Poster: Botcoin – Bitcoin-Mining by Botnets

Danny Yuxing Huang[§], Hitesh Dharmdasani[†§], Sarah Meiklejohn[§], Kirill Levchenko
Alex C. Snoeren, Stefan Savage, Nicholas Weaver[*], Chris Grier[*], Damon McCoy[†]
Dept of Computer Science and Engineering, University of California, San Diego
[*]ICSI & University of California, Berkeley [†]Dept of Computer Science, George Mason University [§]Students

I. Introduction

Bitcoin is a pseudo-anonymous virtual currency that is decentralized and free from any government regulations. Anybody—rather than the central bank—can create, or *mine*, bitcoins through a computationally expensive process that consumes considerable time and energy. In response, some botnets are taking over computers in order to mine bitcoins, earning profits at the cost of a victim's energy bill and even loss in productivity.

To our knowledge, this is the first instance where CPU cycles are stolen, abused and monetized at a large scale. Assuming a botnet with 10,000 compromised hosts each mining at 5 mega-hashes per second, we estimate that the botnet can make at least 3.3 bitcoins per day. At the current exchange rate, the daily profit is close to US $600; this number will be significantly higher if some of the compromised hosts have GPUs, whose parallelism can be further exploited for mining.

Evidently, this is a lucrative business. In this work, we aim to understand the business model of these botnets and estimate their profits. To this end, we analyze and manipulate bitcoin transactions in a way that would expose the true account information of botnets. From here, we can derive their earnings.

II. Technical Background

In order to mine a bitcoin, a user must find a nonce that causes a specific collision in a SHA-256 hash. Typically, this is accomplished by running a mining application that tries all possible nonce values via brute force.[1] Once a nonce is found, the miner is rewarded with 25 bitcoins and is said to have solved a *block*. Even though this process can be parallelized on GPUs, finding the right nonce consumes significant energy and can take a long time. Even with highly optimized GPU implementations running on a top-of-the-line AMD Radeon HD7970 graphics card, a single user can expect to successfully mine a block in two years. Furthermore, mining is competitive: all miners attempt to solve the same block at any given time, and if a miner manages to solve a block first then the other miners will have wasted their effort. Once a block is solved, all the miners start solving the next block.

To amortize such costs, bitcoin miners usually collaborate on the same blocks in groups known as *mining pools*. These are web services that combine the computation power of thousands of miners, allowing a block to be solved significantly faster than if done individually. The reward goes to the mining pool, which subsequently divides the newly mined bitcoins across the accounts of participating miners according to how much work they have contributed.

[1]Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." http://bitcoin.org/bitcoin.pdf

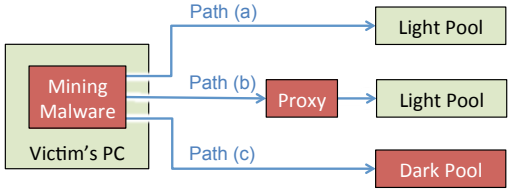

Fig. 1. Architecture of bitcoin-mining malware.

A miner typically registers an account with the pool and associate his *wallet address*—a public key for sending and receiving bitcoins—with the account. He then starts a mining client that communicates with the pool using the account's username and password. This information allows the pool to identify the miner and credit him appropriately. Every time the miner has contributed some work, the pool will automatically transfer bitcoins to the wallet address as payout. Only the mining pool knows the mapping of accounts to wallet addresses; even if a username-password pair is leaked, we are still unable to determine the owner's earnings. Whereas a solo miner can take months to earn 25 bitcoins, a pooled miner can receive smaller payouts on a daily basis.

Despite these advantages of mining pools, individual miners still have to invest considerable time, energy and hardware. Botnets avoid these costs by exploiting the victim hosts' CPU/GPU cycles. A host is compromised when the user downloads malware that contains a generic mining application—identical to what an honest miner would use, except that it is specifically pre-configured to mine for particular pools and accounts. Figure 1 shows three ways in which mining malware can make bitcoins for botnets:

Path (a): Malware can mine at *light pools*, which are public mining pools that anyone can join, and which an honest miner would use. The malware client is pre-configured for the perpetrator's account, so all mining credits accrues to the botnet operator, while the victim bear the brunt of the cost. This is the easiest to set up and requires no extra infrastructure from the botnet. However, the botnet's account information risks being exposed and blacklisted.

Path (b): Botnets can set up proxies to conceal their identity. Proxies connect to a light pool and relay work to malware clients without exposing the account name of the botnet operator. The user, or malware analysts, cannot directly determine the true account information or pool being used.

Path (c): Botnets can set up a *dark pool* for which malware mine. These are private pools that do not have a web interface and which the public cannot join. This conceals the botnet's identity and avoids commission fees associated with light pools. The downside is that extra infrastructure is involved; the botnet has to invest in configuring and maintaining a full-

fledged pool service.

This malware architecture helps botnets hide their identity and earnings. In the next section, we introduce several techniques that allow us to uncover the perpetrators' true account information and estimate their profits.

