GEOST BOTNET. THE STORY OF THE DISCOVERY OF A NEW ANDROID BANKING TROJAN FROM AN OPSEC ERROR

Sebastian García
Stratosphere Laboratory and Czech Technical University in Prague, Czech Republic

Maria Jose Erquiaga
Stratosphere Laboratory and UNCUYO University, Argentina

Anna Shirokova
Avast Software, Czech Republic

sebastian.garcia@agents.fel.cvut.cz; mariajoseerquiaga@gmail.com; shirokova@avast.com

ABSTRACT

Maintaining a good operational security (OpSec) is difficult because it increases the cost of work and decreases the speed of actions. This is true both for security analysts and for attackers. This paper describes a new botnet, which we called Geost, discovered thanks to multiple OpSec mistakes made by the attackers. The mistakes included: the use of the HtBot malware's illegal proxy network; failing to encrypt the command-and-control servers; re-using security services; trusting other attackers that practise even less operational security; and failing to encrypt chat sessions. As far as we know, the Geost botnet has hundreds of malicious domains, 13 C&C servers, approximately 800,000 victims in Russia, and potential access to several million Euros in bank accounts. Moreover, the operational security mistakes led to the discovery of the names of members of an underground group related to the Geost botmasters. It is seldom possible to gain such an insight into the decisions taken by attackers due to failures in their operational security. This paper summarizes the mistakes and the risks taken by the botmasters, provides an overview of the botnet operation, an analysis of the victims, and a study of the social relationships of the developers.

1. INTRODUCTION

It has always been difficult to know exactly how botnet owners (botmasters) operate. It is a complex task to understand the details of their decisions, to see inside their command-and-control (C&C) channels, and to glimpse into their conversations. The three main reasons why it has been difficult to find this information are:

1. Malware authors operate some degree of operational security (from now on OpSec) in order to hide information.
2. The C&C channels are implemented using evasive techniques, such as random domain names, overwhelming analysts with information.
3. It may not legally be possible for analysts to access data and communications in remote servers.

With all these obstacles combined, the security community rarely sees how botmasters operate, make decisions, and protect their communications.

OpSec failures have been the reason for multiple important discoveries in cybersecurity. OpSec can be defined as a 'risk management process that encourages managers to view operations from the perspective of an adversary in order to protect sensitive information from falling into the wrong hands' [1]. The consensus is that OpSec decisions should be carefully designed to be effective against a certain risk. Problems in OpSec are not limited to technical mistakes but include mistakes made in the correct evaluation of the risks taken, and the countermeasures applied for protection.

This paper presents a very rare case of a chain of OpSec mistakes leading to the discovery of a new *Android* banking botnet targeting Russian citizens. It is unusual because the discovery was made when the botmasters decided to trust a malicious proxy network called HtBot. Our security laboratory had already been running samples of the HtBot malware for months when a traffic analysis revealed a group of infected computers being used to manage infected *Android* phones. The HtBot malware provides a proxy service that can be rented to provide *secure* connecting hosts for malicious activity. Our analysis of this HtBot communication led to the discovery and disclosure of a large operation infecting *Android*-based phones.

After the initial discovery of the Geost botnet, the method of analysis consisted of extracting more information about the attacks, the victims, the operations, its capabilities, and finally, about the group of developers related to the Geost botnet. Using pivoting techniques of threat hunting it was possible to uncover the C&C channels, the domains and IP addresses. Given that more than 72,600 victims were uncovered in just one C&C server, and there are at least 13 C&C channels, a conservative estimate of the total number of victims was calculated at 871,200.

The OpSec failures of the Geost botmasters were significant enough to allow us to recover a large amount of information. First, the attackers had a flawed risk model when choosing the appropriate communication platform for hiding their tracks. They picked up an illegal proxy network, not knowing that the network was being monitored by our laboratory. Instead of trusting a good communication provider, they trusted the security of a badly maintained illegal network. Second, the botmasters didn't protect their communications with several layers of encryption protocols – making it possible for us to see the content of their communications. Third, there was a leaked document on a public website that detailed the chatting activities of a group of developers working on the C&C website of the botnet. Since the chat was conducted over *Skype*, it is possible that it was leaked by a member of the group. Fourth, the chat log revealed that credentials were commonly passed unencrypted in the chat, giving access to very important information about them. In summary, a chain of small mistakes was enough to disclose the operation of a large *Android* banking botnet.

This paper makes the following novel contributions:

- Describes for the first time and names the Geost botnet, unknown to the security community until now.
- Provides an analysis of the OpSec mistakes that led to the discovery of the activities of a cybercrime group acting in Russian-speaking countries.
- Describes the complete infrastructure of the botnet and its victims.
- Publishes indicators of compromise (IoCs) and information to enable the community to act upon the Geost botnet.

- Performs a social analysis of the cybercriminal group discovered.
- Makes available for the research community, upon request, all the datasets in reference to the discovery of the Geost botnet.

The remainder of this paper is organized as follows: Section 2 analyses the previous work in this area; Section 3 describes the discovery of the Geost botnet; Section 4 shows how the botnet operates; Section 5 analyses the infrastructure of the botnet; Section 6 studies the victims of the botnet; Section 7 discusses the attackers, botmasters and developers; and Section 8 presents our conclusions.

2. PREVIOUS WORK

There are several examples of mistakes made by malware authors that have led to the discovery of their identities. However, they are usually regarded as technical mistakes rather than OpSec problems [2]. Technical mistakes are usually discovered as a result of poor OpSec criteria, e.g. code review. OpSec problems are hard to mitigate and they usually lead to the discovery of how botmasters operate or who they are [3]. Good OpSec can protect the user, but depending on the adversary, small mistakes can be very costly. One of the most famous OpSec incidents was that of Guccifer 2.0, the alleged persona that attacked the Democratic National Committee in the US, whose real affiliation was supposedly confirmed when Guccifer 2.0 apparently failed to activate their VPN during one login process [3]. This is an example of how hard good OpSec can be, even for experienced attackers.

A similar case of OpSec failure being taken advantage of by a powerful adversary was the identification of the owner of the Silk Road drug-selling site, Ross Ulbricht. Ulbricht was found because he used his personal email account to register other accounts related to his illegal site [4]. Although good OpSec is possible [5], cybercriminals also make mistakes that put them in jeopardy.

Practising good OpSec is hard, and it's harder when others try to force mistakes. In 2009 the Mariposa botmasters were captured because they connected to their servers directly from their homes. They usually used VPN services but after the police took their servers down (to force their hand), the botmasters panicked and connected insecurely. This paper provides an analysis of OpSec mistakes committed by a group of attackers while managing part of a botnet.

Regarding previous work on the Geost botnet, the only previous unnamed reference found was a post from September 2017 on the blog site *Virqdroid* [6]. This blog post analysed one of the malware's APK files, showed its technical qualities, and reported the IoCs. However, the blog lacked data about the threat, the attackers and the victims, and therefore conclusions could not be drawn as to the size of the operation or the identity of the *Android* banking botnet.

