National Security Agency

Information Assurance Directorate

Vulnerability Analysis and Operations

Systems and Network Analysis Center

# Application Whitelisting using Software Restriction Policies

Version 1.1
August 2010

# Contents

# Figures

# Tables

# Abstract

Software Restriction Policies (SRP) enables administrators to control which applications are allowed to run on Microsoft Windows. SRP is a feature of Windows XP and later operating systems. It can be configured as a local computer policy or as domain policy using Group Policy with Windows Server 2003 domains and later. Using this guide, administrators can configure SRP to prevent all applications in their domain from running except applications they explicitly allow. Utilizing SRP as an application whitelisting technique significantly increases the security posture of the domain by preventing many malicious programs from executing.

For some additional government-only implementation details, please obtain the addendum to this document at https://www.iad.gov/library/snac.cfm (DoD PKI required), at http://www.iad.smil.mil/resources/library/ on SIPRNet, by request from your IAD Customer Advocate, or by contacting the NSA IA Service Center at NIASC@nsa.gov or the NSA Systems and Network Analysis Center (SNAC) at snac@radium.ncsc.mil.

# Introduction

The amount of malware on the Internet increases in volume and variety every day. Malware developers and antivirus vendors are in a never-ending arms race. Malware authors continuously modify their creations so they are not detected, and antivirus vendors update their signatures daily to detect new malware variants. Defending against these threats by blocking every known malware sample, a technique known as *blacklisting*, is a reactive technique that does not scale well to the increasing volume and variety of malware. It also does not protect against unknown malware. Many attacks use previously unknown vulnerabilities, also known as zero-day vulnerabilities, which cannot be prevented with blacklisting techniques.

Government and corporate networks are prime targets for attackers. They contain valuable proprietary or sensitive information and have a large, diverse attack surface for an adversary to exploit. As operating systems have become more locked down, attacks have shifted from operating systems to applications. This change has left each individual user, and the applications they use, as the main attack vectors into the network.

Application *whitelisting* is a proactive technique where only a limited number of programs are allowed to run, while all other programs are blocked from running by default. Below are some example scenarios that may be mitigated by using application whitelisting:

- A user receives an enticing email with a link to a program that looks like a greeting card or a streaming video viewer that executes a hidden malicious program.
- A user views a web site that silently exploits a previously unknown or unpatched vulnerability in the web browser, or third-party browser add-on, and then downloads a malicious program to further compromise the network and steal data.
- A user opens a document that exploits a vulnerability in the document viewer and an embedded malicious program is extracted and unknowingly executed.
- A user inserts removable media, such as a USB thumb drive, into their computer that automatically executes a malicious program.
- A user installs a program without notifying the administrator, so the program remains unpatched after a critical vulnerability is publicly disclosed and then is exploited by malware.

Since none of the malicious programs in the above scenarios are included in the list of allowed programs, they would not be executed if application whitelisting was enforced. Whitelisting makes it more difficult for attackers to compromise a network because they must exploit one of the allowed programs on the victim's computer or circumvent the whitelisting mechanism to perform a successful attack. Even if an allowed program is exploited, further malicious activity may still be blocked by the whitelisting mechanism.

Application whitelisting is not a replacement for traditional security software. It should be used as one layer in a defense-in-depth solution. For an application whitelisting solution to be effective:

- All executable code must be blocked by default so only approved programs can run.
- Users must not be allowed to run programs from directories where they can save files.
- Users must not have administrator privileges.

Microsoft Windows operating systems include a feature called Software Restriction Policies (SRP). Administrators can configure SRP as an application whitelisting solution where only specific executables are allowed to run while all other executables are prevented from running. SRP can also limit which application libraries are allowed to be loaded by executables. SRP is a built-in feature of Windows XP and later operating systems. It can be configured as a local computer policy or as domain policy using Group Policy with Windows Server 2003 and later domains.

## About this Guide

This guide describes SRP settings recommended by the NSA Information Assurance Directorate's (IAD) Systems and Network Analysis Center (SNAC) and provides administrators with a walkthrough for implementing the settings.

**Using SRP as an application whitelisting solution will not stop all malicious software. It provides an additional layer in a defense-in-depth strategy. The intent of this guidance is to prevent users from unknowingly or accidentally executing malicious code.**

Please read this document fully before implementing the guidance. Any configuration changes should be validated on a test network or on a small set of test computers to ensure the settings are correct before making changes to the entire domain.

There are many references that describe how to configure SRP. This document is not meant to replace those resources and does not explain all possible configuration options. For more information about SRP, consult the Microsoft documents "Using Software Restriction Policies to Protect Against Unauthorized Software"[1] and "Windows Server 2003 Technical Library: Software Restriction Policies."[2]

For additional government-only implementation details, please obtain the addendum to this document at https://www.iad.gov/library/snac.cfm (DoD PKI required), at http://www.iad.smil.mil/resources/library/ on SIPRNet, by request from your IAD Customer Advocate, or by contacting the NSA IA Service Center at NIASC@nsa.gov or the NSA Systems and Network Analysis Center (SNAC) at snac@radium.ncsc.mil.

## Known Issues

There is a known bug in the SRP implementation for certain versions of Windows. A hotfix from Microsoft is available to fix this bug and should be applied to all affected computers before enabling SRP. More information is available in the Implementation section and in Appendix A of this guide.

Some minor usability issues may occur when using SRP, especially if the hotfix is not applied, that could annoy users. One example is that double-clicking a document on a network share may not launch its associated document viewer application. Another example is software update mechanisms that require users to manually apply patches may no longer function once SRP whitelisting is enforced. **Most automatic update mechanisms are not affected by SRP and will continue to function correctly**. Due to these issues, SRP settings should be thoroughly tested on a limited set of computers that have all deployed software before being applied in a production environment. See Appendix A for more information about these issues.

