitors' computational power to solve hash functions thereby remotely mining without users' consent.

This problem quickly became the main cybersecurity trend in late 2017 - early 2018, as cryptojacking techniques matured and evolved. A number of conventional countermeasures were deployed to prevent web-based mining: major web browsers, browser extensions and adblockers tried to use domain blacklisting, which was easily defeated by URL randomization and domain generation algorithms. Another approach was to monitor CPU load, which immediately led to mining scripts developing a configurable parameter that allowed a pre-setting mining load; thus staying under the radar.

By examining the origin of memory-based cryptomining, following the evolution of web-based cryptomining services and detecting the main monetary aspects of cryptojacking, this paper will provide an in-depth review of the problem and try to outline possible future detection and evasion techniques and solutions.

The remainder of the paper is organized as follows: the review of web-based cryptomining in Section II will be followed by analytical review and comparison between cryptojacking and online advertising in Section III. In Section IV we will provide a case study of two major types of online cryptomining APIs, which will be followed by analysis of every possible step in web-based cryptomining as a potential candidate for definitive detection of script mining activity. In Section V. we note the observations made and conclusions and future work for this paper.

2 CRYPTOCURRENCY

2.1 Background

In the last decade cryptocurrencies have gained considerable popularity, initially as an alternative to government-emitted fiat currencies, and later as a playground for financial speculations, anonymous transactions and as an efficient money-laundering tool. Cryptography and blockchain technology used to produce building blocks of cryptocurrencies – a process called mining – to verify and add transaction records to a write-only database of all previous transactions. As an incentive to add a new block to the blockchain, the network compensates miners' efforts with cryptocurrency, and a newly added block protected by cryptographic techniques to ensure the integrity of the record. To add a block to the blockchain, miners have to solve a cryptographic puzzle, and a valid block will contain a solution to such puzzle with a hash of a previous block, hash of the transactions in the current block and an address for the miner's wallet on which the reward will be issued.

The cryptographic puzzle in the original Bitcoin was designed in such a way that it benefited from GPU and ASIC- based mining. Such implementation was not entirely scalable and constantly-rising complexity of computational puzzles required more investments both in electricity cost and hardware upgrades. As a remedy, the new cryptographic protocol CryptoNote and a corresponding proof-of-work function CryptoNight [11] were invented to allow effective mining on a mainstream CPUs. This hash function uses an extensive read and write operations in a 2 Megabyte region called scratchpad, effectively moving the focus of mining from computing resources to memory access performance. Since the majority

of mainstream CPUs are organized with a cascading multi-level system cache memory (L1, L2 and L3 is a common architecture), this allows CryptoNight and other memory-based algorithms to implement profitable cryptomining on ordinary desktop computers and mobile devices.

As one negative aspect of cryptomining was solved, it was time to overcome the second – cost of electricity, which by different estimates was negating up to 90 percent of the generated revenue. Since all of the main alternative coins share the same CryptoNote protocol, it was relatively easy to combine the previously used idea of mining pools (a distribution of hash computation and generated cryptocurrency reward among a number of miners) and novelty JavaScript delivery and execution systems. Three main technologies behind theemerging web-based cryptomining were WebSockets, WebWorkers and WebAssembly [3].

WebSocket protocol was designed as an application-layer full-duplex protocol and implemented into all the major web browsers as of December 2011 [21]. It allows faster clientserver communication with much less overhead compared to HTTP. WebWorkers is browser implementation which effectively allows multithreading in JavaScript. Although JavaScript has a native concurrency support, WebWorkers provide much easier scaling to a number of available CPU cores. And finally WebAssembly, or WASM (a low-level language standard from 2017) that implements a browser-hosted virtual machine and allows compilation of a high-level languages, such as RUST, thus dramatically improving JavaScript loading and execution time. WebAssembly was the final missing piece in the emerging cryptojacking threat, and as soon as all main web browsers implemented this novelty standard, the onset of a cryptojacking epidemic began. Figure 1 clearly demonstrates this spike by comparing prices of a CPU-bound and memory-bond cryptocurrencies and a proprietary VirusRadar threat level index from the IT security company ESET. Interestingly enough, we can clearly see that the initial spike in memory-based cryptocurrency Monero, that occurred around September 2017 could also be a driver that in just three month lead to an all-time high price of the Bitcoin at USD19,300.

