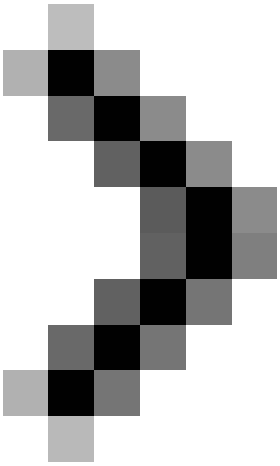
Default denies all applications (Part 2)

Since Windows XP, administrators worldwide have had many options to define Software Restriction Policies (SRP) for their client computers to control what software is allowed, software Come on



Default denies all applications (part 1)

Since Windows XP, administrators worldwide have had many options to define Software Restriction Policies (**SRP**) for their client computers to control what software is allowed, software Which is not allowed to run. In fact, too few organizations have added this function even though it can provide a very high level of security if properly set up. In the second part of this article we will look at how to add what has been mentioned as 'software filtering policies'.

Danger area

Before we start, we need to point out some important points before introducing SRP on computers that you should plan and test them thoroughly. This can be done in Active Directory (**AD**) by isolating the test computer

or user object in the Organizational Unit (**OU**), or by setting the ' *Apply Group Policy* ' permission on the Group Policy Object. (**GPO**) for test computers or user accounts. SRP can even be used on stand-alone computers if you don't want to touch the production environment at all. Here we recommend using virtual machines to perform basic tests and 'production-like' computers for the final tests. After the testing process, the SRP should be implemented on pilot users in groups - it is better to use small steps rather than rushing to "disaster" !.

Whole process

The design or addition errors of the whole process can cause great frustration for users, and even lose money and productivity, so beware when you do this work. . This process requires a lot of work planning, testing and maybe some daily maintenance to be included in the production environment. The list below shows you a brief review of what to have in mind when bringing SRP into a Windows environment.

When disabling SRP settings, other decisions must be made as in the example below :

- Deciding between users and computer GPOs: What AD objects should SRP policies apply?
- Decide between blacklisting Blacklisting (**BL**) and Whitelisting (**WL**) (WL method is recommended if any)
- If using BL method, you must create a list of rejected components.
- If you use the WL method, you must create a list of the components used.
- Decide which SRP options to use (coercion, specify file types, administrator exceptions .)
- Decide what kind of rule to use (path-path, HASH, certificate-certificate or Internet zone - Internet zone)

When checking the SRP setup, different steps must be taken, for example :

- Check basic functions in virtual libraries (using Virtual PC / Server, VMware or similar)
- Test functionality in your environment with 'like production' computers (and user accounts).
- Check with small groups of users who lead the way in steps 5-10 a few weeks, then increase.
- Continue with the next navigation group after successfully adding to the previous navigation users.
- Be sure to check all other types of users and different types of computers you need to secure with SRP.

- Be sure to check for updates to scripts like patching an operating system, upgrading third-party applications, etc.

- Focus on performance on different hardware in your organization. Adding SRP will affect the system performance more or less, this effect depends on how it is added.

- Sometimes when these applications launch other applications, make sure this is strictly controlled.

- By default, Desktop and Start Menu shortcuts (. **LNK**) are locked, change them if necessary.

- Make sure that any admin scripts (such as the login script in SYSVOL / NETLOGON) work fine.

Before introducing the SRP, some procedures need to be considered :

- Introducing workflows on how applications, upgrades must be tested, enabled and included on the network, so people learn about the need for procedures.

- Ensure that managers know SRP experts and researchers they agree to decide to introduce it.

- An information feedback plan is required to quickly remove SRP GPOs for certain users, groups, computers, or pages, etc.

- User training: inform the user why the SRP is important and what procedures to acquire the new software

Other links to SRP

The basic Software Restriction Policy software restriction policy configuration is based on the following steps:

1. Create a User or a Computer GPO and locate it on Site, Domain or OU (or as a local policy) and enable SRP within Group Policy. SRP settings are here (see Figure 1):

   Computer Configuration | Windows Settings | Security Settings | Software Restriction Policies
   User Configuration | Windows Settings | Security Settings | Software Restriction Policies

   First SRP is included in FPO, the ' **New Software Restriction Policies** ' option will be provided.

