In the parts 1 and 2, I wrote how SRP know which file types should be monitored.
This information depends mostly on Enforcement settings: 'No enforcement', 'Skip DLLs', and 'All files'. Yet, it can be modified when the file is opened by the command with Sponsor ('Designated File Types' list is skipped for the file).

In most powerful Enforcement = **'All files'**, SRP can monitor the below file types:

1. Included in 'Designated File Types' list (configurable by the user)
2. Hosted by 'Windows CMD' (CMD Host), 'Windows Script Host', and 'Windows Installer': BAT, CMD, JS, JSE, VBS, VBE, WSF, WSH, and MSI.
3. Native executables: EXE (COM, SCR).
4. Loaded binary libraries: DLL, OCX.

In the **'Skip Dlls'** Enforcement setting, SRP monitor all the above file types, except DLL and OCX.

In the **'No Enforcement'** setting, SRP monitor only the scripts and Windows Installer: BAT, CMD, JS, JSE, VBS, VBE, WSF, WSH, MSI.

So for example, in 'All files' Enforcement setting, EXE files are monitored, and then can be blocked. But, does it mean that all EXE files will be blocked? The answer is negative. **The final decision Block/Allow is made after checking some other SRP settings: Unrestricted/Disallowed rules, and Default Security Level (DSL) settings.**

In Windows Pro, there are also two other options in enforcement window:

★ 'Apply software restriction policies to the following users:' with two settings: (1) All users, and (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, and (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 settings in programs: Simple Software Restriction Policies and Hard_Configurator.

Here are some definitions from 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 (Unrestricted/Disallowed) rules/settings are processed in order, as follows (left wins):
Hash rules -> Certificate rules -> Path rules -> Zone rules -> Default Security Level (DSL) settings**

For example, Unrestricted Hash rule, means that **any file** matching the hash can be opened (monitored or not). Disallowed Hash rule means that **any monitored file** matching that hash will be blocked. Any rule can block the file, only if its file type is monitored by SRP.
So, the hash rule for readme.txt file, will be usually ignored by SRP, because TXT files are usually not monitored by SRP.
‘Default Security Level = Disallowed’ setting, means that any **monitored file** will be blocked by default, if there are not matching (Hash, Certificate, Path, Zone) rules for that file.
Hash, Certificate, Path, and Zone rules win over Default Security Level (DSL) settings, so they can modify DSL.

For home users, the most important are Hash rules, Path rules, and DSL settings.
Certificate and Zone rules can be important in Enterprises.

In Windows built-in SRP, the BlackList and the WhiteList, should be treated as one RuleList with Unrestricted and Disallowed rules. There is also, another blacklist type rule, called 'Basic User'. This rule is an artifact of Windows XP. It works differently in 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 differently manages scripts and shortcuts (LNK files). For simplicity, I skip 'Basic User' rules in this article, and will adress this point later.
I will often replace the words 'WhiteList/BlackList' with 'Unrestricted/Disallowed rules'.
Let's look first, at some examples of the path rules.

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

Assuming 'Skip DLLs' Enforcement setting, the below Unrestricted rules have the meaning:

(A) C:\Windows
C:\Program Files
C:\Program Files (x86)
allow to execute any monitored file from the 'System Space' in Windows 64Bit.

(D) D:\Portable\*.exe
allows to execute any EXE file in 'D:\Portable' folder, but not in its subfolders.

(E) D:\Portable\*.abc
is ignored, because ABC file type is not monitored by SRP.

(F) D:\Portable\**.exe
allows to execute any EXE file in any 'D:\Portable' root subfolder (for example in ‘D:\Portable\Tools’), but not in deeper subfolders.
So, EXE files in ‘D:\Portable\Tools\Sysinternals’ will be blocked).

(G) D:\Portable\**.dll
is ignored in 'Skip DLLs' enforcement setting.

(H) D:\Portable\*
allows to execute any monitored file in 'D:\Portable' folder and its subfolders.

(I) D:\Portable\*\
allows to execute any monitored file in any 'D:\Portable' **subfolders** (no matter how deep).

(J) %LocalAppdata%\Microsoft\Onedrive\??.?.????.????\onedrivestandaloneupdater.exe
allows onedrivestandaloneupdater.exe to run from the User Space. The folder with wildcards '??.?.????.????' is used, because new OneDrive version changes the update folder name (keeping the ??.?.????.???? format).