Probably the most well-studied part of *Android* banking trojans are the binaries themselves. This is because binaries are the first contact with the security community and usually the only source of information. The number of binaries related to *Android* banking trojans suggest that these threats have been rising during 2017 [7] and 2018 [8], although no scientific study has focused on a systematic analysis of the problem. *Android* banking malware is too numerous to describe, but a few important mentions can be made. In the early 2000s trojans Perkele and ibanking were well known for using SMS as a communication channel [9]. From 2014 there was a new era of banking botnets with the appearance of Slempo, Marcher, Shiz, BankBot and MazarBOT [9]. Their infection techniques, C&C protocols, and the attacks performed were significantly improved.

Analysing a malware binary is very useful, but the network traffic provides a different perspective. Even though some binary analysis may reveal network traffic [10], it is very difficult to capture traffic from the botmaster's actions. In this regard, this paper shows a novel discovery of real botmasters' actions while using their C&C servers.

3. DISCOVERY

The Geost *Android* banking botnet was discovered as part of a larger malware analysis operation in our laboratory. During an experiment in which a sample of the HtBot malware was executed [11], the traffic analysis revealed a very unusual communication pattern that stood out from the rest.

HtBot operates by converting its victims into unwilling private illegal Internet proxies. The infected victims relay communications from the HtBot users to the Internet. HtBot is regarded as an underground proxy network that is difficult for security analysts to tap, since its traffic is continually redirected to new victims. The users of the HtBot network pay the HtBot botmasters to provide them with high-speed, semi-private communications for their operations.

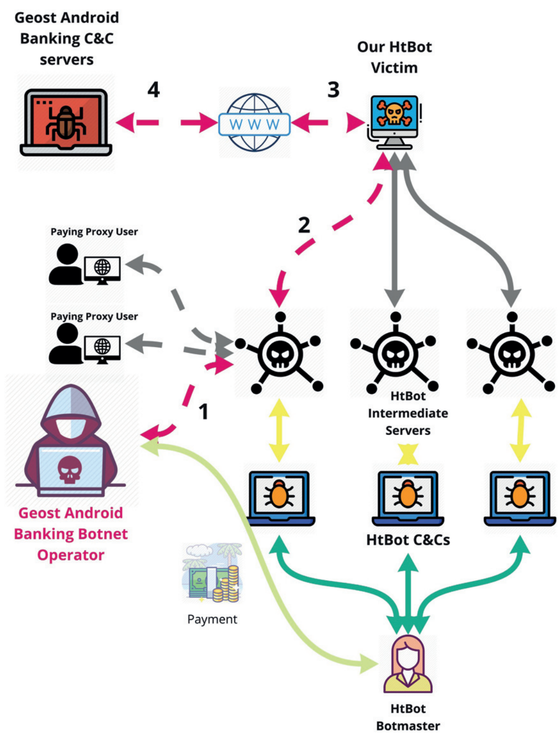

Figure 1: Discovery of the Geost botnet. A monitored bot of the HtBot malware was used by the Geost botmasters. First, the Geost botmaster connected to the HtBot network; second, the HtBot network relayed the data to our bot; third, our bot sent the traffic to the Internet; fourth, the botmaster accessed the Geost C&C server on the Internet.

Our laboratory was running and monitoring HtBot bots that were communicating with the Internet. Since these bots offered illegal proxy connections it was possible to capture all the traffic coming from the illegal users to the Internet. During the analysis of the network traffic of the illegal users, a pattern was discovered; this turned out to be the content of the C&C communication channel of the new Geost botnet.

Figure 1 shows the infrastructure operation of the HtBot malware and how it was used to find the new Geost botnet. When the botmasters of the Geost botnet connected to the HtBot proxy network they sent all their traffic through our victim bot, and therefore through our monitoring service. Therefore, all the information collected about Geost's actions comes from looking at the traffic going through our computer.

The analysis of the HtBot malware traffic revealed the pattern shown in Figure 2. This pattern was discovered thanks to two features that stand out: the large amount of traffic transferred and the lack of encryption. Transfers of such large amounts of unencrypted data are not common in a normal network. The use of unencrypted web servers for the C&C operation was the second OpSec mistake made by the botmasters. It is not clear why they neglected to use TLS encryption, since it is free and easy to install. The main hypothesis is that they may have had a large number of C&C servers and managing the certificates for them all would have been time consuming.

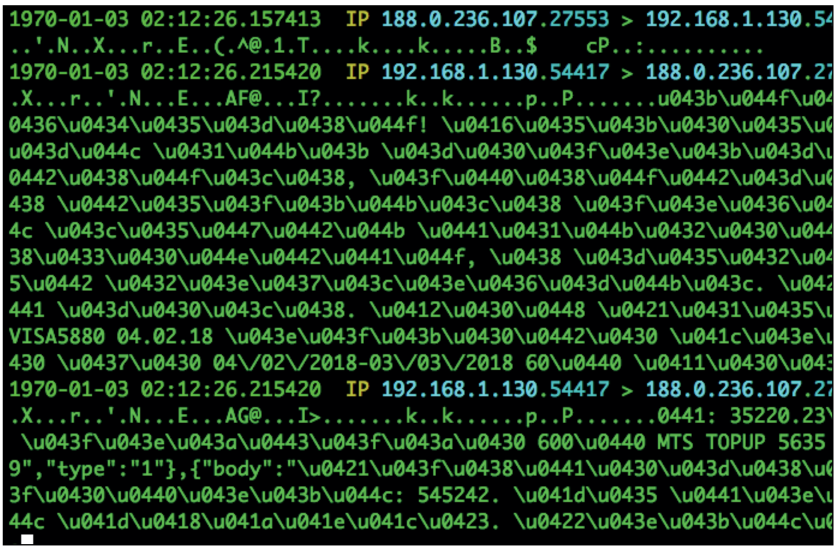

Figure 2: Unencrypted traffic pattern of the Geost botnet that helped to find it. This traffic was later found to correspond with the download of SMS messages from the Android phone victims.

The OpSec decision of the Geost botmasters to use the HtBot proxy botnet is believed to be based on the idea that an illegal proxy network may have better security than other alternatives, such as the Tor network [12], a commercial VPN network, or their own compromised servers on the Internet. The Tor network was probably discarded as a bad OpSec choice since it is known to be monitored [13]. The option of a commercial VPN has the disadvantage that the botmasters would be putting their trust in a private company that may be forced to submit its logs to the authorities. The third option, of compromised servers, may be the best from an OpSec point of view, but it would involve extending the current *Android* banking botnet with another layer of servers, infections, malware, monitoring, and maintenance. This option is much more costly than the rest. The decision to use the HtBot network may have seemed wise since it does not belong to a company, it's not usually monitored, and it handles its own maintenance. In the end, though, the decision to use the HtBot network was the first operational security mistake. It seems that the balance of probabilities and cost-benefit analysis were not correctly evaluated by the botmasters.