Figure 1

2. Set the **Default Security Level** . Figure 2 shows how the level is set by right-clicking on the level you want and selecting ' **Set as default** '.

    - The default level is ' **Unrestricted** ' which means that all software can run and additional rules for disabling software are created - this is also known as **Blacklisting** blacklist.

    - The best protection level is ' **Disallowed** ' which means that no software can run and additional rules for allowed software are created - this can be interpreted as **Whitelisting** .

    - By default, the system creates several rules that allow the operating system to run without any inconvenience.

If the rules are removed or changed without thinking, it can make the system unusable.

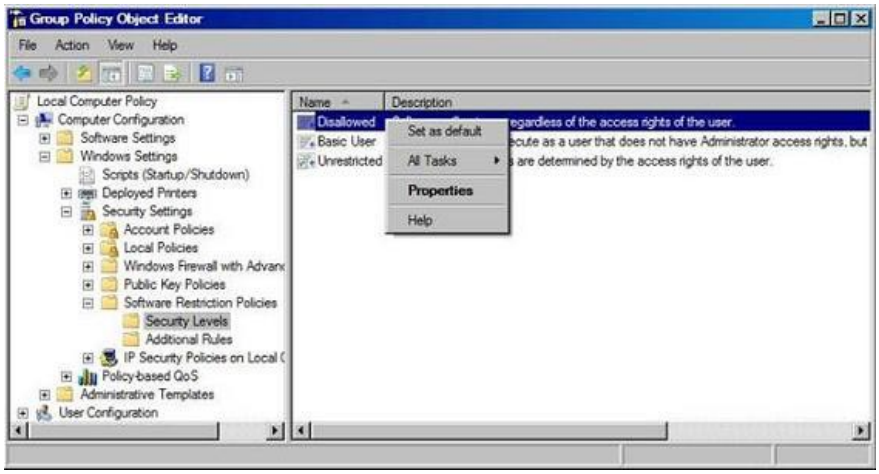


Figure 2

3. With Windows Vista and Longhorn we have added a new level of ' **Basic User** ', which allows programs to execute as a user without administrator access, so users can access it. Access resources only with the usual mode for users.

   File types can be removed or added to fit any environment, but the most common list of file types is: **BAT** , **CMD** , **COM** , **EXE** , **HTA** , **LNK** , **MSI** , **OCX** , **PIF** , **REG** & **SCR** and in addition are: ADE, ADP, BAS, CHM, CPL, CRT, HLP, INF, INS, ISP, MDB, MDE, MSC, MSP, MST, PCD, SHS, URL, VB & WSC.

   Note : The status of the Designated Files Type Properties dialog box has 'in ***addition*** to the *standard program file types* , such as EXE, DLL and VBS' - I was unable to get the entire list but still confirm that **VBS** , **VBE** , **JS** and **JSE** will be locked even if they are not on the list. In our opinion, here is better if there is a single list so that administrators around the world can change it if necessary.

