The Windows OS is like a castle with all its doors open. Software Restriction Policies close these doors in a way that only administrators can open.

Simple facts about Windows built-in Software Restriction Policies

1. Software Restriction Policies can be abbreviated in two ways:

   - SRP as a Windows built-in security feature (policies, rules, API integration, etc.).
   - SRPs as specific policies introduced by SRP.

   I will use the first meaning for simplicity.

2. SRP employs native Windows features, without impacting system performance or introducing software incompatibilities. It does not introduce additional processes, services or drivers.

3. SRP can be activated on all Windows versions, Windows XP and above.

4. On Windows Pro, Enterprise and Education, SRP can be configured using secpol.msc or gpedit.msc. On Windows Home, SRP can be configured only by directly modifying Windows Registry, or by means of utilities, namely: Simple Software Restriction Policies (no GUI), Hard_Configurator (with GUI).

5. SRP can be configured to not affect processes executing with administrative rights, so the Windows system works unhindered. Thus, there is no need to disable SRP protection when applying Windows updates or system-scheduled tasks or when installing/updating Universal Windows Platform (UWP) apps from Windows Store.

6. There are two common approaches to SRP:

★ Default Allow + Blacklist/Whitelist (this makes up a great part of the security in CryptoPrevent and SBGuard Anti-Ransomware).
★ Default Deny + Whitelist/Blacklist (used by Simple Software Restriction Policies and Hard_Configurator).

6. The approach that has the greatest protective power is a properly configured “Default Deny”, but such configuration requires some amount of knowledge about SRP. The “Default Allow” approach has certain advantages (on Windows 7 and below) when installing new programs. Most restrictions remain active during the installation

process, so accidental zero-day malware installations are mitigated. On the other hand, Default Deny is better against exploits, and in a broader sense, whenever malicious code (including zero-day) tries to run, either unknown to the user or by accident. However, most restrictions will be inactive when installing new programs. So if the user unwittingly installs zero-day malware, in most cases it is not mitigated by SRP.

7. It is possible to use Default Deny in daily work, and switch temporarily to Default Allow when installing new programs.

8. Properly configured SRP is easily usable even by inexperienced users, but the initial configuration should be done by advanced users.

9. On Windows 10, SRP + Windows Defender can be used for effective Windows built-in security, without the need for protection from third-party real-time security components. (Such protection can be further enhanced by means of the ConfigureDefender tool, which is incorporated in Hard_Configurator.)

10. On a vulnerable system, SRP is most effective on a Standard (limited) user account, which is much less prone to exploitation (system exploits, UAC bypasses). In this case, it is recommended to use Standard user account (SUA) during daily computer use, and switch to an Administrator account only when needed.

11. Default Deny SRP + No Elevation SUA (ConsentPromptBehaviorUser=0 in UAC settings) can be used to lock down the SUA, so there is no way to run/install new programs other than Universal Windows Platform (UWP) apps from Microsoft Store.

12. Properly configured SRP is compatible with third-party antivirus, antimalware, anti-exe, anti-exploit and HIPS solutions. But in some cases, this can be implemented only by experienced users.

13. SRP can prevent many “drive-by” attacks, but it can be bypassed by totally fileless malware or by VBA macros embedded in documents. However, these attacks are mitigated by other features of Hard_Configurator, as explained in the H_C manual.

14. Depending on SRP configuration, there can be other types of possible bypasses (using LOLBins i.e., Windows system files to bypass whitelisting, etc.). These, too, can be mitigated by SRP, but this will require advanced configuration, for example, blocking many administrative tools and sponsors from the Windows folder.

What happens under the hood when I double-click a file on my desktop?

Something needs to call into SRP (safer APIs) and get information on what should be monitored and blocked. The Windows OS has a special API function that is triggered, in order to manage the opening of files from Desktop (as well as from Windows Explorer and Internet Explorer). It is called ShellExecute().

Suppose you click the file How2BeRich.hta, then ShellExecute() function determines that the sponsor named **mshta.exe** should be used to open the HTA file. ShellExecute() has also a built-in ability to call into SRP, so if SRP is activated, then the How2BeRich.hta can be monitored and blocked. Blocking files by ShellExecute() prevents users from unknowingly or accidentally executing malicious code.

SRP has a list of file extensions called "Designated File Types." If the file extension is on this list, then ShellExecute() can prevent that file from opening. So the file How2BeRich.hta can be monitored and blocked by ShellExecute() if the HTA extension is on the Designated File Types list, as indeed it is by default. The list is configurable, and the user can add or remove file extensions at will.

These are the default 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.

The EXE extension is on the list for informational purposes only. SRP works just the same when it is removed from the list. This is because EXE files do not use the ShellExecute() function.

Now let's suppose you open the HTA file directly, using this command with HTA sponsor **mshta.exe**:

mshta.exe %Userprofile%\Desktop How2BeRich.hta

In this case, ShellExecute() will be skipped, and SRP will not know that a file is executing.

Windows will thus have two possibilities:

1. Allow the file to open.

2. Use another mechanism to call into SRP.

End of part 1.

@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-1.69275/