Digital Investigation 31 (2019) 100884

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

How to detect cryptocurrency miners? By traffic forensics!

Vladimír Veselý*, Martin Žádník

Brno University of Technology, Božetěchova 2, Brno, 612 66, Czech Republic

ARTICLE INFO

Article history:
Received 25 August 2018
Received in revised form
31 July 2019
Accepted 17 August 2019
Available online 22 August 2019

Keywords:
Bitcoin
Cryptocurrency
Mining pool
Mining server
Stratum protocol
GetBlockTemplate protocol
GetWork protocol

ABSTRACT

Cryptocurrencies set a new trend for a financial interaction between people. In order to successfully meet this use-case, cryptocurrencies combine various advanced information technologies (e.g., blockchain as a replicated database, asymmetrical ciphers and hashes guaranteeing integrity properties, peer-to-peer networking providing fault-tolerant service). Mining process not only introduces new cryptocurrency units, but it has become a business how to generate revenue in real life. This paper aims at different approaches how to detect cryptocurrency mining within corporate networks (where it should not be present). Mining activity is often a sign of malware presence or unauthorized exploitation of company resources. The article provides an in-depth overview of pooled mining process including deployment and operational details. Two detection methods and their implementations are available for network administrators, law enforcement agents and the general public interested in cryptocurrency mining forensics.

Introduction

The motivation behind cryptocurrency is to introduce an alternative currency that is not controlled by a government (e.g., the central bank). Trustworthiness of such electronic cryptocurrency lies in the utilization of cryptographical algorithms to verify transactions and fair emission of new units into circulation. Dark web marketplaces utilize cryptocurrencies for their: a) nearly instant and free-of-charge payments; b) easily obtainable and changeable addresses; c) hard to trace transactions (thanks to their peer-to-peer nature). Several studies (Raeesi, 2015) (Grinberg, 2012), (Johnson, 2014) investigate Bitcoin as the key component of any digital black marketplace because cryptocurrencies generally allow criminals to circumvent law enforcement agencies (LEAs) and regulators.

Of all cryptocurrencies, Bitcoin (Nakamoto, 2008), (Bitcoin.org, 2018) had become popular when it gained momentum at the end of 2013 after its exchange price skyrocketed. The current (at the July 2019) total number of Bitcoins (approx. 17.8 million) accounts for more than 202 billion USD (CoinMarketCap.com, 2017). Bitcoin is a peer-to-peer network with the distributed infrastructure of users and miners. A miner verifies ongoing transactions for a reward (either transaction fee or newly emitted Bitcoins). The reward is paid to the first miner who proves transaction by spending its computation power on this process. Other proof-of-work[1] cryptocurrencies also adopted the same mining concept. Anyone can join the solo mining process but the probability of earning a reward is low and the risk of wasted computational power without any profit too high. Therefore, miners form so-called mining pools. When the pool earns a reward, it is distributed by the pool operator among miners according to their contribution.

Apart from alternatives to Bitcoin (e.g., Litecoin, Ethereum, generally referred as *altcoins*), the cryptocurrency universe also contains *tokens*. Tokens (comparing to coins) represent digital asset or utility that leverages another's coin blockchain for being accounted. New tokens are generally not mined but distributed by their authors/owners. In the frame of this paper, we will focus only on the mining process behind coins and refer to them as “cryptocurrencies” interchangeably.

Any organization should be aware of running mining software on its hardware in its network due to at least two reasons: a) the mining activity is often caused by malware, therefore, the mining

* Corresponding author.
E-mail addresses: veselyv@fit.vutbr.cz (V. Veselý), izadnik@fit.vutbr.cz (M. Žádník).
URL: https://www.fit.vutbr.cz/~veselyv, https://www.fit.vutbr.cz/~izadnik

[1] In case of *proof-of-work* mining, the probability of finding a new block is directly proportional to a computational power invested in mining. While for *proof-of-stake* mining, the probability is directly proportional to a number of units owned by a miner.

https://doi.org/10.1016/j.diin.2019.08.002
1742-2876/

activity is an indicator of a compromise; b) the energy (e.g., electricity, cooling, CPU and GPU power) spent on mining is paid by the hosting organization, but the recipient of the reward is a malicious actor. Ali et al. (2015a) informs about various types of cryptocurrency malware dedicated to undercover mining on devices, desktops, and servers but also platforms like webcams, smartphones or network attached storages. Universities (Hern, 2014) or technological centers (Nield, 2018), (BBC.com, 2018) are typical examples of energy exploitation because they offer free computational resources (i.e., servers, network) to academics, researchers and students. Nevertheless, it is possible to start a mining operation in any organization (e.g., subsidized accommodation for Czech members of parliament, see (Frouzová and Zelenka, 2018)).

The malicious actor might exploit these assets resulting in an increased energy bill, depleted resources, endangered work processes, services and other users. For instance, Bitcoin mining has a severe impact on electricity comparable to the energy consumption of Ireland (O'Dwyer and Malone, 2014) in 2014. Another report (de Vries, 2018) provides a more in-depth analysis of how to estimate Bitcoin's hunger for energy concluding that it may reach 7.67 GW (comparable with Austria) during 2018.

In this paper, we focus on the detection of devices participating in the mining pools. Cryptocurrency mining is the only option how users may obtain freshly minted currency units. Moreover, mining is still the prevailing form of how to earn cryptorcurrencies with the existing equipment.

We propose two approaches for cryptocurrency miners detection in the network:

- The first approach employs a mix of passive and active traffic monitoring. The passive monitoring is based on the analysis of IP flow records, while the active monitoring is based on probing. The detection method as a whole slowly learns a list of mining servers which subsequently reduces the need for the active monitoring. Since anyone can set up own mining pool or even mining server, the resulting list of publicly known mining servers cannot be considered complete. However, it may be employed as a baseline for miner detection by any network operator.
- The second approach can be described as a catalog of mining pools. We have created a publicly available web application that stores metadata about existing mining pools. Any user may query our system to check whether a given FQDN,[2] IP address or port number is a part of known pool configuration.

Fig. 1 illustrates a stake-holder (i.e., administrator or LEA operative as *network analyst*) and modus operandi of above-mentioned approaches (i.e., *NetFlow probe* capable of cryptocurrency miners detection + the *pool catalog* validating existing mining servers and optionally feeding probe).

The contribution of this article involves: a) an overview of the current cryptocurrency mining technology; b) two detection methods to detect network traffic related with cryptocurrency mining; c) open-access data samples; and d) publicly available service cataloging mining servers.

The rest of the paper is organized as follows. Section 2 informs about related work on cryptocurrency mining. Section 3 brings details about currently used mining architecture and involved protocols. Section 4 describes passive/active traffic monitoring (the first approach how to detect miners), which also includes its validation and verification. Section 5 explains the implementation and operation of the mining server catalog (the second approach). The article is summarized in Section 6, which also outlines our future work.

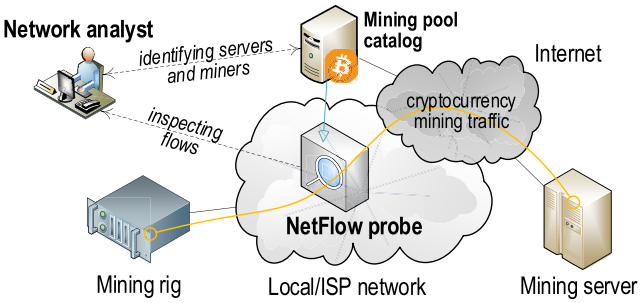


Fig. 1. Illustration of paper outcomes and actors.

Related work

This section summarizes knowledge from the selected articles relevant to cryptocurrency mining. We try to motivate miners detection in a frame of known cryptocurrency issues and research of others.

We consider Courtois et al. (Courtois et al., 1310) work as a great introductory source explaining Bitcoin mining. Despite focusing on Bitcoin mining process improvement, authors provide theoretical background explaining bindings between employed cryptography and cryptocurrency mining. Moreover, this work and other ones mentioned in this section allow us to skip ththughroughcryptographic description of the mining process. Instead of it, we will focus only on protocols and messages exchanged between miner and pool.

Kroll et al. (2013) and Lewenberg et al. (2015) provide an economical point of view on Bitcoin mining. They try to model the mining process as the game-theory problem. Eyal and Sirer (2014) discuss Bitcoin security and mining incentive-compatibility. All of these articles introduce interesting attacks that might disrupt any cryptocurrency mining process. We will briefly mention mining protocol "flaws" that may be used to identify miner and connect its identity with a real person.

