In parts 1 and 2, I described how SRP determines which file types to monitor.

This determination depends on:

1. Enforcement settings: “No enforcement”, “Skip DLLs”, “All files”.
2. The way of running files: “Via command line” or “From Explorer shell”.

In the highest Enforcement level = “All files”, SRP can monitor the following file types:

1. Those on the “Designated File Types” list (configurable by the user).
2. Those supported by “Windows CMD” (Command Prompt), “Windows Script Host”, and “Windows Installer”. For example: BAT, CMD, JS, JSE, VBS, VBE, WSF, WSH, and MSI.
3. Native executables: EXE (COM, SCR).
4. Binary libraries: DLL, OCX, which are loaded by LoadLibrary API function.

In the “Skip Dlls” Enforcement level, SRP monitors all the above file types except DLL and OCX.

In the “No Enforcement” level, SRP monitors only the file types supported by “Windows CMD” (CMD Host), “Windows Script Host”, and “Windows Installer”.

So, for example, in “All files” Enforcement level, EXE files are monitored, and thus can be blocked. But this does not mean that all EXE files will be blocked.

*The final determination whether to block or allow the file is made after checking certain other SRP settings, namely: **”Unrestricted/Basic User/Disallowed” rules, and “Unrestricted/Basic User/Disallowed” Default Security Level (DSL) settings.***

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

★ "Apply software restriction policies to the following users", with 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 which ask for Administrative Rights. The second applies SRP to all users except local administrators, so elevated processes can bypass SRP.

★ "When applying software restriction policies", with 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". Those are the default (hardcoded) settings in Hard_Configurator.

Here are some definitions from a relevant Microsoft page:

You can define a default security level of Unrestricted or Disallowed for a Group Policy Object (GPO) so that software is either allowed or not allowed to run by default. You can make exceptions to this default security level by creating software restriction policies rules for specific software.

For example, if the default security level is set to Disallowed, you can create rules that allow specific software to run.

Working with hash rules

A hash is a series of bytes with a fixed length that uniquely identifies a software program or file. The hash is computed by a hash algorithm. When a hash rule is created for a software program, software restriction policies calculate a hash of the program. When a user tries to open a software program, a hash of the program is compared to existing hash rules for software restriction policies. The hash of a software program is always the same, regardless of where the program is located on the computer. However, if a software program is altered in any way, its hash also changes, and it no longer matches the hash in the hash rule for software restriction policies.

Working with certificate rules

Software restriction policies can also identify software by its signing certificate. You can create a certificate rule that identifies software and then allows or does not allow the software to run, depending on the security level. For example, you can use certificate rules to automatically trust software from a trusted source in a domain without prompting the user. You can also use certificate rules to run files in disallowed areas of your operating system. Certificate rules are not enabled by default.

Working with path rules

A path rule identifies software by its file path. For example, if you have a computer that has a default security level of Disallowed, you can still grant unrestricted access to a specific folder for each user. You can create the path rule by using the file path and setting the security level of the path rule to Unrestricted. Some common paths for this type of rule are %userprofile%, %windir%, %appdata%, %programfiles%, and %temp%. You can also create registry path rules that use the registry key of the software as its path. Because these rules are specified by the path, if a software program is moved, the path rule no longer applies.

Working with Internet Zone rules

Internet zone rules apply only to Windows Installer packages. A zone rule can identify software from a zone that is specified through Internet Explorer. These zones are Internet, Local intranet, Restricted sites, Trusted sites, and My Computer. An Internet Zone rule is designed to prevent users from downloading and installing software.

https://technet.microsoft.com/en-us/library/hh994597(v=ws.11).aspx

All rules and settings are processed in order, as follows (left rule wins over the right one):
Hash rules -> Certificate rules -> Path rules -> Zone rules -> DSL settings

Hash, Certificate, Path, and Zone rules take precedence over Default Security Level (DSL) settings, so they can modify DSL.

