A First Look at the Crypto-Mining Malware Ecosystem: A Decade of Unrestricted Wealth

Anonymous Author(s)

ABSTRACT

Illicit crypto-mining leverages resources stolen from victims to mine cryptocurrencies on behalf of criminals. While recent works have analyzed one side of this threat, i.e.: web-browser cryptojacking, only white papers and commercial reports have partially covered binary-based crypto-mining malware. In this paper, we conduct the largest measurement of crypto-mining malware to date, analyzing approximately 4.5 million malware samples (1.2 million malicious miners), over a period of twelve years from 2007 to 2019. Our analysis pipeline applies both static and dynamic analysis to extract information from the samples, such as wallet identifiers and mining pools. Together with OSINT data, this information is used to group samples into campaigns. We then analyze publicly-available payments sent to the wallets from mining-pools as a reward for mining, and estimate profits for the different campaigns. All this together is is done in a fully automated fashion, which enables us to leverage measurement-based findings of illicit crypto-mining at scale.

Our profit analysis reveals campaigns with multi-million earnings, associating over 4.4% of Monero with illicit mining. We analyze the infrastructure related with the different campaigns, showing that a high proportion of this ecosystem is supported by underground economies such as Pay-Per-Install services. We also uncover novel techniques that allow criminals to run successful campaigns.

1 INTRODUCTION

Mining is a key component responsible for the wealth of Blockchain-based crypto-currencies. This process requires a network of interconnected miners to solve a complex mathematical problem in order to link blocks and maintain the integrity of the transactions. In exchange, miners receive an amount of the mined crypto-currency as a reward.

The high value of crypto-currencies has attracted a large number of malicious actors that use hijacked resources to mine these currencies. The illicit crypto-mining threat has grown considerably over the recent years [1], and it is considered one of the top-most cybersecurity threats, even surpassing ransomware according to recent reports [42].

Illicit crypto-mining is typically conducted using either one of these two modes: (i) by using `browser`-based crypto-mining programs (dubbed cryptojacking [29]), where the mining process is run in scripts (typically JavaScript) embedded in web content; or (ii) by using `binary`-based crypto-mining malware, where the mining process is embedded in the payload of a malware running in infected machines that are connected to the Internet. In both cases, by using hundreds of hijacked machines, perpetrators can obtain a hash-rate similar to medium-sized mining farms. Each mode has different characterizing features and unique challenges, specially when it comes to devising effective countermeasures. For example, in browser-based cryptojacking the damage ceases when the victim stops browsing the site. Also, users can reduce the threat by restricting the use of JavaScript. Meanwhile, crypto-mining malware entails classical malware-related challenges, such as persistence and obfuscation. Also, since mining increases the CPU load, thus reducing the computer's performance, it might be noticed by end-users. Thus, we observe a new paradigm aimed at evading user- rather than AntiVirus-detection using techniques such as *idle mining* (mining only when the CPU is idle) or reducing CPU consumption when monitoring tools (e.g., Task Manager) are running. For readers unfamiliar with the topic, we refer to the *Background* section in §2 for an introduction to cryptocurrency mining and its threats.

Motivation. While illicit crypto-mining has been less notorious than other threats such as ransomware, it poses nonetheless an important threat to users and organizations; and its presence is an indicator of weaknesses in security practices that must be addressed [24]. First, the profits generated by their miners introduce massive incomes to cyber-criminals. These incomes fuel the underground economy and gear other cyber-criminal activities [38]. Second, this threat causes important economical loses to their victims. By draining the CPU-usage, corporations see how their electricity bills increase and how their hardware rapidly deteriorates [36, 42]. Finally, this indirectly causes a non-negligible environmental footprint [15]. Due to these concerns, browser-based crypto-mining has been widely studied recently, both analyzing it as a crime [8, 13, 17, 34] and as an alternative business model to monetize web content [28, 30, 33]. However, the literature lacks of a systematic approach to measure the binary-based mining threat at scale.

The first and only seminal work putting this threat in perspective is from 2014 [9]. Authors analyzed 2K malware samples mining Bitcoin and their methodology relied on the analysis of public transactions. However, there has been a significant increase in the number of malware samples monetizing this threat since 2014 [1, 7]. Also, criminals' attention has shifted to other cryptocurrencies, mainly motivated by: i) the proliferation of ASIC mining, which uses dedicated hardware and renders the use of desktop computers no longer profitable for mining bitcoins, and ii) the development of protocols that provide transaction anonymity (such as those used in Zcash or Monero). Anonymous currencies are used by criminals to thwart traceability and they are on demand in underground markets. Commercial reports, in the form of blog posts [7, 29] or white papers [24] provide a further, but limited, view of the magnitude of the problem and the landscape. Security firms have analyzed isolated cases of decontextualized mining operations [10, 11]. However, these studies are limited by the simplicity of the analysis.

In this paper, we aim to bridge these gaps by addressing the following research questions: (1) What are the preferred cryptocurrencies mined by criminals? (2) How many actors are involved in this ecosystem and what are their profits? (3) What is the level of sophistication used in different campaigns and how does this affect the earnings? (4) What is the role of underground markets and what are the tools and techniques adopted from them? (5) How can we improve current countermeasures and intervention approaches? Due to potential *ethical concerns* arisen from this work (see Appendix), we obtained approval from our REB office.

Novelty. Our work focuses on crypto-mining malware. By looking at a wide-range of underground communities, where knowledge and tools are shared, we have observed increased interest in this malware. This suggests that cybercrime commoditization plays a key role in the wealth of illicit crypto-mining. We design a measurement pipeline to automatically analyze malware samples observed in the wild and to extract information required to identify the miners and pools, using both dynamic and static analysis. Then, we build a graph-based system that aggregates related samples into campaigns based on a series of heuristics. The system is designed to distinguish campaigns using third-party infrastructure such as Pay-Per-Install (PPI) services or binary obfuscators. This allows to analyze to what extent this threat is sustained by different underground markets [38]. Our analysis system enables the research community to leverage crypto-mining measurements at scale.

Findings. Among others, our main findings include:

(1) Monero (XMR) is by far the most popular cryptocurrency among cyber-criminals in underground economies (§4.2). Considering only crypto-mining malware, our profit analysis shows that criminals have mined over 4.37% of the circulating XMR. Although this depends on when criminals cash-out their earnings, we estimate that the total revenue accounts for nearly 58M USD. These criminal earnings should be added to estimations from parallel work focused on browser-based cryptojacking (§7).

(2) Campaigns that use third-party infrastructure (typically rented in underground marketplaces) are more successful. However, this is not always the case. Some of the most profitable campaigns rely on complex infrastructure that also uses general-purpose botnets to run mining operations without using third-party infrastructure. Here, we discover novel malware campaigns that are previously unknown to the community (e.g., the code-named `Freebuf` or the `USA-138` campaign presented as case studies in §5). Moreover, only some criminals keep their infrastructure updated, for example when they are banned in mining pools or when the mining software needs to be updated due to changes in the mining algorithm.

(3) Campaigns use simple mechanisms to evade detection, like using domain aliases to contact mining pools (which prevents simple blacklisting approaches), or *idle mining*.

(4) There are other criminals running successful campaigns with minimal infrastructure. A common yet effective approach is to use legitimate infrastructure such as Dropbox or GitHub to host the droppers, and stock mining tools such as `claymore` and `xmrig` to do the actual mining. We also show what are the most popular Monero mining pools (`crypto-pool`, `dwarfpool` and `minexmr`) among criminals and discuss the role of these and other pools when devising countermeasures.

Contributions. To the best of our knowledge, this paper presents the largest systematic study of malicious binary-based crypto-mining. Our main contributions are:

1. We analyze and describe the role of underground communities for the proliferation of the illicit crypto-mining business (§2).
2. We present a system that uses both static and dynamic analysis to extract relevant mining-related information from crypto-mining malware, such as wallet addresses and pool domains (§3). Our system uses different techniques to aggregate related samples into larger campaigns represented as a graph that is then mined for further analysis. Additionally, we feed the system with information gathered from various Open-Source Intelligence (OSINT) repositories to further classify and analyze the campaigns.
3. We present a longitudinal study of the crypto-mining malware threat using data spanning over more than a decade (§4 and §5). Then, by focusing on Monero, we rely on information gathered from mining pools to measure the earnings gained by each campaign. We also analyze the infrastructure used by criminals and extract the attribution to stock mining software.
4. We propose a number of countermeasures, and discuss the efficacy of existing ones together with the

open challenges (§6). Then, we contextualize the most important findings of our study with respect to relevant works in the area (§7).

Finally, to foster research in the area, we release our dataset in our online repository[1]. We encourage readers to visit this repository as it provides a wider presentation of the measurements left out of this paper due to space constraints.

2 BACKGROUND

In this section, we first provide an overview of the crypto-currency mining process. Then we describe the underground economy supporting the illicit crypto-mining threat.

Cryptocurrency Mining. Crypto-currencies are a type of digital assets that can be exchanged in online transactions. These transactions are grouped into blocks and added to a distributed database known as the blockchain. Each block is linked to its previous block. Addition of new blocks to the blockchain is done by voluntary miners. These must compute a cryptographic hash of the block, which includes complex mathematical puzzle known as 'Proof-of-Work' (PoW). As a reward, miners receive a certain amount of the currency. The mining process maintains the integrity of the blockchain and it is at the core of all crypto-currencies.

The increased value of crypto-currencies such as Bitcoin or Ethereum leads to the growth of mining farms using specialized hardware known as ASICS. Thus, mining these crypto-currencies using end-user machines such as laptops or desktop computers was useless. However, in 2014 a new PoW known as Cryptonote required not only CPU power but also memory, turning ASIC-based mining inefficient and thus gaining again the attention of individuals willing to mine with their home machines. Additionally, the mining algorithm changes periodically, thus discouraging ASIC development (which is optimized for specific algorithms) [19]. Examples of crypto-currencies using Cryptonote as PoW are: Monero (XMR) and Bytecoin (BCN).

When a new block is added to the blockchain, only the first miner being able to mine the block will get the reward. This turns the mining process into a race where speed is the hashrate of a miner. The higher the hashrate, the higher the probability of mining a block and thus getting the reward. Accordingly, mining is typically done using public mining pools, which can be viewed as partnership services between various workers where the complexity of the mining challenge is distributed among the partners. Each partner contributes with a given hashrate aimed at solving the puzzle, and when the pool successfully mines a block, the reward is divided among the partners proportionally to their hashrate. In order to get this reward, workers must provide some form of identification. This can be proprietary site-keys, like in the case of CoinHive (the major provider of browser-based mining services), emails or wallet addresses. The communication between each miner and the pool is done by using Stratum, which is a de-facto TCP based protocol evolved from the *getwork* protocol [32].

Crypto-mining malware and cryptojacking. Crypto-currency mining is a rather easy monetization technique using hardware resources. However, it requires an investment in equipment and also entails a cost in terms of energy. In illicit cryptomining, criminals make use of their victims' computing resources to mine crypto-currencies on their behalf. This threat exists since the creation of Bitcoin in 2009, but it has increased since 2014 due to the inception of Cryptonote and other PoW algorithms resistant to ASIC-based mining.

Illicit crypto-mining is performed by using two techniques: browser-based or binary-based mining. In the former, the mining payload is embedded in web resources which are executed by client browsers without the explicit consent of the users [8, 17]. In the latter, the payload is distributed in the form of malware.

The Underground Economy. Underground markets play a key role in the business of malicious crypto-mining. Users with few technical skills can easily acquire services and tools to set up their own mining campaign. Forums are also used for sharing knowledge. To put our study in context, we have analyzed a dataset of posts collected from various underground forums [31], looking for conversations related to crypto-mining. We observe that crypto-mining malware can be easily purchased online, for a few dollars (e.g., the average cost for an encrypted miner for Monero is 35$). In particular, we have seen an online service which allow to create customized binaries (e.g., for a particular cryptocurrency and/or a given pool) to mine cryptonote based currencies, for $13.[2] It provides several stealthy-related techniques such as idle mining or the use of execution-stalling code [16] targeted to certain conditions (e.g., when the Task Manager is running). Other providers opt to share their miners for free, in exchange for a donation: *"Miner is free, we charge a fee of 2% to cover the time coding."* Figure 1 shows a longitudinal analysis of posts related to crypto-mining in these forums. Here, we show that Monero is the most prevalent currency nowadays.

