InTELL
BY FOX IT

GameOver ZeuS

Backgrounds on the Badguys and the Backends

This whitepaper accompanies our presentation at Blackhat US 2015

Michael Sandee

Principal Security Expert
Fox-IT

This whitepaper accompanies the talk "GameOver ZeuS: Badguys and Backends" on Blackhat in Las Vegas, August 5, 2015. The presenters are Elliott Peterson of the FBI, Michael Sandee of Fox-IT and Tillmann Werner of Crowdstrike.

This paper describes the history of the ZeuS malware and also the background of the GameOver ZeuS group, which has operated for well over five years. Throughout this paper there are sections discussing the ZeuS origin, group composition, methods for fraud, and origin of fraudulent beneficiaries. Additionally, we will be discussing a much lesser known side of peer-to-peer ZeuS: its use for espionage.

Throughout this paper, we refer to acronyms or names commonly used in the malware research world. The reader will require some specialist knowledge of malware, especially the workings of financial malware, to understand the full scope of this document.

GameOver ZeuS, GOZ, peer-to-peer ZeuS, P2P-ZeuS and ZeuS3 are analogous to each other and refer to a ZeuS based malware family, which was active in the wild from September 2011 till May 2014. When we refer to the GameOver ZeuS group or peer-to-peer ZeuS team, we mean the group that operated around this specific malware variant and its predecessors.

Slavik is the nickname of the author of ZeuS, his real name is Evgeniy Bogachev. Slavik was indicted by the FBI in June 2014. Over the years, he used many different nicknames, however people close to him would still call him Slavik.

This paper would not have been possible without the help and hard work of my dear friend and colleague, Frank Ruiz. I would also like to thank all of my colleagues at the InTELL team at Fox-IT and our Senior Management for supporting our work.

Management Summary

In June 2014, the news of a Law Enforcement led operation targeting GameOver ZeuS was announced. In addition to action against the malware itself, the author named Evgeniy "Slavik" Bogachev was indicted.

Although Slavik's endictment was over a year ago, he is yet to be apprehended. An award of 3 million dollars has been announced for information that will lead to his capture.

The GameOver ZeuS group was a crime ring that focused on various financial frauds, most notably corporate banking account takeovers, with an estimated 100 million dollars of losses attributed to the group. However it is likely that the amount is higher, as the group targeted banks and victims in many different countries and has operated for many years, going back to at least 2009. No aggregate numbers of fraud losses attributed to GameOver ZeuS over this period are available, as there was not a single long running international investigation that collected information on this.

One of the methods of fraud that was started in the last year of the GameOver ZeuS, between 2013 and 2014, was the CryptoLocker ransomware, which was a simple way of extorting money from victims by encrypting their files and demanding money for the key. About 3 million dollars in money was paid to the operator of CryptoLocker, which was Slavik (and his affiliates), who also was the author of ZeuS.

The group itself, which called itself "business club", consisted of over 50 individuals who were involved in the various aspects of fraud. This included the fraudsters themselves, the persons recruiting and arranging mule accounts, the technical support team and various third party suppliers of other crimeware kits that could be utilized by the group. The group was well organized and was led by Slavik and one other individual.

While the group was directly associated with the GameOver or peer-to-peer ZeuS malware, it had migrated from the previous ZeuS 2.1.0.X variants, and even prior to that worked together simply utilizing the kit malware of ZeuS. While in the beginning the group was based more on a supplier-consumer relationship within the underground, over the years it grew into a well oiled fraud machine.

During our research of GameOver ZeuS, we encountered a number of search commands that were looking specifically for information regarding Foreign Intelligence services in Georgia, Turkey and Ukraine. This is rather unusual to find in financial malware, and has fed speculation it could be one of the reasons why Slavik has so far been able to evade capture. The search commands were found in 2013 and 2014, but actually it was found that the activity likely even predated the start of GameOver ZeuS in 2011 and was also executed from the ZeuS 2.1.0.X versions.

Overall, due to the size of the group, the amount of activity and the global scope of the attacks, this investigation was a long and complex one. And while the attacks were relatively simple, the international character of the frauds committed made the investigation and prosecution a complex task.

ZeuS history and ecosystem

The ZeuS malware has existed for nearly a decade, and has been one of the most popular and versatile tools used in the underground. While initially used solely by actors in the Eastern European and Russian regions, typically amongst those speaking Russian, it has quickly been adopted by actors all around the globe.

While ZeuS is a versatile malware kit that can be used for a variety of purposes, its key strength is in browser manipulation through the use of its dynamic configuration. This manipulation is achieved by a set of rules that tell the malware on which url pattern to take which actions. This is known in the underground as webinject, and is believed to be named by Slavik. The result is that pages which are loaded by the browser, regardless of the source being an HTTP or HTTPS resource, can be modified prior to rendering by the browser.

The modifications can be relatively simple, such as displaying extra input fields during the login process, allowing the fraudsters to then use that information to execute attacks on the site itself with the additional credentials, or use the information to enroll victims for other services or abuse other services that could be easily monetized. The other end of the scale is injecting entire javascript frameworks that were utilized to social engineer the victim for information, and then, on the bank side, automatically inserting and authorizing transactions.

But ZeuS was not used merely for banking fraud, the webinjects were only used by one specific group of people, others used it just as a piece of malware to log information that ZeuS collected from victims from either the keystroke logging, or the built-in POST data logging, which worked for both HTTP and HTTPS websites and stored passwords from certain programs

Chronological summary of ZeuS highlights

2005 2006 2007 2008 2009

Back in the period of 2005/2006 *"Slavik" had created ZeuS, the first publication about ZeuS was made at the end of 2006.*

In 2007 *the first large scale attacks took place, that used the ZeuS bank attack configuration called "webinjects" that became the defacto standard format for bank attacks since then and to this day. It was also one of the first attacks to use the hybrid attack model to beat two factor authentication, an attack which is still used with success to date.*

Several years of dramatic growth ensued, *with both actual customers but also software piracy leading to hundreds of users of ZeuS worldwide, and it soon was the most popular malware in this space.*

In January of 2009, *and likely even earlier, "Slavik" started working closely together with a group, named by researchers the "JabberZeuS" group. This group had firsthand access to the latest features of ZeuS, but also did feature requests for specific add-on functionality that helped the group execute frauds. ZeuS development continued with new additions of features and increasing version numbers.*

In 2010, *with a popular alternative to ZeuS, named SpyEye, gaining increasing popularity, "Slavik" did his disappearing trick and announced he would no longer support ZeuS, but that instead the SpyEye author would support his work.*

Various variants of ZeuS appeared, *which suggested the source code was in the hands of multiple people. One variant, introducing new advanced features requiring indepth knowledge of the code, was used by the group known as "JabberZeuS", the variant became known as the Murofet/Licat ZeuS variant. We simply called it by its version number ZeuS 2.1.0.X.*

like FTP clients. In this way the ZeuS botnet operators would collect vast amounts of data from their victims, which could range from a few megabytes for a small botnet to many terabytes for GameOver ZeuS. We believe the GameOver ZeuS group had obtained somewhere between twenty to thirty terabytes of data over the period of five years from 2009 to 2014.

Another often used feature in ZeuS was the ability to load other malware, which often was affiliate based malware such as clickfraud, or other pay per install type malware. These would allow botnet operators to increase revenue from their ZeuS infections.

