Are You a Favorite Target For Cryptojacking? A Case-Control Study On The Cryptojacking Ecosystem

Giorgio Di Tizio*, Chan Nam Ngo*
*University of Trento, Trento, IT
giorgio.ditizio@unitn.it, channam.ngo@unitn.it

***Abstract*—Illicitly hijacking visitors' computational resources for mining cryptocurrency via compromised websites is a consolidated activity.**

Previous works mainly focused on large-scale analysis of the cryptojacking ecosystem, technical means to detect browser-based mining as well as economic incentives of cryptojacking. So far, no one has studied if certain technical characteristics of a website can increase (decrease) the likelihood of being compromised for cryptojacking campaigns.

In this paper, we propose to address this unanswered question by conducting a case-control study with cryptojacking websites obtained crawling the web using Minesweeper. Our preliminary analysis shows some association for certain website characteristics, however, the results obtained are not statistically significant. Thus, more data must be collected and further analysis must be conducted to obtain a better insight into the impact of these relations.

1. Introduction

The year of 2008 witnessed the advent of Bitcoin [21], the most successful decentralized cryptocurrency in the history of digital/crypto-cash. Bitcoin utilizes "Proof-of-Work" (PoW) [14], a hard cryptographic computational puzzle, as a means of mitigating Sybil attacks [9] and eventually determining the inherent value for the medium of exchange. The PoW puzzle is hard to solve but easy to check. This process is referred to as "mining", and the individual nodes are commonly referred to as "miners". Once they successfully solve a block puzzle, they will receive a reward and all transaction fees included in the block.

Following Bitcoin, many cryptocurrencies have been developed and deployed widely, e.g. LiteCoin, Ethereum, ZeroCoin, Zcash, and Monero. Since then, cryptocurrencies have also been actively traded in exchanges, for example, the Chicago Board Options Exchange (CBOE) and Chicago Mercantile Exchange (CME) launched Bitcoin futures markets.

The *high trading price* and the *possibility to make anonymous payment* of cryptocurrencies in these exchanges make mining very *attractive to malicious actors*. While Bitcoin only provides pseudonymity, some other cryptocurrencies (e.g. Monero) offer very strong payment privacy, which make them *ideal for illicit mining*; a practice where an attacker hijacks the computational resources of the victim to mine the cryptocurrency for himself.

Among the hijacking methods, cryptojacking, i.e. malicious browser-based crypto-mining programs that start the background mining process with some scripts (e.g. JavaScript) [10], is the most popular method, especially when the CoinHive browser miner was developed in 2017. The attack vectors in cryptojacking could be (i) the malicious website owner himself (the website simply mines without consent from visitors) [10] (ii) a benign website compromised by an attacker (for example the websites of Indian Government [7] and CBS showtime [18]) (iii) some third-party plugins for popular Content Management Systems (CMS), e.g. cryptojackers struck Drupal [20] and Wordpress [17], (iv) advertisements injected via cryptojacking scripts [19], or (v) a man-in-the-middle (MITM) attack, for example, via compromised routers [5].

Different methods have been developed to detect cryptojacking: searching for known strings in the source code (such as *coinhive.min.js* or *load.jsecoin.com*) [10]; analyzing executed JavaScript code and WebSocket traffic frames for obfuscated JavaScript [25]; employing computational resources API-based detection method [26]; performing JavaScript code block analysis on the compiled JavaScript code [16]; observing the call stack and seeking for periodic executions [12]; or identifying mining scripts based on the CPUs L1 and L3 cache usage and cryptomining characteristics in WebAssembly [15].

A general observation is that, to be profitable, web-based mining requires (i) a high number of visitors, (ii) a reasonably long duration of the visit, and (iii) a high computational power from each visitor [12], [15]. As the last two requirements cannot be easily controlled by criminals, they are more interested in obtaining *a high number of visitors* by compromising either (a) a well-known, and highly likely secure, website or (b) a high number of less popular, but at the same time, potentially less secure, websites. In the latter case, websites compromised for cryptojacking are often part of an untargeted campaign. Thus, to *maximize the profit* and *minimize the effort*, attackers must try to find a *set of characteristics* that can be exploited to compromise the *highest number of websites*. The common approach is to exploit known vulnerabilities in web applications, for example, exploiting CMS [20] and their plugins[1]. In this paper, we investigate if some characteristics of a website are positive (negative) risk factors of being compromised for cryptojacking campaigns.