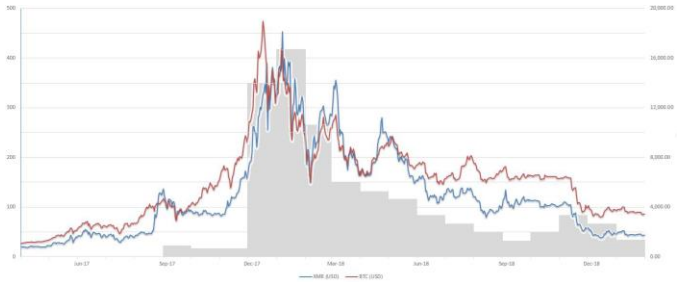

Figure 1. Correlation between price of Bitcoin (CPU-bound, red), Monero (memory-bound, blue) and ESET VirusRadar JS/CoinMiner Threat (grey) [9][20]

CoinHive was the first online product which emerged from a combination of the aforementioned tools. Its distribution consists of a single JavaScript file, which a website's owner links on the page along with a configuration for a public API provided by CoinHive. As soon as the webpage loads on a user's computer, the miner initializes an adequate amount of WebWorkers, creates a WebSocket connection to a mining pool (usually by using protocol called Stratum), fetches a job and begins calculations utilizing WebAssembly technology for increased performance. The ease of use and simple configuration initially led to a wide variety of JavaScript delivery methods ranging from direct webpage placement to exotic methods such as reported vulnerability exploit in MikroTik routers that allowed cryptojacking script injection into traffic passing through [14].

Albeit positioning itself as a valid substitute for an ad-free content, effective spam protection, CAPTCHA-alternative (which does not require any input from a user), in-game money and link forwarding, web-based cryptomining possesses all the possible negative aspects related to cryptojacking: notoriously, the absence of explicit user consent, unauthorized mining and security threats. Moreover, JavaScript mining supports code obfuscation both at initial connection establishing and inside the WASM module, which further depicts its usage as primarily targeted for cryptojacking by effectively denying keyword-based detection for antimining tools. On top of that, a WebSockets connection may also be encoded for further obfuscation. But the most impressive evasive technique is next: malicious code monitors whether in-browser debugging or developer tools are opened by user, and if they are –mining activity stops. From the user's point of view, unauthorized cryptomining manifests in increased electricity consumption, constant high load on the CPU coupled with possible overheating issues and thus the decreased component life expectancy. Users will experience elevated levels of noise in systems equipped with active air cooling, accompanied by notable drop in overall performance and web browser lagging. And since all the modern web browsers on mobile devices provide a full JavaScript support, the impact on non-stationary battery-equipped smartphones, tablets and laptops will be the most tangible: high CPU utilization coupled with increased network activity will drain the battery much faster.

3 FINANCIAL JUSTIFICATION

In this section we will review monetary aspects of cryptojacking and compare them to the well-known online advertisement revenue model. Defining the key differences in these two monetization strategies will lead to a better understanding of underlying concepts and in turn allow us to propose a different approach in developing counter-measures.

One of the main selling points of online advertisement platforms today is a targeted delivery, or a so-called tailored ads. The very fact of displaying an advertisement to an end-user is usually preceded by data mining and extensive research. Online activity of a said user is thoroughly collected and analyzed, later to be combined with any available physical information (such as GPS coordinates, IP geolocation and cellphone signal triangulation) to present an advertisable profile. Completeness of such profile directly relates to diversification of information sources, which is why all major players in the advertisement market constantly invent new products (e.g. Android Auto) to diversify data sources and supplement existing statistics with a new information. Later this profile will be analyzed by AI and transformed into a list of relevant categories. This list is matched with a list of categories provided by an advertisement platform's customers, thus defining a target audience for a marketed specific product.