Several studies (Juels et al., 2016; Hampton and Baig, 2015; Ali et al., 2015b; Kharraz et al., 2015) mention ways and means how cryptocurrencies are being employed in monetizing and as a platform for unlawful activities. Examples include ransomware attacks, botnet command and control operations, private key thefts, spam advertisements, pay-per-click or pay-per-install scams and others. Our research complements these studies by targeting the illicit mining of cryptocurrencies.

Huang et al. (2014) provide a comprehensive study of cryptocurrency mining malware. Authors developed methods, which correlate the mining bot with its mining pool. Moreover, authors were able to estimate the number of infected devices, generated revenue and duration of botnet infection. We consider this paper as a great encouragement for our work because it shows how successful discovery of miners can be crucial not only for proper network operation but also for significant reduction of botnet contagions. There is a connection between (unintentional) cryptocurrency mining and exploitation of resources.

D'Herdt, (2015) analyzed captured traffic samples and suggested to look for well-known ports and IP addresses of mining servers. Besides that, he derived that the communication of miners with mining server is sparse but often cyclic between 30 and 100s. Although it is possible to capture all the network traffic even on a

[2] Fully qualified domain name (FQDN) is complete host identification within a Domain Name System (DNS) tree hierarchy.

high-speed link Pus et al. (2014) so that the raw network data can be analyzed, it is a resource-expensive way of network monitoring from a long-term perspective. Therefore, various meta-data collecting approaches are utilized (i.e., several generations of NetFlow protocols (Claise, 2004) and IPFIX Claise and Trammell, 2013). The research in flow analysis has come up with simplistic as well as complex approaches ranging from statistical methods to machine learning (ML) approaches.

From the perspective of methodology, the closest to our work are methods based on supervised machine learning such as (Livadas et al., 2006). The authors of (Livadas et al., 2006) select descriptive features that should be extracted from the flow data, prepare an annotated data-set and train a classification model which is used to recognize specific events in the flow data. In comparison to (Livadas et al., 2006) we introduce an additional mechanism to reduce the large number of false positives generated by the classifier. To the best of our knowledge, there has been no work specifically focusing on detection of cryptocurrency miners utilizing flow data.

Mining background

This section provides a theoretical background (mostly based on Bitcoin use-case). However, explanation of the whole mining process for all cryptocurrencies is far beyond the scope of this article. Hence, only parts relevant to the miner detection are described. The first subsection lays out the basic theory for any cryptocurrency operation. The second subsection familiarizes the reader with the state-of-the-art of cryptocurrency mining software and hardware. The third subsection provides a deeper description of existing mining protocols.

Theory

Transaction encapsulates transfer of cryptocurrency units between parties, where a single transaction may contain multiple inputs and also outputs. To prevent fake or malicious transactions (e.g., double spending problem), a given user needs to validate the transaction history. Hence, transactions are chained together, where outputs of the previous transaction serve as inputs of the next transaction. Transactions are grouped into *blocks*, which vouches for the validity of contained transactions with timestamps and cryptographic hashes. Blocks are periodically recorded into a public ledger dubbed as *blockchain*. Blocks are bound together in the blockchain as a unidirectional list, where each item (i.e., block) has the pointer to its predecessor. The inception of cryptocurrency is done by starting its history with the first *genesis block*. Blocks are formed and their content verified by *miners*, who compete between themselves in the process (so-called *mining*) of appending new blocks to the blockchain. The winning miner earns reward in the form of newly minted coins (called *coinbase transaction*) as an incentive to participate on cryptocurrency peer-to-peer network operation. The winner is the miner, which would successfully solve the certain cryptographic task (e.g., compute a hash with certain properties from given inputs and nonces) of variable *difficulty* (which acts as a feedback mechanism guaranteeing deterministic time of block creation). Miners are grouped in *pools* in order to increase their chance of successful mining and thus to cash the reward.

Hardware and software

A wide range of different mining hardware/software exist that is mostly differentiated by the hashing algorithm employed and *hashrate* (i.e., computational performance in a number of hashes per second, abbreviated as hash/s). Depending on a given cryptocurrency, the user chooses the appropriate combination of hardware and software that impacts mining operation. Mining hardware capabilities pose an upper-bound limit for a maximum available hashrate. Nevertheless, the choice of mining software may optimize and automatize the mining operation. To generalize it, successful establishment of cryptocurrency mining consists of several steps.

1 Select cryptocurrency - There is no business perspective to mine cryptocurrencies if overall expenses exceed potential income. Hence, it is necessary to take into account: a) trend of exchange price; b) cryptocurrency viability; c) possible increase of mining difficulty; and d) ever-changing total hashrate of the peer-to-peer network.
2 Choose a pool - Participation in the pool (compared to solo mining) offers a more predictable generation of revenue, which is proportional to work done by a miner. It is important to choose a stable pool (in terms of Internet connectivity and denial-of-service protection) with a trustworthy pool operator (who will not embezzle earnings or submitted shares for his/her own profit).
3 Assemble mining rig - Overall power consumption of the mining rig goes hand to hand with its hashrate performance. Any mining rig generally dissipates a lot of heat and its ventilation/cooling produces significant noise. These facts can be used as indicators how to locate mining rig in corporate environment physically.
4 Configure mining software - Mining is controlled and managed either by official cryptocurrency client or dedicated software. Mining software needs a low-bandwidth, but constant connection to the Internet since it periodically exchanges work packages with the server.

Fig. 2 outlines usual deployment scenarios between miners and mining pools. A user may control multiple mining devices (rigs, sometimes also referred to as workers). Each mining device may rotate mining operation (i.e., in a round-robin or fall-back fashion) between multiple pools. Each mining device is connected to a single mining server that belongs to the pool, thou switching to a secondary mining server is quite common in case of the outage of the primary one. The connection between miner and server can be: a) direct without any middle-box (although, it reveals IP address of miner); b) proxied by centralizing communication with mining server via middle-box that may relay or even alter mining protocol data; c) overlayed via VPN,[3] TOR,[4] I2P[5] or any similar service.

Protocols

A mining pool and its members are using dedicated protocols to coordinate distribution of mining process. There are three general mining protocols supported by a majority of cryptocurrencies:

- *GetWork* was the first mining protocol ever. Comparing to its descendants, GetWork is a simple request-response scheme protocol - server assigns work package and miner blindly conducts mining task. Due to its simplicity, GetWork allows double-spent transactions in the case of corrupt pool operator. GetWork

[3] Virtual Private Network. For more, see https://en.wikipedia.org/wiki/Virtual_private_network.
[4] The Onion Router. For more, visit https://www.torproject.org/.
[5] Invisible Internet Project. For more, check https://geti2p.net/en/.

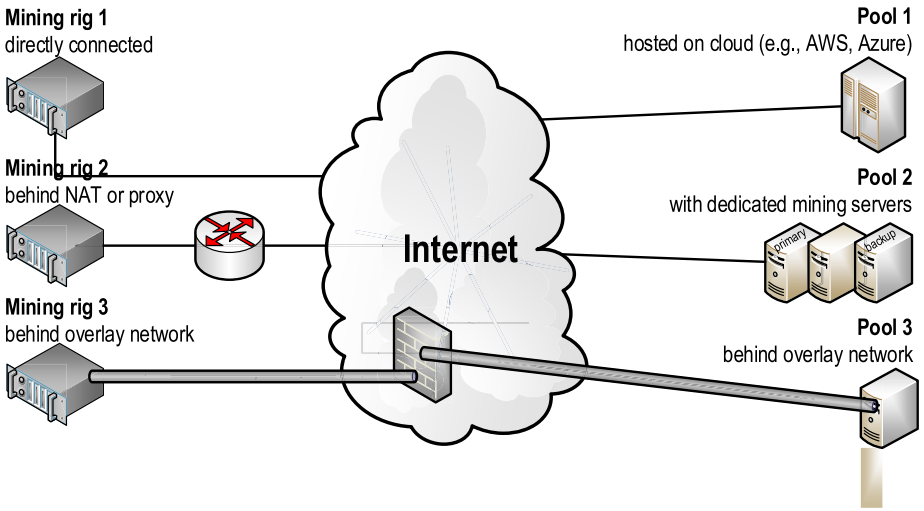


Fig. 2. Deployment scenarios for miners and pools.

messages with JSON[6] syntax are carried inside HTTP.[7] GetWork supports a limited number of protocol extensions using additional HTTP header lines.

