A Retrospective Analysis of User Exposure to (Illicit) Cryptocurrency Mining on the Web

Ralph Holz
University of Twente
r.holz@utwente.nl

Diego Perino
Telefonica Research
diego.perino@telefonica.com

Matteo Varvello
Brave Software
varvello@brave.com

Johanna Amann
ICSI
johanna@icir.org

Andrea Continella
University of Twente
a.continella@utwente.nl

Nate Evans
University of Denver
nathan.s.evans@du.edu

Ilias Leontiadis
Samsung AI
i.leontiadis@samsung.com

Christopher Natoli
University of Sydney
christopher.natoli@sydney.edu.au

Quirin Scheitle
Technical University of Munich
scheitle@net.in.tum.de

Abstract—**In late 2017, a sudden proliferation of malicious JavaScript was reported on the Web: browser-based mining exploited the CPU time of website visitors to mine the cryptocurrency Monero. Several studies measured the deployment of such code and developed defenses. However, previous work did not establish how many users were really *exposed* to the identified mining sites and whether there was a real risk given common user browsing behavior. In this paper, we present a retroactive analysis to close this research gap. We pool large-scale, longitudinal data from several vantage points, gathered during the prime time of illicit cryptomining, to measure the impact on web users. We leverage data from passive traffic monitoring of university networks and a large European ISP, with suspected mining sites identified in previous active scans. We corroborate our results with data from a browser extension with a large user base that tracks site visits. We also monitor open HTTP proxies and the Tor network for malicious injection of code. We find that the risk for most Web users was always very low, much lower than what deployment scans suggested. Any exposure period was also very brief. However, we also identify a previously unknown and exploited attack vector on mobile devices.**

I. Introduction

In September 2017, the filesharing site Piratebay was reported to engage in browser-based mining of the Monero cryptocurrency by including JavaScript and Web Assembly (WASM) code from the Coinhive mining pool. Due to Monero's cryptographic design, such mining was viewed as a possible revenue model for site operators. Dozens of pool operators and mining variants emerged soon (see Figure 1). The term cryptojacking was soon coined for sites that exploited their visitors' CPU without informing them; attackers even compromised websites [1] and exploited known weaknesses of the Drupal web framework [2], [3] to plant mining code. Starting in late 2018, a number of researchers investigated the deployment of cryptomining code on websites and the operator ecosystem [4], [5], [6], [7]. While their results are not quite consistent, they still demonstrated that deployment was occurring on thousands of websites or more.

These previous lines of research left some important questions unanswered, however, as the methods used were almost exclusively short-term, active scans of parts of the web. These are well-suited to gauging deployment, but they cannot establish if users were actually *exposed* to mining as they do not measure *how many* and *how often* users encounter mining sites. This requires *passive* observation of user behavior to understand the actual risks. The above methods also miss out on attack vectors that are known to be of practical relevance, such as the injection of mining code by HTTP proxies [8].

In this paper, we pool existing datasets from several research groups and vantage points that were created during 2018-2019. We present a *retrospective*, longitudinal view of user exposure to browser-based cryptocurrency mining and investigate various forms of exposure due to the injection of mining code by intermediary nodes. Our main contributions are as follows:

1) User exposure. We measure user exposure by leveraging passive traffic monitoring, using monitoring stations in several North American research and education networks, a large European mobile ISP and an existing Chrome extension with a large user base. Our measurements are longitudinal. We show that the problem was by far not as often experienced by users as previous deployment measurement suggested: users were rarely, if at all, ever exposed.

2) Alternative attack vectors. We check the ecosystem of open proxies for malicious injection of cryptomining code, testing up to 250k proxies per day, and monitor exit nodes on the Tor network for injections. We show that the attack vector was real and exploited, but fortunately most users were not at risk.

3) Unknown attack vector. We identify an unknown attack vector in our passive data and trace it to manufacturers and/or suppliers of cheap mobile phones who upload apps without user consent. The apps cause cryptocurrency to be mined on the phones.

The remainder of this paper is organized as follows. We give an overview of related work and position our contribution in Section II. Section III presents our methodology and discusses ethical considerations. We present our results in Section IV and discuss them in Section V.

978-3-903176-27-0 ©2020 IFIP

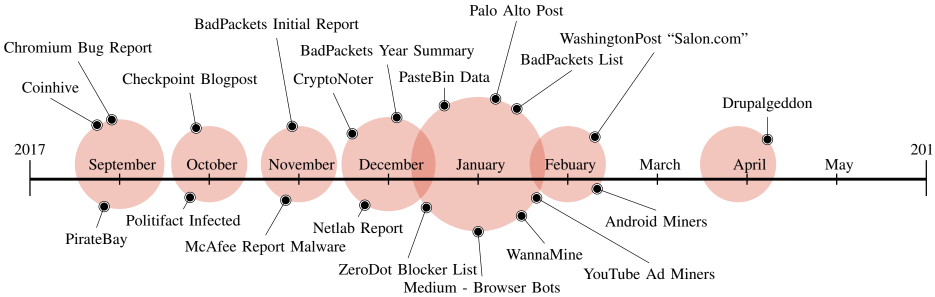


Figure 1. Timeline of key events during the rise of cryptojacking. Circle size is a rough measure of the number of reports in the respective period.

