Endpoint Protection

symantec.com/connect/blogs/brief-look-zeuszbot-20

Zeus/Zbot is one of the most widely known Internet threats today. It's been around since 2007 and has evolved over time, and is still in a constant state of being developed into a stronger, more prolific Trojan.

A few weeks ago we came across a variant of Zbot representing the fact that it has undergone code refactoring and some functional changes in the Trojan's infection technique and behavior. The variant is now known as version 2.0 (named after the Trojan builder kit version).

In overview, for the common PC user, new changes mean that:

- Your PC could have multiple infections of Zbot, thereby sending your personal information to multiple Zbot controllers.
- Zbot is aiming for information from different browsers, including Firefox.
- Zbot is expanding its ability to run in newer operating systems such as Windows 7.
- Zbot is in constant development, so it might be around for few more years to come.
- Removing Zbot manually isn't going to be straightforward. The same applies for other threats that claim to have a 'Kill Zeus' module.

Let's briefly look at some of the changes in 2.0. For the uninitiated, please read our previously published paper on Zeus entitled "Zeus: King of the Bots."

On your "command," Master

> The built-in commands supported by the Trojan have changed; below is the comparison table:

Version 2.0 command	Functionality	Version 1.x command
user_flashplayer_get	**Gets the Flash player data from a user's system**	**--**
user_ftpclients_get	**Steals passwords from FlashFXP, total_commander, ws_ftp, fileZilla, FAR2, winscp, ftp_commander, coreFTP, and smartftp**	**Resetgrab**

user_homepage_set	Set the browser's home page to desired URL	Sethomepage
user_url_unblock	Restore access to an attacker-desired URL	Unblock_url
user_url_block	Disable access to an attacker-desired URL	Block_url
user_certs_remove	Removes certificate	--
user_certs_get	Steal digital certificates	Getcerts
user_cookies_remove	Delete browser cookies	Delmff (partly)
user_cookies_get	Upload cookies	Getmff (partly)
user_execute	Download and execute a file	Consolidation of 'Rexec', 'Lexec' & 'Lexeci'
user_logoff	Log off user	--
bot_bc_remove	Removes a back door connection	Bc_del
bot_bc_add	Initiate back door by back-connecting to a server and allow arbitrary command execution via the command shell	Bc_add
bot_update	Download and update bot config (sets in registry as well), download and execute new bot installer	Consolidation of 'Upcfg' & 'Rename_bot'
bot_uninstall	Remove bot altogether	--
os_reboot	Reboot the computers	Reboot
os_shutdown	Shut down the computer	Shutdown
user_destroy	(not implemented) possibly to delete all user files	Kos
fs_search_remove	(not implemented) possibly to remove file mask for local search	Delsf
fs_search_add	(not implemented) possibly to add a file mask for a local search	Addsf
fs_path_get	(not implemented) possibly to upload a file or folder	Getfile

bot_httpinject_disable	**(not implemented) possibly to disable http inject**	**Block_fake**
bot_httpinject_enable	**(not implemented) possibly to enable http inject**	**Unlock_url**

Built-in commands in 2.0 compared to 1.x

Some of the commands in 2.0 aren't implemented yet, which only leads us to one conclusion: it is under continuous development.

Sneak in

When executed, initial thread injection into other processes is as follows:

The processes "explorer.exe", "taskeng.exe", and "taskhost.exe" are injected with the below four threads:

1. A thread to open a back door channel.
2. A thread to download threat-related data.
3. A thread to inject into other running processes.
4. A thread to perform bot instructions from config stored in the registry.

A single thread is injected into the processes "rdpclip.exe", "ctfmon.exe", and "wscntfy.exe" (the single thread's job is to inject into every other process).

The "Hijack"

The APIs shown in the table below are hijacked from the injected process memory. We can see that version 2.0 is now able to hook NSPR APIs (used by Mozilla Firefox) to read and write to data that it receives and sends:

Zbot 1.x hooked APIs	Zbot 2.0 hooked APIs

WININET.DLL
- HttpSendRequestW
- HttpSendRequestA
- HttpSendRequestExW
- HttpSendRequestExA
- InternetReadFile
- InternetReadFileExA
- InternetQueryDataAvailable
- InternetCloseHandle
- HttpQueryInfoA

