2012 4th International Conference on Cyber Conflict
C. Czosseck, R. Ottis, K. Ziolkowski (Eds.)
2012 © NATO CCD COE Publications, Tallinn

Case Study of the Miner Botnet

Daniel Plohmann
Cyber Defense Research Group
Fraunhofer FKIE
Wachtberg, Germany
daniel.plohmann@fkie.fraunhofer.de

Elmar Gerhards-Padilla
Cyber Defense Research Group
Fraunhofer FKIE
Wachtberg, Germany
elmar.gerhards-padilla@fkie.fraunhofer.de

Abstract: Malware and botnets are one of the most serious threats to today's Internet security. In this paper, we characterise the so-called "Miner Botnet". It received major media attention after massive distributed denial of service attacks against a wide range of German and Russian websites, mainly during August and September 2011. We use our insights on this botnet to outline current botnet-related money-making concepts and to show that multiple activities of this botnet are actually centred on the virtual anonymised currency Bitcoin, thus justifying the name.
Furthermore, we provide a binary-level analysis of the malware's design and components to illustrate the modularity of the previously mentioned concepts. We give an overview of the structure of the command-and-control protocol as well as of the botnet's architecture. Both centralised as well as distributed infrastructure aspects realised through peer-to-peer are present to run the botnet, the latter for increasing its resiliency. Finally, we provide the results of our ongoing tracking efforts that started in September 2011, focusing on the development of the botnet's size and geographic distribution. In addition we point out the challenge that is generally connected with size measurements of botnets due to the reachability of individual nodes and the persistence of IP addresses over time.

Keywords: *miner botnet, botnet analysis, cybercrime*

1. INTRODUCTION

Malicious software (short: malware) is the key enabler for digital crime and thus poses a serious threat to the modern society. One of its many uses is the creation of botnets. These networks of compromised computers (bots) are controlled by a third party (botmasters) and provide a flexible toolset for various illegal activities, promising remarkable financial gain with a low risk of being caught. Examples for activities are the massive sending of unsolicited messages (SPAM), distributed denial of service (DDoS) attacks, or the automated extraction of sensible credentials such as account login information or banking details.

One of the most recent botnet cases is the so-called "Miner botnet", named after its capabilities of mining Bitcoins. It received major media attention after carrying out massive DDoS attacks

against German websites (a detailed list is publicly available at [1]).
In this paper, we provide a comprehensive analysis of the "Miner botnet". Our contributions are the following:

- We analyse design and development aspects of a botnet on a technical level, covering individual binaries, the command & control (C&C) protocol, and its infrastructure.
- We present the results of our botnet tracking efforts since September 2011 and provide a statistical evaluation of the collected data set.
- We motivate current developments of botnet monetisation practices with one of the first specimens using the computational power of infected systems for direct profit generation.

The remainder of the paper is structured as follows. Section 2 covers background information and related work. Section 3 continues with insights on the botnet's infrastructure and outlines the characteristics of this malware specimen including monetisation of different functional aspects. Section 4 details the results of our botnet tracking efforts and Section 5 concludes this paper.

2. BACKGROUND

Centralised Botnets. The concept of botnets originates from the idea of enhancing malware with the ability to connect back to a server upon infection. First known cases of centralised botnets appeared in 1998/1999 and are tied to the so-called "Global Threat Bot" (GTBot), the remote access toolkit SubSeven and the email worm PrettyPark [2]. When infecting the target computer, these specimens joined a chat room on a specified Internet Relay Chat (IRC) server, notified the botmaster about their availability, and posted information gathered to enable further action. Obviously, the server's role in this centralised infrastructure is to provide C&C capabilities to the botmaster. The concept of using central servers evolved over the years, including masking of C&C servers through techniques like DNS Fast-Flux [3] and Domain Generation Algorithms (DGA) [4]. However, one flaw remains to this type of architecture from a botmaster's view: Shutting down all central C&C instances takes control away instantly and renders the botnet useless.

P2P Botnets. In order to overcome the drawback of depending on central components, experiments with peer-to-peer (P2P) mechanisms in malware date back as far as 2002 to the Slapper Worm [5]. The advantage of this technology is that the C&C channel is embedded into the botnet architecture, thus significantly contributing to resiliency against countermeasures when used correctly. A game-changing event was the appearance of the Nugache Worm, first detected in 2005 and considered to be responsible for the creation of one of the first botnets with a successfully distributed C&C infrastructure, based on a P2P protocol [6]. Since then, other P2P botnets have been observed and analysed. Detailed case studies have been performed e.g. for Storm [7], Waledac [8], and Conficker [9].

Bitcoin. Cybercriminals are constantly exploring new ways to generate profit from their botnets. Therefore, it was only a matter of time until bots were abused for generating Bitcoins

(BTC), an experimental digital currency scheme that was published in 2009 [10]. Bitcoins are calculated within a P2P network of competing nodes that iteratively perform SHA256 hashing operations towards certain target hashes. The first node to calculate an output hash based on certain input parameters that is below the target hash can claim a fixed amount of Bitcoins for its solution. The repeated hashing serves as a proof of work among competitors, who frequently join forces in so-called mining pools. Transactions of Bitcoins are cryptographically secured by a public-key infrastructure and the history of transactions is embedded into the calculations. While anonymity of transactions was not a design goal, techniques exist to aggravate tracing the flow of money. Bitcoins appeal to botmasters because they provide a way to immediately exploit the computational power of the compromised machines for financial gain. Bitcoins can be traded against hard currencies like USD or EUR on special trading platforms.