As many versions of ZeuS were pirated and thus freely available, the skillset of the attackers using ZeuS varied a lot, hence it was impossible to generalize all ZeuS related attacks as sophisticated. Actually the majority of attacks were simple, and would because of this sometimes be even more successful than the complicated and advanced attacks. So even from the outside an attack can look rather simple, but it all depends on the capabilities of the attackers to turn an attack into a success.

After support for the official ZeuS stopped in late 2010, a number of variants of ZeuS appeared. This number increased after the source code became public in 2011, which led to the popular and widely adopted ZeuS versions such as Citadel, Ice-IX and KINS. The original ZeuS and variants of ZeuS remained popular tools that typically were stable and reliable. But since the disappearance of GameOver ZeuS and also the lack of updates for the many variants, popularity has dropped. Recently, other supported malware kits have been gaining popularity and have taken a lot of market share from ZeuS.

2010 2011 2012 2013 2014

In 2011 *the source code of ZeuS became public, and this was followed by years of ZeuS variants appearing from small limited distributed variants, to popular widely supported competitors, such as Ice-IX, Citadel and KINS. But also variants of ZeuS, that were tailored to execute click fraud instead of banking fraud.*

In September 2011, *the ZeuS variant known to researchers as Murofet/Licat or simply ZeuS 2.1.0.X, used by the JabberZeuS group, morphed into what we now know as peer-to-peer ZeuS, P2P-ZeuS or GameOver ZeuS (GOZ), named after a C&C gate gameover2.php.*

In spring 2012, *Microsoft DCU announced legal action against P2P-ZeuS/GameOver ZeuS, which actually did no harm to the actual P2P-ZeuS botnet, and devalued a lot of good research by exposing a large amount of intelligence information. The result was that a lot of the actors involved with P2P-ZeuS/GameOver Zeus changed their digital identities, making it hard for many of the researchers to correctly attribute the activity.*

P2P-ZeuS continued to evolve, *also the addition of Cryptolocker as a potential payload for some of the infections was increasing its notoriety. The damage done by Cryptolocker was often far greater than the financial damages. Additionally, Cryptolocker would run on thousands of systems, encrypting all files, while financial fraud was only committed on a small percentage of the systems.*

End of May 2014 *was D-Day for GameOver ZeuS, with both a technical takedown of infrastructure of both GOZ and Cryptolocker, takeover of the Cryptolocker DGA domains, and takeover of the peer-to-peer network of GOZ. Additionally, "Slavik" (Evgeniy Bogachev) was indicted.*

One of the interesting fall outs of the operation against peer-to-peer ZeuS / GameOver ZeuS, was the appearance of a new variant of this ZeuS after the takedown, without the peer-to-peer network. It was dubbed "newGOZ" among researchers, however it never rose to the level of sophistication of the original peer-to-peer ZeuS, and it was likely a trick by the original author to give away the source code and create a distraction. It was only active for a short while until it completely disappeared.

Inside the business club

The group that amongst researchers was known as the "JabberZeuS" group, was internally really known as "business club". The group used a number of communication methods, but most commonly the businessclub.so jabber server. However most members had a number of jabber accounts and could communicate with each other through any of them. This included jabber servers of individual teams, which committed fraud through the managed ZeuS service.

The core team of GOZ consisted of two leaders (of which one was Slavik), a support crew and a number of preferred suppliers. Apart from this core team, a number of users that were very close to the core team was involved in troubleshooting and implementing certain features. Slavik was the main technical operator who was responsible for managing the GOZ operations and arranging the backend infrastructure through various means. Slavik, however, was no Linux expert, and he hired external expertise for setting up various servers and securing them.

The support team was there to support various botnet operational systems, which supported maintaining or creating botnets by the customers, which included loaders and exploit kits. The default hybrid token-grabber attacks, which were optional but included by default, were supported by a dedicated webinject code writer. Other items from preferred suppliers were the Blackhole exploit kit and the Psyche/Cutwail spambot. Individual members could opt for their own services but also make use of the services provided to the team.

Peer-to-peer network

P2P-ZeuS, even though it used one coherent peer-to-peer network, had up to 27 different botnets, each with its own backend instance almost identical to the original ZeuS backend. Note that these 27 also included the debug instances and several botnets which were hardly ever used. Interestingly, many of these botnets already existed prior to the creation of the peer-to-peer version of ZeuS, and bots from the old 2.1.0.X version of ZeuS were migrated using updates to the new peer-to-peer version.

The peer-to-peer layer merely functioned as a reliable and robust communication mechanism, and a way to hide the next layers of the infrastructure in order to become more resistant to takedown activity.

This was quite successful, as for nearly three years the botnet remained active with only minor interruptions, even though it was extremely popular and widespread, with averaging around 200,000 infections active at any point in time.

Each backend was managed by a different person or group, who in some cases had their own jabber server to coordinate activity and attacks, apart from the activity organized as part of business club. This makes it harder to understand the true hierarchy of the group, and one could argue that there is no true hierarchy, just a network of suppliers and consumers of online crime services.

High dollar frauds

Business club's role was not just to be the support platform for the P2P-ZeuS malware, but also to serve as a platform to execute the frauds for the hybrid attacks, that were in the standard webinject configuration offered by P2P-ZeuS. Mainly targeting corporate online banking systems. These were both common mule accounts that were able to handle small amounts of money, and high profile corporate accounts that were set up with great care to handle hundreds of thousands to even millions of US dollars. Typically these accounts were created in countries like China, Hong Kong, Cyprus and Latvia by associates of the businessclub leadership.

Interestingly, while Jabber was used for a lot of the communication, both internally and to external partners and clients, the details of the specialized mule accounts were exchanged over a secured webmail server where non-descriptive aliases were used. Obviously, setting up such mule accounts was a costly operation, as it was also important to make sure the accounts were not blacklisted, if they would have leaked prior to use. The process of using these accounts was orchestrated and planned in great detail, when access was gained to corporate victim accounts with high dollar, in many cases multi-million dollar, value.

The most-used banking malware attack in P2P-ZeuS, which shipped in most configurations, was the hybrid token-grabber attack which was offered by default and was mostly responsible for the high dollar frauds. Internally, this system was also called the "World Bank Center". The users however could choose to not enable these attacks and rather only load their own attacks (webinjects), which could be for any country except Russia, and any type of service, such as online banking, stock trading, creditcard management or other services where victims would feel compelled to fill in their credentials.

Business club membership

To become a member of the business club there was typically an initial membership fee and also typically a profit sharing agreement. Note that the customer and core team relationship was entirely built on trust. As a result not every member would directly get full access, but it would take time until all the privileges of membership would become available.

The Backends

The backends of the GOZ botnets were very similar to the original ZeuS version, and to the untrained eye the differences would be almost unnoticeable. There was also the possibility to create sub-botnets, which were typically used for tracking infection campaigns and their performance, just as regular ZeuS botnets.

The differences of the backends were in the settings where jabber notifications were built in, and technically one of the features was the ability to extract a peer-to-peer seed list from the list of infected systems. Interestingly, instead of changing the database, an existing field “net_latency” was reused without even renaming it, now serving as field to store the peer-to-peer port a bot was listening on.

The peer-to-peer network was presumably initially intended as a backup mechanism to recover the network in case of a takedown, later it became the standard way of communication for everything from the standard command and control and stolen data channels of ZeuS, but also for the built in attack methods, and some custom variants of attacks that were added for customers. For this purpose, specific variables were added to the webinject format, which could be used to reference a peer-to-peer protected backend system.

