# Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking

Hugo L.J. Bijmans, Tim M. Booij and Christian Doerr
Delft University of Technology, Cyber Security Group
2628CD Delft, The Netherlands
{h.l.j.bijmans@student.,t.m.booij@student.,c.doerr@}tudelft.nl

## ABSTRACT

The release of an efficient browser-based cryptominer, as introduced by Coinhive in 2017, has quickly spread throughout the web either as a new source of revenue for websites or exploited within the context of hacks and malicious advertisements. Several studies have analyzed the Alexa Top 1M and found 380 – 3,200 [5, 15, 18, 30, 31] (0.038% – 0.32%) to be actively mining, with an estimated $41,000 per month revenue for the top 10 perpetrators [18]. While placing a cryptominer on a popular website supplies considerable returns from its visitors' web browsers, it only generates revenue while a client is visiting the page. Even though large popular websites attract millions of visitors, the relatively low number of exploiting websites limits the total revenue that can be made.

In this paper, we report on a new attack vector that drastically overshadows all existing cryptojacking activity discovered to date. Through a firmware vulnerability in MikroTik routers, cyber criminals are able to rewrite outgoing user traffic and embed cryptomining code in every outgoing web connection. Thus, *every* web page visited by *any* user behind an infected router would mine to profit the criminals. Based on NetFlows recorded in a Tier 1 network, semiweekly crawls and telescope traffic, we followed their activities over a period of 10 months, and report on the modus operandi and coordinating infrastructure of the perpetrators, which were during this period in control of up to 1.4M routers, approximately 70% of all MikroTik devices deployed worldwide. We observed different levels of sophistication among adversaries, ranging from individual installations to campaigns involving large numbers of routers. Our results show that cryptojacking through MITM attacks is highly lucrative, a factor of 30 more than previous attack vectors.

## KEYWORDS

cryptojacking; MikroTik; router; MITM; cyber threat intelligence

**ACM Reference Format:**
Hugo L.J. Bijmans, Tim M. Booij and Christian Doerr. 2019. Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking. In *2019 ACM SIGSAC Conference on Computer& Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3319535.3354230


*CCS '19, November 11–15, 2019, London, United Kingdom*

ACM ISBN 978-1-4503-6747-9/19/11…$15.00
https://doi.org/10.1145/3319535.3354230

## 1 INTRODUCTION

Cryptocurrencies, which started with the release of Bitcoin in 2009 [24], represent monetary value secured by a blockchain. Transactions are permanently stored in an ever growing list of records, where transaction data can be added by solving a cryptographic challenge. This puzzle is dependent on the last block, the current transactions and their recipients, and once solved a new record gets inserted into the chain consolidating the record of previously conducted activities. Users are incentivized to participate and donate computational resources to the system, as the one solving the puzzle gets a (fraction of a) cryptocurrency unit as reward.

As the Bitcoin blockchain was designed to continuously increase the difficulty of these challenges, Bitcoin mining is no longer profitable on regular PCs, now requiring specialized hardware such as ASICs. As a result, thousands of other cryptocurrencies, so-called alt-coins, have emerged that replace the proof-of-work algorithm of Bitcoin with alternative mechanisms to validate transactions. The Monero cryptocurrency [39], which uses a private blockchain with transactions not publicly visible, relies on the CryptoNight algorithm, a memory-intensive computation of subsequent reads and writes that can be efficiently run using the processor-level cache found in typical consumer-grade CPUs.

The reward that can be gained from these alt-coins has however also attracted the attention of cyber criminals, who have distributed cryptomining code through malware or as part of botnet installations [28]. With the recent introduction of a JavaScript miner by Coinhive in 2017, cryptomining code can now be shipped as part of a web page and be efficiently executed by a web browser, thereby providing an easy, scalable and low-effort method to roll out cryptomining to a large user population. This has led to new business and revenue models, for example replacing advertisements by letting website visitors donate computational resources [40].

The relative ease with which website visitors can be recruited for cryptomining has also led to a major surge in illicit cryptomining, so-called cryptojacking or drive-by mining, in which the visitor's resources are hijacked without knowledge and consent. Aside from cryptojacking that is initiated by the website owner without their visitors' consent, criminals also seek to increase their revenue by compromising websites to install mining code [6, 20], as well as hiding miners in third party software used by web masters and thus inadvertently being deployed [2, 8, 42]. Cryptojacking is also spread through exploitable vulnerabilities in content management systems [23], through the distribution of advertisements including malicious code [22] or through malware [28]. Previous work by Konoth et al. estimate that cryptojacking possibly yields monthly revenues of $41,000 for the 10 most successful perpetrators across the Alexa Top 1M [18].

In this paper, we will analyze a previously unseen attack vector for cryptojacking, namely man-in-the-middle attacks launched through compromised consumer and edge routers that inject mining code into every web page requested by their users. This was made possible by a firmware vulnerability in MikroTik routers discovered in early 2018 [26], which allowed adversaries to change the device configuration and create an outgoing HTTP proxy, and that remained widely unpatched until a year later. By following reconnaissance scans of the perpetrators through a large network telescope, the detection of compromised routers through semiweekly crawls, and the tracing of connection patterns of adversaries, their supporting infrastructure and the compromised routers based on NetFlows from a Tier 1 operator, we are able to provide a comprehensive insight into how this vulnerability scaled out into massive cryptojacking campaigns that drastically overshadow previous mining activities. In this work, we make the following four contributions:

- We are first to investigate a new type of attack that exploits Internet infrastructure for cryptomining, and show how over a period of 10 months after the initial discovery of the vulnerability groups of criminals launch massive campaigns to control 1.4M routers, with a peak of 460,618 simultaneously infected routers.
- We have analyzed adversarial tactics and unveil the supporting infrastructure used within the campaigns, and are able to show differences between groups in how they locate their victims, compromise routers, and run their infrastructure.
- We demonstrate that previously reported vectors are negligibly small in number of affected users and created revenue, compared to the reported MITM attack. We find that this attack yielded monthly revenues, estimated exceeding $1,200,000 per month for the top 10 grossing accounts, a factor of 30 larger than previously estimated cryptojacking revenues from hacked websites, malicious advertisements and website-owner initiated mining combined.
- We have observed high levels of sophistication in three identified campaigns, of which the largest involved 40 mining accounts linked to one single actor.

The remainder of the paper will be structured as follows: Section 2 provides an overview of related work. Section 3 introduces the concept of cryptojacking and previously used modus operandi, and describes the vulnerability and its exploitation used for a MITM-based cryptojacking on routers. Section 4 describes the datasets used in this study. Section 5 presents the techniques, tactics and procedures in use during the identification, exploitation, monetization and maintenance of the compromised systems. Section 6 puts the techniques and sophistication levels of the ecosystem into perspective and quantifies adversarial revenues. Finally, Section 7 summarizes and concludes our work.

## 2 RELATED WORK

The growing interest in cryptojacking by cyber criminals was followed by an interest of the academic world to research this new phenomenon. Only shortly after the release of the Coinhive miner in September 2017, Eskandari et al. made the first explorations into the field by searching source code databases *Censys.io* and *PublicWWW* for strings known to be part of cryptomining libraries [12]. Due to the possibilities of JavaScript obfuscation and other hiding techniques, other research that followed soon focused on detection of these mining applications. Rauchberger et al. built *Mininghunter*, a crawler instructed to analyze both source code and WebSocket traffic of the visited pages [30]. A crawl of the Alexa Top 1M resulted in the identification of 3,178 cryptojacking websites and the discovery of a number of campaigns. Other web crawling studies performed by Parra Rodriguez & Posegga [31] and Carlin et al. [5] used machine learning techniques to determine active mining on a web page while crawling, and both reached high precision scores. The *CMTracker* made by Hong et al. [15] also crawls the web, but detects cryptojacking behavior based upon periodic executions in WebAssembly modules. This robust detection method was able to identify 868 actively mining websites in the Alexa Top 100K. Wang et al. performed a similar study by learning a support vector machine (SVM) on the characteristics of WebAssembly modules and concluded that analyzing WebAssembly modules is a very efficient and robust detection method [41].

The latest web crawling studies involve the work of Rüth et al., who crawled the three largest top-level domains (*.com, .net* and *.org*) as well as the Alexa Top 1M to estimate the prevalence of browser-based cryptomining (0.08% of the probed websites were actively mining) [32], Konoth et al., responsible for creating another crawler which identified 1,735 actively mining websites and performed campaign analysis to gain knowledge about the ecosystem [18], and Kharraz et al. performing a similar study but identified the actively cryptomining websites using machine learning techniques [17]. They also dedicated a section to campaign analysis, in which the authors identified 35 campaigns involving a total of 386 websites. The largest study to date across 55M websites discovered that the prevalence of cryptojacking significantly varied by top-level domain zone and the popularity of websites, and that about half of all cryptojacking activity is organized and part of a campaign [2].

Initial evidence begins to suggest that exploitation of websites for browser-based mining through their visitors might not generate the main source of revenue. Papadopoulos et al. concluded that advertisements are still over five times more profitable than browser-based cryptomining [27]. A longitudinal study performed by Pastrana et al. revealed that in the cryptojacking ecosystem, only a small number of cyber criminals is making large profits and those making profits had mined 4.3% of all Monero in circulation. [28].

Previous studies have primarily focused on either detection or the estimation of the compromised websites attack vector. While cryptojacking as part of a man-in-the-middle (MITM) attack – for example through a malicious WiFi network – is mentioned as being feasible by Eskandari et al. [12], this particular attack vector has never been researched before by the academic community. As a MITM attack will affect all traffic that crosses a particular device, the potential number of victims and with it potential revenue is however much higher. This paper will thus address this gap, and show how and to what extent cryptojacking is deployed in the wild through the attack on Internet infrastructure.

## 3 BACKGROUND

The introduction of memory-bound cryptocurrencies like Monero allowed for new methods of cryptomining, one of them being browser-based cryptomining. These cryptocurrencies, together with new web technologies such as WebAssembly (native speed code execution within the browser sandbox), WebWorkers (separate JavaScript instances), HTML5 WebSockets (simple multiplex TCP connection) and the Stratum Mining Protocol (JSON-RPC formatted mining pool communications) paved the way for the creation of an efficient browser-based cryptominer by Coinhive in 2017. Their miner, and most other mining applications, work as follows: the user visits a cryptojacking website which includes (a reference to) a cryptojacking script. This script explores the host system, downloads a highly optimized WebAssembly module for mining and spawns a number of WebWorkers to run this module. Consequently, it sets up a connection with a mining pool through a proxy server operated by the service, authenticating using a *siteKey* or (Monero) wallet address, which is essentially the account of the adversary. For readability when we mention *siteKeys* in the paper, we will refer to them by the first six characters of the key. The mining pool distributes a job to work on, the WebWorkers start mining and found hashes are submitted to the mining pool. When the browser window is closed, all mining activity stops.