- *GetBlockTemplate* is standardized mining protocol developed by Bitcoin community but also adopted by other cryptocurrencies. GetBlockTemplate was codified in BIP[8] 22 (Dashjr, 2012a). GetBlockTemplate is more decentralized by offloading block creation process onto miners instead of pools. GetBlockTemplate increases potential work package size and reduces mining protocol overhead to support performance delivered by ASIC miners. Moreover, BIP 23 (Dashjr, 2012b) standardizes ways enhancing GetBlockTemplate without any major protocol redesign or non-conformant HTTP header hacks.
- *Stratum* protocol (Palatinus, 2018a) was prototyped by M. Palatinus, inventor of pooled mining and operator of the oldest Slush pool (Palatinus, 2018b). Stratum development was motivated by a need to remedy design flaws of previous two protocols: a) by removing HTTP as a carrier, which reduces unnecessary protocol overhead; b) by removing long polling feature that posed scalability issue for load balancing of traffic between miner and server; and c) by adding the *extranonce* field that allows miner to generate more hashes locally without bothering a mining server for a new batch of work. Stratum is a JSON-RPC 2.0 (JSON-RPC Working Group, 2013) compatible protocol that operates directly above TCP.

All of these protocols leverage TCP[9] as the transport layer protocol. Comparing to official cryptocurrency peer-to-peer clients, mining protocols do not use any “well-known” port number. It depends solely on the preference of mining pool administrator on which ports pool servers accept connections. Hence, port numbers 80, 443, and 25 are often used as a best practice to bypass firewalls between the mining device and its mining server.

The usual message exchange involves several steps. With the initial message, the miner connects to the mining server and provides authentication credentials. Authentication is necessary because based on credentials, mining pool correlates submitted shares with miner's account and credit earnings. Two types of authentication are common:

- *registration-oriented* - Before establishing the mining operation, the user owning mining rig needs to sign up to the pool and create an account. A part of account administration involves workers (i.e., separate mining devices) setup. Authentication credentials inside mining protocol include username and password.
- *registration-less* - Some pools tender their services without any dedicated account registration. In that case, the miner usually provides just cryptocurrency address to inform pool where to send payments. This identifier substitutes username and is enough for authentication.

Regardless of authentication type, the username may contain optional suffixes such as worker identifier (in order to distinguish different workers of the same user) or e-mail address (where the user is notified about any problems occurred during mining).

The next step in mining protocol communication is a recurrent assignment of work packages provided by the server. Each work package contains *data*, *target*, and *nonce* (other fields depend on cryptocurrency). A miner tries to find a hash (from combined data and nonce) that meets difficulty. Different cryptocurrencies are using distinct hashing algorithms - e.g., SHA-256d for Bitcoin, Scrypt for Litecoin, X11 for Dash. Miner either submits correct solution or restarts mining with different inputs upon receiving a new work package. Miner periodically announces its state to the server.

Fig. 3 illustrates Stratum exchange from mining device to its server (with the red color) and vice versa (the blue color). We can observe the typical confluence of messages. A connection to the pool is initiated with the first message (marked as #1), where we can see authentication details. The server confirms it with message denoted as #2. The server sends a work package (#3) that needs to be computed. Upon proper initialization of mining software, the miner asks for a new work package (#4), which the mining server gladly provides (#5). The miner successfully finds the hash and submits (#6) the complete solution back to the server. The server decides whether the miner's result is valid or not (in the case of #7, it is valid) and sends a new work package (#8). The miner starts a new task and meantime periodically updates server about its local

[6] JavaScript Object Notation. For more, see https://tools.ietf.org/html/rfc7159.
[7] Hypertext Transfer Protocol. For more, visit https://tools.ietf.org/html/rfc7230.
[8] Bitcoin Improvement Proposal. See https://github.com/bitcoin/bips/blob/master/bip-0002.mediawiki.
[9] Transmission Control Protocol. For more, see https://tools.ietf.org/html/rfc793.

Fig. 3. Example of Stratum protocol message exchange.

computational speed (message marked as #9) so that server can dynamically adjust the size of subsequent work packages. If we focus on the forensic analysis of metadata related to mining protocol, then we can extract metadata described in Table 1.

Traffic monitoring

Network traffic monitoring provides data for network management, accounting as well as security. In our work, we assume basic network monitoring based on flows. The flow is a set of packets sharing the same key (in most cases, source and destination IP address, source and destination port, protocol). The flows are measured at the observation points and the measured data per each flow are exported to the collector by a flow export protocol (e.g., NetFlow v5). For further details on flow monitoring please refer to (Hofstede et al., 2014).

We assume that the flow monitoring captures the communication of the active mining clients as they connect to the pool, ask for a job and deliver the results. We propose a concept, depicted in Fig. 4, that is capable of identifying mining IP addresses based on the flow data. The incoming flow data are collected and the features are extracted into a feature vector. Subsequently, a passive detector decides if the feature vector looks like miners communication or not. At this stage, false positives usually occur due to an aggregation of information and due to the heuristic nature of the detection algorithm. We address the problem of the false positives by adding the second detection step. During the second detection step an active probe verifies whether a server, the suspicious client connects to, belongs to a mining pool or not. In order to reduce the number of probes, we propose to employ a list of known and probed servers which is fed by the results of the active probe as well as from the catalog of mining pools (described in Section 5).

Design of cryptocurrency network traffic detector

Passive detection is a key component in the schema. Its effectiveness determines the number of false positives that must be verified by the active probe component. Therefore we propose to utilize a supervised machine learning technique to assemble an optimized classifier capable of reaching a low number of false positives as well as false negatives. The supervised machine learning assembles a classifier based on positive and negative examples, in our case, feature vectors belonging to mining and non-mining communication. The machine learning classifier (ML-classifier) is based on a decision tree induction with a particular implementation of J48 in Weka (Witten et al., 2011). The decision tree induction recursively selects features and their thresholds to maximize information gain contributed by the selected feature. As a result, the decision tree contains in its root the feature with the most information value while towards the leaves the information

Table 1
Metadata available in mining protocols.

Metadata	Description
IP address, port number	By inspecting IP addresses, we can geolocate both miner and mining server. Together with port numbers, we can account network traffic with NetFlow. Once we have NetFlow records available, we can answer questions such as for how long is mining operation active, how many mining devices are involved, etc.
Pool information	GetWork or GetBlockTemplate protocol extensions may uncover other useful intel such as alternative mining servers including their IP addresses, fully-qualified domain names, and port numbers.
Miner's username	Based on authentication type, username field may contain either nickname or account name of pool user or its cryptocurrency address. This information may be crucial for successful correlation of real-world person and its electronic identity.
Miner's password	Authentication message of any mining protocol includes a password. However, it is seldom used for authorization or any purpose by a pool. The default value of password field for the most of mining software is 'x'.
Miner's email	Some pools offer email notifications about the progress of mining operation. In case of any problem such as the miner outage, too many rejected shares or disconnection from the pool, the user is warned by email. The email address may be optionally part of mining protocol message filed, which may help to reveal user's identity.

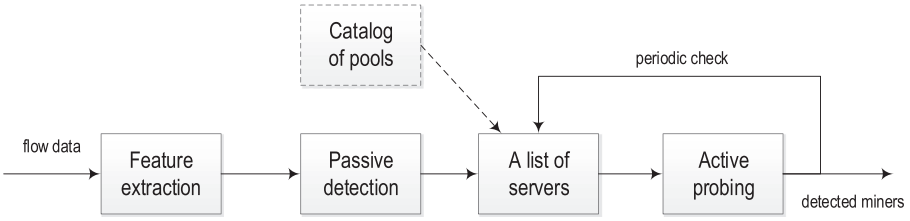


Fig. 4. Two-step detection schema (Catalog of pools is a supplementary component).

value decreases. This leads to the construction of an optimized decision tree.

Data annotation is vital to the supervised machine learning. Our process of data annotation is based on an iterative approach. Each iteration annotates new data utilizing knowledge extracted from the previous iteration. The annotation itself utilizes the two-step detection schema already depicted in Fig. 4. During the first iteration, the incoming data are annotated only by the list of known servers which utilizes information from the catalog of mining pools as well as information about our own mining clients. We select relevant features and manually design a classifier based on the analysis of the first data-set. During the second iteration, all components in the schema are active, i.e., features are extracted, the manual classifier is utilized by the passive detection component and active probing verifies its results. We train the ML-classifier on the new data-set annotated during the second iteration. In the third iteration, the manual classifier is replaced with the ML-classifier and a new data-set is annotated. This final data-set contains no false positives and only such false negatives that were not recognized by the ML-classifier nor by the list of servers.

Feature extraction. In order to derive and select the classification features, we analyzed several packet traces of various known mining clients (e.g., miners deployed by authors such as cpuminer) and the traffic belonging to the well-known mining servers listed in the catalog. According to our analysis, we can state that:

- Mutual communication between a miner and a mining server often lasts for several hours.
- Packets are generally small, often in the range from 40 to 120 bytes.
- Most flows are observed with TCP ACK and PUSH flags set.
- The destination port is either a well-known port of a different service or not well-known but definitely lower than the source port.
- Flows are generally long-lasting, often exported before its end due to an active timeout.
- Communication is not disrupted, i.e., most flows do not contain the RST flag.