For example, Unrestricted Hash rule means that **any file** matching the hash can be opened, even if the file is monitored. Disallowed Hash rule means that **any monitored file** matching the hash will be blocked. Disallowed rule can block the file only if the file type is monitored by SRP. If not, the rule is ignored. Therefore, the hash rule for 'readme.txt' file will normally be ignored by SRP, because TXT file type is not normally monitored by SRP.

Neither the Unrestricted nor the Disallowed rules are necessarily absolute rules. They can be mixed and layered with each other to refine the whitelisting, so it is important to know which

is the winning rule, which takes precedence. In Windows built-in SRP, the BlackList and the WhiteList should be treated as one RuleList, since Unrestricted and Disallowed rules modify one another, as can be seen in the section "**Let's look at some examples of combined Unrestricted and Disallowed path rules."**

For home users, the most important are Hash rules, Path rules, and DSL settings. All of them can be set to Unrestricted, Disallowed, or Basic User.

The Unrestricted/Disallowed rules are common in SRP, but not "Basic User" rule, which is an artifact of Windows XP. It works differently in Windows XP and Vista, as compared to Windows 7+. Starting from Windows 7, the "Basic User" rule was converted to "Disallowed special type" rule. Now, it is almost identical to the standard Disallowed rule, but it manages scripts and shortcuts (LNK files) differently. For simplicity, I will skip "Basic User" rules in this article. I will often replace the term "WhiteList/BlackList" with the term "Unrestricted/Disallowed".

Let's look at some examples of Unrestricted path rules.

"Unrestricted path rules" tell SRP which monitored files should be allowed. The path rules can incorporate the "?" and "*" wildcards.

The "Skip DLLs" Enforcement setting is assumed to apply in all examples. It greatly simplifies the rules because usually there often are many DLLs loaded by a single EXE file.

EXAMPLES

- The below three rules whitelist the System Space in WIndows 64-bit:

  C:\Windows

  C:\ProgramFiles

  C:\ProgramFiles (x86)

  They allow running most of applications already installed in the system.

- D:\Portable*.exe

  allows executing any EXE file in “D:\Portable” folder, but not in its subfolders.

- D:\Portable*.abc

  is ignored because ABC file type is not monitored by SRP.

- D:\Portable**.exe

  allows executing any EXE file in any “D:\Portable” root subfolder, but not in deeper subfolders. Thus, the EXE files in “D:\Portable\Tools\” will be allowed, but not in the folder D:\Portable\Tools\Sysinternals\.

- D:\Portable**.dll

  is ignored in “Skip DLLs” enforcement setting.

- D:\Portable*

  allows executing any monitored file in “D:\Portable” folder, and all its subfolders.

- D:\Portable*\

  allows executing any monitored file in any “D:\Portable” **subfolder** (no matter how deep).

- C:\Users*\Appdata\Local\Microsoft\Onedrive\??.?.????.????*.exe

  allows running OneDrive executables, in the folder for any specific version, for any user. This folder is written with wildcards “??.?.????.????”, because the new OneDrive version changes the folder name (while keeping the ??.?.????.???? format).

- C:\Windows\Temp*.js

  allows executing JS scripts in “C:\Windows\Temp” folder, but not in subfolders.

- %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProgramFilesDir (x86)%

  the registry path rule is equal to whitelisting “Program Files (x86)” folder in the 64-bit Windows version. If we look in the registry key:

  “HKLM\Software\Microsoft\Windows\CurrentVersion”

  we see, under the “ProgramFilesDir (x86)” value, the REG_SZ string with path to this folder (usually “C:\Program Files (x86)”).

