Product Guide

McAfee Application Control 6.1.0

Contents

Preface

Contents

About this guide

This information describes the guide's target audience, the typographical conventions and icons used in this guide, and how the guide is organized.

Audience

McAfee documentation is carefully researched and written for the target audience.

The information in this guide is intended primarily for:

- **Administrators** — People who implement and enforce the company's security program.
- **Users** — People who use the computer where the software is running and can access some or all of its features.

Conventions

This guide uses these typographical conventions and icons.

Book title, *term*, *emphasis*	Title of a book, chapter, or topic; a new term; emphasis.
Bold	Text that is strongly emphasized.
`User input, code, message`	Commands and other text that the user types; a code sample; a displayed message.
Interface text	Words from the product interface like options, menus, buttons, and dialog boxes.
Hypertext blue	A link to a topic or to an external website.
	**Note:** Additional information, like an alternate method of accessing an option.
	**Tip:** Suggestions and recommendations.
	**Important/Caution:** Valuable advice to protect your computer system, software installation, network, business, or data.
	**Warning:** Critical advice to prevent bodily harm when using a hardware product.

Find product documentation

McAfee provides the information you need during each phase of product implementation, from installation to daily use and troubleshooting. After a product is released, information about the product is entered into the McAfee online KnowledgeBase.

Task

1 Go to the McAfee Technical Support ServicePortal at http://mysupport.mcafee.com.

2 Under **Self Service**, access the type of information you need:

To access...	Do this...
User documentation	1 Click **Product Documentation**. 2 Select a product, then select a version. 3 Select a product document.
KnowledgeBase	• Click **Search the KnowledgeBase** for answers to your product questions. • Click **Browse the KnowledgeBase** for articles listed by product and version.

1 Introduction

McAfee® Application Control software offers an effective way to block unauthorized applications from running on your systems. Unlike simple whitelisting, it uses a dynamic trust model to avoid labor-intensive lists.

Today's IT departments face tremendous pressure to ensure that systems and servers comply with security policies, operating procedures, and regulations. Users can unintentionally introduce software that poses a risk to the business, installs malware, creates support issues, and violates software licenses, compromising systems and your business. Businesses of all sizes need an efficient way to standardize systems and servers to make sure that they are running only approved software, without impacting productivity.

As enterprises face unknown software from the Internet, Application Control adds timely control to your system security strategy, and is attuned to the operational needs of enterprises.

Contents

Application Control overview

Application Control is an efficient software to protect your systems from all types of threats.

Increase control over fixed-function systems

In regulated industries like banking, retail, and manufacturing, devices such as point-of-sale (POS) terminals or customer service terminals perform critical functions and often store sensitive data. Application Control extends a layer of protection to fixed function systems. Its low overhead footprint does not impact system performance, requires low initial and ongoing operational overhead, and works effectively in standalone mode. The product is designed to operate in network and firewall configurations. It can even operate on systems that are not connected to a network.

Business efficiency in a controlled environment

Malicious code takes advantage of the flexible software and modular code used in business environments. Application Control extends coverage to Java, ActiveX controls, scripts, batch files, and codes. This give you greater control over application components, and blocks advanced threats without requiring signature updates.

Easy solution

Application Control is an easy solution.

- Easy setup and low initial and ongoing operational overhead.
- Minimal impact on CPU cycles and uses less than 10 MB of RAM.

- No file system scanning that could impact system performance.
- Requires no signature updates.

Dynamic whitelisting using a trust model

Application Control provides flexible, affordable, and secure dynamic management of a whitelist. This dynamic management allows Application Control makes it easy to support multiple configurations for different business needs, such as POS terminals, backoffice servers, and multiple desktop images for different user profiles.

Leveraging a trusted source model, Application Control eliminates the need for IT administrators to manually maintain lists of approved applications. On a protected system, only authorized software is allowed to run and it cannot be changed. Application Control prevents attempts to tamper with protected files, creates an event for each attempt, and writes event entries in a log file.

Application Control requires very initial and ongoing operational costs and provides complete protection from unwanted applications and code. This protection helps system administrators to maintain control on the system status.

Key advantages and uses

- Protection against zero-day threats without requiring signature updates.
- Lower cost of ownership because dynamic whitelisting eliminates manual effort.
- Protection against malware for these fixed function systems.
  - POS terminals (in retail environments)
  - Kiosk devices
  - Customer service terminals
  - Automated teller machines (ATMs) in banking
  - Servers and corporate desktops

Product features

Application Control protects your system from any unauthorized attempt.

Application Control provides these key features.

- **Malware protection** — Protects systems from malware attacks before they occur, by proactively controlling the application execution on the system.
- **Secured-lock system** — Enables the secured - lock system against threats and unwanted changes.
- **Execution protection** — Prevents execution of unauthorized updates that might change the existing applications running on the system.
- **Dynamic whitelisting** — Eliminates the manual maintenance effort that other whitelisting technologies require.
- **Trusted applications** — Enables administrators to adopt a flexible approach with a centralized repository of trusted applications to run on the system.
- **Memory protection** — Prevents execution of all unauthorized software, scripts, and dynamic-link libraries (DLLs), and further defends against memory exploits.
- **Automatically whitelists** — Whitelists new software added through an authorized process.

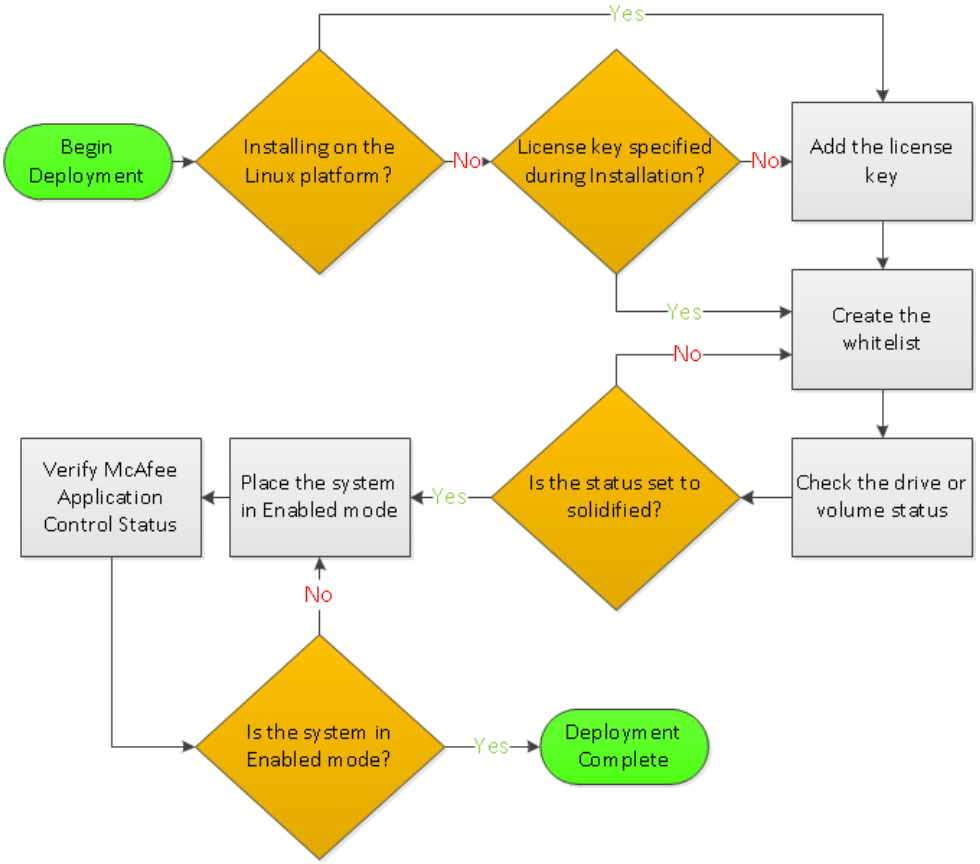
2 Getting started

Application Control workflow details important concepts and includes instructions to help you deploy Application Control on a system.