Table 3
Volumes of utilized data-sets.

	Flows	Packets	Bytes	Interval
Train.-eval.	16M	117M	54G	2018/02/02 14:00 - 14:15
Real	3.6G	27.8G	99.5T	2018/02/09 14:00 - 20:00

Based on the analysis we have selected features (listed in Table 2 in the left column) to be collected per each triplet (source and destination IP addresses and destination port). The triplet and its features are collected in a hash table in a memory. In order to limit the size of the hash table, triplets are evicted from the table if there is no update for more than a defined number of seconds (we set up this inactive timeout to be 1 h). However, the passive detection itself is performed periodically, every 60 s.

The manually-designed classifier is based on a cumulative score which must overcome a threshold T. The score is gradually increased by an increment of $1/n$, where n is the number of satisfied conditions per each feature. The conditions are listed in Table 2 in the right column. The threshold T is experimentally set to reach approx. 90% of true positives.

The list of servers keeps track of the probed servers (positives and negatives) as well as of the servers reported by the Catalog. Not only the list reduces the number of active probes but it also reduces the number of false negatives (also in case of data annotation). If the passive detection fails to recognize the triplet as mining there is a chance that the list corrects the result due to its prior knowledge.

The active probe connects to a mining server pretending to be a regular miner asking for a job. If the server replies with an expected answer, then it is very likely that: a) the server is truly mining server and b) clients connecting to this given server on this particular port are actually miners. The probe itself differs based on the mining protocol, where each probe consists of several queries targeting different cryptocurrencies (namely Bitcoin, Monero, Ethereum, Zcash). The probing itself runs in parallel since it takes time for a server to respond or time out. A snippet of a query for Stratum protocol and a corresponding response is given in Fig. 5.

Table 2
Selected features and conditions of the manual classifier.

Feature	Condition
Average number of bytes per packet	In the range 35–80 bytes or 105–110 bytes
Average number of packets per flow	Out the range of 5–40
Average number of packets per minute	In the range 2–8 or 40 - 5300
Duration of a communication	Greater than 300 s
Percentage of flows with ACK & PUSH	Higher than 90%
Percentage of flows with RST	Less than 1%
Percentage of flows with SYN	Less than 5%
Percentage of flows with FIN	Less than 5%
Percentage of flows with source port greater than destination port	Higher than 90%

```
{"id": 15, "method": "mining.subscribe", "params": ["cgminer/3.7.2"]}
{"error": null, "result": [[["mining.notify", "f21800001"],
["mining.set_difficulty", "f21800002"]], "f218000000000000", 4], "id": 15}
{"method": "mining.set_difficulty", "id": null, "params": [1024]}
{"method": "mining.notify", "id": null, "params": ["1526710821_119016",
"60b9ac6add6f3048fac87d3bb8b926437db7e9da5d81120bf57d2b6247d904a8", <omitted>
```

Fig. 5. Example of Stratum request (red) and response (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Evaluation

Data-sets

The experiments were carried out on data collected in Czech National Research and Educational Network connecting more than 30 organizations (e.g., universities, labs, hospitals) including more than 400 thousand users altogether. However, only 3 subnets conforming to 3 large organizations (over 50 thousand users) were considered for training and evaluation in order to reduce the amount of training data and to allow for manual inspection of the results. Table 3 depicts the parameters of the reduced data-set in comparison to the original real data-set. The observation points are located on the peering links with internet exchange points or other national networks. Therefore, the communication between entities within the national network is not part of the data.

We created an offline data-set[10] consisting of the feature vectors and their classification (i.e., mining/non-mining client). The data-set is annotated utilizing the schema described in Section 4.

Experiments

The selected features were evaluated on the annotated data-set described in the previous section. Fig. 6 depicts cumulative normalized distribution function (CDF) of the selected features. Each feature is assigned two functions – one for samples annotated as mining (positive) and one for samples annotated as other (negative).

In Fig. 6a we can observe that the distribution of positives and negatives differs. Please note that the x-axis of this figure was shortened to display the detail (the maximum size is 1460 B). It can be seen that:

- approx. 40% of positives accounts for packets of average size from 36 to 80 bytes,
- more than 50% of positives accounts for packets of size 105–110 bytes,
- less than 10% of negatives consists of packets of size between 105 and 110 bytes and
- another 20% accounts for packets larger than 110 bytes.

Distribution of a number of packets per flow (in Fig. 6b) and packets per minute (in Fig. 6c) are correlated. Please note that due to a high difference between minimum (one packet per flow) and maximum (over 200 thousand packets), a logarithmic function is applied first to highlight differences in lower order of magnitudes. The CDF is constructed after applying the logarithmic function. Positive triplets belong to either a group with a low number of packets or another group with a high number of packets per flow as well as per minute. Moreover, in the case of negative triplets there exists less than 1% of instances with an extremely high number of packets.

Fig. 6d displays the distribution of the rate of flows with push and ack flag set to all flows. The CDF shows that:

- more than 20% of negatives contains no ack nor push flags,
- CDF of negatives slowly rises to 40% at the rate of 0.8,
- on the other hand, there are nearly zero positives with a rate lower than 0.8 while the majority of positives exhibits rate higher than 0.9.

The opposite holds for the distribution of other flags depicted in Fig. 6e, f, 6g. Positives with zero rate account for the majority of their samples while negatives only for 50% in case of SYN, for 80% in case of RST and 65% in case of FIN.

The detection works upon requests from miners to the mining server. To this end, the feature capturing the rate between the number of flows with source port greater than the destination port to all the flows aims at distinguishing between a prevalent request or response communication. By keeping the rate as one of the features and not an a priori condition we allow the detection algorithm to detect miners even in case of the triplets that aggregate responses if the other features recognize that triplet as potentially positive. This makes the algorithm more robust but for the price of more false positives and the higher number of triplets to work with. The distribution of positives as well as of negatives are almost identical.

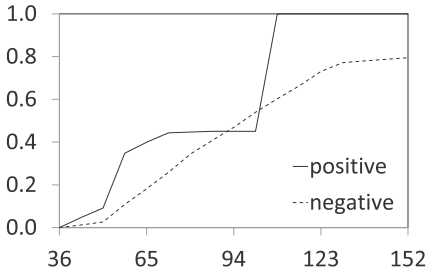
Obviously, none of the features considered is good enough to directly distinguish between positives and negatives. Therefore, the classification algorithm must combine the results of several features to improve detection results. We evaluate two classifiers, one designed manually and one based on machine learning, both described in Section 4.1.

As the training and evaluation data-set contains a significantly lower number of mining communications (273 positive triplets) than of other communications (356,574 negative triplets), we apply *ClassBalancer* filter to balance the weight of both sets in order not to overtrain the detector on non-mining communications. The training process is set up with the 5-fold cross-validation, and the number of instances in a leaf must be at least 100 (other parameters are kept default). The higher the number of instances in a leaf, the shallower tree is generated by the training process. The shallower tree, the faster is its evaluation as well as the tree is less overfitted to a particular data-set.

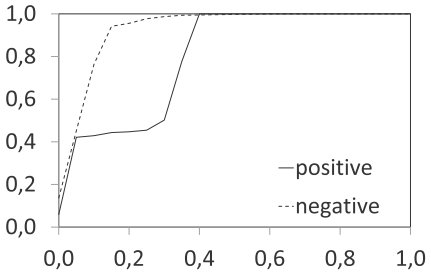
The confusion matrix of the resulting ML-classifier is depicted in Table 4a. The table shows that ML-classifier marks: a) mining communication as mining in most cases; b) another communication as other; c) except in 2.6% of the cases where communication is misclassified as mining (i.e., false positive); and d) although the ML-detector misclassifies 4.7% of the mining triplets, it is a significant improvement over the manually constructed classifier.

The confusion matrix of the manual classifier is depicted in Table 4b. The manual classifier fails to correctly recognize approximately 10% instances of each class. In case of the false positives this would lead to three times higher number of the probes in

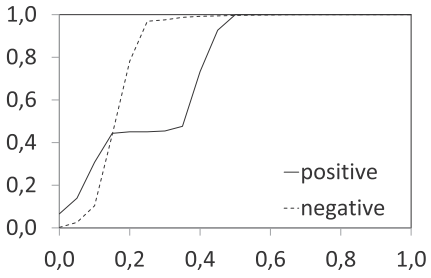
[10] The data-set can be downloaded from the results reproduction page (Vesely and Žádník, 2018). This data-set was created by a streamwise automated analysis framework (Cejka et al., 2016).