- %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\SystemRoot%

  the registry path rule is equal to whitelisting “Windows” system folder. If we look in the registry key: “HKLM\Software\Microsoft\Windows NT\CurrentVersion” we see, under the “SystemRoot” value, the REG_SZ string with a path to this folder (usually “C:\WINDOWS”)

- %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProgramW6432Dir%

  the registry path rule is equal to whitelisting “Program Files” folder in 64-bit Windows version. If we look in the registry key:

  “HKLM\Software\Microsoft\Windows\CurrentVersion”

  we see, under the “ProgramW6432Dir” value, the REG_SZ string with path to this folder (usually “C:\Program Files”).”

- %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProgramFilesDir%

  the registry path rule is equal to whitelisting the default “Program Files” folder. If we look in the registry key: “HKLM\Software\Microsoft\Windows\CurrentVersion” we see, under the “ProgramFilesDir” value, the REG_SZ string with path to this folder (usually “C:\Program Files”).

According to the Microsoft approach, there are certain fixed “Rules of precedence”:

1. When the path rule specifies a folder, it matches any file contained in that folder and any files contained in subfolders (no matter how deep). Software Restriction Policies support local and Uniform Naming Convention (UNC) paths.
2. When the path rule specifies the file(s) in the folder, it does not match files contained in subfolders of that folder.
3. The global rules: “string.*”, “*.string”, “*string”, “string*”,”*string*”, etc. (we are using the word “string” to stand for any valid file extension or file name) take precedence over any folder rule (for example over C:\Windows). One must be very cautious when using global file rules (see examples).
4. Otherwise, the more specific rule takes precedence.

The following is a list of paths, from a very specific match (1) to a very general match (5):

1. Drive:\Folder1\Folder2\FileName.Extension
2. Drive:\Folder1\Folder2*.Extension
3. *.Extension
4. Drive:\Folder1\Folder2\
5. Drive:\Folder1\

It is clear that any file path rule takes precedence over any folder path rule.

Let's look at some examples of combined Unrestricted and Disallowed path rules.

"Disallowed rules" tell SRP which monitored files should be blocked. Disallowed rules often will compete with Unrestricted rules, so one should carefully apply the above-mentioned 'Rules of precedence'.

EXAMPLES:

(a)

Unrestricted

- C:\Windows

Disallowed

- regedit.exe

globally block the system executable "regedit.exe", but allow anything else from the folder "C:\Windows".

(b)

Unrestricted

- D:\Portable*.exe

Disallowed

- D:\Portable

allow any EXE file from “D:\Portable” (subfolders not included), and block any other monitored file in this folder (subfolders included).

(c)

Disallowed

- *.abc

is ignored, because ABC file type is not normally monitored by SRP.

(d)

Unrestricted

- C:\Windows

Disallowed

- C:\Windows*.vbs

allow any monitored file in C:\Windows (subfolders included), except VBS files in C:\Windows (subfolders not included).

(e)

Unrestricted

- C:\Windows

Disallowed

- *.vbs

globally block VBS files. Therefore, whitelisting “C:\Windows” folder is valid in this example for any monitored file type except VBS. One should be very careful when using global rules. They restrict “System Space” as well.

(f)

Unrestricted

- C:\Windows\Temp*.js

Disallowed

- *.js

block JS files globally, except in “C:\Windows\Temp” (subfolders not included).

(g)

Disallowed

- *malware*

blocks any file (but not folder) whose name contains the string “malware”.

(h)

Unrestricted

- C:\MyTestMalware\malware.exe

Disallowed

- malware.exe

globally block the file malware.exe , except for C:\MyTestMalware\malware.exe.

Now let’s look at “Default Security Level” (DSL).

It has three settings: “Disallowed”, “Basic User”, “Unrestricted”.
These settings can be modified by Hash, Certificate and Path rules. Such rules can be also “Disallowed”, “Basic User” or “Unrestricted”, but they take precedence over DSL.

“Disallowed” DSL setting blocks by default all monitored files, except those that match the winning Unrestricted/Disallowed rules (Default_Deny_1).

“Basic User” DSL setting (in Windows 7+) blocks by default *almost* all monitored files, except for those that match the winning Unrestricted/Disallowed rules (Default_Deny_2).