Apart from the peer-to-peer network, which was only the first layer, there were additional layers of proxies, which protected the real IP addresses of the backends from becoming known. Even the users of the malware would log in to the individual backends via a proxy, as to not directly expose the backend IP address in case of an intentional or unintentional leak. However, the last few years GOZ made use of a high profile bullet proof hoster, which offered servers with a virtual IP address assigned to it, which was transported from another network using various tunneling mechanisms.

In some cases these IPs were obtained from cheap VPS systems, in other cases they were entire netblocks announced via BGP and then transported back to the ethernet segment where the actual servers were. In case the IP addresses were cut off, the hoster would simply get a new netblock and assign IPs from the new netblock to the servers and it would be good to go, this typically took less than a few business days.

Exclusive access to the boss

The way in which GOZ worked, was that a lot of the functionality to manage the botnet was part of the peer-to-peer botnet, e.g. sending updates and changing the configuration. To perform these actions, the builder of GOZ was able to join the peer-to-peer network using a seed-list and then required an RSA private key to perform the relevant commands to update the bot and configuration. Slavik was the person who executed these actions.

Apart from the standard bot and standard builder there also was a special bot that had an output file which contained a list of bots that was used as an initial seed list for the builder. Additionally, there was a special debug build that had the capability to provide detailed debug logging of the peer-to-peer network to debug any issue, this was used for example to understand the attacks that security researchers executed against the peer-to-peer network. Typically, the attacks were thwarted relatively quickly, and subsequently the bot was hardened to not allow the same attacks in the future.

Fraud operator groups

Looking at some of the people who were operating the malware, there were individual operators but some were groups with more than five members who worked together to execute fraud. The operators did not exclusively use GOZ, but also other malware variants. Some were using kit malware and others were also members of other private malware systems. Again note that there was a total of 27 different backends, of which some were unused and some used for debugging purposes, however the total amount of members was quite large.

Espionage

Some of the more unusual instances of GOZ, were specific botnets that were not used for typical fraud, but instead for espionage. One instance focused on Georgia and Turkey, the botnets contained a number of commands issued to specifically these countries, with queries which were very detailed, including searches for documents with certain levels of government secret classifications, and for specific government intelligence agency employees, and information about politically sensitive issues in that region. Additionally, some of the activity revolved around information from OPEC members, a clear sign that the information gathering was not purely politically motivated but also quite likely economically.

After the recent political changes in Ukraine, which led to a more pro-western government, one botnet which had been previously used for banking fraud, was then used for a large amount of infections in Ukraine to search for certain types of politically sensitive information.

ZeuS binary building and distribution

The ZeuS builder filename was originally “zsb.exe”, and interestingly, the builder for peer-to-peer ZeuS has the same name. However, it does not name itself ZeuS, but instead is called “Mapp”. But it is hard to find references to that name anywhere else, which is a common trend amongst malware authors, as it is hard to classify something which is not known by a very specific name, and researchers will keep finding new names to describe it, leading to more overall confusion. When operators talked about this version of ZeuS, they did sometimes called it “ZeuS version 3”, although that was likely because they had no better name for it.

```
Mapp Builder.
Build time: 11:25:53 25.09.2012 UTC.

Usage: zsb.exe <command> -<switch 1> -<switch N>

build                          Build bot or(and) configuration.
  -nologo                      Suppresses display of sign-on banner.
  -bid:[number]                Numeric ID of botnet, 0 - if this is update.
  -kbucket:[file]              K-Bucket file, URL or single IP:UDP Port.
  -config:[file]               Source configuration file.
  -private_key:[file]          Private key file of botnet.
  -subbotnet:[name]            Override subbotnet name (form configruation) with this name.
  -obot:[file]                 Output executable file of bot.
  -oconfig:[file]              Output configuration file of bot.
  -toupdate:[file]             Convert PE-file to update of bot (mark, encrypt and sign). File will be converted in place.
  -oproxy:[file]               Output proxy data file of bot.
  -sfile:[file]                Mark PE-file as protected from PE-infection. File will be signed in place.
  -mfile:[file]                Sign file. File will be signed in place.

dht                            DHT operations.
  -nologo                      Suppresses display of sign-on banner.
  -kbucket:[file]              K-Bucket file, URL or single IP:UDP Port.
  -config:[file]               Source configuration file.
  -private_key:[file]          Private key file of botnet.
  -put_config:[file]           Put configuration data to every node.
  -put_update:[file]           Put update data to every node.
  -put_proxy:[enable/disable]  Enable or disable private proxy data for every node.
  -ping                        Ping every node.

crypt                          Cryptographic functions.
  -nologo                      Suppresses display of sign-on banner.
  -newkeys:[bits]              Generate new PRIVATEKEYBLOB keys with bits size. Bits can be set from 384 to 16384 in 8-bit
increments.
```

As we described earlier, the builder has a number of functions, amongst which one is to build updates with a number of configurable settings, and another is to communicate with the peer-to-peer network to interact with it in a number of ways, including distributing configurations and updates. For interaction with the peer-to-peer network the builder needed a list of seed nodes, specified with the kbucket option, one such seed file was available on a system that was actually infected with a specialized version of the malware:

```
hxxp://95.211.XXX.XX:1800 /kbucket.bin
```

When we look at the version of the builder from 2014, compared to the version of 2012, we can notice a number of differences:

```
Mapp Билдер.
Время сборки: 21:03:23 03.03.2014 +04:00.

Использование: builder.exe <команда> -<опция 1> -<опция N>

build                          Build bot or(and) configuration.
  -nologo                      Не выводить стартовый логотип приложения.
  -bid:[number]                Numeric ID of botnet, 0 - if this is update.
  -kbucket:[file]              K-Bucket file, URL or single ID@IP:Port.
  -config:[file]               Source configuration file.
  -private _ key:[file]        Private key file of botnet.
  -subbotnet:[name]            Override subbotnet name (form configruation) with this name.
  -obot:[file]                 Output executable file of bot.
  -oconfig:[file]              Output configuration file of bot.
  -toupdate:[file]             Convert PE-file to update of bot (mark, encrypt and sign). File will be converted in place.
  -oproxy:[file]               Output proxy data file of bot.
  -sfile:[file]                Sign file. File will be signed in place.

dht                            DHT operations.
  -nologo                      Не выводить стартовый логотип приложения.
  -kbucket:[file]              K-Bucket file, URL or single ID@IP:Port.
  -config:[file]               Source configuration file.
  -private _ key:[file]        Private key file of botnet.
  -put _ config:[file]         Put configuration data to every node.
  -put _ update:[file]         Put update data to every node.
  -put _ proxy:[enable/disable] Enable or disable private proxy data for every node.
  -ping                        Ping every node.
  -enum _ text:[file]          Enumeration of all the nodes in the network to text file.
  -enum _ binary:[file]        Enumeration of all the nodes in the network to binary file (NODE _ DATA _ SHORT).

plugin                         Manage plugin
  -nologo                      Не выводить стартовый логотип приложения.
  -input:[file]                Source DLL file.
  -output:[file]               Output plugin.
  -private _ key:[file]        Private key file of botnet.
```

The newer version of the builder came both with built in rootkit (Nercurs) and new options, which included crawling the peer-to-peer network, and the inclusion of support for creating signed plugins from a DLL to load via the C&C server. One specific plugin that was seen, was a VNC component before the plugin VNC was actually built into the malware itself.