Contents

Application Control workflow

This diagram provides an overview of the Application Control deployment workflow.



Understanding Application Control modes

Application Control operates in different modes depending on your requirements.

Disabled Application Control is not running on the system. Although Application Control is installed, its features are disabled.

Disabled mode is supported on Windows and Linux platforms. From the Disabled mode, you can switch to the Enabled or Update mode. See *Disable the product*.

Enabled Only whitelisted applications and files are allowed to run. Execution of unauthorized software, such as a virus or spyware, is prevented. In Enabled mode, Application Control protects all files in the whitelist from unauthorized modification and deletion attempts. After the initial whitelist is created for your system, switch Application Control to Enabled mode. This makes sure that no unauthorized changes are allowed.

Enabled mode is supported on Windows and Linux platforms. You can switch Application Control from Enabled mode to Disabled or Update mode. See *Place Application Control in Enabled mode*.

Update Perform authorized software updates on a protected system. This mode groups all required update actions, such as addition, modification, or removal of software, then executes the actions. When you perform software updates in the Update mode, Application Control tracks and records each change. Also, it dynamically updates the whitelist to make sure that the modified or added binaries and files are authorized to execute when the system returns to Enabled mode. If you delete any software and program files from the system, the respective files are removed from the whitelist.

Update mode is supported on Windows and Linux platforms. You can switch from Update mode to the Enabled mode only. See *Perform emergency changes* .

Observe Unavailable in the standalone configuration; available only when the system is managed by McAfee® ePolicy Orchestrator®.

In Observe mode, the application is in effect but does not prevent any changes made to the endpoints. Observe mode is supported only on Windows platform.

How the whitelist works

When you deploy Application Control to protect a system, it scans the system and creates a whitelist of all executable binaries and scripts present on the system.

The whitelist lists all authorized files and determines trusted or known files. In Enabled mode, only files that are present in the whitelist are allowed to execute. All files in the whitelist are protected and cannot be modified or deleted. An executable binary or script that is not in the whitelist is said to be *unauthorized* and is prevented from running.

Application Control stores the whitelist for each drive or volume at the following location:

- **Windows**: `<drive>\Solidcore\scinv`
- **Linux**: `<volume>/.solidcore/scinv`

Here is a list of the type of files included in the whitelist.

- Application program code (for Windows and Linux)
- Binary executables (.exe, .sys, and .dll files for Windows and ELF file format for Linux)
- Script files (.bat, .cmd, and .vbs files for Windows and files containing #! for Linux)

When the whitelist is created for Windows, Application Control does not include system-specific files that are protected by the operating system. For example, `pagefile.sys` and `hiberfil.sys`.

When you execute a file on a whitelisted system, Application Control compares the checksum and path of the binary with the checksum and path stored in the whitelist and allows the execution only if the checksum value and path matches.

Using the command line interpreter

The command line interpreter (sadmin) allows you to manage the Application Control configuration and features.

The method you use to open the command line interpreter depends on your operating system.

Operating system	Steps
Windows	• On the Windows Vista, Windows 2008, Windows 2008 R2, or Windows 7 (with UAC enabled) platforms, right-click on the **McAfee Solidifier Command-line** icon on the desktop and select **Run as Administrator**. • On other Windows platforms, double-click the **McAfee Solidifier Command-line** icon on the desktop. • Click **Start** | **Programs** | **McAfee** | **Solidifier** | **McAfee Solidifier Command Line** menu option. By default, `sadmin` is added to the PATH environment variable and allows the `sadmin` command to work by opening the CLI from any location.
Linux	1 Open a Linux terminal. 2 Access the command-line interpreter from `<install directory>/mcafee/solidcore/bin/sadmin`.

Use these help commands to get help information.

Syntax	Description
`sadmin help`	Lists basic help information.
`sadmin help <command>`	Provides basic help for the specified command.
`sadmin help-advanced <command>`	Provides advanced help for the specified command.

This document describes using Application Control in the standalone configuration only.

Deploy Application Control

Complete the tasks to deploy Application Control on a system.

Before you begin

Review the deployment workflow.

Tasks

- *Add the license* on page 14
  The license determines if the product features are available to you.
- *Create the whitelist* on page 14
  The whitelist controls the applications and files that can run on a protected system. Create a whitelist of all executable binaries and scripts present on the system.
- *Place Application Control in Enabled mode* on page 15
  Place Application Control in Enabled mode to allow only the whitelisted applications to run on the system.

Add the license

The license determines if the product features are available to you.

- **Windows** – You can specify the license during or after installation. If you don't specify a license during installation, you must when you run Application Control on the system.
- **Linux** – You must specify a valid license after installation when you run Application Control on the system.

Task

1 Verify if a license is already added (provided during installation) by entering the following command and pressing **Enter**.

```
sadmin license list
```

All licenses already installed on the system are listed.

2 If no license is listed, add a license now.

a Run this command at the command prompt.

```
sadmin license add <license_key>
```

b Restart the Application Control service.

Windows	`net stop scsrvc` `net start scsrvc`
Linux	`service scsrvc restart`

Create the whitelist

The whitelist controls the applications and files that can run on a protected system. Create a whitelist of all executable binaries and scripts present on the system.

Before you begin

- Read how Application Control uses the whitelist. For detailed information, see the *How the whitelist works* section.
- Optionally, set the whitelist thread priority before creating the whitelist. For detailed information, see the *Configure the whitelist thread priority* section.

Task

1 Run this command at the command prompt.

```
sadmin solidify
```

The time the system takes to create the whitelist varies from a few minutes to an hour, depending on your system configuration, including CPU speed, RAM, and applications installed on the system. After the whitelist is created, a message similar to this message appears.

```
Solidifying volume C:\
00:04:11: Total files scanned 12265, solidified 6342
```

2 Verify that drive or volume is whitelisted.

a Run this command at the command prompt.

```
sadmin status
```

The status of Application Control is displayed. You can view the operational mode, operational mode on system restart, connectivity with McAfee ePO, CLI access status, and whitelist status of the drives or volumes. However, in the standalone configuration of the product, connectivity with McAfee ePO is not applicable.

b Review the whitelist status of the drives or volumes, and make sure that the status is Solidified.

