

# Hard\_Configurator - Manual

Version 4.0.1.0 (May 2019)

Copyright: Andy Ful  
Developer Web Page: [https://github.com/AndyFul/Hard\\_Configurator](https://github.com/AndyFul/Hard_Configurator)

Malwaretips forum thread:  
[https://malwaretips.com/threads/hard\\_configurator-windows-hardening-configurator.66416/](https://malwaretips.com/threads/hard_configurator-windows-hardening-configurator.66416/)

## Distribution

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

## Disclaimer of Warranty

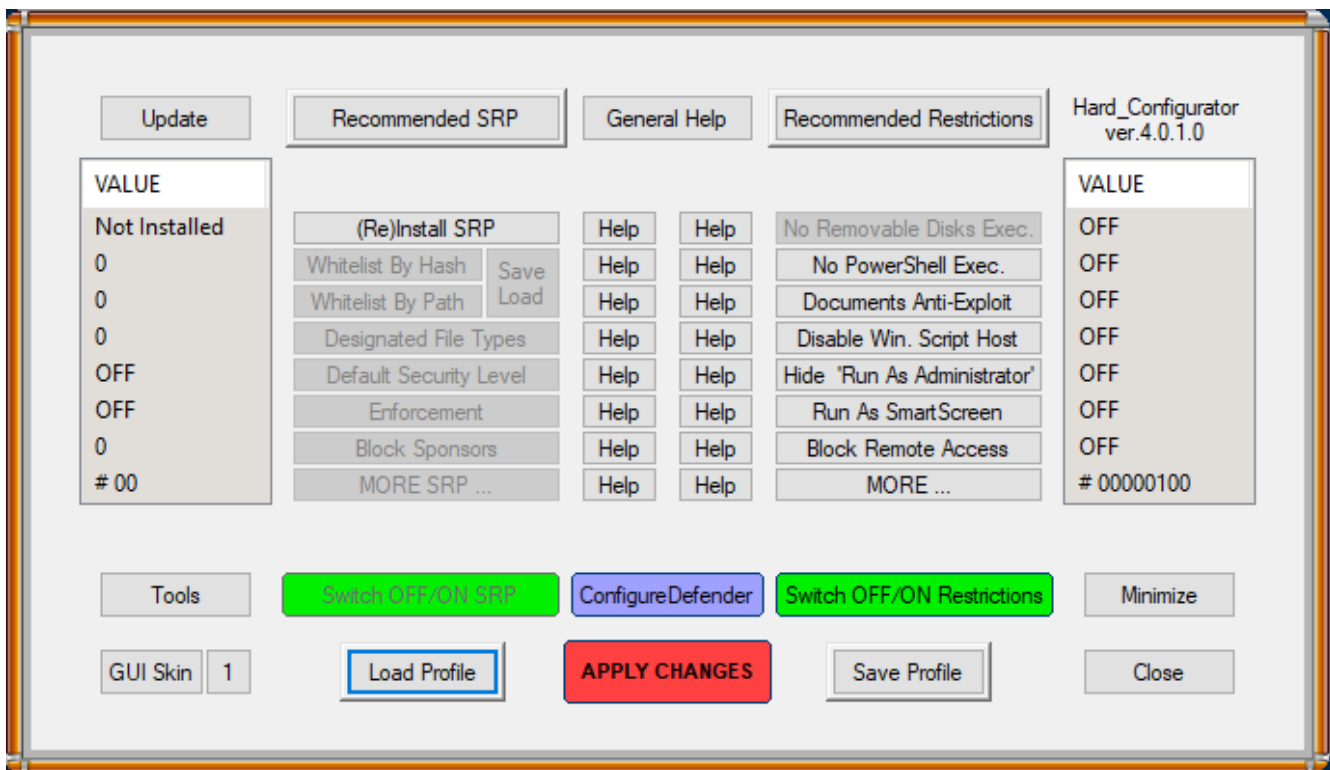
THIS SOFTWARE IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE IT AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING THIS SOFTWARE.

## **TABLE OF CONTENTS**

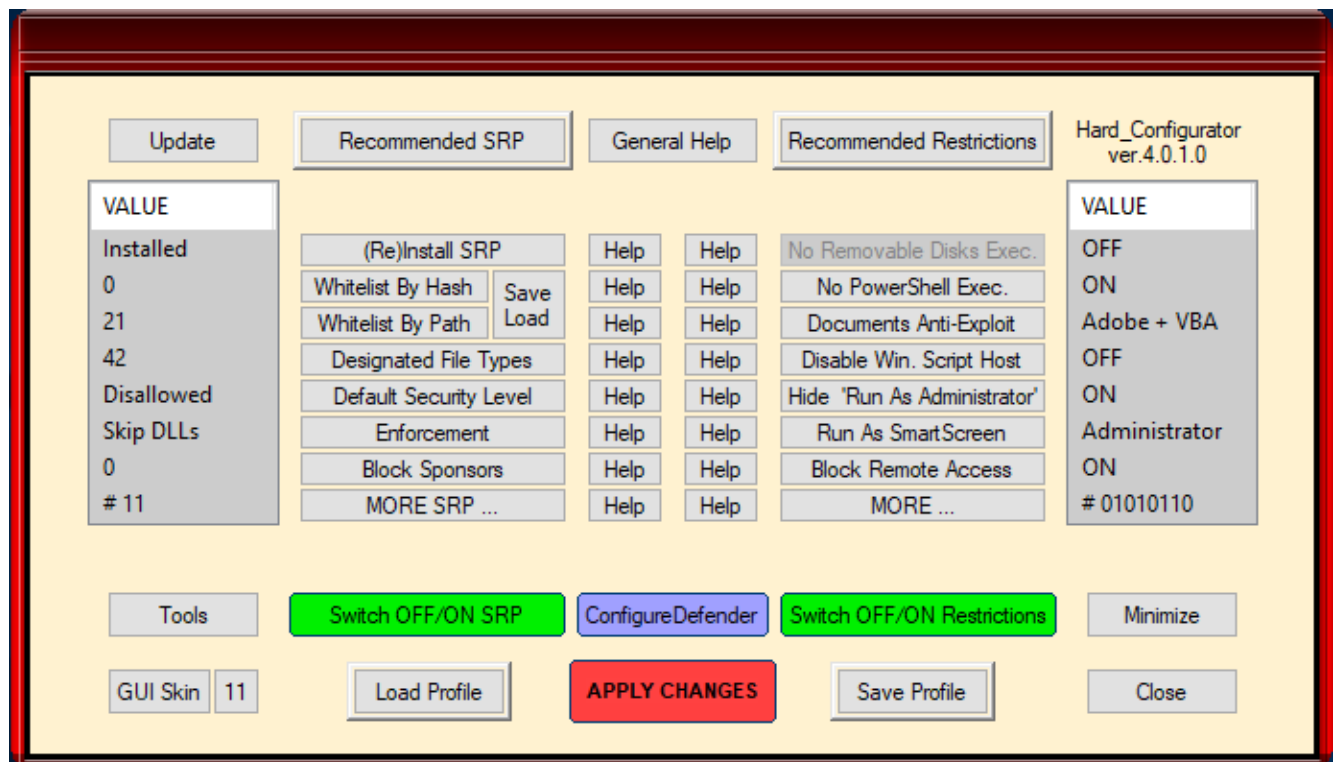
INTRODUCTION .....	3
INSTALLATION / DEINSTALLATION .....	6
SOFTWARE RESTRICTION POLICIES (SRP) .....	8
HOW SRP CAN CONTROL FILE EXECUTION/OPENING..	10
WHITELISTING BY HASH .....	15
WHITELISTING BY PATH .....	16
WHITELIST PROFILES .....	17
DESIGNATED FILE TYPES .....	19
DEFAULT SECURITY LEVELS .....	20
ENFORCEMENT .....	21
BLOCKING SPONSORS .....	23
PROTECTING 'WINDOWS' FOLDER .....	25
PROTECTING SHORTCUTS .....	26
EXECUTION FROM REMOVABLE DISKS .....	27
POWERSHELL SCRIPTS.....	28
WINDOWS SCRIPT HOST.....	29
DOCUMENTS ANTI-EXPLOIT .....	30
SWITCH DEFAULT DENY <DOCUMENTS ANTI-EXPLOIT> ..	33
RUN AS ADMINISTRATOR.....	34
RUN AS SMARTSCREEN.....	35
REMOTE ACCESS .....	39
UNTRUSTED FONTS .....	41
16-BIT APPLICATIONS .....	41
SECURING SHELL EXTENSIONS .....	41
PROGRAM ELEVATION ON SUA .....	42
ELEVATION OF MSI FILES .....	43
DISABLING SMB PROTOCOL 1.0, 2.0, 3.0 .....	44
CACHED LOGONS .....	45
ENABLING SECURE CREDENTIAL PROMPTING .....	46
CONFIGURING WINDOWS DEFENDER .....	47
TROUBLESHOOTING (TOOLS) .....	53
FREQUENTLY ASKED QUESTIONS .....	60

## INTRODUCTION

1. Hard\_Configurator works on Windows Vista and higher versions. It is intended for users, who want to apply some extended features of Windows built-in security. This program can manage Windows built-in Software Restriction Policies (SRP), advanced Windows Defender configuration, hardening of MS Office & Adobe Reader, and some well-known system restrictions to harden the Windows OS. Hard\_Configurator can be seen as a medium integrity level default-deny (via SRP) & Application Reputation Service (forced SmartScreen on Windows 8+) & Windows hardening (restricting vulnerable features).
2. The actual status of all restrictions is shown in 2 panels, on the left and on the right side of the GUI window. The left panel is related to Software Restriction Policies (SRP) settings, and the right panel to non-SRP settings.



3. 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.



4. 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.
5. 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, but remembers the last settings in that column. They can be restored when pressing the green button the second time. 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 will overwrite the previous settings.  
**If you do not <APPLY CHANGES>, when using <Switch OFF/ON SRP>, then only SRP Default Security Level is changed. That can be useful when you want to disable default-deny protection temporarily, install/update something in the UserSpace, and quickly restore the protection.**
6. 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 on Administrator type of account, as an alternative to LOG OFF.
  - If LOG OFF is not required, then only a splash window 'FINISHED' is shown.
7. 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.
  8. Turning on recommended settings (<Recommended SRP> and <Recommended Restrictions>), give users pretty good, set and forget security setup. Please keep <Run As SmartScreen> set to 'Administrator' to safely bypass SRP when installing the new applications. There is no need to turn off recommended settings when installing Windows Updates and performing system Scheduled Tasks.
  9. 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.
  10. When configured on the particular account, the changes apply to all accounts (except some whitelisted entries related to the particular account).
  11. 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.
  12. 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 or find problematic items, that should be whitelisted (see TROUBLESHOOTING paragraph for more info).

## **FRESH INSTALLATION**

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

### **Updating from version 4.0.0.1 or 4.0.0.2**

The users who applied the custom settings with blocked Sponsors or recommended\_enhanced profile, can look at the new entries in <Block Sponsors> option.

### **Updating from the version 4.0.0.0**

After updating, add manually IQY and SETTINGCONTENT-MS file extensions to 'Designated File Types' list or use <Designated File Types><Restore Defaults> option.

### **Updating from the version prior to 4.0.0.0**

After updating but before reconfiguring the new settings introduced in ver. 4.0.0.0 and 4.0.0.1, you can save your custom-made settings to file, by using <Save Profile> from Hard\_Configurator main Window. They can be restored later by using <Load Profile>. This is not required when predefined settings are used. Also, the custom extensions from 'Designated File Types' list can be saved by using <Designated File Types><Save Extensions>. Those file extensions can be restored later by using <Designated File Types> <Restore Saved>. This is not required when default list is used.

