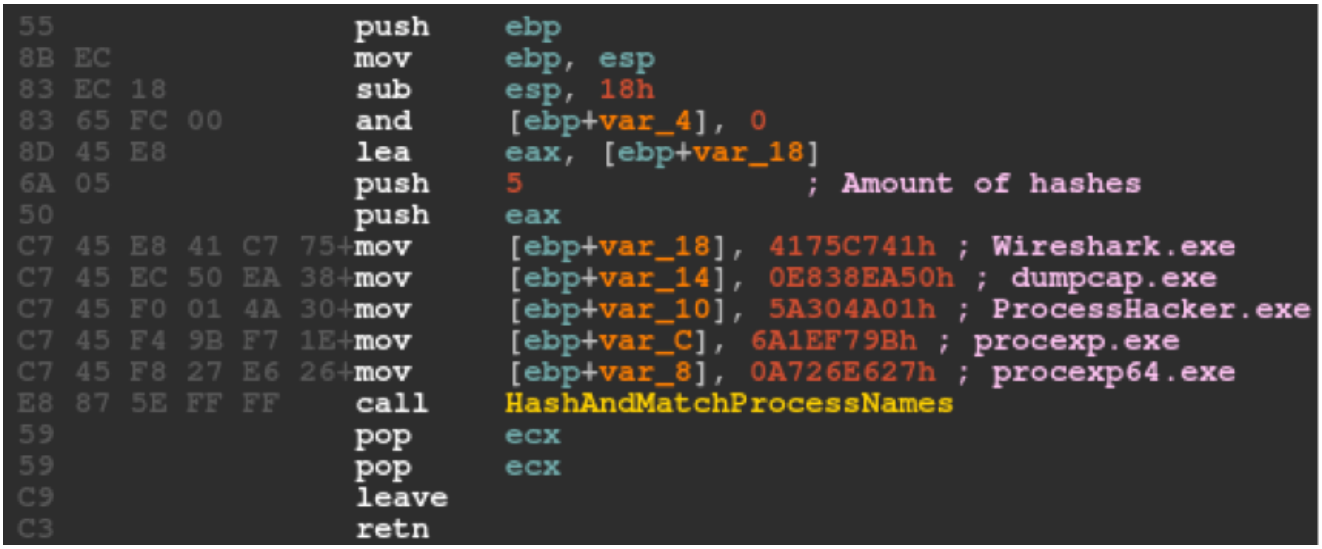
Grabbot is Back to Nab Your Data

blog.fortinet.com/2017/03/17/grabbot-is-back-to-nab-your-data

March 17, 2017



```
55                  push    ebp
8B EC               mov     ebp, esp
83 EC 18            sub     esp, 18h
83 65 FC 00         and     [ebp+var_4], 0
8D 45 E8            lea     eax, [ebp+var_18]
6A 05               push    5               ; Amount of hashes
50                  push    eax
C7 45 E8 41 C7 75+  mov     [ebp+var_18], 4175C741h ; Wireshark.exe
C7 45 EC 50 EA 38+  mov     [ebp+var_14], 0E838EA50h ; dumpcap.exe
C7 45 F0 01 4A 30+  mov     [ebp+var_10], 5A304A01h ; ProcessHacker.exe
C7 45 F4 9B F7 1E+  mov     [ebp+var_C], 6A1EF79Bh ; procexp.exe
C7 45 F8 27 E6 26+  mov     [ebp+var_8], 0A726E627h ; procexp64.exe
E8 87 5E FF FF      call    HashAndMatchProcessNames
59                  pop     ecx
59                  pop     ecx
C9                  leave
C3                  retn
```

Introduction

Fortinet has discovered a new botnet capable of stealing large amounts of user information, as well as remotely manipulating compromised machines. The malware appears to be based on an older botnet known as Grabbot, which was first discovered back in November of 2014. This new variant improves on that existing functionality while adding several dangerous new features. This blog aims to offer a quick insight into how Grabbot functions.

Replication

The bot can be found hosted on a number of compromised websites with a random filename. We currently suspect that Grabbot may arrive on these hosts through Exploit Kits or other malicious campaigns.

The bot may drop several files in the following paths:

●"%AppData%\{GUID}\{generated filename}.exe"

●"%AppData%\{GUID}\{generated filename}.bat"

●"%AppData%\{GUID}\{generated filename}"

Note that each generated filename is different, with the host machine's System Volume Information. Several mutexes are created in the same way. Each drop file also has its file time information set to be the same as "cmd.exe" in Windows.