Place Application Control in Enabled mode

Place Application Control in Enabled mode to allow only the whitelisted applications to run on the system.

Task

1 Run this command at the command prompt.

```
sadmin enable
```

2 Place Application Control in Enabled mode.

Operating system	Action
Windows	Perform one of these steps: • Restart the system to enable Application Control and the memory protection feature. • Restart the Application Control service (using `net stop scsrvc` followed by `net start scsrvc` command), to enable Application Control without the memory protection feature.
Linux	Restart the Application Control service (using `service scsrvc restart` command), to enable Application Control.

3 Verify that Application Control is in Enabled mode with this command:

```
sadmin status
```

Application Control status is displayed. You can view the operational mode, operational mode on system restart, connectivity with McAfee ePO, CLI access status, and whitelist status of all drives. However, in the standalone configuration of the product, connectivity with McAfee ePO is not applicable.

a Review the operational mode.

b Ensure that the current operational mode is Enabled.

3 Protecting file system components

When Application Control is running in Enabled mode, you can choose a set of files, directories, drives (Windows), volumes (on Linux), and registry keys to protect from unauthorized changes.

Contents

How protection works

Application Control prevents unauthorized changes to your system components by write-protecting them.

Application Control can write-protect or read-protect these components.

Feature	Component	Prevented actions
Write-protection	File	• Creating • Modifying • Renaming • Deleting • Creating hard links • Creating Alternate Data Stream (ADS) for Windows
	Directory	• Modifying • Deleting • Renaming
	Drive/Volume	
	Registry Key (Windows)	
Read-protection	File	Reading data
	Directory (Applicable only to the files inside the read-protected directory)	
	Drive/Volume (Applicable only to the files inside the read-protected drive/volume)	

If you specify a component (file, directory, volume) to be write-protected before creating it, you cannot create a component with that name.

If a file is write-protected, you cannot modify its content or attributes. However, on the Windows platform, certain attributes can be modified.

Attribute	Attribute modification allowed
Encryption	No
Compression	No
Hidden	Yes
Read-only	Yes

What is write protection

Write protection is a feature that protects the files, directories, and drives (Windows) or volumes (Linux) from being modified or deleted. Write-protection feature is identified as deny-write in the features list. By default, this feature is enabled.

If you write-protect a directory, drive, or volume, write protection is applied to all files and subdirectories in that directory, drive, or volume. If any file residing in a directory or subdirectory is write-protected, you are not allowed to rename, move, or delete its parent directory. Creation of new files in a write-protected directory, drive, or volume is also not allowed.

Write-protect only files that are not routinely updated by programs. For example, `C:\WINDOWS\system32\drivers\etc\hosts`.

This feature is in effect only when Application Control is operating in Enabled mode.

Any unauthorized attempt to modify the contents of a write-protected component is prevented and an event is generated.

What is read protection

Read protection is a feature that protects the files, directories, drives (Windows), and volumes (Linux) by preventing the data in the files from being read. Read-protection feature is identified as deny-read in the features list.

This feature is disabled by default and can be enabled by using `sadmin features enable deny-read` command. No restart is required for enabling or disabling this feature. Read protection works only when Application Control is running in Enabled mode.

When a directory, drive, or volume is read-protected, read protection is applied only to the files in that directory, drive, or volume. As a result, the files in the subdirectories are also read-protected. If a read-protected file or directory is moved to a different path, it is no longer read-protected.

Be careful when you read-protect directories, drives, or volumes to allow Application Control to operate on a system. For example, if you read-protect a directory, drive, or volume, the whitelisted files in that directory, drive, or volume cannot execute. Also, if you create a file in a read-protected directory, drive, or volume, the file cannot be added to the whitelist.

Make sure that the read-protected files are also write-protected using the deny-write feature to provide extra protection to the read-protected files. This ensures that the contents of the files cannot be read by renaming or moving the files. A read-protected file (that is not write-protected) becomes readable if it is renamed or moved to another location.

Any unauthorized attempt to read data from a read-protected file is prevented and an event is generated.

You cannot read-protect registry keys.

Write-protection feature

You can write-protect specific files, directories, drives (Windows), and volumes (Linux) to prevent unauthorized programs or users from modifying them. Write-protecting these components makes them read-only and prevents unauthorized changes. These components cannot be compressed or encrypted.

Tasks

- *Apply write protection* on page 19
  Write-protection feature makes the components read-only and protects the components from unauthorized changes.
- *Exclude components from write-protection* on page 20
  Exclude specific components from a write-protected directory, drive (Windows), or volume (Linux).
- *List write-protected components* on page 21
  View the complete list of write-protected files, directories, and drives (Windows) or volumes (Linux).
- *Remove write protection* on page 21
  When you remove write-protection, components are no longer protected from unauthorized changes.

Apply write protection

Write-protection feature makes the components read-only and protects the components from unauthorized changes.

Task

1 Write-protect files, directories, drives (Windows), or volumes (Linux).

```
sadmin write-protect [ -i ] pathname1 ... pathnameN
```

Specify the complete paths to the components to be write-protected. Paths can include wildcard character (*). However, it can only represent one complete path component.

- On Windows, using `\abc\*\def` is allowed while `\abc\*.doc`, `\abc\*.*`, or `\abc\doc.*` is not allowed.
- On Linux, using `/abc/*/def` is allowed while `/abc/*.sh`, `/abc/*.*`, or `/abc/doc.*` is not allowed.

For example:

- `sadmin write-protect -i Listerner.ora` (Windows)
- `# sadmin write-protect –i /etc/security/limits.conf` (Linux)

Write-protect the network file system components by specifying the network path with the `sadmin write-protect` command in any of the ways to prevent any modifications to the network share from the client system.

Hard link to a write-protected file should also be write-protected to prevent any modification to the hard link.

This table describes how you can specify the network path with the command.

Syntax	Example
`sadmin write-protect –i \ \server-name\share-name`	Specify the server name that has a network share. Also, specify the name of the network share. For example: `sadmin write-protect –i \\ftpserver\documents`
`sadmin write-protect –i \ \server-ip\share-name`	Specify the IP address of a server and name of the network share. For example: `sadmin write-protect –i \\192.168.0.1\documents`
`sadmin write-protect –i mapped-drive-letter:\`	Specify the drive letter, which is mapped to the server on the client system. For example: `sadmin write-protect –i W:\`
`sadmin write-protect -i / mount-point` (Linux)	Specify the mount point name on the Linux platform. For example: `sadmin write-protect –i /nfs`

2 Write-protect registry keys.

```
sadmin write-protect-reg [ -i ] registrykeyname1 ... registrykeynameN
```

Paths used in registry key-based rules can include the wildcard character (*). However, it can only represent one path component in the registry path. Do not use the character for the component at the end of the complete registry path (if used at the end the path filter will not be in effect). For example, registry path `HKEY_LOCAL_MACHINE\*\Microsoft` is allowed while `HKEY_LOCAL_MACHINE\*` or `HKEY_LOCAL_MACHINE\*\*` is not allowed.

Modifications to the write-protected registry keys are not allowed.