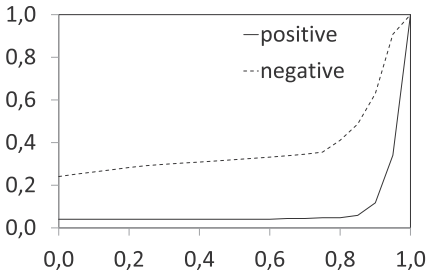
(a) bytes per packet



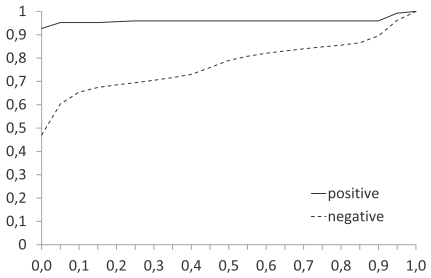
(b) $\log_{10}$ of packets per flow



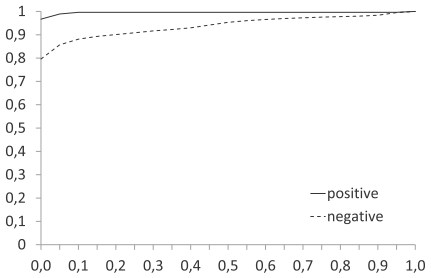
(c) $\log_{10}$ of packets per minute



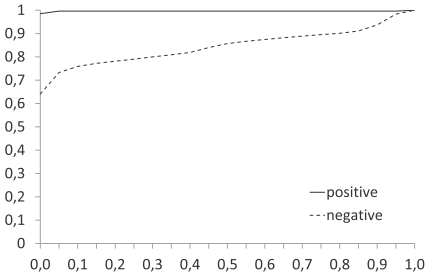
(d) ACK & PUSH / ALL



(e) SYN / ALL



(f) RST / ALL



(g) FIN / ALL

Fig. 6. Cumulative distribution functions of features.

Table 4
Confusion matrices.

(a) ML-classifier.

Class	mining	other
mining	95.3%	4.7%
other	2.6%	97.4%

(b) Manual classifier

Class	mining	other
mining	89.7%	10.3%
other	8.6%	91.4%

comparison to the ML-classifier. Evaluation of the whole two-step detection schema with ML-classifier yields 100% of true positives and true negatives (as the annotation is performed by the schema itself). This highlights the contribution of the list of servers to reduction of false negatives and the active probe to the reduction of the false positives in comparison with the pure passive detection approach.

Catalog of mining pools

We were also looking for a more lightweight solution suitable even for small corporate networks lacking capacities to install dedicated probes performing our active/passive traffic monitoring employing machine learning. We want to offer conclusive detection results with a minimum set of input information.

Network administrator and law enforcement agent (i.e., our main actors for mining detection use-case) shall have basic NetFlow records of investigated device/network segment. These records contain at least source/destination IP addresses, source/destination ports and a protocol identifier. The reasoning behind our second approach is following. If we know IP address of mining pool server, then we can reliably distinguish between mining and non-mining connections. Moreover, if we are aware of the port number employed by a pool operator, then we can tell what cryptocurrency is being mined through the connection.

Design

Both mining server's hostname and port number are publicly available (except mining malware cases) on pool's webpage because they are necessary for successful setup of the mining process. Without this vital information, the miner would not be able to configure mining software properly.

Based on these premises, we have decided to manually collect all mining software configurations announced by the biggest mining pools for several important cryptocurrencies. We gathered all these data in a database, which is accessible through a web application called sMaSheD (Mining Server Detector of cryptocurrency pools). In the rest of this subsection, we briefly outline some of the design choices that we have made during the sMaSheD development.

There are hundreds of coins (and tokens) available in cryptocurrency universe. In order to choose coins supported by sMaSheD, we did due diligence on "the most popular" cryptocurrencies taking into account public news (Kharif, 2018), dedicated reports (Carbon Black, 2018) and consultations with our LEA partners. Bitcoin is dominating this ladder due to its importance (e.g., around 80 million USD worth of Bitcoins stolen from a hacked cloud mining service provider in December 2017 (NiceHash.com, 2017)). However, Monero becomes more and more used by malwares because of its anti-forensic features, which help to cover criminal's tracks (e.g., nearly 5% of all Moneros in circulation worth of 175 million USD were mined using malware (Grunzweig, 2018)). The third is Ethereum thanks to smart-contracts and popularity among token developers (e.g., 30 million USD worth of Ethereum stolen by a wallet breach in July 2017 (Zhao, 2017)). Regardless of the current set of cryptocurrencies, sMaSheD is designed to be a generic catalog of mining pools which should be easy to maintain and operate.

Our tool operates according to the diagram outlined in Fig. 7. Pool information are stored into sMaSheD together with available FQDNs of pool servers. Server names are resolved onto a list of IPv4/IPv6 addresses, which are then verified as mining servers by employing mining protocol probes. Each operational step is described in more detail below.

We investigated mining distribution among available pools for each chosen cryptocurrency. The majority of pools add their signature into the freshly mined block. This marking allows to account the success rate of each participating pool. Moreover, pools are announcing their overall hashrate performance publicly. By combining these data, we receive quite a reliable overview about more and less important pools for every cryptocurrency. Anyone can obtain these data from dedicated web-pages, e.g. (Blockchain.info, 2018) for Bitcoins (Litecoinpool.org, 2018), for Litecoin, and (Etherscan.io, 2018) for Ethereum.

Mining software configurations are collected by web scraping the content of pool web pages. This procedure is currently performed manually by sMaSheD administrators. However, we aim at the automation of this process in the near future. The following set of information is being collected for every pool (from web-pages similar to Appendix A):

- the name of the pool and its home URL;
- the list of pool servers identified by FQDN including ports associated with a mined cryptocurrencies;
- every mining server FQDN is resolved onto a list of IPv4/IPv6 addresses.

Nevertheless, some pools are private (e.g., Bitfury pool with roughly 3% hashrate share (Btc.com, 2018) run by a company[11] with the same name producing ASIC mining solutions). The operator of such pool does not maintain any publicly available web page, which makes any web scraping of configuration impossible. Hence, sMaSheD catalog does not contain a complete list of pools for a given cryptocurrency. Fortunately, private pools constitute a fraction of overall network hashrate.

Mining server FQDNs may include information about location, mined cryptocurrency (e.g., eth-us2.dwarfpool.com) or employed algorithm (e.g., sha256. eu.nicehash.com). However, a single visit of a pool's web page does not take into account the changing nature of

[11] For more, visit https://bitfury.com/.

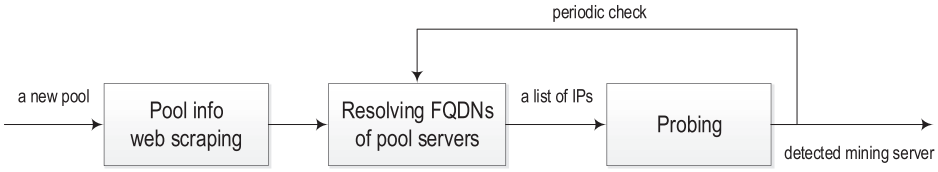


Fig. 7. Operation schema of sMaSheD tool.

pool infrastructure (i.e., mining service availability on new/old servers).[12]

Pool operators provide server FQDNs, which are resolved by miners onto various IP addresses based on miner's geolocation. Based on deployment (see Fig. 2), DNS may resolve a single FQDN onto many IP addresses (e.g., stratum. slushpool.com) in order to guarantee high-availability of a mining service. sMaSheD tries to keep the list of these IP addresses as up-to-date as possible. It is a necessity especially for pools leveraging cloud deployment because cloud providers often rotate available IP addresses among customers' virtual machines. An IP address of mining server today can belong to a completely different machine tomorrow. Because of this changing nature and since a single FQDN may actually represent a set of load-balancing mining servers, sMaSheD periodically renews the list of IP addresses associated with each mining server within the system.

In order to provide more reliable results if a given IP address belongs to a mining server or not, we developed probing similar to one described in Section 4. This probing repeats for all known pools (and their mining servers). During every periodic check of mining server, sMaSheD sends crafted mining protocol message and waits for the response. If counterparty reacts properly (with a message containing work package), then it confirms that this device is really a pool's mining server.

Probing is supported for Stratum and GetBlockTemplate mining protocols. sMaSheD is probing single server for both of these protocols. GetWork is also implemented, but we were not able to test it since this protocol is deprecated and not employed by any pool within our system. There are three probing return codes:

- DOWN - Probing failed because connection had not been even established. This occurs when a port on the server is closed, or some middle-box is blocking the connection.
- LISTEN - The connection was accepted on a specified port, but the alleged server returns an empty response. This happens when a) server is using different mining protocol than the tested one; b) port is opened but bound to a different application.
- UP - Probing succeeded because mining server responded with mining protocol message containing valid content. Message validity depends on employed mining protocol and consists of multiple value presence tests (e.g., *error*, *result* and other JSON fields). This validator can be easily extended to support changes or even new mining protocols.