The malware creates the following registry entry to survive system reboots:

●HKEY_CURRENT_USER\Software\Microsoft\ Windows\CurrentVersion\Run

○{GUID} = "%AppData%\{GUID}\{generated filename}.exe"

During execution, the bot may inject the main payload into explorer.exe and delete the original file.

Browser Targeting

The bot enters a sleep loop and will not perform the rest of its functionality unless one of the following internet browsers is found in the active process list:

●Internet Explorer (iexplore.exe)

●Firefox (firefox.exe)

●Google Chrome (chrome.exe)

●Opera (opera.exe)

Anti-analysis measures

The bot also scans active processes for the presence of certain system analysis tools, such as Wireshark or Process Explorer. If any is found, the bot may branch into a fake set of behaviours instead of the actual payload.

```
55                  push    ebp
8B EC               mov     ebp, esp
83 EC 18            sub     esp, 18h
83 65 FC 00         and     [ebp+var_4], 0
8D 45 E8            lea     eax, [ebp+var_18]
6A 05               push    5                   ; Amount of hashes
50                  push    eax
C7 45 E8 41 C7 75+  mov     [ebp+var_18], 4175C741h ; Wireshark.exe
C7 45 EC 50 EA 38+  mov     [ebp+var_14], 0E838EA50h ; dumpcap.exe
C7 45 F0 01 4A 30+  mov     [ebp+var_10], 5A304A01h ; ProcessHacker.exe
C7 45 F4 9B F7 1E+  mov     [ebp+var_C], 6A1EF79Bh ; procexp.exe
C7 45 F8 27 E6 26+  mov     [ebp+var_8], 0A726E627h ; procexp64.exe
E8 87 5E FF FF      call    HashAndMatchProcessNames
59                  pop     ecx
59                  pop     ecx
C9                  leave
C3                  retn
```

Fig.1: Searching for hashes of specific process names

```
DNS      113 Standard query 0x36de A upg26d02xxedojwta6mc4gpw7oerdhdt93l1wb6tfuckgjzmts.ru
DNS      174 Standard query response 0x36de No such name A upg26d02xxedojwta6mc4gpw7oerdhdt93l1wb6tfuckgjzmts.ru SOA a.dns.ripn.net
DNS      113 Standard query 0x3da6 A qt4km4pt4ltddnubetunbfp8n4wcor4vc5ys8oexdeaeu6ktys.ru
DNS      174 Standard query response 0x3da6 No such name A qt4km4pt4ltddnubetunbfp8n4wcor4vc5ys8oexdeaeu6ktys.ru SOA a.dns.ripn.net
DNS      113 Standard query 0x880c A hjctholhyfk1q9iv8dix9g2xnv4o3r9lkdiy9tr0abyagxa307.ru
DNS      174 Standard query response 0x880c No such name A hjctholhyfk1q9iv8dix9g2xnv4o3r9lkdiy9tr0abyagxa307.ru SOA a.dns.ripn.net
DNS      113 Standard query 0x933a A jqh8zzxz6bge7yduezhxajjr7ivgj6673knhjs1ha7fsx3bqo3.su
DNS      174 Standard query response 0x933a No such name A jqh8zzxz6bge7yduezhxajjr7ivgj6673knhjs1ha7fsx3bqo3.su SOA a.dns.ripn.net
DNS      114 Standard query 0x69e8 A u7dk6ojeduperiuzco7ic3ahiie0703x0i3kk3jcubn9fl361e.com
DNS      187 Standard query response 0x69e8 No such name A u7dk6ojeduperiuzco7ic3ahiie0703x0i3kk3jcubn9fl361e.com SOA a.gtld-servers.net
DNS      114 Standard query 0xb619 A zcjill27gt6br1nkpi41vxmxmlq7kw7l6ieyj3yrfjxzfi6r62.com
DNS      187 Standard query response 0xb619 No such name A zcjill27gt6br1nkpi41vxmxmlq7kw7l6ieyj3yrfjxzfi6r62.com SOA a.gtld-servers.net
DNS      113 Standard query 0xcc80 A ajtql6w0flk3u9rosr0nbv2njh3p9xh9idvh70weciccj6b2o3.su
DNS      174 Standard query response 0xcc80 No such name A ajtql6w0flk3u9rosr0nbv2njh3p9xh9idvh70weciccj6b2o3.su SOA a.dns.ripn.net
DNS      113 Standard query 0x3648 A frcim1xtbjb5jtfwt7smtlevp0a52uuvzizgl2h7pukzynniq0.su
DNS      174 Standard query response 0x3648 No such name A frcim1xtbjb5jtfwt7smtlevp0a52uuvzizgl2h7pukzynniq0.su SOA a.dns.ripn.net
DNS      113 Standard query 0x0f38 A b3nwbb4isx65g12bauprt1hglhs71d4zbnba62banuge2p9o6v.ru
DNS      174 Standard query response 0x0f38 No such name A b3nwbb4isx65g12bauprt1hglhs71d4zbnba62banuge2p9o6v.ru SOA a.dns.ripn.net
DNS      113 Standard query 0x4a4b A chpbtubypq2qjombrk5h29vlo4n3ayebzj3pzcqfxcmx6yjari.su
DNS      174 Standard query response 0x4a4b No such name A chpbtubypq2qjombrk5h29vlo4n3ayebzj3pzcqfxcmx6yjari.su SOA a.dns.ripn.net
DNS      114 Standard query 0x375d A hghg1vyhujy3qwncyrckclj7ezj9vdobb9bmmq1xsc2avro9kn.com
DNS      187 Standard query response 0x375d No such name A hghg1vyhujy3qwncyrckclj7ezj9vdobb9bmmq1xsc2avro9kn.com SOA a.gtld-servers.net
DNS      113 Standard query 0x9dea A w7w4w77w212ve80xbkxhn888oigaoy3t5j2fq2ravg5ti35gm9.su
DNS      174 Standard query response 0x9dea No such name A w7w4w77w212ve80xbkxhn888oigaoy3t5j2fq2ravg5ti35gm9.su SOA a.dns.ripn.net
```