II. Related work

Previous work on in-browser mining focused nearly exclusively on deployment, campaigns, defenses, and estimates of revenue. To the best of our knowledge, passively obtained data is only used in [9] in a 30,000-ft estimate to show that some users were affected; however, this is not put into context of the overall user base. Injection of mining code by proxies, Tor nodes or mobile phone suppliers has not received any attention. Hence, our work is complementary to previous work. Table I visualizes this and summarizes previous contributions.

Previous work used various active detection methods and counted incidents quite differently. This makes it very difficult to compare between the authors. Some authors investigated only the landing pages of websites [4]; others followed internal links [5], [9]; others even counted links to external sites as a mining occurrence [6]. Some authors used static detection and pattern matching in their scans [4], [9], others used dynamic code analysis and/or hardware monitoring [5], [7]. Many investigated only domains from Alexa Top1m lists on single or very few occasions, although the high variability of this data source (up to 50% churn per day) is well established [10]. Others again sampled a large part of domains from public DNS zonefiles [4], [9]. It is usually impossible to give a reliable estimate how many users visited the respectively identified mining sites.

The authors of [6], [7], [5], [4] carried out active scans of Alexa lists. Detection is mostly based on dynamic analysis: monitoring the creation of Websockets, use of the Stratum protocol, and profiling the call stack and JavaScript worker threads. These methods have high accuracy. However, they are not applicable in a passive measurement because the affected machine can only be monitored in terms of network traffic. The publications most closely related to ours are probably the ones by Konoth *et al.* [5] and Bijmans *et al.* [9], which use the same toolset (using dynamic detection of the use of Websockets and the protocol to communicate with a mining pool). Bijmans *et al.* also analyze the nature and purpose of mining sites; this was out of scope for our study. Konoth *et al.* find that static string matching is already sufficient to detect 93% of mining sites. Dynamic detection is useful to eliminate false positives, however. They also propose a defense based on measuring the CPU cache and identifying cryptographic primitives in the call stack. They determine an incidence rate of 0.07% for landing pages and 0.17% when including internal links. Bijmans *et al.* report 0.01% in a smaller scan but 0.07% for domains on the Alexa list, concluding that Alexa scans overestimate the incidence rate. Hong *et al.* and Kharraz *et al.*'s tools produce deviating results (0.50%-0.87% incidence), but they count links to external sites as an incidence relating to the investigated site. Rueth *et al.*'s study [4] is quite different: it uses pattern matching for detection and dissects the Coinhive link-forwarding service.

III. Methodology

We use passive measurements to obtain an accurate picture of user exposure. Some of our passive measurements rely on input from active scans and classifications from blocklists to identify mining sites. We support open science and make code and non-privacy sensitive data in this paper available:

https://github.com/retrocryptomining

A. Deriving Classifiers

Samples. Mining code consists of two parts: a part responsible for the mining (WASM or, rarely, custom encodings) and helper code for configuration. We collected both helper and mining code, including for those mining pools that require manual user opt-in. We collected 78 samples between 2017-10 and 2018-07, from affected sites and by creating accounts with mining pools. This was a manual, iterative process following reports on the Web (especially [11]). We summarize our collection in Table II. In line with previous work, we find two algorithms dominate: Cryptonight and JSECoin. Mining pools use different Cryptonight implementations; however a significant number of samples are just variants of the implementations by Coinhive, Authedmine, and CryptoLoot. Some implementations use a custom byte encoding, not WASM. During our collection period, we noted a shift to heavy obfuscation, especially in the more recent Cryptonight implementations by CoinNebula and DeepMiner. We apply deobfuscation to verify a sample is mining code. For Authedmine, we develop a custom reverser. DeepMiner encrypts its code with AES; the key is located in the JavaScript itself. For other implementations, we apply partial evaluation using the open-source tool JStillery. Owing to our long collection period and the fact that most implementations are closely related variants

Table I
Comparison with related work. Half circles refer to methodological limitations documented in each research paper.

	Contributions						Methodology						Classifiers in scans								Incidence %
													static			dynamic					
	User exposure	Conn. injection	Deployment	Campaigns	Revenue	Mitigation	Monitoring	Active scans	Scan targets	Period	Links followed	Longitudinal	Patterns (samples)	Patterns (blocklists)	Signatures	CPU usage	CPU cache	Code monitoring	Network stack	Library detection	
Bijmans *et al.* [9]	◐	○	●	●	●	○	◐	●	Top1m; 49m	2018-12–2019-01	○	○	○	●	○	◐	○	●	●	○	0.01-0.07%
Hong *et al.* [6]	○	○	●	●	●	○	○	●	Top100k	2018-04	●	○	○	○	○	○	○	●	○	○	0.50-0.87%
Kharraz *et al.* [7]	○	○	●	●	○	○	○	●	Top1m; 600k	2018-02–2018-10	●	●	○	○	○	○	○	●	●	○	0.59%
Konoth *et al.* [5]	○	○	●	●	●	●	○	●	Top1m	2018-02	●	○	○	●	○	◐	●	●	●	○	0.07-0.17%
Rüth *et al.* [4]	○	○	●	◐	●	○	○	●	Top1m; 12m	2018-01–2018-05	○	●	○	●	●	○	○	○	●	○	0.08%
This work	●	●	◐	○	○	○	●	●	Top1m; 265m	2018-01–2018-11	○	●	●	●	●	○	○	○	○	●	0.02-0.12%

Table II
Our samples by family and origin. Helper code counted across families.