Probing return code is usually accompanied with a verbose result (i.e., destination unreachable, unknown method, mining subscribe). sMaSheD records each probing attempt, which creates a history of service availability for a given mining server. These temporal data can later prove that IP address was used by a mining server (at least from the perspective of sMaSheD).

Evaluation

We need to be sure that our probing tool provides trustworthy results. In order to validate them, we compared the behavior of sMaSheD with official mining software. We decided to use cgminer 3.7.2 (Kolivas, 2018) because it is well-established and supports all available mining protocols.

We tested both tools over the same set of mining servers and recorded communication into PCAP file. We compared connection success rate (based on textual console outputs) and messages exchanged between miner (either sMaSheD or cgminer) and mining server. We did not find any differences for detected mining servers when comparing sMaSheD and cgminer connection attempts. Both applications used the same set (1983 entries) of IP addresses and ports of alleged mining servers.

sMaSheD system does not send any authentication credentials towards a pool upon the check, the basic response for mining subscription message is enough to mark a device as mining server positively. This is illustrated in Wireshark message captures depicted in Fig. 8 and Fig. 9.

sMaSheD is coded as a web application employing PHP framework Laravel 5.8 with front-end based on Bootstrap 3. The system is operated in a Docker container on CentOS 7. All source codes are available on (Vesely, 2018).

The database of sMaSheD currently (in August 2018) contains:

- 15 cryptocurrencies mined on 96 various port numbers;
- 58 pools operating 212 servers with 987 addresses;
- 3954 probing associations (dubbed as mining properties) and it takes approximately 60 min to them (i.e., check all IP-port tuples of all known mining servers for both Stratum and GetBlockTemplate).

Mining pools catalog sMaSheD (implemented as the second approach solution for miners detection):

- offers access to all data in JSON format through REST calls;
- periodically probes available pools' servers whether they provide mining services and accounts the result;
- allows privileged users to update database according to the current situation (i.e., add newly established pools).

Conclusion

In this paper, we provided an in-depth analysis of cryptocurrency mining operation. We designed and implemented passive-active flow monitoring and sMaSheD catalog to detect mining devices within the network. We tested the feasibility of these approaches on real-life data as well as published data-sets utilized in this article under open access policy. We conclude that

[12] In order to address potential changes in server names, we conducted experiments to obtain them automatically. We tried to generate hostnames as permutations from a set of keywords, which includes cryptocurrency abbreviations, country codes and pool domains. Unfortunately, this approach: 1) generated way too many false hostnames; and 2) verification of generated hostnames is a time-consuming process. DNS allows listing of all DNS records (including hostname A and AAAA records) through zone transfer, but this is not applicable for our use-case. For more about DNS records and zones, please read RFC 1035 and related ones.

```
{"id": 195, "method": "mining.subscribe", "params": ["cgminer/3.7.2", "0024b7cc"]}
{"result": [[["mining.notify", "0024b7cc"]], "8d337042", 4], "error": null, "id":
195}
{"id": 202, "method": "mining.authorize", "params": ["X", "Bitcoin"]}
{"params": ["00000000000074de",
"547eacc4e9476c71d05dc2e56745486d630191300017ea67000000000000000",
"01000000010000000000000000000000000000000000000000000000000000000000000000ffffffff37
03bbf7070004823df55a0483a8032808", <other output ommited>
```


Fig. 8. Message exchange between cgminer (red) and pool (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)


```
{"jsonrpc": "2.0", "method": "mining.subscribe", "params": ["Miner 1.0"], "id": 1}
{"result": [[["mining.notify", "0024cda8"]], "a5d4f1dc", 4], "id": 1, "error": null}
{"params": ["000000000000751a",
"cdd11ef3fe81084ce97262c42cff4bcb5d8a744700298abf000000000000000",
"01000000010000000000000000000000000000000000000000000000000000000000000000ffffffff37
03c0f70700045948f55a04a58a571508", <other output ommited>
```


Fig. 9. Message exchange between sMaSheD (red) and pool (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

catalog and passive-active approach are complementary - catalog is more focused on maintaining current information about mining servers anywhere on the Internet, while passive-active flow monitoring helps to reveal miners within enterprise networks. Data from sMaSheD can be used for refining detection capabilities of flow monitor.

The results of passive-active detection approach show that although there is a high number of false positives after the passive detection, it is sufficiently low to enable active verification of the results. In comparison to the pure catalog approach, passive-active detection is capable of discovering emerging or deliberately hidden pools. As such it should serve Security Operation Centers, CSIRT, and network security service providers to populate their cyber threat intelligence systems.

The goal of our sMaSheD system is to become a tool as valuable for network administrators and LEA operatives as what is ExoneraTor (The Tor Project et al., 2018) application for TOR overlay network. The sMaSheD prototype including a large data-set is available at (Vesely, 2019a) (see Appendix B for demo screenshots). Moreover, anyone can deploy own installation and feed it with custom pools. Our sMaSheD system can be deployed either from sources (Vesely, 2018) or as a containerized set of Docker images (Vesely, 2019b). Online catalog offers a curated list of the most popular pools and their servers. Data available in sMaSheD offer a neat solution for following use-cases:

- create an access control list that will block unwanted mining traffic (based on IPs and ports known to sMaSheD);
- detect the presence of miners via inspection of their DNS queries (based on FQDNs and IPs contained in DNS requests and answers);
- data-retention proof about mining service availability for a given IP + port tuple.

In the future, we would like to automatize metadata collection for sMaSheD by a periodic scraping of relevant web pages. Currently, the information provided by our catalog is updated manually, which makes the system less dynamic than we would appreciate. Regarding additional future work, we also consider other strategies on how to probe and positively identify pool servers based on different mining protocol messages. Last but not least, we are constantly adding new cryptocurrencies, pools, and servers as they appear in publicly disclosed announcements of illicit mining activities.

Acknowledgement

This article has been supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602). Authors also want to acknowledge work done by Jakub Kelečeni, Erik Šabík, and Martin Cagaš, the students of Brno University of Technology.

Appendix C. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.diin.2019.08.002.