WS2_32.DLL
- closesocket
- send
- WSASend

USER32.DLL
- TranslateMessage
- GetClipboardData

NSPR4.DLL
- PR_OpenTCPSocket
- PR_Close
- PR_Read
- PR_Write

CRYPT32.DLL
PFXImportCertStore

NTDLL.DLL
- LdrLoadDll
- NtCreateThread

KERNEL32.DLL
GetFileAttributesEx

WININET.DLL

- HttpSendRequestW
- HttpSendRequestA
- HttpSendRequestExW
- HttpSendRequestExA
- InternetReadFile
- InternetReadFileExW
- InternetReadFileExA
- InternetQueryDataAvailable
- InternetCloseHandle

WS2_32.DLL and WSOCK32.DLL

- send
- sendto
- closesocket
- WSASend
- WSASendTo

USER32.DLL

- GetMessageW
- GetMessageA
- PeekMessageW
- PeekMessageA
- GetClipboardData

Here is an image showing hooked APIs in process memory:

“Drop” zone

Zbot version 2.0 drops another smaller .exe of equivalent functionality but packed by a different executable packer into %User Profile%\Application Data\[randomfolder]\ [random name].exe (where [random name], [random folder] is generated based on Mersenne's pseudo-random number generator).

For comparison’s sake, version 1.x copied itself to a fixed path with a constant filename under “%System%” and added some random content in its appended data every time, making its file hash different.

Revival

Version 2.0 adds the following registry key:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\"{[VOL_GUID]}" = "%User Profile%\Application Data\[random folder]\[random name].exe"

(Where [VOL_GUID] is the volume GUID of WINDOWS mount point. And [random name], [random folder] is generated based on Mersenne's Pseudo random number generator.)

This is compared to version 1.x, which added HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Userinit" = "%System%\userinit.exe, %System%\<threatfilename>.exe"

“Exclusive” rights

Mutex and PIPE names used for the exclusion and inter-thread communications in 2.0 now use GUID (instead of fixed Mutex names, as in 1.x). This enables multiple instances of Zbot, created by different bot builders, to run alongside one another simultaneously.

“Phish” me not

Disables Internet Explorer’s phishing filter by setting the following keys:

HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\PhishingFilter\"Enabled" = dword:0

HKEY_CURRENT_USER\Software\Microsoft\InternetExplorer\PhishingFilter\"EnabledV8" = dword:0

“Passwords” are (not) safe

Version 2.0 steals passwords from the following FTP clients, which is similar to what was seen in version 1.x:

“FlashFXP”, “total_commander”, ”ws_ftp”, “fileZilla”, “FAR2”, “winscp”, “ftp_commander”, “coreFTP” & “smartftp”.

“Log” my move

It can also take screen shots and log keystrokes (by hooking API 'TranslateMessage')

All your “data” belongs to me

It adds the following extra headers to the http communication through hijacked API’s 'HttpSendRequestW', 'HttpSendRequestA', 'HttpSendRequestExW', and 'HttpSendRequestExA'.

"Accept-Encoding: identity\r\n"
"TE:\r\n"
"If-Modified-Since:\r\n"

All of the network data that is sent out through the APIs 'send' and 'WSASend' is uploaded to a pre-configured anonymous FTP site.

Keys are no longer “private”

Attempts to steal private keys from certificates by hooking API 'PFXImportCertStore', similar to version 1.x.

“Change” is inevitable

By hooking the ‘PR_Read’ and ‘PR_Write’ APIs of NSPR4.dll, it is able to modify and read data sent by Mozilla Firefox, whereas versions 1.x could not. This means it has the ability to inject code forms and code into banking websites browsed through by Mozilla Firefox.

For the sake of “argument”

Somewhat unusually, Zbot 2.0 also accepts command line arguments. It accepts "-f" and “-n” as parameters. “-f” is used to alter registry use and “-n” is to prevent dropper threats from being self-deleted. This was probably used during the testing phase of Zbot and accidentally included in this release.

A little per-“spec”-tive

The Zeus config file has undergone changes too. Additional layers of encryption were added to deter security analysts—below are a few screen shots of a decoded config file:

The screenshot above shows some search strings that are used by Zeus to collect user information from various popular sites.

The above screenshot of a config file shows Zeus's malicious JavaScript code that will be injected into the web browser in order to steal some private questions and answers that the user has provided to a banking website.

The above screenshot shows script code that tricks users into giving away their bank PIN numbers. Such code injection into a Web browser can trick an everyday user into believing that their bank is legitimately requesting the information.

Symantec's protection summary on Trojan.Zbot can be found at http://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-99. We at Symantec continuously monitor and track such Trojans and update generic detection signatures as soon as new variants are released. Please, as always, keep your antivirus and IDS/IPS products up to date to ensure the best possible protection against threats.