Organizing this extensive process serves one goal: by presenting comprehensive user data, advertisement platforms compete among each other and business naturally goes to a company which is able to provide the most in-depth and complete audience data. But once a contract is obtained (i.e. monetary transaction completed) advertisers lose any if not all incentive to ensure future ad delivery. It is safe to assume that any consecutive actions on behalf of an advertiser will be on a best effort basis. That is why it is almost impossible to completely prevent data gathering by major players on the market; however blacklisting *doubleclick.net (and later *adpdx.com), for example, effectively removes almost all the advertisement from Google and its affiliates. Such a structure clearly defines three main parties of this process: advertisement platforms, advertising customers and advertisement subjects. In contrast, in cryptojacking we have only two parties – attacker and user (attacker is defined as a person responsible for placing and configuring malicious code on the webpage). At first, such an attacker will set possible targets for cryptojacking. The main criterion would be the average time spent by a user on the website

[6]. It is obvious that a login page for a web-mail interface is a bad choice, but a video-streaming service is a rather good one. After the target website is defined, the attacker will introduce malicious code to the webpage and start earning cryptocurrency.

But here is where the main difference between cryptojacking and advertisement lies: after successful injection, the attacker will continue to closely monitor and support initially injected code because only by ensuring code execution can she effectively ensure her revenue. Domain blacklisting can be easily avoided by content delivering networks such as Amazon's CloudFront, targeted script blocking may be counteracted by HTML inline scripts; pattern matching can be easily fixed by code obfuscation. Even monitoring CPU load as an indirect indication of mining activity can't be a reliable tool because all the modern cryptominers are equipped with adjustable throttling capabilities. The process quickly becomes a cat-and-mouse game at this point, but the attacker will always maintain an advantage because of the aforementioned direct financial interest.

A lot of research has been focused on predicting and calculating the possible level of revenue generated by cryptojacking [7]. As this technology matured and more data became available, it was possible for web-based mining services to provide related data themselves. For example, Figure 2 shows combined data from the CryptoLOOT website: the growth in revenue is linear and, for example, the average daily visitors count of 1 million unique users that will spend an hour on a website will generate 5.56 Monero, or about 500 US Dollars as of today's (2019) exchange rate. This number is relatively small compared to the other forms of malware (i.e. ransomware) and is positively small in comparison to possible advertisement revenue generated by the same website audience. But in this context, the cryptojacking should be viewed as a complementary source of revenue which will require almost no effort from the webpage owner and will generate anonymous and virtually untraceable gain.

	Time on web-page (min)				
users	*1*	*5*	*10*	*30*	*60*
1 000	0.00009273	0.00046363	0.00092726	0.00278179	0.00556358
10 000	0.00092726	0.00463632	0.00927264	0.02781792	0.05563584
100 000	0.00927264	0.0463632	0.0927264	0.2781792	0.5563584
1 000 000	0.0927264	0.463632	0.927264	2.781792	5.563584

Figure 2. Estimated profit for mostly desktop traffic (XMR, CryptoLOOT)

Estimates provided in Figure 2 are predominantly for desktop traffic; in order to get values for mostly mobile traffic and mixed traffic those figures have to be multiplied by 0.3 and 0.6 coefficients respectively, thus further reducing a possible profit.

Two-Phase Threat Model

Building on the introduction to web-based cryptomining in Section 2, we will perform an in-depth review of two major mining services in the wild – CoinHive and CryptoLOOT. Unfortunately, as of June 2019 CoinHive ceased its operations due to decreased profits and inability to implement necessary changes caused by the latest hard fork and algorithm updateby Monero [22]. This somehow limits accessibility and availability of documentation, but this section was written at the time when the mining service was still active and all the necessarydata was gathered and preserved.

Both mining services provide a similar approach in integration, which allows both authorized mining start (that is usually used by renowned websites which initiate mining activity only after obtaining an

explicit consent from a user and frequently by providing an ability for said user to manually regulate the mining CPU load [23]) and unauthorized mining initiation. In both cases the mining script being placed on a webpage by the site owner either for legitimate purposes (i.e. charity or as a substitute for a common online advertisement as a mean of traffic monetization) or as a silent way of receiving extra revenue. However, there isalso a possibility that the mining code was placed on a webpage without the knowledge of the site's owner through side-loading from a third-party, code injection or malicious attack. No matter the type of code implementation, both mining services support all the aforesaid scenarios.