After updating to the new version:

1. Press <Recommended SRP> and next <Recommended Restrictions> to install the new recommended settings in the Windows Registry. Some new settings will overwrite the old ones. <Default Security Level> will be set to Disallowed, 'Windows Script Host' file extensions will be added to Designated File Types list, <Disable Win. Script Host> will be set to OFF, a few additional Windows subfolders will be protected, shortcut protection will be improved, PowerShell sponsors will be blocked (except Windows 10), and some minor changes will apply.

2. If you are using OneDrive, then press <Whitelist By Path> button and find the label 'OneDrive for accounts'. Use first <Remove All> button to remove all the old OneDrive entries and next <Add All> button to whitelist OneDrive on all accounts.

## **QUICK CONFIGURATION** (after the fresh installation).

1. Run Hard\_Configurator and follow the instructions which are displayed on the first run.
2. It is recommended to allow Hard\_Configurator making the System Restore Point, whitelisting the autoruns and applying Recommended Settings. The restore point can be skipped when the kind of rollback software was installed.
3. After those actions, Windows restart will be required.
4. If Windows Defender is primary real-time protection, then <ConfigureDefender> option in Hard\_Configurator can be used to activate advanced Windows Defender settings.
5. Please read the help files to get info about Hard\_Configurator options. Full information about the program and SRP can be accessed using <Documentation> button, available after pressing <General Help> button.  
It is recommended to visit [hard-configurator.com](http://hard-configurator.com) website for detailed information.

## **FULL DEINSTALLATION.**

1. Run Hard\_Configurator (close ConfigureDefender, SwitchDefaultDeny DocumentAntiExploit, and other instances of Hard\_Configurator if running).
2. Press <Tools> button and next <Uninstall Hard\_Configurator> button.
3. Follow the displayed instructions.

## **REMARKS**

After Hard\_Configurator deinstallation:

1. The registry values tweaked by Hard\_Configurator are set to Windows defaults!!!

2. The System Restore is turned ON. It is good to keep this setting ON, when installing security programs. If not required, it can be manually turned OFF, by using the Control Panel or running the Windows tool: `SystemPropertiesProtection.exe`.
3. The DocumentsAntiExploit tool is copied to the Public Desktop, and it is available for managing the MS Office and Adobe Acrobat Reader XI/DC settings on the particular account. Do not delete it, until you are sure that its protection is turned OFF, on all user accounts.

## **SOFTWARE RESTRICTION POLICIES (SRP)**

From the [technet.microsoft.com](http://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](https://technet.microsoft.com/en-us/library/hh831534(v=ws.11).aspx)

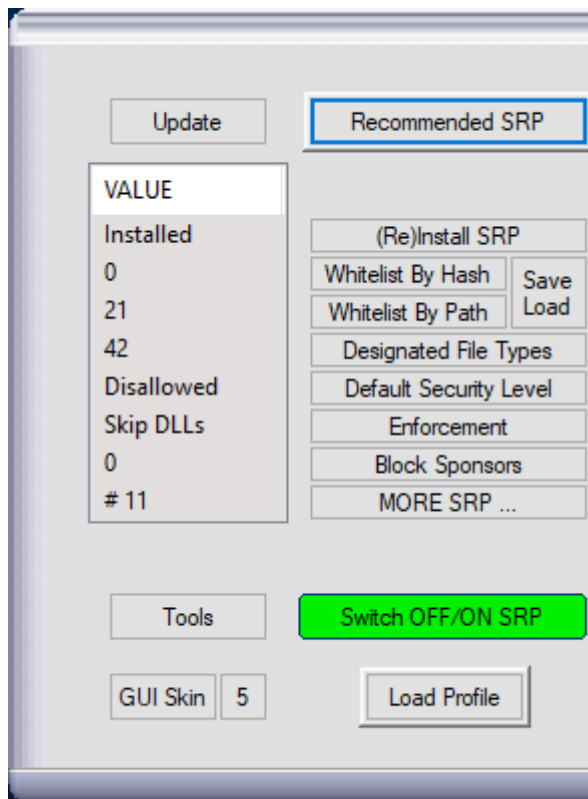
SRP is available via Group Policy for Windows Pro, Windows Enterprise, Windows Server, and Windows Education. Well configured SRP is known in enterprise case studies, as proven protection against virus infections. The development of SRP is now stopped, because Microsoft recommends Windows Defender Application Control in Enterprises. Yet, SRP is still fully functional on all Windows editions.

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 when 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.



In this program, Windows built-in SRP can apply some default-deny and whitelisting/blacklisting features. Executables can be run without SRP restrictions in the **SystemSpace**, that contains UAC protected folders: 'Windows' and 'Program Files' (also 'Program Files (x86)' in 64-bit versions). Outside of those folders (= **UserSpace**), executable files will be blocked by default, when running in the 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’ like: ‘Windows CMD’, ‘Windows Script Host’, and ‘Windows Installer’ have such ability.

Executables from the UserSpace 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 the 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 whitelisting or blacklisting.

SRP in Hard\_Configurator can be completely deactivated by using the button sequence **<Switch OFF/ON SRP>** **<APPLY CHANGES>**.

## How SRP can control file execution/opening.

This section is for the users who want to understand SRP on a deeper level. It is not necessary for using Hard\_Configurator.

How does SRP know what should be monitored (see TABLE (1) and (2))?

- “Designated File Types“ list.
- “Enforcement“ settings.

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). SRP will apply, when Windows Explorer, Edge or Internet Explorer is used to open the files from the local disk. 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 the 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 Command Shell (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 (Edge or IE) and by command using the Sponsor (for example: '**cmd /c path\_to\_malicious.bat**').

3. CreateProcess API function (extended protection).

It calls into SRP while executing **COM/EXE/SCR** files, and SRP is 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 is 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: WSH, WSF, WSC, WS, VBS, VBE, VB, URL, SHS, SETTINGCONTENT-MS, SCT, SCR, REG, PIF, PCD, OCX, MST, MSP, MSC, MDE, MDB, LNK, JSE, JS, JAR, IQY, 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 below 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 (and monitored by SRP).

How SRP knows which monitored files should be blocked.

1. ‘Default Security Level’ settings.
2. Unrestricted/Disallowed rules (by path, by 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 **SystemSpace** = ‘Windows’ + ‘Program Files’ (and ‘Program Files (x86)’ in 64-bit systems).

The below table shows, what files are blocked by SRP in the **UserSpace** (= everything outside of the **SystemSpace**).

**TABLE (3).**

**Files blocked by default in the UserSpace**

**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 **SystemSpace**, 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/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/>

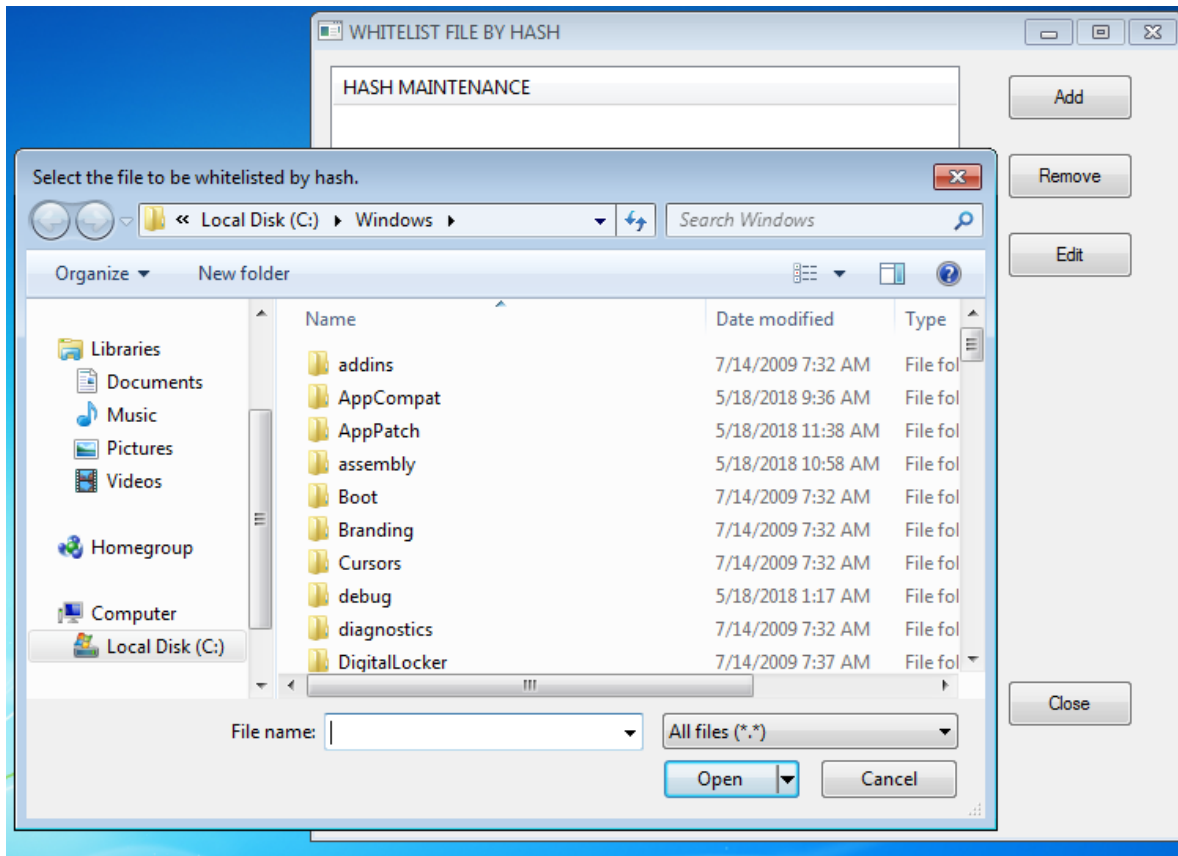
<http://www.wilderssecurity.com/threads/maximising-windows-7-security-with-srp-under-lua-whatever-the-win7-version.262686/>

<http://www.bleepingcomputer.com/tutorials/create-an-application-whitelist-policy-in-windows/>

Registry changes:

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

## WHITELISTING BY HASH



<**Whitelist By Hash**> button opens ADD / REMOVE / EDIT window to manage file whitelisting by hash. It can be useful for running programs located in the UserSpace (outside of the folders: Windows, Program Files, and Program Files (x86) ). The UserSpace 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 this is much safer than whitelisting by path). Hard\_Configurator has the option: <Blocked Events / Security Logs> in the 'Tools' section. It can use NirSoft FullEvent-