Paper Organization. In (§2) and (§3) we describe the related works, we identify the current gap in the academic literature on cryptojacking, and we present the hypotheses

1. https://www.crowdstrike.com/resources/reports/2020-crowdstrike-global-threat-report/

that drive our analysis. Then, we present the data collected and the preliminary results (§4). Finally, we discuss the limitations and future work (§5) and the conclusions (§6).

2. The Previous Looks Into Cryptojacking

In Table 1 we summarize the notable research questions and results in cryptojacking.

The first look into cryptojacking was done by Eskandari et al. [10]. The authors looked for suspicious strings (such as *coinhive.min.js* and *load.jsecoin.com*) in the website's JavaScript code to identify illicit crypto-mining activities. Wang et al. [32] identified Wasm, Domain Whitelisting, Opt-In and CPU Throttle as key components of cryptojacking and employed semantic signature matching for crypto-mining detection. They found that mining and non-mining computations can be distinguished with 98% accuracy. Konoth et al. [15] looked more into the ecosystem of cryptojacking. The authors identified 0.18% of the top 1M Alexa as cryptomining websites and further associated them with 28 mining services. They also clustered the mining pages into 20 campaigns based on site Keys and proxy information. Similarly, Bijmans et al. [4] identified 10k websites that performed illicit mining. They further clustered them into 204 campaigns. Yet, 85% of the mining websites stop mining after one year. Papadopoulos et al. [22] compared the cost and profit of cryptomining with traditional advertisements. They concluded that cryptominers consume significantly more bandwidth, CPU, memory, battery yield higher system temperature than ads. Yet, ads are still more profitable: 3 ads impressions give 5.5x more profit than 1 cryptominer. Tahir et al. [27] showed how to use Hardware Performance Counters to detect (with more than 99% accuracy) cryptojacking websites even when obfuscation is employed. Bijmans et al. [5] investigated MITM attacks exploited for cryptojacking and concluded that such an attack is 30x lucrative than classic cryptojacking. Hong et al. [12] observed hash function calls and repetition in stack execution to detect cryptominers. The authors identified 2770 websites that exhibited mining behaviors. Rodriguez and Posegga [26] suggested monitoring system resources (e.g. CPU, GPU) and their corresponding APIs to detect cryptojacking with 97.84% recall and 99.7% precision.

3. What To Look Into Next?

In Table 2 we classify the academic literature on cryptojacking into three major categories: *fact finding*, *technical analysis*, and *economic analysis*. The majority of the state-of-the-art is mainly focused on large scale analysis of the cryptojacking ecosystem, the techniques implemented in the web-based mining, and the development of detection mechanisms.

A question so far unanswered is if there are technical characteristics[2] of a website that are *likely to be of interest for a cryptojacking attacker* and will *influence the risk of being compromised*. We thus want to determine associations between certain website characteristics and the probability of being compromised. This could be caused by several factors like the market share, the number of vulnerabilities of a product, and the presence of exploits in public and closed forums. However, our goal is not to determine causality as case-control studies are limited in this application [28]. We provide a *preliminary analysis of possible technical factors* that can *positively or negatively impact the odds of being compromised* for cryptojacking campaigns. The goal of the analysis is to help web site administrators to determine if the benefits produced by a certain technology worth the risk of being compromised for criminal activities.

Cyber risk estimation is a well-studied topic in security research. Several approaches estimate cyber risks using machine learning on data of attacks and exploits [6], [13], [34]; regression on big data [2], [3]; and case-control studies [1]. Among these approaches, the latter was utilized to analyze the attack traces in the wild and determine: which people are more likely to be a victim of targeted attacks [29]; which web server characteristics associated with a higher rate of being compromised for phishing and search-redirection attacks [31]; which behaviors can be positively correlated with the probability of being infected by malware [33], and the impact of web security features and patching practices on web compromise rates [28].[3] Our paper follows the approach of [31] but considers a completely different threat.