## Host and Network Performance Concerns

Using path-based SRP rules, as recommended in this guide, have shown no noticeable performance impact on hosts after much testing. Other SRP rule mechanisms, such as file hash rules and certificate rules, may provide

[1] "Using Software Restriction Policies to Protect Against Unauthorized Software"
http://technet.microsoft.com/en-us/library/bb457006.aspx

[2] "Windows Server 2003 Technical Library: Software Restriction Policies"
http://technet.microsoft.com/en-us/library/cc779607.aspx

greater security benefits than path-based rules, but they increase the performance impact on hosts. File hash rules are also much more difficult to manage than path-based rules due to the need to constantly update the list of hashes whenever any files are installed or updated. Certificate rules are also of limited use since many software publishers do not digitally sign all of their applications' files.

In general, the increase in network traffic due to using SRP is minor. The group policy object that contains the SRP rules will only be a few kilobytes larger than the default group policy object size. The more rules that are defined, the larger the policy will become, but a realistic range is 10KB-300KB extra depending on how many rules are added. Since policies are only downloaded to a host when needed, network traffic impact is infrequent and minimal.

# Implementation

The following are the recommended steps for implementing SRP in an Active Directory domain and are described in further detail below. Implement SRP by following these steps:

1. Audit the domain to determine which applications are running on domain computers.
2. Configure SRP to run in whitelisting mode.
3. Decide which applications should be allowed to run and create additional SRP rules as needed.
4. Test the SRP rules and create additional rules as necessary.
5. Deploy SRP to successively larger Organizational Units until SRP is applied to the whole domain.
6. Monitor SRP on an ongoing basis and modify the rules when appropriate.

## Audit the Domain to Determine which Applications are Running

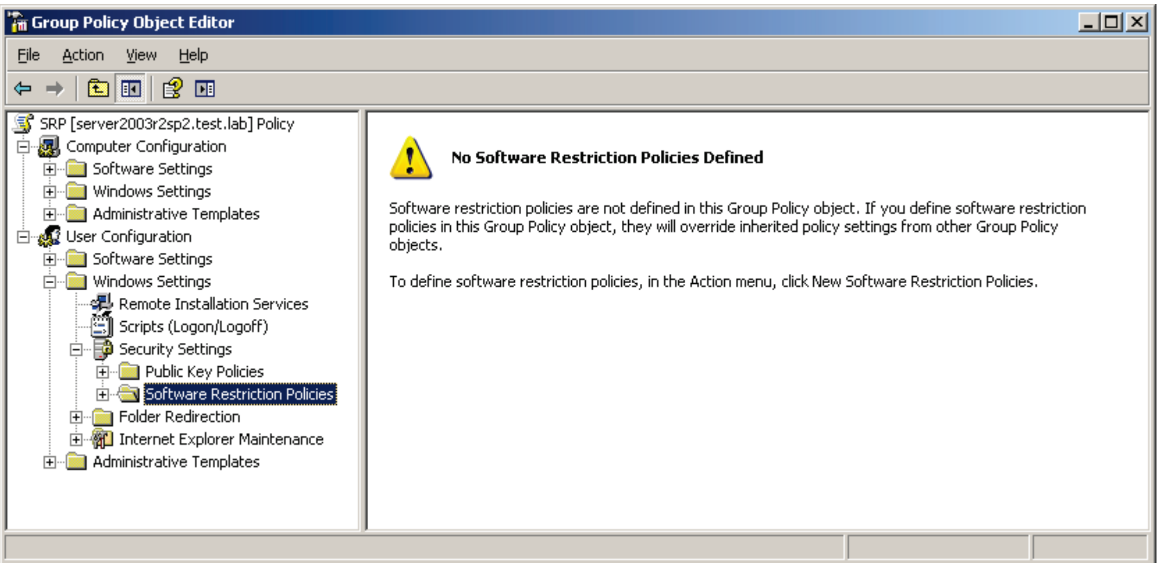
Before applying Software Restriction Policies, it is important to know which applications are running on domain computers. An audit of the domain is essential for creating a set of robust SRP rules that will enable users to continue running authorized programs that are stored in non-default locations. Programs running from the paths specified by the SYSTEMROOT and PROGRAMFILES environment variables, usually C:\Windows\ and C:\Program Files\, can generally be ignored during an audit, except if those programs load libraries from other paths or if they contain subfolders writable by regular users. Those paths are the built-in allowed paths for the default SRP settings. Tools such as Windows Sysinternals Process Monitor and Process Explorer and WMI scripting techniques can be used to list the currently running executables and libraries on computers.

## Configure SRP to Run in Whitelisting Mode

The following section explains how to configure SRP in an Active Directory environment to run in path-based whitelisting mode with the most secure settings. The screenshots are for Windows Server 2003, but differences for Windows Server 2008 have been noted in the text. To apply SRP to the domain:

1. Create a new Group Policy Object (GPO). Give the GPO a name that can be easily associated with SRP.
2. Open the newly created GPO for editing in the Group Policy Object Editor in Windows Server 2003 or the Group Policy Management Editor in Windows Server 2008.
3. Go to **User Configuration** → **Windows Settings** → **Security Settings** → **Software Restriction Policies** as shown in Figure 1. When configuring SRP for the first time, the message shown on the right side of Figure 1 will be displayed. Note that in Windows Server 2008, the **Policies** node exists between the **User Configuration** and **Windows Settings** nodes.

By creating the policy as a user policy rather than a computer policy, the SRP settings can be applied to specific users or groups so that whitelists can be enforced at a more granular level. For example, accounts with domain administrator privileges can be put in an Organizational Unit (OU) that does not have the SRP settings applied. This allows domain administrators to be exempted from the SRP settings, while domain user accounts, even those with local computer administrator privileges, are restricted by them. Another advantage is that a general whitelist can be applied domain wide and then more specialized whitelists can be applied to specific OUs depending on the needs of different parts of an organization.