(K) C:\Windows\Temp\*.js
allows to execute JS scripts in 'C:\Windows\Temp' folder, but not in subfolders.

(L) %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProgramFilesDir (x86)%
the registry path rule is equal to whitelisting 'Program Files (x86)' folder in 64Bit Windows version. If we look into the registry key: 'HKLM\Software\Microsoft\Windows\CurrentVersion', we can see under the 'ProgramFilesDir (x86)' value, the REG_SZ string with path to this folder (usually 'C:\Program Files (x86)').

(M) %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\SystemRoot%
the registry path rule is equal to whitelisting 'Windows' system folder. If we look into the registry key: 'HKLM\Software\Microsoft\Windows NT\CurrentVersion', we can see under the 'SystemRoot' value, the REG_SZ string with path to this folder (usually 'C:\WINDOWS')

(N) %HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProgramW6432Dir%
he registry path rule is equal to whitelisting 'Program Files' folder in 64Bit Windows version. If we look into the registry key: 'HKLM\Software\Microsoft\Windows\CurrentVersion', we can see under the 'ProgramW6432Dir' value, the REG_SZ string with path to this folder (usually 'C:\Program Files').

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

According to Microsoft way of thinking:

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., ('string' is a valid file extension or a valid file name), win over any folder rule (for example over C:\Windows). One must be very cautious using global file rules (see the above (e) rule analysis).
4. Otherwise, the more specific rule wins.

The following is a list of paths, from more specific match (1) to more 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 wins over any folder path rule.

'Disallowed rules' tell SRP which monitored files, should be blocked. Disallowed rules often will fight with Unrestricted rules, so one should carefully apply the above winning criteria.

Here are some examples:

(a)

Unrestricted
  C:\Windows
Disallowed
  regedit.exe
block globally the system executable 'regedit.exe', but allow anything else from '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 monitored by SRP (usually).

(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
block globally VBS files. So, 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 always restrict the 'System Space', too.
(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 (not folder), which name contains 'malware' string.
(h)
Unrestricted
  C:\MyTestMalware\malware.exe
Disallowed
  malware.exe
block the file malware.exe globally, except C:\MyTestMalware\malware.exe.

Let's look now at 'Default Security Level' (DSL). It has three settings:
'Disallowed', 'Basic User', 'Unrestricted'

'**Disallowed'** setting, blocks by default all monitored files, except those that matches the winning Unrestricted/Disallowed rules (Default Deny ①).

'Basic User' setting (in Windows 7+), blocks by default almost all monitored files, except those that match the winning Unrestricted/Disallowed rules (Default Deny ②).

The differences between ① and ② follows from a different way of script blocking, and shortcut maintenance.
The ① can use 'Windows CMD' and 'Windows Script Host' to block monitored scripts: BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, but the ② cannot. So, the ② will try to block the above scripts only if they are on DFT list and are not run by command.
The ① applies standard shortcut blocking, and the ② does it in a non standard way (I return to this point later).

Unrestricted (Default Allow) setting, allows execution/opening of all files, except those monitored files which match the winning Disallowed rules. This 'Default Security Level' setting is adopted in some Anti-Ransomware programs (CryptoPrevent, SBGuard). It is not easy to build SRP around this setting. For example CryptoPrevent uses a few thousands of rules to do it.

'Default Deny' is very convenient, because without it, one has to write many Disallowed rules. This setting is adopted in programs: Simple Software Restriction Policies and Hard_Configurator.

We can define what it means that the file is whitelisted or blacklisted. As we will see below, it is not true, that whitelisted file can be always opened.
The concrete 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.

The concrete File is **blacklisted** when both ③ and ④ are true:

③ It is monitored by SRP.
④ It matches a winning Disallowed rule or there are no matching rules, but ‘Default Security Level’ = ‘Disallowed’ (‘Basic User’) is applied.

**The concrete File can be opened, if both the File and its Sponsor are not blacklisted.
The concrete File cannot be opened, if one element of the pair File/Sponsor is blacklisted.**

The above definitions are true for all files, except shortcuts (*.LNK) when ‘Default Security Level’ = ‘Basic User’ is applied. In this setting shortcuts are treated as a pair of files (shortcut file1, target file2), so things are more complicated (I return to this topic later).