We present a preliminary study of the possible technical website characteristics that can increase (decrease) the odds to be compromised for web-based mining.

3.1. Experimental Design

We want to determine if certain characteristics of a website bring about a higher risk of being compromised for cryptojacking campaigns. We, thus, define the following hypotheses that guide our analysis:

- **H1:** Using certain web server technologies can increase the risk of being compromised for cryptojacking campaigns;
- **H2:** Using certain CMSs can increase the risk of being compromised for cryptojacking campaigns;
- **H3:** *Visible* characteristics associated with hardening can reduce the risk of being compromised for cryptojacking campaigns;

With **H1** and **H2** we want to determine if *certain types of server and CMS are the major targets* for cryptojacking campaigns. With **H3** we want to determine if some *basic practices*, for example version hiding, *can influence the probability of being targeted* by these campaigns. Assuming that cryptojacking campaigns are not performed by Advanced Persistent Threats (APTs),[4] we expect attackers to prefer "low-hanging fruits" matching certain criteria.

To measure the impact of these factors in the risk of being compromised we perform a case-control study as it is commonly done in other fields like biology and medicine [8].

2. As opposite to classification based on the content of a website.

3. Case-control studies have some limitations as the causal relation between attack phases and the attack measurement can be only approximated [24]. However, they are a good solution when it is not possible to run an experiment [1].

4. APTs are characterized by targeted campaigns where, in most of the case, the primary goal is espionage or sabotage. However, some APTs target the cryptocurrency sector.

TABLE 1: Notable Cryptojacking Research So Far

Paper	Research Questions	Results
Eskandari et al. [10]	1. What is the current status of cryptojacking in the Monero ecosystem? 2. What are the (rough) possible mitigations?	1. Over 33K websites were found with illicit-mining; cryptojacking scripts impact about 25% of user's CPU, obfuscation is used to avoid detection, but operators are blacklisting domains associated with cryptomining. 2. Using consent, blacklist or detection of excessive resources.
Wang et al. [32]	1. What are the key components of cryptojacking? 2. Can we detect cryptojacking using semantic signature matching?	1. Key components include Wasm, Domain Whitelisting, Opt-In, and CPU Throttle. 2. Mining and non-mining computations exhibit significantly different behavioral patterns thus, it possible to achieve a 98% accuracy.
Konoth et al. [15]	1. How prevalent is drive-by mining in the wild? 2. How many different drive-by mining services exist currently? 3. Which evasion tactics do drive-by mining services employ? 4. What is the modus operandi of different types of campaigns? 5. How much profit do these campaigns make? 6. Can we find common characteristics across different drive-by mining services that we can use for their detection?	1,2. 1735 websites (out of 1M Alexa) identified as mining cryptocurrency with 28 different mining services. 3. Evasion techniques include code obfuscation, obfuscation of the stratum communication, and anti-debugging checks. 4. 20 campaigns identified based on site Key and proxy information; that are classified into 3 groups: miners injected by 3rd party, miners injected through ads, and compromised websites. 5. On average each of these websites (is estimated to) earn roughly 110$ per month, but it can also be worst with roughly 900 websites that earn less than 10$. 6. All the websites use Wasm for the payload and WebSocket. Roughly 43% of the websites mines only when an internal page is visited.
Bijmans et al. [4]	1. Can we cluster the malicious cryptojacking websites by campaigns? 2. How do cryptojacking activities evolve?	1. The authors observed 10k websites that mine without consent and categorized the cryptomining applications (Coinhive, Cryptoloot, etc.). They clustered campaigns based on site Key and WebSocket proxy and identified 204 cryptojacking campaigns covering 5k websites. 2. 85% of the websites are not mining anymore after one year.
Papadopoulos et al. [22]	1. What is the actual cost of web-cryptomining on the user side? 2. What is the profitability for the attacker or the benign publisher? 3. Can it become an alternative web monetization scheme for benign publishers?	1. Mining utilizes up to 3.4x more bandwidth, 59x more of the visitors CPU, 1.7x more space in real memory, 52.8% higher temperatures, and 2.08x more energy than advertisements. 2. Websites generate more than 5.5x higher revenues by including 3 ad impressions than by including a cryptominer. 3. For cryptominer profit, the users browser tab must remain open for a duration longer than 5.53 minutes.
Tahir et al. [27]	1. What are the (mining and evasion) techniques employed by the attackers in the wild? 2. How can we detect web-based mining using hardware-assisted profiling?	1. Plain mining script; dynamic/platform-aware mining (using logical processor information, battery status or device type), Base64 Obfuscation, and npm Javascript Obfuscator. 2. By using Hardware Performance Counters, one can catch mining applications even when obfuscated (99% accuracy).
Bijmans et al. [5]	1. How can attackers exploit MITM to perform illicit mining? 2. Which are the techniques, tactics, and procedures employed by the adversary? 3. What is the revenue compared to "classic" cryptojacking?	1. Adversaries compromised MikroTik routers to deploy a malicious HTML page to each ongoing connection. 2. Reconnaissance of routers via scanning and Shodan records, exploitation of CVE-2018-14847, routers are often re-compromised by new adversaries. 3. MitM mining is 30x lucrative than "classic" cryptojacking. The life cycle of these infections is longer compared to website infection.
Hong et al. [12]	1. Can we detect cryptojacking scripts using hash functions calls and repetition in the execution stack? 2. What is the profit for the attacker and the cost for the users? 3. What are the techniques employed by cyptojacking pages?	1. Based on hash function calls and stack execution, the authors identified 2770 cryptojacking website from roughly 850k pages from the 100k Alexa domains, among which 868 in the top 100k Alexa domains. 2. The malicious mining pages consume at least 278K kWh electricity energy per day. Malicious miners can gain more than 1.7 million US Dollars, from more than 10 million users per month. 3. Blacklist (NoCoin and MinerBlock) are insufficient: less than 51% of malicious pages detected. Common techniques to avoid detection rely on CPU limiting, code obfuscation and payload hiding (e.g. in another library)
Rodriguez, Posegga [26]	1. How can we detect cryptojacking scripts using the system's resource consumption and API usage information?	1. By monitoring CPU, GPU, Storage, Networking and Inter-Window, (and their corresponding APIs), the classifier achieves 97.84% recall and 99.7% precision, and it is not influenced by obfuscation techniques.