Appendix A. Mining software configuration

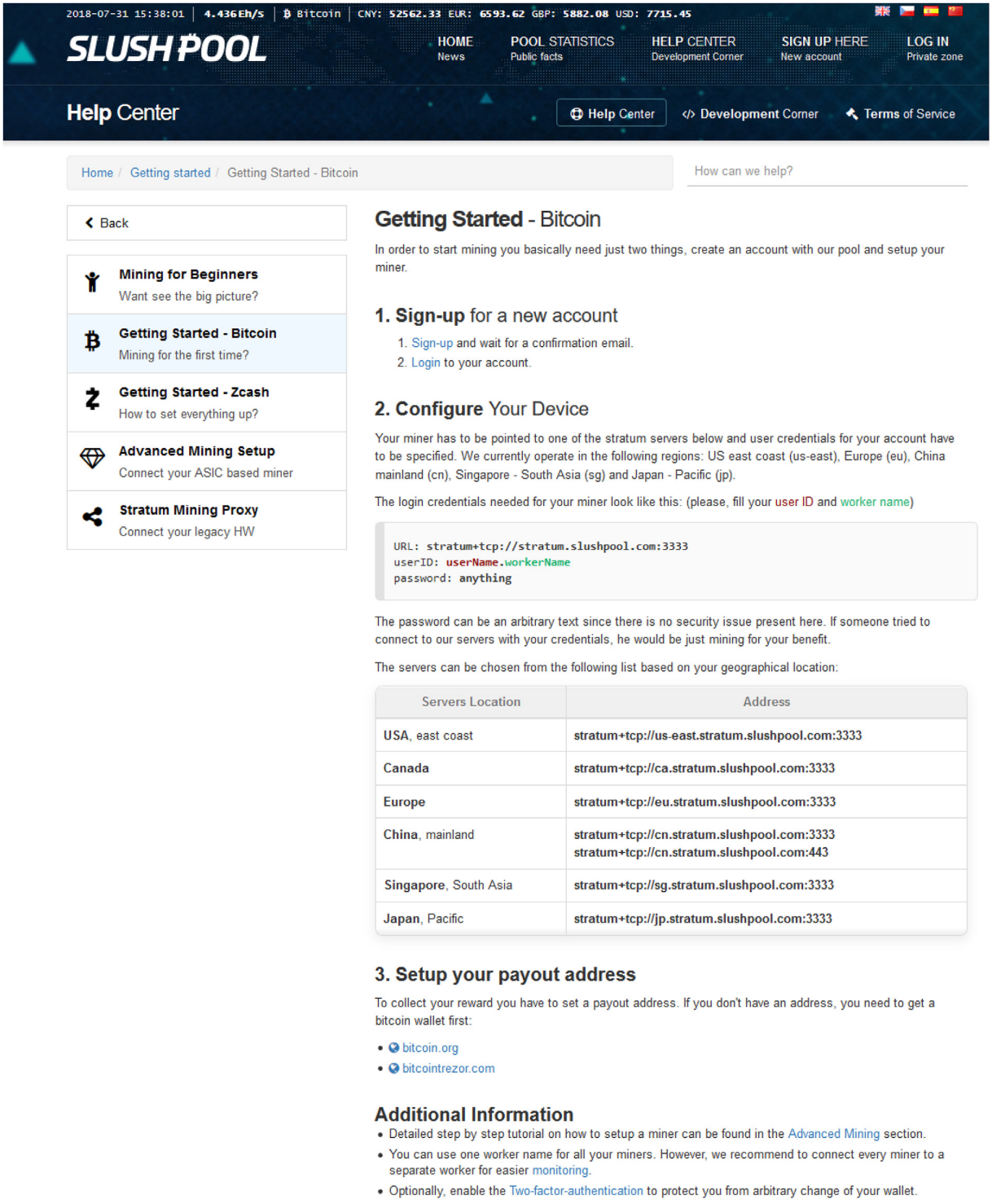


Fig. A.10. Example of mining software setup taken from SlushPool

Appendix B. sMaSheD Demo

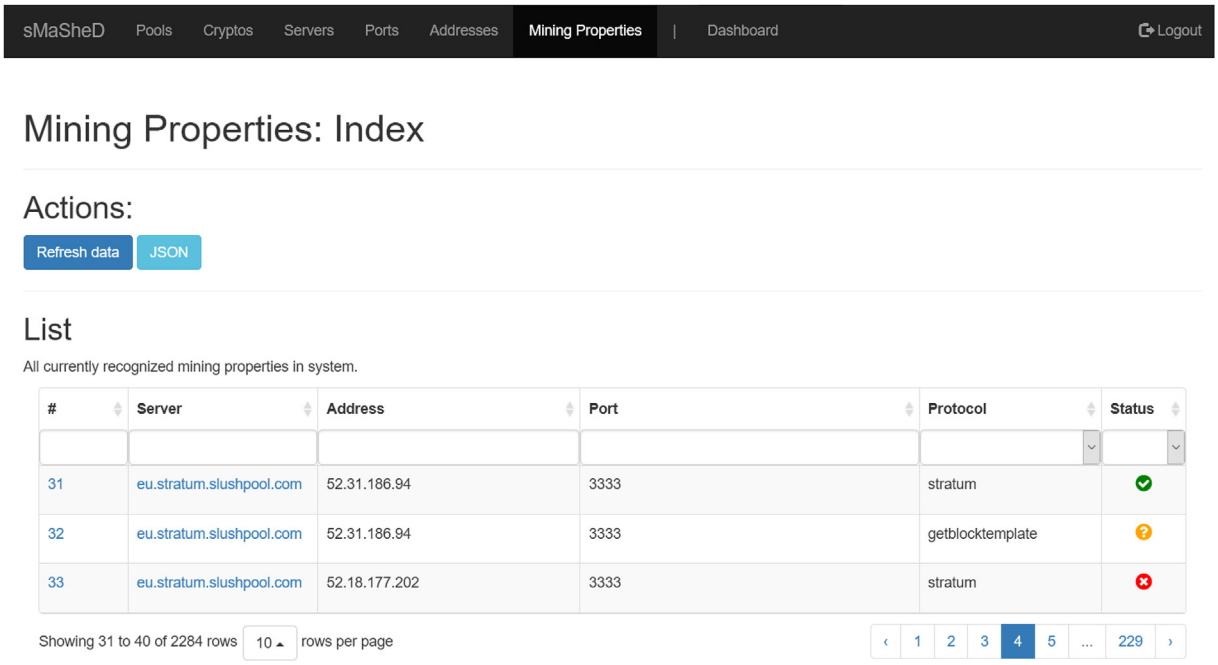


Fig. B.11. Probe results of selected subset from all available mining servers

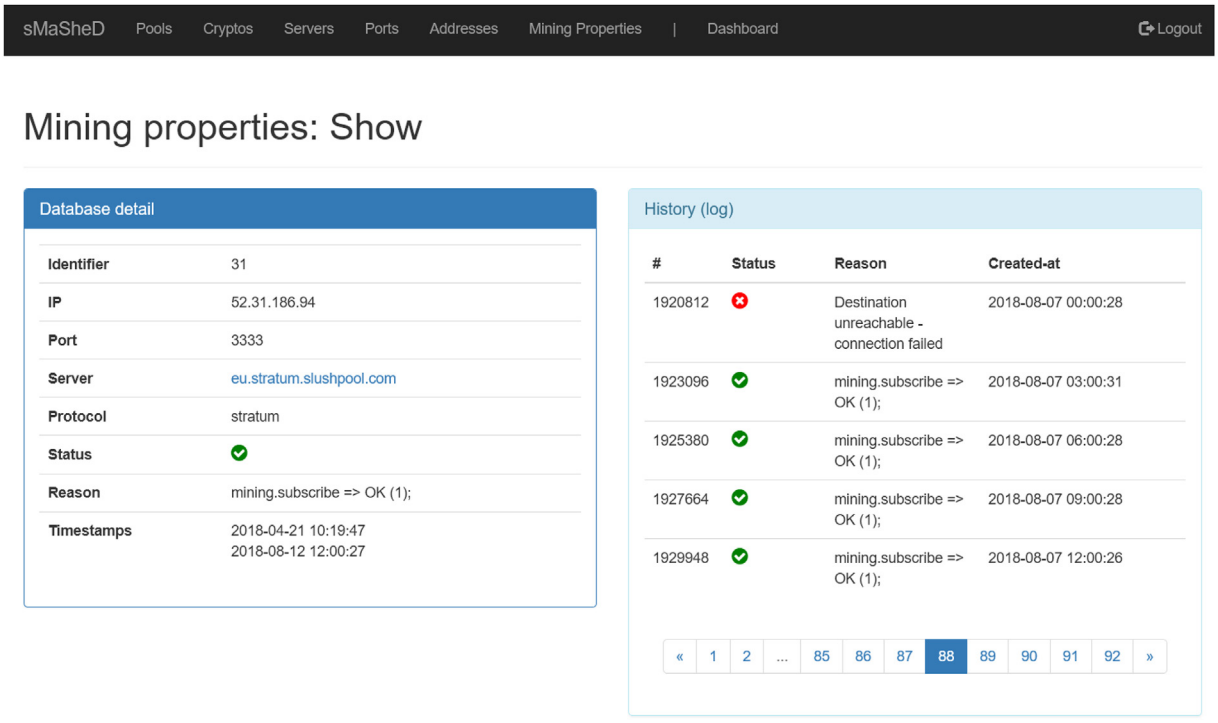


Fig. B.12. Log of probe attempts for a given mining server

References

Ali, S.T., Clarke, D., McCorry, P., 2015. Bitcoin: perils of an unregulated global p2p currency. In: Cambridge International Workshop on Security Protocols. Springer, pp. 283–293.

Ali, S.T., McCorry, P., Lee, P.H.-J., Hao, F., 2015. Zombiecoin: powering next-generation botnets with bitcoin. In: International Conference on Financial Cryptography and Data Security. Springer, pp. 34–48.

BBC.com. Russian nuclear scientists arrested for 'bitcoin mining plot'. published 9 February 2018. https://www.bbc.com/news/world-europe-43003740. (Accessed 31 July 2018).