We also observed that a common topic of conversation concerns (i) "friendly" pools, i.e.: pools that do not generally ban users displaying botnet-like behaviors, or (ii) how to remain undetected otherwise. For instance, users claim that a good trade-off between profitable hash-rates and a long-lasting mining strategy is using botnets with less than 2K bots. For bigger botnets, many discussions and tutorials explain how to configure proxies and provide advice on how

[1] https://github.com/miners2019submission/

[2] For ethical reasons we do not disclose the URL of this service

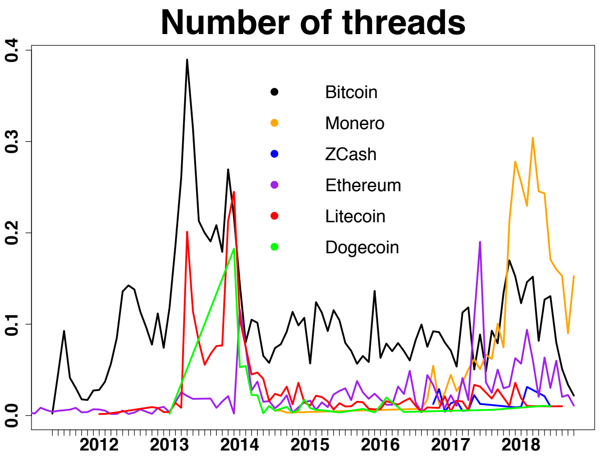


Figure 1: Evolution of the number of threads from underground forums related to mining of different crypto-currencies.

to reduce the risk of being exposed: *"The best option is to use a proxy and you can use any pool. Contact me for PM, I am willing to help"*. Also, we found various conversations with users looking for partners and offering custom (private) mining pools: *"In my pool there is no ban by multiple connections."*

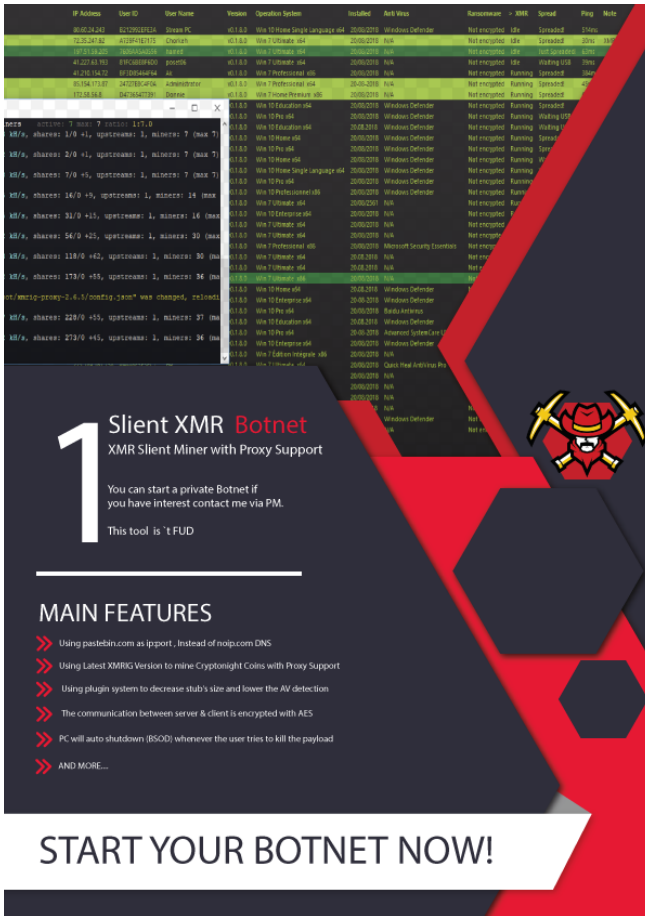
Finally, we note that it is also possible to purchase *all-you-need* packages, including tools and services, with a guarantee period and maintenance (e.g., re-obfuscation when the miner is detected, or updates to new versions). For the curious reader, Figure 2 shows a flyer posted in one of the underground markets, which offers a full Monero botnet.

Take-Away: The support offered by underground communities to criminals explains the sharp growth on the amount of malware monetizing their victims. This motivates the need for a longitudinal measurement of this threat. We show that Monero is currently the most discussed crypto-mining coin by underground forum users.

3 MEASUREMENT METHODOLOGY

A general overview of the measurement methodology is presented in Figure 3. For the sample collection, we query both public and private repositories of malware and different intelligence feeds as described in §3.1. We make a number of sanity checks for each sample to ensure that we only feed crypto-mining malware to our pipeline (see §3.2). We also collect OSINT related to running botnets and relevant Indicators of Compromise (IoCs) observed in malware samples.

A key phase in our pipeline is to analyze relevant samples both statically and dynamically as described in §3.3. (*Binary*, *Sandbox*, and *Network Analysis* in Figure 3.) The goal of this multi-step phase is to extract the following information from each miner: (i) the pool or address which the crypto-mining malware connects to for mining, and the identifier used to authenticate themselves into this pool, (ii) the addresses of the e-wallets where mined cryptocoins are paid to[3], which in most cases coincide with the identifier, (iii) URLs where the malware connects to or is seen at, and (iv) other metadata obtained from intelligence feeds such as when the sample was first seen or related samples (e.g., dropped binaries).


1 Slient XMR Botnet

XMR Slient Miner with Proxy Support

You can start a private Botnet if
you have interest contact me via PM.

This tool is `t FUD

MAIN FEATURES

- Using pastebin.com as ip:port , Instead of noip.com DNS
- Using Latest XMRIG Version to mine Cryptonight Coins with Proxy Support
- Using plugin system to decrease stub's size and lower the AV detection
- The communication between server & client is encrypted with AES
- PC will auto shutdown (BSOD) whenever the user tries to kill the payload
- AND MORE...

START YOUR BOTNET NOW!

Figure 2: Crypto-mining package offered in an underground forum, including botnet setup, XMR miner and proxy. Permanent link: https://perma.cc/4FN8-B98M.

The next step is to analyze the mining pools that the miners work with. We decouple connections made to proxies (that in turn connect to pools) from the connections to the actual pools. We then look at the profit reported by each of the wallets in a pool. These two steps are described in §3.4 and are referred to as *Profit Analysis* in Figure 3.

Finally, we aggregate related samples into campaigns and analyze them separately in the *Aggregation* step as described

[3] The terms *address* and *wallet* are used interchangeably in the literature.

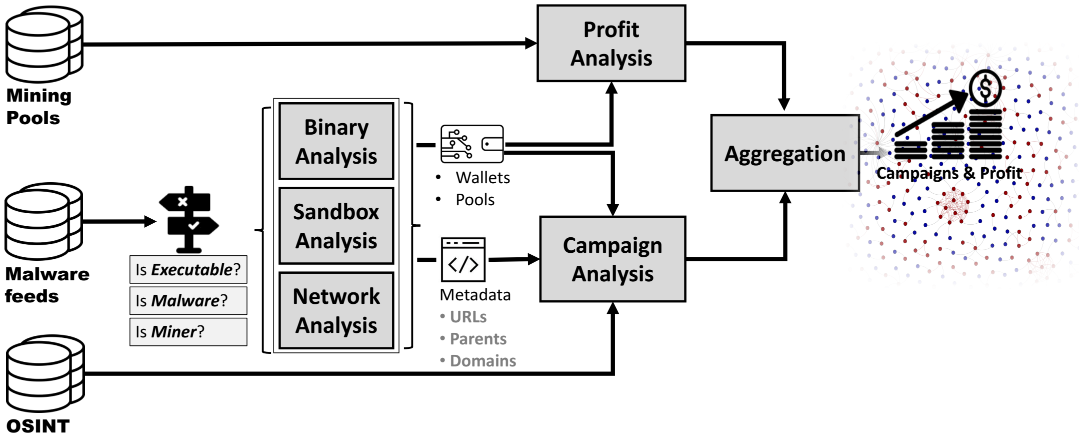


Figure 3: Overview of our processing pipeline and measurement methodology.

in §3.5. For this, we create a graph which interconnects crypto-mining malware that: (i) share a common execution ancestor (i.e., dropper) or are packed together, (ii) accumulate their earnings into the same wallet, (iii) share common infrastructure (e.g., proxies or hosting servers), or (iv) relate to the same IoC related with mining campaigns — gathered both from OSINT reports and our own investigation. We then enrich every interconnected sub-graph (campaign) to include details about related infrastructure used in each campaign (i.e., stock mining software or Pay-Per-Install services).

3.1 Data Gathering

We collect malware samples, metadata and OSINT information, and known mining tools from various sources.

Malware. We rely on public and private feeds from:

(1) **`Virus Share`**. This is an online community that shares torrents to malicious binaries. We use it to gather our initial dataset of raw binaries.
(2) **`Virus Total`**. This is an online service containing both publicly and privately available services to the security community through an API (Application Programming Interface). This service is a subsidiary of Google that runs multiple AntiVirus (AV) engines and offers an unbiased access to resulting reports. We use the private API to download malware binaries. We then query the public API to obtain relevant threat intelligence (which we refer to as metadata in the paper). In particular, we collect metadata from the samples obtained through queries to Virus Total, but also from all the samples obtained through other sources listed in this section.
(3) **`Hybrid Analysis`**. This is an online community providing malicious binaries and threat intelligence obtained from static and dynamic analysis of the samples. Thus, we use this community to fetch readily available intelligence when possible.
(4) **`Other Sources`**. We have developed a crawler to fetch samples from a variety of online communities such as malc0de.com or vxvault.net. Our pipeline also aggregates malware feeds from cybersecurity companies. For the purpose of this paper, we have received feeds from Palo Alto Networks with miners known to them.

Refer to Appendix A.3 for details about the dataset overlaps.

Metadata. When available, we primarily rely on the following metadata to put our study in context: (i) the first time the sample was seen in the wild, (ii) the URLs where the sample was seen, (iii) the list of parents that are known to have dropped the binary under analysis, and (iv) the list of contacted domains.

Stock Mining Software. We also collect binaries from *known mining frameworks*, such as `xmrig`[4] or `xmr-stak`[5], that are hosted in various public repositories. While these binaries are not badware *per se*, their usage is deemed malicious when run by malware. Our assumption is that the usage of proprietary software to mine is not the norm. Anecdotal evidence observed during the course of a preliminary investigation has shown that miscreants rely on legitimate — open-source — mining tools. The modus operandi of the malware is to fetch one of these tools (i.e., acting as a dropper) and run it in infected machines. Mining is configured with the wallet of the miscreant, where the rewards are paid by the network. One of our goals then is to understand if this assumption holds true and how many campaigns are using stock mining software illicitly.

[4] https://github.com/xmrig/xmrig
[5] https://github.com/fireice-uk/xmr-stak

Summary. Our data collection registers over 4.5 million samples (see §4.1 for a breakdown), which have been active between early 2007 and early 2019. This includes about 1K versions of known mining tools from 13 different frameworks. Our initial data contains a wide-range of samples, many of which are irrelevant to this study (e.g., web-based cryptojackers). Thus, we next describe the rules we use to consolidate the dataset where we report our findings with.

3.2 Sanity Checks

One important aspect when systematizing the analysis of malware is properly curating the dataset [20]. We perform the following sanity checks for each sample processed: (i) *is it malware?* (ii) *is it a miner?*, and (iii) *is it an executable sample?*

First, we rely on Virus Total reports to learn if a sample is malware. Virus Total have been shown to perform remarkably well when providing malware feeds according to a recent comparative analysis of Threat Intelligence [21]. In particular, Virus Total was able to detect 99.94% of the threats over one of the largest non-targeted[6] malware aggregators. We assume that a sample is malware if at least 10 AV vendors flag the sample as malicious. While this is a common practice in other works in the area [12, 25, 26], we acknowledge that having a solid ground-truth is essential (see discussion in §6). Thus, we use a white-list with the hashes of known mining tools, to ensure that they are not considered as malware samples in our study. This white-list is compiled from binaries collected from various online open-source repositories.