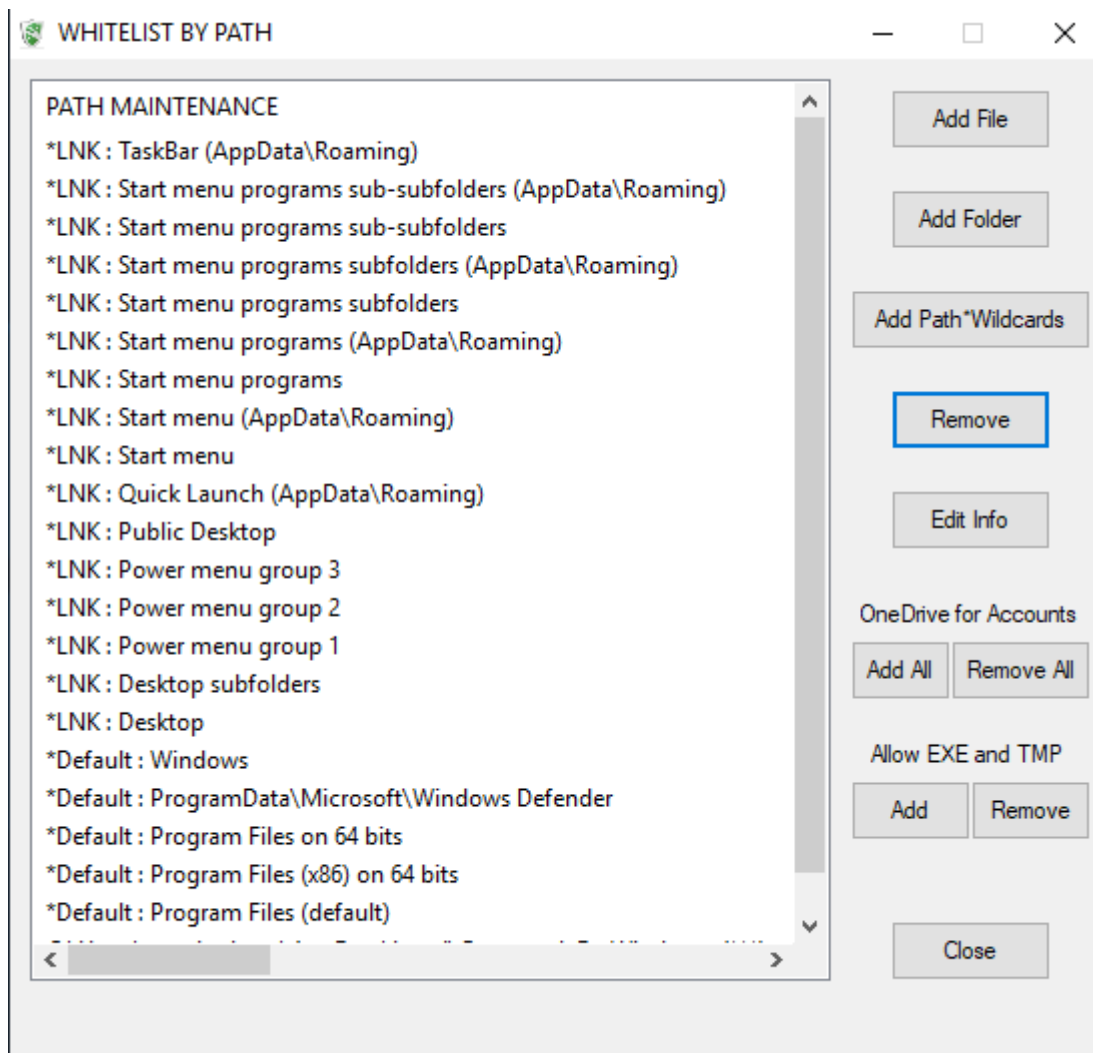


LogView utility to 'filter / view' SRP blocking events and find out which file in the TEMP folder should be whitelisted. This utility is already included in Hard\_Configurator package as an external tool.

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 UserSpace (outside of the folders: 'Windows', 'Program Files ...'). Yet, The UserSpace 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, the below folders should not be whitelisted:

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. The Whitelist can be saved into the file, using <Save Load> button on the right side of the Whitelist buttons.

The users with installed Avast Antivirus set to Hardened Aggressive mode, can apply 'Allow EXE files' option. Those files will be checked by Avast Reputation Cloud and ignored by Hard\_Configurator in the UserSpace.

The above solution is both very safe and usable.

### WARNING!!!

It is forbidden to adopt environment variables when using <Add Path\*Wildcards> option to whitelist the paths!

Registry changes:

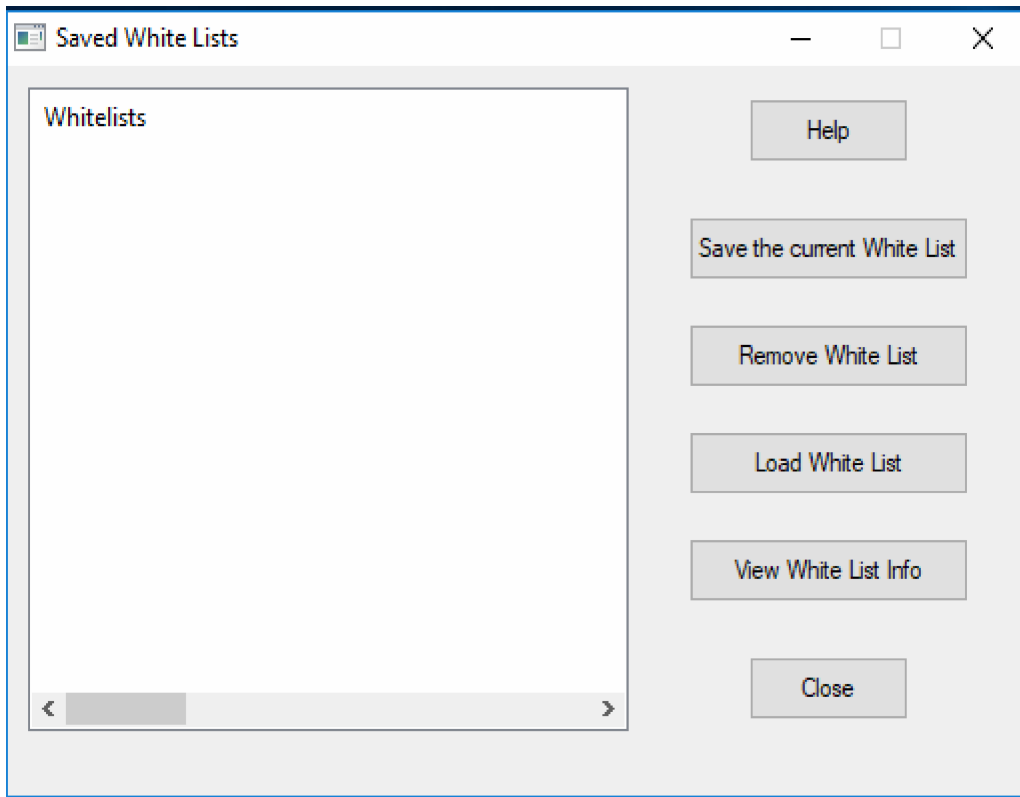
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths

## **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 to 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 for the current, active White List, and next is asked to put some info about the profile (for example the creation date/time, and the short White List characteristics). The names of the saved White List Profiles are visible in the left panel. If the profile with the same name is already in the base, the user is asked if it should be overwritten.



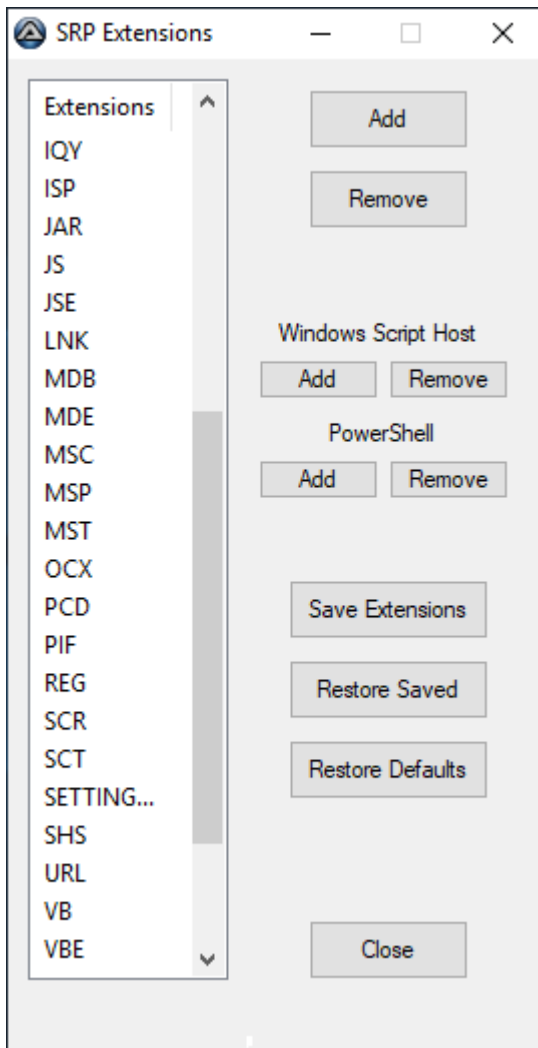
**<Remove White List>** option removes the chosen White List Profile from the Profile Base.

**<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 the 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 characteristics.

## DESIGNATED FILE TYPES

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



Default extensions in Hard\_Configurator (Windows 7+): WSH, WSF, WSC, WS, VBS, VBE, VB, URL, SHS, SETTINGCONTENT-MS, SCT, SCR, REG, PIF, PCD, OCX, MST, MSP, MSC, MDE, MDB, LNK, JSE, JS, JAR, ISP, IQY, 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.

PowerShell script extensions were removed, because Hard\_Configurator has <No PowerShell Exec.> option 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 via <Add> and <Remove> buttons. When using a custom list, it is good to save it (<Save Extensions>). The list can be restored by using <Restore Saved> button.

In some cases, the **below extensions** can be skipped:

- **MSI** extension if <Run As SmartScreen> is set to 'ON'.
- **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 on 64-bit system (for 32-bit programs), 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 level 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 UserSpace:

- ★ protection to all files included in 'Designated File Types' list
- ★ extended security for Windows native executables (COM, EXE, SCR), bi-

nary 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', except when handling LNK, MSI, and script files. In Windows Vista, 'Basic User' works differently, for example, it allows running EXE files even from the UserSpace.

**'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 UserSpace, 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 a 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 UserSpace, 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 libraries (DLL, OCX), due to LoadLibrary API function. This can slow down the system sometimes, and crush Edge browser in older Windows 10 versions.