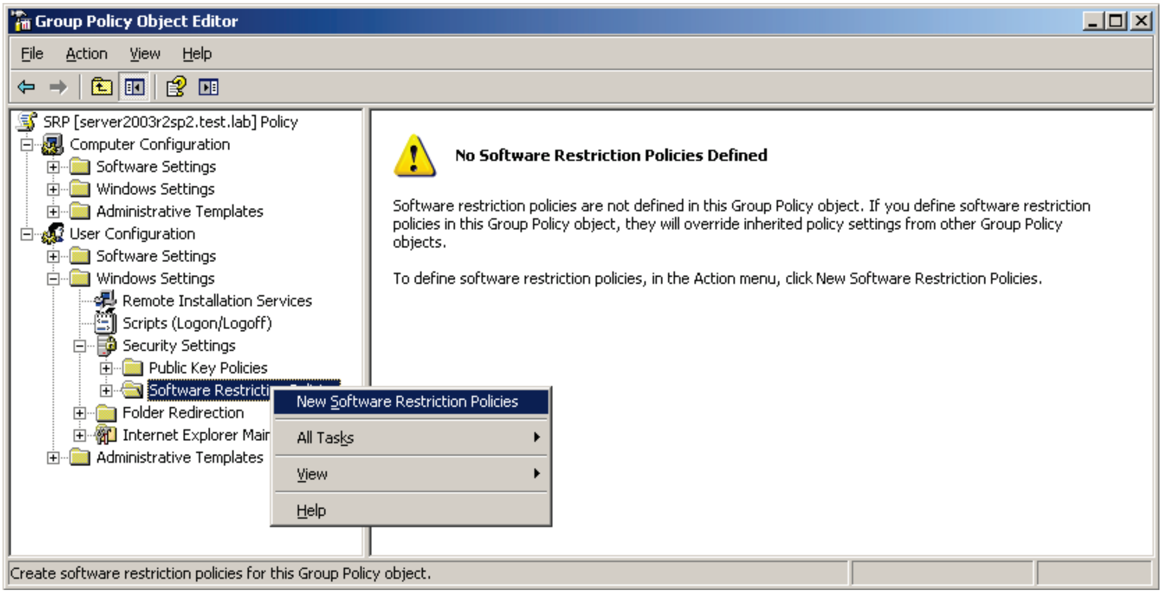
**No Software Restriction Policies Defined**

Software restriction policies are not defined in this Group Policy object. If you define software restriction policies in this Group Policy object, they will override inherited policy settings from other Group Policy objects.

To define software restriction policies, in the Action menu, click New Software Restriction Policies.

**Figure 1: Select the Software Restriction Policies object in the Group Policy Object Editor.**

4. Right-click on the **Software Restriction Policies** folder and select **New Software Restriction Policies** from the menu as shown in Figure 2.



**No Software Restriction Policies Defined**

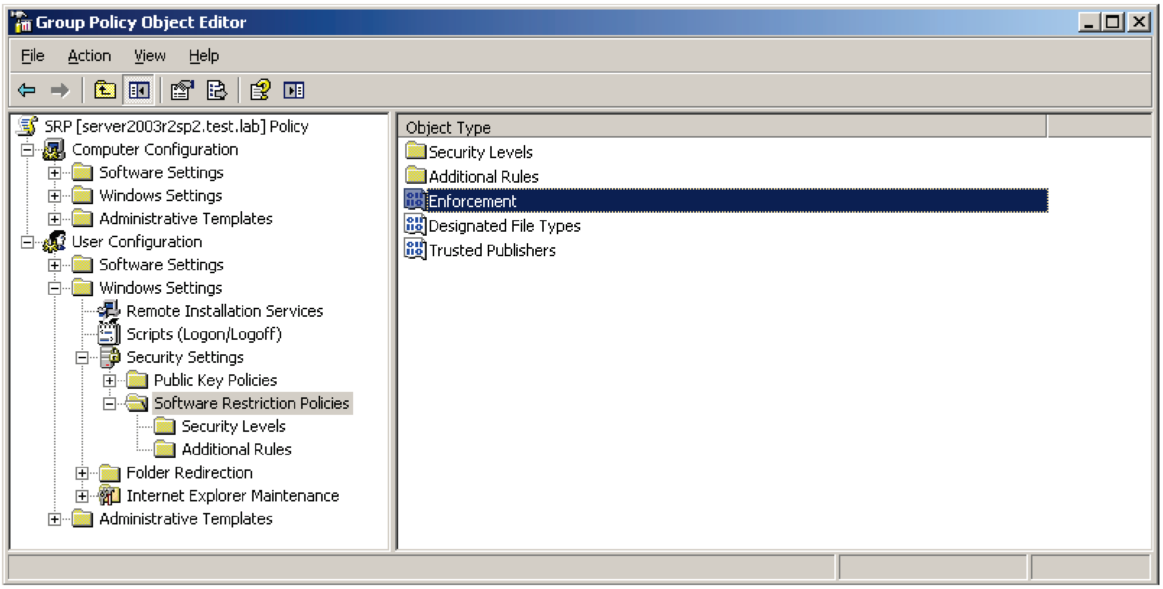
Software restriction policies are not defined in this Group Policy object. If you define software restriction policies in this Group Policy object, they will override inherited policy settings from other Group Policy objects.

To define software restriction policies, in the Action menu, click New Software Restriction Policies.

Create software restriction policies for this Group Policy object.

**Figure 2: Select New Software Restriction Policies from the right-click menu.**

5. Once the new policy is created, select the **Enforcement** setting that is listed in the right-hand pane, as shown in Figure 3, when the **Software Restriction Policies** folder is selected.