Second, we assume a malware is a crypto-mining tool when there are IoCs that reveal the use of cryptophic mining. We apply publicly available YARA rules[7] to our samples. Additionally, we query OSINT information with IoCs extracted from the samples (e.g., file hashes or network data). We also use advanced queries from Virus Total and Hybrid Analysis to look for malware that meet the following criteria: (i) samples that contact domains of known mining pools, (ii) communicate through the Stratum protocol, and (iii) are labeled as "Miner" (or related variants) by more than 10 AVs.

Finally, to understand whether a malware is executable, we rely the magic number from its header, and consider only those related to executables like PE, ELF or JARs. §6 provides discussion of the limitations behind these assumptions.

3.3 Extraction of Pools and Wallets

With our dataset of crypto-mining malware, we perform:

(1) *Static Analysis*: we perform binary inspection to extract evidences of mining activity embedded *into the binary*.
(2) *Dynamic Analysis*: we then use environmental information obtained from the *execution of the binary* in a sandbox. Specifically, we obtain the network traffic, the dropped files, processes opened and command line parameters passed to the binaries.

In some cases we are able to find identifiers (e.g., wallets or emails) and pool names using static analysis. In other cases, we rely on dynamic analysis to extract these identifiers from the network activity or the command line processes. In both cases, we process the output of these two analyses using heuristics and regular expressions to extract the following information:

Cryptocurrency wallets. Miners connect to the pools using the Stratum protocol [32]. Upon connection to the pool, they send a request-for-work packet with the identifier of the miner in a 'login' parameter. This identifier can be extracted from the command line options passed to the mining tool or directly from the network traffic. We also process the type of wallet to understand the cryptocurrency (e.g.: Monero, Bitcoin or Ethereum) the malware is intending to mine.

Mining pools. We collect additional information such as domains and IPs of mining pools and proxies. Similarly to wallet addresses, this information is typically extracted from either the command line of the process invoking the mining tool or from the network traffic. Typically, miners connect to a known pool.[8] In some cases, the miner either uses a proxy or mines against a private/unknown pool.[9] We consider that a miner is using a proxy if we record mining activity for the corresponding wallet in a known pool (see §3.4).

3.4 Collecting Mining Activity

One of the main challenges when measuring the impact of the malicious crypto-mining campaigns is the difficulty to accurately estimate the profits. In the case of browser-based cryptojacking, recent works use estimations of the number of visitors per hour for similar websites and the average hashrate of a single visitor (victim) [5, 8, 17]. This is highly inaccurate as evidenced by the variances reported by concurrent related works (see §7). In the case of crypto-mining malware, the actual wallet which the mining reward is paid to can be extracted. We leverage public information obtained from mining pools (which include total reward paid to wallets) to get a more approximate estimation of the profits.

[6] Meaning that they target malware threats to generic platforms. Other targeted malware aggregators focus on threats that specifically target platforms like Facebook and "that are not as relevant to most Virus Total users" [21].

[7] https://github.com/Yara-Rules

[8] We consider known pools as those listed in public sources, e.g.: http://moneropools.com/ or http://www.blockchain.com/pools.

[9] While the use of private pools is encouraged in certain underground communities, we have observed few samples using private pools.

For all the extracted wallets, we queried the most prevalent mining pools to collect activity associated with these wallets. While the amount of information offered by each pool varies, it always contain the timestamp of the last share, the current (last) hashrate and the total amount of currency paid to the wallet. Additionally, some pools also provide the historic hashrate of the wallet and the list of payments done to the wallet (including timestamp and amount). While the total paid is always available, some pools only provide payment data for the last period (e.g. a week or month). Since we are interested on studying how the payments evolve across time, we use public APIs to collect this information periodically for a period of 10 months (July'18-April'19). As a single wallet can use more than one mining pool, we queried all the wallets against all the pools. Then, to estimate profits, we aggregate all the payments sent by the pools to the wallets. In general, we report payments using XMR. To ease readability we also report the equivalent in US dollars (USD). However, we note that we do not have information about when the criminals have cashed-out their earnings (if ever). Thus, it is hard to extract an exact figure in USD (and other currencies) due to the fluctuations on the value of Monero. To approximate this value, we dynamically extract the exchange rate between XMR and USD of the date when the payments were made, if available. We use the average exchange rate of 54 USD/XMR in cases where historical payments are unavailable.

3.5 Campaign Analysis

Two mayor limitations in related works are: i) the simplicity in which they analyze related mining campaigns, and ii) the inability to study anonymous crypto-currencies such as Monero (as discussed in §9). Thus, in this work, we aggregate samples into campaigns by leveraging various characterizing features observed in the wild. We emphasize that the methodology we use to aggregate samples into campaigns is novel. We also note that our methodology admits a wide range of features.

Spreading Infrastructure. We distinguish two types of infrastructure used to spread the malware: one that can be owned and another one that belongs to a third-party and can be rented (e.g., botnets that are monetized as PPI services and that are used for mining). When available, we link samples to known botnets by querying OSINT information with IoCs extracted from the samples. We refrain from using these botnets to aggregate samples as we detail later. However, we use them to enrich the information of the campaigns in a post-aggregation phase. This way, we can draw conclusions about the number of campaigns using *known* third-party infrastructure. However, since we rely on public intelligence feeds, a limitation of this approach is that samples using *unknown* third-party infrastructure (e.g., offered in underground markets) might be aggregated together in a single campaign. In these cases, we can guarantee that the campaign runs on top of the same infrastructure. This is relevant to law enforcement agencies when devising takedowns strategies. Thus, our analysis considers campaigns that are either from the same actor or a group of actors that use the same infrastructure, independently from the monetizing approach used by the operators of the infrastructure that spreads the samples. Analyzing whether profits from a campaign are given to a single actor or a group of actors is out of the scope of this paper.

Grouping Features. We rely on the following features to group samples into campaigns:

Same identifier: In order to get rewards from the mining pools, workers must mine using a unique identifier, which in most cases corresponds with the wallet address to which payments are made. In other cases, these are e-mails or other identifiers, like user-generated names. If two samples contain the same identifier, it means that they are accumulating earnings in the same wallet and thus they are grouped together. Some mining tools contain donation wallets to reward the developer, which is done by mining for a certain time (typically 2-5%) using the donation wallet. While this is configurable and can be turned off, we have observed some samples doing donations. We note that the CPU cycles donated are also hijacked from the victim and therefore inflict harm to her. However we are primarily interested in measuring the earnings of the miscreants, and thus they are whitelisted and excluded from the aggregation. Currently, we have whitelisted 14 donation wallets directly obtained from the developers' sites.

Ancestors: In many cases, the same sample is used to download additional malware. This is the case of droppers, which adapt based on information gathered from the infected host, e.g.: operating system or processor capabilities. Accordingly, if a sample is parent of two samples with different wallets, these are grouped together. Ancestors and other dropped files that are not directly intended for mining are considered auxiliary binaries and we refer to them as `ancillaries`. This includes samples that do not have a wallet.

Hosting servers: We use metadata from the samples to extract the URL from where the malware was downloaded. A common approach is to host the malware (or even stock mining software) in public cloud storage sites such as Amazon Web Services (AWS), Dropbox or Google Drive (see §4.2). Thus, we aggregate two samples if either they are downloaded from the same IP address which does not resolve to a domain from a public repository, or if they are downloaded from exactly the same URL, e.g: *hxxp://suicide.mouzze.had.su/gpu/amd1.exe*. We also include the parameters to avoid those cases where a parameter is used to uniquely identify the resource being hosted, e.g.*hxxp://file8desktop.com/download/get56?p=19363*.

This approach has as limitation that we are not aggregating resources where a URL contains ephemeral information (e.g., timestamps or click-IDs), even when they point to the same resource in the server. However, this limitation is partially overcome due to other sources for aggregation.

Known mining campaigns: As mentioned, we collect IoCs (e.g., domains or wallets) from mining operations reported publicly. We look at IoCs that are known to belong to a given mining operation, and look for matches against samples in our dataset. We group two samples if they belong to the same operation. In our analysis, we have collected IoC for the following mining operations: Photominer [6], Adylkuzz [10], Smominru [11], Xbooster [39], Jenkins [3] and Rocke [23]. However, our methodology is designed to easily include data collected from new operations.

Domain aliases (CNAMEs): During our investigation, we observed many samples using domain aliases (i.e., CNAMEs) that resolve to known mining pools. In these cases, miscreants create one or various subdomains for a domain under their control, and set these subdomains to be aliases of known mining pools. Since the resolution is done for the CNAME rather than for the mining pool, they thwart defenses blacklisting mining pools (see §6 for a discussion on anti-analysis techniques). To address this evasion method, we perform DNS requests for all the domains extracted from our samples, and look for responses pointing to known mining pools from a CNAME. Since CNAMEs might have changed, we also query a DNS history-resolution service provided by AlienVault (https://www.threatcrowd.org) Accordingly, we aggregate samples using the same domain alias.

Mining proxies: Mining using a large number of machines (i.e., more than 100) with the same wallet raises suspicion of botnet usage, and mining pool operators might opt to ban the miner. To prevent this situation, offenders use mining proxies that gather all the shares from the different bots and forward the aggregated to the pool. Thus, pool operators only receive responses from a single machine, the proxy. As described in §3.3, we identify various samples using proxies. We aggregate together samples that use the same proxy.

Aggregation. To understand how many related campaigns there are and how they are structured, we build a graph where nodes are elements of a given resource (e.g., malware samples, proxies, or wallets) and the edges are determined by the relationships mentioned above. We consider each connected component of the graph as a single campaign, where the internal nodes of the graph represent the crypto-mining malware together with the infrastructure used by the campaign.

Enrichment. After the aggregation, we enrich each campaign with samples related to known Pay Per Install (PPI) services, and mining tools. We emphasize that these features are only informative and they are not used to aggregate campaigns. We next explain the rationale behind this.

Botnets and PPI: A common approach to spread malware is through PPI services, where customers pay a fee to botnet operators in order to spread their malware [2]. Due to commodization of cybercrime services, purchasing a botnet to spread malware is simple and open to anyone with few technical skills, e.g.: by leveraging underground markets [40]. During our analysis, we have observed samples belonging to various botnets that are commonly used as PPI services, such as Virut or Nitol. Since this is a known third-party infrastructure, two samples using these services are not necessarily related to each other and are not aggregated together.

Stock Mining Software: During our exploratory analysis, we have observed that many campaigns use stock mining software. This is, the hash of a file dropped by the malware matches with one of the hashes in our collection of mining tools. Actually, we have observed that some crypto-mining malware fetch this executable directly from the official GitHub repository. However, we have also observed that some miscreants fork these projects and make minor modifications to the mining tool, e.g. to remove donation capabilities.

We use Fuzzy Hashing (FH) to pick up on the aforementioned modifications and to relate these samples with known mining tools. FH is a similarity preserving hash function that allows to compare binary files. Specifically, FH computes a fingerprint of each binary in such a way that any two binaries that are almost identical map to a "similar" hash value. Fuzzy hashing has been shown to be an effective way of comparing malware [22]. In our pipeline, we use context triggered piecewise hashing [18] and compute the distance between the FH of all samples in a campaign and the FH of all known mining tools. We choose a conservative distance threshold of 0.1 as it performs well when comparing malware [22]. Thus, samples with a distance lower than 0.1 are considered as stock mining tools.

4 THE BINARY-BASED MINING THREAT

In this section, we present the analysis of our measurement. We first present our dataset (§4.1), which contains malware seen for over a decade. Next, we perform a longitudinal analysis through the lens of our dataset (§4.2). Then, we characterize the type of mining pools and currencies we have seen (§4.3) and study the earnings of the campaigns in the most prevalent cypto-currency, i.e.: Monero (§4.4).