4. BOTNET OPERATIONS

The main advantage of accessing the botmasters' traffic while they were using the HtBot network was the possibility of a deep study of the attackers' decisions and actions. The analysis helped to identify a large botnet infrastructure, measure the size of the operation, and determine the goal of the botnet. Based on the evidence found, the Geost operation seems to consist of a large number of APK *Android* applications related to several topics, from banks and photo services, to fake social networks. Once the applications are installed it seems that they may be able to interact directly with the web services of five banks in Eastern Europe. It seems that one of the goals of the botnet is to access the personal information of the victims through their SMS messages, including those messages sent by the banks. The rest of this section describes the actions of the botmasters and how they helped identify each part of the Geost botnet. It is worth remembering that this was the traffic traversing our HtBot instance.

Access and actions in the C&C servers

The botmasters accessed the C&C servers through a web server using port 80/TCP. The web server was running *nginx* version 1.12.2. The first connection seen in the traffic was made on Sat, 10 Mar 2018 11:54:08 GMT and it was an access to the C&C server with the following request (not complete):

```
GET /geost.php?bid=c5d72910bd8a97aeb2ce
 7336fbd78a1f HTTP/1.1
Host: wgg4ggefwg.ru
User-Agent: Mozilla/5.0 (Windows NT 6.1;
 rv:48.0) Gecko/20100101 Firefox/48.0
Accept-Language: en-US,en;q=0.5
Referer: http://wgg4ggefwg.ru/geost.php
Cookie: SSE=p6ee96ki2knqrtsahdv84cuj04;
 __lnkrntdmcvrd=-1
```

From this request several things can be learned. First, that the botmaster was already logged in, because the cookie was already set. Second, that the botmaster was probably using a *Windows* computer, given the User-Agent. Third, that the domain was wgg4ggefwg.ru, and that the request was coming from the web page http://wgg4ggefwg.ru/geost.php. After this first request, the botmaster changed a note on one of the victims with the following request:

```
POST /stuff.php?mode=change_notes
(...)
bid=c5d72910bd8a97aeb2ce7336fbd78a1f&
notes=14.50+10.03+68.000
```

The fact that the botmasters put notes on individual victims suggests that they may have been after something more than automatic access to their bank accounts. After changing the note for a victim, the botmaster requested a list of SMS messages from a victim with the HTTP request `POST /stuff.php?mode=showSmsList`. The response to this request was a long list of more than 900 SMS messages from one victim. The SMS messages are analysed in Section 6.

The original HTTP response with the SMS list was a JSON file using Unicode encoding (\u chars) for transferring Russian characters. The following is an example:


```
{"response": [{"conversations":
{"+900":[{"body":"\u0421\u043f
\u0438\u0441\u0430\u043d\u0438
```


The decoded text in Russian is as follows (the password was redacted):

```
Списание средств: Platbox (RUB 120.00); пароль: 342365.Не сообщайте пароль
НИКОМУ. Только мошенники запрашивают пароли.
```

The English translation of this message is:

```
Withdrawal of funds: Platbox (RUB 120.00); password: 342365. Do not disclose
the password to ANYONE. Only fraudsters request passwords.
```

This SMS seems to be a message from the *Platbox* Russian payment system saying that 120 Russian Rubles have been withdrawn. Despite our initial assumption that the botnet was only looking for two-factor authentication messages, it is unclear why the botmasters are monitoring these messages. The first important remark is that the C&C stores the complete list of SMS messages of all the victims since the moment they were infected. The second important remark is that the SMSs were processed offline in the C&C server to automatically compute the balance of each victim. This can be seen in the C&C web page shown in Figure 4.

The SMS messages stored and used by Geost contained highly sensitive information. For example, a victim infected from July 2017 until March 2018 received the following SMS:

- Transfers in bank accounts:

  ```
  [redacted]Bank Online. Lada SE[name
  redacted]NA transferred to you 2500 RUB
  ```

  A message from a bank to a victim about money received.

- VISA balances:

  ```
  VISA5880 03/07/18 18:32 admission
  2500r Balance: 49866.86
  ```

  This information about balances was analysed automatically by the C&C channel.

Botmaster access to the login page

More than eight days after the first access, a botmaster showed up again to access the Geost C&C server. It may have been a different botmaster because the User-Agent of their browser was different from the first time. The first time, the User-Agent was *Mozilla/5.0 (Windows NT 6.1; rv:48.0) Gecko/20100101 Firefox/48.0*, which is a *Windows* computer. The second time it was *Mozilla/5.0 (Windows NT 6.1; rv:45.0) Gecko/20100101 Firefox/45.0*, which is an older version of browser in a *Windows* computer. Since it's very unlikely that the botmaster *downgraded* the browser, the conclusion is that these are different computers.

During this second access, it was possible to observe the complete login process and to obtain the master password of the C&C server. The long-term execution of the malware, which is standard policy in our laboratory, made possible the capture of this important piece of information. This connection also reveals the third OpSec error: the botmasters believed that it was safe to use the HtBot proxy network again. This is a huge underestimation of the security risk of using the same service twice. A better approach would have been to change the connection method every time.

The login request was sent as `GET /geost.php` and resulted in the login page shown in Figure 3. This page was reconstructed in our browsers by extracting the data from the traffic capture. The login page has an option to change the language between Russian and English, which suggests that the botmasters may speak either of those languages.

After the login page was presented, the botmaster logged in with the following request (not complete):

```
POST /stuff.php?mode=autorize HTTP/1.1
Host: wgg4ggefwg.ru
User-Agent: Mozilla/5.0 (Windows NT
 6.1; rv:45.0) Gecko/20100101 Firefox/45.0
X-Requested-With: XMLHttpRequest
Referer: http://wgg4ggefwg.ru/geost.php
Content-Length: 31
Cookie: SSE=epr0dr4qlejbgphtqppmmjrca0

pwd=[redacted]&language=ru
```

The password used was 15 characters long and included nine numbers and six lower-case letters. The fact that the password was leaked means that it would be possible for others to log into the C&C server. The password is not incredibly complex since it lacked symbols and upper-case letters, but it is considered strong enough to resist the casual brute-forcer. It is also worth noting that there is a typo in the name of the request parameter, which is '*autorize*' instead of *authorize*.

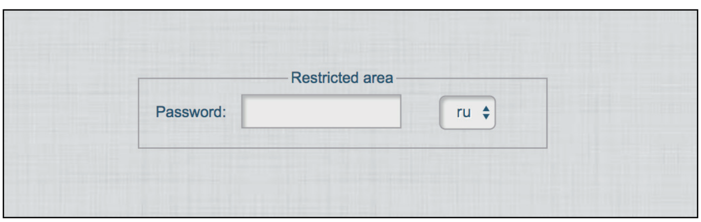

Figure 3: Login page of the C&C server of the Geost botnet. No TLS was used and no username is requested, only a password.