We recommend you write-protect only the `HKEY_LOCAL_MACHINE\SOFTWARE` registry key cluster to protect the Windows components. Other registry key clusters should not be write-protected.

Specify registry key names as parameters with the write-protect-reg (wpr) command to apply write protection to registry keys. For example:

```
sadmin write-protect-reg –i HKEY_LOCAL_MACHINE\SOFTWARE
```

Exclude components from write-protection

Exclude specific components from a write-protected directory, drive (Windows), or volume (Linux).

Task

1 Exclude specific components from a write-protected directory, drive, or volume.

```
sadmin write-protect -e pathname1 ... pathnameN
```

When you specify a file path to be excluded from a write-protected directory, drive, or volume, write-protection is removed from only that specific file.

Specify the complete path to the files, directories, drives, or volumes to be excluded from write protection. For example:

- `sadmin write-protect –e Listerner.ora` (Windows)
- `# sadmin write-protect -e /etc/security/limits.conf` (Linux)

2 Exclude registry keys from write protection.

```
sadmin write-protect-reg -e registrykeyname1 ... registrykeynameN
```

Specify the registry key names as parameters with this command and the exclude argument to exclude registry keys from being write-protected. For example:

```
sadmin write-protect-reg –e HKEY_LOCAL_MACHINE\SOFTWARE
```

List write-protected components

View the complete list of write-protected files, directories, and drives (Windows) or volumes (Linux).

```
sadmin write-protect –l
```

List all write protected registry keys.

```
sadmin write-protect-reg –l
```

Remove write protection

When you remove write-protection, components are no longer protected from unauthorized changes.

- You can remove write protection from specific files, directories, drives (Windows), or volumes (Linux). Also, you can remove write protection applied to specific registry keys.
- You can flush write protection to remove it from all files, directories, drives, and volumes, as well as from all registry keys.

Task

1 Remove write protection applied to specific components.

```
sadmin write-protect [ -r ] pathname1 ... pathnameN
```

When you specify the file path, write protection applied to all files in the specified path is removed.

Specify the complete path to the file, directory, or drive to be removed from write protection.

For example:

- `sadmin write-protect -r Listerner.ora` (Windows)
- `# sadmin write-protect –r /etc/security/limits.conf` (Linux)

2 Remove write protection from specific registry keys.

```
sadmin write-protect-reg [ -r ] registrykeyname1 ... registrykeynameN
```

For example:

```
sadmin write-protect–reg -r HKEY_LOCAL_MACHINE\SOFTWARE
```

3 Flush write protection from all files, directories, drives, or volumes.

```
sadmin write-protect –f
```

4 Flush write protection from all registry keys.

```
sadmin write-protect-reg –f
```

Read-protection feature

You can read-protect specific files, directories, drives (Windows), and volumes (Linux) to prevent unauthorized programs or users from reading the data. These components cannot be compressed or encrypted.

Tasks

- *Apply read protection* on page 22
  The read-protection feature prevents unauthorized programs or users from reading data from the components.
- *Exclude specific components from read protection* on page 22
  Exclude specific components from a read-protected directory, drive (Windows), or volume (Linux).
- *List read-protected components* on page 23
  View the complete list of components that are write-protected.
- *Remove read protection* on page 23
  Removing read-protection allows users or unauthorized programs to read data from the components, putting the critical data at risk.

Apply read protection

The read-protection feature prevents unauthorized programs or users from reading data from the components.
Read-protect files, directories, drives (Windows), or volumes (Linux).

```
sadmin read-protect [ -i ] pathname1 ... pathnameN
```

Specify the complete path to each component to be read protected. Paths can include wildcard character (*). However, it can only represent one complete path component.

- On the Windows platform, using `\abc\*\def` is allowed while`\abc\*.doc`, `\abc\*.*`, or `\abc\doc.*` is not allowed.
- On the Linux platform, using `/abc/*/def` is allowed while `/abc/*.sh`, `/abc/*.*`, or `/abc/doc.*` is not allowed.

For example:

- `sadmin read-protect –i password.docx` (Windows)
- `# sadmin read-protect –i /etc/password` (Linux)

You can apply read protection over mounted network file system components by specifying the network paths with the `sadmin read-protect` command.

Exclude specific components from read protection

Exclude specific components from a read-protected directory, drive (Windows), or volume (Linux).
Exclude specific components.

```
sadmin read-protect -e pathname1 ... pathnameN
```

Specify the complete path to the files, directories, drives, or volumes to be excluded from read-protection.

For example:

- `sadmin read-protect -e password.docx` (Windows)
- `# sadmin read-protect –e /etc/password` (Linux)

List read-protected components

View the complete list of components that are write-protected.

```
sadmin read-protect –l
```

Remove read protection

Removing read-protection allows users or unauthorized programs to read data from the components, putting the critical data at risk.

Remove read protection using one of these two methods.

- Remove read protection.

  Remove read protection from specific files, directories, drives (Windows), or volumes (Linux).

- Flush read protection.

  Flush read protection from all files, directories, drives (Windows), and volumes (Linux).

Task

1 Remove read protection applied to specific components.

```
sadmin read-protect [ -r ] pathname1 ... pathnameN
```

Specify the complete path to the files, directories, drives, or volumes to be removed from read-protection.

For example:

- `sadmin read-protect -r confidential.docx` (Windows)
- `# sadmin read-protect -r /etc/password` (Linux)

2 Flush read protection applied to all components.

```
sadmin read-protect –f
```

4 Overriding applied protection

Describes the techniques used to override the protection. On a protected system, overriding applied protection allows components to execute using checksum values, certificates, or from a trusted directory. If a component is configured as an updater, it can also update the software on a protected system.

Contents

How do I override protection

Authorize a program or file on a protected system to override protection.

You can authorize a program or file on a protected system by using one of these methods.

- Checksum
- Certificate
- Name
- Adding to the whitelist
- Trusted directories
- Trusted users

Whitelisting is the most-common method to determine the trusted or known files.

The order in which the methods are listed indicates the precedence, the software applies to the methods. For example, if you ban a program based on its checksum value and it is present in the whitelist (and hence is authorized), the program is banned. Similarly, if a program is allowed based on its checksum value and is banned by name, the program is allowed to execute and run.

When you authorize a program or file by name, all programs or files that have the same name and are present on the system or network directories are authorized on a protected system. We recommend that you use caution and judiciously authorize a program or file by name. When you authorize by name, you can specify the absolute path of the program or file to authorize only the required program or file by name.

Typically, most applications and executable files remain unchanged over prolonged periods of time. However, if needed, you can allow certain applications and executable files to create, modify, or delete files in the whitelist. To design a trust model and allow more users or programs to modify a protected system, you can use one the methods listed in this table.