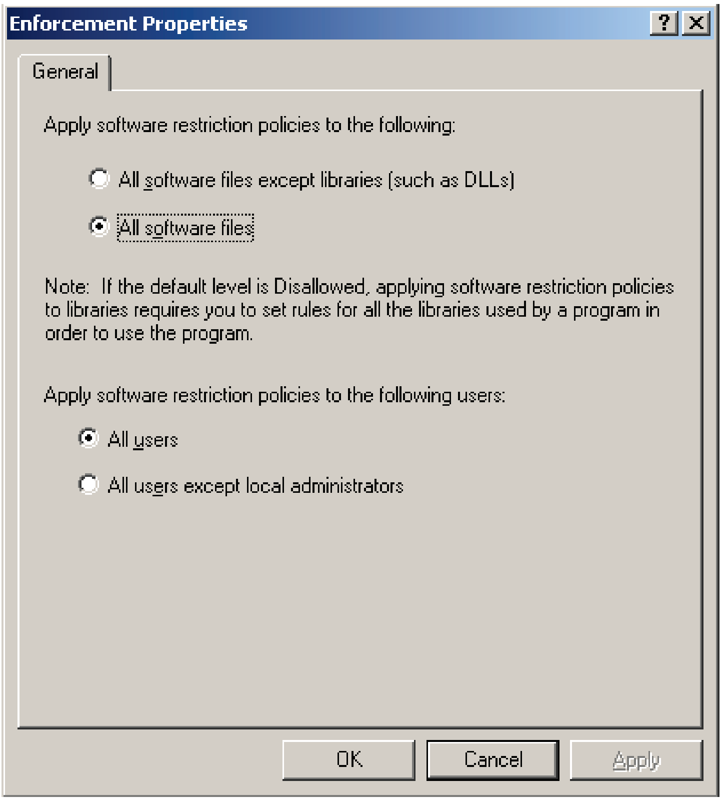


SRP [server2003r2sp2.test.lab] Policy
- Computer Configuration
  - Software Settings
  - Windows Settings
  - Administrative Templates
- User Configuration
  - Software Settings
  - Windows Settings
    - Remote Installation Services
    - Scripts (Logon/Logoff)
    - Security Settings
      - Public Key Policies
      - Software Restriction Policies
        - Security Levels
        - Additional Rules
    - Folder Redirection
    - Internet Explorer Maintenance
  - Administrative Templates

| Object Type |
|---|
| Security Levels |
| Additional Rules |
| Enforcement |
| Designated File Types |
| Trusted Publishers |

**Figure 3: Select the Enforcement policy object.**

6. Double-click the **Enforcement** setting to open the **Enforcement Properties** dialog as shown in Figure 4. Select the **All software files** radio button so SRP will be applied to both executables and libraries. Select the **All Users** radio button so SRP will be applied to all domain users including local administrators. Click the **OK** button when finished.



Apply software restriction policies to the following:

○ All software files except libraries (such as DLLs)

● All software files

Note: If the default level is Disallowed, applying software restriction policies to libraries requires you to set rules for all the libraries used by a program in order to use the program.

Apply software restriction policies to the following users:

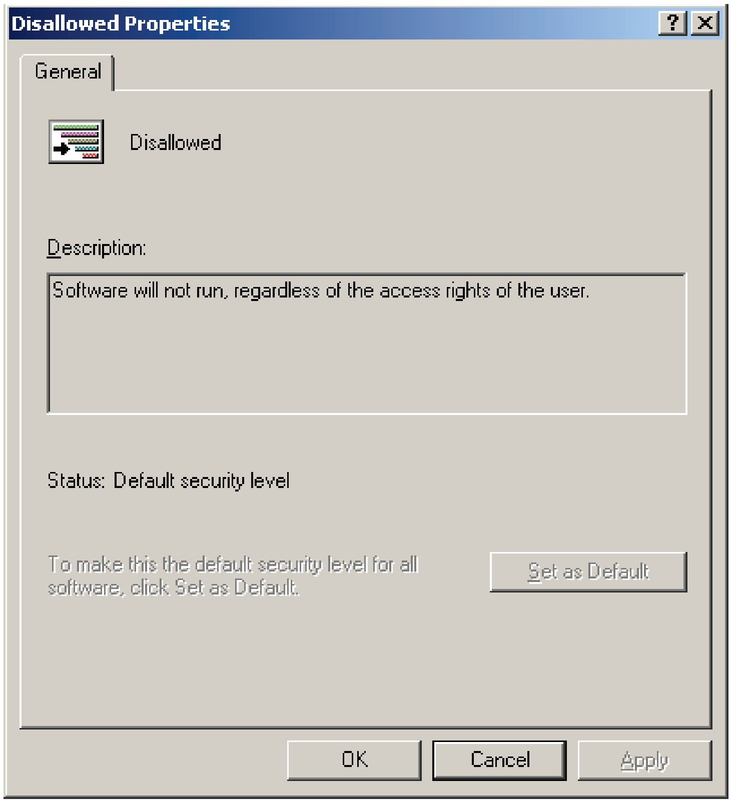
● All users

○ All users except local administrators

**Figure 4: Select the most secure options in the Enforcement Properties dialog.**

7. In the Group Policy Object Editor, click on the **Security Levels** folder to configure the SRP operation mode. SRP can operate in blacklist mode or whitelist mode. Blacklist mode is where *all* applications are allowed to run except the ones an administrator specifically denies. Whitelist mode is where *no* applications are allowed to run except the ones an administrator specifically allows. Configuring SRP to

use whitelist mode is the most secure and recommended mode. Double-click the **Disallowed** security level and then click the **Set as Default** radio button as shown in Figure 5 to configure SRP in whitelist mode.



Disallowed Properties

Disallowed

Description:

Software will not run, regardless of the access rights of the user.

Status: Default security level

To make this the default security level for all software, click Set as Default.

**Figure 5: Set Disallowed as the default security level.**

After clicking the **Set as Default** radio button, a dialog may appear with the warning:
*The default level you selected is more restrictive than the current default security level. Changing to this default security level may cause some programs to stop working. Do you want to continue?*
If this dialog appears, click the **Yes** button.