4.1 Dataset

Our study results from the collection of 4.5 million malware samples from the range of sources described in §3.1. We then apply sanity checks to tailor our analysis to crypto-mining

Category	Type	#Samples
Summary	ALL EXECUTABLES	1,230,248
	Miner Binaries	1,017,323
	Ancillary Binaries	212,925
Sources	Palo Alto Networks	629,125
	Virus Total	956,255
	Virus Share	521
	Hybrid Analysis	858
Resources	Sandbox Analysis	1,143,572
	Network Analysis	258,565
	Binary Analysis	10,204

Table 1: Our dataset of miners and ancillaries, with the collection sources and the number of resources.

malware only, resulting in a total of: (i) 1,017,323 miner binaries, and (ii) 212,925 ancillary binaries. The samples in (i) are samples where we have observed mining capabilities together with an associated wallet and a pool address. The samples in (ii) are samples used by the miners to run the mining operation (e.g., bot clients or loaders). In total, our study leverages over 1.2 million crypto-mining malware samples. Table **??** shows a summary of our dataset, together with the breakdown of data sources and the type of resources we resort from.[10] Our largest source of miners is Virus Total and the smallest is Virus Share. As for the resources, we collect the largest number of wallets and pools through dynamic analysis (sandbox and network analysis). The data collection dates to March 2007 to capture the structure of the third party infrastructure from their early stages. However, malicious mining activity starts getting traction in 2011.

4.2 Longitudinal Analysis

We extract 16,050 different crypto-mining identifiers from a total of 103,894 samples. As mentioned, these mostly include addresses of wallets from various cryptocurrencies, but we also find emails and other identifiers used to authenticate the miner in the pool to later pay them the corresponding reward. In the case of wallets, we use regular expressions to detect the associated currency. Regarding the emails, we observe that the majority (97%) are used as identifiers of one of the most popular mining pools, i.e.: *minergate*.[11]

Overall, we aggregate samples into 11,387 different campaigns. Figure 4 depicts the Cumulative Distribution Function (CDF) for the number of samples, and identifiers in all the campaigns. It also shows an overview of the earnings made in pools that provide public statistics for wallets. Leftmost side of Table 2 shows the breakdown of the number of campaigns per type of identifier (i.e., wallets and other identifiers). Recall that wallet addresses represent the public key of an electronic wallet in a given cryptocurrency. Thus, we show the breakdown for the different cryptocurrencies for which we have wallets. Note that two or more identifiers can be used in the same campaign, for example due to a change of a previous wallet address after being banned [11]. Monero is the cryptocurrency most frequently used, followed by Bitcoin. There are at least 18 campaigns using two or more currencies. While most of the campaigns are composed by one or few wallets (see Figure 4), we observe campaigns having up to 304 different identifiers.

Out of all the samples with wallets, we queried the *first seen* from Virus Total. Rightmost side of Table 2 shows the number of samples with wallets by year for the most prevalent cryptocurrencies, i.e.: Monero (XMR) and Bitcoin (BTC). Overall, the dataset contains 7.6K BTC and 62K XMR malware samples.[12] Judging by the number of samples and the distribution across time, we can confirm the decreasing interest in Bitcoin in favour of Monero. Moreover, we have queried available Bitcoin pools with the BTC addresses, and observed negligible earnings (i.e., less than 5K USD). Note that the data collection ended in early 2019 and thus data from this year is partial. Also, due to constrains in the Virus Total rate limit we could not retrieve the *first seen* entry for some of the most recently discovered samples. However, we attribute this samples to 2019 (denoted as ~*19?* in Table 2).

Table 3 shows an excerpt of the most popular URL domains hosting crypto-miners. [13]. We observe that GitHub is the chosen site used to host the crypto-mining malware. This is because GitHub hosts most of the mining tools, which are directly downloaded — for malicious purposes — by droppers as discussed before. Additionally, GitHub is also used to host modified versions of the miners (e.g., by removing the donation capabilities or adding further capabilities). It is also used to host ancillary malicious tools [37]. We also observe that there are other public repositories and file sharing sites such

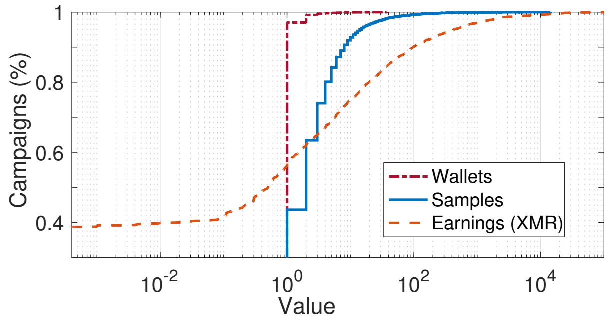


Figure 4: Cumulative distribution number of samples, wallets and earnings observed per campaign.

[10] See Appendix A.3 for details about other sources crawled.

[11] For a detailed analysis, see Table 12 from Appendix A.4.

[12] These are samples with embedded wallets and does not include ancillaries.

[13] For a complete list ordered by prevalence, see Table 10 in Appendix A.4

# campaigns with wallet addresses for:				# samples seen in:		
Monero	2,360	Aeon	57	Year	BTC	XMR
Bitcoin	1,523	Sumokoin	18	2012	9	1
zCash	177	Intensecoin	8	2013	23	3
Ethereum	154	Turtlecoin	3	2014	223	281
Electroneum	140	Bytecoin	2	2015	115	1.6K
Mixed			18	2016	461	8.7K
Sub-total			4,442	2017	3.8K	31K
With other identifiers:				2018	1.3K	6.2K
Email			4,887	2019	1*	49*
Unknown			2,159	~19?	1.7K	14K
TOTAL			11,387	ALL	7.6K	62K

Table 2: Leftmost side of the table: Number of campaigns per currency, amount of e-mails and unknown identifiers (i.e., not associated with a known currency). Rightmost: Number of samples (*partial data) seen in a given year for Bitcoin (BTC) and Monero (XMR).

Domains	**#Samples**	**#URLs**
github.com	163	388
*.amazonaws.com	85	396
www.weebly.com	80	96
*.google.com	38	74
hrtests.ru	37	1
cdn.discordapp.com	34	55
a.cuntflaps.me	32	48
file-5.ru	30	52
TOTAL #: 2755(# domains)	3420	6949

Table 3: Excerpt of domains hosting known mining malware, number of samples hosted under each domain and number of URLs hosting those samples.

as Bitbucket or 4sync, and web hosting sites such as Amazon (AWS), Google, or Dropbox. One can also find mining malware hosted through torrent sites (*b-tor.ru*), entertainment sites (*telekomtv-internet.ro*), or hosted as attachments in the Discord app, a voice and text chat (*cdn.discordapp.com*). There are also URL-shortener sites (*goo.gl*). This altogether shows that crypto-miners largely rely on publicly available third-party servers. The use of these services provides an economical incentive when compared to other approaches that use dedicated infrastructure such as bullet-proof servers — that are more resilient against take-downs.

Our longitudinal analysis confirms previous reports positioning Monero as the preferred currency used by miscreants for crypto-mining malware [7]. Thus, in the rest of the paper we focus our attention on campaigns using Monero.

4.3 Mining Pools

There are two possible strategies for mining: joining a pool or mining alone (which we call *solo-mining*). Using mining pools instead of "*solo-mining* strategies" has several advantages: it increases the chances of receiving payments for mining and reduces the need for specialized mining equipment. Selecting a mining pool is not straightforward because it depends on many dynamic factors such as the current hashrate of the pool, or the complexity required for mining. Pools with a high number of workers are more likely to mine a block faster, but the reward received is lower. To understand the popularity of the different mining pools among criminals, we look at the number of wallets and the amount of XMR mined over the most consolidated pools (according to various benchmarks such as http://moneropools.com, or https://minexmr.com/pools.html) that provide public information about the wallets. Table **??** provides a list of these pools ranked by popularity among criminals (in terms of earnings). We show that the most popular pools are *crypto-pool* and *dwarfpool*, with more than 429K and 168K XMR mined each. When looking at the number of wallets observed, the most common pool used is *minexmr*, with (at least) 608 wallets. An interesting pool not included in our analysis is *minergate*. We have found 4,980 emails mining at this pool in our dataset. Since *minergate* does not provide public information about the rewards paid to the miners, we are unable to estimate profits from this pool.

Our analysis show that 49.3% of the campaigns use or have used more than one pool. Figure 5 shows the number of pools used by different campaigns grouped by the amount of Monero mined. As it can be observed, the 97% of the campaigns with largest earnings (i.e., over 1K XMR) have used more than one pool. However, seven campaigns with earnings over 10K are using just one pool. Out of these, six use *dwarfpool* and one uses *crypto-pool*. This suggests that mining in different pools depends on different strategies, probably driven by the revenues from each pool and their banning policies.

4.4 Monero-based Campaigns

As shown in Table 2, we find 2,360 campaigns mining Monero. Out of those, we are able to get payments to 2,145 campaigns through querying the various mining pools

We summarize the results of our aggregation in Table 5 and show some demographics for the top 10 campaigns. Note the difference between USD and XMR in some campaigns. As explained before, this is due to fluctuations of the XMR value and depends on when payments were made. A note of precaution when looking at the USD figures as we are unaware of when criminals cash-out their moneros. Thus, we prefer to report our findings in primarily in XMR. Overall, we estimate that there are at least 2,142 campaigns that have

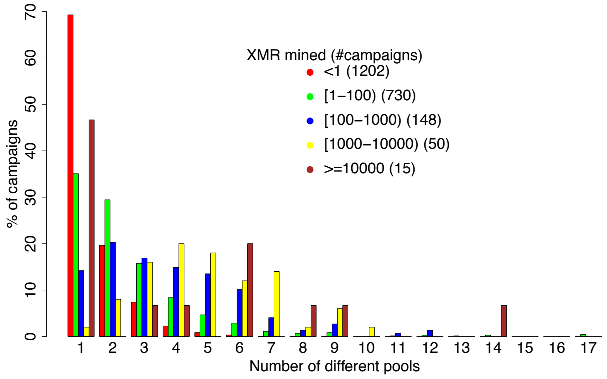


Figure 5: Number of different mining pools used by the campaigns and grouped by earnings

Pool	XMR Mined	#Wallets	USD
crypto-pool	429,393	487	47,261,821
dwarfpool	168,796	461	1,088,516
minexmr	74,396	608	5,320,397
poolto	29,044	38	35,815
prohash	12,833	54	275,471
nanopool	5,205	375	858,949
monerohash	4,046	217	477,557
ppxxmr	3,860	185	518,487
supportxmr	3,217	241	443,087
Others (8)	2,797	314	325,034

Table 4: Overview of the popularity of the different mining pools ranked by the amount of XMR mined by malware.

Campaign	**#S**	**#W**	**Period**	**XMR**	**$**
C#599	66	7	06/16 to active*	163,756	20M
C#2881	20	2	10/16 to 04/18	59,620	8M
C#254	134	4	01/15 to 02/19	42,069	323K
C#6	18290	304	08/14 to active*	41,623	2M
C#97	59	1	09/14 to 04/18	32,886	53K
C#1227	91	14	06/16 to active*	27,086	1M
C#8909	6	1	09/16 to 04/18	23,300	2M
C#3150	9	1	06/16 to 05/18	22,520	5M
C#2510	46	1	09/14 to 04/18	21,389	42K
C#2101	25	1	09/14 to 04/18	20,694	38K
TOP-10	19K	336	2014/08/30 - *	454,943	39M
ALL-2142	68K	2747	2014/07/18 - *	741,094	58M

Table 5: Top 10 campaigns ranked by amount of XMR mined. C=Campaign #S=Num. of samples, #W=Num. of wallets, and active* on `April 2019`. Recall that the exchange rate to USD is computed dynamically based on when the payments were made.

