Hard_Configurator - Manual

Version 2.0.1.0 (February 2017)

Developer Web Page: https://github.com/AndyFul/Hard_Configurator

Distribution

This software may be freely distributed as long as no modification is made to it.

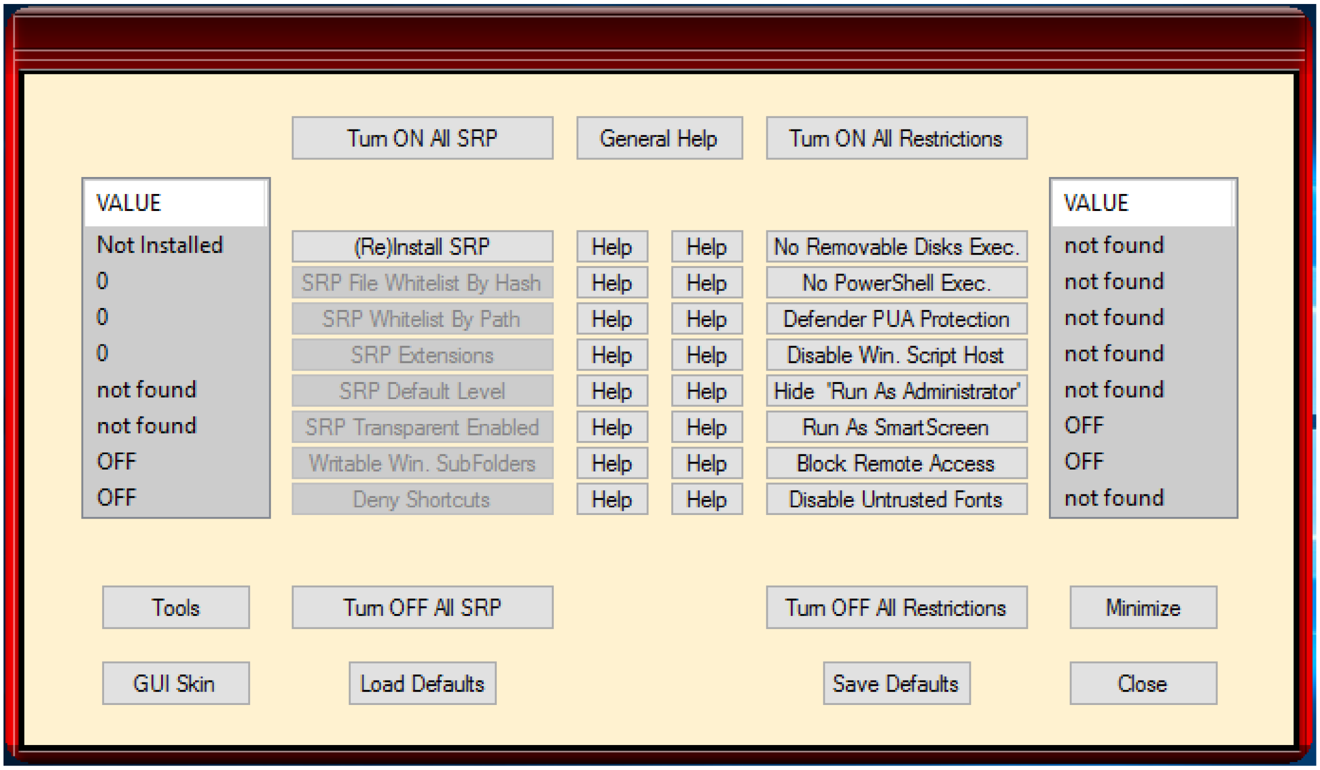
Disclaimer of Warranty

THIS SOFTWARE IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE IT AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING THIS SOFTWARE.

TABLE OF CONTENTS

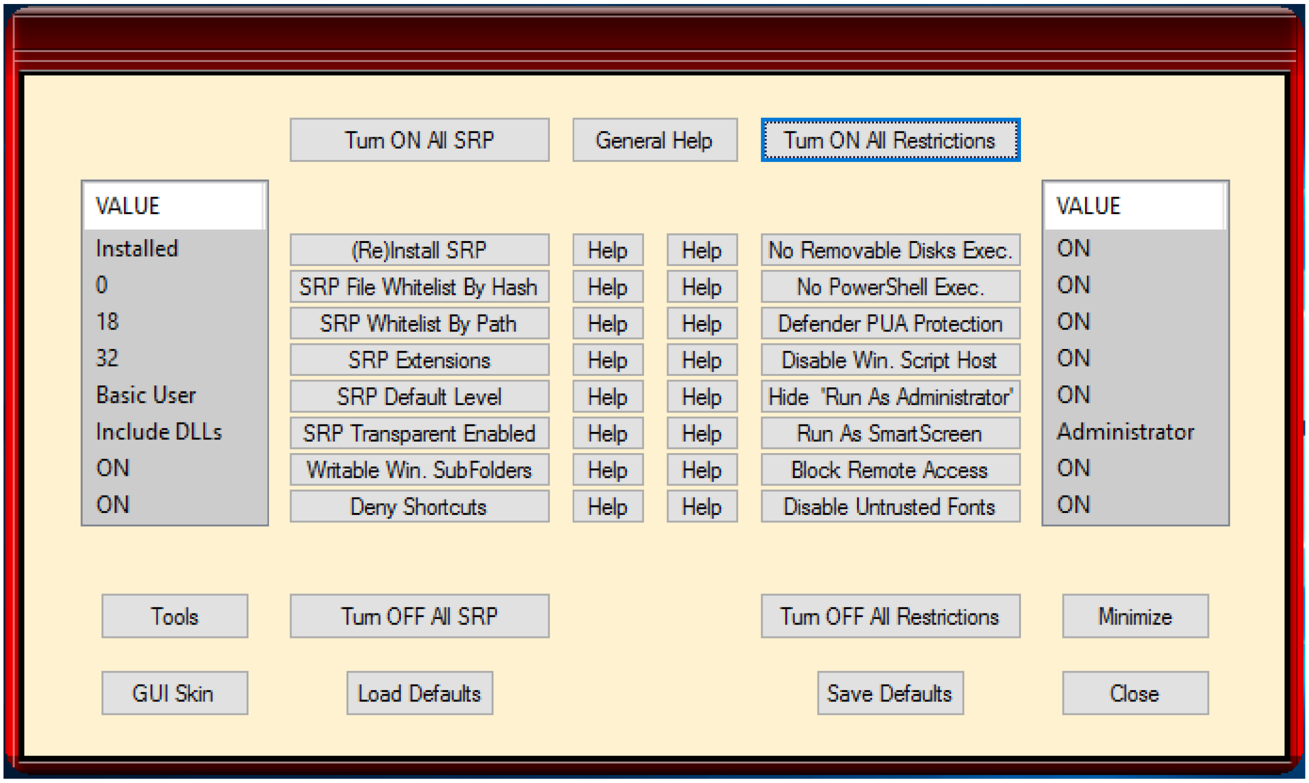
INTRODUCTION

1. Hard_Configurator works with Windows Vista and higher versions. It is intended for users, who want to use Windows built-in security. This program can manage some well-known system restrictions to harden the Windows OS. Actually, they are directed to control code execution, so in the wide sense, Hard_Configurator can be seen as a kind of anti-exe protection.



VALUE					VALUE
Not Installed	(Re)Install SRP	Help	Help	No Removable Disks Exec.	not found
0	SRP File Whitelist By Hash	Help	Help	No PowerShell Exec.	not found
0	SRP Whitelist By Path	Help	Help	Defender PUA Protection	not found
0	SRP Extensions	Help	Help	Disable Win. Script Host	not found
not found	SRP Default Level	Help	Help	Hide 'Run As Administrator'	not found
not found	SRP Transparent Enabled	Help	Help	Run As SmartScreen	OFF
OFF	Writable Win. SubFolders	Help	Help	Block Remote Access	OFF
OFF	Deny Shortcuts	Help	Help	Disable Untrusted Fonts	not found

2. Turning on all options (<Turn ON All SRP> and <Turn ON All Restrictions>) , gives users pretty good, set and forget security setup. Almost all programs can be run as usual by mouse click or pressing the ENTER key. Downloaded programs cannot run from the Web Browser. They should be first saved, and next "Run As SmartScreen" from DOWNLOAD folder using Windows Explorer context menu. Portable applications located outside 'Windows' and 'Program Files' folders can be whitelisted by hash (or by path), and then run as usual.