'All Files' option can protect from many DLL injection attacks, when the system files are used to load malicious libraries (file paths omitted):

```
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, SRP cannot block .NET DLLs or 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 UserSpace (see <Tools> button). Those autoruns and related DLLs (in the UserSpace), 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 files 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. File blocking can be applied, when using combined Disallowed/Unrestricted rules.

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

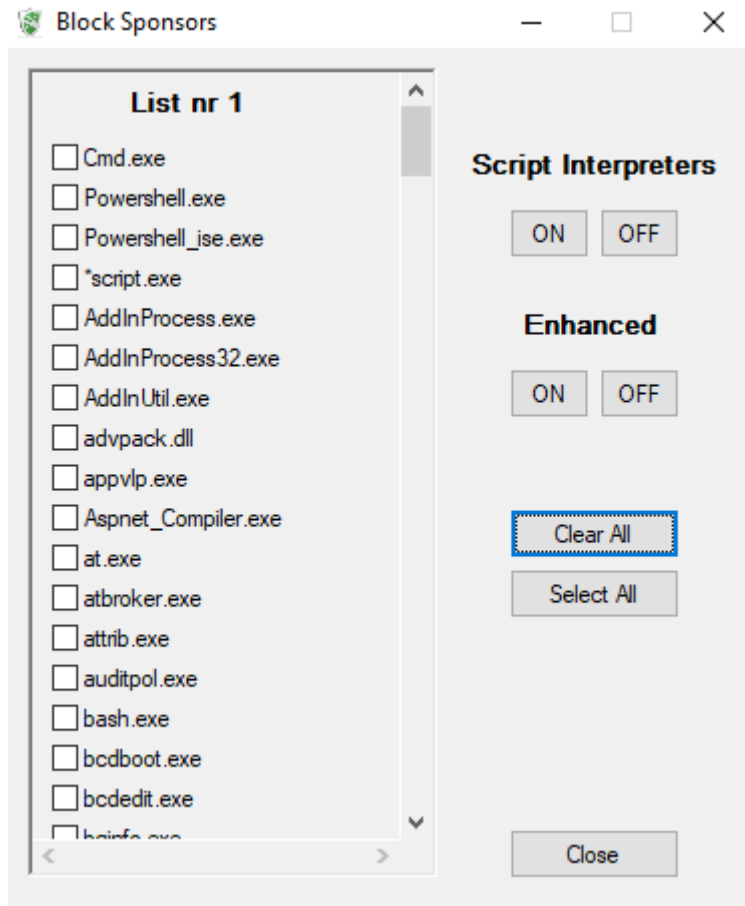
#### Registry changes

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 some system executables. They are known to be used as sponsors when bypassing whitelisting protection. The blacklist is based on the 'Excubits blacklist' (excubits.com).



The sponsors are not blocked in the recommended SRP settings (except PowerShell in Windows Vista, 7, 8, 8.1), but there are situations when they should be blocked temporarily. It can happen, for example, when using the computer 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 powershell\_ise.exe are blocked, then CMD console, PowerShell console, and PowerShell ISE console are disabled.

They can be still used when executed as administrator.

## POWERSHELL SPONSORS

If PowerShell sponsors are blocked, then PowerShell scripts cannot run as standard user when using powershell.exe or powershell\_ise.exe, **even from the SystemSpace**. 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 included in <Recommended SRP> settings on Windows Vista, 7, 8, 8.1. It is not included on Windows 10, because SRP with PowerShell ver. 5.0 (installed by default) can apply Constrained Language mode. The necessary conditions for applying Constrained Language mode are fulfilled, when SRP is 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 and 8.1, 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 of the native Windows executables, Windows CMD, Windows Script Host, and MSI Installer from writable 'C:\Windows' subfolders. So, the execution of EXE, COM, SCR, BAT, CMD, JS, JSE, VBS, VBE, WSF, WSH, and MSI files is blocked, when they are run directly or via command lines with sponsors: cmd.exe, wscript.exe, cscript.exe, and msixec.exe.

This protection uses SRP Disallowed rules, so extended protection is applied to Windows CMD, Windows Script Host, and MSI Installer. It denies the execution even when SRP Default Security Level is set to 'Unrestricted' or mentioned above file extensions are not on SRP Designated File Types list.

In order to block command lines with sponsors of other files (like CHM, HTA, REG, etc.), the sponsors should be blocked via **<Block Sponsors>** (hh.exe, mshta.exe, regedit.exe, reg.exe, regedt32.exe, etc.). Still, the execution is allowed, for programs started with Administrative Rights (or higher) independently of SRP restrictions.

The below writable Windows subfolders are added to SRP blacklist:

C:\windows\debug\WIA

C:\windows\Registration\CRMLog

C:\windows\servicing\Packages

C:\windows\servicing\Sessions

C:\windows\System32\catroot2\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}

C:\windows\System32\com\dmp

C:\windows\System32\FxsTmp

C:\windows\System32\Microsoft\Crypto\RSA\MachineKeys

C:\windows\System32\spool\drivers\color

C:\windows\System32\spool\PRINTERS

C:\Windows\System32\spool\SERVERS

C:\windows\System32\Tasks

C:\Windows\System32\Tasks\_Migrated

C:\Windows\System32\Tasks\Microsoft\Windows\PLA\System

C:\Windows\System32\Tasks\Microsoft\Windows\RemoteApp and Desktop Connections Update

C:\Windows\SysWOW64\Com\dmp

C:\Windows\SysWOW64\FxsTmp

C:\Windows\SysWOW64\Tasks

C:\Windows\SysWOW64\Tasks\Microsoft\Windows\PLA\System

C:\Windows\SysWOW64\Tasks\Microsoft\Windows\RemoteApp and Desktop Connections Update

C:\Windows\Tasks

C:\Windows\Temp

C:\Windows\tracing

Some of them are not writable in Windows 10 (but writable in the prior versions), and a few have not got executable ACL permission.

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-5E544B6A3320}

## **PROTECTING SHORTCUTS**

**<Protect Shortcuts>** button can handle 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.

If <Default Security Level> = 'Disallowed' and <Protect Shortcuts> = 'OFF', then all shortcuts in the UserSpace will be blocked, because the LNK shortcut extension is still on 'Designated File Type' list.

If <Default Security Level> = 'Basic User' and <Protect Shortcuts> = 'OFF', then shortcuts can run EXE files from any location, even if the LNK shortcut extension is on 'Designated File Type' list. This allows also running any script by the shortcut when using script sponsors (cmd.exe, wscript.exe,

cscript.exe, hh.exe, mshta.exe).

Registry changes:

Added GUIDs for Unrestricted rules (whitelisted locations):

[HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\262144\Paths\

{625B53C3-AB48-4EC1-BA1F-A1EF4146FC19}]

...

{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-B029-7FE99A87C645}]

Added GUIDs for Disallowed rules (also due to \*.LNK\\* folder trick):

HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\CodeIdentifiers\0\Paths\

{1016bbe0-a716-428b-822e-5E544B6A3301}

{525B53C3-AB48-4EC1-BA1F-A1EF4146FC19}]

...

{525B53C3-AB48-4EC1-BA1F-A1EF4146FC26}]

{89a0fd77-ed0c-4e30-91ff-9d51428d2f21}]

{89a0fd77-ed0c-4e30-91ff-9d51428d2f22}]

{89a0fd77-ed0c-4e30-91ff-9d51428d2f23}]

{A4BFCC3A-DB2C-424C-B029-7FE99A87C641}]

...

{A4BFCC3A-DB2C-424C-B029-7FE99A87C645}]

## **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, the 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 (not supported on Windows Vista).

If this option is ON, then script file execution is blocked, but the user can still execute PowerShell **commands and cmdlets**. So, **they** are also allowed via Office macros, DDE, etc. Keep this option 'ON', because scripts are the weak point of most antimalware programs.

In Windows 10 the script protection is strengthened by combining SRP with PowerShell Constrained Language mode. In Windows 7, 8, and 8.1 the Constrained Language mode is normally not supported, so Hard\_Configurator in the Recommended SRP settings, blocks also PowerShell Sponsors (powershell.exe and powershell\_ise.exe). It is worth mentioning that Constrained Language can be also introduced to Windows 7 and 8.1 with updated PowerShell to version 5.0 (or higher).

In Windows Vista, <No PowerShell Exec.> and Constrained Language mode are normally not supported, so Hard\_Configurator can apply the protection only via SRP by adding PowerShell script extensions to 'Designated File Types' list, and by blocking PowerShell Sponsors: powershell.exe and powershell\_ise.exe (<Block Sponsors> button).

See also the info in **BLOCK SPONSORS** --> **POWERSHELL SPONSORS**.

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

HKLM\Software\Policies\Microsoft\Windows\PowerShell\EnableScripts

Value (Dword)

0 script execution is disabled

1 script execution 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 (also as administrator). Keep this option ON, if SRP <Default Security Level> is not set to 'Disallowed', because only 'Disallowed' setting can force extended SRP protection for those scripts. It is important, 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 in the home environment, so can be blocked.

In Windows 64-bit there are two Windows Script Hosts (32-bit and 64-bit).

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

## **DOCUMENTS ANTI-EXPLOIT**

### **Important remarks.**

The home users should avoid installing MS Office or Adobe Acrobat Reader, except when it is necessary. Those applications have so many advanced features, that it is hardly possible to fully protect users against all vulnerabilities.

They can be fully protected only in theory, because the more protection, the greater chance to see some documents unreadable. Most home users do not use those advanced features at all, and on the contrary, the most weaponized documents use them to exploit/infect the system.

Generally, it is recommended to use online services or Universal Applications from Microsoft store, for managing documents (Office Online, Google Drive, Word Mobile, Excel Mobile, PowerPoint Mobile, Adobe Reader Touch, Foxit MobilePDF, etc.). Universal Applications are prepared to work in AppContainer isolation from unneeded resources and other applications.

Such popular applications like Libre Office, WPS Office, SoftMaker Office are also the better choice, but they are not as safe as the above solutions. For the compatibility reasons, most of the document active content is still functional (macros, OLE, scripts, etc.) and can be as dangerous as with MS Office or Adobe Acrobat Reader applications.

Anyway, some users have no choice and are forced to use MS Office or Adobe Acrobat Reader. So what can they do to provide the enhanced security?

MS Office 2007 and newer versions provide not so bad default protection against weaponized office documents, but the user has to avoid allowing the active content (macros, OLE, DDE, ActiveX, etc.). That is hardly possible for inexperienced users, who usually do not understand the security alerts.

The situation is even worse for Adobe Acrobat Reader, because in many cases, the active content embedded in PDF documents, is allowed by default without any alert. Furthermore, there is no possibility to silently block the active content, because Adobe Acrobat Reader shows the ‘Yellow Message Bar’ with option to allow the blocked features.

On Windows 10 it is recommended to install Adobe Acrobat Reader DC (from the year 2018 at least), because of AppContainer feature which can mitigate many dangerous actions.

### **What <Documents Anti-Exploit> can do:**

- The VBA interpreter in MS Office is disabled, so VBA Macros (in documents, templates, etc.), VBA Add-ins, and VBA UserForms are blocked. This may have sometimes a direct impact on the proper functioning of OLE Automation, Form/ActiveX/COM controls, etc.
- The dangerous features in Adobe Acrobat Reader DC (version from the year 2018 at least) on Windows 8.1/10 can be blocked with the ‘Yellow Message Bar’, and if allowed by the user, then silently mitigated in AppContainer;
- The dangerous features in Adobe Acrobat Reader XI (all Windows versions) and Adobe Acrobat Reader DC (Windows 8 and prior versions) can be blocked with the ‘Yellow Message Bar’ (the user can allow them);
- The restrictions apply as policies for all accounts and override (but not overwrite) applications' native settings in MS Office and Adobe Acrobat Reader XI/DC;
- The restrictions cannot be modified by the user from within MS Office and Adobe Acrobat Reader XI/DC.

### **The available settings.**

#### **<Documents Anti-Exploit> = Adobe + VBA**

This setting is recommended for anyone on Windows 10, who installed MS Office and Adobe Acrobat Reader XI/DC. It provides enhanced protection, when supported by default-deny Hard\_Configurator setup (Recommended SRP + Recommended Restrictions).

The VBA interpreter is disabled for MS Office XP/2003, and higher versions up to MS Office 2016 (Excel, FrontPage, Outlook, PowerPoint, Publisher,

and Word).

In Adobe Acrobat Reader XI/DC, the important & protective features are turned ON.

The users, who require the protection against the 0-day sophisticated malware, may also consider activating Windows Defender ASR rules on Windows 10.

Hardening MS Office and Adobe Acrobat Reader XI/DC is also possible via DocumentsAntiExploit tool, which is available from SwitchDefaultDeny application. This tool can apply protective features on a particular account (non-system-wide). It is also recommended when the user cannot activate Windows Defender ASR rules.

### **<Documents Anti-Exploit> = Adobe**

The policy restrictions apply only for Adobe Acrobat Reader XI/DC.