4. Preliminary Results

We now present the procedure employed for the data collection and the preliminary results obtained.

4.1. Data Collection

Our primary goal is not to identity new cryptojacking websites but instead determine their technical characteristics. We thus collected the list of websites observed to perform cryptojacking without consent from [12] and [15]. In addition we extracted from publicWWW[5] the list of websites that contains in the source code a keyword or javascript listed in [4], [30], and in the NoCoin[6] and MinerBlock[7] blacklists. However, cryptojacking websites seem to last for a short period of time before being moved to other websites [12] and the presence of mining scripts does not implicate that a cryptojacking campaign is carried on (e.g. inactive pools or discontinued scripts [30]). Thus, we validate the list of domains with a state-of-the-art crawler for the detection of miners from [15]. Tab. 3 summarizes the active cryptojacking websites. From the active cryptojacking websites, we extracted the minimum

5. https://publicwww.com/

6. https://github.com/keraf/NoCoin

7. https://github.com/xd4rker/MinerBlock

TABLE 2: Research Topics addressed by the State of the Art

Category	Research Topics	Papers
Fact Finding	Analysis of the cryptojacking ecosystem	[10], [15], [4], [5], [12]
Technical analysis	Development of a detection mechanism	[32], [15], [27], [12], [26]
	Analysis of modus operandi of web based mining	[15], [4], [27], [5], [12]
Economic analysis	Revenue of cryptojacking	[10], [22], [5], [12]
Risk analysis	Analysis of risk factors for cryptojacking	Our work

TABLE 3: Number of active cryptojacking websites

The list of potential cryptojacking websites obtained from the different sources is validated using the Minesweeper tool [15] to determine which websites are still performing cryptojacking.

	# websites
Initial dataset from sources	5700
Active Cryptojacking websites	86
Potential Case group	86

TABLE 4: Number of websites in the Case and Control

Some websites from both the *case* and the *control* groups were not accessible during the data collection phase performed with `WhatWeb`. We reduced the analysis on the websites that were both accessible and included in the Top 7.5M Tranco ranking.