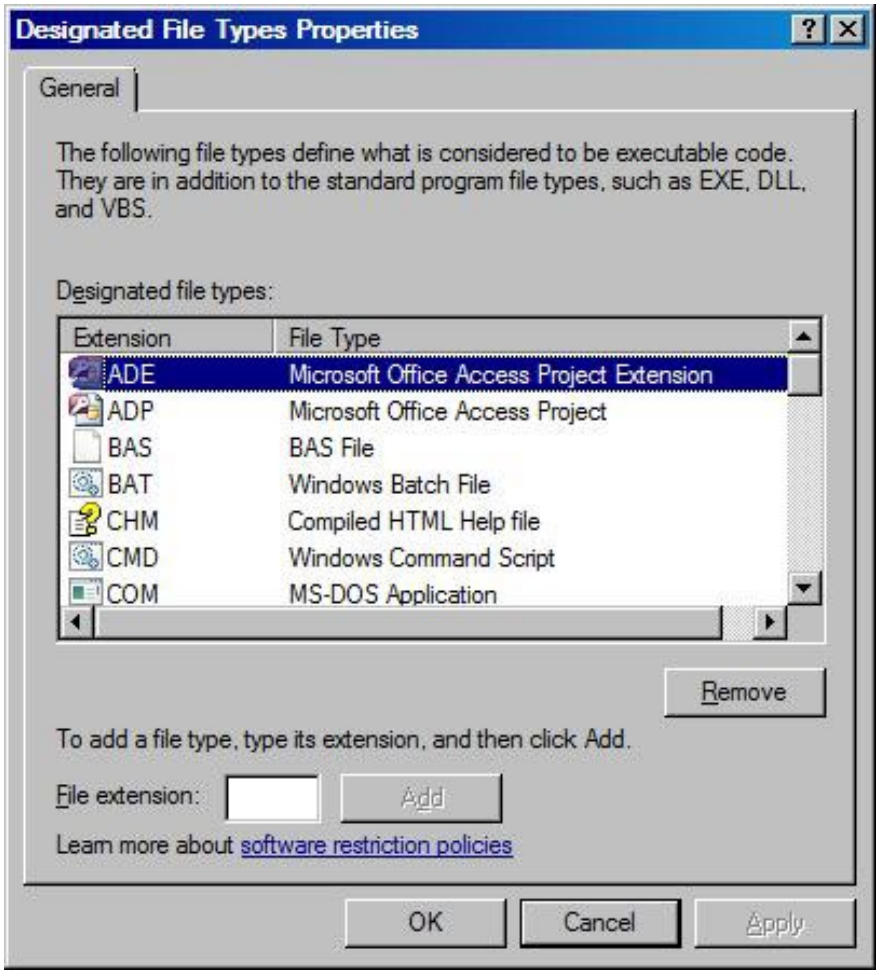


Figure 3

4. Set **exceptions** for Default Security Level. They are known as ' **Additional Rules** ' additional rules. You can see the ' **Additional Rules** " additional rules for more details.

5. ' **Enforcement Properties** ' configuration. See Figure 4, including:

   - ' **All software files** ': We have the option to check **DLL** 's (Dynamic Link Libraries) when they are also executed. This is not set by default and there will be some effects on both performance and planning / addition / maintenance tasks.

   - ' **All users except local administrators** ': Here we can choose whether to apply SRP to local administrators. By default all users must be found by SRP. This policy option only relates to computer policies.

- ' **Enforce certificate rules** ': Option to choose whether to use certificate rules. Note that the status in the Figure 4 dialog box 'Certificate rules will negatively affect the performance of your machine' means that the certificate principles will adversely affect the performance of the machine.