The differences between Default_Deny_1 and Default_Deny_2 follow from a different method of script blocking and shortcut management.

- Default_Deny_1 can use “Windows CMD” and ”Windows Script Host” to block monitored scripts of these types: BAT, CMD, JS, JSE, VBE, VBS, WSF, and WSH.
  However, Default_Deny_2 cannot do this. Therefore, Default_Deny_2 can block the above scripts only if they are on the DFT list and are not run by command line.

- Default_Deny_1 applies standard shortcut blocking, and Default_Deny_2 does it in a non-standard way.

Unrestricted (**Default Allow**) DSL setting allows execution/opening of all files, except those monitored files that match the winning Disallowed rules. This "Default Security Level" setting is adopted in some Anti-Ransomware programs (CryptoPrevent, SBGuard). However, it is not easy to build SRP around this setting. For example, CryptoPrevent uses thousands of rules to apply its protection.

Disallowed/Basic User (**Default Deny**) DSL setting is much more convenient to use for restricting the system, because it is not necessary to write a great number of Disallowed rules. This setting is adopted in the recommended settings of Hard_Configurator.

We will now define what it means when we say that a file is whitelisted or blacklisted. As we will see below, it is not always true that whitelisted files can be opened.

A specific file is called whitelisted when both 1 and 2 are true:

1. It is monitored by SRP.
2. It matches a winning Unrestricted rule, or there are no matching rules but "Default Security Level" = "Unrestricted" is applied.

A specific file is blacklisted when both A and B are true:

A. It is monitored by SRP.

B. It matches a winning Disallowed rule, or there are no matching rules but "Default Security Level" = "Disallowed" ("Basic User") is applied.

DEF1

A specific file can be opened if both the file and its Sponsor are not blacklisted.

DEF2

If *either* the file *or* its Sponsor is blacklisted, that file cannot be opened.

The above definitions are true for all files, except for shortcuts (*.LNK) when "Default Security Level" = "Basic User" is applied. In this setting, shortcuts are treated as a pair of files (shortcut as file1, target as file2), so the case is more complicated.

It is worthwhile to know that all files can be split into three mutually exclusive groups: ignored by SRP, whitelisted, or blacklisted. This means, for example, that a non - blacklisted file will either be ignored by SRP or be whitelisted.

Let's look at some examples with a typical default deny SRP config (Windows 64-bit):

- Designated File Types: ADE, DP, BAS, BAT, CHM, CMD, COM, CPL, CRT, EXE, HLP, HTA, INF, INS, ISP, LNK, MDB, MDE, MSC, MSP, MST, OCX, PCD, PIF, REG, SCR, SHS, URL, VB, WSC.
- Enforcement = "Skip Dlls" -> all Designated File Types and also COM, EXE, SCR, BAT, CMD, JS, JSE, VBS, VBE, WSF, WSH, MSI files are monitored.
- Unrestricted rules for "System Space" folders:
  C:\Windows, C:\Program Files, C:\Program Files (x86)
- Default Security Level = "Disallowed" (**Default Deny**), so all monitored files are blocked outside the above folders (= "User Space").

EXAMPLES

In the first example, we want to open the file "D:\config.vbs" by means of Windows Explorer. The file is monitored (Windows Script Host calls into SRP). It does not match any rules, but Default Security Level = "Disallowed" is applied. So, it is blacklisted and cannot be opened.

In the second example, we want to open the script file "D:\config.vbs" using the command:

"C:\Windows\system32\wscript.exe D:\config.vbs".

The Sponsor "wscript.exe" is monitored (CreateProcess calls into SRP). It matches the winning Unrestricted rule (C:\Windows). So, it is whitelisted and can be run.

The script file is monitored (Windows Script Host calls into SRP). The file does not match any rules, but Default Security Level = “Disallowed” is applied. Therefore, it is blacklisted and cannot be opened.

In the third example, we want to open the BAT file “D:\config.bat” by means of Windows Explorer.