After logging in, the botmaster accessed the main panel of the C&C, shown in Figure 4. The main C&C web page is quite large, showing more than 7,500 infected phones and information about the version of the malware, IMEI of the phones, permissions of the malware, country of the phones, balance in the bank accounts, and much more. Figure 4 shows the following information for each victim:

- Status: whether the victim is online.
- ID: identification number of the victim assigned by the botnet.
- IMEI: code that identifies each cell phone.
- Rights: probably whether the malware has admin rights, or only SMS access, or both.
- Version: version of the *Android* operating system.
- Operator: phone operator.
- Country: country of the phone – it is not clear how this is obtained, but probably using the phone number.
- Balance: balance in the bank of the user.
- Category: it is not clear what is this menu for – the options are: Balance, Spam, Dead, Lok, Tupyat, Sliv, Credit, OTKLU4en, NULOVKI and ONLIKI.
- Flow: probably to identify how the phone was infected, given that the options are: marion1, dea and sitedub, which are related to APK applications.

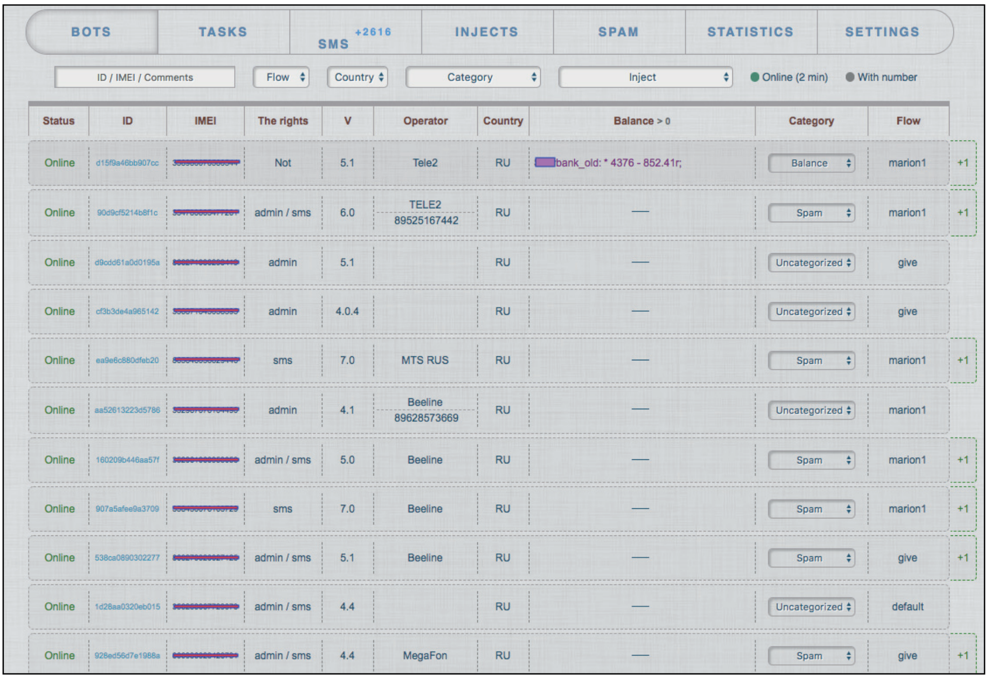

Figure 4: Main page of the C&C server of the Geost botnet. The C&C shows actions for injecting in banks and managing spam.

Features of the C&C

By looking at the options on the C&C page it is possible to infer the goals of the botnet and its main activities. From the top menu it can be seen that the management of injects (specific applications for each bank) is important, as well as the management of spam, SMS and Tasks. Under the menu `Поток`, which means Flow or Stream, there are the following options:

- svd2
- iYl5i8 (Photo Youla). [Site youla.ru]
- EBtiym (Photo Avito). [Site www.avito.ru]
- apkmontman
- CvKa5S (321)
- 2s1Kb1 (Antivirus_PRO)
- marion1
- sTPYWM (`Установка`) (means Installation)
- zGnI6m (`Вконтакт`) (translates to *VKontakte*, a *Facebook*-like Russian social network)
- dea
- `Установка` installation
- `Установить` to install
- sitedub
- BPg5nZ (123)
- wdbX4p (OK). OK.RU. https://ok.ru/
- q5Q9PR (Skype)
- QX3YrO (WhatsApp)
- GHf5Bt (Ula). https://youla.ru/
- I97CiN (Imo). ImoOnline.ru. Instant messaging app and VoIP.
- VAm5bd (VK)
- 2SUeYJ (Viber)
- wsmQDO (Telegram)
- 6NiFak (Yandex navigator)
- ge4twN (Badoo)
- mHhP71 (Shazam)
- udc13a (QIWI)
- gEc0m2 (Aliexpress)
- 9ObVTr (2GIS). https://2gis.ru/
- HaBxsX (ccleaner)

- RA6XMX (Clean Master)
- resur
- All
- NEuVxP (updateplayer)
- bjAVX1 (updateplayer2018)

The meaning of this menu is not completely clear but we suspect that it refers to a filter as to how the victim was infected, since all the options refer to *Android* applications. This theory was confirmed later when it was found that each botnet operator has its own 'Flow id' to determine how many infections they produced. After accessing the main C&C web page the botmaster requested to filter the victims by their online status using the following request:

```
/stuff.php?mode=filter_online
```

After the victims were sorted by online status, the botmaster sorted them by balance amount using the following request:

```
/stuff.php?mode=filter_balance
```

These two actions suggest that the intention was to see the online victims that have the largest balance of money, probably to act on them in some way, but no action was witnessed.

Banks attacked

By accessing the client-side source code of the web page in the network traffic, it was possible to identify which banks were the focus of the Geost botnet. The fact that only five banks were listed suggests that there is a special type of action that can only happen with those banks. It may seem as if the malware APKs or the C&C code could access and make transfers in accounts of those banks, but this hypothesis was not proven. For security reasons the complete list of banks will not be published until the banks acknowledge our contact with them. However, it is possible to provide the following characteristics of the targeted banks:

- The first bank is a Russian commercial neobank. One of the top five providers of credit cards in Russia.
- The second bank is one of the five largest private commercial banks in Russia and one of the top 1,000 world banks.
- The third bank is one of the three largest banks in Russia and Eastern Europe, and one of the top 40 banks in the world.
- The fourth bank is one of the 500 largest organizations in Europe and one of the leading banks in Russia.
- The fifth bank is part of a large group of cooperatives with subsidiaries in more than 15 countries, being in the top seven banks in Russia.
- The sixth bank is a publicly traded Russian payment service provider operating electronic online payment systems in Russia, Ukraine, Kazakhstan, Moldova, Belarus, Romania, the United States and the United Arab Emirates.

5. BOTNET INFRASTRUCTURE

The infrastructure used by Geost is large but not extremely complex. To date, 13 C&C IP addresses, more than 140 domains, and more than 150 APKs files have been found. The domains seem to be randomly generated, but not with a complete domain generation algorithm.