Method	Supported Operating System	Description
Using updaters	Windows and Linux	Updaters are authorized components that are permitted to update the system. If a component is configured as an updater, it is allowed to install new software and update existing software components on a protected system. For more information, see *Using updaters*.
Using certificates	Windows	Application Control allows trusted certificates associated with software packages to run on a protected system. After you add a certificate, you can run all software signed by the certificate on a protected system. For more information, see *Using certificates*.
Using checksum values	Windows	Override protection applied to a system by authorizing certain binaries based on their checksum value. Authorizing binaries by their checksum (SHA1) value allows them to execute on the protected system. You can also provide updater permissions to an authorized binary. For more information, see *Using checksum values*.
Using trusted directories	Windows and Linux	On a protected system, you can add directories (local or network directories) as trusted directories to run any software present in these directories. Trusted directories are identified by their Universal Naming Convention (UNC) path. For more information, See *Using trusted directories*.
Using trusted users	Windows	Trusted user refers to an authorized Windows user with permissions to dynamically add to the whitelist. If you provide the updater permissions to a user, the user is defined as a trusted user. You can add a user as a trusted user to allow the user to install or update any software. While adding the user details, you must also provide the domain details. For more information, see *Using trusted users*.
Using Update mode	Windows and Linux	Update mode is one of the authorized modes to perform software updates on a protected system. When Application Control is in Update mode, all changes are allowed on a protected system. Place the system in Update mode to perform software updates. Use this method when none of the other methods, such as using trusted users, trusted directories, certificates, or checksum values meet your requirements and the software is not present in the updaters list. For example, you can use Update mode to complete maintenance tasks, such as install patches or upgrade software. For more information, see *Understanding Application Control modes*.

Using updaters

On a protected system, most software applications and executable files are not updated regularly. However, you can override the protection and tamper proofing that is in effect by specifying certain legitimate files or programs as updaters.

When there are certain components that are frequently required to install new software or update existing software components, specifying the components as updaters is a recommended way over using the Update mode. Adding such components as updaters is recommended because you can select the components and provide updater permissions to the intended components only whereas using the Update mode authorizes all components to perform update actions, such as addition, modification, or removal of software. However, in the Update mode, all read and write protection that is in effect is overridden.

What are updaters

Updaters are authorized components that are permitted to update the system.

By default, if you provide the updater permissions to a component, the child components automatically inherit the updater permissions. For example, if you specify Adobe 8.0 program as an updater, it can periodically patch all needed files.

Updaters work at a global-level and are not application- or license-specific. When a program is defined as an updater, it can modify any protected file.

To qualify as updaters, components must match one of these requirements:

Requirement	Description
Components must be present in the whitelist.	Only the supported type of components present in the whitelist can be added as updaters. For example, if you specify whitelisted `AcroRd32.exe` as an updater, it is allowed to automatically update the Adobe Reader software for new updates.
Components must be defined as authorized binaries.	Adding authorized binaries as updaters is a workaround and should be done only when it is necessary to update the software components using the allowed binaries. Be cautious and judiciously assign updater permissions to binary files. We suggest you to remove the updater permissions from the binary files soon after the update is done. If authorized binary files are specified as updaters, they can allow other associated binary files to make changes on the protected systems. For example, if you set `cmd.exe` as an updater and invoke any executable from it, the executable can perform any change on the protected systems. To avoid a security gap, it is not recommended to have a file configured as an allowed binary and updater concurrently. For more information on how to allow binary files, see the *Using checksum values* section.

Application Control also includes predefined default updater permissions for commonly used applications that might need to update the systems frequently. These applications are known as default updaters. For example, default updater permissions are defined for Yahoo, Oracle, and McAfee products.

When do I add updaters

There are certain programs, which frequently update software components on the system automatically. Add such programs as updaters to allow them to update the software components.

Add scripts, installers, binaries, and users as updaters when they are frequently required to make changes on the system.

What can I add as updaters

Add components such as installers, scripts, binaries, users, or certificates as updaters.

Components	Examples

Installers (for Windows)	Add installers as updaters to allow them to automatically update the software components on the protected systems.
	• Windows installer.
	For example, to add Windows installer (for a Hotfix, KB893803) as an updater and perform automatic updates on protected files or registry keys, specify this command:
	`sadmin updaters add WindowsInstaller-KB893803-v2-x86.exe`
	• Microsoft installer (MSI)-based installer
	For example, to add MSI-based installer `Ica32Pkg.msi` and perform automatic updates on protected files or registry keys, specify this command:
	`sadmin updaters add Ica32Pkg.msi`

Scripts	Add scripts as updaters to provide updater permissions to the scripts. Scripts with updater permissions are allowed to perform update operations on the protected systems.
	`sadmin updaters add <scriptname>`
	`sadmin updaters add myscript12.bat`
	Adding scripts as updaters functionality is available on all Windows platforms except Windows Server 2003 (IA64).

Binaries	Add binaries as updaters to provide updater permissions to the binaries. Binaries with updater permissions are allowed to update the protected binaries and software components. Binaries also include executable (`.exe`) files on Windows platform.
	`sadmin updaters add <binaryname>`
	`sadmin updaters add update.exe`

Components	Examples

Users (for Windows)	Add users as updaters to allow the users to perform update operations on the protected system.
	`sadmin updaters add –u <username>`
	`sadmin updaters add –u <username>`
	`sadmin updaters add –u john_smith`
	For domain users:
	`sadmin updaters add –u john_smith@mycompany.com`
	`sadmin updaters add –u mydomain\john_smith`

Certificates (for Windows)	Add selected certificates as updaters to assign updater permissions to all components signed by the selected certificates. All components signed by selected certificates are allowed to make changes to the binaries on the system and launch new applications. Be cautious and judiciously assign updater permissions to the certificates. See the *Using certificates* section.
	You can add certificates as updaters only on the Windows platform.
	For example, if you add the Microsoft certificate that is used to sign the Internet Explorer application as an updater, it allows the Internet Explorer to download and execute any application.
	Application Control supports only X.509 certificates (base 64 encoded).
	`sadmin cert add –u <certfilename>`
	`sadmin cert add –u firefox.cer`

Processes that are currently running can be added as updaters.

While creating the whitelist, all temporary folders are ignored and are not whitelisted. The exception is when a process with updater permissions creates binaries in the temp folder, the binaries are added to the whitelist. You can add, list, or remove the updaters using the `sadmin updaters` command with the required arguments.

Also, you can modify the default configuration of Application Control to allow more commonly used applications to execute and add them to default updaters. You can add these types of applications to default updaters:

- Software provisioning systems that download, install, and runs new code. For example, Microsoft software update and custom scripts.
- Self-updating applications. For example, anti-virus.

After creating the whitelist on a system, Application Control configures the default updaters. Application Control updates the default configuration to allow the default updaters to execute and update the Commercial-Off-The-Shelf (COTS) applications. You can use the Finetune utility to configure default updaters. For more information on Finetune utility, see the *Update the default updaters using Finetune* section.

Add updaters

You can add various components as updaters to allow them to update the software components.

Run this command at the command prompt.

```
sadmin updaters add <filename>
```

This table lists the supported arguments, descriptions, and examples.