### 3.1 Past modus operandi of cryptojacking

As stated by a New Jersey Attorney General in 2015 [14], mining cryptocurrencies with the computing power of others is not considered illegal when a clear notification of such activities is shown and the possibility of opting-out exist. However, most cryptojacking cases lack these and are therefore considered illegal. There have been cryptojacking scripts found on malware infected PCs [28], but since the release of the Coinhive miner, cryptojacking in the form of browser-based mining gained enormous popularity. There is a large number of websites running a cryptominer to increase their revenues, such as The Pirate Bay [40], but cryptojacking has also occurred on websites where the owner did not initiate it. Website compromises, such as government pages of the Indian government [6] in 2018, have lead to cryptojacking infections, but cyber criminals are constantly searching for more efficient methods to deploy their miners. To spread an infection over a large number of websites, attackers abused third party software (such as infecting WordPress plugins [42] or exploiting Drupal CMS vulnerabilities [34]) with cryptojacking scripts as well as injected advertisements with mining code and served them through ad networks to websites unaware of any infection [22].

### 3.2 Pervasive Cryptojacking through Man-In-The-Middle Attacks

As mentioned in the previous section, cryptomining code is included as part of the served HTML page, which requires the website owner to explicitly install a cryptominer or inadvertently embed it due to a compromised component. It is however also possible to modify the request in transit, by modifying the HTML as a man-in-the-middle.

In the attack reported in this paper, adversaries compromised the routers' operating system, and reconfigured the system causing requests from clients to any website to be rewritten and channeled