Family/type	Samples (obfuscated)	Sources
Cryptonight family...		
Authedmine	6 (6)	*authedmine.com*; affected sites
Coinhive	9 (0)	*coinhive.com*, *npm*; affected sites
Cryptoloot	10 (8)	*Cryptoloot* (Github); affected sites
Other	37 (27)	respective pools; affected sites
JSECoin	2 (0)	*JSECoin*; affected site
Helper code	14 (1)	respective pools; affected sites

of Cryptonight, we believe our samples provide excellent coverage of most cryptocurrency mining in use in 2018-2019.

Blocklists. We use classifications from five important blocklists as further input. All are available on Github or via the respective browser extension: NoCoin, coinhive-block, coinhive-blocker, crypto-miner-blocker, and MinerBlock. We consolidate these from 2017-11 to 2018-07. We eliminate overambitious regular expressions (e.g., single-letter filenames). The regular expressions we deem suitable for detection cover URLs of mining domains, URLs of Websockets, JavaScript filenames, and a small number of highly specific keywords.

Further input. The authors of [5] provide us with a list of mining sites they identified in 2018-02.

From our samples and regular expressions, we derive a number of classifiers. Two forms of classifiers are for static analysis in active and passive measurements. One classifier can be used to inspect JavaScript code at runtime.

C1: pattern matching. Classifiers in this category are string patterns for both obfuscated and non-obfuscated mining code, including various alternative encodings of WASM. In 2018-01, there are 45 classifiers in this category (one was later found to be redundant). Our set grows to 58 in 2018-03 and to 364 in 2018-07. This was due to fast growth of the blocklists, which included ever longer lists of proxy domains for Websockets, used by mining scripts to avoid detection.

C2: signatures. We implement MD5, fuzzy hashing[1], and Yara signatures[2] to identify and differentiate between variants of mining scripts.

C3: namespaces. We find that global variables are commonly used in mining code to simulate namespaces, with variable names often unique to a family of related implementations. We use this for classification in our browser extension.

An obvious limitation of our approach is that we can only detect known samples and closely related derivates that preserve the key characteristics we extracted. We argue that this limitation should be of little relevance for this study due to our broad coverage.

B. Active Scanning of Domains

We use scans of landing pages to validate the coverage and accuracy of our classifiers and create input data for our passive measurements. Our scanner (*cm-screener*) is custom-built and written in Go. We pick scan targets from two sources. We scan domains on the Alexa Top 1m, with a fresh list downloaded before each scan. We also reuse the methodology from [12]. We collect domains from ICANN's CZDS (*czds.icann.org*), Certificate Transparency logs, and the *.com*, *.net*, *.org* zones. We add domains from the Umbrella and Majestic top lists. We resolve all domains with *massdns* and scan IP addresses with *zmap* to eliminate unreachable domains.

Initally, we use *cm-screener* to test landing pages for the presence of links to JavaScript mining code, in particular links to the respective mining pools. This was a very common deployment pattern in the early phase of in-browser mining. Such a scan can screen nearly 100m domains in under 24h. With deployment diversifying and attempting to evade detection, we extend the scanner to parse landing pages and retrieve all JavaScript (inline and linked) and match the code against our classifiers. The parsing is considerably slower; we limit ourselves to one full scan of the Alexa list and just under 20m random domains from our larger input list. We evaluate our detection rate by statistical sampling and determine the sample proportion of true and false positives (we count commented mining code as a false positive). We show in Section IV-A

[1] github.com/DinoTools/python-ssdeep

[2] github.com/VirusTotal/yara

that this identifies mining sites with the accuracy required for our passive measurements.

C. Passive Monitoring

Our passive monitoring is carried out on traffic data from research and education networks and a mobile ISP.

Research and education networks. We have access to data from a long-running project [13] monitoring TLS network connections from several North American research and education networks with more than 100k users. This includes campus networks and student accomodation networks. We use the Server Name Indication (SNI) of TLS to determine the destination domain of a connection. This allows us to identify HTTPS connections to suspected mining domains and Websocket proxies.

Mining sites cause the browser to open Websocket connections to mining pools and proxies. We check for these using the Websocket URLs from category C1. The Websocket protocol is an in-band upgrade to HTTPS, *i.e.,* they are not directly inferable. However, mining pools typically host their Websocket endpoints on subdomains beginning with *ws*, e.g., *ws1.coinhive.com*, and most code samples we collect follow this pattern and connect to the proxies provided by mining pools. Bijmans *et al.* independently confirm this in [9]; the use of other proxies is rare. Consequently, we count only connections to domains of this format and to the pools known to us as a mining connection.

We inspect data from our passive monitoring going back to 2015 and ending in 2018-11, 280 billion connections in total. We pick up the first mining-related connections (to Coinhive) in 2017-08; this was a month before the first reports.

Mobile ISP. We inspect our samples and find that very few deactivate mining on mobile browsers. Concluding that users of handheld devices should also be affected by in-browser mining, we leverage our access to data from a large European mobile ISP with tens of millions of subscribers in one country to identify mining activity. We collect passive traces and summary statistics from 2018-01 to 2018-08. Specifically, we leverage two separate data sources. The first is a transparent middlebox used by the ISP to optimize mobile traffic. The second is a device database linking a device ID (IMEI) to a specific device model, OS, and manufacturer. We refer to entries in our first source as ‘transactions’. Note that our methodology does *not* allow us to identify tethering.