VALUE			
Installed	(Re)Install SRP	Help
0	SRP File Whitelist By Hash	Help
18	SRP Whitelist By Path	Help
32	SRP Extensions	Help
Basic User	SRP Default Level	Help
Include DLLs	SRP Transparent Enabled	Help
ON	Writable Win. SubFolders	Help
ON	Deny Shortcuts	Help

		VALUE
Help	No Removable Disks Exec.	ON
Help	No PowerShell Exec.	ON
Help	Defender PUA Protection	ON
Help	Disable Win. Script Host	ON
Help	Hide 'Run As Administrator'	ON
Help	Run As SmartScreen	Administrator
Help	Block Remote Access	ON
Help	Disable Untrusted Fonts	ON

3. There is no need to turn off the program options to install Windows Updates and perform system Scheduled Tasks. If everything works well, the settings can be saved (<Save Defaults> button). After temporary setting changes, the default settings can be retrieved using <Load Defaults> button.
4. Some precautions should be taken when turning on SRP and Restrictions. In some hardware/software configurations, a few autoruns located outside the 'Windows', 'Program Files' or 'Program Files (x86)' folders, may be blocked. It can be for example something in Hardware Manufacturer Folder (Intel, AMD, Lenovo, Asus, Toshiba, etc.). Hard_Configurator can utilize Sysinternals Autorunsc (command line), NirSoft FullEventLogView, and Advanced SRP Logging to filter out autoruns, and find problematic items, that should be whitelisted (see TROUBLESHOOTING paragraph for more info).
5. Some options are not supported by earlier Windows versions:
   <Disable Untrusted Fonts> - Windows Vista, Windows 7, 8, 8.1
   <PUA Protection> - Windows Vista, Windows 7
   <Run As Smartscreen> - Windows Vista, Windows 7
   <No Removable Disk Exec.> - Windows Vista
   <No PowerShell Exec.> - Windows Vista

INSTALLATION

0. Uninstall and delete the previous version of Hard_Configurator. Please do not install Hard_Configurator on laptops with the system on eMMC flash memory, because the program was not fully tested on this hardware.
1. Unzip Hard_Configurator.zip and copy the unzipped folder 'Hard_Configurator' to 'C:\Windows' (Administrative Rights are needed).
2. If Windows is 64Bit, run Hard_Configurator(x64).exe from 'C:\Windows\Hard_Configurator' folder.
3. If Windows is 32Bit, run Hard_Configurator(x86).exe from 'C:\Windows\Hard_Configurator' folder.
4. On the first run, Hard_Configurator makes by default the System Restore Point and adds User Space autoruns to the Whitelist. Next, the ‘Tools’ window can be closed, and the main window should be visible.
5. Press <Turn OFF All SRP> and next <(Re)Install All SRP> from the main window.
6. Choose restrictions by pressing appropriate buttons in the main window. You can also, press <Turn ON All SRP> and <Turn ON All Restrictions> to activate default Hard_Configurator settings.

* Do not change the path and the name of:
RunAsSmartscreen(x64).exe, RunAsSmartscreen(x86).exe,
RunBySmartscreen(x64).exe, RunBySmartscreen(x86).exe .
The path 'C:\Windows\Hard_Configurator' and above executable filenames are hard-coded.
* Do not run executables RunAsSmartscreen(x64).exe or RunAsSmartscreen(x86).exe, they are invoked by "Run As SmartScreen" option in Explorer context menu.
* Do not run executables RunBySmartscreen(x64).exe or RunBySmartscreen(x86).exe, they are invoked by "Run By SmartScreen" option in Explorer context menu.

DEINSTALLATION

1. Run Hard_Configurator.
2. Press <Turn OFF All SRP>, and next <Turn OFF All Restrictions>.
3. Close Hard_Configurator.
4. Delete Hard_Configurator folder from C:\Windows.

SOFTWARE RESTRICTION POLICIES (SRP)

From the technet.microsoft.com :

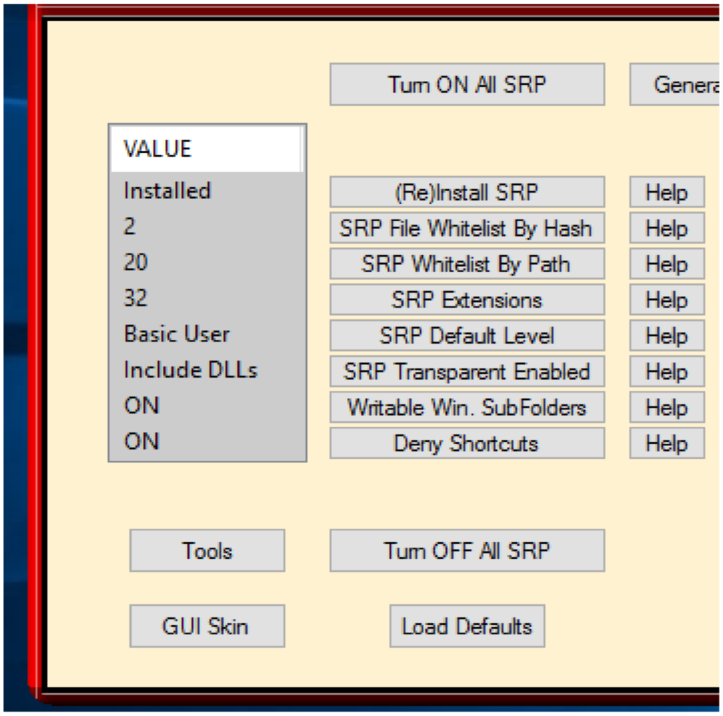
"Software Restriction Policies (SRP) is Group Policy-based feature that identifies software programs running on computers in a domain, and controls the ability of those programs to run. Software restriction policies are part of the Microsoft security and management strategy to assist enterprises in increasing the reliability, integrity, and manageability of their computers.

You can also use software restriction policies to create a highly restricted configuration for computers, in which you allow only specifically identified applications to run. Software restriction policies are integrated with Microsoft Active Directory and Group Policy. You can also create software restriction policies on stand-alone computers. Software restriction policies are trust policies, which are regulations set by an administrator to restrict scripts and other code that is not fully trusted from running."

https://technet.microsoft.com/en-us/library/hh831534(v=ws.11).aspx

Software Restriction Policies are available through Group Policy, for Windows Pro, Windows Enterprise, Windows Server, and Windows Education. Well configured SRP is known in enterprise case studies, as one of the best protections against virus infections.

Hard_Configurator program can apply several SRP settings in Windows Home, too. Some settings were skipped because they are not needed for home users (for example Zone and Certificate Whitelist/Blacklist rules).



VALUE		
Installed	(Re)Install SRP	Help
2	SRP File Whitelist By Hash	Help
20	SRP Whitelist By Path	Help
32	SRP Extensions	Help
Basic User	SRP Default Level	Help
Include DLLs	SRP Transparent Enabled	Help
ON	Writable Win. SubFolders	Help
ON	Deny Shortcuts	Help

<(Re)Install SRP> button makes changes in the Registry to install Windows SRP. The SRP parameters can be changed using the buttons:
<SRP Whitelist By Hash>, <SRP Whitelist By Path>, <SRP Extensions>, <SRP Default Level>, <SRP Transparent Enabled>.
Two SRP options: "Ignore certificate rules" and "All users except local administrators" are hardcoded, and set to ON. Some Disallowed (Blacklist) rules are applied by <Writable Win. Subfolders>, and <Deny Shortcuts> options.
There are no other options in Hard_Configurator to customize Blacklist rules.

In this program, SRP is used as a simple anti-exe, that is based on Windows built-in security features. Executables can be run without SRP restrictions in the System Space, that contains UAC protected folders: 'Windows' and 'Program Files' (and 'Program Files (x86)' in 64Bit versions). Outside those folders (= User Space), executables will be blocked by default, when running by mouse click, pressing the ENTER key or using “Open“/“Open With ...“ from Explorer context menu. The list of protected file extensions (Designated File Types) can be accessed by pressing <SRP Extensions> button.
There is a group of privileged file types, that can be blocked by the SAFER API, even if they are not on the list of protected extensions (see **‘How SRP can control file execution/opening**’). This type of execution control, relates to API functions: CreateProcess, and LoadLibrary, that can call into Safer APIs. Also, the privileged sponsors: cmd.exe, wscript.exe, cscript.exe, and msiexec.exe have such calling ability.
There's a possibility to run executables from User Space, if they were whitelisted by hash or by path.

