In part 1 we saw the first layer of SRP protection.

This was related to the "**Designated File Types"** list (DFT) and the **ShellExecute()** API function. If a file extension is on the DFT list, then ShellExecute() can ask SRP what to do and prevent opening the file by the user. But still, the file can be opened by using a command line with sponsor, for example:

mshta.exe %Userprofile%\Desktop\How2BeRich.hta

In this case, ShellExecute() will be skipped. This leaves Windows with two possibilities:

1st: Allow opening the file.

2nd: Use another mechanism to call into SRP.

In most cases, Windows will follow the first way, but in certain cases, another mechanism can be used to call into SRP. It is based on so-called **Privileged Objects: "CMD Host", "Windows Script Host", and "Windows Installer".** They can call into SRP regarding supported file types. Thus, "How2BeRich.vbs" file can be monitored (blocked) because "Windows Script Host" can call into SRP regarding VBS files (even when VBS is not on the DFT list).

The above-mentioned **Privileged Objects** can control file execution independently of ShellExecute() and the DFT list. Furthermore, file blocking by **Privileged Objects** is more comprehensive than by ShellExecute(). For example, let's run How2BeRich.vbs by:

(A) Double click from the Desktop

(B) Executing the command line:

wscript.exe %Userprofile%\How2BeRich.vbs

If we add the VBS extension to the DFT list (in Hard_Configurator, it is on the list by default), then:

ShellExecute() can block (A) but not (B).

Privileged Objects can block both (A) and (B).

Additionally, **Privileged Objects** can block supported files even when those files have changed extensions.

Although users mostly run files by clicking on them or pressing the Enter key (not executing command lines), protection from (B) is quite important, for the following reason: An Office document macro or any shortcut can execute command lines. The same is usually true when the system or application is exploited.

Based on the above, we see that scripts in BAT, CMD, JS, JSE, VBS, VBS, WSF, WSH, and MSI files can be blocked in two independent ways:

1st: By extension (DFT list, ShellExecute()).

2nd: By Privileged Objects ("CMD Host", "Windows Script Host", and "Windows Installer").

The same applies to COM and SCR files which use ShellExecute() to find out that they are just a special kind of EXE file.

The native Windows executable EXE files always run without invoking Shell Execute(). The API function CreateProcess() is used instead, and it can also call into SRP to block COM, EXE, and SCR files.

There is also LoadLibrary() API function that can call into SRP. It is invoked when an EXE file is executed and tries to load binary libraries (DLL and OCX files). SRP can be configured either to allow or to block loading libraries from specified locations. This

layer of protection can stop many DLL attacks, when system files are used to load malicious libraries from User Space. For example (paths omitted for simplicity):

InstallUtil.exe /logfile= /LogToConsole=false /U malicious.dll

regsvcs.exe malicious.dll

regasm.exe /U malicious.dll

regsvr32 /s /u malicious.dll

rundll32 malicious.dll,EntryPoint

SRP has three basic enforcement settings:

1st: No Enforcement.

It ignores DFT list, ShellExecute(), CreateProcess(), and LoadLibrary() calls. SRP will only answer calls from **Privileged Objects** ("CMD Host", "Windows Script Host" and "Windows Installer").

2nd: Skip DLLs.

It ignores only LoadLibrary() calls. Other security layers: DFT list, ShellExecute(), CreateProcess(), CMD Host, Windows Script Host, and Windows Installer calls are not ignored.

3rd: All files.

It activates all SRP security layers. Protection via CreateProcess(), LoadLibrary(), CMD Host, Windows Script Host, and Windows Installer will apply even if files have changed extensions.

The “All files” enforcement setting is the most powerful. However, programs that use dlls or other binary libraries located in UserSpace will not run properly. Thus, all such libraries must be whitelisted.

In Windows Pro, there are two additional options displayed in the “enforcement” window:

(A) “Apply software restriction policies to the following users”.

This has two settings: (1) All users. (2) All users except local administrators. The first applies SRP to all users, including local administrators, so SRP can control processes running with Administrative Rights. The second applies SRP to all users except for local administrators, thus allowing elevated processes to bypass SRP.

(B) “When applying software restriction policies”.

This has two settings: (1) Enforce certificate rules, (2) Ignore certificate rules.

In Windows Home (Vista and later versions), the above options are usually set to “All users except local administrators” + “Ignore certificate rules”. These are the default settings in Hard_Configurator.

The three basic enforcement settings (“No enforcement”, “Skip DLLs”, “All files”) are very important because they tell SRP what kind of files should be monitored (the rest are simply ignored by SRP).

It is worthwhile to remember that DFT list and most of the executables are ignored with “No enforcement” setting − only files supported by CMD Host, Windows Script Host, and MSI Installer are monitored.

End of part 2.

@andyful

text correction @shmu26

This is a corrected version of text available on the MALWARETIPS thread:

https://malwaretips.com/threads/how-do-software-restriction-policies-work-part-2.69451/