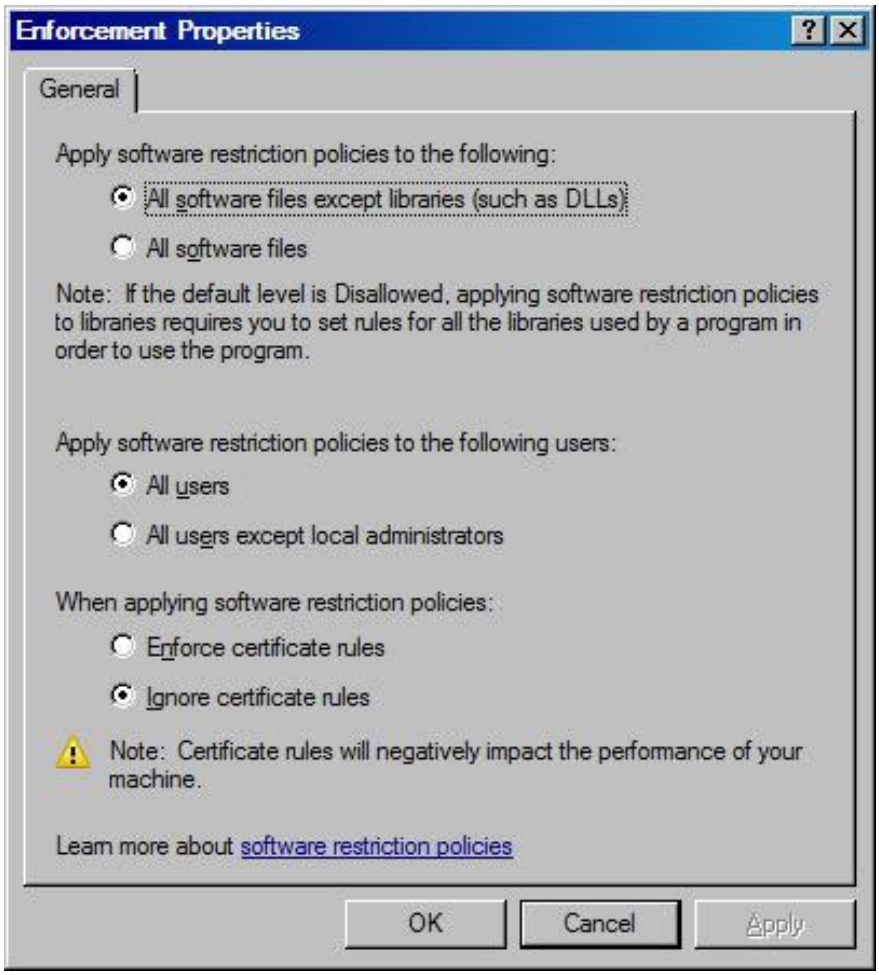


Figure 4

6. Configuring ' **Trusted Publishers Properties** ', this configuration is known as authentication policy options (see Figure 5). In this dialog box we can choose who can choose Trusted Publishers for certificate rules.

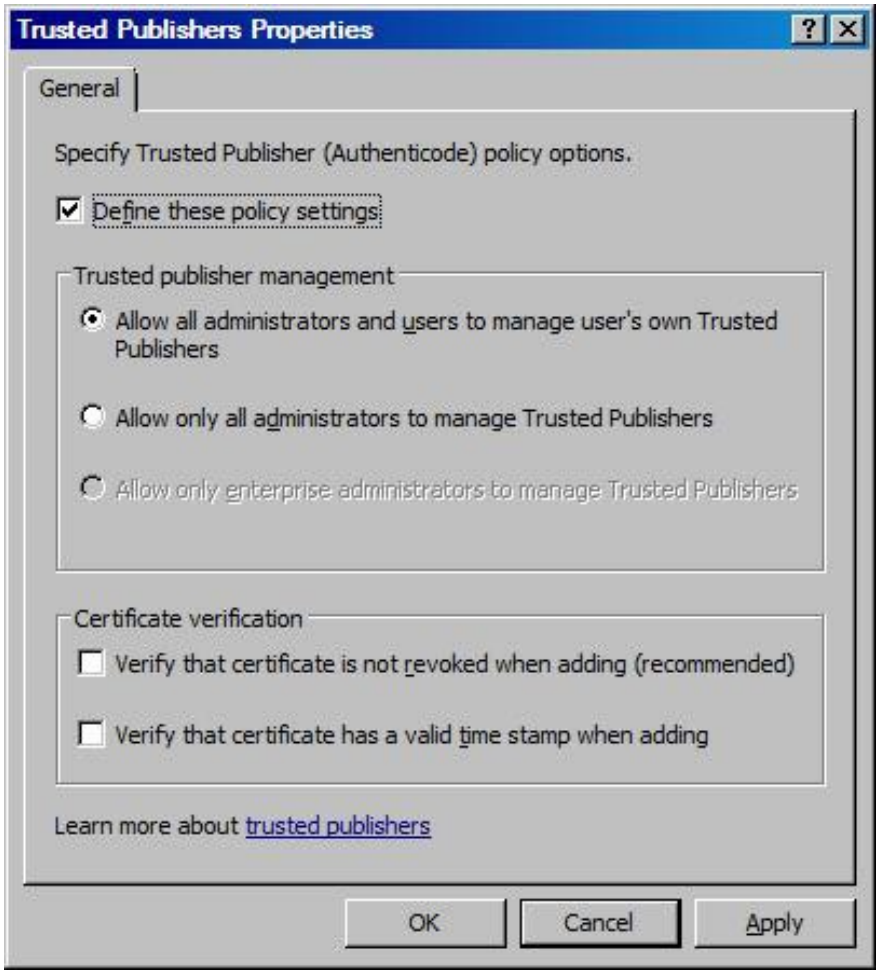


Figure 5

Additional principles

When configuring additional rules, how to recognize software is decided. We have 4 different software identification methods that are:

1. **HASH principles**

   HASH is maintained cryptographic fingerprints regardless of file name and location. It is particularly good to use WL, but it is also ineffective for BL (the reason for this was introduced in many articles): MD5 or SHA-1 HASH provides binary file software verification 'ProduKey .exe ', so it allows a specific HASH value to run to ensure that only that version can execute. However, not allowing specific HASH only affects one version (or compilation) of an application and an intelligent user can change this file quite easily and it will be identified as another software. The word 'unknown' is very important in this regard - it's really the key to deciding between BL and WL - do you want users to be able to execute software that we don't know? Not good?