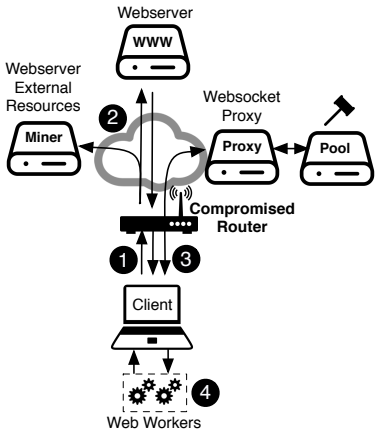


**Figure 1: Through a MITM attack on routers, adversaries performed cryptojacking on websites visited by users.**

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html;charset=windows-1251">
  <title>"http://www.facebook.com/"</title>
  <script src="https://coinhive.com/lib/coinhive.min.js"></script>
  <script>
    var miner = new CoinHive.Anonymous('<mining key>', {throttle: 0.1});
    miner.start(CoinHive.FORCE_EXCLUSIVE_TAB);
  </script>
 </head>
 <frameset>
  <frame src="http://www.facebook.com/"></frame>
 </frameset>
</html>
```

**Figure 2: HTML returned by the proxy of an infected router, with a Coinhive miner and the actual page in an iframe.**

to an internal HTTP proxy server running on the device. With the compromise of the router, the perpetrator installs a script to change the firewall rules of the device, opening telnet and SSH to the Internet if not already exposed, and introduces a firewall rule to redirect outgoing requests on port 80 to a proxy port. Finally, it deploys an HTML page sent by the proxy to each outgoing connection.

While different groups of actors followed slightly different techniques, tactics and procedures as we will show in Section 5, it meant as shown in Figure 1 from the perspective of the user that any outgoing connection to port 80 was redirected to the proxy on port 8080 (1). This served a web page based on a common template, shown in Figure 2 for a connection to `facebook.com`. This led the client's browser to fetch two web resources: the outer frame containing a JavaScript that loaded cryptomining code (2), and within the frame the actual website the user intended to visit was displayed (2). The client's web browser would setup a WebSocket connection to a WebSocket proxy or to a mining pool in order to retrieve instructions (3), and spin up WebWorkers to mine for a specific *siteKey* (4).

From the perspective of the perpetrator, this design has a number of advantages. First, as the iframe opens the original page, the user will at first sight not notice anything wrong, as the requested web page loads within the borderless iframe. Second, as the interaction

with the loaded website functions normally, the victim will remain on the Web page for an extended period of time, thus increasing the time the miner will run in the background. Third, as clicks on the embedded page do not reload the outer frame, the cryptominer keeps running during navigation on the visited web page, thus maximizing mining cycles.

*Susceptibility of HTTP(S) connections.* While the browser address would show a connection to the router instead of the requested URL, the hijack from a usability perspective is both comparatively frictionless and effective. The original URL is displayed as the title of the page, and experimentation on recent versions of both mobile and desktop browsers showed that websites can even be loaded via HTTPS within the iframe without triggering a warning by the browser. In this case, the HTTP proxy loads an unencrypted HTTP page with an iframe showing the secured HTTPS contents. Thus, unless the rewritten URL raises suspicion with the user, we can expect the activity to go by relatively unnoticed.

### 3.3 Vulnerability CVE-2018-14847

The exploited vulnerability in this attack is CVE-2018-14847 and affected MikroTik RouterOS through version 6.42, allowing *"unauthenticated remote attackers to read arbitrary files and remote authenticated attackers to write arbitrary files due to a directory traversal vulnerability in the WinBox interface."* [26]. Of special significance to the attack is that MikroTik uses RouterOS across their entire product line, making the vulnerability applicable to a large number of both consumer and carrier-grade routers. As we will see later, the vulnerability of carrier-grade devices explains the magnitude of cryptomining activity that could be realized in this attack.

WinBox is a small Win32 binary that allows for the administration of RouterOS using a graphical user interface. The functionalities of the WinBox interface are almost identical to the console functions, but some advanced and critical system configurations changes, like changing the MAC address, cannot be made from the WinBox GUI. Several WinBox commands did not require authentication, e.g., an attacker could open files for reading while being unauthenticated, while another allows an attacker to write files to disk given some authentication [38]. By sending a carefully crafted package to the WinBox service on port 8291 exploiting one of these commands, the attacker would retrieve the user credential store `user.dat`, and using these credentials drop files to disk to enable a developer backdoor [38]. Triggered if a specific file, `/pckg/option` or `flash/nova/etc/devel-login`, is present on the system, the developer mode sets up a root BusyBox shell accessible over port 22 (SSH) or 23 (Telnet) giving complete control over the device.

## 4 DATASETS

The study was made possible through a combination of three datasets each covering a different angle of the reported malicious activity: first, we use the traces from a large network telescope to trace adversarial scanning activity. Second, we rely on a periodic crawl for the proxy status page by Censys [10] and Shodan [35] to discover which routers were infected. And third, we use NetFlow data to visualize the communication patterns between the infected routers and the remaining Internet to identify their staging hosts and quantify the volume and revenue of this large scale exploitation.

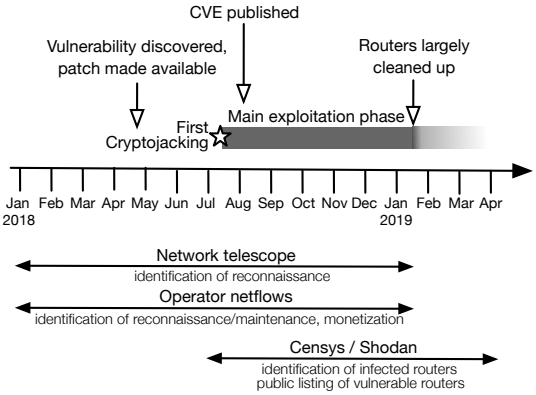


**Figure 3: Timeline of the cryptojacking campaigns and the used datasets.**

Figure 3 shows a timeline of the main phases of the cryptojacking exploitation of MikroTik routers, together with the timeline and purpose of the used datasets. While the vulnerability was discovered in April 2018, the MikroTik routers were only exploited for MITM cryptojacking from middle of July onwards, until in January 2019 the bulk of the ecosystem was cleaned up. We use telescope traffic and operator NetFlows already months prior to the abuse from January 2018 onwards to observe prior knowledge of the vulnerability and any preparation activities by the adversaries as discussed in [3, 13], to trace the activities of actors in finding these devices as well as to identify their installation, maintenance and monetization strategies. After the monetization through MITM cryptojacking has emerged in July, we then follow the state of the compromised devices through the public lists Censys and Shodan until the general wind-down of these campaigns. Details about each dataset are presented below, table 1 lists all the datasets collected within this analysis, and links where they are used later for the analysis in sections 5 and 6.

### 4.1 Network Telescope

In order to exploit routers using the WinBox vulnerability, the attacker must first know where vulnerable routers are located. This identification and localization could be done in one of two ways: either the adversary scans the Internet for open ports or banners that would identify the devices, or obtains a list of devices.

To discover which adversaries are actively scanning the Internet for devices with the WinBox vulnerability, we rely on a large network telescope of three partially populated /16 networks, through which a total of approximately 130K dark IP addresses are monitored. In order to discover whether TCP port 8291 is open and to send a payload triggering CVE-2018-14847, adversaries first need to complete a TCP handshake. This ensures that perpetrators cannot spoof their source IP as otherwise the handshake couldn't complete, and reveals the location of the adversary or a potential proxy. The telescope collected approximately 21.7 TB between January 2018 until January 2019, out of which only the small part of 1.6 GB were probes on port 8291. The size of the used telescope provides tight approximations of network activity estimations as shown in [4].

**Table 1: Summary of the datasets and their usage in this analysis.**

| Dataset | Time frame | Size | Usage in analysis |
|---|---|---|---|
| Telescope | Jan 2018 – Jan 2019 | 1.6 GB | Adversarial identification through port scanning (5.1) |
| Censys | Jul 2018 – Apr 2019 | 43 GB | Adversarial targeted scanning (5.1), Infections and re-infections (5.2), System architecture (5.3), Monetization configuration (5.4), Ecosystem (6) |
| Shodan | Jul 2018 – Apr 2019 | 236 GB | Adversarial use of public datasets (5.1), System architecture (5.3) |
| NetFlows | Jan 2018 – Jan 2019 | 3.2 TB | Characterization of port scanning (5.1), System architecture (5.3), Evolution of monetization (5.4), Maintenance patterns (5.5), Revenue and ecosystem (6) |

## 4.2 Active Scans of Censys and Shodan

In addition to Section 3, the exploitation through the rewriting proxy was unusual as it unnecessarily exposed the web page to the Internet instead of just presenting it to the users on the inside. Since RouterOS allows both port 80 and port 8080 to be used by a HTTP proxy, an Internet-wide survey of these ports made it possible to discover which MikroTik routers are currently infected as they are serving the proxy page, and based on the embedded *siteKey* track who currently "owns" the device.

*Censys.* To trace infections and their evolution, we thus rely on Censys [10], which scans and archives the responses of all IPv4 addresses on a number of common ports, among them 8080 and 80. As the vulnerability became exploited for cryptomining in July 2018, we retrieved these Internet surveys twice a week from July 2018 until the end of the study in April 2019. We identify a router as a MikroTik system if the proxy header was set to `MikroTik HttpProxy` and mark it as infected if it contained scripts or code for crytomining. The regular expressions used for this detection step are listed in Appendix A, and resulted in a dataset of 43 GB. This yielded a total of 1,452,550 unique IPs belonging to an infected router at some point during the study.

*Shodan.* A second service that scans devices for open ports is Shodan [35]. Besides listing ports, the service additionally extracts banners to link it with known vulnerabilities, and makes it possible to conveniently search specific devices and credentials. Given the Internet surveys of Censys, we queried the databases of Shodan and recorded when a particular IP that could be identified as compromised due to the HTTP proxy page including a cryptomining script appeared in Shodan's database. Therefore, we queried the host information endpoint of Shodan's API with the history flag enabled and searched for the timestamp when Shodan encountered the open proxy ports for the first time in their crawls and listed them with the annotation `mikrotik` or `routeros` in their public search results. For the 1.4M routers, this dataset of historical open ports and services comprised of 236 GB of records.

## 4.3 Operator NetFlows

While the aforementioned datasets provide insights into vulnerable devices and which routers are exploited at a given moment, these data sources do not reveal anything about the scale of the operation and how concretely the infrastructure is managed and controlled. In order to fill this gap, we analyzed NetFlows from the network of a Tier 1 operator between January 2018 and January 2019, which were collected at a 1:8192 sampling ratio at each of their edge routers. For the quantification of traffic volumes in Section 6, the flow aggregates were scaled up by this sampling ratio.

*Anonymization.* While the IP addresses of vulnerable MikroTik devices are public knowledge as they appear in both Censys and Shodan, we need to ensure the privacy of users and their traffic during our study. For our analysis, we obtained NetFlow records for all connections from or to the 1.4M infected MikroTik routers in a tuple consisting of time, source and destination addresses and ports, as well as packet size, which allowed us to investigate when and how the routers made connections. The identity of the other endpoint is however irrelevant, and was anonymized to a pseudo-random value. For this, the operator applied the CryptoPan algorithm [43] to the remote points of the NetFlows, which does a prefix-preserving deterministic randomization of IPv4 addresses based on AES as a source of randomness. The algorithm was proven to be semantically secure by Xu et al. [43] and the key to the data randomization remained with the Tier 1 operator. The procedure was developed in collaboration with and approved by relevant departments of the operator. This protocol will thus allow an analysis whether devices connecting and controlling the vulnerable routers are located for example in the same /24 network, but not which one. We can furthermore investigate whether there are specific anonymized IP addresses that connect to multiple vulnerable or infected routers to do exploitation or quantify the amount of hijacked flows due to source/destination port combinations, but cannot tell the identity of these devices nor the destinations visited by the victims. In order to help the presentation of the results and elaboration on certain strategies and patterns, the subsequent discussion will include anonymized IP addresses, however these do not allow any inferences on networks except that addresses in the same netblock – for example a /24 – were also in the same subnet in the original trace. Whenever we use an anonymized IP address in the text, it will be printed in *italic*, while the publicly known and thus unanonymized IP address of an infected router would be shown in regular font.