We provide the ISP with a list of regular expressions, built from our classifiers in C1 of 2018-01. The expressions cover Coinhive, JSECoin, and several providers using Cryptonight. They also identify Websocket endpoints and complete URLs of JavaScript mining code. The ISP builds traffic capturing rules based on these and applies them on our behalf. For HTTP, we match full URLs. For HTTPS, we match the domain name in the SNI. We collect several TB of transactions per day, with the following information for each matching transaction: timestamp, anonymized subscriber ID, URL, device model, and transaction size. Once we have obtained our statistical information, the original data is deleted by the ISP. One limitation to note is that we cannot update our list of regular expressions. However, all important mining providers (e.g., Coinhive, Authedmine, and JSECoin) are covered thanks to our collection of samples.

D. Alternative Attack Vectors: Tampering with Connections

We explore two alternative attack vectors that have not been investigated before.

Open proxies. Open web proxies relay and pseudonymize HTTP connections free of charge. Typical use cases include hiding the geographic origin and accessing geo-blocked content. The authors of [14] and [8] showed that such proxies may tamper with connections and modify content to inject advertisements and malware. We test whether open proxies inject mining code, following the methodology published in [8]. We use an array of scanners and aggregator lists to identify reported open proxies. Between 2017-10 and 2018-08, we access 250k reported open proxies *per day*. Of these, we classify only 25 826 as truly operational, *i.e.,* actually performing any proxying. We set a timeout of 60s to eliminate slow proxies; this results in 15 892 proxies we test for injections. We check for content manipulation by requesting a crafted bait webpage and comparing the received page with the original. We use classifiers from C1 and C2 to identify the kind of injection. We *manually* inspect and verify *all* results.

Injection in Tor. Tor hides client IP addresses by forwarding data through multiple routers in the network. The authors of [15] showed that some last hops (the *exit nodes*) may also tamper with traffic. We monitor Tor exit nodes daily from 2017-12 to 2018-08. Each day, we retrieve the list of active exit nodes from Tor directory servers. Testing each exit node, we download static, small bait pages pages via HTTP and compare them to versions served without Tor. Due to the early start, we use only 11 strings as classifiers. However, we choose them to ascertain breadth, including keywords for Coinhive and JSECoin. We manually verify all incidents of non-matching sites.

E. Understanding User Visits to Mining Sites

We leverage a Chrome extension deployed to investigate the ecosystem of open proxies [8]. Our extension tracks connections made via *benign* open proxies. It offers users to choose a proxy from a list of operational proxies that is continuously updated. Proxies that modify content are *never* added to this list. There are currently 7940 installations of this extension, with about 1400 active users per week. About 200GB traffic per month are inspected. To analyze whether proxy users are affected by cryptocurrency mining activity, we instrument the extension to use classifiers from category C1. We update the used classifiers as our list in category C1 grows. As the extension’s purpose is to prevent users from routing their traffic via tampering proxies, any injection that is still found must come *from the visited site*. This gives us useful information about users accessing such sites. In 2018-05, we add the detection of mining libraries (C3) as the browser extension gives us the necessary access for this form of dynamic analysis. We analyze our dataset until 2018-08, when we deactivate the analysis functionality.

F. Ethical Considerations

All our measurement methods have been cleared by the responsible entities and IRBs at each participating institution. This includes all institutions contributing data to our passive measurement of research and education networks. The latter data collection also excludes or anonymizes sensitive information such as client IP addresses before we see it. Data collection and short term retention at network middle boxes of the mobile ISP is in accordance with the terms and conditions of the ISP and the local regulations. These terms include data processing for research and publications as allowed usage of collected data. We only extract aggregated information and have no access to any personally identifiable identifiers. The Chrome extension to analyze proxy user traffic has a clear privacy policy enabling data collection and analysis for research and publication purposes. Analysis happens on the fly; no payloads are collected. Only anonymous data is collected; no personally identifiable information is included. Our active scans follow best practices such as rate limiting, blacklisting, and maintaining informative rDNS pointers.

IV. Results

We first present our results from the screening scans, which support our passive monitoring and allow us to have confidence in our classifiers. We then present results supporting our conclusion that very few users were exposed to cryptocurrency mining on the web, and only for very short periods of time. We discuss this in depth in Section V.

A. Providing input from active scans

We list our screening scans in Table III. For every suspected mining domain, *cm-screener* stores page dumps.[3] We take a simple random sample of size 250 (which meets the success/-failure condition for binomial experiments) from each scan and give the sample proportion of true positives ($\hat{p}$) together with 95% confidence intervals.