REMARKS

SRP restrictions can be bypassed using "Run As Administrator" option from Explorer context menu, but this works only for some extensions (for example: BAT, CMD, EXE). Running new files with Administrative Rights can be dangerous for many users, so Hard_Configurator can replace "Run As Administrator" option in Explorer context menu with safer "Run As SmartScreen" (see <Run As SmartScreen> manual section).
Known folder GUIDs were used for whitelisting folders: C:\Windows, C:\Program Files, and C:\Program Files (x86) . Additionally, the program

uses GUIDs based on Simple Software Restriction Policies to handle file whitelisting by hash.
This program cannot uninstall SRP, but that is not really necessary. SRP in Hard_Configurator can be completely deactivated by pressing <Turn OFF All SRP> button.

How SRP can control file execution/opening.

This section is for users that want to understand SRP on the deeper level. It is not necessary when using Hard_Configurator.

1. ShellExecute API function.
   It calls into Safer APIs, while opening files with extensions included in the SRP ‘Designated File Types’ list (the list of protected extensions, <SRP Extensions> button). Then, the software restriction policies will only apply when either Windows Explorer or Internet Explorer is used to open the files. So, if the file extension is on this list, then the file access will be controlled by SRP, while double clicking, pressing ENTER key or choosing “Open“/“Open With ...“ from Explorer context menu, etc. If the file is blocked by SRP, then the program (sponsor), that can manage that extension (for example regedit.exe for the REG file) is not invoked at all. Yet, the file can still be opened from within this program (in Regedit the REG file can be imported) or opened indirectly by command, using the sponsor ('regedit.exe path_to_file.reg').
2. Sponsor's calling into Safer APIs (extended protection). Only some privileged sponsors can do it: **cmd.exe, wscript.exe, cscript.exe,** and **msiexec.exe** - they can manage the files: **BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH,** and **MSI**. This is much safer than blocking files by extension (see point 1.). The file cannot be run, both: using Explorer or IE, and indirectly using the sponsor (for example: 'cmd.exe /c path_to_malicious.bat' will be blocked too).
3. CreateProcess API function (extended protection).
   It calls into Safer APIs, while executing **COM/EXE/SCR** files, and software restriction policies are applied directly to those **COM/EXE/SCR** files. The **COM** and **SCR** files can be protected by both ShellExecute and CreateProcess API functions, if those extensions are added to the list of protected extensions (‘Designated File Types’).

4. LoadLibrary API function (extended protection).
   It calls into Safer APIs, while loading libraries **DLL/OCX**, and software restriction policies are applied directly to those **DLL/OCX** files.
   The **DLL** and **OCX** files can be protected by both ShellExecute and LoadLibrary API functions, if those extensions are added to the list of protected extensions (‘Designated File Types’).
5. The above four ways can be modified by the values of <SRP Default Level> or <SRP Transparent Enabled> settings, and Whitelist/Blacklist rules (by path, hash, wildcards supported). It is worth mentioning, that some file path Whitelist/Blacklist rules can be ignored (accepted) by SRP, depending on <SRP Transparent Enabled> settings.
6. Shortcut settings 1:

- ‘Basic User’ + (protected LNK extension, global Blacklist *.LNK, some LNK locations Whitelisted - so LNK can be run only in those locations).
- ‘White List’ + (protected LNK extension, some LNK locations Whitelisted - so LNK can be run only in those locations).

  Only targets covered by extended protection can be blocked (see TABLE 2.) when using shortcuts.
  If the target is **not under extended protection, it will be opened**, even if direct target opening in Explorer is blocked. So, for example (with the above settings), the REG file can always be opened by whitelisted shortcut.
  All Whitelist/Blacklist rules for the REG file will be ignored.

7. Shortcut settings 2:

- ‘Basic User’ + (protected LNK extension, some locations Whitelisted) without global LNK Blacklist .

  Shortcuts can be run everywhere, even in User Space!
  Shortcuts from System Space behave like in 6.
  Shortcuts from User Space behave in unusual way:
  They will execute any target EXE file, in any location (not good) !
  ‘Designated File Types’ + files under extended protection (except EXE) are blocked in the User Space.
  Target scripts (CMD or Windows Script Host), are also blocked (by extension), as if they were silently added to ‘Designated File Types’.

The default SRP config, assume whitelisting by path the **System Space** = ‘Windows’ + ‘Program Files’ (and ‘Program Files (x86) in 64Bit systems).

The below table shows, what files are blocked by SRP in the **User Space** (= everything outside System Space). The <SRP Default Level> settings are included in the first column, and <SRP Transparent Enabled> settings in the first row.

From **TABLE 1.** (below) we can see, that with 'Basic User' + 'No Enforcement' settings, only MSI files are blocked by SRP in the User Space. On the contrary, with 'White List' + 'Include DLLs' settings the files: **BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR, DLL, OCX,** + all **'Designated File Types'** will be blocked by SRP in the User Space.

TABLE 1.
Files blocked in the USER SPACE

	No Enforcement	Skip DLLs	Include DLLs
Allow All (Unrestricted) Windows Vista+ Extended Protection	all files allowed	all files allowed	all files allowed
Basic User Windows 7+ Extended Protection	MSI	MSI, COM, EXE, SCR	MSI, COM, EXE, SCR, DLL, OCX,
Basic User Windows Vista Extended Protection	MSI	MSI	MSI, DLL, OCX,
White list (Disalowed) Windows Vista+ Extended Protection	BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI,	BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR,	BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR, DLL, OCX,
Blocking by Extension	DISABLED	Designated File Types	Designated File Types

Hard_Configurator default 'Designated File Types' in Windows 7, 8, 8.1, 10: WSC, WS, VB, URL, SHS, SCT, REG, PIF, PCD, **OCX,** MST, MSP, MSC, MDE, MDB, LNK, JAR, ISP, INS, INF, HTA, HLP, **EXE,** CRT, CPL, **COM, CMD,** CHM, **BAT,** BAS, ADP, ADE

If we want to add other Whitelist/Blacklist rules for specific **file types (not folders),** the value of <SRP Default Level> is irrelevant:

TABLE 2 - Valid file types for Whitelist/Blacklist rules.

	No Enforcement	Skip DLLs	Include DLLs
Extended Protection Controlled by Sponsor calling into Safer APIs (cmd, wscript, cscript, msiexec), and API calling (CreateProcess, LoadLibrary)	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI**	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR,**	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR, DLL, OCX,**
Blocking by Extension Controlled by ShellExecute calling into Safer APIs	**DISABLED**	**Designated File Types**	**Designated File Types**

EXAMPLE

From the above, we can see, that with **'No Enforcement'** setting, the Blacklist (Disallowed) file path rules :

c:\Program Files*.reg
c:\Program Files*.scr
c:\Program Files*.ocx
c:\Program Files*.bat
c:\Program Files*.vbs

are only valid for **BAT** and **VBS** files, and they will be applied by calling cmd.exe and wscript.exe into Safer APIs. The rules for REG, **SCR, OCX** files will be ignored with 'No Enforcement' setting.

With Hard_Configurator default **'Designated File Types'** + **'Include DLLs'** settings all that rules are valid. The REG file will be blocked only by extension. The **OCX, BAT, VBS** files will be blocked by extension, and by calling into Safer APIs by: LoadLibrary, or sponsors cmd.exe, wscript.exe. The **SCR** file will be blocked only due to CreateProcess calling into Safer APIs.