# 5 ADVERSARIAL TECHNIQUES, TACTICS AND PROCEDURES

In this section, we analyze the techniques, tactics and procedures (TTPs) adversaries use in the exploitation of 1.4M MikroTik routers and their subsequent abuse. We will split this discussion based on the stages in the life cycle of a router infection as shown in Figure 4. This life cycle begins with the identification of candidate victims, the exploitation of the vulnerability, and methods used to gain a foothold and consolidate the infection. After a device is compromised, actors will install tools to monetize the exploited routers and perform maintenance, until the infected system is removed from the pool due to decommissioning or patching. As we will see in this section, each of the individual steps can be accomplished in a variety of ways, and we find adversaries using different techniques

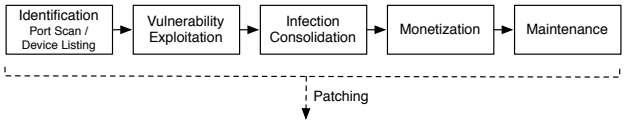


**Figure 4: Life cycle of the vulnerable routers.**

**Table 2: Top 10 most affected Autonomous Systems (AS).**

| AS | Count (%) | AS | Count (%) |
|---|---|---|---|
| Telekomunikasi Indonesia | 55,082 (3.8%) | Cat Telecom | 12,883 (0.9%) |
| Telefonica Brasil S.A | 33,589 (2.3%) | Rostelecom-AS | 11,352 (0.8%) |
| TCI | 21,357 (1.5%) | TOT-NET | 11,136 (0.8%) |
| PTC-Yemennet | 13,585 (0.9%) | UKRTELNET | 10,993 (0.8%) |
| BSNL-NIB | 13,046 (0.9%) | IR-THR-PTE | 9,248 (0.6%) |

and tooling in each of the life cycle phases. In Section 6, these findings on the individual stages will be combined into an overview of the actor landscape.

## 5.1 Identification

In order to gain a foothold on a machine, adversaries first need to know where exploitable devices are located. This also holds for vulnerable MikroTik routers, of which according to market surveys approximately 2M units were installed worldwide [33]. Routers are usually deployed in one of three ways on the Internet: (a) they are either provided by the Internet Service Provider (ISP) to the customer who uses the device to connect to the ISP's network, (b) they are bought, deployed and operated by the customer to connect to the Internet, or (c) they are part of the network infrastructure of the ISP. As RouterOS was used across the entire MikroTik product line, we see vulnerable devices of all three types in practice.

Figure 6 shows a heatmap of all MikroTik routers that were exploited at least once during the study period, mapped to a geographic location by using the MaxMind GeoIP database [19]. The devices are very prevalent in select parts of the world, especially Brazil or Indonesia, where such a device responded at 29%, and 35% of all publicly accessible IP addresses of the largest operators in these countries, thereby indicating that these devices were provided by the ISP to the customers. Table 2 lists the number of compromised MikroTik routers for the 10 most affected autonomous systems and their share of the overall infected population. We can see that 136,659 exploited MikroTik routers could be linked back to the 5 most compromised ISPs. The heatmap however also shows sparse deployments throughout the world, with clusters appearing in densely populated areas, proportionally to the number of IP addresses located in an area, suggesting that these routers were owned and operated by end customers.

*5.1.1 Discovery using Port Scanning.* To localize potential victims, adversaries could make use of port scanning to test remote IPs whether they have TCP port 8291, the port associated with the WinBox vulnerability, open. This reconnaissance could be done at different levels of granularity and sophistication: on the low end, attackers could blindly trawl through the entire Internet in a horizontal port scan to discover any potential victim, albeit at the disadvantage of creating much noise and potentially being

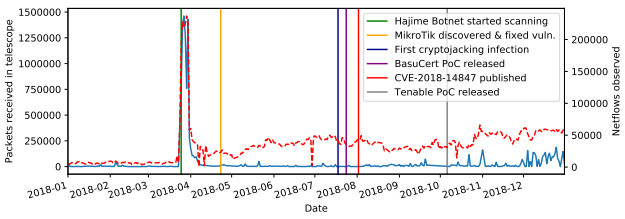


**Figure 5: Packets received on port 8291 in our network telescope (in solid blue) and NetFlows observed (in dashed red).**

identified, blocked and blacklisted. A sophisticated scanner could however do some prior background research, and determine in which networks large MikroTik installations exist, as a result of these devices being used within an ISP's network or being given out to its customers.

We can differentiate between these type of strategies using the data provided by the network telescope and general flow statistics of the Tier 1 operator. Figure 5 shows the absolute number of packets directed against port 8291 in our telescope as well as traffic carried by the operator during 2018 aggregated by day. The vertical lines show important milestones in the lifespan and news coverage of the exploited vulnerability. On March 24, the average daily traffic towards TCP 8291 exploded by 6 orders of magnitude, as the Hajime botnet executed a short, but concentrated horizontal scan for the port across the Internet [25]. On April 23, the vulnerability was discovered and patched by MikroTik, and the resulting news coverage only lead to a very minor continuous increase in scanning traffic. This is interesting, as for example the media reporting around the memcached DDoS vulnerability in early 2018 led to a major influx of actors and probing activity [13]. Starting mid-July, the first cryptojacking installations started to appear in the wild, followed by a public proof-of-concept for the exploit. Finally, in the beginning of August the CVE report was published in the National Vulnerability Database [26].

As we can see from the graph, the general characteristics of telescope and NetFlow traffic resemble each other. Both record the same sudden increase in network traffic due to the Hajime botnet at the same moment and with a similar magnitude, demonstrating that the botnet initiated an unspecific worldwide trawl for the vulnerability. While after this burst the telescope traffic returns to business-as-usual, aside from selected worldwide scans, we see in the NetFlow data that geographically targeted scans – not targeting our network telescope – immediately followed, and continued to run until the end of the observation period. As the number of infections started to rise in December 2019, we observe increased worldwide scanning activity as both our telescope and NetFlow data report more connections towards port 8291.

Out of a total of 1.7M IP addresses that probed the three /16 network ranges in our telescope as well as the rest of the Internet during the late March burst, only 124K IPs continued to probe specific parts of the Internet for router vulnerabilities. This seems to indicate that the scanners used the data collected from previous tests (as our passive monitors would not respond to 8291), or that

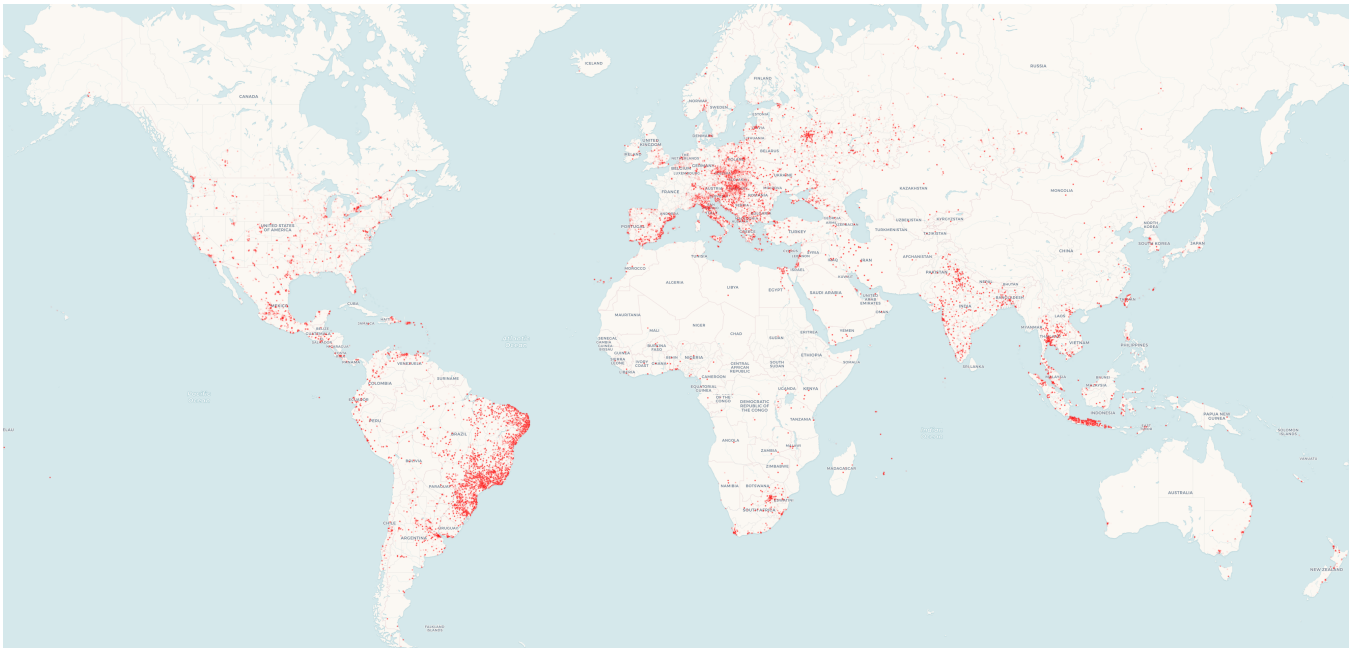

**Figure 6: Geographical location of the MikroTik routers compromised during the study period.**

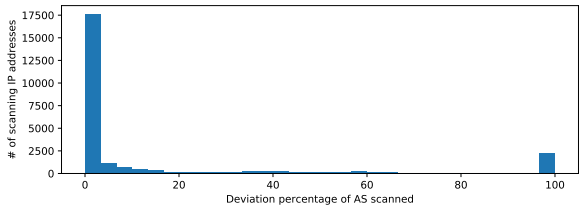


**Figure 7: Histogram of the specificity of scans for port 8291.**

additional knowledge – such as the popularity of MikroTik in specific parts of the world – is used to steer the search. In order to determine the specificity of these scanners, we compared the traffic distributions of the Tier 1 operator towards all autonomous systems (AS) with the traffic distribution for the anonymized scanning source IP addresses. This relative comparison accounted for the fact that the operator would not be part of an exact random sampling of all worldwide traffic flows, but that due to BGP policies and specific IXP and PoP presences certain autonomous systems would be preferred. From this relative comparison we can determine whether sources showed specific preferences for select networks, or scanned the Internet non-discriminantly. Figure 7 shows a summary of all scanners as a histogram of the scanners' deviation from the expected non-discriminatory baseline. As we can see in the graph, there exist three basic behaviors: the bulk – which is also visible in our telescope – targets the entire Internet unspecifically, a smaller but significantly sized group that specializes and concentrates the scan on a specific AS, while a small portion of adversaries scan a large but apparently curated list of destinations.

*5.1.2 Localization using Public Datasets.* In addition to actively scan and probe IPs on the Internet to test whether they are running RouterOS and are potentially exploitable, attackers could try to get a pre-made list of device IPs to connect to potential targets directly, for example by searching on Shodan. To determine whether the attacker uses such services to locate vulnerable routers, we consider the moment Censys retrieved a proxy page from a router with a mining *siteKey* on port 8080 or 80, which means that at this moment the device was compromised. If at that moment the router was not yet listed in Shodan, the perpetrator must have found the vulnerable router by independently scanning for it. If prior to the Censys publication, there already existed a record in Shodan, the attacker could have obtained knowledge from this service.

When we track this relationship for every *siteKey* on the date it first appeared on the 1.4M routers, we find that 54% of the cases a new *siteKey* is installed on routers that were already listed in Shodan, whereas 29% of the new installations were derived from independent scanning. In the rest of the cases, too few routers were compromised with the same *siteKey* to significantly categorize them. Figure 8 shows the percentage of unlisted routers used by actors within the first 14 days of their activity. We clearly see two regimes. Innovators and early adopters such as *d68a7a* and *hsFAjj* which are shown as dashed lines (for *siteKey* emergence see Figure 13) all perform their own discovery, and start off with a high number of new, unlisted routers. This percentage drops over time, as the compromised devices are then included in Shodan. The long lasting campaign *tD2a2P* starts out with 42% unlisted routers on its first, and kept adding unknown devices to its installed base for the months to come. On the other hand, we find a large number of campaigns which primarily feed off public lists to populate their setups. One of the most profitable campaigns *6a9929* had at its peak

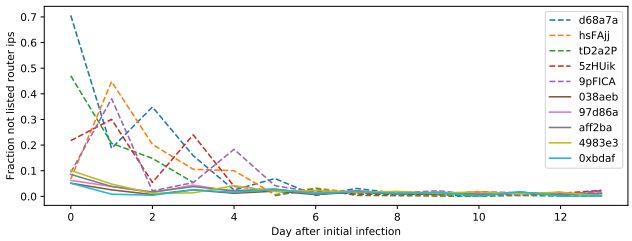


**Figure 8: Percentage of infected unlisted routers per key.**

13,815 routers infected simultaneously, almost exclusively drawn from public lists. As we will see in Section 6, the degree of innovation is not a proxy for the amount of revenue these campaigns make – innovation does not always seem to pay off. To summarize this phase in the life cycle, there is a wide variance of MikroTik spreading over the world, we have seen a steadily increasing interest in scanning of port 8291 throughout 2018 and half of the newly installed *siteKeys* are installed on a router that was already listed in Shodan, whereas only 29% of all new installations was the result of independent scanning by the adversary.

## 5.2 Vulnerability Exploitation

With the vulnerable routers identified, adversaries can trigger the vulnerability by sending a simple payload, as discussed in Section 3.3. While the activities of the perpetrators on the devices cannot be inferred using our datasets, we can investigate patterns on how adversaries infect devices, and how infected devices are taken over.

*5.2.1 Infections and Reinfections.* From previous discussion, we have seen in the NetFlow data and our telescope that a large number of IPs scanned for port 8291, and that actors additionally used records such as Shodan to find exploitable targets. Once a device however appears in Shodan, it could already be infected, due to a proxy service running on port 80 or 8080. This naturally raises the question whether and how reinfections occur, in other words whether actors are grabbing compromised devices from others or are updating *siteKeys* on routers they already "own".

Figure 10 depicts the transition behavior of the 1.4M routers between *siteKeys*, filtered to only include edges if more than 500 devices are taken over from the original "owner" by a particular new actor. The size of the circle is the number of routers which transition away from this *siteKey*, the thickness of the arrow and the color of a circle are the number of routers that newly infected with a particular key. Since we can not retrieve the persona behind a *siteKey*, we assume in this section that every *siteKey* is a different persona. However, as we reveal in Section 6.2, we have strong suspicions that this is not the case. In the figure, we see two types of transition behaviors. First, we see *siteKeys* which draw their installation base from pools of already infected routers. An example of this is *4983e3* which relies on lists of infected devices and then reinfects them with a new *siteKey*, which we could already infer for this *siteKey* from Figure 8. Second, we see *siteKeys* on routers being replaced in a specific sequence by another account. When the routers of different keys before and after the update share