Randomness in Geost

Domain generation algorithms (DGAs) are algorithms that generate domains in a pseudorandom way. This is used as a mechanism to avoid detection and hide the C&C server by resolving a new IP address very quickly. Since the algorithm is unknown to the analyst, they are usually unpredictable. However, the malware author knows the algorithm and therefore can predict which domain will be requested. The attacker then registers the domain with an IP address they control. There are usually three main ways to identify a DGA algorithm: (1) the domains seem random, (2) dozens of domains are requested very quickly, and (3) most of the domains do not have an IP assigned to them. However, in the case of Geost, the domain generation algorithm is very unusual. It looks random enough, but each sample only attempts to contact one domain. Also, all the domains found so far *do* have an IP address assigned. It is not clear, then, how the domains are assigned to each sample, but it appears that each domain is assigned to one sample. The DGA used in the Geost botnet is character-based, uses letters and numbers, and the TLD is .ru or .xyz. The only domain that broke the rule was g877855hrg.ru.com.

The following is a sample list of Geost C&C domains:

- w23t2t2tfwg.ru
- wg34gh34t.xyz
- 32r3t23wef.ru
- ijsdggrur.ru
- wgg4ggefwg.ru
- 52t34tyt43.xyz

Another novel feature of Geost in reference to randomness is the use of an algorithm to generate PHP file names. This is not strictly DGA since they are not domains, but the random principle is the same. The main difference between a classic DGA and the PHP file generation algorithm is the purpose. While classic DGAs are intended to prevent the discovery of the botnet domains and subsequent takedown, the PHP file generation algorithm prevents the generation of signatures to find and block those names. It is not simple, for example, to create a YARA rule that matches a DGA domain using a random PHP file. The PHP filenames are 32 characters long, the same as an MD5 string. The following is a sample list of the filenames for the domain 2ve3gh53h3yh.ru:

- m99h49wtp1g35b5721d64mfs5p8ese1x.php
- n7co2vpu098x85ctgdn689rf4d18n5jz.php
- fhdkqgyfux4gj2t6zwu434ptw0i0mefu.php
- csbu72ow56i9qq7yg1ufbo3ql1phb1s6.php
- f8t8d5tnqvwwi1l2qf0itr97cdibre6i.php
- hgkvf2riqt49z33isl978pj17aivc0nw.php

The final characteristic of Geost domains is that some of them have a large number of subdomains. For instance, the domain 2ve3gh53h3yh.ru has exactly 1,024 subdomains, such as 0hu, 00n, 03, 06p and 090.

IP addresses

At least 13 IP addresses have been found so far. Table 1 shows a summary of the IP addresses with, for each one, the Autonomous System (AS), country, number of domains related to the IP, and the number of APK hashes that communicate with it. It is worth noting that most IP addresses belong to Mauritius.

IP	AS	Country	WHOIS	Domains	Hash
104.18.61.144	CloudFlare, Inc.	US	Cloudfare	>100	3
104.24.109.180	CloudFlare, Inc.	US	Cloudfare	>100	19
162.222.213.6	QuadraNet Enter	US	USWHSS.COM	14	20
162.222.213.25	Admo.net	US	USWHSS.COM	20	20
162.222.213.29	Admo.net	US	USWHSS.COM	8	20
154.16.244.26	NetStack	MU	Madanambal Annauth	3	0
154.16.244.27	NetStack	MU	Madanambal Annauth	9	2
154.16.244.28	NetStack	MU	Madanambal Annauth	19	12
154.16.244.30	NetStack	MU	Madanambal Annauth	8	0
154.16.244.138	NetStack	MU	Madanambal Annauth	10	0
154.16.244.139	NetStack	MU	Madanambal Annauth	3	0
154.16.244.140	NetStack	MU	Madanambal Annauth	1	0
81.177.6.88	OJSC RTComm	RU	Sergey Ulyashin	5	84

Table 1: Summary of the IP addresses used as C&C.

APK hashes

The Geost botnet is associated with at least 150 APK (*Android* package) files. Most APKs share some similarities with each other: each one mostly communicates with only one domain, and each one accesses one unique random PHP file. Regarding the phone permissions, all of them requested access to read, receive, and send SMS messages, to write on the external storage, to access contacts, and to change Wi-Fi status. For the rest of this section we will refer to the APK binaries with their MD5 hash. The list of SHA256 sashes for the APK files related to this paper can be found in the Appendix.

An example of a Geost APK is the file with MD5 4e1af25f84200c7f63e315fe7ca07a9c, that, according to *VirusTotal*, communicated with the domain w23t2t2tfwg.ru and PHP file q15m9gdhybzfznkgexdld9lk3tigg08w.php.

Another example is the APK 9d8702dafbcad82a4603e1fd2e2869b4, which contacted the domain w23t2t2tfwg.ru and the PHP file pyh32o0ezfguw1xl4382wzm8tnr1tyng.php. The domain w23t2t2tfwg.ru is one of the most commonly used by APK samples in Geost. Table 2 shows the complete list of 25 APK hashes that contacted the domain w23t2t2tfwg.ru together with their detection ratio in *VirusTotal*: the number of anti-virus engines that detected them positively on 23 February 2019 against the total number of anti-virus engines that checked the sample. For space reasons it was not possible to include the complete list of 150 APK hashes for the complete Geost botnet.

An example of an APK resolving several domains is 92a3a69c6c0922ace36ca3ac95fcbbb6, which was first seen in the wild in September 2017. The domains resolved by this sample were: 23r23e23er.xyz, fwefr434r3.xyz, rgrer43e2e.xyz, wef34r34rs.xyz and ge5t5t54trtr.xyz.

SHA256 hash	Detect / Total AVs
1e13f46e3833e0a002c499a611b8f4b57b9716a0686b2a04ee701260c3f729e4	36 / 61
1bc3a740bf994d49301fac2f976a7e6887a2f869a09a66d273538d44b2c990b6	34 / 59
91c032d905a92a3dc69c2ba163dd9978ce843fbb2f434f2254a1b7d69b411aff	32 / 62
4d73fc6eb4099bb4b27225ea6c19f7a1f5d276a540d42d244a1b38566aacdcea	26 / 62
e31986c1309e9aae27fec1d3a279b816f6610e54c06c154589a4f72f694d1161	29 / 60
34a01cedf6b94d4979a81275fa8cd4e99e9691b13339ce8763d2362d7fe8faec	32 / 62
2dc56dc14d8c352813c3c6d7026f830a940a716ab291f90bd9aacdc9a236af69	29 / 63
22bac0179306a5bfd7e1d90d458298f487c67d3f84b2ab9bd6f2e399c86cfdc7	27 / 62
051a942a724fb1c5485f1e14f7899dc237c9bf1d7e4db900b0c03e2e3e42e8eb	32 / 61
298601b71f2c4d5db132ad9d972cdabb61bdebb69980fac411fdd9a6e9275860	32 / 62
8ddd48b104bd8805a1c5c98bc6fb7165924d3b3206ada973297c2b511ed2b555	27 / 62
e26d52647bc345232aa904987dc872ee500a1278fdfd65fcfbae58be774dcc96	23 / 59
c9a64286bc7e921d150a64e678705b4fcb99389eafc658c623455ba498009212	26 / 62
56ed2cbb764748b95d893ba1b1c58d0dd801ef1a98958cd5a36eff0995d90999	35 / 62
6d6d79f259943c02d1f39fa7212e0dc3c95650e5aab516e90d083120cff9ee60	32 / 62
72808f79b8c1b5d26324e7c30a1ae61eba2775dbe68d92fa2c85cab7329b5d04	27 / 62
781f84749667a9cc588b46671077111f5f433c4e3635c8e832ada54ee72a0421	38 / 63
4c41694a957419fe79173f802f3167df865fbbe78d8a2747e15018acfbdfe86e	31 / 61
ccfff0a7d44fa7d0ff81029c3871be118dad82bc7012a4a5162e979798e2a6fc	19 / 63
f9ae476484cf27a2fff5095f9c0a278debd9794aedefe986d912c95fc3e82f26	28 / 62
4f6524c3748369228e381198213b7eab2fcffb29f4b01a0a6b4c3af2e06f5464	28 / 62
7bea49e9e60beb5e7fe95c29d8f11da4a6ea36d7ab8787f442125ef111284811	32 / 60
d305f1f13cdf9bbfb2c1fb16b73771d13a7ca0b6a417e93583ad3d0aa78fac2a	33 / 60
d499e64697b9cf2ba61036acf389939ec91c2c2dae9d3672603fe60c80c85432	28 / 62
78d2ed73571c9f39432143ece31cd92d05b39b7f6590b4841adf33764ac3f816	30 / 62