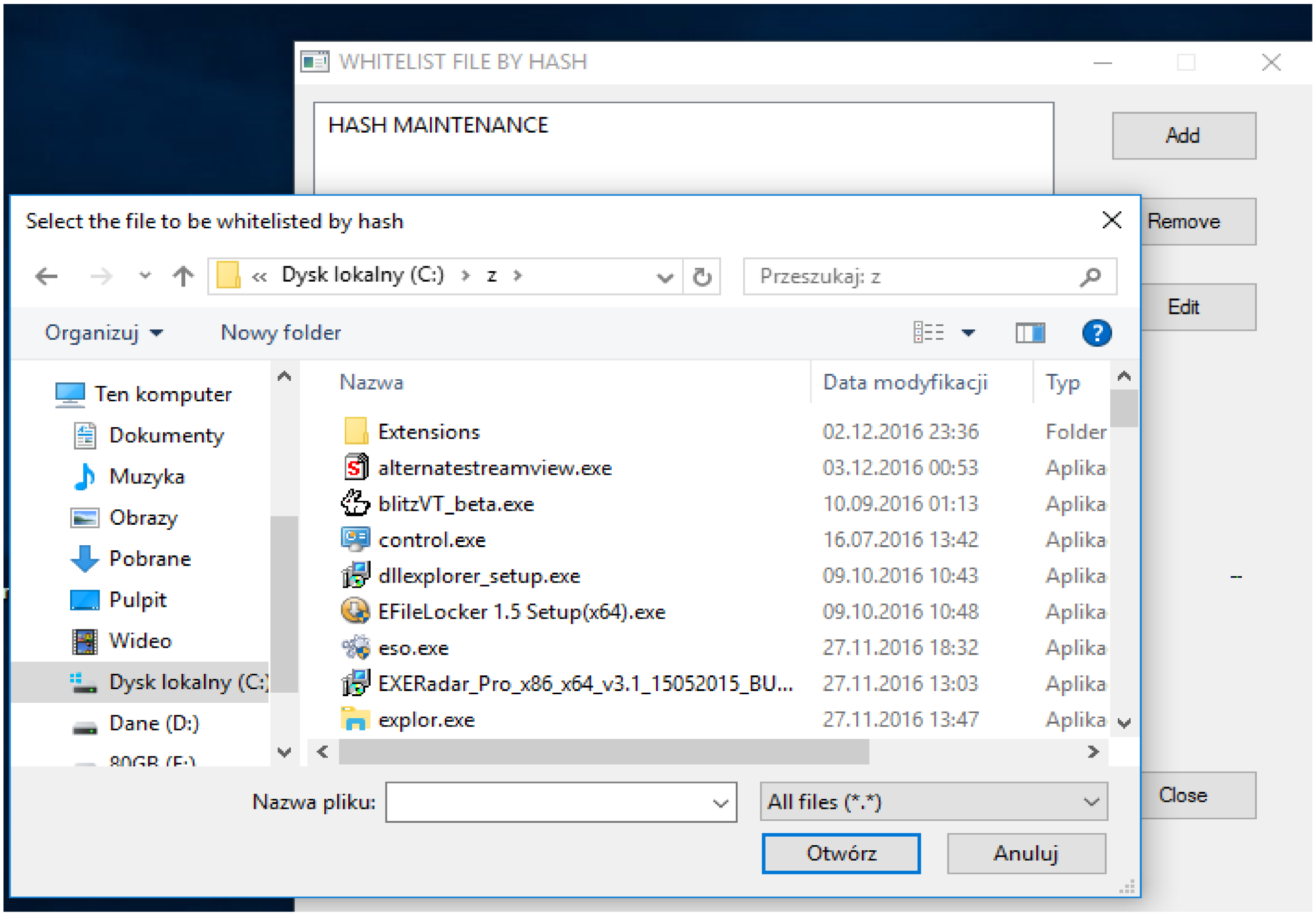
Useful links:

https://technet.microsoft.com/en-us/library/cc786941(v=ws.10).aspx
https://technet.microsoft.com/en-us/library/bb457006.aspx
https://malwaretips.com/threads/windows-pro-owner-use-software-restriction-policies.61871/
http://www.wilderssecurity.com/threads/maximising-windows-7-security-with-srp-under-lua-whatever-the-win7-version.262686/
http://www.bleepingcomputer.com/tutorials/create-an-application-whitelist-policy-in-windows/

Registry changes:

HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers

WHITELISTING BY HASH



Nazwa	Data modyfikacji	Typ
Extensions	02.12.2016 23:36	Folder
alternatestreamview.exe	03.12.2016 00:53	Aplika
blitzVT_beta.exe	10.09.2016 01:13	Aplika
control.exe	16.07.2016 13:42	Aplika
dllexplorer_setup.exe	09.10.2016 10:43	Aplika
EFileLocker 1.5 Setup(x64).exe	09.10.2016 10:48	Aplika
eso.exe	27.11.2016 18:32	Aplika
EXERadar_Pro_x86_x64_v3.1_15052015_BU...	27.11.2016 13:03	Aplika
explor.exe	27.11.2016 13:47	Aplika

<SRP Whitelist By Hash> button opens ADD/REMOVE/EDIT window to manage file whitelisting by hash. It is very useful for running programs located in the User Space (outside the folders: Windows, Program Files, and Program Files (x86)).

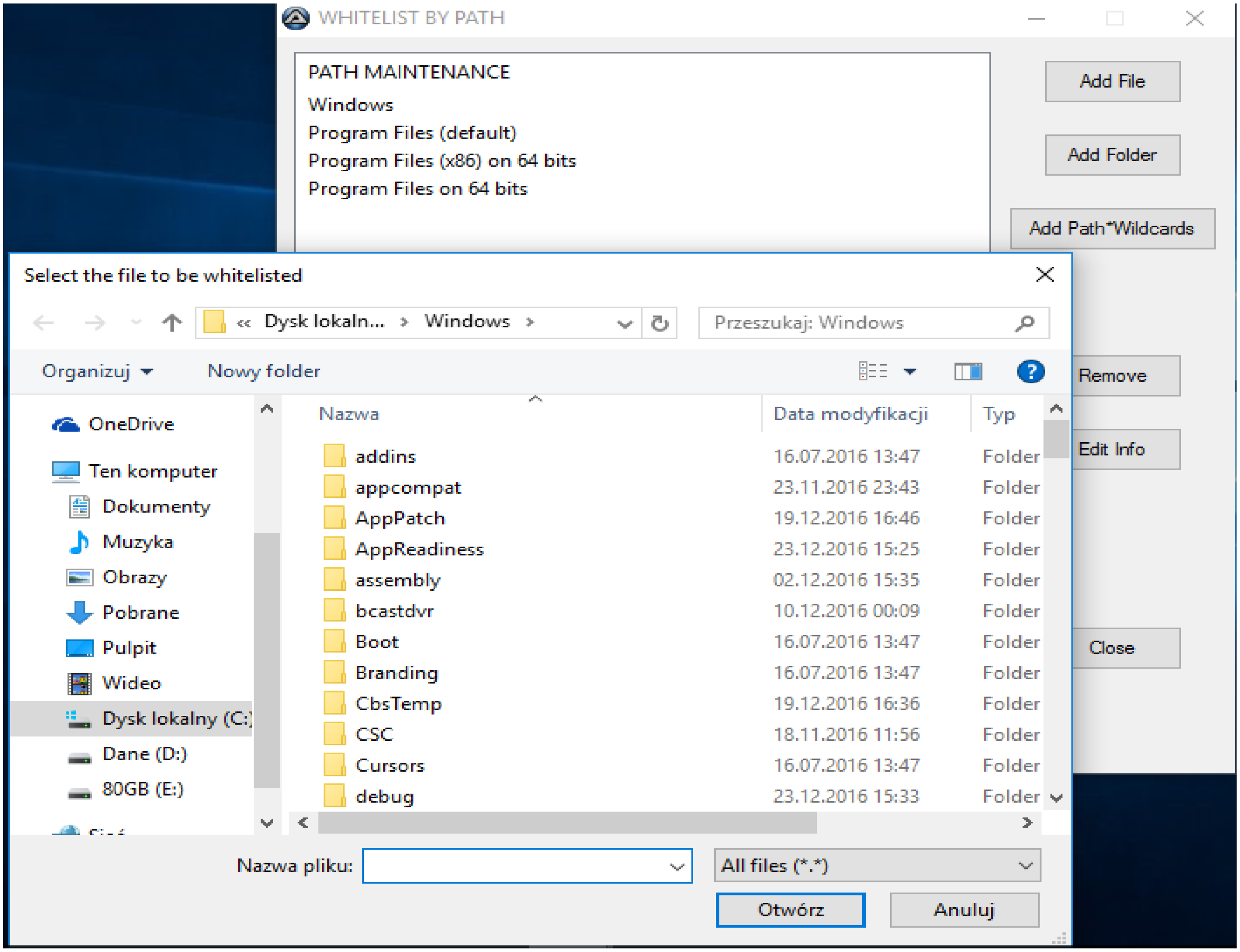
User Space is not protected by UAC, so the file can be silently modified by virus infection. Yet, this also changes the file hash, and then SRP will block file execution.
Managing file hashes is not comfortable. Use this function only if you have to. The program tries to extract some info about the file to make hash entries more readable.

REMARKS
Sometimes programs are wrapped and have to use TEMP folder in the User Profile to execute (most frequently it is ...\AppData\Local\Temp).
Execution in the TEMP folder will be blocked by SRP, so the unwrapped file

should be whitelisted by hash too. Sometimes, such files are quickly deleted, and then utilities similar to "Moo0 FileMonitor" may be needed to find out which files were temporary dropped to TEMP folder. Alternatively, Hard_Configurator has the option: <Run SRP/Scripts EventLogView> in the 'Tools' section. It can use NirSoft FullEventLogView utility to filter and view SRP blocking events. This utility is already included in Hard_Configurator package.

Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Hashes

WHITELISTING BY PATH



PATH MAINTENANCE
Windows
Program Files (default)
Program Files (x86) on 64 bits
Program Files on 64 bits

Nazwa	Data modyfikacji	Typ
addins	16.07.2016 13:47	Folder
appcompat	23.11.2016 23:43	Folder
AppPatch	19.12.2016 16:46	Folder
AppReadiness	23.12.2016 15:25	Folder
assembly	02.12.2016 15:35	Folder
bcastdvr	10.12.2016 00:09	Folder
Boot	16.07.2016 13:47	Folder
Branding	16.07.2016 13:47	Folder
CbsTemp	19.12.2016 16:36	Folder
CSC	18.11.2016 11:56	Folder
Cursors	16.07.2016 13:47	Folder
debug	23.12.2016 15:33	Folder

<SRP Whitelist By Path> button opens ADD/REMOVE/EDIT window to manage file/folder whitelisting by path. It is very useful, for running programs located in User Space (outside the folders: 'Windows', 'Program Files', and 'Program Files (x86)'). Yet, User Space is not protected by UAC, so in theory, the malware file can bypass SRP when running from the whitelisted path.

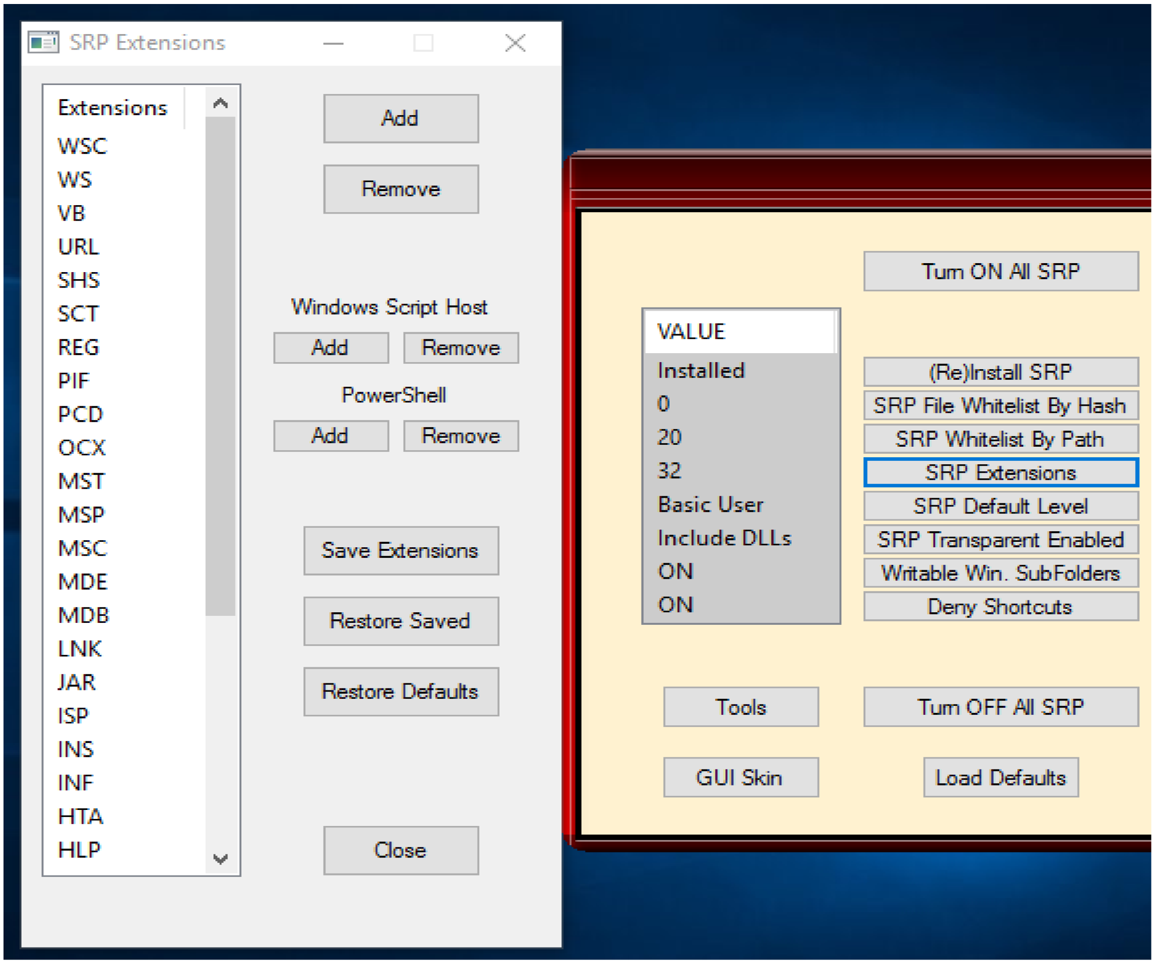
Whitelisting well known locations in User Profile is especially dangerous, for example:
%USERPROFILE%\AppData\Local
%USERPROFILE%\AppData\Local\Temp
Music, Pictures, Videos, Documents, Desktop, Downloads, etc.

The safer method (but less convenient) is whitelisting the file by hash.
Whitelisting the shortcuts or paths with wildcards is only possible with the <Add Path*Wildcards> option.

Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths

PROTECTED EXTENSIONS

<SRP Extensions> button opens ADD/REMOVE window with the list of actually protected extensions. Default extensions (Windows 7, 8, 8.1, 10):
WSC, WS, VB, URL, SHS, SCT, REG, PIF, PCD, OCX, MST, MSP, MSC, MDE, MDB, LNK, JAR, ISP, INS, INF, HTA, HLP, EXE, CRT, CPL, COM, CMD, CHM, BAT, BAS, ADP, ADE.
In Windows Vista, PowerShell extensions are added by default:
PS1, PS2, PSC1, PSC2, PS1XML, PS2XML
because the option <No PowerSell Exec.> is not supported.
The above extensions differ from SRP defaults in Windows Pro. ‘Windows Script Host’ and ‘Powershell script’ extensions were removed, because Hard_Configurator has <Disable Win. Script Host> and <No PowerShell Exec.> options to deal with them. Also, the MSI and SCR extensions were removed to work with <Run As SmartScreen> option (SRP can still protect them by SAFER APIs, even if they are not on the extension list).



Extensions
WSC
WS
VB
URL
SHS
SCT
REG
PIF
PCD
OCX
MST
MSP
MSC
MDE
MDB
LNK
JAR
ISP
INS
INF
HTA
HLP

VALUE
Installed
0
20
32
Basic User
Include DLLs
ON
ON

You can customize this list using <Add> and <Remove> buttons. When using no default list, it is good to save it (<Save Extensions>). The list can then be restored after Hard_Configurator updates (<Restore Saved>).

Warnings.
Do not add MSI and SCR extensions if <Run As SmartScreen> is set to ‘ON’.
Do not add JS, JSE, VBE, VBS, WSF, and WSH extensions, if the option <Disable Win. Script Host> is set to ‘ON’.
Do not add PS1, PS2, PSC1, PSC2, PS1XML, and PS2XML extensions, if <No PowerShell Exec.> is set to ‘ON’.

REMARKS
Some executable files are protected by SAFER APIs even if the list is empty.
Windows Script Host protection depends also on the registry value:

HKLM\SOFTWARE\Microsoft\Windows Script Host\Settings
UseWINSAFER = 1 (Windows default value)
and for 64Bit system, the same in the key:
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Script Host\Settings

See also: **SOFTWARE RESTRICTION POLICIES (SRP) / How SRP can control file execution/opening.**

Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers!ExecutableTypes

SECURITY LEVELS

<SRP Default Level> button changes the security levels between:
'Basic User' ---> 'Allow All' ---> 'White List'

'White List' corresponds to default deny (Disallowed) security level. With <SRP Transparent Enabled> option set to 'Include DLLs', it can apply in the User Space:
* protection to all files included in 'Designated File Types' list
* extended security to Windows native executables (COM, EXE, SCR), binary libraries (DLL, OCX), scripts (BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH), and MSI installers.

'Basic User' (in Windows 7+) is very similar to 'White List', but it differently supports LNK, MSI, and script files (see help for <SRP Extensions> button). This is default setting in Hard_Configurator, except Windows Vista, where 'Basic User' setting allows to run EXE files, so <SRP Default Level> is changed there to 'White List'.