A typical cryptojacking attack consists of the following stages: 1) Script from the webpage makes an HTTP request to the same webserver that hosts the webpage itself. 2) The webserver responds with a so-called orchestrator part of the mining: a script that contains all the detection and initialization tools. 3) Upon detection the user's environment, the orchestrator contacts the mining service server and receives cryptomining payload. 4) Payload is executed by establishing a connection with a mining pool, fetching work and submitting hashes.

This outline is general for all the mining services; the difference is in tools and techniques used to implement stages of this process, most notably steps 3 and 4. These dissimilarities allow us to clearly define two stages in the evolution of web-bases mining services: the first wave emerged around September 2017, and the second wave (gradually superseded the first one) on the verge of 2018-2019. Now we will look deeply into the two most common mining service types in order to show the evolution of web-based mining and elusiveness that accompanies cryptojacking. CoinHive was the most notable example in the first stage, pioneering the cryptojacking across a variety of different categories of websites. It was also the first commercial public mining service, so it is only logical to start with it.

A.1 CoinHive: CoinHive began operating in September 2017 and quickly gained widespread propagation; being first it defined the initial wave of cryptojacking services. Although its history dates back to 2007 and is rooted to an online game community, in its final form it emerged owning to a trio of German entrepreneurs with a history of conviction for fraud, spam anddrug-related offences [24].

In what later became a cryptojacking standard, mining activity is initiated by embedding a script file into the webpage: the script file was available both as a downloadable and linkable version and with or without minimization. Initially, there was no native code obfuscation, but later this option was added. Figure 3 provides an example of script file integration and launching of the mining code.

```
<script src = "https://coinhive.com/lib/coinhive.min.js"></script>
<script>
    var miner = new CoinHive.Anonymous(
                          'YOUR_SITE_PUBLIC_KEY',
                          {throttle: 0.9});
    miner.start();
</script>
```

Figure 3. Example of CoinHive integration

Upon contacting the server and obtaining an orchestration part across the HTTP connection, the orchestrator code detected a number of available cores in the CPU and the size of L3 cache. If the size is less than 2MB, execution stops; otherwise a respective number of WebWorkers is initiated. The HTTP connection to the mining is server updated via a WebSocket handshake request and connection to the mining pool's WebSocket proxy is established. After that, the WebAssembly module is initiated and a mining fetch-execute-response cycle will be performed for as long as the webpage stays open. In later

versions of public API the ability to manually select between native WASM and low-level JavaScript code was added, with WebAssembly still executed as a default option. CoinHive was pioneer among mining services and it fully benefited from all the aforementioned tools, which drastically increased web-based computational performance. It also enjoyed a lack of competition for a long period of time, so it had the liberty to define its service fee rate. And since the price of cryptocurrencies hit an all-time high around that time, the service fee of 30% did not cause any objection from the market, meaning for all the Monero that was mined by the embedded script, the owner of the script received 70% of the mined cryptocurrency.

As the market share of this mining service became significant and more anti-malware companies raised concerns and placed cryptojacking as a leading online threat, the question of obtaining consent from users arose. To address multiple complaints related to silent execution, CoinHive duplicated its services under the name of AuthedMine, designed to provide mining only upon receiving explicit consent from a user. Added overhead resulted in an increase to a 35% service fee, which further declined attractiveness of this mining service in particular and consented mining as such. Latest statistics show that the CoinHive to AuthedMine instance ratio balanced at 200:1.

A.2. CryptoLOOT: CryptoLOOT is a perfect example of mining services from the second wave of cryptojacking. Although it launched in October 2017, it only raised to be a valid threat in late 2018 before becoming the most dominant service on the market after CoinHive seized its operations.