3. THE MINER BOTNET

In this section we present the characteristics of the Miner botnet. First, we provide chronological context of the operation of the Miner botnet. Next, we outline the development methodology used by the malware authors. We then focus our analysis on the set of executables specified by the botnet version number 1999. This version was the most recent on September 12, 2011 when we started our activities. The analysis is split by functionality aspects; for each we motivate the monetisation connected to it, namely:

- Pay-per-install (PPI) service for third parties
- Bitcoin mining
- Extortion via DDoS attacks
- Theft of social network identities

A. Timeline of Events

We were able to identify activities related to Miner back as far as December 20, 2010. On this day, a URL that can be linked to the botnet because of identical filenames was listed for the first time in the Abuse.ch Malware Database (AMaDa) [11]. Continuing our research, we concluded that at the beginning of this botnet, the malware was exclusively deployed and controlled via central servers using domain names of the following pattern: “<word>-<number>.ru”, where <word> is a string e.g. “baza”, “golos”, “vn” and <number> an arbitrary number with two or three digits. Further related entries in AMaDa and investigation of binaries extracted from the botnet indicate it was mainly used for pay-per-install of adware and FakeAV in the first quarter of 2011. Beginning in March 2011, we found the distribution of a module for blocking access to the Russian social networks VKontakte.ru and Odnoklassniki.ru. We also identified the presence of an HTTP DDoS module since April 2011, but it is not known if the botnet was already used for attacks at this point. The first Bitcoin mining module appeared in late May/ early June 2011, at a time when mining became popular and Bitcoin calculation speed increased dramatically [12]. All of this information was gathered by comparing MD5 hashes of malware samples against their initial scan date on VirusTotal and other malware identification services available on the Internet. This type of botnet operation continued until July 2011, when the botnet infrastructure was migrated to a hybrid centralised/ P2P network as indicated in [13,14,15]. In August and September 2011, the Miner botnet carried out widespread DDoS

attacks against approximately 580 German websites. After September 17, 2011, only Russian websites have been targeted [1].

B. Botnet Topology and Command-and-Control Protocol

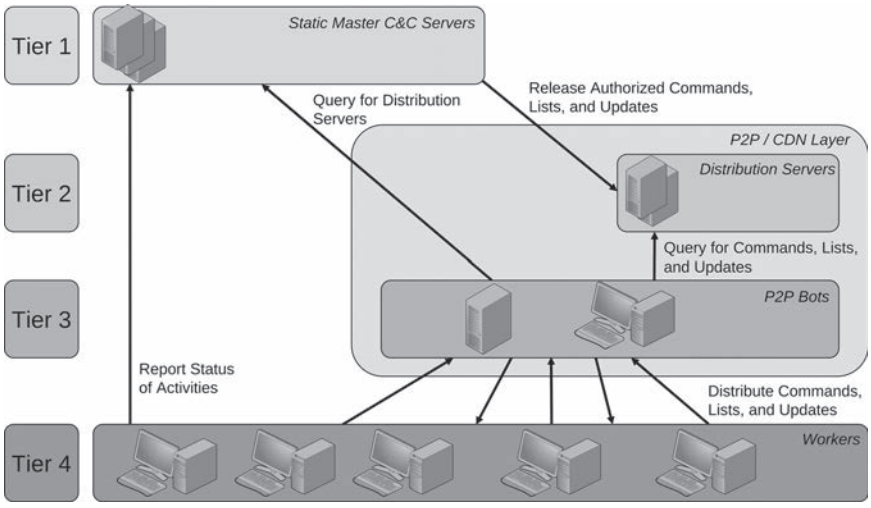
The topology of the Miner botnet can be divided into four tiers (Figure 1), sorted by descending relevance from the botmaster's perspective.

The top tier is represented by master C&C servers that are reachable through domain names and hard-coded IP addresses. These point to instances of nginx reverse proxy servers that are used to conceal the identity of the real C&C servers, which are operated by the botmasters and allow direct control and management of the botnet. Altogether, we identified less than 30 definitions of these fixed contact points. Most of them were not reachable during our investigations.

The second tier is defined through IP address lists published by the master C&C servers and consists of trusted peers that are internally called distribution servers. These servers are used to gather population information from the botnet, and to manage the connectivity as well as the flow of malware updates to the underlying tiers. They authorise and distribute lists for various purposes to third tier nodes. The third tier consists of all bots that are reachable from the public Internet and thus can be used as redistribution layer. In the following, we call nodes of this layer P2P bots.

The second and third tiers together form the P2P network of the Miner botnet. This network is primarily used as a malicious Content Delivery Network (CDN) and allows load balancing of binary transfers among its peers. It also serves as a backup layer for C&C in case the upper tiers are removed. The fourth layer consists of all remaining bots not reachable from the public Internet, e.g. because they reside in a private network. These bots serve as workers for operations like Bitcoin mining or DDoS attacks.

FIGURE 1. MINER BOTNET INFRASTRUCTURE.



The structure of the P2P communication protocol is shared by all tiers. The port used is fixed to 8080. In general, the protocol resembles HTTP GET-requests of the following structure "/search=<command{.txt}> HTTP/1.1". The URI path "/" and query variable "search" is static, while the actual command is appended as an argument. A query with the ".txt" extension serves as a status request and returns general information, e.g. the botnet version number or MD5 hash of contents to be transmitted by the actual command. For the full set of commands, see Table I. Answers to these requests have the structure of legitimate HTTP responses as generated by an nginx server, but are composed by the malware on the remote host.