This setting is recommended when MS Office is not installed and other applications are used for managing office documents.

### **<Documents Anti-Exploit> = OFF**

Generally, the system-wide policies can override but do not overwrite non-system-wide restrictions.

The OFF setting removes policy restrictions for MS Office and Adobe Acrobat Reader XI/DC, so the non-system wide restrictions can apply (via DocumentsAntiExploit tool or from within MS Office or Adobe Acrobat Reader applications).

This setting is also displayed when both MS Office and Adobe Acrobat Reader are not installed.

### **<Documents Anti-Exploit> = Partial**

This setting is displayed when the policy restrictions were applied via external program (in the HKLM Registry Hive), and do not match predefined Hard\_Configurator settings (Adobe + VBA, Adobe, OFF). It is related only to the system-wide restrictions, so does not show the restrictions made for the particular account via DocumentsAntiExploit tool or from within MS Office and Adobe Acrobat Reader applications.



## **SwitchDefaultDeny <Documents Anti-Exploit>**

SwitchDefaultDeny is a companion utility to Hard\_Configurator. It allows the user to:

- switch between default-deny and default-allow modes in SRP, without running Hard\_Configurator;
- harden MS Office and Adobe Acrobat Reader XI/DC (DocumentAntiExploit tool is used).

This feature was prepared for users who:

- cannot apply ASR rules on Windows 10,
- cannot use 'Adobe + VBA' option in Hard\_Configurator,
- use earlier Windows versions (no ASR mitigations).

The option <Documents Anti-Exploit> in SwitchDefaultDeny application runs DocumentsAntiExploit tool (external application). It can be used to Harden MS Office and Adobe Acrobat Reader XI/DC applications. On the contrary to Hard\_Configurator <Documents Anti-Exploit> system-wide feature, the DocumentsAntiExploit tool is focused on the current account from which was started. So, the user can apply different restrictions on different accounts.

In MS Office, the below settings are applied (valid up to MS Office 2016):

- Disabled Macros in MS Office XP and MS Office 2003+ (Word, Excel, PowerPoint, Access, Publisher, Outlook).
- Disabled Access to Visual Basic Object Model (VBOM) in MS Office 2007+ (Access, Excel, PowerPoint, and Word).
- Disabled DDE in Word 2007+ (requires Windows Updates pushed in January 2018, see Microsoft Security Advisory ADV170021).
- Disabled auto-update for any linked fields (including DDE and OLE) in Word 2007+, Excel 2007+, Outlook 2007+, One Note 2013+.
- Disabled ActiveX in MS Office 2007+.
- Disabled OLE in MS Office 2007+ (Word, Excel, PowerPoint).

- Disabled 'Run Programs' option for action buttons in PowerPoint 2007+.
- Disabled automatic download of linked images in PowerPoint 2007+.
- Disabled TrustBar notifications in MS Office 2007+ .

In Adobe Acrobat Reader XI/DC, the below settings are applied :

- The dangerous features in Adobe Acrobat Reader DC (version from the year 2018 at least) on Windows 8.1/10 can be blocked with the 'Yellow Message Bar', and if allowed by the user, then silently mitigated in App-Container.
- The dangerous features in Adobe Acrobat Reader XI (all Windows versions) and Adobe Acrobat Reader DC (Windows 8 and prior versions) can be blocked with the 'Yellow Message Bar' (the user can allow them).
- The restrictions apply for the current account and overwrite native settings in Adobe Acrobat Reader XI/DC.
- The user can apply different restrictions on different accounts.

DocumentsAntiExploit tool is copied to the Public Desktop when uninstalling Hard\_Configurator, so the user can still use it to harden/unharden the MS Office and Adobe Acrobat Reader XI/DC applications.

It is portable, but before removing from the computer, the user should apply the settings: <MS Office> = OFF, <Adobe Acrobat Reader> = OFF on his/her accounts to recover default values of changed settings. If not, then a few settings can be locked (non-configurable) in MS Office applications. See the help files in DocumentsAntiExploit tool, for more help.

## **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 UserSpace (= 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 fully replace the functionality of "Run as administrator", 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 SystemSpace (= inside 'Windows', 'Program Files', and 'Program Files (x86)' folders), and their location in the UserSpace, 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 on Explorer context menu. Both options **force** file execution with **SmartScreen check** for files located in the UserSpace. If the file is located in the SystemSpace (inside 'Windows', 'Program Files ...'), then SmartScreen works as usual (**not forced**).

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 these options from the Explorer context menu.

**'Run As SmartScreen'** option can **check** EXE and MSI files (also via shortcuts), and run them elevated if accepted by SmartScreen filter. All other files are blocked with notification when 'Run As SmartScreen'.

**'Run By SmartScreen'** option can be conveniently used to strengthen default-allow SRP and has some default-deny SRP features (like blocking the potentially dangerous files). It can also **check**/run more executables (COM, EXE, MSI, SCR) via SmartScreen. The main difference is that SRP applies the real-time protection and 'Run By SmartScreen' is on demand protection (see more in the next section).

'Run As SmartScreen' and 'Run By SmartScreen' are not designed to run files from the **root 'c:\'** location (requires Administrative Rights).

#### **(A) Keep the 'Administrator' setting when SRP is activated.**

If so, then the users can safely:

1. Run programs (with a mouse click or pressing ENTER button) which have been already installed in the SystemSpace 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. Install new programs from the UserSpace by using 'Run As SmartScreen' option from 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> to 'Standard User' and <Hide 'Run As Administrator'> to 'OFF', as an alternative solution. Then, 'Run By SmartScreen' + SRP can serve as a second opinion scanner for executables located in the UserSpace.

In the (B) solution the user should always use first 'Run By SmartScreen' option from the right-click Explorer context menu to check all new files.

The main difference between (A) and (B) solutions is that in (A), the new application installers (EXE and MSI files) are allowed to run if accepted by SmartScreen, but instead, they are always blocked in (B) when 'Run By SmartScreen' is used.

In (B) the user has to use additionally 'Run as administrator' from the Explorer context menu to run application installers.

In (A) 'Run As SmartScreen' can replace 'Run By SmartScreen' + 'Run as administrator' for application installers (EXE, MSI files).

**(C) Keep the 'Standard User' setting + <Hide 'Run As Administrator'> set to 'OFF' for using 'Run By SmartScreen', when SRP is deactivated.**

## REMARKS

The SmartScreen Filter in Windows 8+ allows some vectors of infection if you have got the executable file (BAT, CMD, COM, CPL, DLL, EXE, JSE, MSI, OCX, PIF, SCR, VBE) using:

- \* the downloader or torrent application (EagleGet, utorrent, etc.);
- \* container format file (ZIP, 7Z, ARJ, RAR, ...) except Windows built-in ZIP;
- \* CD/DVD/Blue-ray disc or disc image (iso, bin, etc.);
- \* non-NTFS USB storage device (FAT32 pen drive, FAT32 USB disk);
- \* Memory Card;

so the file does not have the proper Alternate Data Stream attached.

Registry changes:

HKCR\\*\shell\Run As SmartScreen\ ,    HKCR\\*\shell\Run By SmartScreen\

The default shortcut flag `IsShortcut` is changed to `NoIsShortcut` under the below Registry keys:

HKCR\Application.Reference,	HKCR\IE.AssocFile.URL,	HKCR\IE.AssocFile.WEBSITE
HKCR\InternetShortcut,	HKCR\piffile ,	HKCR\Microsoft.Website,
HKCR\WSHFile		

## **RUN BY SMARTSCREEN OPTION IN EXPLORER CONTEXT MENU.**

"Run By SmartScreen" option is added to Explorer right-click context menu when Hard\_Configurator <Run As Smartscreen> feature is set to '**Standard User**'. So, when 'Run By SmartScreen' option in Explorer context menu is

used, the files are opened/run **as standard user** (medium rights = no elevation).

If the user tries to open the file with "Run By SmartScreen" it works as follows:

1. Executables (COM, EXE, MSI, SCR) which are located in the SystemSpace ( 'C:\Windows', 'C:\Program Files', 'C:\Program Files (x86)' ) are opened normally, without SmartScreen check.
2. Executables located in the UserSpace (= outside 'C:\Windows', 'C:\Program Files', 'C:\Program Files (x86)') are checked by SmartScreen before running.
3. Files from the UserSpace with potentially dangerous extensions (scripts, most MS Office files, etc.), are not allowed to open, and the program shows an alert.
4. Shortcut (\*.lnk) to any file (target file) is managed as the target file.
5. Shortcuts with a command line in the 'Target' area are always blocked, and the program shows an alert.
6. Compressed archives (7Z, ARJ, RAR, ZIPX) are not opened - only the short instruction is displayed.
7. Popular file formats related to MS Office and Adobe Reader: DOC, DOCX, XLS, XLSX, PUB, PPT, PPTX, ACCDB, PDF are opened with the warning instruction. They are recognized by Office applications and Adobe Reader 10+/DC, as downloaded from the Internet (MOTW is added to the files).
8. Other files (ZIP archives, media, photos, etc.) are opened normally without warnings.

The program has hardcoded list of unsafe file extensions:

ACCD, ACCDA, ACCDE, ACCDR, ACCDT, ACM, AD, ADE, ADN, ADP, AIR, APP, APPLICATION, APPREF-MS, ARC, ASA, ASP, ASPX, ASX, AX, BAS, BAT, BZ, BZ2, CAB, CDB, CER, CFG, CHI, CHM, CLA, CLASS, CLB, CMD, CNT, CNV, COM, COMMAND, CPL, CPX, CRAZY, CRT, CRX, CSH, CSV, DB, DCR, DER, DESKLINK, DESKTOP, DIAGCAB, DIF, DIR, DLL, DMG, DOCB, DOCM, DOT, DOTM, DOTX, DQY, DRV, EXE, FON, FXP, GADGET, GLK, GRP, GZ, HEX, HLP, HPJ, HQX, HTA, HTC, HTM, HTT, IE, IME, INF, INI, INS, IQY, ISP, ITS, JAR, JNLP, JOB,

JS, JSE, KSH, LACCDB, LDB, LIBRARY-MS, LOCAL, LZH, MAD, MAF, MAG, MAM, MANIFEST, MAPIMAIL, MAQ, MAR, MAS, MAT, MAU, MAV, MAW, MAY, MCF, MDA, MDB, MDE, MDF, MDN, MDT, MDW, MDZ, MHT, MHTML, MMC, MOF, MSC, MSH, MSH1, MSH1XML, MSH2, MSH2XML, MSHXML, MSI, MSP, MST, MSU, MUI, MYDOCS, NLS, NSH, OCX, ODS, OPS, OQY, OSD, PCD, PERL, PI, PIF, PKG, PL, PLG, POT, POTM, POTX, PPAM, PPS, PPSM, PPSX, PPTM, PRF, PRG, PRINTEREXPORT, PRN, PS1, PS1XML, PS2, PS2XML, PSC1, PSC2, PSD1, PSDM1, PST, PSTREG, PXD, PY, PY3, PYC, PYD, PYDE, PYI, PYO, PYP, PYT, PYW, PYWZ, PYX, PYZ, PYZW, RB, REG, RPY, RQY, RTF, SCT, SEA, SEARCH-MS, SEARCHCONNECTOR-MS, SETTING-CONTENT-MS, SHB, SHS, SIT, SLDM, SLDX, SLK, SPL, STM, SWF, SYS, TAR, TAZ, TERM, TERMINAL, TGZ, THEME, TLB, TMP, TOOL, TSP, URL, VB, VBE, VBP, VBS, VSMACROS, VSS, VST, VSW, VXD, WAS, WBK, WEBLOC, WEBPNP, WEBSITE, WS, WSC, WSF, WSH, XBAP, XLA, XLAM, XLB, XLC, XLD, XLL, XLM, XLSB, XLSM, XLT, XLTM, XLTX, XLW, XML, XNK, XPI, XPS, Z, ZFSENDTOTARGET, ZLO, ZOO

The above list is based on SRP, Outlook Web Access, Gmail, and Adobe Acrobat Reader file extension blacklists.

The files with extensions: BAT, CMD, CPL, DLL, JSE, OCX, and VBE are supported by SmartScreen Application Reputation. But, their SmartScreen detection is not good, so they are added to the list of unsafe file extensions. Even if they are accepted by SmartScreen, then will be blocked with notification.

## **REMOTE ACCESS**

**If <Block Remote Access> is set to ON**, then the below remote features are disabled by Administrator:

- \* Remote Assistance (solicited and unsolicited)
- \* Remote Desktop
- \* Remote Shell Access

### \* Remote Registry Access

The user cannot enable those remote features via System Properties and Control Panel.

**If <Block Remote Access> set to OFF**, then the user can enable or disable the remote features via System Properties and Control Panel.

It is recommended for home users to keep <Block Remote Access> set to 'ON'. Remote connections are frequently exploited by malware and hackers.