	# Case	# Control
Initial dataset	86	415
Successfully Crawled w/ `WhatWeb`	82	378
Ranked in Tranco 7.5M	53	378
Final group size	53	378

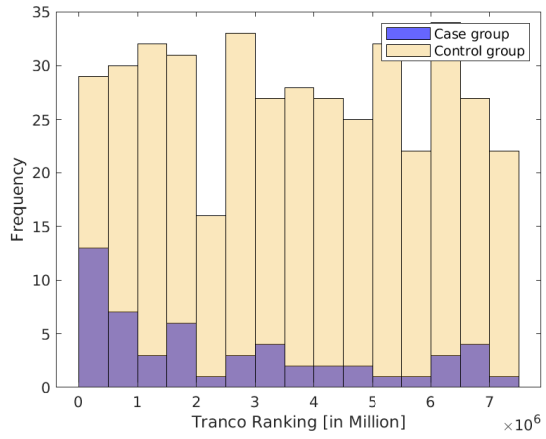


Figure 1: Distribution of Tranco Ranking in *Case* and *Control* group.

and maximum Tranco ranking [23] to determine the population, in terms of the interval in the Tranco top sites ranking, that generated the case. We found that some of the websites were not included in the ranking, thus we ignored them in the analysis (Tab. 4) to correctly compare the *case* with a *control* group from the same population. Fig. 1 shows the distribution of the case in the Tranco 7.25M ranking. To perform a population-based case-control study, the *control group*, i.e. the group of websites that did not develop the "disease", can be *randomly* sampled from the same population of the cases [11]. We thus randomly extracted a number of websites of roughly 7 times the size of the *case* group from the Tranco Top 7.25M list in the period from 01/04/2020 to 30/04/2020 to get a variety of website configurations representative of the population in that range of ranking (Fig. 1). We controlled to not include in the *control* group websites present in the *case group* and we checked that the websites in the *control* were not performing cryptojacking using the Minesweeper tool [15].

From each website in the *case* and the *control* group, we collected information about its configuration using the `WhatWeb` tool.[8] To bypass common anti-crawler countermeasures implemented by websites we employed a realistic user-agent. Not all the websites were successfully crawled by `WhatWeb` either due to downtime problems[9] or due to anti-crawling countermeasures. Thus, we ignored them for our analysis. Table 4 summarizes the size of the *case* and *control* groups at the end of the entire procedure.

For our analysis we identified the following information from the `WhatWeb` output:

- Server type (e.g. *Apache*, *Nginx*, and *Microsoft-IIS*) and version;
- CMS type (e.g. *WordPress*, *Drupal*, and *Joomla!*) and version;
- The *X-Powered-By* header.

The JSON file obtained from `WhatWeb` contains a sequence of HTTP responses. We developed a script to parse the file, follow the redirections, and extract the information of interest. In many cases, the responses from different websites were not standardized and therefore we observed some small differences during the extraction of certain information, for example, some websites reply with a server type *nginx-rc*, while others with *Nginx*. Our script can classify both cases within the same category.[10]

4.2. Data Overview

Fig. 2 shows the percentage of software information we were able to discover for the *case* and the *control*. Tab. 5 and Tab. 6 show the number of instances for each server technology and for each CMS considered in the *case* and *control*. The *others* category contains other less known technologies accordingly to their market share[11].

We computed the odds ratio for the different web server technologies compared with *Apache*, the web server

8. https://github.com/urbanadventurer/WhatWeb

9. Scanning with [15] and with `WhatWeb` were done one day after the other.

10. We performed this operation only for the most known type of servers: *nginx*, *apache*, *litespeed*, *microsoft-IIS*, *cloudflare*, *openresty*, and *amazons3*.