The crawling of the network resulted in a file with peer-to-peer network unique ids combined with their IP and peer-to-peer service port. While running the tool would simply iterate over nodes:

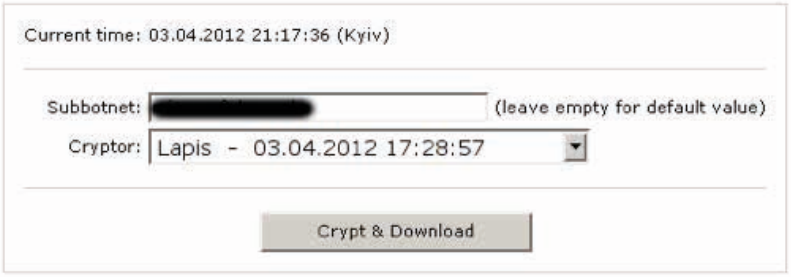
```
Node ID: aafd746a948c3dd249b1a6eb127399bd35f6258c.
UDP: 79.184.108.58:1393...OK
UDP: 79.184.108.58:1393...OK
       aaf06c12fc1db30f1733f11613e14b602146e99e     81.198.65.192:6600
              node already exists.
       aaf3fd3f2c0768145b14835fd5e5c769f5d7ba40     2.25.49.5:6882
              node already exists.
       aa98bacdf55997269c0b69aef279aafa9ae44194     195.235.31.90:3693
              node already exists.
       a8f3350e36ffa94d060a170d9c56eaef1a39fbcc     201.170.247.128:6891
              node already exists.
       a8711bc24a7d81088d6a019022133a2650223913     119.231.123.251:7782
              node already exists.
       af8c953678b96e2cde2354bd4e1544906971c368     81.130.105.174:5539
              node with this id already exists with other address 81.130.24.27:5539.
       afbd1037b4321efa4a93bf222e5cadb0dbccfc75     84.156.119.58:4674
              node already exists.
       afeacaa2005f42daf05c59de914a090a85e2316e     180.197.54.75:9226
       acc4fe3556402d5d45225015ed2f65273ad228ea     86.157.28.254:2553
              node already exists.
       ac9353a062f7eda103c0d75b15ea486d3e639341     172.245.217.122:3091
              node with this address already exists with other id c88937ba06ed
89bb67dab34171f02c775a79f75b.

Node ID: a878ee02272b2a5c4d6277c4d985d9b244774356.
UDP: 81.64.81.176:3879...TIMEOUT
```

When building an executable using the builder, a number of options have to be specified, which included the botnet id (bid) and the subbotnet name (subbotnet). The botnet id was limited to a number of 16bit, which was hardcoded into the output binary, where each defined number corresponded to a specific backend of ZeuS. The subbotnet names were typically used for identifying specific campaigns for spreading. Some where dated, and others were named descriptively after the spam service used or the spam theme such as "irs". Below you can find an overview of the different backend names and the botnet ids associated with them.

botnet name	botnet id	botnet name	botnet id	botnet name	botnet id	botnet name	botnet id
aqua	1111	it	9999	main6	3006	zpz	102
aqua2	2222	main	1212	vp	2000	play	101
azz	104	main1	3000	mr	1616	publo	1414
chrome	5555	main2	103	milan	2828	directoria	6666
fav	7777	main3	3002	spa	1717	debug	2222
grutik	1515	main4	3004	morgan	1144	debugr	65000
hard	8888	main5	3005	amr	100	solo	105

The operators of the individual botnets had access to a web based interface, which issued an executable crypted with one of the specified available crypters, tied to the botnet id assigned to the operator and also containing the embedded subbotnet name specified by the operator. The crypter services that were directly available to the operators were "lapis" (lps), "crypt4you" (c4u), "hardsys" (hrd) and "twcr". When operators tested crypters, the subbotnet name in some cases contained the abbreviation of the crypter such as "lps", "hrd", or "c4u". Below you can find a screenshot of the webbased builder:



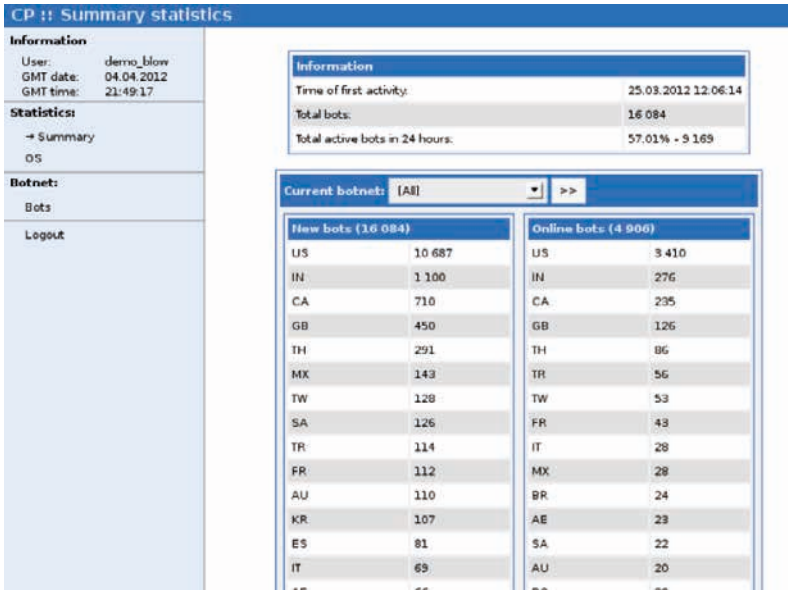

```
Address   Hex dump                                        ASCII
0012FAEC  6D 00 69 00|63 00 72 00|6F 00 73 00|6F 00 66 00| s.u.b.b.o.t.n.e.
0012FAFC  74 00 64 00|63 00 75 00|73 00 75 00|63 00 6B 00| t.n.a.m.e.t.e.s.
0012FB0C  73 00 00 00|9A 5B 5C 3C|44 FB 12 00|7C 24 52 77| t...?[\<D?R.|$Rw
```

The above memory dump shows how, after the static configuration in peer-to-peer ZeuS has been decrypted, the subbotnet name shows up. The "subbotnetnametest" was entered as input in the builder.

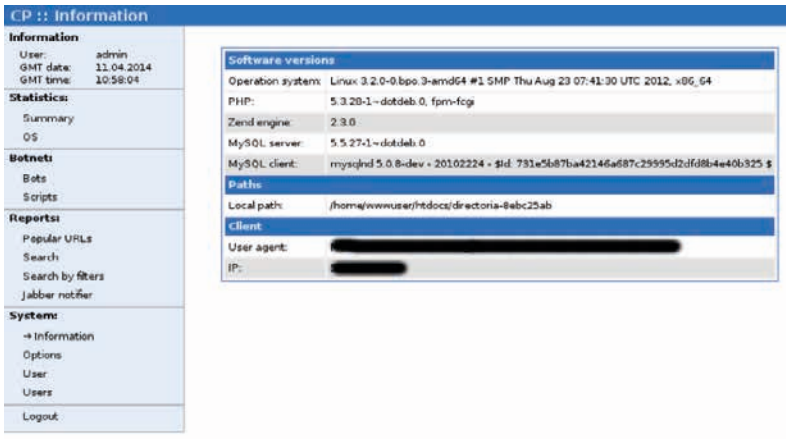
Operator's view of peer-to-peer ZeuS

The operators have access to a number of resources, the most basic is the ZeuS command and control panel that allows access to basic information from the infected systems, and allows the operator to issue commands. The panel is identical to the standard ZeuS panel which had been used for many years.