### REMARK

Changing this option (either set to 'ON' or to 'OFF') always stops 'Remote Registry' service, if it was started. The potential problems may occur when disabling Remote Access:

“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:

For <Block Remote Access> 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 below values are deleted:

fAllowUnsolicited, fAllowToGetHelp, fDenyTSConnections, AllowRemoteShellAccess,

Remote Registry setting does not change on Windows 8+ ("Start=dword:00000004"), but on Windows 7 and Vista it will be changed to "Start=dword:00000003".



## **UNTRUSTED FONTS**

**<Disable Untrusted Fonts> - this feature is disabled in Hard\_Configurator from version 4.0.0.0**

“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.”

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

## **16-BIT APPLICATIONS**

If **<DISABLE 16-BITS>** = 'ON', then 16-bit applications are disabled.

The 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).

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 run only 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. See also the possible bypass:

[http://oalabs.openanalysis.net/2015/06/04/malware-persistence-hkey\\_current\\_user-shell-extension-handlers/](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 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> = ON (one cannot bypass SRP).

There are no problems with Windows Updates, scheduled system tasks, and installing/updating Universal Applications from Windows Store. But, the new installations/updates of desktop applications, should be made on 'Administrator Account' (two accounts are required) or via scheduled tasks.

If the application uses %UserProfile% for updating as standard user (like One Drive) then the specific update/application folder in the UserSpace (but not all %UserProfile%) should be whitelisted and then the updating can be performed on SUA.

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 very secure, even when not using third-party security software (anti-exe, anti-exploit, HIPS).

Registry changes:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System!Consent-PromptBehaviorUser

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 is activated (MSI files are blocked by default). Then, one can bypass SRP, choosing 'Run as administrator' from the right-click Explorer context menu.

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 uninstalled 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 the Windows system tool 'OptionalFeatures.exe'.

**<Disable SMB>** option is not included in recommended restrictions, because sometimes (rarely), it can be required 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 (removed on Windows 10 ver 1803+).

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!CachedLogons-Count'

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.

### **Warning.**

This feature sometimes fails to show the SAS prompt, that can have unexpected consequences while updating Windows. It is recommended to turn it off while making Windows Updates.

Registry changes:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\CredUI!EnableSecureCredentialPrompting

Value (REG\_DWORD)

1	Secure Credential Prompting enabled
0	Secure Credential Prompting disabled

## CONFIGURING WINDOWS DEFENDER

**<ConfigureDefender>** button opens ConfigureDefender application. It can be useful for changing the hidden features of Windows Defender on Windows 10. It mostly uses PowerShell cmdlets (with a few exceptions).

Most settings available in ConfigureDefender are related to Windows Defender real-time protection and work only when Windows Defender real-time protection is set to "ON".

Important: *These two settings (below) should **never** be changed because important features like "Block at First Sight" and "Cloud Protection Level" will not work properly:*

"Cloud-delivered Protection" = "ON"

"Automatic Sample Submission" = "Send"

### **ConfigureDefender Protection Levels (pre-defined settings):**

"DEFAULT"

Microsoft Windows Defender default configuration which is applied automatically when installing the Windows system. It provides basic antivirus protection and can be used to quickly revert any configuration to Windows defaults.

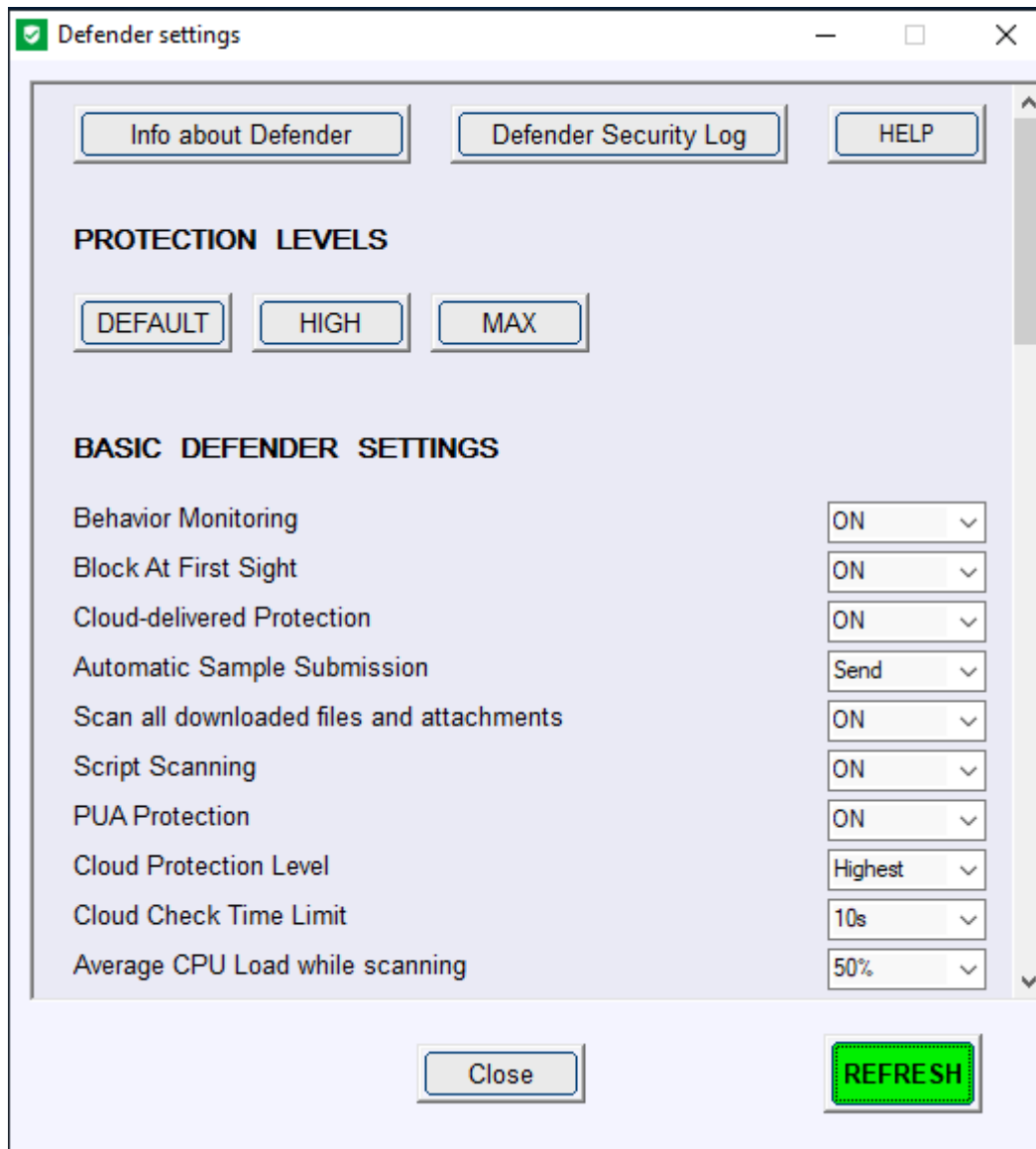
"HIGH"

Enhanced configuration which enables Network Protection and most of Exploit Guard (ASR) features. Three Exploit Guard features and Controlled Folder Access ransomware protection are disabled to avoid false positives. This is the recommended configuration which is appropriate for most users and provides significantly increased security.

"MAX"

This is the most secure protection level which enables all advanced Windows Defender features and hides Windows Security Center. Configuration chan-

ges can be made *only* with the ConfigureDefender user interface. The "MAX" settings are intended to protect children and casual users but can be also used (with some modifications) to maximize the protection. This protection level usually generates more false positives compared to the "HIGH" settings and may require more user knowledge or skill.



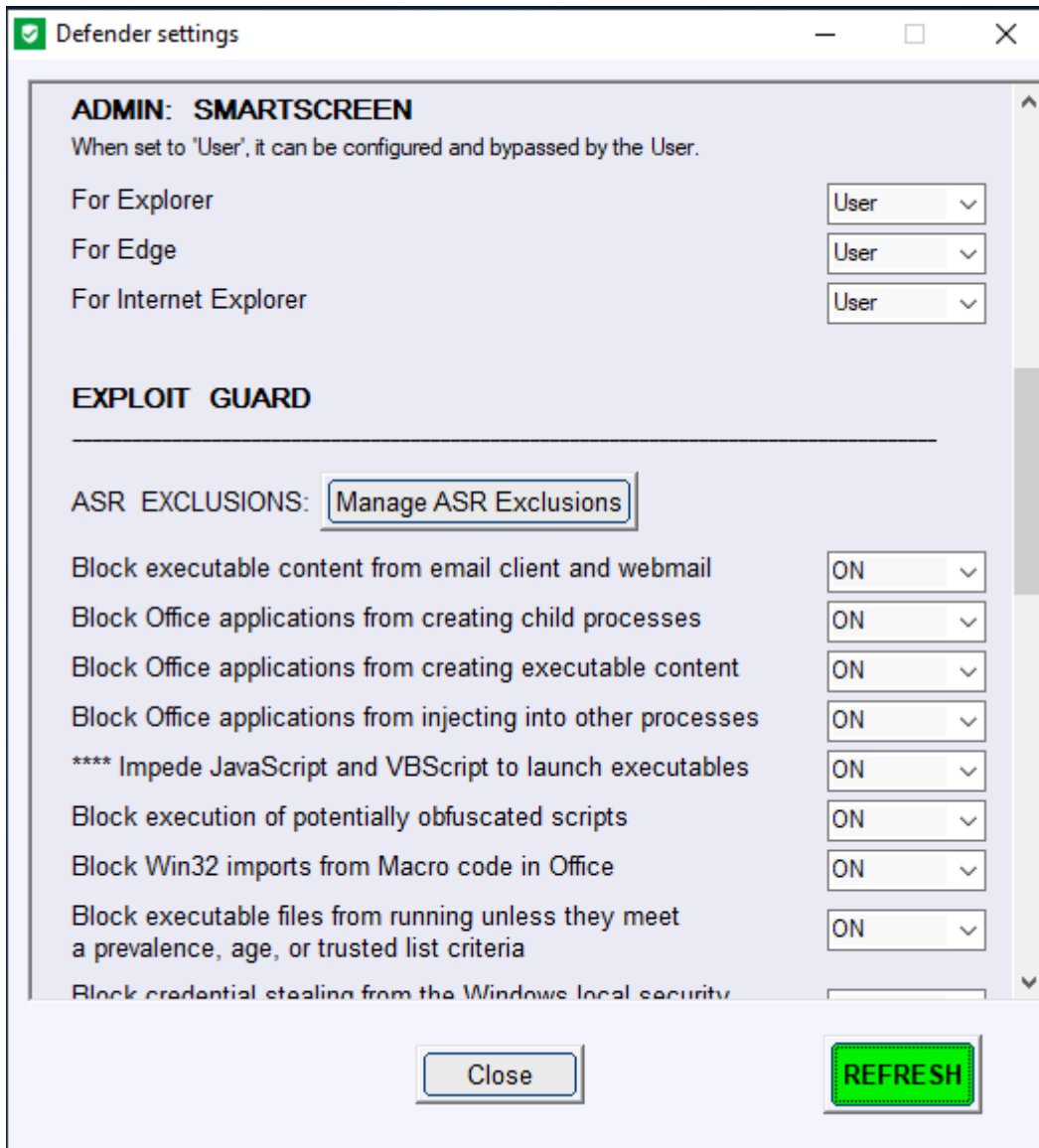
### **ConfigureDefender custom settings:**

You may customize your configuration by choosing any of the three protection levels and then change individual features.



## How to apply the settings:

Select a Protection Level or custom configuration, press the "Refresh" green button and let ConfigureDefender confirm the changes. ConfigureDefender will alert if any of your changes have been blocked. ***Reboot to apply chosen protection.***




## Audit mode:

Many ConfigureDefender options can be set to "Audit". In this setting, Windows Defender will log events and warn the user about processes which would otherwise be blocked with this setting "ON". This feature is available

for users to check for software incompatibilities with applied Defender settings. The user can avoid incompatibilities by adding software exclusions for ASR rules and Controlled Folder Access.

## Defender Security Log:

This option can gather the last 200 entries from the Windows Defender Anti-virus events. These entries are reformatted and displayed in the notepad. The following event IDs are included: 1006, 1008, 1015, 1116, 1117, 1118, 1119, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 3002, 5001, 5004, 5007, 5008, 5010, 5012. Inspecting the log can be useful when a process or file execution has been blocked by Windows Defender Exploit Guard.

 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
<hr/>					
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

## **Development of Windows Defender features:**

ConfigureDefender works on Windows 10. Windows 8.1 and earlier versions are not supported. Microsoft has added new Windows Defender features with successive Windows 10 feature updates. Below is the list of ConfigureDefender features available on different versions of Windows 10:

At least Windows 10

Real-time Monitoring, Cloud-delivered Protection, Cloud Protection Level (Default), Cloud Check Time Limit, Automatic Sample Submission, Behavior Monitoring, Scan all downloaded files and attachments, Average CPU Load while scanning, PUA Protection.

At least Windows 10, version 1607 (Anniversary Update)  
Block At First Sight

At least Windows 10, version 1703

Cloud Protection Level (High level for Windows Pro & Enterprise), Cloud Check Time Limit (Extended to 60s)

At least Windows 10, version 1709 (Fall Creators Update)

Attack Surface Reduction, Cloud Protection Level (extended levels for Windows Pro & Enterprise), Controlled Folder Access, Network Protection.

## **Registry management.**

Windows Defender stores its native settings under the registry key (owned by SYSTEM): HKLM\SOFTWARE\Microsoft\Windows Defender

These can be changed when using PowerShell cmdlets. A few settings can be also changed from Windows Security Center.

Administrators can use Group Policy Management Console to apply policy settings for Windows Defender. They are stored under another registry key (policy key owned by ADMINISTRATORS):

HKLM\SOFTWARE\Policies\Microsoft\Windows Defender

Group Policy settings can override but do not change native Windows Defender settings. The native settings **are automatically recovered when removing** Group Policy settings.

The ConfigureDefender utility removes the settings made via direct registry editing under the policy key: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender

This is required because those settings would override ConfigureDefender settings.

The ConfigureDefender utility may be used on all Windows 10 versions. **But, on Windows Professional and Enterprise editions it will only work if your Administrator has not applied Defender policies by using another management tool, for example, Group Policy Management Console.**

These policies are set to "Not configured" by default. If they have been changed by Administrator, then they should be reset to "Not configured". Group Policy settings may be found in Group Policy Management Console:

Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Defender Antivirus

The settings under the tabs: MAPS, MpEngine, Real-time Protection, Reporting Scan, Spynet, and Windows Defender Exploit Guard should be examined.

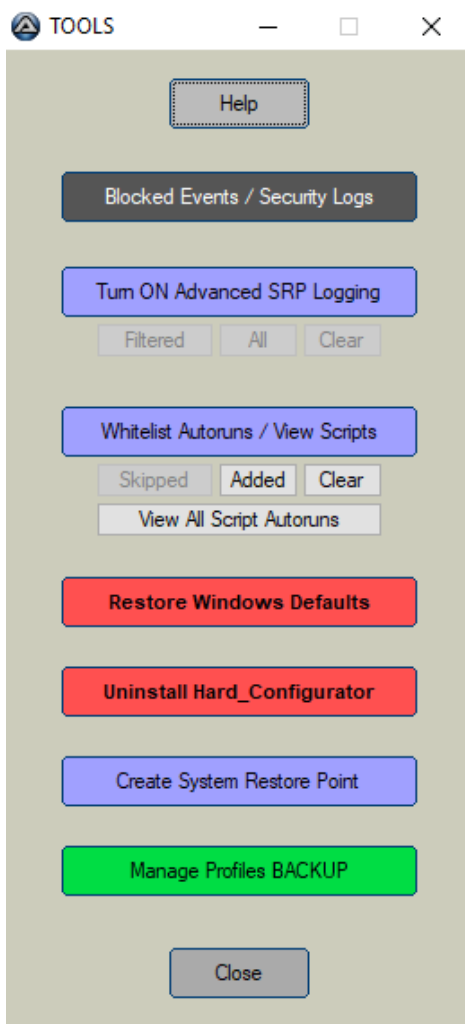
***Please note:** Group Policy Refresh feature will override ConfigureDefender settings if Defender Group Policy settings are not reset to "Not configured"! ConfigureDefender should not be used to configure the settings, alongside other management tools deployed in Enterprises, like Intune or MDM CSPs.*

# TROUBLESHOOTING

## **Hard\_Configurator troubleshooting.**

1. If the system hangs after reboot (very rarely), then it can be a sign, that SRP or one of the 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 whitelist the blocked entry or 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 UserSpace, restore Windows defaults, make a System Restore Point, backup and restore predefined profiles, or uninstall Hard\_Configurator.

## <Blocked Events / Security Logs>

When the program/script is blocked by Hard\_Configurator, the information is usually written in the Windows Event Log. This option filters the output of NirSoft tool: FullEventLogView to retrieve information about the blocked events and some security – related events.

The config file uses events ID as follows:

### ★ **SRP** (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

### ★ **Non-SRP related**

- 1000 -> provider: Windows Script Host, only when scripts were run with Administrative Rights
- 4100 -> provider: Microsoft-Windows-PowerShell
- 1121-1128 -> provider Microsoft-Windows-Windows Defender (Exploit Guard ASR, Controlled Folder Access, Network Protection)
- 1000, 1006, 1007, 1008, 1015, 1116, 1117, 1118, 1119, 3002, 5001, 5004, 5007, 5008, 5010, 5012 -> provider Microsoft-Windows-Windows Defender (Windows Defender)

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

<Blocked Events / Security Logs> 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 the info about processes, which **were run with Administrative Rights**, to the file SRP.log. Yet, this log has usually many entries from the SystemSpace, so some filtering is required. The <Filtered> button checks SRP.log and leaves only entries related to scripts or processes which were run from the UserSpace.

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 <Filtered> Log.

Those DLLs should be whitelisted. For example, if 'EagleGet Downloader' application is installed in the folder: D:\Portable\EagleGet\_\n then after "Run As Administrator" EagleGet.exe the <Filtered> Log shows some UserSpace 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 de-  
fault rule, Guid = {11015445-d282-4f86-96a2-9e485f593302}  
.... (many other dlls)
```