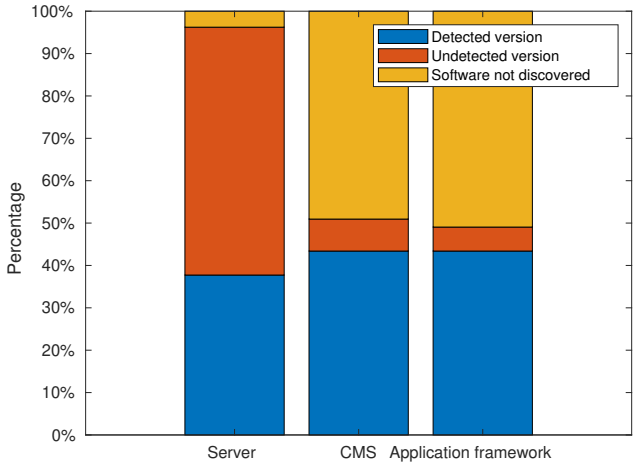
11. https://w3techs.com/

TABLE 5: Number of web server in each group by type

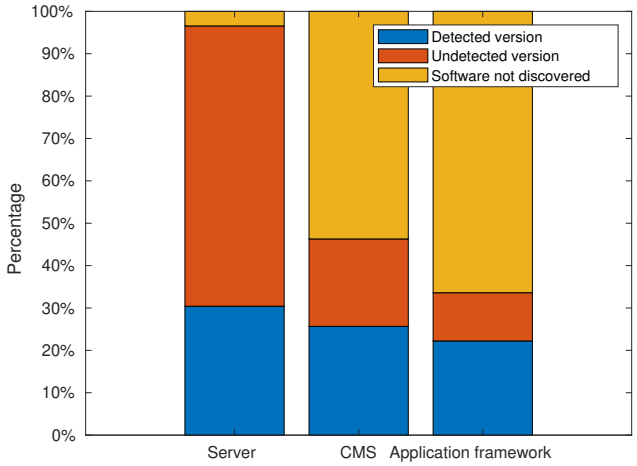
	# in Case group (%)	# in Control group (%)
Nginx	20 (37.7%)	116 (30.7%)
Apache	13 (24.5%)	121 (32.0%)
Cloudflare	9 (16.9%)	47 (12.5%)
Microsoft-IIS	3 (5.7%)	27 (7.1%)
Litespeed	3 (5.7%)	16 (4.2%)
Others	3 (5.7%)	38 (10.1%)
Unknown	2 (3.8%)	13 (3.4%)
Total	53	378

TABLE 6: Number of CMS in each group by type

	# in Case group (%)	# in Control group (%)
WordPress	22 (81.5%)	130 (74.3%)
Joomla!	1 (3.7%)	5 (2.9%)
Shopify	0 (0%)	9 (5.1%)
Drupal	2 (7.4%)	8 (4.6%)
Squarespace	0 (0%)	2 (1.1%)
Wix	1 (3.7%)	8 (4.6%)
Blogger	0 (0%)	2 (1.1%)
Magento	1 (3.7%)	0 (0%)
Others	0 (0%)	11 (6.3%)
Total	27	175



Percentage of software discovered in the *case* group via crawling.



Percentage of software discovered in the *control* group via crawling.

Figure 2: Percentage of software information discovered in the *case* and *control*. Information about the web server type is always available. Roughly 50% of the websites in both groups employ a CMS. Information about the application framework seems to be hidden more in the *control*.

with the highest market share on the Internet. We observed that the odds ratio for *Microsoft-IIS* is roughly 1, thus this web server is neither a positive nor a negative risk factor. Interestingly, *Nginx* has an odds ratio of roughly 1.6 i.e. this technology is more likely to be targeted for cryptojacking campaigns compared to *Apache*. As of now, the results obtained are not statistically significant because their 95% confidence interval is (0.76,3.37) for *Nginx* and (0.27,3.88) for *Microsoft-IIS*.

A recent study claimed that *WordPress* is a driving factor for cryptojacking campaigns [4]. We want to provide further analysis to determine if these claims are true or if these observations are influenced by the fact that *WordPress* is very common on the Internet. We computed the odds ratio of websites with different CMS compared to the websites that do not employ any CMS. We observed that some CMSs present a certain association with the presence of cryptojacking activities. Interestingly these CMSs are not the most common ones. For example, *Drupal* has an odds ratio of roughly 2, while *Wordpress* presents an odds ratio of 1.32. In other words, there is not a very *strong* positive association with the risk of being compromised for the most deployed CMS. As of now, due to the limited amount of subject in the *case* group, the results are not statistically significant because the 95% confidence interval of *Drupal* is (0.39,9.69), while for *Wordpress* is (0.71,2.43).