Argument	Description

-d	To exclude the child processes of a binary file to be added as an updater from inheriting the updater permissions.
	`sadmin updaters add –d <filename>`
	`sadmin updaters add –d winlogon.exe`

-n	To disable event logging for a file to be added as an updater.
	`sadmin updaters add –n <filename>`
	`sadmin updaters add –n winlogon.exe`

-l	To add an execution file as an updater only when the specified library name is loaded for the execution file (for Windows).
	`sadmin updaters add -l <associated libraryname> <filename>`
	`sadmin updaters add –l system32\wuauserv.dll svchost.exe`

-t	To perform these operations:
	• Include the tags for a file to be added as an updater.
	`sadmin updaters add -t <associated tag> –l <associated libraryname> <filename>`
	`sadmin updaters add –t Win_up_schedule1 –l system32\wuauserv.dll svchost.exe`
	• To add a user with a tag name as an updater.
	`sadmin updaters add –t <tagname> –u <username>`
	`sadmin updaters add –t McAfee001 -u john_smith`

-p	To add a binary file as an updater only when its parent execution file (for Windows) or parent program (for Linux) is running.
	`sadmin updaters add -p <parentname> <filename>`
	`sadmin updaters add –p svchost.exe iexplore.exe`

-u	To add a user as an updater (for Windows). All update operations by the specified user name are allowed. When you specify the `–u` argument, other arguments, such as `-l`, `-p`, `-d`, and `–n` are not applicable. `sadmin updaters add –u <username>` The table lists the type of user names that can be added as updaters:

User name	Example
Simple name	john_smith
Domain name	username@domain name john_smith@mycompany.com
Hierarchical domain name	domain name\user name mydomain\john_smith

If you specify a simple name, users with this name in all domains are added as updaters.

If you right-click on a binary and select "Run as <updater user name>", the binary can execute and run as an updater only if the binary is added to the whitelist and authorized to run.

Specify files to be added as updaters

Specify files using the file name or checksum value.

This table describes the methods to specify a file to be added as an updater.

Method	Description
Specify the file name	If the file name is added as an updater, the updater permissions are applicable on the file name only and even if file path is changed, the updater permissions are in effect. You can specify the absolute or relative path of the file. However, if you specify the absolute path of the file as an updater, the updater permissions are applicable only on that specific path. For example, if dir\file.exe is specified, the updater rule applies only if file.exe is in a directory named dir. On the Windows platform, if you specify full path names containing the drive letter, the drive letter is not considered. For example, if you specify C:\foo\bar.exe, the updater rule is added for \foo\bar.exe only and does not include the drive letter.
Specify the file checksum (for Windows)	If the file checksum is added as an updater, it allows only the file with that checksum value to be added as an updater. This ensures that regardless of the source of the file, if the checksum value matches, the file is allowed to be added as an updater. You can add the file checksum is added as an updater only on the Windows platform. You can specify the checksum value to be added as an updater by using the `sadmin auth -a -u -c <checksumvalue>` command. For example, if you have multiple versions of the Adobe Acrobat product on your system but you want to run only a particular version, you can specify checksum value of the executable file as an updater. Adding checksum value of the executable file as an updater ensures the execution of only the required version of the product. See the *Using checksum values* section Specifying checksum value to be added as an updater is not supported for scripts. Scripts cannot be added as updaters by this method.

Update the default updaters using Finetune

Finetune utility enables you to update the default system configuration to execute the Commercial Off-The-Shelf (COTS) applications and add them to default updaters.

Finetune authenticates with the KnowledgeBase that these applications are authorized by Application Control to execute configuration changes. You can deploy Finetune using the batch file, `finetune.bat`, which is available where Application Control is installed. However, you can use this utility to add or remove the whitelisting customization to run a particular application.

To get help about the options that Finetune supports, run this command.

```
finetune.bat help
```

You can use this utility with Windows operating system only.

Task	Description

Add an application to default updaters.	To add an application to default updaters in the configuration file, run this command.
	`finetune.bat add A-Application`
	For example:
	`finetune.bat add A-McAfee`

Remove an application from default updaters.	To remove the application from default updaters in the configuration file, run this command.
	`finetune.bat remove A-Application`
	For example:
	`finetune.bat remove A-McAfee`

The attribute "A" refers to the application identifier. You can view all identifiers by running the `finetune.bat help` command.

Add suggested programs as updaters

You can identify a list of possible updaters that can be added on a Windows system. This feature is identified as *discover-updaters* in the feature list.

When running in the Enabled mode, Application Control protection can prevent a legitimate application from executing (if the required rules are not defined). Application Control tracks all such failed attempts made by authorized executable to modify protected files or run other executable files. You can review information for failed attempts to identify updater rules to allow legitimate applications to run successfully. This feature is available only on the Windows platform.

To get a list of components that can be added as updaters, run this command.

```
sadmin diag
```

The output shows the list of possible updaters that can be configured on the system to perform update operations.

McAfee recommends you to review the diag list to ensure that no restricted program or programs with generic names such as, `setup.exe`, are set as authorized updaters.

The output of executing this command displays these configuration parameters.

Symbol	Configuration Rules
!	The configuration for the program exists. The existing configuration is displayed on the next line.
*****	The configuration is for a *restricted* program, which can provide capability to change the system. Hence, such programs should have restricted configuration.
*** and !**	The configuration of the program exists but some modifications are required in the configuration to execute the program successfully.

Use this command to apply the diagnosed configuration changes.

```
sadmin diag fix
```

This command does not fix the rules marked with * (* is the restricted program)

To apply the diagnosed configuration changes for restricted programs, use this command.

```
sadmin diag fix -f
```

Restricted programs are Windows critical programs. For example, `services.exe`, `winlogon.exe`, `svchost.exe`, and `explorer.exe`.

List updaters

View the list of all components defined as updaters on the system.

```
sadmin updaters list
```

Remove updaters

Remove updaters added on the system to restrict them from making changes to the software components.

You can remove updaters by using any of these methods.

- Flushing all components from the updaters list by using this command:

  ```
  sadmin updaters flush
  ```

- Removing a specific component from the updaters list by using this command:

  ```
  sadmin updaters remove
  ```

  Specify the name or path of the component that you want to remove from the updaters list.

This table lists how to remove specific components as updaters.

Component	Example

Installers (available only on Windows)	Remove installers from the updaters list.
	`sadmin updaters remove <installername>`
	`sadmin updaters remove Ica32Pkg.msi`

Scripts	Remove scripts from the updaters list.
	`sadmin updaters remove <scriptname>`
	`sadmin updaters remove myscript12.bat`

Binaries	Remove binaries from the updaters list.
	`sadmin updaters remove <binaryname>`
	`sadmin updaters remove update.exe`
	On Windows, after using this command, restart the processes to remove them from the updaters list. However, on the Linux platform, process restart is not required.

Users (on Windows)	Remove users from the updaters list.
	`sadmin updaters remove -u <username>`
	`sadmin updaters remove -u john_smith`
	After using this command, restart the system to remove users from the updaters list.

Using certificates

Manufacturers of ATMs, Storage systems, and Point-Of-Sale systems, embed Application Control in their systems for protection. These manufacturers are the primary consumers who use the method of

adding certificate to perform update operations. However, this method of performing updates can also be used by a commercial enterprise.

Application Control allows trusted certificates that are associated with software packages to run on a protected system. After you add a certificate as a trusted or authorized certificate, you can run all software, signed by the certificate on a protected system without entering the Update mode. For example, if you add Adobe's code signing certificate, all software issued by Adobe and signed by Adobe's certificate are allowed to run.