Integration and orchestration steps for this mining service are universal and do not differ much from its predecessors. A JavaScript file must be embedded or side-loaded on the webpage, after page load it fetches the orchestrator code from a webserver, sets up the mining environment, contacts the mining server and starts the mining cycle. Figure 4 provides an example of an integrated script file.

```
<script src="//statdynamic.com/lib/crypta.js"></script>
<script>
    var miner = new CRLT.Anonymous('YOUR_SITE_PUBLIC_KEY');
    miner.start();
</script>
```

Figure 4. Example of CryptoLOOT integration

Figure 5 provides an example of code obfuscation that is applied to the orchestrator script. Not only is this being updated frequently to avoid detection by the anti-mining software's heuristic, but also the public API provides an option to enable auto-update of the orchestrator's code on the user's webserver to obtain the latest code with an updated obfuscation pattern.

Figure 5. Example of code obfuscation
(Source:https://github.com/Crypto-Loot/cryptoloot/blob/master/lib/crypta.js)

In contrast to the first-wave services, CryptoLOOT is using neither WebWorkers nor WebSockets. Since both technologies are rare across the other web applications, they become easy markers for mining activity. Although their presence does not guarantee definitive detection, in combination with other auxiliary factors (such as WASM activity and high CPU load) detection rates proved to be significant. To overcome this vulnerability, all mining traffic is sent via regular encrypted TCP protocol; and instead of relying on the in-browser WebWorkers multithreading, now mining service provide a setting through public API that allows manual selection for a number of cores/threads (recommended setting is 8).

As a second-wave service, CryptoLOOT operated on a well-established market and faced multiple competitors. This manifested in a moderate 12% service fee and increased frequency in revenue transfer; now the payments were made to the owners' wallets every two hours regardless of transaction amount. Any type of consent-driven mining is absent from public API and its implementation is left entirely to webpage administrators.

Based on this overview, it is possible to find the following characteristics among cryptojacking services:

- General structure and component modality is identical between mining services of the first and the second wave.
- In order to minimize dependency on a hashing algorithm, mining services from the second wave step away from a mono-currency affiliation and offer flexibility for mining among different cryptocurrencies.
- While earlier services benefited from aggregation of the tools that ensured optimal performance, thus maximizing profit, later mining services moved toward more general but less detectable technologies, albeit substantial profit loss.
- Operational fees decreased dramatically, which means that both initial development and setup expenses were fully recovered and now mining services agree to operate on smaller margins.

4 PROPOSED APPROACH TO DETECTION AND PREVENTION

In Section 2 we reviewed WebSockets, WebWorkers and WebAssembly as a tools that made possible effective script-based web browser cryptomining. But solely relying on identifying these functionalities as a prerequisite for successive cryptojacking detection is not enough because in anticipation of such a straightforward approach many evasive techniques have been developed. In order to successfully defeat and, more importantly, prevent any future cryptojacking variations, it is imperative to correctly detect execution of the CryptoNight hashing algorithm. Unfortunately, such detection requires direct access to the CPU's L1 and L3 caches and elevated user privileges, which makes browser extension form implementation problematic. Therefore, we will have to dissect the web-based cryptomining process into definitive steps and stages in order to explore the possibility of linking them with absolute certainty to mining activity. Of course, the brute approach would be a complete ban on script execution, but the reality of the modern Model-View-Controller/Binder web architecture dictates that scripts are an integral part of web pages. Earlier, every web browser contained a native option of disabling script execution, but already for many years such an option was removed as one that will prevent correct webpage rendering.

The initial idea behind this paper was to develop a browser extension for one of the major web browsers (in this case, Mozilla Firefox) which would require minimum attention from a user, will run silently in the background, detect cryptomining scripts on the fly and halt their execution. This later part could be easily achieved by granting such extension access to privileged parts of the extensions API (such as activeTab, unlimitedStorage, webRequest and webRequestBlocking); this way any script file in DOM can be stopped with a simple .stop() command. This shifts the focus of attention entirely to the detection part of the process.

Stages Of Detection

Section III.A provides an overview of a typical cryptojacking attack and defines key stages of this process. Consecutively, we will have to extract a clear marker for mining script, assert the validity of this marker in order to attain the minimal false-positive and false-negative results and determine the possibility of implementing such an approach as one that could or could not be implemented using only the built-in native browser extensions API without resorting to developing a standalone application.