After completing the above steps, programs are not allowed to run except for ones in paths specified by the SYSTEMROOT and PROGRAMFILES environment variables, usually C:\Windows\ and C:\Program Files\. These path rules are automatically added when the **Disallowed** security level is set as the default. The rules can be viewed by clicking on the **Additional Rules** folder and are also documented in Table 1.

| Server | Default Rules |
|---|---|
| Windows Server 2003 | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%<br>%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%*.exe<br>%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%System32\*.exe<br>%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ProgramFilesDir% |
| Windows Server 2008 | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%<br>%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ProgramFilesDir% |

**Table 1: Default SRP Path Rules**

There are some subfolders within the default built-in allowed paths which allow users to create or modify files. This would allow users to circumvent SRP since they would be able to write files to these locations and then execute them. An administrator should create a blacklist of SRP path rules with the Disallowed security level to prevent executables from being run from these locations. Table 2 lists some path rules that should be applied to restrict common user writable locations within the SYSTEMROOT folder and prevent easy bypass of the intended application whitelisting policy.

| Default Rules |
|---|
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\Debug |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\PCHEALTH\ERRORREP |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\Registration |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\catroot2 |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\com\dmp |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\FxsTmp |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\spool\drivers\color |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\spool\PRINTERS |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\spool\SERVERS |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\System32\Tasks |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\SysWOW64\com\dmp |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\SysWOW64\FxsTmp |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\SysWOW64\Tasks |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\Tasks |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\Temp |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%\tracing |
| runas.exe |

**Table 2: Common Blacklist Rules for Built-in Default SRP Rules**

Do not create a blacklist rule for the %SYSTEMROOT%\System32\spool folder because users need to load libraries from its drivers subfolder in order to print. Instead use the more specific rules listed in Table 2. Also note that the runas.exe rule is used to prevent a known SRP bypass that works on Windows XP, but not on later versions of Windows. If runas is a utility that is used regularly, then Administrators should consider renaming runas with a different name, so that exact copy-and-paste use of the known SRP bypass will not work.

In addition to the Windows folder, the administrator should examine the Program Files directory since some programs may create subfolders that grant write privileges to users. When new software is installed, the administrator should scan the Program Files folder again. **Scanning for user writable subfolders and creating associated blacklist rules should be done by administrators on a regular schedule**. Several tools, such as Microsoft Sysinternals AccessChk[3], can aid an administrator in searching for user writable subfolders. See the Monitor SRP section for more information.

## Create SRP Rules for Authorized Applications

Now that SRP is configured in whitelisting mode with the most secure settings, new rules can be added based on the results from the domain audit. The results from the domain audit will determine the additional paths that may be needed for applications installed in non-standard locations. SRP supports four rule types, but only path rules are used in this guide because they are the easiest to administer and have the least impact on system performance. When more software publishers consistently digitally sign all their application files and computers are faster so the performance impact will be less noticeable, then SRP rules should be transitioned to use the more secure digital signature rules.

Path rules use local or universal naming convention (UNC) paths (e.g., \\server\share) of a file or folder. They support the wildcard characters of ***** (match many characters) and **?** (match one character). Path rules also support registry paths. Registry path rules are identified by percent signs that surround the entire path of the

[3] Windows Sysinternals. AccessChk v5.0. Published April 28, 2010. http://technet.microsoft.com/en-us/sysinternals/bb664922.aspx

registry entry. When the rule is evaluated, the value of the specified registry entry is used. Do not confuse registry path rules with regular path rules that use environment variables. Environment variables are also surrounded by percent signs but only for the variable rather than the entire path. Some recommended path rules that administrators should consider allowing are listed in Table 3.

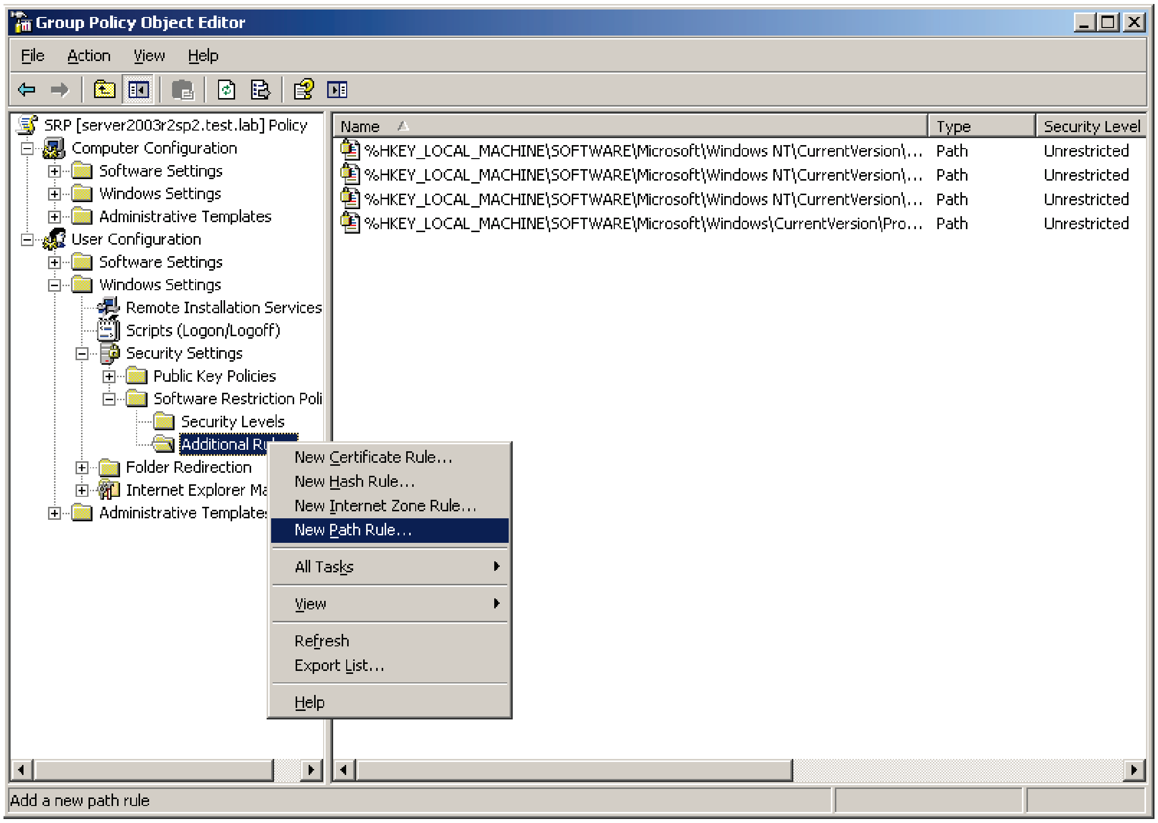
| Description | Type | Rule |
|---|---|---|
| The path of the Program Files (x86) folder on 64-bit computers. | Registry Path | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\ProgramFilesDir (x86)% |
| Domain login scripts. | Path | \\%USERDNSDOMAIN%\Sysvol\ |
| Shortcuts such as those used on the Start menu. | Path | *.lnk |

Table 3: Common Paths to Consider for New Path Rules

Since SRP also restricts the libraries loaded by programs, adding a path rule for the program alone may not be sufficient to allow the program to execute. Rules may need to be added for any libraries that are not loaded from an allowed path. **Before adding a new path rule, make sure the path is not writable by regular users**. If regular users can write to the location specified in the new path rule, then they can easily bypass the intended SRP policy and run any program they want.