Bitcoinorg. Bitcoin - open source P2P money. https://bitcoin.org/en. (Accessed 11

August 2018).
Blockchaininfo, 2018. Hashrate Distribution an Estimation of Hashrate Distribution Amongst the Largest Mining Pools. Blockchain Luxembourg S.A. https://www.blockchain.com/en/pools?timespan=4days. (Accessed 3 August 2018).
Btccom, 2018. Bitfury - Pool - btc.Com. Bitmain. https://btc.com/stats/pool/BitFury. (Accessed 3 August 2018).
Carbon Black, June 2018. Cryptocurrency Gold Rush on the Dark Web. Tech. rep.. Carbon Black, Inc. https://www.carbonblack.com/wp-content/uploads/2018/06/Cryptocurrency_Gold_Rush_on_the_Dark_Web_Carbon_Black_Report_June_2018.pdf
Cejka, T., Bartos, V., Svepes, M., Rosa, Z., Kubatova, H., 2016. Nemea: a framework for network traffic analysis. In: 2016 12th International Conference on Network and Service Management. CNSM), pp. 195–201. https://doi.org/10.1109/CNSM.2016.7818417.
Claise, B., October 2004. Cisco Systems NetFlow Services Export Version 9, RFC 3954. IETF. https://tools.ietf.org/html/rfc3954.
Claise, P.A.B., Trammell, B., September 2013. Specification of the IP Flow Information Export (IPFIX) Protocol, RFC 7011. IETF. https://tools.ietf.org/html/rfc7011.
CoinMarketCapcom, 2017. Bitcoin (BTC) — CryptoCurrency Market Capitalizations. CoinMarketCap OpCo, LLC. https://coinmarketcap.com/currencies/bitcoin. (Accessed 30 May 2017).
N. T. Courtois, M. Grajek, R. Naik, The unreasonable fundamental incertitudes behind bitcoin mining, arXiv preprint arXiv:1310.7935
Dashjr, L., February 2012. Getblocktemplate - Fundamentals, BIP 22, Bitcoin Project. https://github.com/bitcoin/bips/blob/master/bip-0022.mediawiki.
Dashjr, L., February 2012. Getblocktemplate - Pooled Mining, BIP 23, Bitcoin Project. https://github.com/bitcoin/bips/blob/master/bip-0023.mediawiki.
de Vries, A., 2018. Bitcoin's growing energy problem. Joule 2 (5), 801–805.
D'Herdt, J., 2015. Detecting Crypto Currency Mining in Corporate. Tech. rep.. SANS Institute https://www.sans.org/reading-room/whitepapers/threats/detecting-crypto-currency-mining-corporate-environments-35722.
Etherscanio. Ethereum top 25 miners by blocks. https://etherscan.io/stat/miner?range=7&blocktype=blocks. (Accessed 3 August 2018).
Eyal, I., Sirer, E.G., 2014. Majority is not enough: bitcoin mining is vulnerable. In: International Conference on Financial Cryptography and Data Security. Springer, pp. 436–454.
Frouzová, K., Zelenka, J., 2018. Pirát Ve Služebním Bytě Těžil Kryptoměny, Sněmovnu Zaskočil Účet Za Elektřinu. Byla Mi Zima, Hájí Se! Aktualne.cz. https://zpravy.aktualne.cz/domaci/pirat-ve-sluzebnim-byte-tezil-kryptomeny-snemovnu-zaskocil-u/r5c8095665ea211e8b19b0cc47ab5f122/. (Accessed 31 July 2018). published 23 May 2018.
Grinberg, R., 2012. Bitcoin: an innovative alternative digital currency. Hastings Sci. Technol. Law J. 4, 159–208.
Grunzweig, J., 2018. The Rise of Cryptocurrency Miners. Palo Alto Networks, Inc.. https://researchcenter.paloaltonetworks.com/2018/06/unit42-rise-crypto-currency-miners/. (Accessed 3 August 2018). published 11 June 2018.
Hampton, N., Baig, Z.A., 2015. Ransomware: emergence of the cyber-extortion menace. In: The Proceedings of 13th Australian Information Security Management Conference. Edith Cowan University, pp. 47–56.
Hern, A., 2014. Student Uses University Computers to Mine Dogecoin. The Guardian. https://www.theguardian.com/technology/2014/mar/04/dogecoin-bitcoin-imperial-college-student-mine. (Accessed 31 July 2018). published 4 March 2014.
Hofstede, R., Čeleda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A., 2014. Flow monitoring explained: from packet capture to data analysis with netflow and ipfix. IEEE Commun. Surv. Tutorials 16 (4), 2037–2064. https://doi.org/10.1109/COMST.2014.2321898.
Huang, D.Y., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., McCoy, D., Savage, S., Weaver, N., Snoeren, A.C., Levchenko, K., 2014. Botcoin: monetizing stolen cycles. In: Proceedings of the 2014 Network and Distributed System Security Symposium. NDSS, p. 16. https://doi.org/10.14722/ndss.2014.23044.
Johnson, B., 2014. The Advantages and Disadvantages of the Deep Web, Tor Network, Virtual Currencies and the Regulatory Challenges Thereof. Master's thesis. Utica College, USA.
JSON-RPC Working Group. JSON-RPC 2.0 specification. http://www.jsonrpc.org/specification. (Accessed 17 August 2017) published 4 January 2013.
Juels, A., Kosba, A., Shi, E., 2016. The ring of gyges: investigating the future of criminal smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 283–295.
Kharif, O., 2018. The Criminal Underworld Is Dropping Bitcoin for Another Currency. Bloomberg L.P. published 2 January 2018. https://www.bloomberg.com/news/articles/2018-01-02/criminal-underworld-is-dropping-bitcoin-for-another-currency. (Accessed 3 August 2018)
Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E., 2015. Cutting the gordian knot: a look under the hood of ransomware attacks. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, pp. 3–24.
Kolivas, C., 2018. Asic and Fpga Miner in C for Bitcoin. GitHub, Inc.. https://github.com/ckolivas/cgminer. (Accessed 4 August 2018). published 5 June 2018.
Kroll, J.A., Davey, I.C., Felten, E.W., 2013. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In: Proceedings of WEIS, vol. 2013, p. 11.
Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J.S., 2015. Bitcoin mining pools: a cooperative game theoretic analysis. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 919–927.
Litecoinpoolorg. Hash rate distribution (last 22 hours). https://www.litecoinpool.org/pools. (Accessed 3 August 2018).
Livadas, C., Walsh, R., Lapsley, D., Strayer, W.T., 2006. Usilng machine learning technliques to identify botnet traffic. In: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, pp. 967–974. https://doi.org/10.1109/LCN.2006.322210.
Nakamoto, S.. Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. (Accessed 30 May 2017) published 31 October 2008.
NiceHashcom. Official press release statement by nicehash. https://www.reddit.com/r/NiceHash/comments/7i0s6o/official_press_release_statement_by_nicehash/,note=H-BIT. (Accessed 3 August 2018) d.o.o., (published 6 December 2017.
Nield, D.. Student secretly used harvard's supercomputer to mine dogecoin. https://www.digitaltrends.com/computing/student-secretly-used-harvards-supercomputer-mine-dogecoin/. (Accessed 31 July 2018).
O'Dwyer, K., Malone, D., 2014. Bitcoin mining and its energy footprint. In: IET Conference Proceedings. The Institution of Engineering & Technology, p. 6.
Palatinus, M.. Stratum mining protocol. Slushpool.com. https://slushpool.com/help/manual/stratum-protocol. (Accessed 17 August 2018).
Palatinus, M.. Homepage — slushpool.com. https://slushpool.com. (Accessed 17 August 2018).
Pus, V., Kekely, L., Spinler, M., Hummel, V., Palicka, J., February 2014. HANIC100G: Hardware Accelerator for 100 Gbps Network Traffic Monitoring. Tech. rep. CESNET. https://www.cesnet.cz/wp-content/uploads/2015/01/hanic-100g.pdf.
Raeesi, R., 2015. The silk road, Bitcoins and the global prohibition regime on the international trade in illicit drugs: can this storm Be weathered? Glendon Int. J. Stud./Revue d'études internationales de Glendon 8 (1–2), 20. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.960.6973.
Exonerator, 2019. The Tor Project, Inc. https://exonerator.torproject.org/. (Accessed 2 September 2019)
Vesely, V., 2018. Mining Server Detector of Cryptocurrency Pools. GitHub, Inc.. https://github.com/kvetak/sMaSheD/. (Accessed 5 August 2018). published 4 August 2018.
Vesely, V., 2019a. Smashed - online catalog of cryptocurrency mining pools. Brno University of Technology. http://smashed.fit.vutbr.cz/. (Accessed 3 July 2019).
Vesely, V., 2019b. Smashed Docker Deployment. GitHub, Inc.. https://github.com/nesfit/jane-smashed/. (Accessed 31 July 2019). published 31 July 2019.
Vesely, V., Žádník, M., 2018. Pcap Files and Data-Sets for Digital Investigation Article. GitHub, Inc.. https://github.com/nesfit/DI-cryptominingdetection. (Accessed 11 August 2018). published 11 August 2018.
Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, third ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
Zhao, W., 2017. $30 Million: Ether Reported Stolen Due to Parity Wallet Breach. CoinDesk, Inc.. https://www.coindesk.com/30-million-ether-reported-stolen-parity-wallet-breach/. (Accessed 3 August 2018). published 19 July 2017.