A.1 Domain filtering: Domain filtering is the most common technique to this day when it comes to blocking unwanted content or denying unwanted host communication. It comes in various formats and is widely used in advertisement-blocking software. As a web browser extension, it proved to be powerful and useful, especially in combination with an element-hiding capability which allows selection of any element on the web page by means of addressing it using vanilla CSS syntax.

As we already showed in Section III.A, this approach is somewhat flawed based on the different times monetary incentive was created. While common online advertisers will rarely implement anything to ensure the ad would actually be seen, the cryptojacker will be willing to go the extra mile in order to guarantee script loading and continuous, uninterrupted execution.

For example, web infrastructure companies such as Cloudflare could provide a reverse proxy service or even register a new randomized web address to provide the malicious script delivering URL that is still not on a blacklist.

Even more, the domain filtering could be easily defeated by loading the script file from the same domain as the website itself, a situation common when the website administration is the one benefiting from the cryptomining. Orchestration script is small enough to be loaded or even embedded inline, thus leaving the user with a dilemma: to disable all the scripts for this domain, eliminating cryptomining

threat, but most certainly rendering the useful content crippled; or to accept both evident and hidden qualities of the website and continue using it as is, caveat emptor.

A.2 Pattern matching: A pattern matching approach somewhat echoes with the previous detection method, but extends it a bit further: in this case not only is the origin of the script file checked against a blacklist of domains in the form of a list of strings, but the name of a script file alongside with its content is analyzed and if a particular match is detected, the script execution is stopped.

Although this routine may sound promising and could be easily implemented by extensions API, there are in reality multiple tools that could be used to avoid or defeat such detection. File name randomization takes care of filtering by script name and code minification and obfuscation deters effective script code scanning.

A.3 DOM Monitoring: Regardless of its exact implementation, the orchestration code of the cryptomining script will possess a number of unique features that are governed by JavaScript programming language. Irrespective of class or variable names that can be obfuscated to the point of complete incognizance, they would still hold characteristics such as type, prototype property of a constructor in its prototype chain or a specific type for return functions. This approach benefits from ease of implementation and immunity from code obfuscation, but it also lacks in absolute efficiency. Script code could be easily re-written in a less elegant but entirely different way, not to mention that any major changes in the underlying hashing algorithm inevitably will lead to structural deviations rendering this detection obsolete.

Several popular web browser extensions utilize this approach either on their own or in combination with other detection methods[2].

A.4 CPU load observation: The very essence of the web-based cryptomining process is a hashing operation with an extensive number of cryptographic and read and write operations, which in result manifests in high CPU load. The nature of this load is relatively stable and consistent, without ever presenting major spikes or periods of low activity. Although it is theoretically possible for the cryptomining script to set the mining load at a relatively moderate CPU utilization rate in attempt to blend with other legitimate CPU operations and avoid raising suspicion, in reality this arrangement will produce such a negligible revenue that this approach was never observed. On the other hand, there are quite a few situations in which CPU load could express similar behavior: online games with elaborate graphics, high-definition streaming services or even a poorly coded legitimate scripts.

Albeit possessing all the necessary characteristics of the perfect detection marker, CPU load monitoring cannot be considered for this role for a number of reasons: the high probability of the false-positive results, complicated if not impossible implementation in a browser extension form and non-linear correlation between selected desirable processor utilization rate and real output. Figure 6 represents the data ratio between nominal mining load setting and actual measured CPU load for a selected mainstream laptop CPUs.

	actual		
load	Intel® Celeron® N3450	Intel® Core™ i3-5010U	Intel® Core™ i7-8565U
20%	>99%	>97%	>99%
30%	>99%	>99%	>99%
40%	>99%	>99%	>99%
50%	>99%	>99%	>99%
60%	>99%	>99%	>99%
70%	>99%	>99%	>99%
80%	>99%	>99%	>99%

Figure 6: Nominal vs. actual CPU load for CryptoLOOT implementation of CryptoNight hashing algorithm

A.5 Traffic analysis: Script mining initially generated network traffic with a unique footprint, utilizing WebSocket protocol to speed up communication between infected host and a mining pool.