Table 2: SHA256 hashes of APKs related to the domain w23t2t2tfwg.ru.

Most of the APK binaries of the Geost botnet are identified in *VirusTotal* as 'Android Hqwar' or 'Banking Trojan'. However, both terms are generic and used to identify thousands of binaries that are protected by a software packer or an obfuscating method. Therefore, this particular botnet has not been identified by the community until now. As an example of how each APK was detected by *VirusTotal*, Table 2 shows a subsample of the total number of APKs and their detection ratios.

Relationships in the infrastructure

Among the uncommon characteristics of the Geost infrastructure are: (i) each domain corresponds to a unique IP address; (ii) no domain was ever seen without an IP address registered; (iii) each IP address has more than one domain assigned to it; (iv) domains always refer to the same IP address. It is worth noting that the Geost malware used random generation of words at least in three places: (1) to generate its domain names, (2) to generate the names of PHP files, and (3) to generate the names of the APK packages.

An example relationship between the pieces of the botnet's infrastructure is the sample 92a3a69c6c0922ace36ca3ac95fcbbb6. This sample communicated with the domain wef34r34rs.xyz, which resolved to the IP address 154.16.244.28. This sample targeted three of the top five Russian banks and the name of its package is 'com.vuzbswbpv.ipapszyud'. The same IP address was also assigned to the domain t43r43r43.xyz that is requested by the sample 92a8aa2c6dd86aeab67e687de2c9e6a9591bee17.

6. VICTIMS

The traffic generated by the botmasters when accessing the C&C server revealed information about the victims of this botnet. It seems that the botmasters kept a detailed summary of the victims, and that this summary was important for the operation of the botnet. The victims of this botnet not only probably lose money but they had their privacy and identity completely compromised. The *minimum* amount of information that the botmasters know about each victim can be seen in the following list:

- IMEI of the phone
- Brand of the phone
- Phone service provider
- Phone number
- Country of the phone number
- Current balance of bank accounts
- History of balance in each bank account (the history of the balance is not even available to the victims themselves)
- Whether they have a credit card tied to the phone
- From the SMS of the victims:
  - Name of victim
  - Home address
  - Social relationships

- Religion
- Purchases
- Expenses
- Financial problems.

Regarding the number of victims, it is only possible to speculate. In the C&C server of the IP address 162.222.213.28 there were 50 victims per page, and there were 1,452 pages, which gives an estimation of 72,600 victims in that C&C alone. Extrapolating this to the 13 C&C servers, a rough estimation of the total number of victims may be 871,200. It is possible that even more victims exist, given that there may be more C&C servers.

According to the 50 victims shown in one of the C&C screens, there is a column labelled 'Balance' that shows the amount of money (in Rubles) in the bank accounts of the victims. The total sum of this column of 50 victims is 1,129,152 Rubles, which is approximately 15,000 Euros. Extrapolating this number to the estimated 800,000 victims in this C&C there may be an estimated maximum total amount of money close to 240,000,000 Euros. However, the real total for this C&C could be much lower if we consider that the web page is sorted by balance.

IMEI

Of all the information stolen from the victims, the IMEI is important because it can be used to identify them. The IMEI is a unique code assigned to cell phones and, by searching for it online, it is possible to find out information about the device. The IMEI number is divided into parts. The initial eight-digit portion of the IMEI, known as the Type Allocation Code (TAC), details the phone model and origin. The remainder of the IMEI is manufacturer-defined, with a Luhn check digit at the end. Given the IMEI, it is possible to determine the victim's phone model and characteristics. From the IMEI it was possible to learn the brands of the phones of the victims, which were all *Android*-based. From the IMEI numbers it was also possible to identify the victims' phone operators, including *Tele2*, *MTS RUS*, *Beeline*, *MegaFon*, *Yota* and *Motiv*. The last one is a Russian regional provider.

SMS data

The access to SMS messages was probably one of the more invasive actions of the botnet. SMS messages potentially contain a lot of private information about the user. An analysis of the two SMS lists downloaded revealed that users shared very private conversations with friends and lovers, the status of their financial accounts, and sensitive private data about themselves. It was particularly interesting to find that most of the private information was leaked by the phone operators, including users' real name, birthday, the last four numbers of their credit card, the amount of money in their balance, and the password for mobile banking applications. The following is an example SMS stolen by the attackers (without personal information):

```
07/03/18 18:59 VISA5880 purchase 120r
MTS TOPUP 5635 Balance: 49746.86r
```

7. ATTACKERS

One of the most important breakthroughs of this analysis was the discovery of a file in a public web page that referenced one of the Geost domains. This file proved to be the chat log of a group of people related to the Geost botnet operation. It is not clear how the file was leaked, but since it was a *Skype* chat log it was probably created (whether on purpose or not) by one of the participants in the chat. The use of *Skype* as a communication medium is consistent with previous reports on the modus operandi of the Russian malware community [14]. The existence of this file marks another OpSec error on the part of the botmasters: they trusted part of the operation to a group of users with very low or non-existent OpSec practices.

It was possible, then, to conduct an open-source intelligence (OSINT) investigation to find out more about the group in this chat log. The file has more than 6,200 lines, covering eight months of chats, and shows the private conversations of 29 people. Not all of them seem to be related to the Geost botnet since the group had several alternative streams of revenue. By analysing the top participants in the chat log it was possible to determine that the user 'powerfaer' was the only one talking with all the participants, making this user the probable owner of the chat log.