TABLE I. P2P COMMAND&CONTROL PROTOCOL

Command	Answer (".txt")	Answer
error	-	returns an error code for the previously executed command
get_my_ip	0|<21 times "0">	returns the IP address as seen from the queried host
listen_test	0|<21 times "0">	requests the queried host to perform a connection check against local port 8081 in order to determine if the victim computer is reachable from the outside
test_server_r	0|<21 times "0">	this command is sent by a tier 3 node to a distribution server in order to validate if it is reachable from the public Internet
test_server	-	this command is induced by receiving the test_server_r command and performed by a distribution server
ip_list	0| <MD5 hash>	returns the most recent IP address list for subnet 1, the well-connected bots
Ip_list_2	0| <MD5 hash>	returns the most recent IP address list for subnet 2, the remaining bots
Ip_list_3	0| <MD5 hash>	returns an (empty) IP address list for the subnet 3
ddos_http_list	0| <MD5 hash>	returns a list of domain names to perform an HTTP-based DDoS attack against
ddos_udp_list	-	returns a list of domain names to perform a UDP-based DDoS attack against
btc_list	0 | <MD5 hash>	returns an IP address list for Bitcoin relay hosts
txt_server_list3	0 | <MD5 hash>	if the botnet is in fixed distribution mode, this command returns static download locations for the different modules
soft_list	<ver> | <MD5 hash>	returns a list of modules, each with its botnet version number, protected signature, filename and type ID
<filename>	<ver> | <signed MD5 hash>	returns the contents of the requested file

The communication protocol itself is not encrypted or obfuscated. For example, IP address lists are transmitted in plain text, with one quad-dotted IP address per line. The only mechanism of protection applied is a signature scheme for executable updates. Malware updates are delivered with an RSA-encrypted MD5 hash of the expected content. This is decrypted after downloading and checked against the actual MD5 hash calculated on the received data.

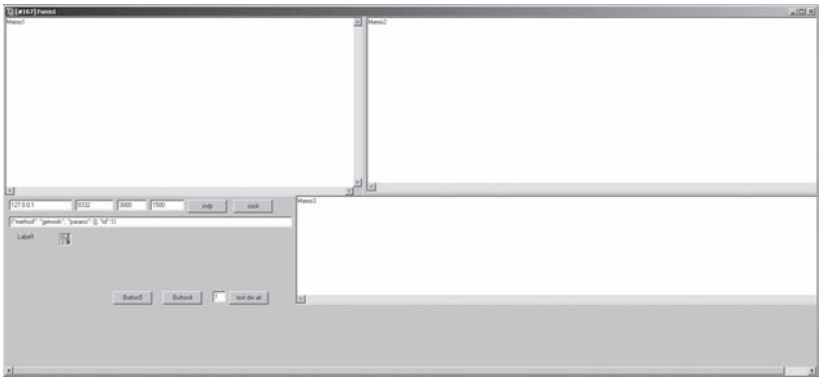
Besides the P2P communication, the static C&C servers and distribution servers are also contacted by bots on ports 80 and 62900 for submission of status reports and extracted data.

C. General Characteristics

Miner's code base can be characterised as a cluster of different malicious executables that are divided into modules according to their functionality. The functional groups identified can be categorised as infrastructure/ distribution, Bitcoin-related, DDoS-related, Social network-related, and additional utilities. The individual modules have no interdependencies. For example, the DDoS module can function on its own.

The dominant programming language used is Delphi version 2007. The majority of the malicious binaries have been developed by composing Delphi Graphical User Interfaces (GUI). Control elements such as memo and edit fields, labels, and buttons are placed on a form and assigned functionality, as shown in Figure 2. The overall behaviour is orchestrated by timers. While this approach may seem odd from a software development view, using a GUI allows easy live visual debugging of the modules by the malware authors. The visibility of the GUIs during runtime when deployed to victims is of course disabled so as not to raise immediate concern. The overall code base is structured into classes with different functional aspects, e.g. a class for nesting as a service, a class for refreshing IP address lists or a class for resolving its own IP address to geographical information. These classes are heavily reused among different modules.

FIGURE 2. A RUNTIME INVISIBLE GUI IS THE BASIS FOR FUNCTIONAL COMPOSITION OF SEVERAL MODULES, HERE FOR A BITCOIN MODULE (CLIENT_8.EXE). EXTRACTED WITH INTERACTIVE DELPHI RECONSTRUCTOR [16].



From 372 kilobytes (UDP DDoS module) to 1560 kilobytes (browser manipulation module), the footprint of the individual binaries is large compared with other malware. The reasons for the size are the mentioned graphical components and statically linked libraries. The authors make heavy use of third party open source products. The following libraries are present in all of the main modules: Internet Direct (Indy) [17] is used to implement communication interfaces and local web servers, Fast Giant Integers (FGInt) [18] supports implementation of a custom RSA signature scheme to protect malware updates, and RegExp Studio [19] allows matching of strings with regular expressions in various situations.

The binaries are not protected by any scheme that would harden them against analysis. Only

Ultimate Packer for eXecutables (UPX) is used in order to reduce the file size. The summarised file size of executables from botnet version 1999 is 17 MByte in decompressed state.