'Allow All' turns off SRP Default Deny Protection. Users can still use some Whitelist/Blacklist rules.
See also: **SOFTWARE RESTRICTION POLICIES (SRP) / How SRP can control file execution/opening.**

If you want to run executable file in the User Space with SRP set to 'Basic User' or 'White List', then it can be done with "Run As Administrator" option in Explorer context menu. But, bypassing SRP with Administrative Rights can be dangerous. Hard_Configurator provides the safer option by replacing "Run As Administrator" with "Run As SmartScreen".
If you want to use frequently the application that is located in the User Space, then consider whitelisting it by hash.

REMARKS

In theory, you can "Run As SmartScreen" the files: BAT, CMD, COM, EXE, MSI, SCR, JSE, and VBE. Yet, in default settings (blocked scripts), only COM, EXE, MSI, and SCR can be executed. If you want to bypass both SRP and forced SmartScreen, then run ‘Total Commander’ with "Run As SmartScreen", and then use it to execute files. If so, all child processes will be run automatically with Administrative Rights.

Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers
DefaultLevel

Value (Dword)	
0	'White List (Disallowed)'
131072	'Basic User' (131072 = 20000 hex)
262144	'Allow All' (262144 = 40000 hex)

ENFORCEMENT OPTIONS

<SRP Transparent Enabled> button changes the Enforcement options :
'Skip DLLs' -> 'Include DLLs' -> 'No Enforcement'

'Skip DLLs' can control file execution by extension (Designated File Types), provides extended protection for scripts (BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH), MSI installers, and native Windows executables (COM, EXE, SCR).
This setting is default in Hard_Configurator, because it is more usable for the average users.

'Include DLLs' setting, additionally turns on the extended protection of binary libraries (DLL, OCX), due to LoadLibrary API function.
This option can protect from many DLL attacks, when system files are used to load malicious libraries:

```
PATH\InstallUtil.exe /logfile= /LogToConsole=false /U pathToDLL
PATH\regsvcs.exe pathToDLL
PATH\regasm.exe /U pathToDLL
regsvr32 /s  /u pathToDLL
regsvr32 /s pathToDLL
rundll32 pathToDLL,EntryPoint
```

where ‘PATH’ is the path to Microsoft Net Framework, for example:
“C:\Windows\Microsoft.NET\Framework64\v4.0.30319“
and ‘pathToDLL’ is the full path to the DLL file in the User Space.
'Include DLLs' does not protect from sophisticated DLL attacks initiated by exploits, for example ReflectiveDLLInjection.
This setting is the most restrictive one, and may sometimes result in performance degradation.
Before using 'Include DLLs' setting, the user should first analyze autoruns in the User Space (see <Tools> button). Those autoruns and related DLLs (in the User Space), should be whitelisted to avoid startup problems.
With 'Include DLLs' setting, you may consider also, turning on Advanced SRP Logging from <Tools> menu.

'No Enforcement' option turns off blocking by extension (Designated File Types), and disables extended protection of binary libraries (DLL, OCX). The extended protection for BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, and MSI files is still active - the sponsors: cmd.exe, cscript.exe, wscript.exe, and msiexec.exe, can call into Safer APIs, controlling the execution of sponsored scripts. File blocking can be applied, when using Whitelist\Blacklist file path rules.
See also: **SOFTWARE RESTRICTION POLICIES (SRP) / How SRP can control file execution/opening.**

Registry changes

HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers
TransparentEnabled
Value(Dword)

0	No Enforcement
1	skip DLLs
2	include DLLs

WRITABLE ‘C:\WINDOWS’ SUBFOLDERS

Setting **<Writable Win. SubFolders>** to 'ON' denies the execution from ‘C:\Windows’ subfolders, that are writable (no UAC protection).
This protection uses SRP blacklist, so it denies the execution even
if SRP security level is set to 'Allow All'.
The execution is allowed, for programs started with Administrative Rights independent of SRP security level.

For Windows 8.1 and prior versions, the below subfolders are added to SRP blacklist:
c:\windows\debug\WIA
c:\windows\Registration\CRMLog
c:\windows\System32\catroot2\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}
c:\windows\System32\com\dmp
c:\windows\System32\FxsTmp
c:\windows\System32\spool\drivers\color
c:\windows\System32\spool\PRINTERS
c:\windows\System32\Tasks
c:\windows\SysWOW64\com\dmp
c:\windows\SysWOW64\FxsTmp
c:\windows\SysWOW64\Tasks
c:\windows\Tasks
c:\windows\Temp
c:\windows\tracing

For Windows 10 the below subfolders are added to SRP blacklist:
c:\windows\servicing\Packages
c:\windows\servicing\Sessions
c:\windows\System32\Microsoft\Crypto\RSA\MachineKeys
c:\windows\System32\spool\drivers\color
c:\windows\System32\Tasks
c:\windows\SysWOW64\Tasks
c:\windows\Tasks
c:\windows\Temp

Registry changes:
[HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\0\Paths\

EXECUTING SHORTCUTS

<Deny Shortcuts> button disables/enables shortcut execution restrictions.
If this option is set to 'ON', then shortcuts can be executed only in 'Windows', 'Program Files', 'Program Files (x86)', 'Desktop', 'Power Menu', and 'Start Menu' locations.
This restriction is applied, because specially crafted shortcuts can bypass Software Restriction Policies.
<Deny Shortcuts> is suited to work with SRP 'Basic User' security level.
If the security level is changed to 'White List' the LNK extension should be removed from SRP Extensions - if not, shortcuts in the User Space will be blocked (counterintuitively), when <Deny Shortcuts> is set to OFF!

Registry changes:
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\262144\Paths\

Added GUIDs for whitelisted locations:
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC19}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC20}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC21}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC22}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC23}]

{625B53C3-AB48-4EC1-BA1F-A1EF4146FC24}]
{99a0fd77-ed0c-4e30-91ff-9d51428d2f21}]
{99a0fd77-ed0c-4e30-91ff-9d51428d2f22}]
{99a0fd77-ed0c-4e30-91ff-9d51428d2f23}]
{B4BFCC3A-DB2C-424C-B029-7FE99A87C641}]
{B4BFCC3A-DB2C-424C-B029-7FE99A87C642}]
{B4BFCC3A-DB2C-424C-B029-7FE99A87C643}]
{B4BFCC3A-DB2C-424C-B029-7FE99A87C644}]
{B4BFCC3A-DB2C-424C-B029-7FE99A87C645}]

Added Guid for blacklisted *.lnk:
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\0\Paths\{1016bbe0-a716-428b-822e-5E544B6A3301}

EXECUTION FROM REMOVABLE DISKS

<No Removable Disk Exec.> button disables/enables file execution from removable disks (Pendrives, USB disks, Memory Cards). The files cannot be executed by mouse click, pressing Enter key, or using "Run As Administrator" from Explorer context menu. Yet, if the <Run As SmartScreen> is set to 'Administrator' or 'Standard User', then this restriction can be bypassed using Explorer context menu options: "Run As SmartScreen" or "Run by SmartScreen". If additionally SRP is activated, only "Run As SmartScreen" from Explorer context menu will work, because "Run By SmartScreen" cannot automatically elevate file execution to bypass SRP.

Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\RemovableStorageDevices\
{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}

Deny_Execute
Value (Dword)

1	block execution
0	allow execution

POWERSHELL SCRIPTS

<No PowerShell Exec.> button disables/enables PowerShell script execution.
If this option is ON, then script execution is blocked, but you can still execute script commands in the Powershell window. Keep this option ON because scripts are the weak point of most antimalware programs. Alternatively, you can activate SRP add PowerShell script extensions.