Despite the obvious advantages, said protocol did not gain widespread circulation and remained in limited use throughout the web. That is why it was abolished in favor of a more generic TCP, thus effectively avoiding possible distinction in the second wave of cryptomining services. Furthermore, initially only the orchestration part of the mining process was obfuscated and the actual hash load were sent without any cryptographic changes. Later, this was changed and nowadays all the traffic between a host and a mining pool is encrypted: this adds a considerable overhead and a tangible loss of mining efficiency, but completely eliminates any possibility of traffic analysis and pattern-matching detection. In addition, there are no direct correlations between mining load and generated traffic in terms of packet count (not that such analysis could be implemented with browser extensions API). Traffic intensity varies greatly; short bursts of data exchange spikes alternates with dormant periods characterized by small number of packets or no packets at all.

In conclusion, we can state that a flexible and adaptive cryptojacking script ecosystem in combination with limited functionality of the native extensions API denies us a possibility of creating a 100%-effective, user-friendly and error-free in-browser preventive solution. The efficiency of the existing extensions derives in a form of combined effectiveness of domain filtering, pattern-matching and DOM monitoring approaches. The widely used extension MinerBlock is a perfect example: it superposes a domain blacklist and script name pattern matching blocking and it monitors DOM for instances with specific variable properties and names [2]. Apparently, this approach is deemed effective enough, even though the number of false-positive results should not be ignored as a statistical error.

5 CONCLUSIONS AND FUTURE WORK

Albeit the recent cryptocurrency market has declined and interest in the problem of unauthorized mining has increased from both the government agencies and software companies developing web browsers [4][13], cryptojacking still remains a hazardous online threat. Moreover, the initial spike in cryptojacking occurred in September 2017 at the price levels which were almost two times as low as the price on the time this paper being written, meaning that today the monetary incentive for cryptojacking development is even higher. And all this is aggravated by the fact that cryptojacking has already passed the infant stage and matured into a fully functioning infrastructure, thus almost entirely eliminating previously necessary costs associated with initial code development. We reviewed the mechanism of

cryptomining scripts in their evolution, while coming to the conclusion that they possess quite a few praiseworthy features: a modular design allows almost seamless interchange between network protocols and execution environments, comprehensive utilization of latest web browser technologies guarantees maximum efficiency and creative approach ensures detection avoidance. This unique adaptability and robustness determines almost absolute elusiveness and invulnerability, rendering regular in-browser solutions useless and leaving users exposed.

Although abusing visitors' resources for cryptomining is a relatively new trend, it has already attained extensive work on its origin and underlying mechanism [18]. Unfortunately, most of the research is done on quantification of different aspects of cryptojacking such as website propagation and possible profitability in relation to more conventional revenue generating activities [5]. Some of most promising proposed preventive measures include monitoring CPU cache events by using performance counters to detect the execution of cryptographic applications [18]. As we hypothesized, although this approach is promising in terms of the outcome, its actual implementation may lack the necessary simplicity to be effectively implemented in form of a lightweight browser add-on. This is indirectly confirmed by one of the market leaders on the web browsers market which opted for a simple domain blacklisting approach [25].

Data presented in this paper reviews the major components and stages of cryptojacking, thus providing the outline for a number of potential vectors of future research. We foresee the possible implementation as of one not being focused on a single part of the web-based cryptomining process, which at the same time will maintain the overall memory and CPU footprint at a minimum in order to achieve usability and resource preservation.

REFERENCES

[1] CoinHive Documentation. Accessed: Mar. 14, 2019. [Online]. Available: https://coinhive.com/documentation/miner

[2] MinerBlock. Accessed: Oct. 21, 2019. [Online]. Available: https://github.com/xd4rker/MinerBlock

[3] WABT: The WebAssembly Binary Toolkit. Accessed: Oct. 21, 2019. [Online]. Available: https://github.com/WebAssembly/wabt

[4] FTC. Equiliv Investments (Prized). Accessed: Oct. 21, 2019. [Online]. Available: https://www.ftc.gov/enforcement/cases-proceedings/142-3144/equiliv-investmentsprized

[5] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Web-based Cryptojacking in the Wild," Aug. 2018, [Online]. Available: https://arXiv:1808.09474v1