The only known spreading vector of Miner is social engineering of users through the social networks Facebook and VKontakte. As described in [14,15], the malware sends messages through stolen accounts to their friend lists and points users to fake YouTube videos that directly address the victim. In order to play the video, installation of a new Adobe Flash plugin is demanded, which is actually malware. The fake websites are directly hosted on P2P bots.

When a module is executed, it nests into the system in the following way. First, it copies itself to the Windows directory, either in the root directory or in a subfolder named "update.<number>" where <number> is a single digit number. The filename used for this purpose imitates typical Windows filenames (svchost.exe, svchostdriver.exe, sysdriver32.exe). It then restarts itself as a service and maintains presence on the system by enabling execution on system start-up. Malware configuration parameters are additionally stored in the Windows registry in module-specific subkeys that are also used for data sharing of timestamps or IP address lists. Most modules have a "close"-subkey that allows shutdown of the application through the registry by setting its value to "1".

D. Analysis of Botnet Version 1999

In the following, the different functionality aspects of Miner's malicious executable set of version 1999 are analysed. Each aspect is explained in context to the related binaries that represent the functionality. Furthermore the monetisation concept connected to the functionality is explained. We do not detail the changes between version 1999 and the current version 2103 because we did not identify major changes to the functionality.

1) Infrastructure/ Distribution

We first describe the core files responsible for integrating an infected machine into the botnet infrastructure. Next, we explain the different mechanisms of loading updates as well as SOCKS proxy and PPI as monetisation aspects.

loader2.exe – The first module to be executed on a freshly infected system is a loader that nests as a service called "srvsysdriver32" and then proceeds by performing an online connectivity test. If successful, it continues by contacting a random IP address from the embedded hard coded list of contact points, the so-called bootstrap list. As soon as a successful connection on port 62999 is established with a contact point, the loader continues by acquiring updated IP address lists of botnet peers with the commands "ip_list" and "ip_list_2". These first steps are similar for almost all modules. IP addresses from obtained lists are queried by the loader with the command "soft_list" in order to obtain the most recent list of modules. All files on this list are downloaded from different peers to establish full functionality on the infected system. Furthermore, a reachability test is performed with the command "listen_test" in order to determine whether the victim's computer can be accessed from the public Internet or not.

All of the downloaded files are first checked for a valid signature according to the implemented protection scheme. After successful signature validation, another check is performed against the modules type ID. If the type equals the ID of the distribution module and the reachability

test was positive, the node becomes a P2P bot, or else the module is not installed on the system, and the victim becomes a worker bot. All downloaded modules are registered to run on start-up and are executed to let them perform their initialisation.

The loader finally resolves its own IP to the corresponding country code and reports this information back on port 62900 to a list of master C&C servers, together with its module version number (1.66) and a unique identification number derived from the system's drive information and the computer name.

wdistrib.exe – The distribution module is the fundamental component of the flexible infrastructure of the Miner botnet. This module is only installed in case the machine is reachable over the public Internet. It does not install itself as a system service. When executed, hard-coded master C&C servers are contacted. Their authenticity is checked by comparing the content of a queried certificate against a fixed 13-digit number. After the authentication phase, a distribution level is queried from the server. This level decides whether a centralised or decentralised mechanism is used for distribution of malicious binaries. In either case, an IP address list of distribution servers is obtained. The entries of this list are then requested to ensure the own machine's reachability.

In the case of the centralised distribution level, a list of filenames is queried with the command "txt_server_list3" and the contents from the specified URLs are downloaded. The same is valid for current IP address lists and DDoS targets.

In the case of the two decentralised distribution levels, the value decides whether the victim's computer will be responsible for a network segment identified by an IP address list 1 or 2. From the list of distribution servers, a recent list of P2P bots from the chosen segment is requested. These bots have the same status as the victim's computer. The list is sequentially scanned for possible software updates, which allows injection of updates from any machine of the P2P layer. The refresh rate for IP address lists is set to 45 minutes. Independently from the distribution mode, downloaded files are offered to other infected machines that may contact the victim's computer.

Furthermore, a web server is opened with the purpose to serve a fake YouTube page that is used as the previously described spreading vector.

Lastly, a random port in the range of 10000 to 65000 is opened to serve as a SOCKS proxy service. This type of proxy is regularly used as an anonymisation mechanism and a well-known service in the cybercrime economy. The port number, IP address, country code, and result of a connection speed test performed against popular websites are reported back to the list of distribution servers. Depending on the speed test, nodes may be reassigned to subnet 1 or 2, the former containing the nodes that surpass a certain speed threshold.

This is also the first monetisation aspect of the Miner botnet, as the given information allows renting of compromised machines for the use as SOCKS proxy servers. We assume that the detailed information of country code and connection speed is used to justify individual pricing.

loader_rezerv.exe – This is a network-based downloader with the ability to install arbitrary executable files on a victim's computer. It contacts a range of hard-coded C&C URLs with the system identification number as well as its module version number (1.08). Upon connection, it can be commanded to download a file identified by a download ID from a given URL, together with the protection signature of the file. If download and validation are successful, a status message is reported back to the same C&C server.

gbot_loader.exe – The third loader has the malware to be spread directly embedded in its PE resource section. Upon execution, the system's geographical location is deferred and the payload is only installed when the IP address is associated with one of the following countries: USA, Canada, Australia, Great Britain, New Zealand, France, Germany, Sweden, The Netherlands, Italy, Belgium, Denmark, Swiss, Norway, Iraq, Israel, Qatar, Oman, Bahrain, or Japan. In the following, we refer to this module as the PPI module.