In Windows 64Bit there are two PowerShell Hosts (32Bit and 64Bit), but both are disabled/enabled by the below key:

Registry changes:
HKLM\Software\Policies\Microsoft\Windows\PowerShell

EnableScripts
Value (Dword)

0	script execution is disabled
1	script execution is enabled

PUA PROTECTION

<Defender PUA Protection> button activates/deactivates Windows Defender PUA protection.
"By default, PUA protection quarantines the file so they won’t run. PUA will be blocked only at download or install-time. A file will be included for blocking if it meets one of the following conditions:
* The file is being scanned from the browser
* The file has Mark of the Web set
* The file is in the %downloads% folder
* Or if the file in the %temp% folder "
https://blogs.technet.microsoft.com/mmpc/2015/11/25/shields-up-on-potentially-unwanted-applications-in-your-enterprise/

REMARKS
PUA = Potentially Unwanted Application ~ PUP ~ PUS
PUP = Potentially Unwanted Program
PUS = Potentially Unwanted Software

"A potentially unwanted program is bundled software which computer users are fooled into installing along with a wanted program.
Such software can compromise privacy or weaken the computer's security. Companies often bundle a wanted program download with a wrapper application. This may install an unwanted application, without providing a clear opt-out method.[1][2] Unwanted programs often include no sign that they are installed, and no uninstall or opt-out instructions.[3]
Antivirus companies define the software bundled as potentially unwanted programs (PUP)[3][4] which can include software that displays intrusive advertising, or tracks the user's Internet usage to sell information to advertisers, injects its own advertising into web pages that a user looks at, or uses premium SMS services to rack up charges for the user.[5][6] The practice is widely considered unethical because it violates the security interests of users without their informed consent.
Some unwanted software bundles install a root certificate on a user's device, which allows hackers to intercept private data such as banking details, without a browser giving security warnings."
https://en.wikipedia.org/wiki/Potentially_unwanted_program

Registry changes:
HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine

MpEnablePus

Value (DWORD)

0	Potentially Unwanted Application protection is disabled.
1	Potentially Unwanted Application protection is enabled.

WINDOWS SCRIPT HOST

<Disable Win. Script Host> button disables/enables Windows Script Host.
If this option is ON, then execution of JS, JSE, VBS, VBE, WSF, and WSH scripts is blocked. Keep this option ON because scripts are the weak point of most antimalware programs. Some scripts can be executed at the boot time, for example:

c:\windows\system32\gathernetworkinfo.vbs
c:\windows\syswow64\gathernetworkinfo.vbs
c:\windows\system32\gatherwiredinfo.vbs
c:\windows\syswow64\gatherwiredinfo.vbs

c:\windows\system32\gatherwirelessinfo.vbs
c:\windows\syswow64\gatherwirelessinfo.vbs

The above scripts are not essential for the Windows system, so can be blocked.

Alternatively, you can activate SRP to block script extensions. But, then Windows Script Host extensions must be added to SRP, because they are omitted in default settings.
In Windows 64Bit there are two Windows Script Hosts (32Bit and 64Bit).
Registry changes:

HKLM\SOFTWARE\Microsoft\Windows Script Host\Settings

Enabled
Value (Dword)

0	script execution is disabled
1	script execution is enabled

HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Script Host\Settings
Enabled
Value (Dword)

0	script execution is disabled
1	script execution is enabled

RUN AS ADMINISTRATOR

<Hide 'Run As Administrator'> button hides/shows "Run As Administrator" option in Explorer context menu. It is useful when you choose to replace this option by "Run As SmartScreen".
Set <Hide 'Run As Administrator'> to "ON" if <Run As SmartScreen> is set to 'Administrator'.
Otherwise, it is better to turn <Hide 'Run As Administrator'> 'OFF'.

REMARKS
When <Hide 'Run As Administrator'> is set to 'ON', then "Command Prompt (Administrator)" option in Windows Power Menu, and "Run As Administra-

tor" option in the Search context menu, are hidden too. Furthermore, files with extensions: BAT, CMD, CPL, and MSC will not run from the User Space (= outside 'Windows', 'Program Files', and 'Program Files (x86)' folders), when default SRP protection is active. "Run As SmartScreen" cannot replace functionality of "Run As Administrator" in this case, because access to BAT, CMD, CPL, and MSC extensions is blocked by SRP default settings. It should not be a problem, since files with above extensions are mostly run from System Space (= inside 'Windows', 'Program Files', and 'Program Files (x86)' folders), and their location in the User Space is very suspicious.

Registry changes:
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer

HideRunAsVerb

Value (Dword)

0	"Run As Administrator" is not hided
1	"Run As Administrator" is hided

RUN AS SMARTSCREEN

<Run As SmartScreen> button adds/removes "Run As SmartScreen" or "Run By SmartScreen" option in Explorer context menu. Those options force file execution with SmartScreen check for files located in the User Space. If the file is located in the System Space (inside 'Windows', 'Program Files', 'Program Files (x86)' folders), then SmartScreen check is skipped.
Pressing <Run As SmartScreen> button changes between values:
'Administrator' -> 'Standared User' -> 'OFF'

The setting 'Administrator' corresponds to "Run As SmartScreen" option in Explorer context menu.
The setting 'Standard User' corresponds to "Run By SmartScreen" option in Explorer context menu.
The setting 'OFF' removes each of the above options from Explorer context menu.

Keep the value 'Administrator' when SRP is activated. If so, the users can safely:

1. Run programs (with a mouse click or pressing ENTER button) already installed in the System Space.
2. Open the media files, documents, etc., that are not on the list of SRP protected files.
3. Install new programs from the User Space using "Run As SmartScreen" option in Explorer context menu.

If SRP is activated (with program defaults) and <Disable Win. Script Host> option is 'ON', then only DLL, EXE, MSI, and SCR files are allowed to "Run As SmartScreen" in the User Space. Other files, located in the User Space, will not be allowed to execute.
Keep the 'Standard User' setting when SRP is deactivated and set <Hide 'Run As Administrator> to 'OFF'. “Run By SmartScreen“ option in Explorer context menu does not automatically elevate the Rights of the executed program.

REMARKS

The SmartScreen Filter in Windows 8+ allows some vectors of infection listed below:
A) You have got the executable file (BAT, CMD, COM, CPL, DLL, EXE, JSE, MSI, OCX, PIF, SCR and VBE) using:
* the downloader or torrent application (EagleGet, utorrent etc.);
* container format file (zip, 7z, arj, rar, etc.);
* CD/DVD/Blue-ray disc;
* CD/DVD/Blue-ray disc image (iso, bin, etc.);
* non NTFS USB storage device (FAT32 pendrive, FAT32 usb disk);
* Memory Card;
so the file does not have the proper Alternate Data Stream attached.
B) You have run the executable file with runas.exe (Microsoft), AdvancedRun (Nirsoft), RunAsSystem.exe (AprelTech.com), etc.

<Run As SmartScreen> covers all vectors of infection listed in the A) point.

Alternatively to "Run As SmartScreen", you can simply upload the file to One Drive (or mailbox) , and download it again. This procedure also activates SmartScreen check automatically. So, if you are executing files, downloaded on NTFS hard drive by most popular Internet Browsers, Windows Store or from One Drive, then the SmartScreen Filter gives you very good protection against malware files (especially 0-day), even without "Run As SmartScreen".

Registry changes:
HKEY_CLASSES_ROOT*\shell\Run As SmartScreen\

REMOTE ACCESS

<Block Remote Access> button disables/enables:
* Remote Assistance
* Remote Shell Access
* Remote Registry Access

For home users, it is recommended to keep this setting ‘ON’.
Remote connections are frequently exploited by malware and hackers.

REMARKS
If this setting is ‘ON’, then local user cannot request remote assistance from a friend or a support professional. Also, Unsolicited Remote Assistance is blocked. Computer management using Remote Shell or Remote Registry is disabled.

“Note that print spooler and directory services replication require access through the remote registry service for certain functions to work properly. Other custom applications may also depend on remote registry access.“
http://www.blackviper.com/windows-services/remote-registry/

Registry changes:
If the setting <Block Remote Access> is set to ON:
HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services
"fAllowUnsolicited" = dword:00000000
"fAllowToGetHelp" = dword:00000000

"fDenyTSConnections" = dword:00000001
HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\WinRS
"AllowRemoteShellAccess"=dword:00000000
[HKLM\SYSTEM\CurrentControlSet\Services\RemoteRegistry]
"Start"=dword:00000004

If "Block Remote Access" is set to OFF the keys values are changed to:
"fAllowUnsolicited" = 00000001
"fAllowToGetHelp" =00000001
"fDenyTSConnections" =00000000
"AllowRemoteShellAccess"=dword:00000000
The Remote Registry setting does not change Windows 8+ (default value)
"Start=dword:00000004"
but in Windows 7 and Vista it will be changed to "Start=dword:00000003".

UNTRUSTED FONTS

<Disable Untrusted Fonts> button activates/deactivates the Blocking Untrusted Fonts feature in Windows 10.
Blocking untrusted fonts help prevent attacks that can happen during the font file-parsing process.

"This security feature provides a global setting to prevent programs from loading untrusted fonts. Untrusted fonts are any font installed outside of the %windir%\Fonts directory. This feature can be configured to be in 3 modes: On Off and Audit. By default it is Off and no fonts are blocked. If you aren't quite ready to deploy this feature into your organization you can run it in Audit mode to see if blocking untrusted fonts causes any usability or compatibility issues."
http://winintro.com/Category=Windows_10_2016&Policy=Microsoft.Policies.GroupPolicy
::FontMitigation&Language=en-en

REMARKS

"Potential reductions in functionality
After you turn this feature on, your employees might experience reduced functionality when:

Sending a print job to a remote printer server that uses this feature and where the spooler process hasn’t been specifically excluded. In this situation, any fonts that aren’t already available in the server’s %windir%/Fonts folder won’t be used.