Tool	**#I**	(#S)	#V	#C
xmrig	415	(299)	59	262
claymore	861	(853)	14	98
niceHash	108	(21)	11	67
learnMiner	2	(2)	2	2
ccminer	1	(1)	1	1

Table 6: The most popular mining tools used. I=Instances, S=Samples, V=Versions, C=Campaigns

accumulated about 741K XMR (58M USD). Some of them are still active. Interestingly, just a single campaign (C#599) has mined more than 163K XMR (20M USD), which accounts for about 22% of the total estimated. This campaign is still active at the time of writing and it is later studied in §5.1. We observe that only the top 10 campaigns mine more than the remaining 2,132 ones. Overall, we observe that 99% of the campaigns earn less than 100 XMR (see Figure 4). We also observe that while majority of the campaigns earn very little, there are a few campaigns overly profitable. This indicates that the core of this illicit business is monopolized by a small number of wealthy actors.

There are campaigns with a large number of samples, with up to 18K in the case of C#6. However, some of the most profitable campaigns have few samples (e.g., C#3159 or C#8909). This means that either the samples in those campaigns have infected a large number of victims, or that other samples from the campaign are not detected by any AV. In either case, it suggests that there are some miscreants that are proficient in remaining undetected. In the next section we analyze the infrastructure and stealth techniques used by the different campaigns, and how this affects their efficiency.

While most of profitable campaigns started in 2016 or earlier, we observe recent campaigns with large earnings. In particular, 21 campaigns that started in 2018 have mined more than 100 XMR, 12 of which are active at the time of writing (April 2019).

4.5 Infrastructure

We next analyze the third-party infrastructure used in the different Monero campaigns.

Mining software. Table 6 shows an overview of the stock mining tools used by the different campaigns. We show that `xmrig`, `claymore` and `niceHash` are the most popular tools we account for. With the current distance threshold in our Fuzzy Hashing algorithm, we found no evidence pointing to the use of other less popular tools such as: `cast-xmr`, `jceMiner`, `srbMiner`, or `yam`. When using a higher threshold, we found one campaign using `xmr-stak`. Overall, the top most popular frameworks account for approximately 18% of the Monero campaigns. Note that obfuscated versions of these tools are sold in underground marketplaces. Thus, these numbers should be viewed as a lower approximation.

Domain aliases for mining pools. A common mitigation strategy often suggested in commercial reports [24] is to block known mining pools, using blacklists. Criminals create CNAME domain aliases to evade this mitigation. In our analysis, we observe 215 different CNAMEs. Most of these are aliases of *minexmr* (176), *crypto-pool* (21) and *dwarfpool* (14). Interestingly, there are two aliases (*x.alibuf.com* and *xmrf.fjhan.club*) which have been eventually used to hide two different pools each. This suggests again dynamic changes in the mining strategy used by criminals to maximize their revenue. We note that the former alias is actually part of the most profitable campaign (C#599), which is detailed in §5.1.

Pay-Per-Install services. In order to spread malware, criminals use commodity botnets offered as PPI services in underground markets [2, 40]. We find samples from 3 different botnets offering PPI services. In particular, we observe 511 samples associated with the Virut botnet (in 44 different campaigns), 46 from Ramnit (in 10 campaigns) and 27 from Nitol (in 3 campaigns). Also, in one of the biggest campaigns (C#6), known as Photominer [6], we find 349 samples (1.87%) of the samples belonging to this campaign) using Virut to deploy the mining operation. Recall that campaigns are automatically extracted. Observing campaigns from botnets that are know to the community shows that our heuristics provide a reliable aggregation. Yet, our framework steps up finding novel campaigns as shown in §5.

Obfuscation. A common practice when spreading malware is to obfuscate the binary to avoid detection. Criminals typically use existing tools, such as well-known packers (e.g., UPX) or crypters. Packers can be fingerprinted more easily than crypters, but crypters — which are usually purchased in underground markets — increase the cost of the operation. By leveraging the F-Prot unpacker [4], we extract packer information associated with each sample (when applicable). We also measure when binaries are obfuscated by looking at the entropy. In our implementation, we choose a threshold of 7.5 (where 8 means total randomness) to decide when a sample is obfuscated. We found that around 30% of the samples are obfuscated. We consider that a campaign uses obfuscation if a large proportion of their samples (i.e., 80%) are obfuscated. While this is the ratio in the overall dataset, we found that only 4.16% of Monero campaigns use obfuscation. Table 7 summarizes the number of samples using obfuscation together with the tool used to obfuscate it. UPX is by far the most common tool used. Interestingly, we have seen many binaries created using AutoIt (a Windows-based scripting language) which by default packs the script into an PE file using UPX. In §6 we discuss the limitations of analyzing obfuscated binaries.

UPX	328,513	eval	2,032
NSIS	17,468	docwrite	1,490
maxorder	5,988	ARJ	858
SFX	3,931	CAB	721
INNO	2,430	Enigma	710
Others			4,019
Not packed			862,905

Table 7: Packers used for binary obfuscation.

Analysis. Table 8 shows the third-party infrastructure, stealth techniques and period of activities for the different Monero campaigns (both divided according to their profits, and overall). While only 1.1% of the campaigns use domain aliases, a higher proportion is found in most profitable campaigns (10% of those mining between 1K and 10K XMR, and 26.7% of those mining more than 10K XMR). A similar situation happens with proxies and PPI services, which are more common in successful campaigns (i.e., with larger earnings).

Most profitable campaigns have longer period of activity (46.7% have been active since 2014). However, we also observe a high portion of campaigns (26.7%) operating only for 1 or 2 years and still having large profits. We also note the percentage of campaigns active before and after changes in the PoW (Proofs-of-Work): on 06/04/2018, 18/10/2018 and 09/03/2019. These changes require the update of mining software. This means that either botnet operators have to update their bots, or customers of PPI services must buy new installs. We show that most of the campaigns stopped due to PoW updates: around 71.7% in April 2018, 88.2% in October 2018 and 96.4% in March 2019.[14] This means that changes in the PoW algorithm might be an effective (though unwitting) countermeasure, as discussed in §6.

4.6 Take-Aways

In summary, the main take-aways of our analysis include:

(1) We observe that it is no longer profitable to mine Bitcoin, and current criminal efforts focus on mining ASIC-resistant currencies. We also show that there are a small number of actors that monopolize the crypto-mining malware ecosystem. Recent works found similar conclusions in web-based cryptojacking [8, 17] and crypto-mining malware targeting Bitcoin [9] (although this study was from 2014, when mining Bitcoin using desktop computers was profitable).

(2) We note that some successful mining campaigns are very complex in terms of the size and infrastructure that supports the campaign. Our data shows that about 12% of these are supported by other underground economies such as third-party Pay-Per-Install botnets. On the contrary, we also observe very profitable mining campaigns that do not

[14] Given that our data is from April 2019, some of these campaigns might not be defunct, since it might take some time to update the miners.

	< 100	**[100-1k)**	**[1k-10k)**	**>10k**	**ALL**
#Campaigns	**1932**	**148**	**50**	**15**	**2,145**
THIRD PARTY INFRASTRUCTURE					
PPI	1.3%	4.1%	6.0%	20.0%	1.7%
Mining SW	8.6%	16.2%	28.0%	20.0%	9.7%
Both	0.5%	2.0%	4.0%	6.7%	0.7%
STEALTH TECHNIQUES					
Obfuscation	4.0%	5.4%	4.0%	0.0%	4.1%
CNAMEs	0.4%	5.4%	10.0%	26.7%	1.1%
Proxies	2.3%	6.8%	6.0%	20.0%	2.8%
PERIOD OF ACTIVITY					
+ Apr-18	25.1%	60.8%	52.0%	33.3%	28.3%
+ Oct-18	9.9%	31.1%	22.0%	33.3%	11.8%
+ Mar-19	2.7%	12.8%	6.0%	20.0%	3.6%
Start: 2014	0.3%	4.7%	10.0%	46.7%	0.3%
Start: 2015	0.3%	2.0%	4.0%	13.3%	0.3%
Start: 2016	5.4%	25.7%	42.0%	40.0%	4.9%
Start: 2017	38.3%	54.7%	44.0%	0.0%	34.5%
Start: 2018	51.0%	12.2%	0.0%	0.0%	45.9%
Start: 2019	0.3%	0.7%	0.0%	0.0%	0.3%
Years: 0	68.0%	6.1%	0.0%	0.0%	61.2%
Years: 1	29.4%	60.8%	42.0%	6.7%	26.5%
Years: 2	2.4%	26.4%	42.0%	20.0%	2.1%
Years: 3	0.3%	3.4%	6.0%	20.0%	0.2%
Years: 4	0.0%	3.4%	8.0%	46.7%	0.0%
Years: 5	0.0%	0.0%	2.0%	6.7%	0.0%

Table 8: Summary of infrastructure, techniques and period of activity for the different campaigns targeting Monero grouped by profit.

appear to use a large supporting infrastructure. Instead, they are effective campaigns (due to their long lifetime) with obfuscation and novel evasion techniques, e.g.: using CNAMEs to bypass blacklist-based detection.

(3) We estimate that the malicious ecosystem has currently mined at least 4.37% of the total Monero in circulation (approximately 58M USD). These numbers must be added to estimations made using web-browser cryptojacking in paralell work.

(4) It is common to see campaigns mining in various pools. We observe that the most popular mining pools are *crypto-pool*, *dwarfpool* and *minexmr*. We show that a large number of samples mine to *minergate*, an opaque mining pool for which there is no publicly-available information about the rewards received.

(5) When looking at the activity period, we find long-lasting campaigns — some of which are active at the time of writing. In particular, we can see multi-million campaigns operating for a continuous period of five years (see Top-10 in Table 5). This shows that AVs have not addressed this threat appropriately. We argue that crypto-mining malware has not been given enough attention by the industry and the research community and novel countermeasures are required as discussed in §6.

5 CASE STUDIES

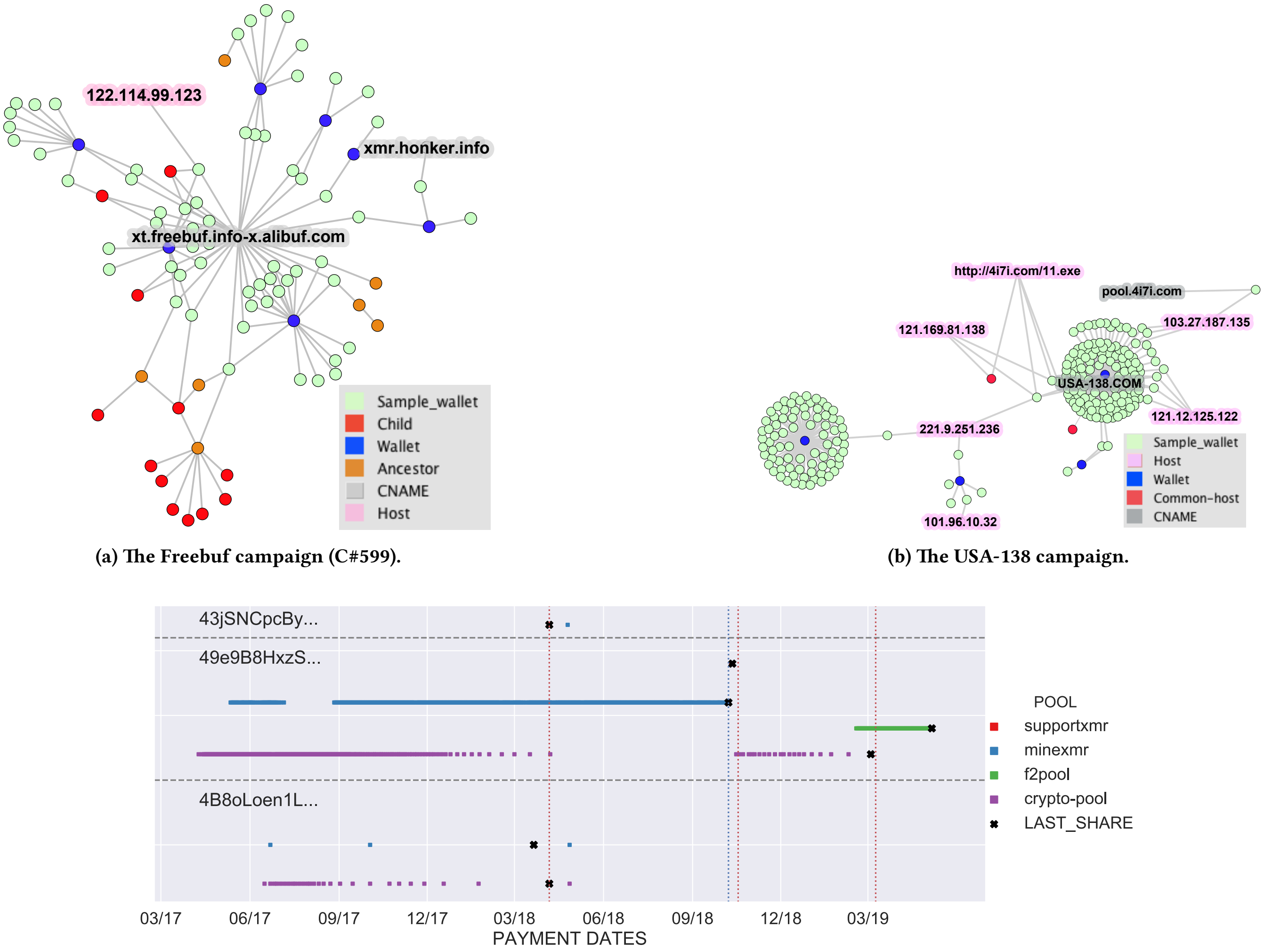
We next present two case studies related to high-profit campaigns that have not been previously reported. Fig. 6 presents an overview of how the campaigns are structured. In the graphs, nodes in blue represent wallets and nodes in light-green represent malware miners. Thus, various light-green nodes connected to a blue node represent a group of samples using the same wallet. Nodes in gray and pink represent the infrastructure of the campaign, with gray nodes portraying contacted domain servers and the pink ones the malware hosts. Finally, ancillary malware are depicted in red and orange. The edges represent the connections described in §3.5.