A detailed analysis of the observed payloads is out of the scope of this paper; therefore, we only give a short overview. In total, we extracted 11 different unique payloads from various gbot_loader.exe samples. All payloads differ massively from the binaries related to the Miner botnet. They are not written in Delphi and are protected against analysis. In 10 cases, a variant of the Max++/ZeroAccess rootkit was embedded. In one case we identified a variant of GBot/CycBot, a trojan downloader mainly connected to clickjacking.

The last two modules exemplify another monetisation aspect of the Miner botnet, pay-per-install. The presence of two separate mechanisms shows the importance of this feature to the botmasters. While loader_reserv.exe relies on the availability of the hard-coded domains and servers, gbot_loader.exe can and has been published through the Miner CDN to install third party malware.

2) Bitcoin-related Modules

The capability of Bitcoin mining is the most characteristic feature of the botnet and also responsible for its name.

btc_server.exe – This module is responsible for managing work distribution in the botnet and is only executed if the victim's computer fulfils the same properties as for the infrastructure distribution module. It serves as a proxy for the worker bots towards a selection of Bitcoin mining pools, clusters of miners that cooperate in order to increase their chance of gaining Bitcoins. It downloads one of the Bitcoin clients, namecoind or bitcoind, and joins a random mining pool chosen from a hard-coded list. These clients are used to backup the Bitcoin wallet containing earned Bitcoins. The wallet is posted every twenty minutes to a master C&C server. Furthermore, the module opens the ports 9442 and 9332 for Bitcoin communication with worker bots. Messages received by the workers are based on the Bitcoin JSON RPC protocol and delegated to the local Bitcoin client, which in turn forwards them to the chosen mining pool.

client_8.exe – This Bitcoin mining module is executed on bots of both tier 3 and tier 4. After nesting as service "srvbtcclient", a connection to the botnet is established and multiple operations are started in parallel.

Initially, an executable file named myunrar2.exe is downloaded from the Miner CDN. It works comparably to the utility geoip_unrar.exe and extracts the three embedded Bitcoin miners UFA Miner, RPCminer and Phoenix Miner. It then checks if the name of the video driver contains the string "radeon" and if so, checks for the driver revision installed. In case they are too old too perform Bitcoin mining on the Graphics Processing Unit (GPU) of the graphics card, the drivers are updated in the background through the vendor's website. After this, a speed test for the system is performed; the results including the system identifier, hardware information, the mining programme used and the hashing speed are submitted to a master C&C server. If an ATI graphics card is present, another test is executed on the GPU and the results augmented with detailed information about the graphics card are again posted to a master C&C server.

Further actions are the following. Every hour, a recent IP address list is obtained from tier 3 nodes. The IP addresses of this list are queried via a JSON RPC method for their current Bitcoin block count, which is returned if a btc_server.exe is running on queried node. In parallel, the successfully queried IP addresses are queried with another request for a portion of work. This allows the module to keep its own Bitcoin calculations at the current global network state. Finally, every five hours a status update about the mining operation is sent to a master C&C server.

The usage of Bitcoin mining on compromised machines is a remarkable development in botnets as it allows direct capitalisation through exploitation of computational power. While the Bitcoin currency has practically existed since early 2009, the first reports on malware used for stealing wallets of users were published in June 2011 [20]. After the Bitcoin exchange rate increased dramatically since late April 2011, with a peak of almost US$ 30 per Bitcoin in June 2011 [21], Bitcoin mining became economically justifiable for botmasters.

3) DDoS-related Modules

Next we explain the mechanisms of the DDoS modules and how the botmasters monitor attacked websites.

ddhttp.exe - The core module for DDoS attacks web servers via the HTTP protocol. It installs itself as a system service called "ddservice". After a connectivity check, it tries to download a list of DDoS targets. If the target list is acquired successfully, a status report with the unique system identifier and module version number (2.63) is sent to the contact point every 10 minutes. The following DDoS attack is performed with a randomly chosen User-Agent from a list of eight popular operating system and browser configurations. First, the IP addresses of the target host names are resolved and sanitised, i.e. the address 127.0.0.1 (localhost) is removed. Next, 10 concurrent threads are created to carry out the attack. In a first step, a connection to the target is tried in order to check if it is at all reachable. If the connection attempt is successful, the root path of the website is fetched. This page is then spidered for all links except RAR and ZIP archives, XLS, PDF, and DOC documents, executables, URLs containing "google.ru" or "cycounter" or email addresses. The attack then proceeds to request all the identified link targets to create even more load on the server.

udp.exe - The UDP DDoS component is a secondary attack module that was used during the massive attacks in August and September 2011. While the HTTP variant's goal is to exhaust the

web server application, the UDP module aims at saturating the network link of the target server. The module we analysed performs an attack against the hard-coded target “zenprotection.com”, a DDoS protection service. It will send fragmented UDP packets with a payload size of 32001 identical, randomly chosen bytes to a random port in the range of 10 to 65000.

pele.exe - The last module related to DDoS allows the botmasters to evaluate the success of their attacks. Similarly to the HTTP module, it tries to obtain a list of currently attacked websites. It proceeds by requesting the root page of the attacked website and evaluates the HTTP status code. In cases where the status code indicates a redirection (3xx), the redirection is recursively resolved and queried.