Printing using fonts provided by the installed printer’s graphics .dll file, outside of the %windir%/Fonts folder. For more information, see Introduction to Printer Graphics DLLs.

Using first or third-party apps that use memory-based fonts.

Using Internet Explorer to look at websites that use embedded fonts. In this situation, the feature blocks the embedded font, causing the website to use a default font. However, not all fonts have all of the characters, so the website might render differently.

Using desktop Office to look at documents with embedded fonts. In this situation, content shows up using a default font picked by Office."
https://technet.microsoft.com/en-us/itpro/windows/keep-secure/block-untrusted-fonts-in-enterprise

Registry changes
HKLM\SOFTWARE\Policies\Microsoft\Windows NT\MitigationOpions

MitigationOptions_FontBocking

Value (REG_SZ)	
1000000000000	Enable (block untrusted fonts)
2000000000000	Disable (do not block untrusted fonts)
3000000000000	Audit (log events without blocking untrusted fonts)

TROUBLESHOOTING

Hard_Configurator troubleshooting.

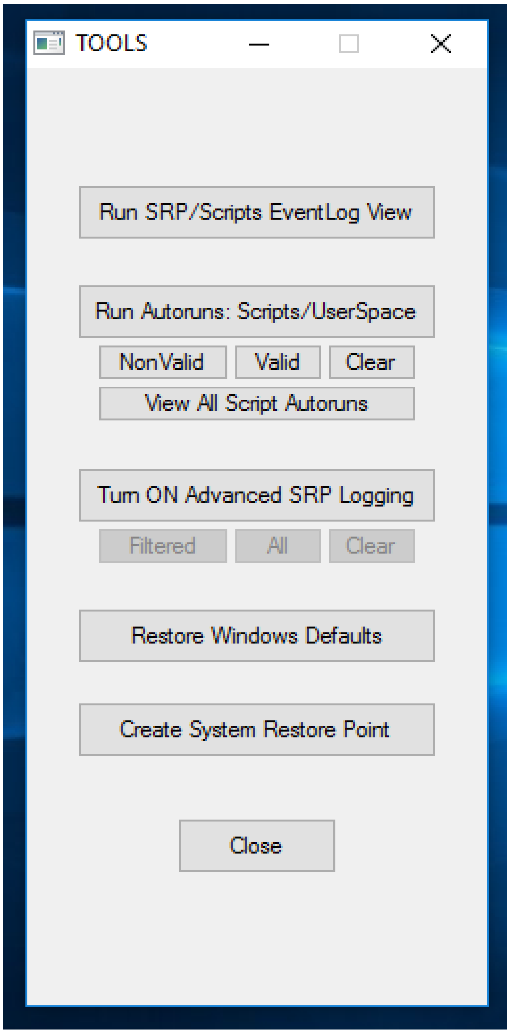
1. If the system hangs after reboot, then it can be a sign, that SRP or one of program restrictions has blocked something important from loading at the boot time.
2. The simplest method to solve this problem is using one of system restore points.
3. Another solution is booting into Safe Mode or using bootable media to access the Command Prompt, and then editing the Registry offline. In most cases the problem would be with SRP, so one must edit the key:
   HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers!DefaultLevel
   change **hex** value to **40000**

4. If the above did not help, then it is possible to edit or remove any registry changes made by Hard_Configurator. The Registry keys altered by the program are enumerated in this manual at the end of each paragraph.

Using TOOLS button.

Pressing this button allows some tools, that can help to prevent blocking important processes in the User Space.

<Run SRP/Scripts EventLogView>
When the program/script is blocked by Hard_Configurator, the information is written in the Windows Event Log.
This option filters the output of NirSoft tool: FullEventLogView to retrieve information about blocked events. The config file uses events ID: 865, 866, 867, 868, 882, 1000, 1007, and 1008 (see below):


TOOLS
Run SRP/Scripts EventLog View
Run Autoruns: Scripts/UserSpace
NonValid
Valid
Clear
View All Script Autoruns
Turn ON Advanced SRP Logging
Filtered
All
Clear
Restore Windows Defaults
Create System Restore Point
Close

SRP related, provider: Microsoft-Windows-SoftwareRestrictionPolicies
* Blocked EXE file
865 -> restricted by policy level
866 -> restricted by path rule
867 -> restricted by certificate rule
868 -> restricted by hash or zone rule
882 -> other

* Blocked MSI file
1007 provider: MsiInstaller
1008 provider: MsiInstaller

* No SRP related
1000 -> provider: Windows Script Host, only when scripts were run with
Administrative Rights
4100 -> provider: Microsoft-Windows-PowerShell

<Run Autoruns: Scripts/UserSpace>

Some processes can be loaded at the boot time from the User Space (= outside 'Windows', 'Program Files', 'Program Files (x86)'). They should be whitelisted by path in SRP to load properly. Sysinternals Autorunsc command line utility allows to find the paths of those processes. This is very useful, because stopping something important from loading at the boot time may hang the system. 'Run Autoruns: Scripts/UserSpace' option, can filter out all numerous autoruns from the System Space leaving only a few entries from the User Space. They can be seen when pressing 'Valid' button. Rarely, the autoruns can have complicated structure, and the filtering algorithm may give up. Those entries should be checked manually - they can be seen when pressing 'NonValid' button.
Pressing 'View All Script Autoruns' shows all scripts (from System and User Space). This option is necessary when we want to disable Windows Script Host and PowerShell.

<Turn ON Advanced SRP logging> (Verbose trace logging of SRP).

'SRP/Scripts EventLogView' option can handle EXE, MSI, and script files, but sometimes the information about DLLs is needed. 'Advanced SRP logging' option activates Verbose trace logging of SRP by changing the Regi-

stry:
HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers
LogFileName
Value REG_SZ
c:\Windows\Hard_Configurator\SRP.log

‘Advanced SRP logging’ option puts info about processes, that **were run with Administrative Rigthts**, to the file SRP.log. Yet, this log has usually many entries from the System Space, so some filtering is required. The ‘Filtered’ button checks SRP.log and leaves only entries related to scripts or processes that were run from the User Space.
This can be used to identify the problems with blocked DLLs, when <SRP Transparent Enabled> is set to ‘Include DLLs’. Simply, run the blocked application with "Run As Administrator" or "Run As SmartScreen" (bypassing SRP), and then look which DLLs are in the log - those DLLs should be whitelisted too. For example, if ‘EagleGet Downloader’ application is installed in the folder: D:\Portable\EagleGet_\
then after "Run As Administrator" the log shows some User Space entries:

```
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\util.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\CrashRpt.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\libcurl.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\sqlite3.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\zlib.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\SSLEAY32.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\LIBEAY32.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\ssl.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\sslQuery.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified \??\D:\Portable\EagleGet_\dl.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EagleGet.exe (PID = 4704) identified D:\Portable\EagleGet_\EGMonitor.exe as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EGMonitor.exe (PID = 5240) identified \??\D:\Portable\EagleGet_\util.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EGMonitor.exe (PID = 5240) identified \??\D:\Portable\EagleGet_\sqlite3.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
EGMonitor.exe (PID = 5240) identified \??\D:\Portable\EagleGet_\dl.dll as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}
```

All the above DLLs and the file EGMonitor.exe must be whitelisted.

Another example, when NoVirusThanks ‘dllexplorer_setup.exe’ is "Run As SmartScreen", then the entries in the log will look like:

```
dllexplorer_setup.exe (PID = 5236) identified C:\Users\Admin\AppData\Local\Temp\is-PPQV9.tmp\dllexplorer_setup.tmp as Unrestricted using default rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}"
```

So, we know that dllexplorer_setup.exe is using dllexplorer_setup.tmp to execute in temporary folder ‘C:\Users\USERNAME\AppData\Local\Temp\is-ASDAD.tmp\’.
Now, dllexplorer_setup.tmp can be whitelisted, and the program can be run normally.

<Restore Windows Defaults>
This option allows restoring all Windows Registry keys, that can be changed by Hard_Configurator, to default values. Those values are mostly the same, as before installation of Hard_Configurator program, except when programs that utilize SRP (Crypto Prevent, SBGuard, etc.) were installed or the user tweaked himself the Registry.