5.1 The Freebuf Campaign

The most profitable Monero campaign (C#599 in Table 5) has mined more than 163K XMR in 3 years using 7 wallets. We have named it 'Freebuf' because the main grouping feature is the domain *xt.freebuf.info*, which is an alias (CNAME) of the *minexmr* pool.

Structure. Figure 6 shows how the campaign is structured. We observe that the aggregation is dominated by three grouping features: (i) same identifier, (ii) ancestors, and (iii) domain aliases (CNAMEs). Interestingly, the graph of this campaign reveals groups of samples with wallets that reach out to one another through paths traversing these three grouping features. In other words, the three grouping features are the key to map the structure of the campaign. In particular, there are two domains linked through a common wallet: *x.alibuf.com* and *xmr.honker.info*, which in turn are connected to *xt.freebuf.info*. Note that both have been aliases of *minexmr*, and *x.alibuf.com* has also been an alias of *crypto-pool*. We can observe that the backbone of the graph is established through connections of samples linked to CNAMEs. From there, there are other clusters that are linked via *same identifier*, and to those, there are some samples which are connected by common ancestors.

Payments. By analyzing the different payments received through time, we observe that before the update of the PoW in April 2018, this campaign was mining in various pools simultaneously. However, after the update all mining efforts were put into *minexmr*. In September 2018 we reported the wallets to the largest pools, resulting in two wallets being banned in October 2018. Upon request, one pool operator kindly provided us with statistics regarding the number of different IPs behind the wallets. The two banned wallets connected from 5,352 and 8,099 different IPs and had mined 362.6 and 1,283.7 XMR respectively. As a consequence of banning, we observe that the campaign operator decided to



(a) The Freebuf campaign (C#599).

(b) The USA-138 campaign.

(c) Payments per wallet for USA-138.

Figure 6: Overview of the analysis of our case studies. For Fig. 6c, the dashed red lines correspond to the two changes in the Monero PoW algorithm. Dashed blue lines show the day when the wallets were banned in *minexmr*.

move their mining efforts to another pool (*ppxmr*) — which indeed was used before the update on April 2018. We visualize these payments in Figures 7 and 8 from Appendix A.5. We have seen that as a result of this intervention, together with the change in the PoW algorithm in October 2018, the payments received by the wallets associated to this campaign have been considerably reduced.

5.2 The USA-138 Campaign

We have also named this campaign following the name of the CNAME that characterizes this campaign: *xmr.usa-138.com*. Overall, this campaign has mined (at least) 7,242 XMR and it has 137 samples and 4 wallets. None of the samples from the campaign are known mining software and the campaign does not use proxies. Recall that this campaign is still active at the time of writing in December 2018. Out of all the samples, 43 are obfuscated with UPX.

Structure. Figure 6b shows how this campaign is structured. We see that there are two clusters of samples connected through a common host (221.9.251.236). This host, which operates in China, is still online at the time of writing and it is still hosting the malware used to run the criminal operation. One of the wallets (on the left-most side of the graph) is an Electroneum (ETN) wallet (another crypto-currency based on the CryptoNote algorithm). The remaining wallets are Monero. We observe a domain, *4i7i.com*, which is used both as a domain alias (*pool.4i7i.com*) for *crypto-pool*, and as a malware host (e.g., *hxxp://4i7i.com/11.exe*). We note that the samples using the Electroneum wallet connect to a pool in *etn.4i7i.com*. This domain is probably the domain alias of a Electroneum pool, but we are unable to find passive DNS data.

Payments. As mentioned, the campaign has mined 6,709 XMR (around 651K USD). When looking at the amount of earnings made with the Electroneum wallet, we observe earnings of 314,18 ETN. The equivalent of this is currently less than 5 USD. While this might seem little at the moment, it might be worth much more in the future. Regarding Monero, this campaign mainly relies on *minexmr* and *crypto-pool*. As depicted in Figure 6c, the most active wallet (named as *49e9B8H...*) operated mainly with *minexmr* after the April 2018 update. Once again, we reported the wallets to the pools and soon after we found a similar behavior as with the Freebuf campaign: after the wallet was banned in *minexmr* the malware operators moved to *crypto-pool* again. Different to Freebuf, this campaign 'survived' the PoW change in October 2018 and is still receiving payments from this pool.

When looking at the number of connections made by the samples using the wallet *49e9B8H...* to the *minexmr* pool, we observed over 13K IPs. These samples have been mining after the update in April'18. This indicates that the campaign uses a medium-sized botnet which is kept updated.

6 DISCUSSIONS

In this section, we analyze existing countermeasures, presenting their main challenges and weaknesses and looking at potential directions to address this challenge. We also discuss the limitations of our work.

Reporting illicit wallets. Reporting illicit wallets to the pools, while being a common — and important — practice [7], is not an effective countermeasure. First, it is costly and requires cooperation and coordination from all (or at least the main) pool operators. Second, criminals have developed mechanisms to bypass detection (e.g., using mining proxies). During our study, we have reported the illicit wallets we found to the largest pools, together with evidence of criminal behavior. We found two issues. First, there are non-cooperative pools that chose not to ban wallets found within crypto-mining malware; and second, those that are cooperative pools tend to err on the safe side. For example, the pool *minexmr*, while being remarkably cooperative, has a policy that only blocks wallets with a large number of associated connections. Recall that there are criminals that leverage on a small set of machines (i.e., proxies) to hide botnet-related mining activity. Thus, only banning botnet-related mining activity proves ineffective.

Additionally, we found that many successful campaigns use several pools at the same time. While this practice has drawbacks for criminals (mining workers compete with each other), it also makes their campaigns more resilient to take-down operations. Criminals respond to such take-downs by changing the mining pool being used (as we have seen in the case studies presented in §5) or by creating new wallets and setting proxies up [11].

Changes in the Proof-of-Work algorithm. ASIC-based mining uses hardware support customized for specific algorithms to compute the faster PoWs [19]. Frequent changes in the algorithm are intended to hinder ASIC-based mining, due to the cost of creating new hardware with each change. These changes also require updates in the software, which is straightforward for benign miners. However, in the case of crypto-mining malware, it requires botnet operators to update their bots. In turn PPI users have to purchase further installs to push the updated version of their miners. We have monitored two changes in the PoW of Monero in 2018 (April, 6th and October, 18th). In each change, about 73% and 90% of the campaigns ceased their operations. Due to the cost of updating the mining infrastructure, we observe a large number of campaigns not providing valid shares after changes in the mining algorithm (i.e. due to mining with an outdated algorithm). This does not mean that the mining has ceased: a non-updated miner does not provide valid hashes, but it is still mining using infected computers. Thus, the victim is still being harmed as long as the mining continues.

Although changes in the algorithms do not dismantle consolidated campaigns, they can dissuade new ones. Thus, a potential countermeasure against crypto-mining malware is to increment the frequency of such changes, and design these changes to not only be anti-ASIC, but also anti-botnet.

Security by design and liability. In addition to non-cooperative pools, we have observed that a large number of samples mine through opaque pools. We position that the community should work on devising protocols that enforce transparency to nodes that act as pools. Mining is a process which requires cooperation from the miners, who get rewards in exchange. Similarly, a potential solution would be to design protocols or techniques that reward transparent and cooperative pools. Likewise, pools could also improve the mechanisms they use to detect malicious miners, and the actions they take against them. Additionally, we argue that a legal framework should be formulated to regulate the pool industry.

Third-party Infrastructure. We observe that most mining campaigns use third-party infrastructure, both illicit and legitimate. The former includes PPI services, malware packers or private mining pools allowing botnets to mine. The latter includes cloud hosting services such as Dropbox or AWS, and stock mining tools such as *xmrig* or *claymore-tool*. During our analysis, we have observed proficient campaigns making use of both types. Using public OSINT, we have analyzed

IoCs observed in the samples to associate them to known third-party infrastructure. In particular, we have observed three *known* PPI services (i.e. Virut, Nitol and Ramnit) used by campaign operators to spread their miners across 355K malware variants using *known* packers. However, we have learned about third-party infrastructure (e.g., non-reported botnets, custom malware obfuscators or bullet-proof hosting servers) being offered in underground markets. Thus, a limitation of our approach is that campaigns that use such *unknown* third-party infrastructure (i.e., for which there is no OSINT information) can be grouped together. Detecting such new third-party infrastructure is out of the scope of this study, since this requires investigating each campaign separately and applying other type of intelligence — by further investigating tools exchanged in underground markets or infiltrating these campaigns. Having said it, we have argued that having campaigns grouped by unknown third-party infrastructure, while it hinders our ability to account for the number of individual actors, it is nonetheless useful for law enforcement when prioritizing take-downs and for security practitioners when understanding the magnitude of the problem and devising novel mitigation strategies.

Quality of the Ground-Truth. One obvious countermeasure to this threat is to keep educating users to have updated AVs. This countermeasure requires AVs — provided the magnitude of the threat — to have a very comprehensive dataset of signatures. However, judging by the activity period of some of the campaigns we have seen that AV vendors have not been up to this task. This is a known limitation of the Threat Intelligence industry [21].

Likewise, in our work, we rely on different independent AV vendors to set our ground-truth. This might introduce two types of errors that limit our work: False Positives (FP), where a legitimate sample is erroneously flagged as malware, and False Negatives (FN), where malware is flagged as legitimate. Moreover, the boundaries between 'malicious' and 'legitimate' samples are unclear: a legitimate mining software, when used maliciously in infected computers (i.e. without user consent) might be considered malware (and indeed, most AV classify these tools as malware). We note that establishing an optimal trade-off between benign and malicious mining is not straightforward [8, 17].

We deal with FP by setting up a relatively high number of positive detections (i.e., 10 AVs).[15] However, by doing this we are also introducing FN. In this paper, we err on minimizing the number of FP knowing that our findings have to be seen as an under-approximation of the current threat. We would like to explore a more greedy trade-off by setting the number of positive detections to lower values (e.g., 5 AVs). We argue that this should not incur into many FPs as we have introduced a white-listing of known mining tools — which are more prone to be misclassified. In addition, we note that these tools do not contain hard-coded wallets and/or Stratum connections are not observed when run alone in a sandbox. Exploring this in detail is precisely the scope of our future work.

Anti-analysis Techniques. A limiting factor of the quality of the ground-truth is the ability of malware to hinder detection. On the one hand, malware uses obfuscation to thwart static analysis. We have seen that this generally comes in the form of packers and crypters, although advanced adversaries might be using polymorphic or metamorphic malware.