Besides, a WHOIS lookup is performed to identify the “netname”, i.e. the hosting service responsible for this domain. Next, the gathered data from these status checks is conducted into a report and submitted via a HTTP POST method to the master C&C servers. This procedure is repeated every two minutes. By inspecting the aggregated information from all reporting bots, the botmasters receive an almost real-time impression of how the attacked websites are reacting to the attack, and the detection of redirects or changes in reported features also allows them to adapt to countermeasures taken by their victims.

The monetisation aspect related to the DDoS functionality is extortion. While the first attacks were not connected to publicly known demands, following attacks were accompanied by emails from randomly generated yahoomail addresses requesting a payment of 100 Bitcoins to different account numbers for each target. We inspected eight Bitcoin account numbers identified through Google searches. According to the lookup service BlockExplorer [22], none of the accounts received incoming payment. The statement of Bitcoin as the desired payment method again underlines how firmly the botmasters pursue this virtual currency.

4) Social Network-related

The spreading success of the Miner Botnet in summer 2011 was heavily driven by the ability to interact with the social networks Facebook.com and VKontakte.ru.

iecheck12.exe – This module changes the Windows hosts file for static resolution of domain names with the intention to reroute the victim to a local proxy server when Facebook or VKontakte is accessed. This local proxy allows arbitrary interception and manipulation of website contents. On start-up, it queries current JavaScript files that are used to adjust the appearance of the social networks’ websites to the malware’s needs. Additionally, it downloads spam templates to be used for spreading.

Furthermore, the Geo IP database is used to determine the country the victim is situated in. Interestingly, the module also carries out the same functionality as the loader l_rezerv.exe by polling a list of hard-coded C&C servers for additional executables to download and install.

The core functionality of the module activates as soon as a victim logs into one of the mentioned social networks. The credentials are recorded and stored in the registry for multiple purposes. First, the credentials consisting of email address and passwords together with the system’s unique identifier and geolocation are reported back to C&C servers. Next, the credentials are

abused in order to initiate communications based on the downloaded spam templates with individuals from the victim's friend list. These communications are not visible to the victim. The goal of the communications is tricking the contacted person via social engineering to download and install malware. In terms of monetisation, this provides the botmasters with stolen identities that serve as a tradable good.

Further investigation of the binary revealed the presence of a telephone number that appears to be connected to another fraud scheme. The telephone number appeared in multiple forums of Russian language where users reported that a popup blocked their access to VKontakte, demanding for a payment to obtain an unlock code via the Russian mobile service MTS. The forum entries date to January and February 2011 and might give a hint of the early uses of the botnet. However, we were not able to reproduce the mentioned functionality.

Analysis of further files received by the module revealed a scamming scheme based on the injection of an advertisement for a fake Groupon offer. In the advertisement, a payment of US$ 200 via PayPal is offered in exchange for a transfer of 25 Bitcoins to a given account number. We checked the addresses on BlockExplorer and did not notice any payments to the Bitcoin account.

5) Utilities
Miner makes use of two additional executables that support functionality to prolong its presence on the infected system. Both are only used once when initially infecting the system.

geoip_unrar.exe – This module uses the RAR archive algorithm to decompress an embedded Geo-IP database that allows derivation of a geolocation from an IP address. The structure of the Geo-IP database used by the Miner Botnet is similar to a reduced version of the free IP address to country database available from MaxMind [23].

resetr.exe – In order to reduce the chance of being detected or removed from the system, this utility disables and deletes the services responsible for Windows Update functionality and removes the Microsoft Background Intelligent Transfer Service (BITS) that is used for the roll-out of said updates and is also used for signature loading functionality of Microsoft Security Essentials (MSE).

4. MONITORING THE MINER BOTNET

In this section we present the results of our monitoring operation for the Miner botnet. First, we explain the focus of our efforts and the methodology we used, followed by an analysis of the data gathered.

A. Focus and Methodology

The focus of our operation was to get insights into the population and activity of the Miner botnet. Based on our findings on the botnet infrastructure (cf. section 3.B), we concentrated our efforts on the P2P layer. Information about active peers, commands, and malware updates can all be observed on this layer.

Our approach is an adaption of the techniques that were used for the monitoring of other P2P botnets [7,8]. The general methodology applied is recursive enumeration, also known as crawling. Starting with a set of bootstrap nodes, each of the nodes is queried for IP addresses of its known peers. By collecting these IP addresses and repeating the procedure on the growing set, the network can be enumerated until no new IP addresses are observed.

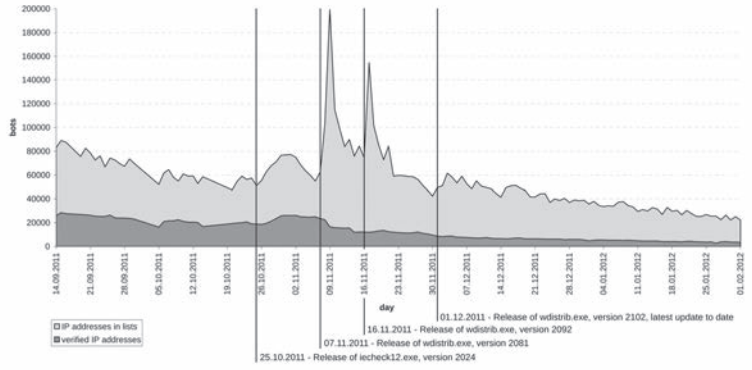
Applied to the Miner botnet, an initial bootstrap IP address list was extracted from the malware. The enumeration is done through the C&C protocol by using the command “ip_list” to query bots for their peer lists. The size of IP lists for segment 1 ranges between 80 and 250 and for segment 2 between 500 and 800 entries.