Finally, we observed that hiding software information like CMS, Server, and application framework versions (*X-Powered-By* header) is a negative risk factor with an odds ratio of 0.27. However, there no statistical significance because the 95% CI is (0.03,2.11).

5. Limitations and Future Work

The list of websites in the *case* and *control* group can potentially contain false positive and false negative respectively. However, the author of Minesweeper validated all the websites found by the tool and did not find any false positive [15]. We plan to collect a larger *case* and *control* group and a more complete list for the ranking (e.g. Alexa) to identify the factors that increase the risk of compromise for cryptojacking with statistical confidence. We also plan to extend the analysis to less common software.

The *case* group contains websites compromised to perform cryptojacking campaigns but can potentially contain websites that intentionally performed cryptojacking. This can influence the odds ratio for the characteristics we observed. However, since cryptojacking is now not as remunerative as ads [22], [30], the owners of the websites are not incentivized to use miners. Furthermore, previous studies showed that a high number of websites that perform cryptojacking are part of campaigns [4], [15]. We thus think that the majority of the websites in the *case* are compromised. However, we plan to eliminate the outliers via manual inspection.

We collected visible characteristics associated with hardening practices based on what can be observed from an external client, thus we ignored protections mechanisms[12] that require invasive analysis of a website.

12. That in many cases require an internal point of view. E.g. SQL injection sanitization mechanisms, etc.

We plan to enrich our analysis considering the security headers available from the server response (e.g. *X-XSS-Protection*, *CSP*, etc.).

In the future, we plan to determine if attacker's preferences on technologies change depending on the malicious activity by comparing the risk factors obtained from cryptojacking websites with phishing websites (e.g. [31]) and compromised websites that deploy malware.

Finally, we would like to perform a longitudinal analysis to determine if and how the risk factors change over time.

6. Conclusions

We have investigated the problem of cryptojacking, where websites illicitly hijack the computational resources of the visitors to mine anonymous cryptocurrency. We have analyzed the cryptojacking websites found in the wild and shown that certain technical characteristics of the web applications could potentially be positive and negative risk factors using a case-control study. However, further analysis must be conducted to obtain results that are statistically significant. Our analysis is still preliminary and with some limitations, which we plan to address in future work, but at the same time, we have obtained some interesting results, that we hope to spark the discussion in the security community.

Acknowledgments

This work is partly supported by the European Unions H2020 grants 830929, CyberSec4Europe.

References

[1] L. Allodi and F. Massacci. Comparing vulnerability severity and exploits using case-control studies. *ACM TOPS*, 2014.

[2] L. Allodi and F. Massacci. Security Events and Vulnerability Data for Cybersecurity Risk Estimation. *Risk Analysis*, 2017.

[3] L. Allodi, F. Massacci, and J. Williams. The work-averse cyber attacker model. evidence from two million attack signatures. In *Proc. of WEIS'17*, 2017.

[4] H. L. J. Bijmans, T. M. Booij, and C. Doerr. Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale. In *Proc. of USENIX'19*, 2019.

[5] H. L. J. Bijmans, T. M. Booij, and C. Doerr. Just the tip of the iceberg: Internet-scale exploitation of routers for cryptojacking. In *Proc. of CCS'19*, 2019.

[6] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker. Beyond heuristics: Learning to classify vulnerabilities and predict exploits. In *Proc. of SIGKDD'10*, 2010.

[7] N. Christopher. Hackers mined a fortune from Indian websites, 2018. https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/hackers-mined-a-fortune-from-indian-websites/articleshow/65836088.cms. Accessed: 2020-02-20.

[8] R. Doll and A. B. Hill. Smoking and carcinoma of the lung. *British medical journal*, 1950.

[9] J. R. Douceur. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[10] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark. A first look at browser-based cryptojacking. In *Proc. of EuroS&B Workshop-18*, 2018.

[11] B. U. P. Health. Case-control studies. [online] http://sphweb.bumc.bu.edu/otlt/MPH-Modules/EP/EP713_Case-Control/EP713_Case-Control_print.html.

[12] 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 *Proc. of CCS'18*, 2018.