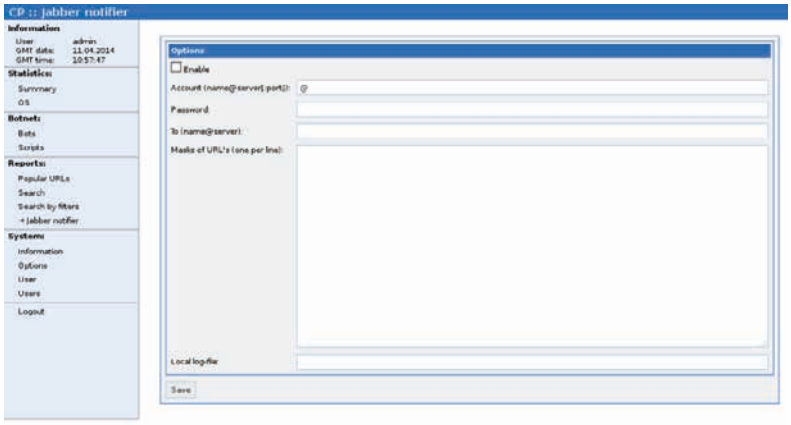
Basic overview of infected systems, using a restricted demo account that only has viewing rights:



The following screenshot shows the additional options in the menu that are available as panel administrator:



The jabber support, while not configured in this instance, allows jabber notification when reports for specified URL patterns are sent to the drop server:



In the Search options, an operator could search for data that was logged by the bots. This could provide additional data when defrauding a specific victim, both for complementing the regular banking frauds, and for looking for creditcard data even including the additional password, allowing the attackers to purchase online services easily. Additionally, the information could be used to assist certain operational actions, such as hosting of additional malware components on a site for which the credentials were compromised.

In the botnet scripts option, much like the traditional ZeuS command, scripts can be formatted that allow specification to which systems the commands should be sent, including for example bot id, subbotnet name and country.

Some of the most commonly used commands used by attackers are:

```
user _ destroy
user _ execute <url>
os _ reboot

bot _ bc _ add vnc <ip> <port>
user _ url _ block <urlpattern>
user _ url _ unblock

user _ flashplayer _ get
user _ cookies _ get
user _ certs _ get
user _ cookies _ remove
user _ flashplayer _ remove
```

Most of these commands are used in conjunction with fraudulent activity, to install additional tools to make fraud easier, to block a victim's access to their bank, connect to the victim's desktop, get a session cookie and soft certificate files, and removing session cookies so that the victim is forced to login, making ZeuS automatically log the credentials.

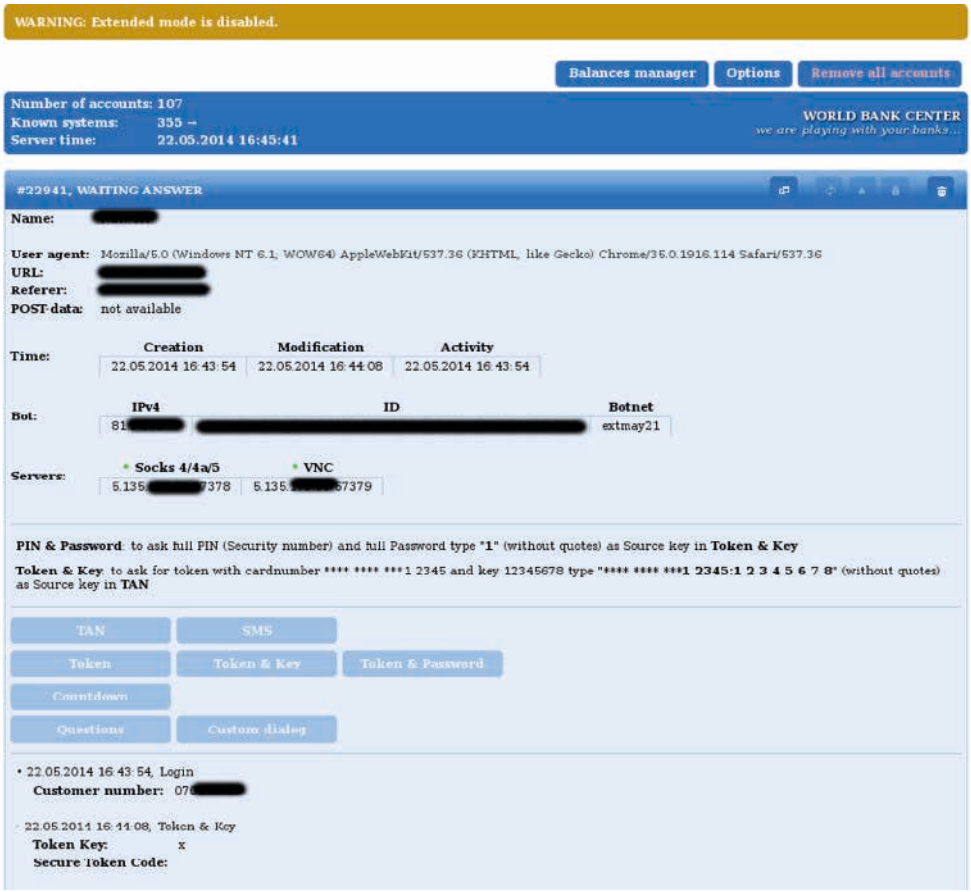
The user_execute command was used specifically for CryptoLocker installations too, where the "user_execute" command was issued only to US, Canada, Great Britain, Australia and New Zealand. Not all botnets that were spreading CryptoLocker were so specific, but in most cases they were specific, as CryptoLocker was only available in English.

How the operators execute fraud

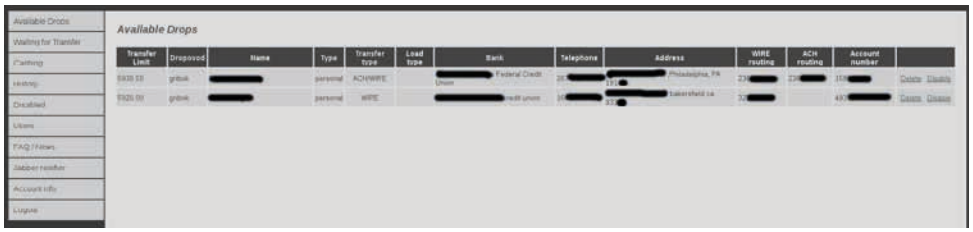
As most of the targeted bank accounts will have some forms of extra authentication when executing a transaction, merely grabbing the credentials using ZeuS is not enough. This is where the browser manipulation functionality of ZeuS comes into play which will modify the web responses from the bank of the victim, prior to rendering them.

The token-grabber attack in peer-to-peer ZeuS was a simple one, which was more or less always similar. The victim would see a normal, or almost normal, login page of their bank. For example, in case the login process of a bank consisted of two steps, it would be easier to have the initial page just with one extra field for an OTP code required for the second page. Directly after this step, the victim would be placed on hold, being shown the famous “Please wait...” loading message. During the victim being on hold, the browser would continuously poll the backend to check if new questions were available to ask the victim.

On the fraud operator side, a new “account” would appear in the Token Grabber panel, which contained various login details to start the session with the bank from the fraud operator side. The operator could use VPN services, socks proxies in the same country as the victim, but also a socks proxy of the victim machine to use the same IP address as the victim. Another option was to use a VNC connection to the victim system to use the same browser software as the victim. Seen below is the control panel the fraud operator used to communicate with the victim.