```
<script src="https://xmr.omine.org/assets/v7.js"></script>
<script>OMINEId(\"4983e34ef01b4b579725b3a228e59e79\",\"-1\");
throttleMiner=10; </script>

<script src="https://xmr.omine.org/assets/v7.js"></script>
<script>var _0xdafb=['\\x34\\x39\\x38\\x33\\x65\\x33\\x34\\x65\\x66\\x30\\x31
\\x62\\x34\\x62\\x35\\x37\\x39\\x37\\x32\\x35\\x62\\x33\\x61\\x32\\x32\\x38
\\x65\\x35\\x39\\x65\\x37\\x39'];
OMINEId(_0xbdaf('0x0'),'\\x2d\\x31');throttleMiner=0xa;</script>
```

**Figure 9: The original Omine infection on top, the obfuscated variant listed on the bottom.**

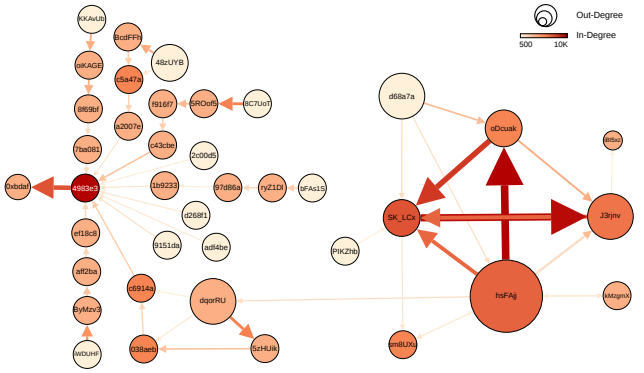


**Figure 10: Reinfections of compromised devices with different keys with >500 overlapping IPs.**

properties, for example they are accessed by the same person or share infrastructure components, we can conclude that these are examples where an actor is performing a key rotation system. The sequence from *iWDUFD* to *ByMzv3* to *aff2ba* to *ef18c8* shows an example of such a transition, which is visible as all routers in a node leave towards the same destination as can be seen in the identical color of node and arrow. A special form of these updates also appears in the application of obfuscation techniques. For example, the routers infected by the *siteKey 4983e3* were at some point being updated to the iframe code shown in Figure 9. The obfuscated Javascript expression *0xbdaf('0x0')* however decodes to *4983e3*, so here the transition is an evolution in technique rather than a progression of accounts. Account rotations however do not necessarily occur in chains, an example being the cluster on the right, where the routers originally mining for *hsFAjj* transition towards *SK_LCx* and *oDcuak*, and where a little over 15K routers shift back and forth between the receiving *siteKeys*. While the reinfection graph only displays the largest transitions for readability, there is a lot of change happening, especially in the long tail of the distribution. Overall, 55% of all routers are infected with more than one key, and 15% of all MikroTik devices had 5 or more *siteKeys* in 2018.

## 5.3 Infection Consolidation

After the adversary has been able to obtain the system credentials and activate the developer backdoor, root access is used to establish a foothold on the device. As described in Section 3, the firewall configuration is changed, the proxy activated, and additional files

downloaded to the system. We defer a discussion on the monetization, the cryptomining, to the next section, and discuss the infrastructure used to perform the scanning, logins and loading of additional components.

*5.3.1 Node to node reconnaissance.* Based on Censys and Shodan data we obtained a list of infected devices over time, and could in the NetFlows thus trace which anonymized IP addresses would connect to the WinBox service on vulnerable and infected routers. While the bulk of these connections came from a variety of anonymized IPs, 6.5% of the flows towards port 8291 were sent from infected MikroTik routers to other MikroTik routers. We observed 948 infected routers which were systematically scanning their local subnet for additional vulnerable routers on port 8291. While based on NetFlows it is not clear whether these infected routers only enumerate vulnerable hosts or also perform the compromise itself, we find this additional structural component noteworthy. Interestingly this behavior was only implemented in geographic regions where MikroTik routers seemed to be rolled out structurally by ISPs as we observed this behavior specifically in Brazil.

*5.3.2 Infrastructure.* In August 2018, the first router infections spread throughout Brazil and were under the control of a sophisticated adversary. After successfully locating vulnerable MikroTik devices, it exploited the WinBox vulnerability and injected both a miner into the HTTP proxy page and installed a script which would fetch new updates and commands from a *staging server* on port 2008 every 30 seconds. These updates could involve changes in miner service or a new *siteKey*. Using our NetFlow data we have identified six of these staging servers in the subnet of *211.164.222.**, which confirms the research of [37]. We have identified that these staging servers are active from 26 July to 21 September 2018, and these servers have connected to 220 distinct infected routers during this period. The most prominent *siteKey* involved in making these connections was *hsFAjj*. However our NetFlow data also shows that *SK_LCx* and *oDcuak*-infected devices begin to make contact to these servers towards the end of this period, suggesting a link between these *siteKeys*.

Based on the connection patterns of the compromised routers and the maintenance activities (which we discuss in Section 5.5), we can deduct the system architecture as depicted in Figure 11. While a handful of infected routers are performing scanning and infections within the same prefix, compromised routers remain unconnected among themselves. They only have two flows in common: the connection on port 2008 to a handful of staging or Command & Control (C&C) servers, as well as SSH flows on port 22 from a shared origin. When a router is taken over, the new perpetrator does not seem to always aim to eradicate a previous infection after having replaced the proxy template and *siteKey*. In fact, we find numerous examples where the routers taken over by a different *siteKey* keep beaconing to staging servers associated with an unrelated actor, who shows no other commonalities or features with the new owner.

## 5.4 Monetization

With the vulnerability triggered and a foothold on the routers established, the adversaries moved to the exploitation of the routers

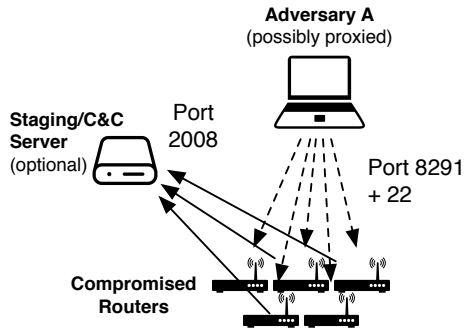


**Figure 11: Schematic overview of the system architecture.**

for monetary gain. Over the course of the study period we observed the evolution of two monetization strategies. First, the use of the routers as a (free) proxy service, and second, the injection of cryptomining code into users' web browsing sessions.

*5.4.1 HTTP Proxies.* The first use case of the compromised MikroTik routers was the establishment of HTTP proxies. Here traffic from a web browser to a web server is tunneled through the HTTP proxy, thus masking the IP address of the client towards the server. HTTP proxies are used as a basic variant of a VPN service, although application-protocol specific and with limited authentication options if at all implemented. Starting from July 9, 2018, the first MikroTik routers were repurposed as HTTP proxies, which we identified from the emergence of large incoming traffic towards specific high TCP ports, namely 36551, 53281 and 58833. This use case remained however relatively rare, with only 3,216 of the total 1.4M infected routers being abused in this way.

Interestingly, the usage as an HTTP proxy did not seem to serve a monetary gain, as within 3 days 95% of the routers for which these unusual spikes appeared were posted to free public proxy lists [29], and allowed a connection without user credentials. This usage was only relatively short-lived, as most were disabled within 40 days, at which point SOCKS proxies were spun up at TCP 4145.

*5.4.2 SOCKS Proxies.* In contrast to HTTP proxies, SOCKS proxies work at the transport layer and forward traffic transparently with regard to the application layer protocol. This allows this proxy type to be used in combination with any application and thus extending the monetization potential. Shortly after the emergence of this new use case, the HTTP proxies on the MikroTik routers are replaced by SOCKS proxies, and 1,530 MikroTik routers remained in use as SOCKS proxies even until the end of the study. Further characterization of the NetFlows is not possible, as the application traffic itself would be forwarded inside the tunnel and the router would rewrite the outgoing flow to an ephemeral source port. However, we do find that the exploitation as SOCKS proxy was under the control of a few and not deployed pervasively.

This is possible to conclude, as the use of SOCKS proxies was never encountered alone, but only in combination with a cryptomining infection. As we discussed in the previous section, adversaries were routinely reinfecting devices and by changing the cryptomining *siteKeys* effectively snatching the devices away from their

**Table 3: Router "ownership" based on cryptomining *siteKey* and corresponding SOCKS proxy activity**

| SiteKey | *hsFAjj* | *J3rjnv* | *SK_LCx* | *oDcuak* | *d68a7a* |
|---|---|---|---|---|---|
| **% of all SOCKS traffic** | 53.6% | 29.1% | 7.8% | 3.4% | 1.2% |

competitors. With the infection script reconfiguring the device including firewall and proxy settings, we can thus assess that the "ownership" with respect to an active cryptomining would also indicate who had control over the SOCKS proxy at that point in time. As we discuss in the next section, we identified a total of 140 cryptomining keys on the 1.4M MikroTik routers, but as shown in Table 3, only five *siteKeys* were in use on a router whenever the device was proxying traffic. Their impact is however huge: more than 95% of all SOCKS activity that originates from MikroTik routers is the result of proxies operated by these five *siteKeys*, with *hsFAjj* being one of the early adopters of MITM-based cryptomining. The small number of *siteKeys* related to SOCKS proxy activity suggests a relation between those *siteKeys*, as others do not exhibit this behavior.

*5.4.3 Cryptojacking Proxies.* While the usage as HTTP proxies was not commercialized and only few actors repurposed a limited number of devices as SOCKS proxies, a large number of actors engaged in cryptojacking user connections, with a total of 140 different cryptomining *siteKeys* being installed on the routers during the study, and a maximum of 106 different *siteKeys* being active at the same time.

*Mining services.* Figure 12 depicts the number of infected routers over time, categorized by the mining service provider used. As we see in the figure, the MITM-based mining started out based on Coinhive, which was at that time the obvious choice to be introduced in the MITM vector as it was the first service for cryptomining and already widely deployed in website-based mining [2, 18, 30]. Starting in middle of September, this homogeneity shattered with first the emergence of CoinImp, and later on Omine, all taking on approximately equal market shares which led to a peak of cryptojacking activity on December 19, 2018, as 460,618 routers were infected concurrently. This activity continues relatively unchanged until January 26, 2019, when suddenly mining activity disappeared from the bulk of infected routers. The distribution of miner applications between Coinhive, CoinImp and Omine remained similar and relatively unchanged.

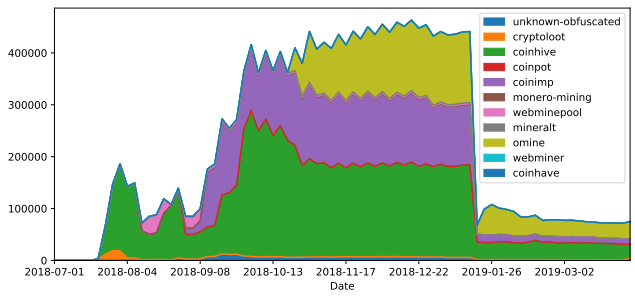


**Figure 12: Evolution of infections over time per service.**

**Table 4: Top 10 largest campaigns identified.**

| SiteKey | Service | Total Infected | Max Concurrently | Date first seen |
|---|---|---|---|---|
| *hsFAjj* | Coinhive | 223,844 | 167,182 | Jul 21, 2018 |
| *4983e3* | Omine | 117,502 | 64,539 | Nov 3, 2018 |
| *f6c7f3* | Omine | 102,241 | 36,059 | Jan 23, 2019 |