All files split into 3 not overlapping groups: ignored by SRP, whitelisted, blacklisted. It means, for example, that not blacklisted file has to be ignored by SRP or whitelisted.

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

✶ Designated File Types: ADE, DP, BAS, BAT, CHM, CMD, COM, CPL, CRT, EXE, HLP, HTA, INF, INS, ISP, LNK, MDB, MDE, MSC, MSI, 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 (only 'System Space' is whitelisted):
✶ C:\Windows
C:\Program Files
C:\Program Files (x86)
✶ Default Security Level = 'Disallowed', so all monitored files are blocked outside the above folders (= ‘User Space’).

In the first example, one wants to open the File 'D:\config.vbs' from Explorer.
The File is monitored (Windows Script Host calls into SRP).
The File does not match any rules, but Default Security Level = 'Disallowed' is applied.
So, it is blacklisted, and cannot be opened.

In the second example, one wants to open the script using command:
'C:\Windows\system32\wscript.exe D:\config.vbs'.
The Sponsor is monitored (CreateProcess calls into SRP).
The Sponsor matches the winning Unrestricted rule (C:\Windows) so it is whitelisted, and can be run.
The File is monitored (Windows Script Host calls into SRP).

The File does not match any rules, but Default Security Level = 'Disallowed' is applied.
So, it is blacklisted, and cannot be opened.

In the third example, one wants to open the BAT File 'D:\config.bat' from Explorer.
The File is monitored, because BAT extension is on DFT list (ShellExecute calls into SRP)
The File does not match any rules, but Default Security Level = 'Disallowed' is applied.
So, it is blacklisted, and cannot be opened.
The 'Windows CMD' is not invoked.

In the fourth example, one wants to open the REG File 'D:\config.reg' from Explorer.
The File is monitored, because REG extension is on DFT list (ShellExecute calls into SRP).
The File does not match any rules, but Default Security Level = 'Disallowed' is applied.
So, it is blacklisted, and cannot be opened.

In the fifth example, one wants to open the REG File, using command:
'C:\Windows\regedit.exe D:\config.reg'.
The Sponsor is monitored (CreateProcess calls into SRP).
The Sponsor 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 and ask about the File.
Both the File and the Sponsor are not blacklisted, so the file can be opened.

In the sixth example, one wants to open the installation File 'd:\setup.exe'.
The File is monitored (CreateProcess calls into SRP).
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, one wants to open the File readme.txt (in any location).
The File is ignored by SRP.
The Sponsor = 'C:\Windows\system32\notepad.exe'. It 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, one wants to open 'D:\helloworld.xyz' File, which is the VBS File with changed extension vbs -> xyz.
It can be done by running the command:
'C:\Windows\system32\wscript.exe /e:vbscript D:\helloworld.xyz'.
The Sponsor is monitored (CreateProcess calls into SRP).
The Sponsor matches the winning Unrestricted rule (C:\Windows) so it is whitelisted, and can be run.
The File is monitored (Windows Script Host calls into SRP).
The File does not match any rules, so Default Security Level = 'Disallowed' is applied.
So, it is blacklisted, and cannot be opened.
If we try to open 'D:\helloworld.xyz' from the Explorer, it fails, because Windows does not know which Sponsor should open that File.

If we add:
'Disallowed' global rules:
notepad.exe
regedit.exe
then the Sponsors: 'C:\Windows\system32\notepad.exe' and 'C:\Windows\regedit.exe' will be blacklis-

ted, so the files 'd:\config.reg' (fifth example) and readme.txt (seventh example) will not be processed (even in the 'System Space').

Warnings!

- With 'Default Deny' type security level ('Disallowed' or 'Basic User'), one has to remember adding Unrestricted rules for the 'System Space' folders - if not, then SRP will block many important files and make Windows unusable!
- 'Disallowed'/'Basic User' rules should not be confused with 'Disallowed'/'Basic User' settings in 'Default Security Level'.
- If Disallowed folder path rule wins for the concrete folder, then extended protection of CMD host, Windows Script Host and MSI Installer applies for files in that folder. This works for that concrete 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 to run EXE files - which are blocked in Windows 7+ with the same settings.
  It can be confusing, so it is better to skip 'Basic User' security level and 'Basic User' rules in XP/Vista.

End of part 3.