[6] Cybersecurity Trends 2019: Privacy and Intrusion in the Global Village. Accessed: Oct. 21, 2019. [Online]. Available: https://www.eset.com/ca/trends-2019/

[7] 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 CCS '18: 2018 ACM SIGSAC Conference on Computer & Communications Security, Toronto, ON, Canada, Oct. 2018, doi: 1145/3243734.3243858

[8] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Accessed: Oct. 21, 2019. [Online]. Available: https://www.bitcoin.org/bitcoin.pdf

[9] [9] ESET Virus Radar Threat Encyclopaedia. JS/CoinMiner. Accessed: Oct. 21, 2019. [Online]. Available: https://www.virusradar.com/en/JS_CoinMiner/chart/history

[10] J. Segura. Malicious cryptomining and the blacklist conundrum. Accessed: Oct. 21, 2019. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2018/03/maliciouscryptomining-and-the-blacklist-conundrum

[11] Seigen, M. Jameson, T. Nieminen, Neocortex, and A. M. Juarez. CryptoNight Hash Function. Accessed: Oct. 21, 2019. [Online]. Available: https://cryptonote.org/cns/cns008.txt

[12] CryptoLOOT Documentation JavaScript Miner. Accessed: Oct. 21, 2019. [Online]. Available: https://cryptolootminer.com/dashboard/documentation.php

[13] L. Wagner, "Turbocharging the Web," IEEE Spectrum, vol. 54, no. 12, pp. 48–53, Dec.2017.

[14] C. Cimpanu. Massive Coinhive Cryptojacking Campaign Touches Over 200,000 MikroTik Routers. Accessed: Oct. 21, 2019. [Online]. Available: https://www.bleepingcomputer.com/news/security/massive-coinhive-cryptojackingcampaign-touches-over-200-000-mikrotik-routers/

[15] A. Viquez. Opera introduces bitcoin mining protection in all mobile browsers – here's how we did it. Accessed: Oct. 21, 2019. [Online]. Available: https://blogs.opera.com/mobile/2018/01/opera-introduces-bitcoin-mining-protectionmobile-browsers

[16] A. Rossberg. WebAssembly core specification. W3C First Public Working Draft. Accessed: Oct. 21, 2019. [Online]. Available: https://www.w3.org/TR/2018/WD-wasm-core-1- 20180215

[17] N. van Saberhagen. Cryptonote v2.0. Technical report, CryptoNote. Accessed: Oct. 21, 2019. [Online]. Available: https://cryptonote.org/whitepaper.pdf

[18] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark. A First Look at Browser-based Cryptojacking. In Proc. of the IEEE Privacy and Security on the Blockchain Workshop, London, UK, Apr. 2018, [Online]. Available: https://arXiv:1803.02887

[19] T. Zhang, Y. Zhang, and Ruby B. Lee, "CloudRadar: A Real-Time Side-Channel Attack Detection System in Clouds," In Proc. of the International Symposium on Recent Advances in Intrusion Detection, Evry, France, Sep. 2016, doi: 10.1007/978-3-319-45719-2_6

[20] Bitcoincharts. Accessed: Oct. 21, 2019. [Online]. Available: https://bitcoincharts.com/charts

[21] I. Fette, A. Melnikov. The WebSocket Protocol. Accessed: Oct. 21, 2019. [Online]. Available: https://tools.ietf.org/html/rfc6455J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[22] P. Muncaster. Coinhive Monero Miner Set to Close. Accessed: Oct. 21, 2019. [Online]. Available: https://www.infosecurity-magazine.com/news/coinhive-monero-miner-set-toclose-1

[23] UNICEF Australia. The Hope page. Accessed: Oct. 21, 2019. [Online]. Available: https://www.thehopepage.org

[24] B. Krebs. Who and What Is Coinhive? Accessed: Oct. 21, 2019. [Online]. Available: https://krebsonsecurity.com/2018/03/who-and-what-is-coinhive

[25] Let Firefox help you block cryptominers from your computer. Accessed: Oct. 21, 2019. [Online]. Available: https://blog.mozilla.org/firefox/block-cryptominers-with-firefox

[26] CryptoNote's. Accessed: Oct. 21, 2019. [Online]. Available: https://cryptonote.org