In 2018-01, we identify 240 of the 250 samples as true positives of sites loading mining code that would execute in the browser. In 5 further cases, the mining code had been commented; in five more cases our hits are false positives. This corresponds to a true positive rate of 0.95, with a 95% confidence interval of (0.94, 0.98). The vast majority of the 240 cases are inclusions of Coinhive. For the scans in 2018-03 (Alexa and large-scale scan of 265m domains), we find very similar values. The scans in 2018-07, however, produce a lower true positive rate. Inspecting our results, we note an overreach in blocklist entries. Removing just three entries reduces the false positives by nearly half. On the whole, however, our incidence rates are remarkably consistent with related work. With $\hat{p} = 0.96$, the identified mining sites from the Alexa scan in 2018-01 correspond to an estimated incidence rate of 0.12. For 2018-03, we obtain 0.09. We hence conclude that our classifiers produce results that are consistent with other authors. We do not eliminate the false positives of 2018-07 as they are not damaging to our passive measurements. As we show in Section IV-B, user exposure is so low that checking for too many suspected mining sites in our passive monitoring has no further impact.

B. Passive Monitoring

We analyze our data from passive observations of network traffic and contrast the results from research and education networks with those from the mobile ISP.

1) Research and education networks: We first inspect all connections in our entire observation period for mining-related domains to which Websocket connections are made. We use the Websocket domains/URLs from our samples and blocklists as input (C1). Over the entire period, we identify only 1.4m connections (out of billions) to a total of 398 domains. The domain names differ only in the digits following the *ws* prefix. Going through these manually and grouping them together, we arrive at only 15 distinct domains to which 995k connections are made. Recall that category C1 is based on real, verified samples and entries from blocklists, with more than 200 string patterns that identify Websocket endpoints. Most of these receive no traffic at all. The amount of network traffic going to known Websocket endpoints must be described as minuscule. The finding implies that none of these sites belong to the popular domains on the Web that attract many users.

Grouping endpoints for mining pools and proxies together under the term *dropboxes*, we list the top domains by number of connections in Table IV. The Coinhive mining pools, *i.e.,* including Authedmine, dominate by number of connections. More than 85% of our dropbox connections terminate there.

Coinhive. Focusing on Coinhive connections only, Figure 2 shows the longitudinal development over time. The percentage of connections per day is very low. In absolute numbers, the peaks are at 22k connections per day; typically, this number is much lower at several hundred. We observe most connections after 2017-09, when activity picks up rapidly and peaks a first time in 2017-12, when in-browser mining was in the news. There are two more peaks in 2018-04 and 2018-05. They coincide with the widely reported vulnerability in the Drupal CMS ('Drupalgeddon') known to be exploited at that time for cryptomining [2], [3]. According to *coinmarketcap.com*, there was also a short price surge around this time. We see an ever lower level of connections in the second half of the year—at most a few thousand connections per day. We believe the reasons to include Drupal updates, more comprehensive blocklists, and a general decline of the cryptojacking practice due to lower prices. Note that our sample collection is evergrowing. In 2018-07, it already contains many of the newer, alternative mining pools.

Casting a wider net. We check how many identified mining domains appear in our scans and in our passive observations.

[3] Due to a technical issue, in a small number of cases only the mining code was stored to disk. We used the Wayback machine to confirm it had been included in the HTML.

Table III

SCREENING SCANS. $\hat{p}$ IS THE SAMPLE PROPORTION OF *true positives* WE DETERMINED. WE GIVE A 95% CONFIDENCE INTERVAL.

Method	Date	C1	Input	Linked JS	Suspected mining sites	$\hat{p}$ (TP)	95%-CI $\hat{p}$ (TP)
cm-screener v1	31 Jan 2018	45	Top1m	-	1239	0.96	(0.94, 0.98)
cm-screener v1	12 Mar 2018	58	Top1m	-	994	0.95	(0.93, 0.98)
cm-screener v2	29 Jul 2018	364	Top1m	4.4m	742	0.78	(0.73, 0.84)
cm-screener v1	14–17 Mar 2018	58	265.3m	-	41 359	0.94	(0.91, 0.97)
cm-screener v2	28 Jul–7 Aug 2018	364	19.7m	30.0m	4398	0.82	(0.77, 0.87)

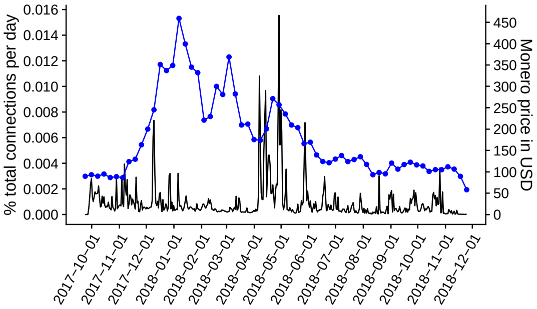


Figure 2. Black: Coinhive connections, passive measurement (%). Blue (dotted): Monero price.

Table IV

TOP 10 DROPBOXES BY CONNECTIONS.

Domain	Conns
ws.coinhive.com	760 164
ws.rocks.io	83 793
ws.authedmine.com	56 956
ws.coin-hive.com	35 068
ws.pzoifaum.info	19 135
ws.crypto-loot.com	17 683
ws.coin-have.com	7548
ws.staticsfs.host	3553
ws.aalbbh84.info	3245
ws.hemnes.win	3060

In the large scan of 2018-03, which covered 265m domains, we identify 41 359 mining domains. We search for these in our passive data, with a window reaching back 30 days. In the 8 billion connections of this 30-day window, we find only 4202 connections to just 441 of the 41 359 domains. There is a long-tail distribution: ignoring Coinhive and Authedmine, which host the mining code that is often linked to, we find mostly connections to movie streaming sites (roughly 30%). These are not the known, clearly legitimate providers but rather sites that seem to target the ‘grey area’ of streaming where the illegality of consumption of such streaming is unclear, disputed, or at least not acted upon. Just nine domains account for more than half the connections. Almost a third of domains receive just one connection. We repeat the analysis for the results of the smaller scans in 2018-01 and 2018-07. We find 156 domains in 49 742 connections (out of 5.4 billion in the preceeding 30 days) for 2018-01. For 2018-07, we find 44 domains in 1832 connections out of a total of 2.5 billion connections in the 30-day window.