During the time period from 2017-06-11 11:14 to 2018-04-17 18:41, powerfaer held business discussions with the other 28 people in relation to different projects. The conversations between powerfaer and the user with the nickname 'mirrexx777' seem to be the most notable since they showed a connection with the Geost botnet. For instance, on several occasions powerfaer and mirrexx777 exchanged links to the control panel of the Geost botnet, sharing information that nobody would possess unless they were insiders. The following is a human translation from Russian:

```
On 2017-10-18 07:24:07
From powerfaer to mirrexx777:
   http://2[redacted]e.xyz/stats.php?sid=
   7NDNI0aercTtwPA
   title:Statistics

   Re-crypt, Kaspersky got cleaned

From mirrexx777 to powerfaer:
   ok. will do. according to the old
   recordings how many of them remains?
   i want to start to keep a record
```

The fact that they shared information from inside the C&C channels – information that you need to be logged in to see (the stats.php file) – and the fact that they discuss the need to fix them, is strong evidence that they possess internal information with complete knowledge of its purpose. There were many pieces of evidence in the chat log showing a relationship with malware actions, such as asking to re-encrypt links because *Kaspersky* was able to detect them.

It seems that the user powerfaer has operated since 2010. This is supported by one conversation where there was a remark about the income from traffic in 2010 having been better (translated from Russian):

```
On 2017-12-06 18:14:46
From powerfaer to mirrexx777:
 That would be nice to get back in to 2012
 Or 2010
```

Some conversations in the chat got serious and resulted in the use of real names as a means to call the attention of the other. This confirmed the names of some aliases. The following log confirmed the name of 'taganchik.ru' when powerfaer talked to him (translated from Russian):

```
Alexander, really, if we started together we need to finish it. Because for now
this is working and we can earn money. Not every day we are getting 100k for
promotion
```

Later on, however, it seems that the user taganchik.ru tried to leave the group:

```
2017-10-15l14:53
From taganchik.ru to powerfaer:
 (...) But now im saying i am working but
 in fact I dont. I am getting demotivated
 and do not want to do anything

From taganchik.ru to powerfaer:
 i thought about it, and im not in

From powerfaer to taganchik.ru:
 Understand, ok. Shame. If you change
 your mind write to me
```

Showing a complete lack of OpSec, the chat log also revealed credentials for several servers and services, such as fttkit.com (an *Android* application protection service advertised on the Russian underground site crimina.la). The log also disclosed the IDs of online wallets, and credit card numbers. This information helped us find sensitive information about the identity of some individuals. For instance, 'taganchik.ru', 'elkol95' and 'dmitrixxx89' all advertise their services on the same web marketing forum, https://searchengines.guru/.

The user powerfaer also engaged in conversations with several money launderers. The log confirms that online payment systems such as *WebMoney*, *Qiwi*, and *Yandex Money* remain popular among Russian cybercriminals [15]. However, these services are not anonymous and it would be possible to see the payments through third-party money launderers. The following is an example chat with the user 'cyberhosting.ru':

```
On 2017-12-04 11:21
powerfaer wrote to cyberhosting.ru
   And another question,
   can you exchange cash to BTC?
```

A challenge for us during the analysis was to understand the Russian underground slang. For example, the term *white accounting* should be translated to Russian as `Белая бухгалтерия`. However, cybercriminals used the term `белка`, which in English means *squirrel*. The same issue applies to other words like *application*, which translates to `прила` in Russian and has no direct translation in English.

After a deep OSINT analysis it was possible to infer a list of probable real names for the following nicknames: 'mirrexx777', 'powerfaer', 'cyberhosting.ru', 'taganchik.ru', 'doktorsaitov', 'dmitrixxx89', and 'maximchik700'. However, the names will not be published since their implication in the Geost operation has not been confirmed.

8. CONCLUSION AND FUTURE WORK

The discovery of the Geost *Android* banking botnet inside the traffic of another malware proxy shows that operational security is very hard to get right, and that simple mistakes can lead to deep understanding of the operations of malware authors. After the discovery of the Geost botmasters accessing their C&C servers it was possible to find more and more pieces of their botnet infections, leading to a very large mapping of their attack infrastructure, their APK binaries, the number of victims infected, and an estimation of the economic size of the operation. Finally, it was possible to use open-source intelligence to relate a group of developers to part of the infrastructure-building process of the botnet. The developers do not seem to be the Geost botmasters, but an underground group related to them.

Despite operating since at least 2016, the Geost botnet remained unknown until its traffic was captured on the HtBot malware. This may suggest that the best OpSec may be to hide operations among thousands of other malware. However, once the operation was found, it was clear that the group's OpSec measures were not good since there were several mistakes that have led to information about the operation.

The following is a summary of the operational security mistakes that led to the identification and understanding of the botnet:

- Use of the illegal proxy network HtBot. Wrong estimation of the risk of using a service that was being tracked in a security laboratory.
- Failure to encrypt C&C traffic. It was possible to identify the traffic and the content of the communications.
- Use of the same protection service multiple times. This allowed repeated monitoring of the attackers and the capture of credentials.
- The hiring of a group of developers with very low OpSec, who disclosed links, names and credentials in their chats.
- Failure to encrypting chats. This allowed a document to be leaked containing important information about the privacy of some attackers and leads about their identities.

The amount of information collected on the Geost botnet was so large that it has not been possible to include all the details of the infrastructure, the victims found, banks accounts disclosed, phones infected, credit cards used, and the very interesting view of the social relationships within a group of underground cybercriminals. Therefore, our analysis of the Geost botnet will continue in several directions. The name 'Geost' was selected after the only web page that didn't seem to change in the C&C servers.

ACKNOWLEDGEMENTS

We would like to thank Veronica Valeros for her help during the analysis and extraction of information. We also thank Professor Sebastian Garcia.

REFERENCES:

[1] Zhang, E. What is Operational Security? The Five-Step Process, Best Practices, and More. Digital Guardian. 2018. https://digitalguardian.com/blog/what-operational-security-five-step-process-best-practices-and-more.

[2] Ilascu, I. Flaw in Telegram Reveals Awful OpSec from Malware Author. Bleeping Computer. 2017. https://www.bleepingcomputer.com/news/security/flaw-in-telegram-reveals-awful-opsec-from-malware-author/.

[3] Newman, L.H. Yes, even elite hackers make dumb mistakes. Wired. 2018. https://www.wired.com/story/guccifer-elite-hackers-mistakes/.

[4] Paul, K. How Silk Road's Founder Could Have Avoided Getting Busted. Vice. 2015. https://motherboard.vice.com/en_us/article/ezvkg7/the-five-hidden-service-commandments.

[5] Otten, B. Cybercriminal intent: When good OpSec met bad OpSec. Tech Beacon. 2016. https://techbeacon.com/security/cybercriminal-intent-when-good-opsec-met-bad-opsec.

[6] Virqdroid. Mobile Threats targeting Russian Banks. https://www.virqdroid.com/?m=1.

[7] Wei, F.; Li, Y.; Roy, S.; Ou, X.; Zhou, W. Deep ground truth analysis of current android malware. Lecture Notes in Computer Science, vol. 10327 LNCS, pp.252–276, 2017.

[8] Štefanko, L. Android banking malware: sophisticated trojans vs. fake banking apps. ESET. 2019. https://www.welivesecurity.com/wp-content/uploads/2019/02/ESET_Android_Banking_Malware.pdf.