To add a new path rule:

1. Right-click on the **Additional Rules** folder as shown in Figure 6.
2. Select **New Path Rule** from the right-click menu.



| Name | Type | Security Level |
|---|---|---|
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\... | Path | Unrestricted |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\... | Path | Unrestricted |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\... | Path | Unrestricted |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Pro... | Path | Unrestricted |

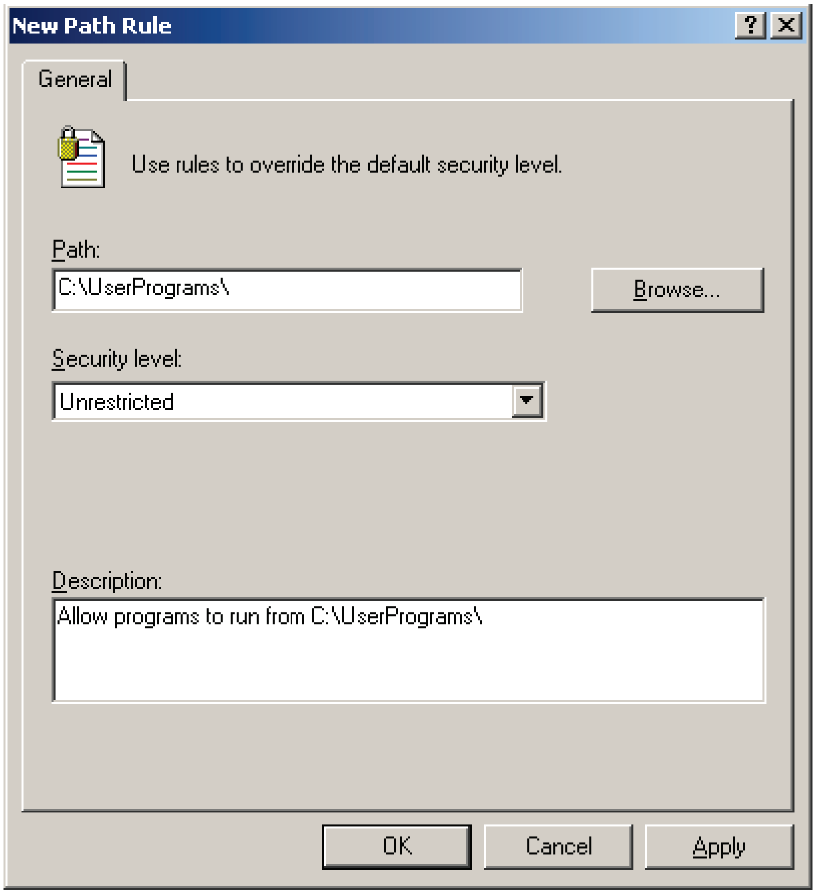
Add a new path rule

Figure 6: Select New Path Rule from the Additional Rules right-click menu.

3. In the **New Path Rule** dialog, enter a path in the **Path** textbox or click the **Browse** button to select a path as shown in Figure 7.
4. Make sure the **Security level** dropdown list has the **Unrestricted** option selected.

5. Enter a description in the **Description** textbox if desired.
6. Click the **OK** button to close the **New Path Rule** dialog.

Figure 7 is an example of adding a new path rule for the path of C:\UserPrograms\. By adding this rule, all programs in that path will be allowed to run as long as all the libraries loaded by the programs are loaded from this path or other allowed paths.



New Path Rule

Use rules to override the default security level.

Path: C:\UserPrograms\

Security level: Unrestricted

Description: Allow programs to run from C:\UserPrograms\

Figure 7: Add an additional path rule using the New Path Rule dialog.

In addition to adding rules for allowed applications that are not in the default allowed paths, administrators should also examine the results of the domain audit to see if there are any unauthorized applications that have been installed within allowed paths and their subfolders. If an unauthorized application is discovered, then uninstall the application. If the application cannot be uninstalled, then create a blacklist SRP path rule specifically for the path of the unauthorized application’s subfolder. This blacklist rule can be created by selecting the **Disallowed** option from the **Security level** dropdown list when creating a rule as discussed in step 4 above. However, adding blacklist rules to block unauthorized applications makes the effective SRP policy more complex.

## Test SRP

After applying SRP, it is important to test that the settings are being applied as intended. First ensure that any test computers with operating systems that are affected by the SRP implementation bug discussed in Appendix A have installed the appropriate patch or workaround. Since group policy settings are only refreshed approximately every 90 minutes, test computers will not enforce new settings immediately. Before testing SRP, apply the policy in a GPO and then run the gpupdate command and log off the test computer so the new policy will be applied without waiting for the next group policy refresh. If the Fast Logon group policy setting is enabled, then the client computer may need to be rebooted for the new policy to be completely applied. If this behavior is not desired, then the Fast Logon option can be turned off by going to **Computer Configuration** → **Administrative Templates** → **System** → **Logon** and setting **Always wait for the network at computer startup and logon** to **Enabled**.

### Test Allowed and Disallowed Paths

Now that the settings are applied as intended, try executing programs from paths that should and should not be allowed by the SRP rule set. When running a program from a disallowed path, a message box should be displayed with the following message on Windows XP and Windows Server 2003: *Windows cannot open this program because it has been prevented by a software restriction policy. For more information open Event Viewer or contact your system administrator*.

