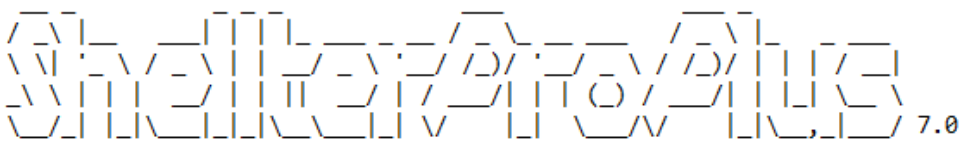

INTRO

This document provides an overview of the exclusive features added in Shellter Pro Plus.

Shellter Pro Plus introduces several new features that provide runtime evasion against AV and EDRs.

It is recommended to operate Shellter Pro Plus via our GUI interface which provides access to all the features and offers a simple and intuitive way of operating our software for best results.

For more details about how to use these features, run Shellter with the -h argument, and/or take a look at the demos that are available through our official website.

If you still need to know more about a specific feature, send an email to support@shellterproject.com.

SHELLTER PRO PLUS 7.X – ADDITIONAL FEATURES

EXTENDED RUNTIME EVASION CAPABILITIES II

This release brings additional enhancements to the runtime evasion capabilities by adding code that monitors in real time for modules being loaded into the process.

Advanced payloads usually require to load additional modules in order to complete several tasks. Since security software will commonly monitor these events through kernel-mode callbacks, it may optionally hook additional modules beyond the usual suspects such as kernel32 and ntdll.

Our latest additions, to the code that is bundled with your chosen payloads, will now monitor for newly loaded modules that are by default found under the 'KnownDlls' directory and will make sure that these will also be checked for hooks and other artefacts.

THIS PART INTENTIONALLY LEFT BLANK

SHELLTER PRO PLUS 6.X – ADDITIONAL FEATURES

EXTENDED RUNTIME EVASION CAPABILITIES

This release includes multiple updates towards runtime evasion against various techniques used by security software to intercept searching and calling system functions; especially those exported by kernel32, kernelbase, and ntdll DLLs.

This particular entry might not look too fancy, but the internal updates associated with this enhancement offer a lot more than meets the eye.

AMBUSH PAYLOAD EXECUTION

When this feature is enabled, it will set all injected payloads into hibernation until the specified benign DLL is loaded by the process.

This special feature offers the ability to perform deep infection of an application while evading automated analysis systems, and execution emulation.

It also offers the capability to simulate threat actors that compromise supply chains in order to distribute their malware though legitimate software that have previously infected by using a more advanced technique.

In order to take advantage of this feature, the specified DLL must not be statically linked, and only loaded when the user activates/interacts with a specific feature of the infected application.

There is a demo on our website which demonstrates the required steps to use this great feature effectively.

ANTI-DLL LOAD MONITORING

This feature removes user-mode registered callbacks that may be set by modules injected by security software inside the process in order to monitor for new DLL loading events.

THIS PART INTENTIONALLY LEFT BLANK

SHELLTER PRO PLUS 5.X - ADDITIONAL FEATURES

ADVANCED DEBUGGER DETECTION (KM + UM)

Shellter Pro Plus offers the ability to insert additional code that is able to detect both kernel and user mode Windows debuggers. This feature is enabled by default in 'Auto' mode for both types.

If you use Shellter Pro Plus via 'Manual' mode and/or through our GUI/CLI interfaces you can choose the level of detection in order to target one or the other type only if you wish.

If user mode debugger detection is enabled, the added code will check if the process is being debugged by a user mode debugger.

If kernel mode debugger is enabled, the added code will check if the kernel debugger is currently enabled in the OS, which is normally disabled by default.

If a debugger is detected, then the payloads will not run.

If remotely-fetched AES-128 key/iv pair feature is used then there will be no attempt to download this data.

This feature can be combined with 'Decoy Payloads' feature.

ADVANCED VM/SANDBOX DETECTION

Shellter Pro Plus offers the ability to insert additional code that is able to detect both type-1 and type-2 hypervisors.

In a few words, both bare metal such as ESXI, Hyper-V, KVM etc..., and the usual suspects such as VMWare Workstation, Oracle VirtualBox and so on.

If you use Shellter Pro Plus via 'Manual' or 'Auto' modes then you can only choose a VM detection profile based on certain hardware resources available to the system, such as number of CPU cores, available RAM etc....

If you use Shellter Pro Plus via our GUI/CLI interfaces you can choose additional options, including our 'SecretSauce' which is able to detect both hypervisor types by using low level checks.

If a hypervisor is detected, then the payloads will not run.

If remotely-fetched AES-128 key/iv pair feature is used then there will be no attempt to download this data.

This feature can be combined with 'Decoy Payloads' feature.

DECOY PAYLOADS

Shellter Pro Plus offers the ability to insert a payload that will be used as a decoy in case the Debugger/VM detections trigger.

This can be used to tamper with both automated analysis sandbox systems and with manual analysis of your binary.