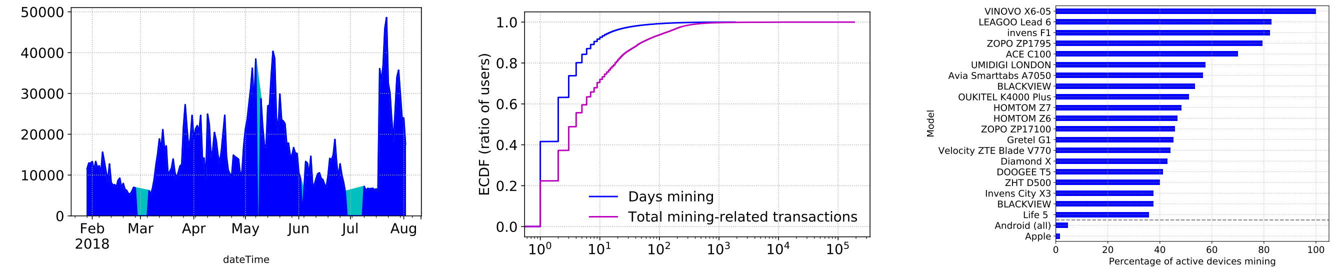
The clear picture emerging from our observations is that of a striking discrepancy between deployment and actual user exposure. The long-tail distribution offers evidence that blocklists in browsers and network gateways, while definitely incomplete, can provide a much better protection than related work assumed. We return to this in Section V.

2) Mobile ISP: The data from our mobile ISP covers a much larger user base and a long observation window (half a year). The picture that our data establishes is entirely consistent with our network monitoring of research networks. We identify 91m transactions over six months as mining-related. Of tens of millions of subscribers, only 1.3m unique users are ever affected. Coinhive dominates: we identify 30m transactions as access to Coinhive (860k unique users). Figure 3(a) shows the daily number of unique users who access domains that we have reason to suspect of hosting mining code. On average, 15k devices access these per day (out of tens of millions of active devices we monitor). The vast majority (97%) of transactions relate to Coinhive.

We identify the same peaks in 2018-04 and 2018-05 in this data source, coinciding with the Drupal vulnerability. A sudden peak at the end of 2018-07 occurs only in this data set, however. We have no compelling theory why this peak appears only in this data source. Inspecting the transactions manually, we confirm that they constitute increased download activity of the Coinhive script.

Frequency of user exposure. Mining on resource-constrained mobile devices is more detrimental to the browsing experience than mining on desktop computers. We are interested how long users are exposed, given that the authors of [6] claim that deployed mining code remains on a site for several weeks. Figure 3(b) shows the number of transactions and days that users access mining-related domains. Most users access such domains fewer than four times over six months. There is a long tail (0.5% of users) who have more than 100 mining-related transactions. We also observe that most users are only affected for a couple of days over the entire period of six months. Surprisingly, however, some users seem to access mining domains very regularly: 90k users do so for seven or more days and 14k access the domains for 90 days or more. As we show now, this is because of a previously unknown attack vector.

Unknown injection vector. Figure 3(c) plots the percentage of actively mining devices per model. We compute this by normalizing the number of unique devices accessing mining domains with the total number of such devices in the network over the same period. We only consider models with at least 50 unique active devices. We identify certain Android devices that are much more prominent than others. *All* 56 Vinovo X6-05 devices active during this period access mining websites. 136



(a) Number of unique users per day who accessed a mining domain over a period of 6 months. Cyan indicates periods where no collection was performed due to operational issues.

(b) Distribution of i) number of days each user was observed accessing mining domains and ii) the total number of transactions per user.

(c) Per-model percentage of active devices that accessed mining domains during 6 months. For comparison we also include the percentage for all Apple and Android devices.

Figure 3. Mobile ISP analysis.

out of 164 Leagoo Lead 6 devices and 353 out of 613 Umidigi London devices do so as well. Statistically, this is quite unlikely, and hence we hypothesize that some process other than normal user behavior causes the access. We survey the Amazon product pages for the devices and find that although they are very affordable, they are also equipped with relatively powerful hardware. For example, the Umidigi London is a quad-core, 5-inch smartphone with 1GB of RAM that cost only $US 75 in 2018. Interestingly, many buyers complain in reviews about advertisements appearing on the home screen. Others notice background activity, especially when charging, or applications that are installed without user interaction.

We buy one such device for testing purposes. A few weeks after activation, the phone starts displaying pop-ups, advertisements, and various apps are installed without user intervention. Google recently reported the Triada malware to have been injected into the supply chain for some of our affected devices [16]. Triada is known to cause similar behaviors. We test for the typical signatures identified by Google; however, none of them are present in our firmware. This suggests that our device is controlled in a different but possibly related way. It may well be that the apps installed without user permission load web sites in the background that contain mining code. It is unclear if this is the intention behind the installed apps or only a side-effect.[4]

C. Alternative Attack Vectors: Tampering with Connections

In line with our previous findings, we find few incidences of open proxies or Tor exit nodes tampering with traffic for the purpose of mining cryptocurrency.