| *tD2a2P* | Coinhive | 71,513 | 61,835 | Aug 22, 2018 |
| *oDcuak* | Coinhive | 55,437 | 47,310 | Aug 1, 2018 |
| *48zUYB* | Coinhive | 52,181 | 26,122 | Sep 12, 2018 |
| *dqorRU* | Coinhive | 50,566 | 27,808 | Sep 15, 2018 |
| *9pFICA* | Coinhive | 50,376 | 25,928 | Sep 15, 2018 |
| *BOvlp3* | Coinhive | 49,640 | 22,921 | Sep 15, 2018 |
| *8C7UoT* | Coinhive | 47,981 | 24,773 | Sep 15, 2018 |
| **Total** | | 1,452,550 | 460,618 | |

Interestingly, *siteKeys* found as related in previous analyses do not necessarily use the same mining application, possibly to avert risks from accounts becoming frozen by an individual cryptomining service. Despite risk being shared across providers, several actors also spread out their activities across multiple *siteKeys*, as can be inferred when the same maintenance hosts connect to routers with multiple *siteKeys* (as we will show in Section 5.5). These movements between providers and the market shares of CoinImp and Omine might also be explainable based on fees: while Omine charges a 2% fee and CoinImp is entirely free, Coinhive takes a 30% cut.

*Evolution of siteKeys.* Figure 13 shows the evolution of *siteKeys* installed on MikroTik routers between July 2018 and April 2019, ordered by the time they were first encountered on a router. The size of circle indicates on how many routers this *siteKey* was installed on a given day. We can see that MITM-based cryptomining was pioneered by three *siteKeys*: first, *d68a7a* emerged first but beside a small peak remained only a minor player. Second, *hsFAjj* who followed one week after, temporarily controlled 70% of all infected routers, and introduced new strategies for controlling and otherwise monetizing the routers, remaining a steady force until the general decline. And third, *oDcuak*, like the first mover *d68a7a* experiencing a small surge followed by steady but comparatively low-volume activity.

Approximately one week after these first movers, a large number of new *siteKeys* started to appear, frequently co-emerging in groups that stay relatively similar in size. Four sequential blocks of 10 *siteKeys* can be clearly observed, two of them using Coinhive, the other two are using Omine as their mining service. While all other *siteKeys* never reach the same size as *hsFajj*'s initial deployment (167,182 infections), each of them is able to hold control over up to 64,539 routers at a time. While we find a total of 1.4M routers to be vulnerable and at some point infected, the perpetrators are never rolling their cryptojacking infections out to all potential victims simultaneously. Instead, we see a constant flux, with new routers being infected so that the mining deployments stay consistent in size. This is necessary, because once infected, most of the routers are patched quickly, as shown in Figure 15. This figure shows the cumulative density function (CDF) of the number of days a router is infected on a logarithmic scale. We see that 50% of the devices are patched within 18 days after compromise, whereas only 30% of the devices remain active for more than 50 days, urging actors to constantly replace disappearing routers to maintain their installation base.

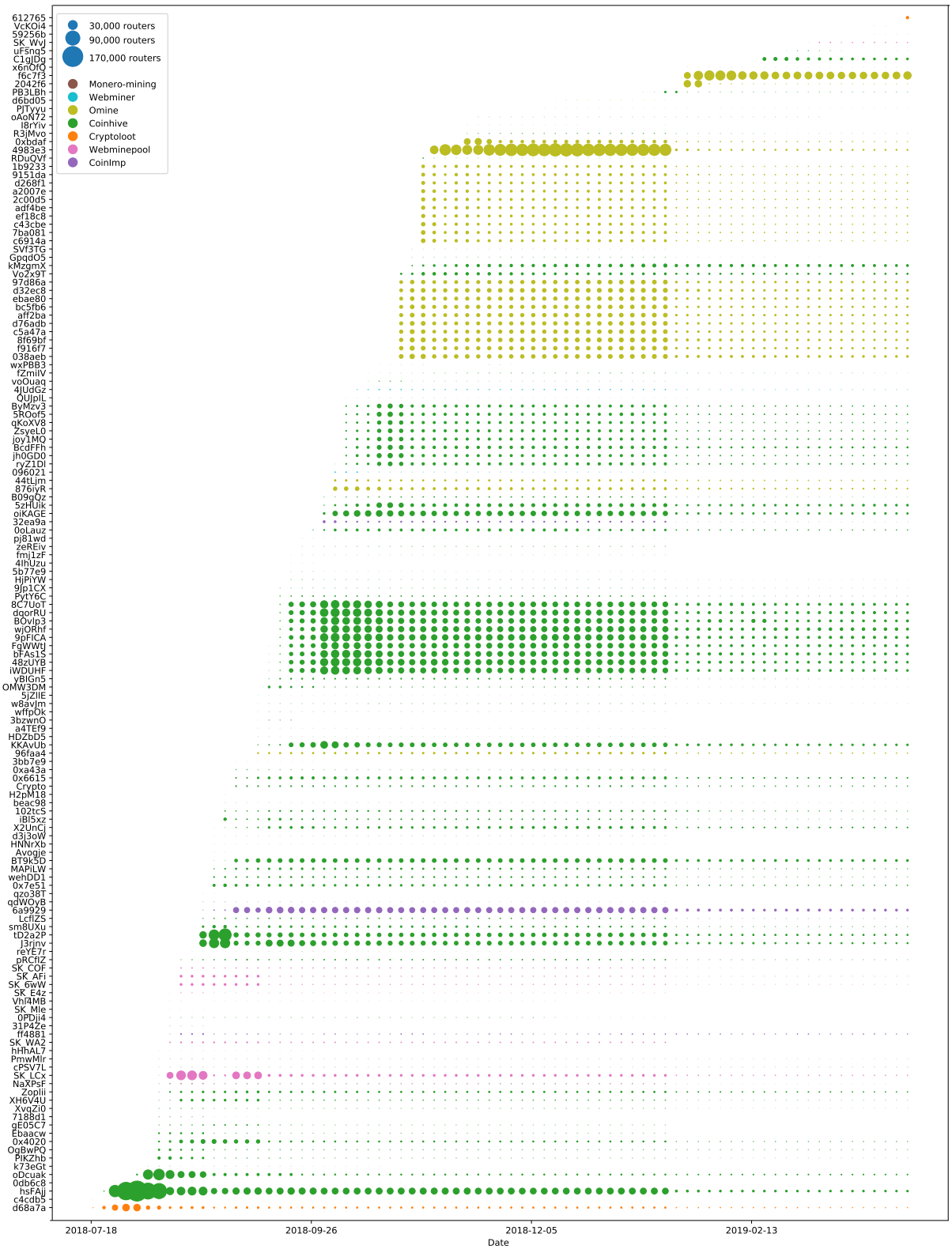


**Figure 13: Evolution of detected *siteKeys* over time, colored per mining service.**

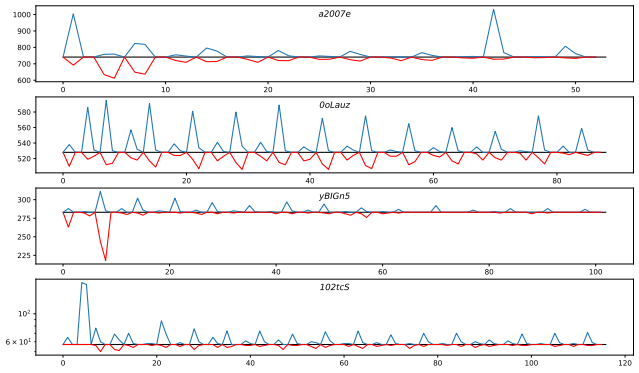


**Figure 14: Additions/deletions over time per *siteKey*.**

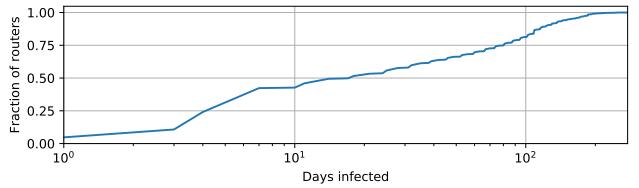


**Figure 15: CDF of the infection duration per IP address.**

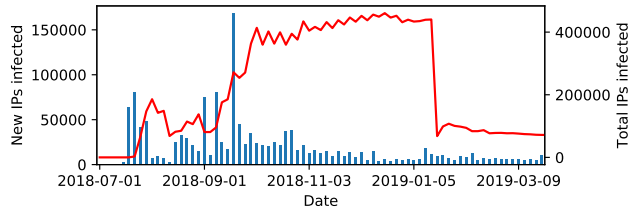


**Figure 16: New and total of IP addresses infected per day.**

This is best observed when we look at the *siteKeys* in Figure 13 that remain relatively constant over time. Four of these *siteKeys* are depicted in Figure 14 with respect to daily additions and removals from the pool, indicated in blue and red respectively, starting from the day the *siteKey* first became active. This behavior, as well as the sets of *siteKeys* that appear together, might indicate a strategy to offset risk. If a particular *siteKey* gets blocked by a mining service, others will still generate profits. The same might hold for the deployment size in general, where an all-out operation from becoming too greedy could lead to increased press coverage and faster cleanup of the vulnerability than maintaining a smaller infection size and thus lower profile. This diversification however stops from December 2018 onwards, where we see that most actors no longer replenish routers lost. This might be explained by Monero's significant drop in value, which decreased by 60% from early November until a month later.

*A sudden drop in mining activity.* Indeed, we observe a steady decline of new devices that are added to the pool from November 2018 onwards, as shown in the bottom of Figure 16, which leads to a flattening out of the overall installation base. As we have already seen in Figure 12, the ecosystem of router-based cryptomining drastically changes in late January. Most apparent is the major drop in participating devices, approximately 87% of all infected routers disappear, which affects the installation base of all *siteKeys* across all autonomous systems and countries. While such a large and universal movement would indicate some external trigger or cause, we could not find any evidence for a coordinated cleanup action, for example by an ISP or a grey hat hacker (aside from one who has taken credit for patching 100,000 routers in November 2018 [7]). Additionally, we contacted Censys whether they had made any changes to their crawling strategy, but that was not the case. After the sudden cut, we also see a rotation of remaining actors towards new *siteKeys*, where the new *siteKey f6c73* partially takes over the efforts of *4983e3*, however only a few continue to re-establish their activities and forego previous practices, whereas *f6c73* is responsible for most new infections.

*Geographical Focus.* Based on the heatmap in Figure 6 and the large deployment of MikroTik devices in certain autonomous systems as shown in Table 2, we have seen that a number of countries seemed prime candidates when looking for MikroTik devices, which would logically mean that advanced adversaries should focus their activities there. As RouterOS is used in both consumer devices and carrier-grade routers, we would naturally expect some devices to be more lucrative than others, immediately posing the question whether reinfection of devices – in other words "stealing" routers – would primarily occur in popular areas and target those devices where a lot of money could be made.

Figure 17 shows the number of *siteKeys* as a function of the amount of NetFlows on port 80 this router processed during its infection. Counterintuitively, there is no trend that high-value targets are more fought over than low-value ones. Especially routers with much traffic tend to stick with just a low number of *siteKeys*. This is surprising, as a cryptomining operation on a large router would clearly affect more people, lead to more complaints and thus logically faster patching. The lack of a fight for high-grossing routers can however partially be explained based on the location of the routers, indicated by the color of the data point. While routers in Indonesia and Brazil – the hotspots of the infection – cover the entire spectrum and are changing keys considerably, the most stable infections – and the highest grossing ones for that matter as we will see in Section 6 – are in countries that do not appear anywhere near the top in MikroTik installation counts, for instance, 6 out of the 10 most grossing routers are located in Iraq. This means that actors targeting niche markets accomplished much more valuable deployments, as these routers mined longer for them.

## 5.5 Maintenance

When we look at the life cycle of a malware infection as for example a botnet, after the initial exploitation the compromised device remains in contact with the perpetrator or a C&C server to download additional components or receive a new configuration. While we would expect a similar behavior for malware targeting routers, we saw little evidence for post-compromise maintenance operations.

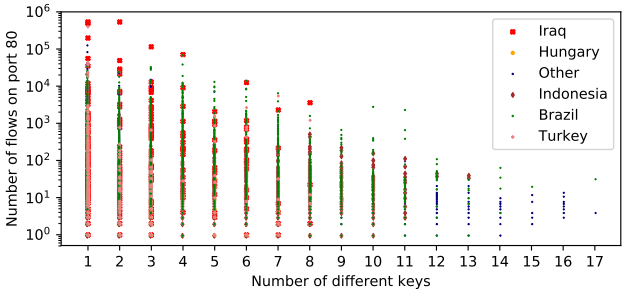


**Figure 17: Relation between the number of flows to port 80 and the number of keys per router.**

*5.5.1 Configuration Access and Periodic Updates.* As a *siteKey* is directly linked to a particular actor, we analyzed whether any connections were made between an end point and the group of routers that were at a certain moment compromised by the same key. Using the association rules methodology described by Agrawal & Srikant [1], we have searched for maintenance patterns where specific keys have a large probability to coincide with a specific anonymized IP address or port number, as maintenance would likely be performed from a set of C&C servers or the attacker's PC. As connections to port 22 (SSH) and 23 (Telnet) in NetFlows are also caused by prevalent port scanning, we differentiate between port scanning and active SSH sessions in NetFlows based on the packet size and only include connections with a confidence $c$ and support $s$ of at least 40% among our router/key set, in other words we require that at least 40% of infected routers had been contacted by a common origin, while being significantly present in the data.

We have observed maintenance connections on port 22 (SSH), which was only pursued by the actor(s) responsible for routers infected with one of three keys *oDcuak*, *SK_LCx* and *hsFAjj*, while other strains and actors do not seem to deploy such coordinated access. Surprisingly, routers with any of these keys were in contact with the same remote host at a given moment in time, strongly suggesting that the keys were actually related to the same persona. In addition, when a new IP address appeared to make contact with the compromised devices, routers with all three keys were always contacted by the same source. For example, routers with these keys made SSH connections to *236.197.108.8* between 3 and 20 August 2018, while between 11 and 14 August 2018 these routers were contacted by *236.247.130.64*. Each of these IPs seemed to employ automation, contacting routers either at midnight or during the timeframe 16–19h. Besides these IPs, almost no evidence of scripted interactions between a controlling source and the infected routers has been found, which would be evident from a large number of connections being made at the same time, or sequentially within a short time period. In total we observed 5 IPs making such common connections over time, matching our earlier observation about the link between the aforementioned three keys as discussed in Section 5.4.2.

# 6 DISCUSSION

The analysis of the tactics, technique and procedures of the actors demonstrated different levels of sophistication. This section translates flow volume into a revenue estimation per campaign and finally we review the previous findings, and use them to describe the ecosystem of cryptojackers and their differences of sophistication.

## 6.1 Quantification of Revenue

The results from the previous sections already suggested that MITM-based cryptomining operates at an entirely different scale than previously reported attack vectors. This is due to three reasons:

(1) The volume of compromised entities is much higher. Instead of a few thousand websites [5, 15, 30, 31], here a total of 1.4M infected routers is involved. Instead of mining on the web browsers of the users who visit one of the select infected websites, the MITM attack vector through routers would greatly amplify earnings, as cryptomining is introduced into *any* web page visited by *any* user connected behind an infected router.
(2) MikroTik uses the vulnerable RouterOS on consumer grade and carrier-grade devices. A carrier-grade router will likely serve significant user populations, and thus within a short time amass large volumes of revenue.
(3) While 30% of all website-based cryptomining is removed within 15 days [15], we find that 30% of the MITM-based mining remains active for more than 50 days. Although also routers are often patched quickly, the pool of vulnerable devices is so large that it barely affects the installation base.

In this section we will extend the previous results towards a quantification of adversarial revenue per key using this new attack vector. Unfortunately, as we have shown in Section 5.4.3, most mining is deployed through a cryptojacking service, such as Coinhive or Omine. This prevents us from performing a similar analysis as Huang et al. [16] did, who queried the (Bitcoin) mining pools directly to estimate the profits of a campaign. In our analysis, only 3 of the 140 discovered *siteKeys* were mining directly in a mining pool. However, we can leverage our datasets and using to the method established by Konoth et al. [18] we will conduct a quantification for a direct comparison with website-based mining, but make some adjustments for the shifted attack vector. For their analysis, Konoth et al. built a three-step estimation model:

- *Estimation of monthly visitors and visit duration:* They estimate visitor count and the average time spent for the 1,705 sites they detected to be cryptojacking based on visitor statistics from SimilarWeb [36].
- *Average computing power of visitors in hash rate per second:* Cryptocurrency is mined during the visit on the website. They measure the hash rate of two desktop CPUs and 16 mobile devices, and determined an average rate of 40.5 and 14.56 per second, respectively. Afterwards, information on MineCryptoNight [21] is used to convert that to $XMR/s$.
- *Current value of cryptocurrency:* The overall mining power of the visitors is then mapped to and monetized in Monero cryptocurrency, which was valued at $\$253/XMR$ at that time. Based on this value, the top 10 grossing actors generated an overall revenue of some $41,000 per month.

**Table 5: Revenue estimation parameters**

| Parameter | Methodology in [18] | This study |
|---|---|---|
| Number of visitors | SimilarWeb estimations | # of NetFlows on port 80 |
| Average hashing rate | SimilarWeb estimations | desktop / mobile: 25 $H/s$ |
| Monero market value | $ 253 as of May '18 | $253 for equal comparison |
| Time on website | SimilarWeb estimations | Average, 1st / 3rd quartile |

In the following analysis, we are following the same equation:

$$\text{traffic [\# flows]} \times \text{avg. time [s]} \times \text{mining rate [XMR/s]}$$
$$\times \text{ value [\$/XMR]} = \text{profit [\$]}$$

but adjust them for the specific attack vector observed. First, our NetFlow traces allow for an extrapolation of the actual number of HTTP connections on port 80, and we attribute the count of flows to the revenues of a *siteKey* installed on the proxy page at that time. While the embedded miners also work for iframed-HTTPS connections, we did not find evidence that this attack was pursued in the wild. This will thus be a lower bound on the amount of traffic.

Second, Konoth et al. estimated average visiting times for each of their 1,705 detected websites using SimilarWeb data, but the MITM attack works across all pages of the Internet. As the actual end point of the outgoing connection has been anonymized for privacy, we can approximate the average visiting time as we query the average visiting duration of websites listed in the Alexa Top 10k – the 10,000 most popular websites – on SimilarWeb. The average visiting time for these websites is 293 seconds. We will for our calculation work with three values for visit duration to provide a range of the revenues made by the attackers. We will use the average visiting time, as well as the first and third quartile of visit durations. Yet, already the highly conservative estimation based on the first quartile highlights the magnitude of this new attack vector. Table 5 compares the parameters used in [18] to our study.

Third, Konoth et al. also used SimilarWeb data to estimate the hashing rate for both mobile and desktop visitors, being 14.56 and 40.5 respectively. We estimated the hashing rate based on the desktop/mobile device ratio found across the Internet as a whole, which is listed in [11] as 0.58, resulting in a weighted hash rate of 25 $H/s$. Since we want to compare the profitability of MITM-based to website-based cryptojacking, we could either compare the amount of Monero mined, or translate the Monero amount into more intuitive currency such as USD. Currency exchange rates are however volatile and in between Konoth's May 2018 study and our study, the average Monero price had dropped during August and December 2018 to $92.2/XMR. To compare both attack vectors side by side, we thus use the same exchange rate as in [18], which still makes a fair comparison, as the decline woud have equally scaled down the revenues attackers could have generated using website-based mining during our observation period. Even if we scale the revenue down with the declined value of Monero, the MITM-based revenues would still be a factor of 10 higher than the website-based earnings made half a year earlier.

Based on the parameters chosen above, Table 6 shows the estimated monthly revenues for the top 10 grossing actors for the average visit duration on the Alexa 10K, as well as the first and third quartile. As we can see, in the average case the top 10 campaigns total a profit exceeding $1,200,000 per month, in which the highest grossing *siteKey* earns $222K. To put this into perspective, the 10 most successful campaigns that are deploying cryptojacking by installing miners on the websites themselves (for example by hacking the site) were reported by [18] to yield monthly revenues of some $41,000. Cryptojacking through a MITM attack on routers is thus a factor of 30 more lucrative than previously observed attack vectors, and the most successful MITM actor earns 5x more revenue than the top 10 website-based cryptojackers combined.

**Table 6: Estimated monthly revenue of top 10 grossing actors based on the average visit duration on the Alexa Top 10K, as well as the first and third quartile according to SimilarWeb.**

| SiteKey | Total # routers | First quartile 2'27" stay | Median stay 4'53" stay | Third quartile 6'19" stay |
|---|---|---|---|---|
| *48zUYB* | 52,181 | $111,447.18 | $222,136.22 | $287,336.61 |
| *6a9929* | 30,135 | $97,626.82 | $194,589.52 | $251,704.53 |
| *8C7UoT* | 47,981 | $90,532.54 | $180,449.21 | $233,413.82 |
| *BOvlp3* | 49,640 | $82,573.82 | $164,585.92 | $212,894.42 |
| *4983e3* | 117,502 | $70,017.28 | $139,558.26 | $180,520.75 |
| *FgWWtJ* | 39,384 | $50,719.01 | $101,092.99 | $130,765.33 |
| *J3rjnv* | 45,934 | $40,551.39 | $80,826.92 | $104,550.86 |
| *hsFAjj* | 223,844 | $35,396.11 | $70,551.44 | $91,259.37 |
| *BT9k5D* | 8,459 | $31,494.11 | $62,773.97 | $81,199.10 |
| *wjORhf* | 42,342 | $27,671.96 | $55,155.67 | $71,344.70 |
| **Total top 10** | | **$638,030.22** | **$1,271,720.11** | **$1,644,989.49** |

In our analysis above, we have seen the different roles the actors have played in the development and rollout of this attack vector and the different levels of innovation they have embraced. Curiously though, we find that innovation and a first mover advantage does not manifest in earnings. The actor with the key *hsFAjj*, who was among the first, dominated proxying and controls extensive infrastructure, did not translate this advantage into earnings at the same rate as for example the key *6a9929* who would pick up data on vulnerable routers from public lists to roll out infections.

Also the number of infected routers is not necessarily an indicator for the amount of revenue an adversary has generated, as the size (and thus also the type) of the router matters more than the number of compromised devices. Table 6 also lists the total number of routers a particular *siteKey* ever had under its control during the 10 month study, and we clearly see that the volume of routers is not an adequate predictor of monetary success. Another unexpected story emerges when we look at the routers that are providing the most revenue. Out of the top 10 most grossing routers, 6 are located in Iraq, and one each in Turkey, France, Brazil and the Netherlands, which is counterintuitive looking at the worldwide distribution of MikroTik installations shown earlier in Figure 6.

## 6.2 Charting the Ecosystem of Actors

While looking at the life cycle of router infections, we observe different levels of sophistication in every stage. In the identification stage, we discover a clear distinction between *siteKeys* installed as a result of scanning and infection based on public sources, such as Shodan. In the exploitation of the routers afterwards we observe a constantly changing landscape in which actors are regularly infecting new devices and stealing from each other. After infection, only a limited number of actors demonstrate a high level of sophistication by setting up an infrastructure. To monetize the hijacked routers,

actors initially set up HTTP proxies, but subsequently increased their revenues by installing SOCKS proxies with cryptojacking scripts. The used cryptomining scripts diverge to multiple services, and we have noticed a continuous flow of router infections and removals. Clear geographical differences in mining characteristics are identified, where Brazil and Indonesia are the most infected, while Iraq seems to have the most lucrative infrastructure to infect. Observed maintenance patterns show that specific anonymized IPs can be linked by behavior to *siteKeys*.