The decoy payload can be as complex as you wish such as a reflective DLL that performs multiple actions, or just a simple stager that connects to a fake IP/Port.

In order to use this feature, Debugger and/or VM detection must be enabled.

You will then have to choose a minimum of 2 payloads to inject into the clean binary. Keep in mind that in this case the first payload will be the decoy and the rest will be any 'real' payloads that you wish to execute normally.

So, make sure that you don't use one of the good payloads as the first one if you enable this feature.

If remotely-fetched AES-128 key/iv pair feature is used then there will be no attempt to download this data. The decoy payload will be encrypted using a separate AES-128 key/iv pair which will be embedded inside the binary.

If you don't enable the decoy payload feature and a Debugger/VM are detected, then none of the injected payloads will run.

AES-128 Payload Encryption

Shellter Pro Plus encrypts all payloads with AES-128 (CBC mode) algorithm by using a key/iv pair that is generated randomly every time.

Normally, this pair is embedded inside the binary that will run your payloads, but there's also the option to fetch the decryption key/iv pair from a remote location.

This allows you to control for how long someone can potentially analyse your special payloads if they get a copy of your binary. Once you have removed the keys from your server/endpoint, then nobody will be able to decrypt and analyse them.

Two methods are currently supported:

1. HTTP/HTTPS URL
2. UNC

Shellter Pro Plus will ask you to supply the full remote path to the file that contains the AES key information data.

For example, *"https://myserver.com/dir1/dir2/key.aes"* or *"\\myserver.com\\dir1\\dir2\\key.aes"*.

A key file with the specified name will be saved locally in the working directory using the file name that you specified in the path (i.e "key.aes"). This file must be placed at the exact remote path as specified previously.

When using a URL to fetch the data the following rules apply:

1. Can use both HTTP and HTTPS protocols.
    a. Redirections from HTTP to HTTPs are allowed.
    b. Redirections from HTTPS to HTTP are not allowed.
2. SSL Certificates checks.
    a. Validity dates.
    b. Hostname given in the request.

You can also use UNC type paths to easily fetch the key from another computer on the same or another network.

Keep in mind that whichever path type you use, you need to make sure that the file is accessible without requiring any type of authentication.

Advanced Self-Unhooking

Shellter Pro Plus offers the ability to insert additional code that is able to remove hooks placed in native and other Ntdll functions.

These hooks are commonly used by security software to monitor for abnormal behaviour before a system call is executed.

This additional injected code will run before your payloads in order to remove such 'redirections' to the monitoring code that is already loaded inside the process by the security software during process initialisation.

Advanced Heuristic Unlinking of AV/EDR modules

Shellter Pro Plus offers the ability to insert additional code that is able to unlink decoy modules placed inside the "*Process Environment Block"* (PEB) "*Loader Data"* linked-list structures that provide information for all loaded modules in the process.

Some security software may use those decoy modules in order to detect manual parsing of the export table of system modules, most commonly those of kernel32 and ntdll, by renaming the original modules and setting the decoys earlier in the linked list.

Advanced native Imports Redirection for loaded modules

Shellter Pro Plus offers the ability to insert additional code that is able to redirect *"Import Address Table"* (IAT) pointers (of already loaded modules) that hold addresses of native ntdll functions in order to evade those hooks even if they get replaced by the security software.

Advanced Stealth Payload Thread Creation

Shellter Pro Plus Plus offers the ability to insert additional code that is able to kick of the main thread of each of the injected payloads by using legitimate code pivots that reside inside the already loaded modules of the process.

This helps to evade detections that look for suspicious thread start activity that occurs when the start address resides outside of a loaded module.

Advanced ETW Evasion

Shellter Pro Plus Plus offers the ability to insert additional code that is able to blind the *"Event Tracing for Windows"* (ETW) functionality for the current process.

This is normally used in order to log specific events that occur inside a process which are then evaluated by the security software.

Advanced AMSI Evasion

Shellter Pro Plus Plus offers the ability to insert additional code that is able to blind the *"Antimalware Scan Interface"* (AMSI) of Windows.

This is normally used by Windows Defender Windows Defender and other security software in order to scan memory buffers of the process upon request.

Advanced Self-Process and Payload Threads Protection

Shellter Pro Plus Plus offers the ability to insert additional code that is able to modify the security information of the process executing your payloads as well the security information of their main threads that are kicked off by our code.

This can make manual analysis through debugging to become harder since some usermode debugger's process access requests may be blocked.

Advanced Direct SysCalls-Based Runtime Evasion

Shellter Pro Plus Plus does all the above by using direct syscalls in order to evade hooks and other monitoring code injected in the process that executes your payloads.

In addition, extra low-level techniques are implemented in order to tamper with the ability of security software to detect the usage of direct syscalls.

Author: Kyriakos Economou
Insainted Ltd - www.ShellterProject.com

Twitter: @kyREcon / @ShellterProject
Email: kyrecon@shellterproject.com