After a successful login, and when the account has enough balance and accounts with transaction capability, the fraud operator would then create a transaction. To create a transaction he needed a destination account for which the operators had a system containing mule accounts where transactions could be sent. Below you can find a screenshot of the system used by the peer-to-peer ZeuS group.



Social engineering the victim

The next step would be to further social engineer the victim for any additional authentication and authorization challenges, which would for example be TAN cards, index TAN, mobile TAN, Token OTP based, EMV Challenge-Response based, or even relatively simple knowledge based systems. For this you can see in the Token Grabber a number of pre-set buttons which will allow the fraud operator to quickly ask for any additional information that is required. In case of a challenge response system, the fraud operator will have to enter the challenge, which typically is just a number. In case the bank asks for a non-predefined question, the fraud operator can choose custom dialog and ask the question from the bank directly to the victim. This shows how this attack is a true man in the middle, still using relatively simple browser manipulation and scripts.

The victim will receive the questions, and based on the server side configured code and set parameters, the victim will see social engineering messages which are designed to make the victim understand why he has to enter the information. In some cases where victims could create transactions but required a second person to authorize them, the victims were social engineered by calling the authorizer over to the compromised computer to “unlock” the victim bank account.

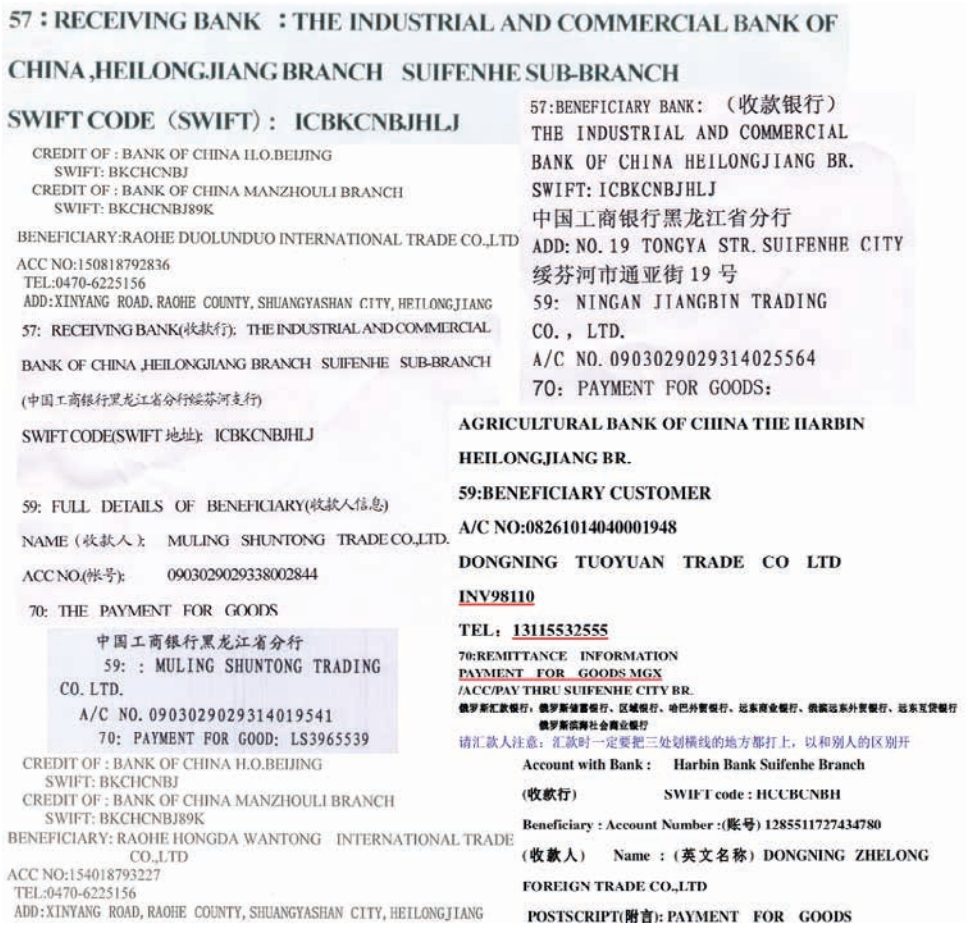
Typically, the result was that the victim received an error message, which is designed to make the victim believe he should not try to connect for a little while. In case of large frauds the attackers might even try more harsh measures, such as executing a distributed denial of service attack against the online banking site, or on one of the components required for login. This would stall the victim more as he could not log in, it would typically be hard to reach the bank as many customers would call due to the denial of service attack, and the bank itself would be in disorder due to this incident, potentially allowing the fraudulent transaction to slip through.

The corporate mule cities

While the group used a wide variety of mule accounts over time, one interesting collection of mules were corporate accounts located in two cities in China, both adjacent to a border crossing with East-Russia, north of Vladivostok. One is Raohe county and the other, further south, is Suifenhe, both are in the Heilongjiang province. They are marked on the map below.

While these cities were not the only places where mule accounts were opened, they did at least account for a large amount of corporate mule accounts over a certain period. The documents we obtained, showed a number of companies were opened, that operated under various names, pretending to be trading or shipping companies. We will show some of the examples we encountered that helped us understand the pattern.

The following are extracts of remittance information forms that were found to be used by the peer-to-peer ZeuS team. They were used to siphon large amounts of money, up to millions of US dollars, to these accounts from corporate and fund accounts, typically located in the US. As mentioned previously, all are located in either Suifenhe or Raohe county.



57 : RECEIVING BANK : THE INDUSTRIAL AND COMMERCIAL BANK OF CHINA ,HEILONGJIANG BRANCH SUIFENHE SUB-BRANCH

SWIFT CODE (SWIFT) : ICBKCNBJHLJ

CREDIT OF : BANK OF CHINA H.O.BEIJING
SWIFT: BKCHCNBJ
CREDIT OF : BANK OF CHINA MANZHOULI BRANCH
SWIFT: BKCHCNBJ89K
BENEFICIARY:RAOHE DUOLUNDUO INTERNATIONAL TRADE CO.,LTD
ACC NO:150818792836
TEL:0470-6225156
ADD:XINYANG ROAD, RAOHE COUNTY, SHUANGYASHAN CITY, HEILONGJIANG
57: RECEIVING BANK(收款行): THE INDUSTRIAL AND COMMERCIAL BANK OF CHINA ,HEILONGJIANG BRANCH SUIFENHE SUB-BRANCH
(中国工商银行黑龙江省分行绥芬河支行)
SWIFT CODE(SWIFT 地址): ICBKCNBJHLJ
59: FULL DETAILS OF BENEFICIARY(收款人信息)
NAME (收款人): MULING SHUNTONG TRADE CO.,LTD.
ACC NO.(帐号): 0903029029338002844
70: THE PAYMENT FOR GOODS
中国工商银行黑龙江省分行
59: : MULING SHUNTONG TRADING CO. LTD.
A/C NO. 0903029029314019541
70: PAYMENT FOR GOOD: LS3965539
CREDIT OF : BANK OF CHINA H.O.BEIJING
SWIFT: BKCHCNBJ
CREDIT OF : BANK OF CHINA MANZHOULI BRANCH
SWIFT: BKCHCNBJ89K
BENEFICIARY: RAOHE HONGDA WANTONG INTERNATIONAL TRADE CO.,LTD
ACC NO:154018793227
TEL:0470-6225156
ADD:XINYANG ROAD, RAOHE COUNTY, SHUANGYASHAN CITY, HEILONGJIANG