We created a tracking framework that implements this method and started crawling Miner’s subnets 1 and 2 on an hourly basis on September 14, 2011. We chose an hourly interval according to the refresh rates set in the P2P bots (between 30 and 120 minutes) and in order to keep a low profile on the network. To gather additional information, we used the commands “soft_list” to identify files offered and “ddos_http_list” to obtain attack targets when a successful connection is established with a bot.

B. Results

For this analysis, we take the data gathered between September 14, 2011 and February 01, 2012 into concern. We present the data of subnets 1 and 2 combined, because we did not notice any discrepancy caused by the separation by network speed.

FIGURE 3. DAILY POPULATION OF THE MINER BOTNET.



We differentiate between the number of IP addresses observed through lists and the number of bots we were actually able to communicate with. On average, we could connect to 22.91% of the IP addresses listed, with a maximum of 46.26% on November 06, 2011. These percentages are mainly influenced by the embedded bootstrap lists and dynamics of bots joining or leaving the botnet, as well as the timeliness of IP address lists published by the distribution servers.

Figure 3 shows the development of the daily botnet population over time. We observed between 23,000 and nearly 200,000 peers in the IP lists and between 3,000 and 29,000 actually reachable

hosts. We have linked four remarkable events to activities in the botnet.

- On October 25, 2011, an update of the module for interfering with social networks was released. This caused a visible increase in infections.
- On November 07, 2011, updates to the PPI module and distribution module were published. The temporary spike in observed IP addresses in the lists can be explained with changes to the botnet backend. The actual decrease in population is probably caused by the updated PPI module. We assume that the botmasters of the Miner botnet sold a part of their population at this time.
- On November 16, 2011, the distribution module is updated again, causing another temporary spike.
- On December 01, 2011, an update to the distribution module was published, which was the last update to date. Since then, a constant decay is observable.

Table II shows that the geographical distribution of the botnet is centred on the countries Ukraine, Russia, Poland, Romania and Belarus. These countries made up about 70% of all infected hosts during our entire monitoring time. The only remarkable observation is that Poland and Romania change their position in November 2011. This is related to the above-mentioned event of an update to the PPI module.

TABLE II. GEOGRAPHICAL DISTRIBUTION OF REACHABLE BOTS

	14.09.11	**12.10.11**	**16.11.11**	**14.12.11**	**11.01.12**	**01.02.12**
Ukraine	28,24	29,49	22,27	25,88	25,29	27,04
Russia	18,56	20,76	17,11	19,35	21,74	17,83
Poland	10,40	9,15	10,55	7,25	7,24	8,06
Romania	8,97	7,77	12,07	11,44	11,71	9,98
Belarus	4,20	4,86	4,70	4,40	4,15	3,58
remaining	29,62	27,97	33,30	31,68	29,87	33,50

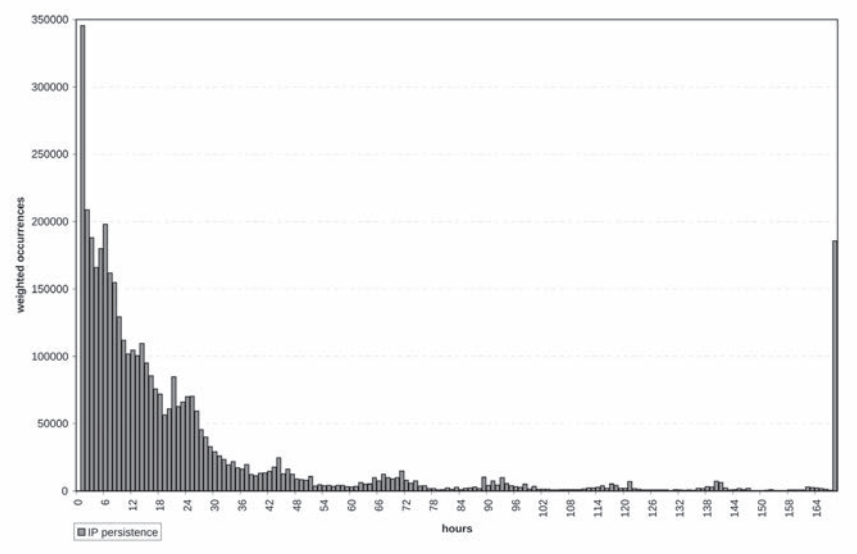
Furthermore, we analysed for how many hours single IP addresses were continuously present in the merged set of IP address lists. We took seven days from September 19, 2011 0:00 to September 25, 2011 23:59 as a sample, as shown in Figure 4. For possible intervals from 1 to 168 hours, the number of occurrences of an interval weighted with the number of hours it represents is shown.

The largest fraction is represented by short persistences with one hour occurrences being the most frequent. We assume this is at least partially influenced by the experience of a system infected with malware, especially in the case of Miner which increases system load through Bitcoin operations. The first significant drop-off is at about 6-8 hours which matches with the expected uptime of an office computer. The next decrease occurs after 24-25 hours and we conclude this is related to the enforced disconnect that many Internet Service Providers (ISPs) apply to their customers. The summarised weighted occurrences for 1 to 24 hours account for

72.43% of all occurrences, indicating that the majority of all observed persistences last for one day or shorter. This underlines that it is disputable to perform size measurements of botnets by counting observed IP addresses over longer time periods without taking the dynamics of the underlying systems and networks into concern, as has already been pointed out in [4].