III. Methodology

When a piece of malware is blacklisted, the embedded mining pool, username and password are exposed. To determine botnets' profit, we have to pinpoint their wallet addresses, look them up in the public transaction record, and add up the revenue. Although all bitcoin transactions are public, in general a wallet address is not directly connected with specific usernames. Thus, to find the profit, we must find the username-wallet associations, the techniques of which are described below.

A. Finding Wallet Addresses Directly

In the simplest case, the username is the wallet address for one particular light pool: Eligius. This, presumably, eliminates the pool's cost in maintaining a username-wallet database. Several pieces of exposed malware were found to to mine at this pool. By looking up the exposed wallet addresses in bitcoin's public record, we were able to compute the respective revenue easily

For the majority of the mining pools, usernames and wallet addresses are disjoint. Fortunately, a handful of light pools publish payout statistics, including usernames and the respective earnings. For malware samples that mine for these pools, we looked up the exposed usernames in the payout statistics table and simply calculated the corresponding profit.

Still, most of the other light mining pools, especially high-traffic ones, are not as helpful. Some only announce top miners and their earnings; none of these top miners' usernames were found in exposed malware. Some pools publish per-user payout statistics under nicknames, which can be different from the usernames that mining clients use. We attempted to contact pool operators for specific exposed accounts; most were reticent, and we could only determine the wallet addresses for less than five accounts in this manner. Therefore, we need other means to indirectly establish username-wallet associations.

Dark pools, on the other hand, are entirely dedicated to malware mining. To determine the profit, we only need to find out the pool's wallet address, rather than those of individual users. Again, we need other ways to indirectly make pool-wallet connections.

B. Injecting Payout Signals into Light Pools

A pool gets a 25-bitcoin reward every time it solves a block. If it is a light pool, a user will be paid based on how much work he has contributed. All of these reward transactions appear in the public record, but we need to isolate the ones that exclusively belong to botnets from those of honest miners.

To this end, we have developed a technique that introduces patterns in how often botnets receive payouts from light pools. In short, we mined on their behalf. Every day, we randomly chose a malware instance from our collection. We mined for the pool using credentials embedded in the malware, increasing the botnet operator's earnings in that 24-hour period. Our GPUs were capable of computing more than 1.1 billion hashes per second in total. Having injected such mining signals, we needed to detect them across all the wallet addresses. Given a sufficiently long period, if a wallet address receives more payouts on or a short time after the days we mine, there is strong evidence that the wallet is associated with the botnet account. We can compute its revenue accordingly.

This approach is not applicable to dark pools. Even though exposed mining malware at dark pools contains usernames and passwords, all of them are used by botnets. It is sufficient to estimate the botnet's profit by examining the pool's wallet address alone, rather than the wallets of individual accounts. Our GPU's computation power is dwarfed by the overall mining rate of the pool. It is unlikely that we can sway the frequency at which the pool solves a block.

C. Pool Proxies

Pool proxies are essentially light pools. Because they run on private hosts, they look like dark pools. For the signal-injection technique to work, we need to distinguish dark pools from pool proxies.

Since the pool mining protocol is HTTP-based, any HTTP proxy can also be used to proxy mining requests. In the simplest case, the proxy does not modify the data, making it possible to connect to the mining pool through the proxy as a normal pool user. To test for this behavior, we created accounts at major light pools and attempted to log in to each account via a suspected proxy. As a control, we also attempted logins using non-existent randomly-generated account names. We managed to find at least one proxy to a light pool. However, this approach is not applicable if the proxy rewrites the authentication data. We needed a more robust way to find conclusive evidence for proxies.

To this end, we introduce a technique that can determine which pool has solved which block. Normally, a block can only be solved by exactly one pool. Once a block is solved, all the pools move on and tackle the next block. If a block is found to be solved by multiple pools, we can conclude that some of these pools are proxies to the real pool.

A block's content changes over time until it is solved and saved in the bitcoin public record. Such changes vary across different pools, but for a given pool the variations follow some pattern. First, we record the history of all changes in a block's lifetime—from its construction to its solution—across all known dark and light pools. Then, starting from the solved block in the public record, we explore the space of all possible changes that could have occurred to the block in reverse chronological order. If a block from our search space matches our historical record for some pool, then we conclude that the pool has solved the block. If more than a pool is found, with one of them being a light pool, then the other pools are proxies.

IV. Result and Conclusion

Our analysis shows that 74% of our malware samples mine at light pools, and 26% at dark pools and proxies. For light-pool malware whose usernames are traceable to wallet addresses, we estimate that the revenue is close to 7 bitcoins in the previous year. We also have conclusive evidence that links a particular dark pool as a proxy to a major light pool, because both of them have solved the same blocks. Given our progress so far, we are confident in finding more wallet addresses that are associated with malware usernames. These wallets will allow us to estimate the profit of botnet mining with greater accuracy. Our hope is to fully understand the ecosystem of malware mining and curb such monetization of stolen processor cycles.