Like in many of the current countermeasures, we have partially addressed this in two ways. First, we looked at the usage of known packers and saw that a small number of campaigns used them. As not all packing algorithms are known, and thus not all samples can be unpacked, we have also looked at the entropy of the binaries.

On the other hand, malware uses evasion techniques to thwart dynamic analysis. While there are many forms of evasion we are vulnerable to (e.g., sandbox detection [27]), we put special efforts to address those targeting the crypto-mining malware realm in particular. Specifically, we have attempted to de-anonymize domain aliases that masked connections to mining pools (c.f., §3.5). However, despite our efforts, our study inherits the limitations of both static dynamic analysis and thus can unavoidably miss samples from advanced adversaries. A way to cope with other advanced adversaries that use other forms of evasion, such as anti-emulation techniques, would be to use bare metal solutions [15].

Mining-tailored Solutions. One common strategy when assuming adversaries leveraging advanced obfuscation and evasion techniques, is to devise solutions that are tailored to the type of threat. Since miners have a distinctive CPU usage, one can rely on this to build an anomaly detection system for crypto-malware. Related works rely on modeling of the CPU usage [17] or on instrumenting web browsers to detect suspicious activity [13, 41]. While these approaches are effective for web-based cryptojacking, these types of defenses are not effective with crypto-mining malware for one reason: the malware controls the infected computer and thus it can evade any local defenses (e.g., by acting as a rootkit and tampering with the CPU monitoring module). Other works propose to monitor the CPU usage of a computer from a hypervisor to protect Infrastructure-as-a-Service clouds [36]. However, this approach focuses on protecting cloud providers, and it is not applicable to end-users. An alternative is to offload the usage monitor to an external system and look at the energy consumption fingerprint. While power-aware anomaly detection systems have been proposed to detect smartphone

[15] There is one exception to this: we keep a sample with less than 10 AV positives when it contains a wallet observed in another sample having 10 or more AV positives.

malware in general [14, 35], we are not aware of a solution tailored to crypto-mining malware for general-purpose computers. We position that these solutions could be deployed by electric-companies to end-users with smart-meters.

7 RELATED WORK

Illicit mining has been a threat since the emergence of Bitcoin in 2009. However, it has not been properly addressed in academia until recently. The first analysis of crypto-mining malware was published in 2014 by Huang et al. [9]. Authors analyzed botnets and campaigns mining bitcoins. They found that malicious malware mined at least 4.5K bitcoins (which was worth around $3.2M in 2014). Since mining bitcoins using end-user computers is no longer profitable, both crypto-currency malware and web-based cryptojacking rely on cryptocurrencies resistant to ASIC mining, such as Monero or Bytecoin. Thus, most of the illicit mining focuses on Monero nowadays [7, 8, 17]. Recent works analyzed web-based mining, both as an alternative to advertisements to monetize web content [30, 33] and as cryptojacking, where mining is done without the consent of users [8, 17, 34]. Konoth et al. analyzed the Top 1M Alexa sites looking for web-browser cryptojacking [17]. They used a mixture of code analysis and network monitoring to identify whether a web is trying to connect to a mining pool using the Stratum Protocol. Hong et al. proposed a dynamic analysis method to detect cryptojacking in web content [8].
To distinguish cryptojacking from benign mining, it is important to properly identify user consent. One approach is to search for keywords indicating mining activity [8]. This approach misses informed consent acquired by other means, such as images or additional documents. Thus, some works also look for AuthedMine scripts, which require explicit action from users to start mining [13, 17]. In our work, we rely on AV reports and other heuristics to classify binaries into malware or goodware.

Previous works are characterized by the simplicity in which they aggregate campaigns. In particular, related works mostly look at mining pool identifiers (e.g., wallets) alone [8, 13]. However, criminals use concurrent miners with different identifiers to retake operations when wallets are banned [11]. Konoth et al. includes information about the servers when performing the aggregation [17]. Unfortunately, this does not scale as it requires manual efforts vetting the code of the scripts (i.e., to get the verification code). In their analysis of Bitcoin, Huang et al. use information gathered from the Blockchain to aggregate campaigns [9]. However, this approach is not valid for crypto-currencies that obfuscate transactions (e.g., Monero or Zcash).

The overall earnings obtained from malicious mining have increased in the last years. For example, Konoth et al. discovered 1,735 domains, estimating overall revenues of $188,878 per month. In parallel work, Hong et al. detected 2,770 domains, estimating overall revenues of $1.7M. However, estimations obtained from web-browser cryptojacking are not reliable. This is because analyzing profits from web activity relies on estimates of i) the number of monthly visitors, ii) the time spent by each visitor on average, and iii) the type of device they use. Instead, we are able to get wallets used by the malware and the payments given by the pools as a reward. This allows us to analyze not only the earnings of each wallet, but also the pools used for mining and the exact dates of the payments. Our findings increase the understanding of this threat. In particular, we estimate that earnings are — at least — 56 million USD obtained in 4.5 years of operation (more than 1M/month). Table 9 summarizes the related works and compares each of the measurements.

8 CONCLUSION

In this paper, we have presented a longitudinal large-scale measurement study of crypto-mining malware, analyzing samples spanning over more than a decade. We show that Monero is currently the preferred currency used by criminals, who have obtained massive earnings. This criminal activity is rooted within the underground ecosystem, which allow criminals to externalize operations, e.g. to avoid AV detection using packers or to spread their malware through PPI. Through static and dynamic analysis, we extract information from the samples which is used to group them into campaigns. While some campaigns rely on third-party infrastructure, others use simple and effective evasion mechanisms such as *domain aliases*. Our profit analysis on Monero shows that a small number of actors hold sway the mining illicit business. Using crypto-mining malware, criminals have mined (at least) 4.4% of the moneros in circulation, earning up to 56 million USD. One of the main reasons of the success of this criminal business is its relatively low cost and high return of investment. Also, since it is considered a lower threat to their clients, the AV industry has not paid due attention. Our findings complement related studies focused on Bitcoin and web-based cryptojacking, corroborating that malicious crypto-mining is a growing and complex threat that requires effective countermeasures and intervention approaches. Due to the need of updating mining software, our findings suggest that regular changes in the PoW algorithm might discourage criminals, since this will increase the cost of acquisition (e.g., customers of PPI will need to buy new installs) and maintenance of their botnets. Finally, we present technical details about the way this ecosystem operates and discuss open challenges and countermeasures. In particular, we analyze two novel campaigns and fully release the data of all campaigns to foster research in the area.

	Focus (currency)	Size		Profits
		Analyzed	Detected	
Huang et al. [9] (2014)	Binary-based mining (BTC)	Unknown	2K crypto-mining malware	14,979 BTC
Ruth et al. [33] (2018)	Web-based mining (XMR)	10M websites	2,287 websites	1,271 XMR/month
Hong et al. [8] (2018)	Web-based cryptojacking (XMR)	548,624 websites	2,270 websites	7,692.30 XMR
Konoth et al. [17] (2018)	Web-based cryptojacking (XMR)	991,513 websites	1,735 websites	746,55 XMR/month
Papadopoulus et al. [30] (2018)	Web-based mining (XMR)	3M websites	107.5K websites	N/A
Musch et al. [28] (2018)	Web-based cryptojacking (XMR)	1M websites	2.5k websites	N/A
Our work	**Binary-based mining (various)**	**4.5M malware samples**	**1.2M crypto-mining malware**	**741K XMR (15K XMR/month)**

Table 9: Summary of related work. *Last row shows our measurement.

REFERENCES

[1] C. Beek, T. Dunton, S. Grobman, M. Karlton, N. Minihane, C. Palm, E. Peterson, R. Samani, C. Schmugar, R. Sims, D. Sommer, and B. Sun, "Mcafee Threat Report," June 2018.

[2] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring pay-per-install: the commoditization of malware distribution." in *Usenix security symposium*, 2011, pp. 13–13.

[3] Checkpoint, "Jenkins miner: one of the biggest mining operations ever discovered," February 2018. [Online]. Available: https://perma.cc/SVN4-C5B4

[4] Cyren, "F-prot antivirus," 2014. [Online]. Available: http://www.f-prot.com

[5] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark, "A first look at browser-based cryptojacking," in *IEEE Security & Privacy on the Blockchain*, 2018.

[6] D. Goldberg, "Guardicore: The Photominer campaign," June 2016. [Online]. Available: https://perma.cc/JE3Y-F42L

[7] J. Grunzweig, "Palo Alto Networks: The Rise of the Cryptocurrency Miners," June 2018. [Online]. Available: https://perma.cc/4VZL-J45Q

[8] G. Hong, Z. Yang, S. Yang, L. Zhang, Y. Nan, Z. Zhang, M. Yang, Y. Zhang, Z. Qian, and H. Duan, "How you get shot in the back: A systematical study about cryptojacking in the real world," in *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 2018.

[9] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. McCoy, S. Savage, N. Weaver, A. C. Snoeren, and K. Levchenko, "Botcoin: Monetizing stolen cycles." in *NDSS.* Citeseer, 2014.

[10] Kafeine, "Proofpoint: Adylkuzz cryptocurrency mining malware spreading for weeks via EternalBlue/DoublePulsar," June 2017. [Online]. Available: https://perma.cc/3V7G-CDEN

[11] ——, "Proofpoint: Smominru Monero mining botnet making millions for operators," January 2018. [Online]. Available: https://perma.cc/V5UR-TDLU

[12] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. D. Tygar, "Better malware ground truth: Techniques for weighting anti-virus vendor labels," in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security.* ACM, 2015, pp. 45–56.

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

[14] H. Kim, J. Smith, and K. G. Shin, "Detecting energy-greedy anomalies and mobile malware variants," in *Proceedings of the 6th international conference on Mobile systems, applications, and services.* ACM, 2008, pp. 239–252.

[15] D. Kirat, G. Vigna, and C. Kruegel, "Barecloud: Bare-metal analysis-based evasive malware detection." in *USENIX Security Symposium*, 2014, pp. 287–301.

[16] C. Kolbitsch, E. Kirda, and C. Kruegel, "The power of procrastination: detection and mitigation of execution-stalling malicious code," in *Proceedings of the 18th ACM conference on Computer and communications security.* ACM, 2011, pp. 285–296.

[17] R. K. Konoth, E. Vineti, V. Moonsamy, M. Lindorfer, C. Kruegel, H. Bos, and G. Vigna, "Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense," in *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 2018.

[18] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digital investigation*, vol. 3, pp. 91–97, 2006.

[19] K. Kurokawa, "Forking for ASIC resistance: A Monero case study," June 2018. [Online]. Available: https://perma.cc/5JL6-RPPS

[20] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis, "A lustrum of malware network communication: Evolution and insights," in *Security and Privacy (SP), 2017 IEEE Symposium on.* IEEE, 2017, pp. 788–804.

[21] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, and K. Levchenko, "Reading the tea leaves: A comparative analysis of threat intelligence," in *Proceedings of the USENIX Security Symposium*, 2019.

[22] Y. Li, S. C. Sundaramurthy, A. G. Bardas, X. Ou, D. Caragea, X. Hu, and J. Jang, "Experimental study of fuzzy hashing in malware clustering analysis," in *8th workshop on cyber security experimentation and test (cset 15)*, vol. 5, no. 1. USENIX Association Washington, DC, 2015, p. 52.

[23] D. Liebenberg, "Cisco Talos: Rocke, the champion of Monero miners," August 2018. [Online]. Available: https://perma.cc/ZH4B-DBG3

[24] D. Liebenberg, C. McFarland, M. Martinez, C. Jerome, F. Gutierrez, A. Giandomenico, T. DeJesus, J. Grunzweig, M. Martinez, A. Brandt, N. Jenkins, and S. Scher, "The illicit cryptocurrentcy mining threat," September 2018. [Online]. Available: https://perma.cc/Z5ZH-H5TG

[25] M. Lindorfer, M. Neugschwandtner, and C. Platzer, "Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 2. IEEE, 2015, pp. 422–433.