57:BENEFICIARY BANK: (收款银行)
THE INDUSTRIAL AND COMMERCIAL BANK OF CHINA HEILONGJIANG BR.
SWIFT: ICBKCNBJHLJ
中国工商银行黑龙江省分行
ADD: NO. 19 TONGYA STR. SUIFENHE CITY
绥芬河市通亚街 19 号
59: NINGAN JIANGBIN TRADING CO., LTD.
A/C NO. 0903029029314025564
70: PAYMENT FOR GOODS:

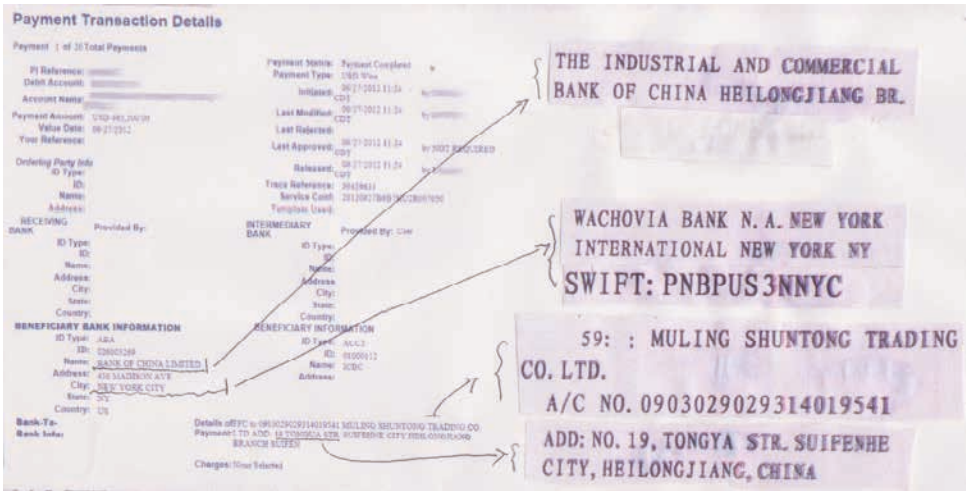
AGRICULTURAL BANK OF CHINA THE HARBIN HEILONGJIANG BR.
59:BENEFICIARY CUSTOMER
A/C NO:08261014040001948
DONGNING TUOYUAN TRADE CO LTD
INV98110
TEL: 13115532555
70:REMITTANCE INFORMATION
PAYMENT FOR GOODS MGX
/ACC/PAY THRU SUIFENHE CITY BR.
俄罗斯汇款银行，俄罗斯储蓄银行、区域银行、哈巴外贸银行、远东商业银行、俄属远东外贸银行、远东互贷银行
俄罗斯联邦社会商业银行
请汇款人注意：汇款时一定要把三处划横线的地方都打上，以和别人的区别开
Account with Bank : Harbin Bank Suifenhe Branch
(收款行) SWIFT code : HCUBCNBH
Beneficiary : Account Number :(账号) 1285511727434780
(收款人) Name : (英文名称) DONGNING ZHELONG FOREIGN TRADE CO.,LTD
POSTSCRIPT(附言): PAYMENT FOR GOODS

With a large amount of manufacturing happening in China, it is not uncommon for large transactions to occur to China. However the specific region of Heilongjiang is more known for Sino-Russian trade as there are no major shipping lanes from there to the US. So it would be uncommon for US companies to buy goods at companies in this specific region.

The specific area of Suifenhe started to develop several major projects for economic cooperation between China and Russia, which started in the first half of 2012. So it is not unlikely that peer-to-peer ZeuS associates would have made use of the positive economic climate and business friendly environment to open their businesses right there.

This shows that all around the world Free Trade Zones and other economic incentive areas are some of the key places where criminals can set up corporate accounts, as they are promoting business. And without too many problems, and with limited exposure, can receive large sums of money.

The fraudulent transactions created using the Token Grabber panel, were by themselves relatively straightforward. But as banking systems vary and international transactions can be complicated to execute, the fraud operators were given examples of how to set up the transactions and what methods would work best.



Note that while large transactions were more complicated to pull off, they did yield larger profits when the heists were successful. Still the peer-to-peer ZeuS group did not solely target corporate accounts but was also still targeting consumer accounts and credit card data, which seems a way to maximize profits from the investments.

Keeping tabs on the neighbors

As previously mentioned, some of the clearest espionage attacks in peer-to-peer ZeuS, and also prior to that in ZeuS 2.1.0.X, were targeted against Georgia, Turkey, and Ukraine. During our research, we found a large amount of search queries which were executed on the victim systems.

The search queries consisted of a number of keywords, which included contact information like email addresses, names and nicknames. Both the contact information and the generic keywords showed the type of information that was searched for. We have partially masked the names of the persons to protect their identity, and omitted the nicknames and personal email addresses. The total of Georgian entries was 106 and the total of Turkish entries was 11.

Set of Georgian keywords we have found:

Georgian Foreign Intelligence Service:
dir.int (at) fiss.gov.ge
admin (at) fiss.gov.ge
z.******vili (at) fiss.gov.ge
g.****nia (at) fiss.gov.ge
k.*****odze (at) fiss.gov.ge

Georgian Ministry of Internal Affairs:
z.******dze (at) mia.gov.ge
d.*******vili (at) mia.gov.ge
z.******vili (at) mia.gov.ge
r_******vili (at) mia.gov.ge
n.******vili (at) mia.gov.ge
a****n (at) security.gov.gew

Set of Turkish keywords we have found:

Turkish Ministry of Foreign Affairs:
g****n (at) mfa.gov.tr

Turkish KOM (Specialized police unit):
k*****i (at) kom.gov.tr

Set Georgian keywords used in 2013, mostly focused on locating "government classified" material:

ცნობა + საიდუმლო
საიდუმლო + ეგზ.
თანამსრომელი + თარირი + ეგზ.
დაზვერვ + ეგზ.
გეგმა + საიდუმლო
საგარეო + დაზვერვა
საიდუმლო + ანგარისი
ანგარისი + რუსეთი
სანდო + ურთიერთობ
სეიარარებ + რუსეთი
სსდსს + სამმართველო
სდდ + სამმართველ
ს.დ.დ + სამმართველ
დაზვერვ + სამმართველ
თანამსრომელი + ვყარო
ფათაქი + ურთიერთობ
საიდუმლო + ფირი
ფირად + ქონთაყთ
ფ.ს.ს + ვყარო
ფათაქი+ საიდუმლო
სფეცსამსახური + საიდუმლო
ფირად ქონთაყთ + საიდუმლო
დაზვერვ + ქრასნოდარ
დაზვერვ + სოვი
საიდუმლო + რუსეთი