[9] Neto, P.D. The new era of Android banking botnets. https://www.botconf.eu/wp-content/uploads/2017/12/2017-Drimel-The_new_era_of_Android_Banking_Botnets.pdf.

[10] Shishkova, T. The rise of mobile banker Asacub. Kaspersky. 2018. https://securelist.com/the-rise-of-mobile-banker-asacub/87591/.

[11] White, J. ProxyBack Malware Turns User Systems Into Proxies Without Consent. Palo Alto Networks. 2015. https://unit42.paloaltonetworks.com/proxyback-malware-turns-user-systems-into-proxies-without-consent/.

[12] McCoy, D.; Bauer, K.; Grunwald, D.; Kohno, T.; Sicker, D. Shining light in dark places: Understanding the tor network. In Privacy Enhancing Technologies, N. Borisov and I. Goldberg, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp.63–76.

[13] Chakravarty, S.; Portokalidis, G.; Polychronakis, M.; Keromytis, A.D. Detecting traffic snooping in tor using decoys. Lecture Notes in Computer Science, vol. 6961 LNCS, pp.222–241, 2011.

[14] Terrelonge III, L. Cybercrime Economy. An Analysis of Criminal Communications Strategies. Flashpoint 2017. https://forensicfocus.files.wordpress.com/2017/05/flashpoint_cybercrime_economy.pdf.

[15] Goncharov, M. Russian Underground Revisited. Trend Micro. 2014. http://www.trendmicro.com/vinfo/us/security/special-report/cybercriminal-underground-economy-series.

APPENDIX: SHA256 HASHES OF ANDROID APKS FILES RELATED TO GEOST BOTNET

70e6454910b1c4e1ff1a86a6e7506e6e5c234fca2fe77e44a00287aacc86853e
0bf2fc434ae4ab98e0a25388042ae011048d54404e0b94bd513bd6927d9f918a
934ae455b77216544358061091 6b3af352c3c46a83cb17cb7f380d6835d84552
b9862f5f097e2c05577b602022ffd7429af448b5ff485bfa8f3d8919d819eec5
299c3916838e527986c5d252322386add8c320a5da2138986a59e2b667a00945
3d32fb91da5ed45ecc8e7880b85e817e05d2134f5ecd69f5b4478be8013ae2da

5627c1d1ea942bab7134396dd7ba89009e6ff921c1e1a608a6dcdbdda2b14744
2a307a34de0b9d33bfc225e60c393c380b981a9fc52ce1277fc30445237f151d
6a7782b019566becbe0a7c06e56abbe54e3d72726f26b1bf95499b21b076d39e
0367d4e913b28fad8c57a37ac21cac5cda347846bb2b0f5d505fa47696ba2f2a
7d49950323cf0eae8b5ae36e4aefc688a1bfa1a651457382e9f9a4a4e28073c1
302c2d88fba26235b3229dd1b146a767449d47ede008556ef0d79a3c7b44d382
6e6dd2329188b334e519845804bef6e52454620dfb37ae46a457a81c478d2f77
dddfcd90fbb5b02756ec03ea75d2d98b6d1f29e14fbdbebfe6e2c77026591056
7659e30f3d8d45d7c595cb03ffe6ad6706b9c4b17d8c284a0fa6c90e226f44e6
f265608593e47c25a6bbdf31179776b401e08f08c4930dcac50684be70aa8902
4748c004a3e4b35b0daddd054e22c393c7c66aaa1d08ee3cba7c3bddc26b0a6f
4727b7727ee4ae5d9f041dc7f066da70b8cfb7417d0904e34b7b4028c38f2c76
8d1cd474f4aefcaf5f2fd6ce890ca49398194c796631b73c090fbcce2ed4f2dc
c63e7ccf63feeaf145c0303bd91bf46f43a4b2170cba0b9939492eae88b0175a
b1a376b1427a0373915f228d51eb26ea6cd009b4dd11796902f3fee6f8af122e
18ab096f1d2cd8a2759204838114e5ab4ff82f07adc8efce393cf5a807790e4d
04957fe15f8d9df2bf03f6660a55dbf57570416cdb4c225203b99a4e5c7d632b
de963c011fad513f8ced3e2911b02bfe514ca8991be31b4338262e76939a5dfa
f446e1c58cd7d8ebbfdfeaa2ae1eabf361e75ecd92dd5b9d9c09fa085949baf7
c92c09e4aaf9c3f9531a92964077d6fa6b118f87f106ee1b7f430a43c783a7f6
28c864aa54ab9c4f2b254258f3db807638becdacd11d23f793978f03863f065d
931d011f1343979f233ec9767005a492e76c5434cf4fd863c9969e8b461c04dd
cccb82d3b9f98b34678333c7f4e3e9fcf00cc2515a2c731965074af2c9f85f00
a70210a109aa4bd9eec9f495378027e9aadd83dc65d5344e26739e98b2e3aa7d
13776897f46add32b1dda3f7862c53bb069ce839334f9b1d7cd7e93cc4b9a3b6
ba3ecf85544e09d4e31b912b19d47728767933ccdc4e1b7c337a7a18ade7aa7d
77d88c936db100e77290abc4131cf41fdc092f77c8fcb488dfc1d08a3937b94c
8c3ac248e798e6f1fb5e349cc558f0b62ed9a23393b4bf11117c1d9de19e57a6
3fcec3bda7d044848a3aaf5f893a319982b545a7736adde036eb47c3bb4ea0d5
29030672718236978 76b4c153e0bbc222cb8fdbd1b936fb8cfd5f35ae8401dfa
50c82f9ed9e91a1e10997cc707aec1587c8488c35e7dc76ac3d3d25eb60753b4
bd9ef6aa820164ea76def200f47abad38edbb4a1df13aa602ee8673af85f6aea
00a5f79d610759c6dd88e1c6108be24daad5b18187f0abde7bd9056e0d513ee2
9ff5dc79a6d7d1369ee113b0250a75a5ce3ce9caeb66fc46f602564086c525b5
45c7feeca4784dd6c5bc91d4e02a81d36f9ee56a954730ccc66c7e36671f1c3c
9706ca42aa8fef8a8c9463d647e5ecf7671180024e78988c4e5a36c1d86e0615
d36b04ae800000300c351cee1ee0f708340f9cb5b5da5a9a97799e8368a6a3c4
513c649370052ee0934175854037eac7c2cf5eb147414fa61df42b35530babaf
8fb1f54434f2966751d7ae221466c50e5deb5f51ed6e2a042fd71e3d2a53cf5b
e2e8a472b3bdf1ba785d5e78bb12ecb31f14bfcb43d4d0043b6116fd197f6e33
4f0e801a6d0f4898b0874da31d63d2dda0620e347d72b35f5086fb22cde9a9cd
5f216ae10a3972b5a90d6178f4d6f0d2c995b4248a9f329edbc854ead89ce904
2ba2a567c91086112c63f09ace11d725537dceba1cc56c14fc86d63d1c6585c8
e8bf2615d8d9c3d768f687cd05d0f9305fd3118168d2b94eabdfc365fafc9d06