To allow in-house applications to run on protected systems, you can sign the applications with an internal certificate and define the internal certificate as a trusted certificate. After you do so, all applications signed by the certificate are allowed.

You can also provide updater permissions to the certificate. All applications and binary files that are either added or modified on a system and signed by a certificate that has the updater permissions are automatically added to the whitelist. For more information on updaters, see the *Using updaters* section.

We recommend that you use this option judiciously because selecting this option ensures that all binary files signed by trusted certificate acquire updater permissions. For example, if you set the Microsoft certificate that signs the Internet Explorer application as an updater, Internet Explorer can download and execute any application. In effect, any files added or modified by an application signed by the trusted certificate (with updater permissions) are added to the whitelist automatically.

Extract certificates

The ScGetCerts utility is used to extract certificate from a binary file. This utility can also run on systems on which the whitelist is not created.

This utility is shipped with the product and it gets installed in the Application Control installation directory. Default location of this utility is `C:\Program Files\McAfee\Solidcore\Tools\ScGetCerts`.

Here is the syntax of the command to extract certificates:

```
scgetcerts.exe [<FILEPATH: filename|directory>] [OUTPUT PATH] [--cab] <-A> <-O> <-n|-c>
[<DOMAIN>] [<USERNAME>] [<PASSWORD>]
```

To extract certificate from a binary file, specify the file path with the file name or the directory path where the binary file is located. If you specify a directory name, certificate, or installer information, certificates are extracted recursively from all binary files to the specified directory. Also, specify the output directory path where you want to store the extracted certificates, installer information, or both.

This table describes the supported parameters:

Parameter	Description
`--cab`	Specify this parameter to extract certificate from a cab file. When you specify the `--cab` parameter, you must specify the `-O` parameter with it.
`-A`	Specify this optional parameter to extract all certificates from a binary file. By default, only the root certificate is extracted.
`-O`	Specify this optional parameter if only the certificates are required to be extracted and not the additional information. However, this parameter is not optional if `--cab` parameter is also specified.
`-c`	Specify this parameter to check if the path of the binary file is accessible on the network.
`-n`	Specify this optional parameter to provide authentication to the directory path on the network. The `-n` option is specified only when you specify the directory path.

Mention the domain, user name, and password when `-n` or `-c` parameter is used.

Add certificates

Add certificates as trusted or authorized certificates to run all software signed by those certificate on a protected system.

Run this command at the command prompt.

```
sadmin cert add
```

Use an existing certificate available to you or extract certificates from one or more signed binary files. You can extract certificate from any signed binary using `ScGetCert.exe` (`<Install_dir>\Tools\ScGetCert\ScGetCert.exe`). See the *Extract certificates* section.

Application Control supports only X.509 certificates.

Syntax	Description
`sadmin cert add <certificatename>`	Adds a certificate as a trusted certificate. For example: `sadmin cert add mcafee.cer`
`sadmin cert add –c <certificatecontent>`	Use this command with the –c argument to specify the certificate content as trusted. For example: `sadmin cert add –c` `MIIFGjCCBAKgAwIBAgIQbwr3oyE8ytuorcGnG3VhpDANBgkqhkiG9w0BAQUFADCB tDELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSAoYykwNDEuMCwGA1UEAxMl VmVyaVNpZ24gQ2xhc3MgMyBDb2RlIFNpZ25pbmcgMjAwNCBDQTAeFw0wNTEyMTAw MDAwMDBaFw0wNjEyMTAyMzU5NTlaMIHdMQswCQYDVQQGEwJVUzETMBEGA1UECBMK Q2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxIzAhBgNVBAoUGkFkb2JlIFN5 c3RlbXMgSW5jb3Jwb3JhdGVkMT4wPAYDVQQLEzVEaWdpdGFsIElEIENsYXNzIDMg LSBNaWNyb3NvZnQgU29mdHdhcmUgVmFsaWRhdGlvbiB2MjEcMBoGA1UECxQTSW5m b3JtYXRpb24gU3lzdGVtczEjMCEGA1UEAxQaQWRvYmUgU3lzdGVtcyBJbmNvcnBv cmF0ZWQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJcS8Tyuz/JSB6XlyV5Z d02tIo4iZoXANFxbGVXS3Yg4v7zIR8k2K0Tzzpmz3Y00Qr237nTslDnLb4rMx9Fr +DmH1Fq2CwQBCVTnrwbtdUyv2v977Fc05B09WEJvZmvcm22iNrfpCV5wqd7OTp1F qP2HIMa0ihztWAc3R9cn8xPPAgMBAAGjggF/MIIBezAJBgNVHRMEAjAAMA4GA1UdDwEB/ wQEAwIHgDBABgNVHR8EOTA3MDWgM6Axhi9odHRwOi8vQ1NDMy0yMDA0LWNy bC52ZXJpc2lnbi5jb20vQ1NDMy0yMDA0LmNybDBEBgNVHSAEPTA7MDkGC2CGSAGG +EUBBxcDMCowKAYIKwYBBQUHAgEWHGh0dHBzOi8vd3d3LnZlcmlzaWduLmNvbS9y cGEwEwYDVR0lBAwwCgYIKwYBBQUHAwMwdQYIKwYBBQUHAQEEaTBnMCQGCCsGAQUF BzABhhhodHRwOi8vb2NzcC52ZXJpc2lnbi5jb20wPwYIKwYBBQUHMAKGM2h0dHA6 Ly9DU0MzLTIwMDQtYWlhLnZlcmlzaWduLmNvbS9DU0MzLTIwMDQtYWlhLmNlcjAf BgNVHSMEGDAWgBQI9VHo+/49PWQ2fGjPW3io37nFNzARBglghkgBhvhCAQEEBAMC BBAwFgYKKwYBBAGCNwIBGwQIMAYBAQABAf8wDQYJKoZIhvcNAQEFBQADggEBAFY7 rAYt9WjCDFQ+YNHfnEZxav3zhGhTdTwqGpWZJh/wg9IgLnyRqMnoQNjDFsSCduxf FryGREMwCHI/PvEYq7hKZsUXSGWRNl+Auuomg0OFGG1ZlBv/ rWtQEbwmGKgtwXMDDm2IYY3t7O7shG3KW4qHg +Tq04pR8VGTGJodwZWEsj9JavErsujI7SFDMkj9xFz4 VD/ilkWF+AyzSLAyUTq797y/ q7TsG5Y1SeMtze49cVbJVRrbGtq3kSzF56adsA4Hvv2CjM379GkYX0Atro74YLEwcfwdAogZ +F+XtOU9CR48bPvkFP5xMLUJ46HPs1+u83Jk2lrr5OYmtMqd7f0`
`sadmin cert add –u <certificatename>`	You can add certificates as updaters using the `-u` argument. This command provides updater permissions to a certificate that is added as a trusted certificate. For example: `sadmin cert add –u mcafee.cer`

View certificates

View certificates in the Application Control certificate store to verify if the trusted certificates are added to the system.

Run these commands at the command prompt.