   If a user cannot run an older version of a specific application, assume that it is corrupted and cause a

malfunction of your system, using this HASH restriction rule would be a good decision. . You should also note that a new version of a certain application will always return a new HASH value that is allowed or not allowed.

2. **Certificate principles**

The rules for using HASH certificates are marked and provide strong software verification, but when we trust a certain certificate, we will trust all the software that is assigned to the certificate. only that specific. This can be a good or bad thing. It might be good if we have an application from the third group, someone has marked all the important files in the application (may include DLLs), so instead of creating some integers HASH rules, we just need to create a single rule that trusted the certificate and ran it. However, should we rely entirely on digital certificates that are used to signify some tools from Microsoft (or from third parties) - now users can run all applications. marked with that specific certificate . Obviously the problem here is how to know exactly what application has been marked with that certificate? Without that information we will not be able to know what is allowed. Instead of allowing single applications for users, we can allow hundreds of applications from vendors to run on our systems.

Checking the certificate principles can be done using the tools: File Signing Tool (Signcode.exe) & Certificate Creation Tool (Makecert.exe).

3. **Path rules**

This principle is the most common and easiest principle we have. This rule can deny or allow a file on a specific location (eg ' **C: ScriptsScript.VBS** '), a file name (eg ' **Script.VBS** '), a directory (for example ' **C: Scripts** '), a **UNC** path (eg ' **SERVERSHAREFile.VBS** ') or a registry path (**'% [Registry Hive] [Registry Key Name] [Value Name]%** '). Can these principles be used in environment variables (' **% WINDIR%** ') and replace ' **?** '= a character (eg' **SERVER ?? ShareScript.VBS** ') and' ***** '= a character cluster (eg' *** .VBS** ').

If you are using this principle, some things to consider are :

- If the directory path is set, this also affects all possible executables in subdirectories.

- If the user has write access to the folder set to *Unrestricted* , it is also possible to copy any file to that directory and execute it from there. The design must be cautious about this problem, perhaps by including **ACL** (Access Control List) settings by using Group Policy.

- If a user with write access to an executable specified as *Unrestricted*, then the user can override that executable on another user to corrupt the SRP.

- If a directory path has environment variables such as **% TEMP%** , even a restricted user can change user environment variables using the **SET** command for another path (**SET TEMP = C: MyFolder**) - this can very well lead to accidental user control of certain software that can be run by copying the executable or script to that path.

As above, users can try changing names or transferring unauthorized files - or overwrite unrestricted files to corrupt SRP - this explains why HASH or certificate principles Often considered the best choice.

4. **Internet zone**

   Internet Explorer security zones can be used to control software installation and apply only to Windows Installer packages (**MSI**) that are run from one of the default Internet zones. Those principles are not commonly used.

   When many rules are used they will be evaluated to match what is above, the default rule will be evaluated as the final rule - the most commensurate will be prioritized.

Restrictions of SRP

Some limitations of SRP must be considered. The scope of SRP is not for all operating systems as you expect. SRP does not apply to the following sections:

- Other drivers or kernel mode software are installed
- Any program installed by the local SYSTEM account
- Macros inside Microsoft Office documents (we have many ways to lock them using group policies)
- Programs are written for the common language (these programs use code access security policies).

Conclude

SRP requires a lot of time, workflow and upgrades, but besides it gives you a higher level of security than the default settings allow. Some environments may not be suitable for SRP, but our guess is that most networks have multiple computers and users of SRP technology are correct.

It is necessary to make decisions, learn and work hard to supplement this default rejection policy, which is why it is rarely used . However there is one thing clear here. The fact that adding policies in the right way ensures that network administrators can sleep better.

You finished reading the article "**Default denies all applications (Part 2)**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

© 2019 TipsMake.com