Open proxies. Out of 15 892 proxies active and fast enough to deliver a site within our 60 seconds timeout, only 282 (1.7%) modify content. 6% of these proxies inject mining code, *i.e.,* a total of 0.11% of the tested proxies. We verify manually that each case is a true positive. All injections are related to Coinhive. Figure 4 shows our results over time. We see most mining activity from 2017-12 to 2018-03, which coincides with the ramp-up phase of in-browser mining and higher Monero prices. We never find more than ten proxies injecting mining code at any time. As in our other data sources, we see a decline beginning in mid-2018. We track whether proxies also manipulate TLS certificates, *i.e.,* stage a Person-in-the-middle attack on HTTPS connections. We find a similar fraction of proxies doing this—but when we inspect these cases, they are never the same proxies that inject mining code into HTTP.

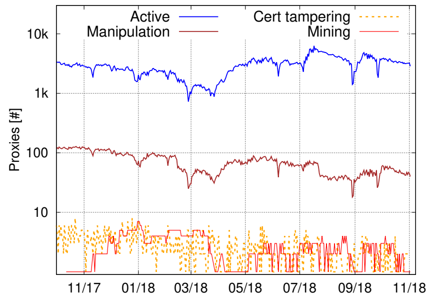


Figure 4. Open proxy ecosystem and miners evolution.

We measure how long proxies attempt code injection and establish a median of 22 days. A similar number has been reported for other malicious tampering by open proxies in [8]. It is also close to the number reported in [6] for websites hosting mining code: a third stops the mining within two weeks. The short duration during which injection of code by proxies happens contrasts with the overall lifetime of proxies, which spans months. One proxy, for example, is active for about one year but only injects mining code for one month.

Injection in Tor. Our analysis of injections at Tor exit nodes spans more than 200 days, with an average of 680 successful downloads of the bait page per day. The number of exit nodes in our measurement period is between 900 and 1100, although not every exit node provides us with a result. This is due to the way how onion circuits are created and the fact that exit nodes set policies for exit traffic. We exclude exit nodes that do not forward our HTTP requests. Although we use a very broad range of keywords, we find only one Tor exit node injecting mining code. This specific exit node injects an unobfuscated Coinhive script. It is only active for 13 days, during which we see the injection on four days. It is instructive to compare this with previous results on content modification in exit nodes. Winter *et al.* [15] found significant malicious behavior among Tor exit nodes in 2014. The number of exit nodes has not

[4] We make the firmware of our device available to interested researchers.

Table V
VISITS TO MINING SITES IDENTIFIED BY OUR BROWSER EXTENSION.

Month (2018)	Total visits	Visits to mining sites	%
Feb	113 257	18	0.016%
Mar	206 518	33	0.016%
Apr	171 555	89	0.052%
May	153 256	53	0.034%
Jun	195 092	40	0.021%
(Jul)	(160 373)	(461)	(0.287)%

changed significantly since then, but Winter *et al.* found 40 exit nodes tampering with HTTPS connections and 27 exit nodes stealing credentials from HTTP connections. However, Tor has a tech-savvy user base likely to notice tampering. The network now encourages users to report malicious exit nodes, which can be flagged in the directory service. We believe that Tor was hence never particularly appealing to attackers.

D. Understanding User Visits to Mining Sites

We investigate the extent of mining experienced by users of our browser extension. Note that it is configured in such a way that any mining code we find must come from the visited site as users cannot access malicious proxies.

We updated and extended the classifiers over time to ensure coverage and accuracy. Table V summarizes our findings. Once again, we find that users are very occasionally affected by mining. Mining sites are rarely accessed by users: they account for just 0.03% of the total number of domains our users visited. We find the characteristic bumps in 2018-04 and 2018-05, which coincide with the attacks on the Drupal content management system. This corroborates our previous results from passive measurement. We notice a peak in 2018-07. However, closer inspection reveals that this is an automated tool using our plugin repeatedly to visit a miner domain. Such tools have also been reported in [8]. The peak is hence not related to the one we find in our ISP data. Coinhive is the most common mining pool we encounter—we find it in 93% of sessions. This is followed by Authedmine, JSECoin, Cryptoloot, and Coinimp, attesting to the growing significance of alternative providers.

We investigate the duration of HTTP sessions with and without mining taking place. The median grows from six seconds for regular sessions to 30 seconds when mining occurs. This seems counter-intuitive as one would expect users to be annoyed by high CPU usage. However, it is likely that some mining sites offer attractive content, as also suggested in [4], [9] (entertainment and adult content). However, most sessions are too short to contribute significant mining results to a mining pool: just 10% of mining sessions last more than 16 minutes.

V. Discussion and Conclusion

In this paper, we set out to analyze the exposure of Web users to cryptocurrency mining. In a nutshell, do reported deployment rates between 0.01–0.1% imply a great risk for users or not? The conclusion that we offer is that the risk to users was greatly overestimated. This is due to the nearly exclusive use of active scans in previous work. In contrast to previous work, our study is *retrospective* and *longitudinal* in nature: we pool datasets raised by several groups around the planet to paint a more comprehensive picture of user exposure. In the following, we combine and discuss the results from our various data sources in context.