When running a program from a disallowed path at the command prompt, the following message should appear on Windows XP and Windows Server 2003: *The system cannot execute the specified program*.

Windows Vista, Windows Server 2008, and later operating systems use the same error message regardless of how the program is executed: *This program is blocked by group policy. For more information, contact your system administrator*.

If one of the above messages appears for a program that is running from an allowed path, then the program may be loading libraries from a disallowed path. Additional rules may need to be added for the libraries to allow the program to execute. The results from the domain audit will have the paths of the libraries loaded by a program.

### Troubleshoot Rules

SRP has some logging abilities that can help when testing or troubleshooting SRP rules. When SRP blocks a program from executing, a Windows Event Log entry should appear in the Application log. Table 4 shows the different Windows Event Log entries related to SRP and their meanings.

| Event | Message | Meaning |
|---|---|---|
| 865 | Access to %program% has been restricted by your Administrator by the default software restriction policy level. | A program was prevented from executing due to the default rule which automatically blocks all programs unless they are specifically allowed. |
| 866 | Access to %program% has been restricted by your Administrator by location with policy rule %guid% placed on path %path%. | A program was prevented from executing due to a configured path rule. |
| 867 | Access to %program% has been restricted by your Administrator by software publisher policy. | A program was prevented from executing due to a SRP certificate rule from a software publisher's certificate. |
| 868 | Access to %program% has been restricted by your Administrator by policy rule %guid%. | A program was prevented from executing due to a SRP hash or zone rule. |
| 882 | Access to %program% has been restricted by your Administrator by policy rule %guid%. | A program was prevented from executing by SRP but the SRP notification dialog was blocked from showing. |

**Table 4: Windows Event Log Entries for SRP**

When implementing this guidance, event ID 865 in the Application log will be the most common event. The event's description will list the path of the program that was prevented from running. If the path of the program appears to be an allowed path, then it may have been prevented from running due to loading a library from a path that does not have an allow rule associated with it. Consult the domain audit to see what libraries are loaded by the program in question and create new path rules as needed.

The Microsoft Sysinternals Process Monitor[4] utility can also help discover which libraries may have prevented the program from running. To troubleshoot with Process Monitor, run it on a computer that has SRP disabled and then follow these steps:

1. Start Process Monitor and create a new filter.
2. In the **Process Monitor Filter** dialog, configure the drop down menu options so it reads as **Process Name is *program.exe* then Include**.
3. Click the **Add** button and then click the **OK** button.
4. Once the capture has started, ensure that only the **Show File System Activity** option is enabled.
5. Run the program.
6. Look under the **Path** column in the output for any DLLs that may not be located in an allowed path.
7. Create new path rules as needed.

If the program is still being prevented from running, then check the Application log for event ID 866. This event will be logged due to a specific disallowed path rule that is used to blacklist specific locations where programs are not allowed to execute. Check the current SRP policy for any specific disallowed path rules that may apply to the path listed in the event's description.

In addition to Windows Event Log entries, a computer specific SRP log file can be created. The SRP log file records the specific rule used by SRP when it examined a program or library to determine if the program should be allowed to run. Create a new registry string value (REG_SZ) named **LogFileName** under the registry key of **HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\**. The registry value's data can be set to a file path such as C:\srp.log. New log entries are appended to the end of the file. An example entry from the log file looks like:

*cmd.exe (PID = 1022) identified C:\Windows\system32\cscript.exe as Unrestricted using path rule, Guid = {191cd7fa-f240-4a17-8986-94d480a6c8ca}*

The Globally Unique Identifier (GUID) from the log file maps to rules stored in the registry. Whitelist rules correspond to the Unrestricted security level and are located under **HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\**. Blacklist rules correspond to the Disallowed security level and are located under **HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\0\**. Since SRP is configured as a user policy in this guide rather than a computer policy, the rules are stored under the HKCU hive instead of the HKLM hive.

For the specific example log file entry show above, the rule is located under the registry key **HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers\262144\Paths\**. Below the **Paths** registry key, there is a registry key named **{191cd7fa-f240-4a17-8986-94d480a6c8ca}** that matches the GUID from the log file entry. Under that GUID registry key there will be a registry value named **ItemData** with its data set to a folder path or registry path. In the case for the above GUID, the data is a registry path rule for the registry value that the SYSTEMROOT environment variable gets its value from. This is one of the built-in default SRP rules listed in Table 5 below.

[4] Windows Sysinternals. Process Monitor v2.92. Published August 30, 2010. http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

| GUID | Rule | OS |
|---|---|---|
| 191cd7fa-f240-4a17-8986-94d480a6c8ca | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot% | Server 2003, Server 2008 |
| d2c34ab2-529a-46b2-b293-fc853fce72ea | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ProgramFilesDir% | Server 2003, Server 2008 |
| 7272edfb-af9f-4ddf-b65b-e4282f2deefc | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot%*.exe | Server 2003 |
| 8868b733-4b3a-48f8-9136-aa6d05d4fc83 | %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ SystemRoot%System32*.exe | Server 2003 |

**Table 5: Common GUIDs for Built-in Default SRP Rules**

In addition to checking the Windows Event Log and using SRP log files, see Appendix A for other issues that may occur and some workarounds.

## Deploy SRP throughout the Organizational Unit Hierarchy

Since SRP rules are being configured as user policy within GPOs, unique sets of rules can be attached to individual OUs within the Active Directory organizational structure. The SRP rules should be tailored to each group of users within the OU, tested, and then applied to the OU. This technique should be applied successively up the OU hierarchy until SRP has been deployed successfully to the entire domain.

As SRP is rolled out, ensure that all computers with operating systems that are affected by the SRP implementation bug discussed in Appendix A: Known Issues and Workarounds have installed the appropriate patch or workaround prior to enforcement of SRP on that computer.