All the above DLLs and EXEs (EagleGet.exe , ...) must be whitelisted.

Another example, when some application 'myapp.exe' is "Run As SmartScreen", then the entries in the log may look like:

```
myapp_setup.exe (PID = 7246) identified C:\Users\Admin\AppData\Local\Temp\is-  
PPQV9.tmp\myapp_setup.tmp as Unrestricted using default rule, Guid = {11015445-  
d282-4f86-96a2-9e485f593302}"
```

So, we know that myapp.exe is wrapped and uses **myapp\_setup.tmp** to execute in the temporary folder:

'C:\Users\USERNAME\AppData\Local\Temp\is-PPQV9.tmp\.'

Now, the **myapp\_setup.tmp** file can be whitelisted by path:

'C:\Users\USERNAME\AppData\Local\Temp\is-?????.tmp\myapp\_setup.tmp'

or by hash (if the file **myapp\_setup.tmp** was not deleted).

### <Whitelist Autoruns / View Scripts>

Some processes can be loaded at the boot time from the UserSpace (= 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.

<Whitelist Autoruns / View Scripts> option can filter out all numerous autoruns from the SystemSpace leaving only a few entries from the UserSpace. They can be seen when pressing <Added> button.

Rarely, the autoruns can have a complicated structure, and the filtering algorithm may give up. Those entries should be checked manually - they can be seen when pressing <Skipped> button.

Pressing <View All Script Autoruns> shows all scripts (from System and User Space) started at the boot time. This option may be helpful when the user wants to disable Windows Script Host, PowerShell or Windows CMD.

### <Restore Windows Defaults>

This option allows restoring all Windows Registry keys that could be changed by Hard\_Configurator, to default values (except Windows Defender). 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> does not change the System Restore settings.

After Hard\_Configurator uninstallation, the System Restore is typically turned ON, which is the default setting on 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.

This option requires rebooting the system.



### <Uninstall Hard\_Configurator>

After using Hard\_Configurator, it cannot be uninstalled via Windows uninstall feature, because Hard\_Configurator entry is deleted from the list of installed applications.

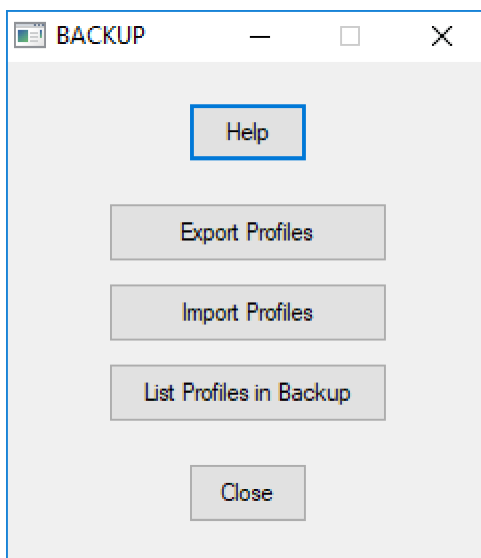
This option can restore Windows default settings, including Windows Defender settings, and removes Hard\_Configurator files from disk. The Document-AntiExploit tool is copied to the PUBLIC Desktop.

This prevents users from uninstalling Hard\_Configurator without restoring Windows default settings.

### <Create System Restore Point>

Makes Windows restore point named Hard\_Configurator. If the System Restore feature was turned off, then it will be turned on, and the 1GB of the disk space will be reserved for the restore points.

### <Manage Profiles Backup>



Hard\_Configurator can back up its Profile Base (all saved Whitelist Profiles and Settings Profiles) into the 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 the 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 pen drive or another 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.

 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

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).

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.

# Frequently Asked Questions

This FAQ is a copy of FAQ from <https://hard-configurator.com/faq.html>

## Abbreviations used in this FAQ:

- H\_C - Hard\_Configurator
- AV - Antivirus application
- SRP - Software Restriction Policies (Windows built-in security feature)
- UAC - User Account Control
- SUA - Standard User Account
- AA - Administrator Account; not to be confused with 'Built-in Administrator Account' (disabled by default), that can be used to boot Windows to 'Audit mode'.

## Basic concepts:

### Standard rights (standard user rights)

These are standard (default) rights granted by the Windows system to processes initiated by the user on AA or SUA. Access to higher rights is controlled by User Account Control (UAC). This feature was introduced with Windows Vista.

An Administrator Account (AA) created during a fresh installation of Windows, or any account created manually by the user (AA or SUA), is limited to standard rights by UAC.

### Administrator rights (Administrative rights)

A process initiated by the user on AA or SUA may be elevated to Administrator rights and access important, new privileges. Process elevation is controlled by User Account Control (UAC). If the elevated process is initiated on AA (with standard rights), then process creation and elevation take place on AA, and the process continues to run on AA (account change not required). If it is initiated on SUA, then process creation and elevation also take place on AA, except the process no longer runs on SUA (account change SUA ---> AA, admin password required).