Fig. 2: Part of the fake behaviour - Random domain name generation and contact

C&C Connection

Before the bot attempts to contact the command and control (C&C) server, it first makes a connection to *www.microsoft.com* to verify internet connectivity. If a connection can be established, the bot will iterate through a list of possible C&C servers and contact each until a response is received. The list of C&Cs observed in this sample are:

●http://de{REMOVED}is.site

●http://ge{REMOVED}et.site

●http://bi{REMOVED}ys.info

●http://on{REMOVED}nc.site

●http://de{REMOVED}is.info

●http://ss{REMOVED}rs.info

When a connection is established, the bot may attempt to download the following data files:

●/wordpress/ajax/d.dat

●/wordpress/ajax/e.dat

●/wordpress/ajax/f.dat

●/wordpress/ajax/out.dat

●/wordpress/ajax/g.dat

●/wordpress/ajax/h.dat

The files are saved on the disk with a generated filename. Notably, the file “out.dat” is renamed to the executable file in the autorun registry. All communication between the bot and the C&C are encrypted and done through HTTP. In any contact with a C&C, the bot will try twice to establish connection before trying a different C&C.

```
HTTP      756 POST /wordpress/forumpost.php HTTP/1.1
HTTP     1422 HTTP/1.1 200 OK  (text/html)
HTTP      357 GET /wordpress/ajax/d.dat HTTP/1.1
HTTP      975 HTTP/1.1 200 OK  (application/x-ns-proxy-autoconfig)
HTTP      357 GET /wordpress/ajax/e.dat HTTP/1.1
HTTP      503 HTTP/1.1 200 OK  (application/x-ns-proxy-autoconfig)
HTTP      357 GET /wordpress/ajax/f.dat HTTP/1.1
HTTP     1001 HTTP/1.1 200 OK  (application/x-ns-proxy-autoconfig)
HTTP      359 GET /wordpress/ajax/out.dat HTTP/1.1
HTTP     1061 HTTP/1.1 200 OK  (application/x-ns-proxy-autoconfig)
HTTP      357 GET /wordpress/ajax/g.dat HTTP/1.1
HTTP      546 HTTP/1.1 404 Not Found  (text/html)
HTTP      357 GET /wordpress/ajax/g.dat HTTP/1.1
HTTP      546 HTTP/1.1 404 Not Found  (text/html)
HTTP      357 GET /wordpress/ajax/h.dat HTTP/1.1
HTTP      546 HTTP/1.1 404 Not Found  (text/html)
HTTP      357 GET /wordpress/ajax/h.dat HTTP/1.1
HTTP      546 HTTP/1.1 404 Not Found  (text/html)
```

Fig.3: C&C communication

C&C Commands

The botnet is capable of responding to the following commands:

Command	Function
user_execute	Executes a file on the host machine
bot_update	Updates the bot itself
conf_update	Updates the bot configuration
bot_uninstall	Removes the bot from the host machine

conf_update2	Updates the bot configuration
send_debug	Send system information and log data to C2
socks_bc	Establish SOCKS proxy
run_vnc	Creates VNC connection
install_bd1	Install Teamviewer backdoor
url_block_add	Blocks specific URLs from being accessed
url_block_rem	Removes URL blocking
grab_ftp	Retrieves FTP information from host
grab_cookies	Retrieves browser cookies from host
grab_sol	Retrieves local shared objects/flash cookies
grab_certs	Retrieves client certificates
grab_all	Retrieves data from all grab functions
del_cookies	Deletes browser cookies
grab_pop	Retrieves Outlook POP accounts
run_plugin_exe	Downloads and injects an executable into svchost.exe and runs it
run_plugin_dll	Downloads and injects a library into svchost.exe and runs it

Compared to the previous known version of Grabbot, there are several new commands labeled “conf_update2”, “install_bd1”, “grab_pop”, “run_plugin_exe” and “run_plugin_dll”.

Sending Back Debug Information

The bot is able to extract current system information, including a list of active processes, detected AV products, and a list of installed applications. The bot may send this information to the C&C on command.

Fig.4: System debug information

Banking Backdoor

The bot is also capable of tracking if specific sites, namely financial institutions and services, are accessed, and may launch a proxy or remote access backdoor to steal information. Some targeted sites from the list are as follows (in the format of *[URL]*;[backdoor cmd] [arguments]):

●*paypal.com*;socks_bc 5.{REMOVED}.250:7777

●*hxxps://www1.royalbank.com/cgi-bin/rbaccess/*;run_vnc

●*hxxps://easyweb.td.com/*;run_vnc

●*hxxps://www1.bmo.com/onlinebanking/cgi-bin/netbnx/NBmain?product=5*;run_vnc

Crypto-Currency Wallet Stealing

The bot recursively scans the %AppData% directory looking for files with the name “wallet.dat”, “electrum.dat” or “wallet”. If any match is found, the contents of the file are read and encrypted, then stored into a temporary file for retrieval.

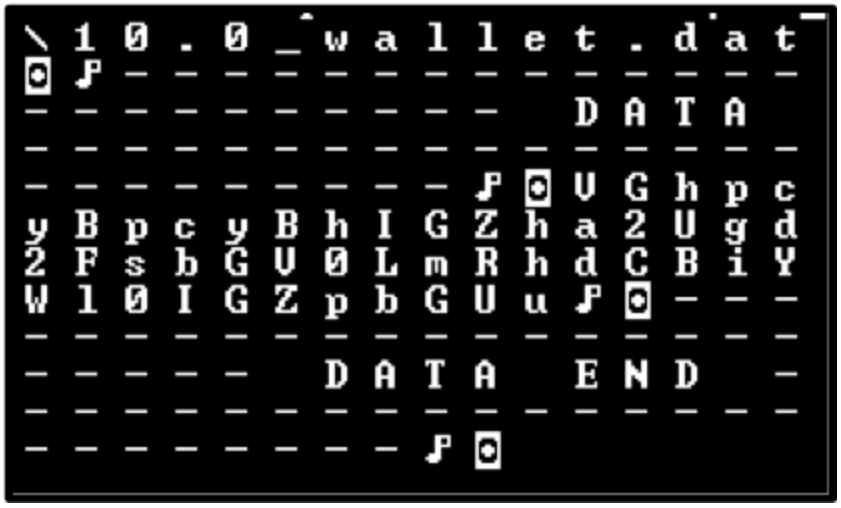


Fig.5: Wallet data to be retrieved

Conclusion

Grabbot was a relatively unknown bot in the past, but from our brief analysis of this new variant it is apparent that Grabbot now has the potential to be very dangerous. Although we are still investigating its current distribution method, Fortinet is able to detect this new variant and we will keep you updated on any further changes.

Sample MD5: d439c468d59f117c584bda463b03aea9

Sample SHA256: 6d8ce2d1b33ff42ba04ded09fe79cff158e6dfffa82f6ceada12f4fda6d0c221

Fortinet Detection Name: W32/Kryptik.VVV!tr

Sign up for weekly Fortinet FortiGuard Labs Threat Intelligence Briefs and stay on top of the newest emerging threats.