## Monitor SRP

Once SRP has been customized, tested, and applied to the domain, monitor that SRP is working as desired. To accomplish this task, use the domain auditing tool that was used at the beginning of this process. Check the results for any executables that may be running from paths that are not allowed. Examine the Windows Event Log to see which applications are being blocked. If they should be allowed to run, then add new SRP path rules for them. Otherwise, educate users about the official policies regarding use of authorized and unauthorized applications. Enabling SRP logging as described in the previous section may also help monitor SRP operation.

When users require additional programs, they should request that their administrators install the new programs to paths allowed by SRP. If the program needs to be installed into an alternate location, the administrator should ensure that the path is not writable by regular users and then add a new path rule to the SRP policy.

Another important monitoring task is scanning for additional paths that may need to be blacklisted. As new software and operating system components are installed, new blacklist rules may need to be created. An administrator can use AccessChk to scan their computers for user writable paths. An example command is:

**accesschk -w -s -q -u *<group> <path>***

Administrators should scan paths mentioned in Table 1 and Table 3 plus any other paths defined in their SRP policy. Administrators should scan every path with each group below plus any site specific groups for their domain:

- Users
- Everyone
- Authenticated Users
- Interactive

## Conclusion

Configuring SRP as described in this guide can significantly increase the security posture of a domain while still allowing users to run the applications they need to perform their duties. For an effective SRP implementation an administrator must:

- Set the default SRP security level to **Disallowed** and select the **All software files** and **All users** options.
- Create whitelist rules for authorized applications as needed.
- Create blacklist rules for subfolders that users can write to within a whitelisted path.
- Ensure that users cannot both write to and execute from any location.

# Appendix A: Known Issues and Workarounds

There are some known usability issues and one bug that may affect organizations that adopt this guidance. Most of the usability issues will only be encountered on systems affected by the bug prior to patch installation.

## Bug

There is a known bug in specific operating system versions when using SRP with the **All software files** option enabled as recommended in this guide. Microsoft has documented this issue in knowledge base article 959074[5]. The operating system internally generates invalid paths when SRP tries to resolve the locations of libraries loaded by an executable. This problem occurs when running a program from drives other than the system drive (usually C:\). These drives include other physical hard drives, additional drive partitions, removable drives, and mapped network drives. The problem occurs when a user is logged into the following operating systems:

- Windows Server 2003, including R2, with SP1 or SP2
- Windows Vista, Windows Vista SP1, Windows Vista SP2
- Windows Server 2008 (Windows Server 2008 is based off Vista SP1), Windows Server 2008 SP2

Windows XP, Windows 7, and Windows Server 2008 R2 are not affected by this bug. Microsoft Help and Support has a patch available at http://support.microsoft.com/kb/969972 for Windows Vista SP1, Windows Vista SP2, Windows Server 2008, and Windows Server 2008 SP2. Installing the patch fixes this bug. The patch will be included in Windows Vista SP3 and Windows Server 2008 SP3 as well. Unfortunately patches are not available for:

- Windows Server 2003, including R2, with SP1 or SP2
- Windows Vista with no service pack installed

For the above operating system versions, the knowledge base article recommends disabling the **All software files** option. This recommendation *drastically* lowers the protection offered by Software Restriction Policies and is not recommended. For Windows Vista with no service pack, upgrade to the latest service pack and install the patch if SP3 is not available.

This leaves Windows Server 2003, including R2, with SP1 or SP2 as the only versions that require a workaround. Since users do not log into servers, the problem will only affect administrators while logged into Windows Server 2003 computers, so the impact will be limited.  Here are two workarounds that may help:

1. Create a junction to the Program Files and Windows folders on the other drives.
2. Create empty Program Files and Windows folders on the other drives.

Both methods require that SRP path rules be created for the new paths.

## Issues

The following items are some examples of possible issues, and some workarounds, that users may experience once SRP rules are applied. The majority of these issues will no longer happen once the previously mentioned SRP patch is installed.

[5] Software Restriction Policy Enforcement set to “All Software Files” causes checks against paths/files that are invalid. http://support.microsoft.com/kb/959074

### Using the Open With Submenu from a Disallowed Path

Using the **Open With** submenu from a disallowed path may not work and an error message may not be displayed. A workaround is to move the target of the item in the **Open With** submenu to an allowed path.

### Using the Send To Submenu from a Disallowed Path

Using the **Send To** submenu from a disallowed path may not work and an error message may not be displayed. A workaround is to move the target of the item in the **Send To** submenu to an allowed path. Also check if the specific **Send To** submenu item uses a shortcut. If it uses a shortcut then the same workarounds for shortcuts discussed below may apply.

### Opening Documents from a Disallowed Path

Double-clicking certain types of documents from a disallowed path may not work and a misleading error message may be displayed or an error message may not be displayed at all. Microsoft Office and Adobe PDF files are example documents that may have this behavior. A workaround is to open the program first and then open the document by using the program's **File** menu. Sometimes opening the document using the right-click **Open With** submenu can fix this problem too.

### Using Shortcuts that Contain Disallowed Paths

Shortcuts that have the **Start in** property set to a disallowed path may not work and an error message may not be displayed. A workaround is to modify the shortcut's **Start in** property so it is empty or set to an allowed path. Also check that the shortcut's **Target** property is in an allowed path.

### Running Programs that use Disallowed Paths as a Working Directory

A program that uses a working directory that is a disallowed path may not run correctly. A workaround is to start the program using a shortcut with the shortcut's **Start In** property set to an allowed path.

### Running Software Update Mechanisms that use Disallowed Paths

A user attempts to install an update and it appears to install correctly without any visible error messages, but depending on how the update mechanism works, it may not have actually installed the update.

Some software update mechanisms write files to the Temp folder and then attempt to execute them. Since the Temp folder is writable by users, files should **not** be allowed to execute from there, and will be blocked by SRP. If the software update is not allowed to execute, there will be an event in the Windows Event Log under the Application log with an event ID of 865. One example of this scenario is the Adobe Flash updater. Application updates should be automatically distributed without relying on user intervention. Use a patch management solution or develop a script to install these updates without requiring user intervention.