Exposure was very low. All our measurements show users were very rarely exposed to mining. We note that we verified high coverage and accuracy for our classifiers. Our own active scans, which provide some input to our passive observations, are consistent with related work. An important take-away to consider in future studies of Web attacks is the degree to which the use of deployment numbers alone is a very insufficient metric to express risk to users. The long-tail distribution of browsing preferences must be accounted for to quantify risks meaningfully. In our data, we were able to establish the long-tail distribution for mining sites that users actually connect to. Taking this into account, we believe that blocklists in browsers or on network gateways were actually also more effective than previous work gave them credit for; they should not be dismissed.

Exposure was very short. The number of connections to mining proxies alone do not reveal how users react to mining sites, and whether there is any measurable contribution to a pool's overall mining. The data we obtain from our browser extension gives us a unique perspective here. It corroborates that users encounter mining very rarely but also shows that most users stay briefly if they come across mining sites: usually around 30 seconds. As revenue is a driving force in the operations of cybercriminals, it is important to take this into account when new, different attacks exploit users' CPU time.

Risk from software monoculture. The peaks in our passive measurements correspond to compromises of a widely used content management system. This should be taken as yet another warning that software monocultures, combined with the known slow update rates in the web ecosystem, put users at risk too easily.

Uncommon attack vectors were tried. We expose one attack vector that was hitherto unknown: mobile devices loading apps that lead the devices to visit mining sites, possibly in the background. While a relatively infrequent occurence, our findings support the call for more stringent checks of operating systems, including in supply chains.

Acknowledgements

This work was supported partially by the Australian Government through the Australian Research Councils' Discovery Projects Scheme (project DP180104030). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or Australian Research Council. Ralph Holz received partial funding from an Early Career Researcher Award by the Faculty of Engineering, University of Sydney. This work was supported by the US National Science Foundation under award CNS-1528156. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators, and do not necessarily reflect the views of the NSF.

References

[1] B. Fung, "Hackers have turned Politifact's website into a trap for your PC," Washington Post, 13 Oct 2017.

[2] SecurityTrails, "Cryptojacking campaigns continue to target vulnerable websites," https://securitytrails.com/blog/cryptojacking-campaigns, 4 June 2018.

[3] Palo Alto Networks, "Exploit in the wild: #drupalgeddon2 – analysis of CVE-2018-7600"," https://researchcenter.paloaltonetworks.com/2018/05/unit42-exploit-wild-drupalgeddon2-analysis-cve-2018-7600, 1 May 2018.

[4] J. Rüth, T. Zimmermann, K. Wolsing, and O. Hohlfeld, "Digging into browser-based crypto mining," in *Proc. ACM Int. Measurement Conference (IMC)*, 2018.

[5] 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. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[6] 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. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018.

[7] A. Kharraz, Z. Ma, P. Murley, C. Lever, J. Mason, A. Miller, N. Borisov, M. Antonakakis, and M. Bailey, "Outguard: Detecting in-browser cryptocurrency mining in the wild," in *Proc. of the International Web Conference (WWW)*, 2019.

[8] D. Perino, M. Varvello, and C. Sorrente, "ProxyTorrent: Untangling the free HTTP(S) proxy ecosystem," in *Proc. of the International Web Conference (WWW)*, 2018.

[9] H. L. J. Bijmans, T. M. Booij, and C. Doerr, "Inadvertently making cyber criminals rich: a comprehensive study of of cryptojacking campaigns at Internet scale," in *Proc. USENIX Security*, Aug 2019.

[10] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A long way to the top: Significance, structure, and stability of Internet top lists," in *Proc. ACM Int. Measurement Conference (IMC)*, 2018.

[11] T. Mursch, "Bad packets report," https://twitter.com/bad_packets, 2018.

[12] Q. Scheitle, T. Chung, J. Hiller, O. Gasser, J. Naab, R. van Rijswijk-Deij, O. Hohlfeld, R. Holz, D. Choffnes, A. Mislove, and G. Carle, "A first look at Certification Authority Authorization (CAA)," *ACM SIGCOMM Computer Communication Review*, vol. 48, pp. 10–23, Mar 2018.

[13] J. Amann, M. Vallentin, S. Hall, and R. Sommer, "Extracting certificates from live traffic: A near real-time SSL notary service," TR-12-014, ICSI, Berkeley, CA, USA, Tech. Rep., 2012.

[14] G. Tsirantonakis, P. Ilia, S. Ioannidis, E. Athanasopoulos, and M. Polychronakis, "A large-scale analysis of content modification by open HTTP proxies," in *Proc. Network and Distributed System Security Symposium (NDSS)*, 2018.

[15] P. Winter, R. Köwer, M. Mulazzani, M. Huber, S. Schrittwieser, S. Lindskog, and E. Weippl, "Spoiled onions: Exposing malicious Tor exit relays," in *Proc. Privacy Enhancing Technologies Symposium*, Amsterdam, Netherlands, July 2014.

[16] L. Siewierski, "PHA family highlights: Triada," Google Security Blog. https://security.googleblog.com/2019/06/pha-family-highlights-triada.html.