[13] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker. Improving vulnerability remediation through better exploit prediction. 2019.

[14] M. Jakobsson and A. Juels. Proofs of Work and Bread Pudding Protocols. In *Secure Information Networks*, pages 258–272. Springer, 1999.

[15] 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 *Proc. of CCS'18*, 2018.

[16] J. Liu, Z. Zhao, X. Cui, Z. Wang, and Q. Liu. A novel approach for detecting browser-based silent miner. In *Proc. of DSC'18*, 2018.

[17] M. Maunder. WordPress Plugin Banned for Crypto Mining, 2017. https://www.wordfence.com/blog/2017/11/wordpress-plugin-banned-crypto-mining/. Accessed: 2020-02-20.

[18] K. McCarthy. CBS's Showtime caught mining crypto-coins in viewers' web browsers, 2017. https://www.theregister.co.uk/2017/09/25/showtime_hit_with_coinmining_script/. Accessed: 2020-02-20.

[19] M. Murphy. YouTube shuts down hidden cryptojacking adverts , 2018. https://www.telegraph.co.uk/technology/2018/01/29/youtube-shuts-hidden-crypto-jacking-adverts/. Accessed: 2020-02-25.

[20] T. Mursch. Over 100,000 Drupal websites vulnerable to Drupalgeddon 2 (CVE-2018-7600), 2018. https://badpackets.net/over-100000-drupal-websites-vulnerable-to-drupalgeddon-2-cve-2018-7600/. Accessed: 2020-02-20.

[21] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. https://bitcoin.org/bitcoin.pdf. Accessed: 2020-02-20.

[22] P. Papadopoulos, P. Ilia, and E. P. Markatos. Truth in web mining: Measuring the profitability and the imposed overheads of cryptojacking. In *Proc. of ISC'19*, 2019.

[23] V. L. Pochat, T. van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proc. of NDSS'19*, 2019.

[24] S. Ransbotham and S. Mitra. Choice and chance: A conceptual model of paths to information security compromise. *Inf. Sys. Res.*, 20, 2009.

[25] J. Rauchberger, S. Schrittwieser, T. Dam, R. Luh, D. Buhov, G. Pötzelsberger, and H. Kim. The other side of the coin: A framework for detecting and analyzing web-based cryptocurrency mining campaigns. In *Proc. of ARES'18*, 2018.

[26] J. D. P. Rodriguez and J. Posegga. Rapid: Resource and api-based detection against in-browser miners. In *Proc. of ACSAC'18*, 2018.

[27] R. Tahir, S. Durrani, F. Ahmed, H. Saeed, F. Zaffar, and S. Ilyas. The browsers strike back: Countering cryptojacking and parasitic miners on the web. In *Proc. of INFOCOM-19*, 2019.

[28] S. Tajalizadehkhoob, T. van Goethem, M. Korczynski, A. Noroozian, R. Böhme, T. Moore, W. Joosen, and M. van Eeten. Herding vulnerable cats: A statistical approach to disentangle joint responsibility for web security in shared hosting. In *Proc. of CCS'17*, 2017.

[29] O. Thonnard, L. Bilge, A. Kashyap, and M. Lee. Are you at risk? profiling organizations and individuals subject to targeted attacks. In *Proc. of FCDS'15*, 2015.

[30] S. Varlioglu, B. Gonen, M. Ozer, and M. F. Bastug. Is cryptojacking dead after coinhive shutdown? *ICICT-20*, 2020.

[31] M. Vasek, J. Wadleigh, and T. Moore. Hacking is not random: A case-control study of webserver-compromise risk. *IEEE Trans. Dependable Sec. Comput.*, 2016.

[32] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao. SEISMIC: secure in-lined script monitors for interrupting cryptojacks. In *Proc. of ESORICS'18*, 2018.

[33] T. Yen, V. Heorhiadi, A. Oprea, M. K. Reiter, and A. Juels. An epidemiological study of malware encounters in a large enterprise. In *Proc. of SIGSAC'14*, 2014.

[34] S. Zhang, X. Ou, and D. Caragea. Predicting cyber risks through national vulnerability database. *Information Security Journal: A Global Perspective*, 2015.