[26] M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V. Van Der Veen, and C. Platzer, "Andrubis–1,000,000 apps later: A view on current android malware behaviors," in *Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2014 Third International Workshop on.* IEEE, 2014, pp. 3–17.

[27] N. Miramirkhani, M. P. Appini, N. Nikiforakis, and M. Polychronakis, "Spotless sandboxes: Evading malware analysis systems using wear-and-tear artifacts," in *Security and Privacy (SP), 2017 IEEE Symposium on.* IEEE, 2017, pp. 1009–1024.

[28] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Web-based cryptojacking in the wild," *arXiv preprint arXiv:1808.09474*, 2018.

[29] A. Norton, "Crypto-malware fi?! a look at the latest malware threat," September 2018. [Online]. Available: https://perma.cc/6K3K-E632

[30] P. Papadopoulos, P. Ilia, and E. P. Markatos, “Truth in web mining: Measuring the profitability and cost of cryptominers as a web monetization model,” *arXiv preprint arXiv:1806.01994*, 2018.

[31] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, “CrimeBB: Enabling cybercrime research on underground forums at scale,” in *Proceedings of The Web Conference (WWW).* ACM, 2018.

[32] R. Recabarren and B. Carbunar, “Hardening Stratum, the Bitcoin pool mining protocol,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 3, pp. 57–74, 2017.

[33] J. Rüth, T. Zimmermann, K. Wolsing, and O. Hohlfeld, “Digging into browser-based crypto mining,” in *Proceedings of the 2018 Internet Measurement Conference.* ACM, 2018.

[34] M. Saad, A. Khormali, and A. Mohaisen, “End-to-end analysis of in-browser cryptojacking,” *arXiv preprint arXiv:1809.02152*, 2018.

[35] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and S. Pastrana, “Power-aware anomaly detection in smartphones: An analysis of on-platform versus externalized operation,” *Pervasive and Mobile Computing*, vol. 18, pp. 137–151, 2015.

[36] R. Tahir, M. Huzaifa, A. Das, M. Ahmad, C. Gunter, F. Zaffar, M. Caesar, and N. Borisov, “Mining on someone elsefis dime: Mitigating covert mining operations in clouds and enterprises,” in *International Symposium on Research in Attacks, Intrusions, and Defenses.* Springer, 2017, pp. 287–310.

[37] T. I. Team, “Avast: Greedy cybercriminals host malware on GitHub,” March 2018. [Online]. Available: https://perma.cc/DV7R-Q7CQ

[38] K. Thomas, D. Yuxing, H. David, W. Elie, B. C. Grier, T. J. Holt, C. Kruegel, D. Mccoy, S. Savage, and G. Vigna, “Framing dependencies introduced by underground commoditization,” in *In Proceedings of the Workshop on Economics of Information Security (WEIS*, 2015.

[39] A. Vamshi, “Netskope: Technical analysis of Xbooster parasitic Monero miner,” May 2018. [Online]. Available: https://perma.cc/8RZG-5QBS

[40] R. Van Wegberg, S. Tajalizadehkhoob, K. Soska, U. Akyazi, C. H. Ganan, B. Klievink, N. Christin, and M. Van Eeten, “Plug and prey? Measuring the commoditization of cybercrime via online anonymous markets,” in *27th USENIX Security Symposium.* USENIX Association, 2018, pp. 1009–1026.

[41] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao, “SEISMIC:SEcure In-lined Script Monitors for Interrupting Cryptojacks,” in *European Symposium on Research in Computer Security.* Springer, 2018, pp. 122–142.

[42] Webroot, “Webroot threat report: Mid-year update,” September 2018. [Online]. Available: https://perma.cc/3M2Z-Q76Y

A APPENDIX

A.1 Ethical issues

Most of the data collected is publicly available. However, both Palo Alto and Virus Total shared some non-public information with us and we requested their permission to use it for this paper. Another ethical concern relates to the implications of reporting any misuse activity to the pools. In particular, we have reported evidence of wallets seen in crypto-mining malware. Our actions might produce an intervention over the reported users due to criminal activity. This entails potential ethical implications when non-criminal wallets are mistakenly banned. We have taken due precautions to guarantee that we only report wallets of samples used by malware as discussed in the paper. Thus, our study has been approved by the designated ethics officer at the Reseach Ethics Board (REB) of our institution.

In addition to our precautions, we have provided the pools with accompanying evidences proving illicit activity, including a pointer to the Virus Total report. However, despite our involvement, the final decision of banning the wallets relies on the pool operators. These operators have additional insights about the *modus operandi* of their users (e.g., the number of IP addresses that are currently mining with a wallet) that can be used to further corroborate any type of misuse. In fact, it is our understanding that the pools that took actions against some of the reported users based their decision solely on the number of connections per wallet. Multiple connections from the same wallet evidences the use of a botnet, which it is against the terms and conditions of some of the pools. However, we note that a botnet can always hide behind a single IP addressed using proxies. Also, we have observed that the pools do not proactively ban wallets that display botnet-like activity. We discuss the ethical implications of the banning policies of the pools in §6.

A.2 Crypto-mining Trends in Underground Forums

As discussed in §2, the underground economy plays a key role in the proliferation of the malicious crypto-mining threat. After analyzing our dataset of posts collected from several underground forums, we observe a large number of posts discussing the use of illicit crypto-malware mining. Figure 1 shows the proportion of threats discussing the use of crypto-malware mining across time for different crypto-currencies. We observe that Monero is the most prevalent crypto-currency in 2018. We also observe that while Bitcoin was the most popular crypto-currency among illicit miners, its popularity has dropped over time. We also note that the actors of the underground economy have experimented with other less popular crypto-currencies such as Dogecoin or Litecoin during the 2013 and 2014. However, criminals shifted to Bitcoin and Monero probably when they realized that their value was becoming more profitable.

We have studied a large portion of these posts as discussed in §2. We showed that cybercrime commoditization is key to the wealth of illicit crypto-mining. Figure 2 provides evidence of this. In particular, it shows an advertisement posted in one of the largest English-speaking underground forums, offering a “Silent XMR” (i.e. using obfuscated binaries) Botnet. Among others, one of the characteristics claimed is the use of xmrig as a mining tool with support for proxies. We have seen a wide range of similar adverts in our analysis of underground economies.

A.3 Other Sources

Although our data collection originated from all the data sources described in §3.1, we later discovered overlaps between the different sources. During our dataset consolidation, we observed that Virus Total, Palo Alto Networks, Virus Share, and Hybrid Analysis together accounted for (at least) all the samples observed in the remaining sources. We also observed that most of the malicious miners appear in these four datasets. For this, we consider them to be the main sources for collecting binaries. For simplicity, we only refer to `Virus Total`, `Palo Alto Networks`, `Virus Share`, and `Hybrid Analysis` when labeling the source of the dataset in this paper. However, we highlight that the alternative sources of data provide valuable complementary metadata that is used in our study. We also extend the metadata available for each sample with targeted queries ran using binary or network inspection as described in §3.3.

A.4 Additional Measurements

In this Section we provide additional measurements obtained from our analysis:

- Table 10 shows an extension of the URLS hosting the malware observed during our analysis.
- Table 11 shows the top 10 wallets sorted by how much they gained. Note that these numbers include donation wallets, which we have filtered for our profit analysis. Nevertheless, we include them in Table 11 to show the similarity between the findings reported in [7].
- Table 12 shows the number of emails detected for each associated domain. As it can be observed, most of the emails are used to mine in *minergate*. This is an opaque pool which allows mining in various crypto-currencies.

A.5 Payments in the Freebuf Campaign

In this section, we visualize the payments done as a mining reward for wallets belonging to the Freebuf campaign, which is the longest campaign and the one with the highest earnings. Concretely, Figure 7 shows the payments made to all the wallets related to the campaign across time. A more detailed analysis is shown in Figure 8, focusing on the payments done in 2018 to the two wallets that were banned in *minexmr*. It can be observed that, after being banned, campaign operators decided to return to the *ppxmr* pool. However, while the campaign is still active, both the intervention (i.e., wallets being banned) and the change in the PoW of October 2018 has considerably reduced the payments made to this campaign, nearly turning it off.

Domains	#Samples	#URLs
github.com	163	388
*.amazonaws.com	85	396
www.weebly.com	80	96
*.google.com	38	74
hrtests.ru	37	1
cdn.discordapp.com	34	55
a.cuntflaps.me	32	48
file-5.ru	30	52
telekomtv-internet.ro	30	30
mondoconnx.com	26	26
free-run.tk	25	18
brafisaplay1.name	25	23
b.reich.io	23	23
mysuperproga.com	22	21
goo.gl	22	32
tyme.one	21	21
gatsoed9.beget.tech	20	18
mysupflax.name	19	18
bluefile.biz	19	18
pack.1e5.com	18	16
directxex.com	18	18
dropbox.com	17	50
*.4sync.com	16	142
store4.up-00.com	16	16
www.murphysisters.org	16	12
fireass.ru	16	16
weebly.com	15	17
4.program-iq.com	14	15
xmr.enjoytopic.tk	14	11
jkhskdjhsakdjas.info	14	7
a.pomf.cat	14	16
giantsto.com	14	12
daniltinkov228.website	13	8
root.mcs-katwijk.nl	13	8
plalium.pw	13	12
mm.cnxc.tk	13	14
debittech.ro	12	5
365experts.com.au	12	15
www.teamlunyr.com	12	12
murphysisters.org	12	9
store6.up-00.com	12	13
folderfiles10.ru	11	20
dl.x420.me	11	6
play.best01011.com	11	2
garant-ural.ru	11	12
callfor.info	11	17
v91049e6.beget.tech	11	12

Table 10: Extended list of domains hosting known mining malware, number of samples hosted under each domain and number of URLs hosting those samples.

Wallet	XMR mined	USD
496ePyKvPB...	82,985	10,655,849
49s5yfpFvE...	74,643	8,964,789
44N9sqiizw...	55,025	7,940,287
454HDLDtqC...	42,024	322,267
42yJMfdGHQ...	32,886	52,830
42NCdZTvv3...	26,273	42,295
44cwDVn9cQ...	23,300	2,288,329
46GGhVFZq8...	22,520	4,775,043
42ychz53ap...	21,389	42,351
46hoCjuFZB...	20,694	37,975
Total for 2,433 wallets	733,586.75	56,605,132.78

Table 11: Amount of Monero mined and corresponding USD for the top 10 wallets.

Pool	**#emails**
minergate	4980
50btc	41
crypto-pool	4
supportxmr	4
nanopool	4
btcdig	3
slushpool	2
moneropool	2
minemonero	2
moneromіners	1
monerohash	1
dwarfpool	1
suprnova	1
minexmr	1
f2pool	1
OTHERS	105
TOTAL	**5153**

Table 12: Number of emails in pools

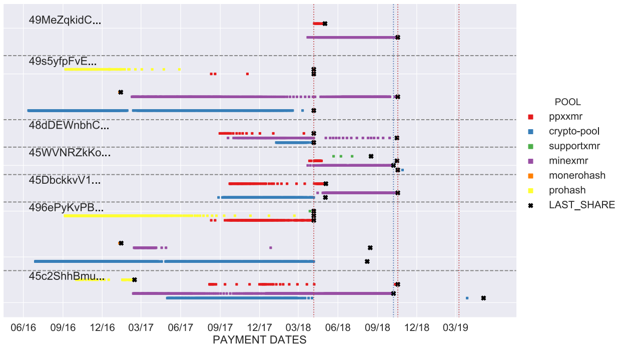


Figure 7: Payments per wallet for the Freebuf campaign. Dashed red lines correspond to the two changes in Monero PoW algorithm.

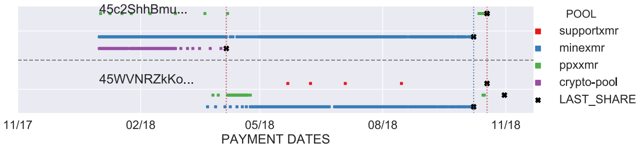


Figure 8: Detailed view of the wallets banned in the Freebuf campaign. Blue vertical line shows when the wallets were banned.