## H\_C smart default-deny setup

Selected Windows built-in security features can restrict Windows, MS Office, and Adobe Acrobat Reader with smart default-deny protection. These features are normally disabled in Windows. H\_C allows the user to enable them, make configuration changes, and displays the user's chosen settings. After configuration, real-time protection comes *only* from Windows' built-in security features.

## SystemSpace

The following file locations (folders and subfolders) are defined as SystemSpace and are whitelisted by default in H\_C:

C:\Windows

C:\Program Files

C:\Program Files (x86) - only on Windows 64-bit

C:\ProgramData\Microsoft\Windows Defender.

## UserSpace

All locations on the *user's local* drives (also USB external drives) which are not included in SystemSpace, are defined as UserSpace. *Network locations* are excluded either from UserSpace or SystemSpace. UserSpace locations are writable by processes running with standard rights. All executables in the UserSpace are blocked by default with H\_C's default-deny setup, except when whitelisted or initiated by the user via "Run As SmartScreen" (see also the Elevated Shell).

**PLEASE NOTE:** *The terms SystemSpace and UserSpace are specific to H\_C settings. They should not be confused with the terms 'System Space' and 'User Space', which can have a more general meaning.*

## Elevated Shell

Normally, the user on AA or SUA may initiate applications *only with standard rights*. However, this can be changed by accessing an elevated shell: PowerShell (Administrator), Command Prompt (Administrator), etc. An alternative solution is to run Total Commander via "Run As SmartScreen". The user who wants to access the elevated shell must first accept the UAC prompt. As long as the applications are initiated from the elevated shell, SRP and UAC

will ignore them (i.e., no UAC alerts or SRP restrictions). This can be useful when doing administrative tasks on the computer.

## **Questions and answers:**

### **What is conventional default-deny protection?**

It allows all installed applications and system processes but blocks by default all new executables, except those which are whitelisted. Some executables may be whitelisted automatically, e.g. by certificate or path rules. Others must first be whitelisted by the user in order to run. It is the user's responsibility to whitelist clean files.

### **What are the advantages of H\_C's smart default-deny vs conventional default-deny protection?**

Smart default-deny makes the computer more usable, while maintaining a high level of protection in the home environment. Hard\_Configurator includes three smart features:

- Forced SmartScreen (replaces "Run as Administrator"), which can be activated by the setting <Run As SmartScreen> = Administrator. Forced SmartScreen is supported on Windows 8, 8.1, and 10.
- SRP set to allow executables initiated with Administrator rights.
- SystemSpace folders/subfolders whitelisted by default.
- Some files in C:\Windows may be blacklisted by the user when using <Block Sponsors> settings.

These features allow installing most applications without whitelisting or turning OFF the protection. Furthermore, Windows Updates and system scheduled tasks can automatically bypass SRP restrictions. It is worth mentioning that Forced SmartScreen significantly extends the SmartScreen protection.

### **Are H\_C's smart features safe?**

They are very safe in the home environment, against malware in the wild. Smart features can be bypassed in Enterprises because of targeted attacks. Also, certain H\_C restrictions, e.g. "Block remote access", are not practical in enterprises.

### **Will H\_C smart default-deny setup block system processes, Windows Updates, or system scheduled tasks?**

No. System processes, Windows Updates, and system scheduled tasks are not started directly by the user. These are initiated with higher than standard rights and automatically bypass SRP restrictions configured with H\_C.

### **Will H\_C smart default-deny block updates of user applications?**

Occasionally. Some applications download the updater and run it from the Temp folder in user profile with standard rights. In this case, the update will be blocked by the H\_C default-deny settings.

### **How to update applications on *Administrator* account with H\_C's default-deny settings.**

If the update is blocked, then the application or updater should be run with Administrator rights by using "Run As SmartScreen" (on Windows 8, 8.1, 10) or "Run as administrator" (on Windows Vista or Windows 7).

### **How to update applications on *SUA* with H\_C's default-deny settings.**

There may be a problem if the application is installed in the user profile, because then an update should not be performed with Administrator rights.

Why? If it is run with Administrator rights, then it will usually search the application files in the administrator profile and not in the SUA profile. *The update will thus fail, or will be installed in the wrong user profile.*

H\_C users should check as follows:

- If the application *is not installed in the user profile*, then the update can be done on Administrator account as described above.
- If the application *is installed in the user profile* (e.g. in the folder C:\Users\Alice when the user name is Alice), then the user must:
  - turn OFF protection temporarily using "Switch Default-Deny";
  - run the update with standard rights;
  - turn ON the protection using "Switch Default-Deny".

### **Is it safe to whitelist SystemSpace?**

Generally, it is safe in smart default-deny setup. SystemSpace locations are usually not writable with standard rights. There are known exceptions, but they are covered by H\_C's <Protect Windows Folder> setting. The exploit or

malware cannot silently drop payloads to SystemSpace when running with standard rights.

### **Are all applications installed in SystemSpace?**

Usually they are, and this is recommended by Microsoft. However, some legal applications still install in UserSpace. These applications have to be whitelisted manually. For users who frequently install such applications, default-deny protection may be inconvenient.

### **What is the difference between an AA and SUA?**

Processes initiated by the user cannot run with Administrator rights on SUA. If a process running on SUA requires Administrator rights, then the UAC prompt appears, and the user must provide an Administrator password to log on to the AA. After accepting the UAC, the process is no longer running on SUA, but on AA (*user account is switched for that process only: SUA ---> AA*).

This behavior is quite different when a process is initiated on AA, because the user is not obliged to provide the Administrator password. Instead, the UAC prompt asks for a simple "Yes" or "No". After accepting the UAC prompt, the process continues running on the same AA (user account is not switched for that process).

### **Is SUA more secure than AA?**

Yes, most definitely. On SUA, unelevated processes (running with standard rights or lower) do not share the same user account as elevated processes. This is not true on AA. It is much easier to exploit something when both unelevated and elevated processes are running on the same AA account. Malware or exploits cannot run with Administrator rights on SUA - they must first escape to an Administrator account. This is hardly possible, because Microsoft usually patches any system vulnerabilities which might allow malware to escape from SUA. H\_C's smart default-deny setup relies on blocking unelevated programs (running with standard rights), so SUA is an ideal companion to H\_C.



## **When should SUA be used instead of AA?**

SUA should be considered a vital part of any security solution when using *a* vulnerable system, or popular & vulnerable software. However, it is not necessary to use SUA with H\_C's smart default-deny *when Windows 10 and all installed software are updated regularly*. A well maintained system which includes H\_C is a dead end for malware/exploits in the home environment.

## **Does Forced SmartScreen work well on SUA?**

Yes, if the application installs in SystemSpace (usually in C:\Program Files). There can be a problem if it installs in user profile, which lies in UserSpace. Why? Because with H\_C smart default-deny, Forced SmartScreen uses Administrator rights. Applications which are intended to install in SUA profile, are installed in Administrator profile - even when the installation is initiated from SUA. The user on SUA cannot run applications from Administrator profile, since Windows isolates user profiles from one another. In this case, the user must disable default-deny protection temporarily, and install the application without using "Run As SmartScreen".

## **How to install applications on SUA:**

1. Run the application installer by using "Run As SmartScreen" option from the Explorer right-click context menu.
2. Check the default installation folder.
3. If it is in the Administrator profile, then cancel the installation and continue with Steps #4-7. If not, then continue with the installation and skip Steps #4-7.
4. Use "Switch Default-Deny" to turn OFF the protection temporarily.
5. Install the application normally (by left mouse-click or pressing the Enter key).
6. Whitelist the application in the UserSpace.
7. Use "Switch Default-Deny" to turn ON the protection.

## **Why Recommended H\_C settings are best as a starting setup?**

New users of default-deny protection should be aware that it requires more skill than using an AV alone. Please use *only* the Recommended H\_C settings along with your AV, until you are comfortable and familiar with H\_C. Prema-

turely adding advanced H\_C settings or more security software to this configuration may lead to complications, and user discouragement, with default-deny protection.

### **Who should consider applying advanced H\_C settings?**

Recommended H\_C settings provide strong preventive protection against running malware in the system.

Advanced H\_C settings can mitigate the malware or an exploit which is already running in the system. When using well-patched software on updated Windows 10, advanced settings are not required.

### **Will advanced settings spoil the system?**

On most computers, even maximum H\_C settings cannot break anything important in the system, but some applications may be not fully functional. Enabling advanced settings will usually require more whitelisting, more researching of logs, etc., and may be annoying for most users. If so, then the user should restore Recommended settings.

### **How to restore Recommended settings.**

1. Press <Recommended SRP> left panel button,
2. Press <Recommended Restrictions> right panel button,
3. Press <APPLY CHANGES> button.

Restoring the Recommended settings preserves the user's whitelisted entries and blocked file extensions.

**PLEASE NOTE:** *If SRP is deactivated, then order of the pressed buttons is important! Pressing the buttons in another order (for example 2,1,3) will prevent installation of new applications. This kind of a locked setup is much more restricted as compared to Recommended settings.*

### **How to apply advanced H\_C settings.**

Advanced settings can be activated by turning ON additional individual H\_C options, or by loading the setting profile (<Load Profile> button).

It is advisable to begin with the Recommended\_Enhanced profile. This may be done by loading the file: Windows\_\*\_Recommended\_Enhanced.hdc, where the asterisk replaces the Windows version (7, 8, or 10). This will enable

the Recommended settings, and some well known Sponsors will be blocked (including Script Interpreters).

**PLEASE NOTE:** *It is not advisable to use multiple advanced settings at once. When using advanced settings, the user should occasionally check for blocked entries (<Tools><Blocked Events / Security Logs>). This is because sometimes there is no alert when a process is blocked by Windows policies.*

## **What is a Sponsor?**

A Sponsor is an executable from the SystemSpace (usually from C:\Windows), that can be used by an attacker to bypass default-deny protection. Sponsors are frequently used in targeted attacks on organizations and businesses, especially via exploits. Blocking some Sponsors in the home environment can be important for people who use a vulnerable system or software. In H\_C's Recommended settings, Windows Script Host Sponsors (wscript.exe and cscript.exe) are blocked by SRP. Furthermore, PowerShell Sponsors (powershell.exe and powershell\_ise.exe) are restricted by Constrained Language mode in Windows 10 and blocked by SRP in Windows Vista, 7, 8, 8.1. These Sponsors are the most popular Script Interpreters. Some other Interpreters (mshta.exe, hh.exe, wmic.exe, scrcons.exe) can be blocked in H\_C by <Block Sponsors> option. Unfortunately, a few of them can be used occasionally by older software, usually those related to peripherals. Applications and web browser plugins may also use Interpreters for some actions, though most applications and plugins do not use them at all. In H\_C, Sponsors are blocked for processes running with standard rights, but allowed for administrative processes running with higher rights.

## **Can wildcards be used for whitelisting files and folders?**

Yes, they can. Here are some examples, where the random characters are replaced by wildcards to whitelist the particular EXE file:

- C:\Users\Alice\Fly2theMoon\App.1928-0928\setup\_101989873.exe
- C:\Users\Alice\Fly2theMoon\App.????-????\setup\_?????????.exe
- C:\Users\Alice\Fly2theMoon\App.\*\setup\_?????????.exe
- C:\Users\Alice\Fly2theMoon\App.\*\setup\_\*.exe
- C:\Users\Alice\Fly2theMoon\App.\*\\*

Those rules (except the first) are correct, and the EXE file will be whitelisted even when the random numbers will change after some time. The last rule is most general because it will whitelist many other files and folders, for example: C:\Users\Alice\Fly2theMoon\App.malware\virus.js