The smaller peaks past the day mark are nearby multiples of 24 hours. We assume this to be caused by the way IP address lists are generated. The peak at 168 hours is caused by systems with a dedicated line and IP addresses that are constantly announced by the distribution servers. While having a strong impact in the representation chosen by us, these IP addresses account for less than 0.45% of all IP addresses observed in the given timeframe.

FIGURE 4. ANALYSIS OF IP PERSISTENCE IN THE WEEK FROM SEPTEMBER 19, 2011 TO SEPTEMBER 25, 2011.



5. CONCLUSION

In this paper, we have provided an overview of the Miner botnet. By taking this botnet as an example, we have motivated a selection of current techniques used by botmasters to extract money from their botnets. We outlined the chronological development of the botnet and its general characteristics. By this, it became obvious that the botnet owners have experimented with various methods for generating profits over time, adding and removing aspects, probably depending on how successful their activities were. We explained the layout of the hybrid infrastructure used in the botnet and detailed its capabilities and its C&C protocol. Furthermore, we presented our statistical data on its population and activities, gathered during four months of tracking efforts.

While the design and implementation used in this botnet are technically not on the same

level as of its more prominent competitors, the use of advanced concepts like a peer-to-peer infrastructure and RSA-signed updates indicate a trend that such features will become more and more common in all kinds of botnets in order to increase their resiliency against takedowns.

REFERENCES:

[1] F.Pfeiffer. *Minerbot Target List* [Online]. Available: http://www.ax10m.de/minerbot, Jan. 2, 2012 [Feb. 12, 2012].

[2] R. Ferguson. *The history of the botnet – Part 1* [Online]. Available: http://countermeasures.trendmicro.eu/the--history-of-the-botnet-part-i/, Sep. 24, 2010, [Feb. 12, 2012].

[3] The Honeynet Project & Research Alliance. "Know Your Enemy: Fast-Flux Service Networks," in *The Honeynet Project KYE Paper Series*, July 2007.

[4] B. S. Gross et al., "Your Botnet is My Botnet: Analysis of a Botnet Takeover" in *Proceedings of the 16th ACM conference on Computer and Communications Security*, 2009, pp. 635-647.

[5] D. Dittrich and S. Dietrich. "New Directions in Peer-to-Peer Malware," in *Sarnoff Symposium*, 2008, pp. 1-5.

[6] D. Dittrich and S. Dietrich. "P2P as botnet command and control: a deeper insight," in *Proceedings of the 3rd International Conference on Malicious and Unwanted Software*, 2008, pp.41-48.

[7] T. Holz et al., "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm" in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[8] B. Stock et al., "Walowdac - Analysis of a Peer-to-Peer Botnet," in *Proceedings of the European Conference on Computer Network Defense*, 2009, pp. 13-20.

[9] F. Leder and T. Werner, "KYE: Containing Conficker", in *The Honeynet Project KYE Paper Series*, March 2009.

[10] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System* [Online]. Available: http://www.bitcoin.org/bitcoin.pdf, May 24, 2009 [Feb. 2, 2012].

[11] abuse.ch. *abuse.ch Malware Database* [Online]. Available: http://amada.abuse.ch [Feb. 12, 2012].

[12] P. Wuille. *Bitcoin Charts* [Online]. Available: http://bitcoin.sipa.be/ [Feb. 12, 2012].

[13] T. Werner. *The Miner Botnet: Bitcoin Mining Goes Peer To Peer* [Online]. Available: http://www.securelist.com/en/blog/208193084/The_Miner_Botnet_Bitcoin_Mining_Goes_Peer_To_Peer, Aug. 19, 2011 [Feb. 12, 2012].

[14] R. Lipovsky. *Win32 DELF.QCZ: Trust Me, I'm Your Anti-Virus* [Online]. Available: http://blog.eset.com/2011/08/03/win32delf-qcztrust-me-i%E2%80%99m-your-anti-virus, Aug. 03, 2011 [Feb. 12, 2012].

[15] S. Duquette. *Win32 DELF.QCZ: Additional Details* [Online]. Available: http://blog.eset.com/2011/08/29/win32delf-qcz-additional-details, Aug. 29, 2011 [Feb. 12, 2012].

[16] *Interactive Delphi Reconstructor* [Online]. Available: http://kpnc.org/idr32/en [Feb. 12, 2012]

[17] *The Indy Project* [Online]. Available: http://www.indyproject.org [Feb. 12, 2012].

[18] Triade Systemm *Fast Gigantic Integers (FGInt)* [Online]. Available: http://www.submanifold.be/triade/GInt/gint.html [Feb. 12, 2012].

[19] A. Sorokin, *RegExp Studio* [Online]. Available: http://regexpstudio.com [Feb. 12, 2012].

[20] S. Doherty. *Infostealer.Coinbit* [Online]. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2011-061615-3651-99 Jun. 16, 2011 [Feb. 12, 2012].

[21] *Bitcoin Charts* [Online]. Available: http://www.bitcoincharts.com [Feb. 12, 2012].

[22] *Bitcoin Block Explorer* [Online]. Available: http://www.blockexplorer.com [Feb. 12, 2012].

[23] MaxMind Inc. *GeoLite Country* [Online]. Available: http://geolite.maxmind.com/download/geoip/database/ [Feb. 12, 2012].