*Relating actors and siteKeys.* Based on the results of the different independent components analyzed in the previous sections, we are able to link certain *siteKeys* to each other and/or to IPs. To start with, three *siteKeys hsFAjj*, *SK_LCx*, *oDcuak* show similar behavior as the same infrastructural patterns can be found on routers infected with these *siteKeys*, as well as regular contacts with the same set of attacker IPs for maintenance over SSH. Figure 10 confirms this hypothesis by showing numerous routers transitioning between those *siteKeys*. Interestingly, the analysis of SOCKS traffic also links *J3rjnv* to this set. Additionally, this figure depicts the sophistication level of the actor behind *siteKey 4983e3*, as this actor hijacks vulnerable routers infected with numerous other *siteKeys*, but subsequently changes his own *siteKey* to a masked variant, as listed in Figure 9. Revisiting Figure 13, which shows 4 clear sequential blocks of 10 *siteKeys* having similar installation sizes and evolutional patterns. This in combination with the aforementioned figure, which shows 5 clear *siteKey* transition chains, an even larger number of *siteKeys* can be linked to one single adversary. By following each *siteKey* within these transition chains in Figure 10, we noticed that these transitions resemble transitions between the sequential blocks in Figure 13. All the *siteKeys* in the transition chains are located inside these blocks in the same sequence. For each of the *siteKeys* inside these four blocks, the first two blocks (highlighted in green in Figure 13) use a Coinhive miner, with the uncommon option `CoinHive.FORCE_EXCLUSIVE_TAB` enabled, and the latter two (highlighted in yellow) use Omine as a mining service. Additionally, all 40 mining scripts within these blocks were set to the same throttle value of 0.1. As a result, this common behavior across multiple *siteKeys* strongly suggests that we can thus link these 40 *siteKeys* to one single actor.

## 7 CONCLUSION

In this paper, we have reported on a new attack vector for cryptojacking, which does not infect websites but compromises Internet infrastructure itself. This vector greatly overshadows any cryptojacking campaigns known to date by orders of magnitude in installation size, and we find groups of actors compromising a total of 1.4M vulnerable routers, approximately 70% of all deployed MikroTik routers, with various degrees of sophistication. As the injection of miners into network traffic affects any user visiting any website, we find this attack vector to be highly profitable, estimated to exceed $1,200,000 per month in revenue for the top 10 actors.

Curiously, we find that innovation and the first mover advantage does not pay off in terms of revenue made. The highest grossing actors are not the ones creating new monetization options, deploying sophisticated infrastructure or creating the largest deployment, but those finding the most productive niche where they can operate relatively undisturbed.

In April 2019, Interpol has begun an investigation into the cryptojacking campaigns using MikroTik routers to investigate the perpetrators, clean up the infected routers and take the supporting infrastructure out of service [9]. To assist with this effort, the research team has shared the results and additional outcomes with the involved law enforcement agencies.

## REFERENCES


[1] Agrawal, R., Srikant, R., et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (1994), vol. 1215, pp. 487–499.

[2] Bijmans, H. L., Booij, T. M., and Doerr, C. Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale. In *Usenix Security Symposium* (2019).

[3] Bilge, L., and Dumitras, T. Before we knew it: An empirical study of zero-day attacks in the real world. In *ACM Conference on Computer and Communications Security* (2012).

[4] Blenn, N., Ghiette, V., and Doerr, C. Quantifying the spectrum of denial-of-service attacks through internet backscatter. In *International Conference on Availability, Reliability and Security (ARES)* (2017).

[5] Carlin, D., O'Kane, P., Sezer, S., and Burgess, J. Detecting cryptomining using dynamic analysis. In *Annual Conference on Privacy, Security and Trust* (2018).

[6] Christopher, N. Hackers mined a fortune from indian websites, Sep 2018. https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/hackers-mined-a-fortune-from-indian-websites/articleshow/65836088.cms (December 2018).

[7] Cimpanu, C. A mysterious grey-hat is patching people's outdated MikroTik routers, Oct 2018. https://www.zdnet.com/article/a-mysterious-grey-hat-is-patching-peoples-outdated-mikrotik-routers/ (February 2019).

[8] Claburn, T. Crypto-jackers enlist google tag manager to smuggle alt-coin miners, Jan 2018. https://www.theregister.co.uk/2017/11/22/cryptojackers_google_tag_manager_coin_hive/ (December 2018).

[9] CSIRT, N. MikroTik Routers Compromised in Cryptojacking Campaign, Apr 2019. https://csirt.cy/mikrotik-routers-compromised-in-cryptojacking-campaign/ (April 2019).

[10] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. A search engine backed by Internet-wide scanning. In *22nd ACM Conference on Computer and Communications Security* (Oct. 2015).

[11] Enge, E. Mobile vs Desktop Traffic in 2019, Apr 2019. https://www.stonetemple.com/mobile-vs-desktop-usage-study/ (May 2019).

[12] Eskandari, S., Leoutsarakos, A., Mursch, T., and Clark, J. A first look at browser-based cryptojacking. *European Symposium on Security and Privacy Workshops* (2018).

[13] Ghiette, V., and Doerr, C. How media reports trigger copycats: An analysis of the brewing of the largest packet storm to date. In *ACM SIGCOMM Workshop on Traffic Measurements for Cybersecurity (WTMC)* (2018).

[14] Hoffman, J. J., Lee, S. C., and Jacobson, J. S. New jersey division of consumer affairs obtains settlement with developer of bitcoin-mining software found to have accessed new jersey computers without users' knowledge or consent, May 2015. https://nj.gov/oag/newsreleases15/pr20150526b.html.

[15] Hong, G., Yang, Z., Yang, S., Zhang, L., Nan, Y., Zhang, Z., Yang, M., Zhang, Y., Qian, Z., and Duan, H. How you get shot in the back: A systematical study about cryptojacking in the real world. In *SIGSAC Conference on Computer and Communications Security* (2018).

[16] Huang, D. Y., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., McCoy, D., Savage, S., Weaver, N., Snoeren, A. C., and Levchenko, K. Botcoin: Monetizing stolen cycles. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014* (2014).

[17] Kharraz, A., Ma, Z., Murley, P., Lever, C., Mason, J., Miller, A., Borisov, N., Antonakakis, M., and Bailey, M. Outguard: Detecting in-browser covert cryptocurrency mining in the wild.

[18] Konoth, R. K., Vineti, E., Moonsamy, V., Lindorfer, M., Kruegel, C., Bos, H., and Vigna, G. Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense. In *SIGSAC Conference on Computer and Communications Security* (2018).

[19] MaxMind, Inc. MaxMind GeoIP2. https://www.maxmind.com/en/geoip2-services-and-databases/ (April 2019).

[20] McCarthy, K. Cbs's showtime caught mining crypto-coins in viewers' web browsers, Jan 2018. https://www.theregister.co.uk/2017/09/25/showtime_hit_with_coinmining_script/ (December 2018).

[21] MineCryptoNight. MineCryptoNight - Making mining profits great again! https://minecryptonight.net/ (May 2019).

[22] Murphy, M. Youtube shuts down hidden cryptojacking adverts, Jan 2018. https://www.telegraph.co.uk/technology/2018/01/29/youtube-shuts-hidden-crypto-jacking-adverts/ (November 2018).

[23] Mursch, T. Over 100,000 drupal websites vulnerable to drupalgeddon 2 (cve-2018-7600), Jun 2018. https://badpackets.net/over-100000-drupal-websites-vulnerable-to-drupalgeddon-2-cve-2018-7600/.

[24] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system, 2009.

[25] NetLab360. Quick summary about the Port 8291 scan, Mar 2018. https://blog.netlab.360.com/quick-summary-port-8291-scan-en/ (April 2019).

[26] NIST National Vulnerability Database. NVD - CVE-2018-14847 Detail, Jul 2018. https://nvd.nist.gov/vuln/detail/CVE-2018-14847.

[27] Papadopoulos, P., Ilia, P., and Markatos, E. P. Truth in web mining: Measuring the profitability and cost of cryptominers as a web monetization model. *CoRR abs/1806.01994* (2018).

[28] Pastrana, S., and Suarez-Tangil, G. A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth. *CoRR abs/1901.00846* (2019).

[29] Proxy Lists 24. Proxy Lists 24 - Daily Free Proxy Server Lists. http://www.proxyserverlist24.top/ (April 2019).

[30] Rauchberger, J., Schrittwieser, S., Dam, T., Luh, R., Buhov, D., Pötzelsberger, G., and Kim, H. The other side of the coin: A framework for detecting and analyzing web-based cryptocurrency mining campaigns. In *International Conference on Availability, Reliability and Security* (2018).

[31] Rodriguez, J. D. P., and Posegga, J. RAPID: resource and api-based detection against in-browser miners. *Annual Computer Security Applications Conference* (2018).

[32] Rüth, J., Zimmermann, T., Wolsing, K., and Hohlfeld, O. Digging into browser-based crypto mining. In *Internet Measurement Conference* (2018).

[33] Schwartz, M. J. Cryptojackers Keep Hacking Unpatched MikroTik Routers, Oct 2018. https://www.bankinfosecurity.com/cryptominers-keep-hacking-unpatched-mikrotik-routers-a-11627 (April 2019.

[34] Segura, J. A look into drupalgeddon's client-side attacks, Jun 2018. https://blog.malwarebytes.com/threat-analysis/2018/05/look-drupalgeddon-client-side-attacks/ (December 2018).

[35] Shodan. Shodan - The search engine for Internet-connected devices. https://www.shodan.io/.

[36] SimilarWeb. Similar web. website traffic statistics & market intelligence. https://www.similarweb.com (May 2019).

[37] SonicWall. Massive cryptojacking campaign compromised ̃200,000 MikroTik routers, Aug 2018. https://securitynews.sonicwall.com/xmlpost/massive-cryptojacking-campaign/ (March 2019).

[38] Tenable. MikroTik RouterOS Vulnerabilities: There's More to CVE-2018-14847, Oct 2018. https://www.tenable.com/blog/mikrotik-routeros-vulnerabilities-there-s-more-to-cve-2018-14847 (March 2019).

[39] The Monero Project. Monero: What is monero (xmr)? https://www.getmonero.org/get-started/what-is-monero/ (December 2018).

[40] The Pirate Bay. The pirate bay-miner, Sep 2017. https://thepiratebay.org/blog/242.

[41] Wang, W., Ferrell, B., Xu, X., Hamlen, K. W., and Hao, S. SEISMIC: secure in-lined script monitors for interrupting cryptojacks. In *Computer Security - 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part II* (2018), pp. 122–142.

[42] Wordfence.com. Wordpress plugin banned for crypto mining, Nov 2017. https://www.wordfence.com/blog/2017/11/wordpress-plugin-banned-crypto-mining/ (January 2019).

[43] Xu, J., Fan, J., Ammar, M., and Moon, S. B. On the design and performance of prefix-preserving ip traffic trace anonymization. In *ACM SIGCOMM Workshop on Internet Measurement* (2001).

# A CENSYS DATASET MINING DETECTION

**Table 7: Regular expressions used to detect mining code in the Censys datasets**

| Miner type | Regular expression |
|---|---|
| Coinhive | `new CoinHive\.Anonymous`<br>`coinhive.com/lib/coinhive.min.js`<br>`authedmine.com/lib/ \| coinhive \| cnhv\.co` |
| Cryptoloot | `CRLT\.anonymous`<br>`webmine.pro/lib/crlt.js \| cryptoloot`<br>`verifier.live/lib/crypta.js \| crypta` |
| Coinimp | `coinimp \| new CoinImp.Anonymous`<br>`new Client.Anonymous \| scrip`<br>`freecontent.data \| freecontent.date`<br>`hostingcloud.science \| hashing\.win`<br>`srcips \| freecontent.stream \| priv\.su` |
| Omine | `omine\b \| omineID` |
| Webminer | `coinwebmining.com \| cwm\.js \| serv1swork`<br>`mining711 \| gazanew` |
| Mineralt | `ecart\.html\?bdata= \| amo\.js\"`<br>`mepirtedic\.com \| gramombird\.com`<br>`tulip18\.com \| mineralt\.io \| dinorslick`<br>`istlandoll\.com \| feesocrald\.com`<br>`besstahete\.info \| nexioniect\.com`<br>`pampopholf\.com \| feesocrald` |
| Coinhave | `minescripts\.info` |
| Coinpot | `coinpot \| wait\.php` |
| Monero-mining | `perfekt` |
| Webminepool | `webminepool\.com/lib/base\.js`<br>`WMP\.Anonymous` |
| Obfuscated | `147\.135\.234\.198 \| 91\.134\.24\.238`<br>`unescape \| pastebin` |