Set of Turkish keywords used in 2013, focused on "government classified" material and information pertaining to the Syrian conflict and involvement of Russia with mercenaries and arms shipments:

gizlice + nüs
gizlice +zata mahsustur
Çok gizli + nüs
son +derece +mahrem
salt +kişiye + özel
hizmete özel + nüs
gizlice + operativ memuru
gizlice + harekatçı
gizli olmayan + operativ memuru
gizli olmayan + nüs
gizli olmayan + harekatçı
milli istihbarat teşkilati + gizlice
genelkurmay başkanlışı şstihbarat dairesi + gizlice
mit + gizlice
turhan + dilmaç
Çagatay +turkistan
gokhan +turan
aykut +unal
gizli+ emniyet genel mudurlugu
gizli + silahli teslim
gizli + parali askerleri
gizli + rus parali askerleri
emniyet genel mudurlugu + suriye
emniyet genel mudurlugu + parali askerleri
emniyet genel mudurlugu + silahli teslim
rus parali askerleri + suriye
kafkas parali askerleri + suriye
silahli teslim + suriye
militan kampi + suriye
gizli + emniyet genel müdürlügü
gizli + silahlı teslim
gizli + paralı askerleri
emniyet genel müdürlügü + suriye
emniyet genel müdürlügü + paralı askerleri
emniyet genel müdürlügü + silahlı teslim
rus paralı askerleri + suriye
kafkas paralı askerleri + suriye
silahlı teslim + suriye
militan kampı+ suriye
gizli+ milli istihbarat teşkilati
gizli+ emniyet müdürlüşü
gizli + emniyet müdürü
gizlilik kararı vardır
gizli + başkomiser
Çok gizli + emniyet müdürlügü
Çok gizli + emniyet müdürü
hizmete özel + emniyet genel müdürlügü
sınıf emniyet müdürü
hizmete özel + milli istihbarat teşkilatı
gizli + kişiye özel
istihbarata karşı koyma
casusluşa karşı koyma
anket + milli istihbarat teşkilatı
gizli +suriye +askeri operasyon
milli istihbarat teşkilatı
askeri + suriye

Set of Ukrainian keywords used in 2013, mostly focused on locating "government classified" material:

Особливої важливості
Цілком таємно
агент БЕЗПЕКИ
Головне управління безпеки
Федеральна служба безпеки
оперативний джерело
СЛУЖБА БЕЗПЕКИ УКРАІНИ
оперативно-розшукової
УСБУ ПЕОМ №

Concluding remarks

After looking at the whole set of search queries, it is quite likely that Slavik, who had set up and enjoyed full access to these specific ZeuS command and control servers, was involved in more than just the crime ring around peer-to-peer ZeuS. We could speculate that due to this part of his work he had obtained a level of protection, and was able to get away with certain crimes as long as they were not committed against Russia. This of course remains speculation, but perhaps it is one of the reasons why he has as yet not been apprehended.

For further enquiries please contact Eward Driehuis driehuis@fox-it.com +31 6 43824529

GameOver ZueS
Backgrounds on the Badguys and the Backends

In June 2014, the news of a Law Enforcement led operation targeting GameOver ZeuS was announced, in addition to action against the malware itself. The author named Evgeniy "Slavik" Bogachev was indicted. Although this was over a year ago, he has as yet not been apprehended. An award of 3 million dollars has been announced for information that will lead to his capture.

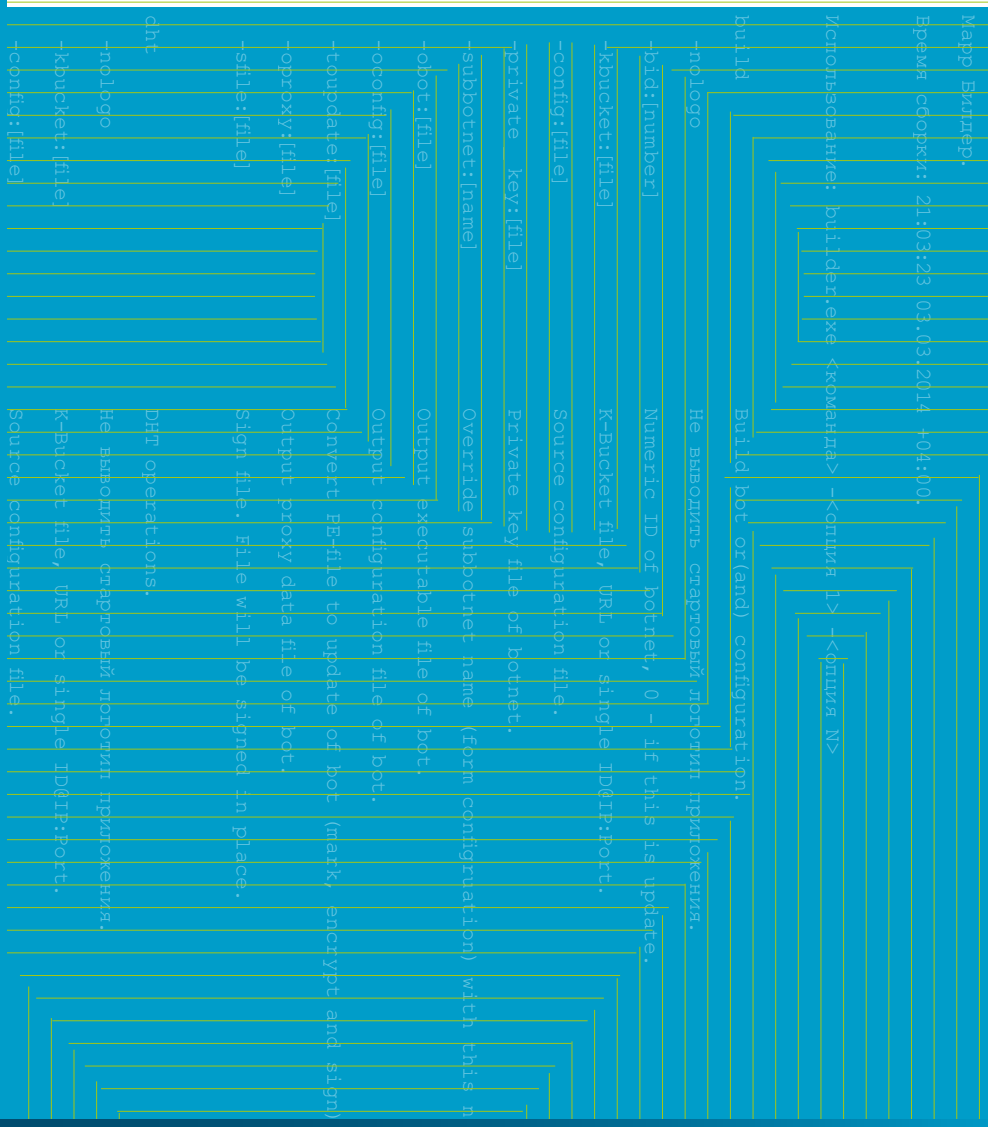
InTELL by Fox-IT

InTELL is Fox-IT's real-time threat intelligence product. Servicing over 50 banks, with hundreds of branch offices in four continents, InTELL is one of the fastest expanding intelligence propositions in the world. Our intelligence is based on actor attribution and context, giving a unique insight into the criminal ecosystem. InTELL features:

Global visibility - Learn about the global threat landscape; peer & sector threats, geographical trends, technical trends and actor attribution. Visualized, structured and indexed, full of relevant context.

Investigations - see threats to you, your peers and your technology as they unfold in real-time. InTELL scours the criminals online hangouts, and provides you with world class intelligence.

STIX / TAXII feeds - InTELL features one of the largest STIX repositories in the world. You will know where the threats are, who's behind it, how they do it and more, feeding into your infrastructure in real-time.



InTELL BY FOX IT

FOX-IT
Olof Palmestraat 6, Delft
PO box 638, 2600 AP
Delft
The Netherlands

t +31 (0) 15 284 79 99
f +31 (0) 15 284 79 90
e fox@fox-it.com

www.fox-it.com

601-001-EN