The file is monitored because BAT extension is on the DFT list (ShellExecute calls into SRP). It is also monitored via “Windows CMD”. The file does not match any rules, but Default Security Level = “Disallowed” is applied. So, it is blacklisted and cannot be opened (“WindowsCMD” is not invoked.)

In the fourth example, we want to open the REG file “D:\config.reg” by means of Windows Explorer.

The file is monitored because the REG extension is on the DFT list (ShellExecute calls into SRP). It file does not match any rules, but Default Security Level = “Disallowed” is applied. Thus, it is blacklisted and cannot be opened.

In the fifth example, we want to open the REG file using the command:

“C:\Windows\regedit.exe D:\config.reg”.

The Sponsor “regedit.exe” is monitored (CreateProcess calls into SRP). It matches the winning Unrestricted rule (C:\Windows), so it is whitelisted and can be run.

The file is ignored by SRP because nothing can call into SRP to query about the file (DFT list is ignored when opening REG file by command line).

Since both the file and the Sponsor are not blacklisted, the file can be opened.

In the sixth example, we want to open the installation file “d:\setup.exe”.

The file is monitored (CreateProcess calls into SRP). It is a native Window executable, so does not need a Sponsor to execute.

The file does not match any rules, but Default Security Level = “Disallowed” is applied. So, it is blacklisted and cannot be run.

In the seventh example, we want to open the file readme.txt (in any location).

The file is ignored by SRP, so it is not blacklisted

The Sponsor “C:\Windows\system32\notepad.exe” matches the winning Unrestricted rule (C:\Windows). So, it is whitelisted and can be run.

Both the file and the Sponsor are not blacklisted, so the file can be opened.

In the eighth example, we want to open the file “D:\helloworld.txt”, which is the normal VBS script hidden in the TXT file (to avoid detection). This can be done by running the command:

“C:\Windows\system32\wscript.exe /e:vbscript D:\helloworld.txt”.

The Sponsor “wscript.exe” is monitored (CreateProcess calls into SRP) and matches the winning Unrestricted rule (C:\Windows). So, it is whitelisted and can be run.

The file “helloworld.txt” is monitored now, because the Sponsor triggers "Windows Script Host" that calls into SRP. The file does not match any rules, but Default Security Level = “Disallowed” is applied. Therefore, it is blacklisted and cannot be opened. If we will try to open “D:\helloworld.txt” from Windows Explorer, then it will be safely opened in the notepad, as in the previous example.

In the ninth example, we apply two additional “Disallowed” global rules:

- notepad.exe
- regedit.exe

Now, the Sponsors: “C:\Windows\system32\notepad.exe” and “C:\Windows\regedit.exe” will be blacklisted. So, the files “d:\config.reg” (fifth example) and readme.txt (seventh example) will not be processed (even in “System Space”).

Warnings!

- With a “Default Deny” type security level (i.e., “Disallowed” or “Basic User”), one has to remember to add Unrestricted rules for the “System Space” folders. If not, SRP will block critical files and make Windows unusable! (Hard_Configurator adds Unrestricted rules for “System Space” by default.)

- “Unrestricted/Disallowed/Basic User” **rules** should not be confused with the more general “Unrestricted/Disallowed/Basic User” **settings** of “Default Security Level”.
- If a Disallowed **folder path rule** takes precedence over a certain folder, then extended protection of “Windows CMD, Windows Script Host and MSI Installer” applies to supported files in that folder. This works for that specific folder independently of Enforcement and Default Security Level settings.
- In Windows XP and Windows Vista, the “Basic User” security level and “Basic User” rules work differently as compared to Windows 7 and later versions. In XP and Vista, for example, “Basic User” allows running EXE files, but they are blocked in Windows 7+ with the same settings. This can be confusing, so it is recommended not to use “Basic User” security level and “Basic User” rules in XP/Vista.

End of part 3.

@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-3.70582/