Syntax	Description
`sadmin cert list`	Lists the SHA1 of certificates that are added as trusted or authorized certificate in the Application Control certificate store.
`sadmin cert list -d`	Lists all details of the issuer and subject of the certificates added to the system.
`sadmin cert list -u`	Lists all certificates with updater permissions.

Remove certificates

Remove certificates from theApplication Control certificate store to remove their trusted or authorized status. Such certificates cannot run the software signed by the trusted certificates on a protected system.

Run this command at the command prompt.

`sadmin cert remove`

Syntax	Description
`sadmin cert remove <SHA1>`	Removes a certificate that is added as a trusted certificate using the SHA1 value. Specify the SHA1 value of the certificate to remove the certificate from the Application Control certificate store. For example: `sadmin cert remove 7ecf2b6d72d8604cf6217c34a4d9974be6453dff`
`sadmin cert remove –c <certificatecontent>`	Use this command with the –c argument to remove specified certificate content from the Application Control certificate store. For example: `sadmin cert remove –c` `MIIFGjCCBAKgAwIBAgIQbwr3oyE8ytuorcGnG3VhpDANBgkqhkiG9w0BAQUFADCB tDELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSAoYykwNDEuMCwGA1UEAxMl VmVyaVNpZ24gQ2xhc3MgMyBDb2RlIFNpZ25pbmcgMjAwNCBDQTAeFw0wNTEyMTAw MDAwMDBaFw0wNjEyMTAyMzU5NTlaMIHdMQswCQYDVQQGEwJVUzETMBEGA1UECBMK Q2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxIzAhBgNVBAoUGkFkb2JlIFN5 c3RlbXMgSW5jb3Jwb3JhdGVkMT4wPAYDVQQLEzVEaWdpdGFsIElEIENsYXNzIDMg LSBNaWNyb3NvZnQgU29mdHdhcmUgVmFsaWRhdGlvbiB2MjEcMBoGA1UECxQTSW5m b3JtYXRpb24gU3lzdGVtczEjMCEGA1UEAxQaQWRvYmUgU3lzdGVtcyBJbmNvcnBv cmF0ZWQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJcS8Tyuz/JSB6XlyV5Z d02tIo4iZoXANFxbGVXS3Yg4v7zIR8k2K0Tzzpmz3Y00Qr237nTslDnLb4rMx9Fr +DmH1Fq2CwQBCVTnrwbtdUyv2v977Fc05B09WEJvZmvcm22iNrfpCV5wqd7OTp1F qP2HIMa0ihztWAc3R9cn8xPPAgMBAAGjggF/MIIBezAJBgNVHRMEAjAAMA4GA1UdDwEB/ wQEAwIHgDBABgNVHR8EOTA3MDWgM6Axhi9odHRwOi8vQ1NDMy0yMDA0LWNy bC52ZXJpc2lnbi5jb20vQ1NDMy0yMDA0LmNybDBEBgNVHSAEPTA7MDkGC2CGSAGG +EUBBxcDMCowKAYIKwYBBQUHAgEWHGh0dHBzOi8vd3d3LnZlcmlzaWduLmNvbS9y cGEwEwYDVR0lBAwwCgYIKwYBBQUHAwMwdQYIKwYBBQUHAQEEaTBnMCQGCCsGAQUF BzABhhhodHRwOi8vb2NzcC52ZXJpc2lnbi5jb20wPwYIKwYBBQUHMAKGM2h0dHA6 Ly9DU0MzLTIwMDQtYWlhLnZlcmlzaWduLmNvbS9DU0MzLTIwMDQtYWlhLmNlcjAf BgNVHSMEGDAWgBQI9VHo+/49PWQ2fGjPW3io37nFNzARBglghkgBhvhCAQEEBAMC BBAwFgYKKwYBBAGCNwIBGwQIMAYBAQABAf8wDQYJKoZIhvcNAQEFBQADggEBAFY7 rAYt9WjCDFQ+YNHfnEZxav3zhGhTdTwqGpWZJh/wg9IgLnyRqMnoQNjDFsSCduxf FryGREMwCHI/PvEYq7hKZsUXSGWRNl+Auuomg0OFGG1ZlBv/ rWtQEbwmGKgtwXMDDm2IYY3t7O7shG3KW4qHg +Tq04pR8VGTGJodwZWEsj9JavErsujI7SFDMkj9xFz4 VD/ilkWF+AyzSLAyUTq797y/ q7TsG5Y1SeMtze49cVbJVRrbGtq3kSzF56adsA4Hvv2CjM379GkYX0Atro74YLEwcfwdAogZ +F+XtOU9CR48bPvkFP5xMLUJ46HPs1+u83Jk2lrr5OYmtMqd7f0`
`sadmin cert flush`	Removes all certificates from the Application Control certificate store.

Using checksum values

Override the protection applied to a system by authorizing certain binaries based on their checksum values.

Authorizing binaries by their checksum (SHA1) value allows them to execute on the protected system. If a binary is not added to the whitelist but configured as an authorized binary, it is allowed to execute on the system. Authorizing a binary based on the checksum (SHA1) value ensures that regardless of the source of a binary, if the checksum value matches, the binary is allowed to run.

You can also provide updater permissions to an authorized binary. Configuring an authorized binary as an updater provides the updater permissions in addition to the execution. An authorized binary that is configured as an updater is allowed to run and update software on a protected system. Installers can also be authorized by checksum and configured as updaters to allow them to install new software and update the software components. For example, if you authorize the installer for the Microsoft Office 2010 suite by checksum and also configure the installer as an updater, if the checksum matches, the installer is allowed to install the Microsoft Office suite on the protected systems.

Authorize binaries

Authorize binaries to allow them to execute on a protected system.

Use this command to authorize binaries:

```
sadmin auth -a [-t tagname] -c <checksumvalue>
```

Syntax	Description
`sadmin auth –a –c <checksumvalue>`	To specify the checksum value of the binary to be authorized. For example: `sadmin auth –a –c 803291bcc5aa45a0221b4016f62d63a26d3ee4af`
`sadmin auth –a [–t tagname] –c <checksumvalue>`	To include the tag name and the checksum value of the binary to be authorized. For example: `sadmin auth –a –t Win_up_schedule1 –c 803291bcc5aa45a0221b4016f62d63a26d3ee4af`
`sadmin auth –a –u –c <checksumvalue>`	To authorize a binary and also provide updater permissions. Specify the checksum value of the binary to be authorized and added as an updater. For example: `sadmin auth –a –u –c 803291bcc5aa45a0221b4016f62d63a26d3ee4af`

Ban binaries

Restrict binaries from executing on a protected system.

Use this command to ban binaries:

```
sadmin auth -b -c <checksumvalue>
```

Syntax	Description
`sadmin auth –b –c <checksumvalue>`	To specify the checksum value of the binary to be banned. For example: `sadmin auth –b –c 803291bcc5aa45a0221b4016f62d63a26d3ee4af`
`sadmin auth –b –t <tagname> –c <checksumvalue>`	To include the tag name and checksum value of the binary to be banned. For example: `sadmin auth –b –t AUTO_1 –c 583291bcc5aa45a0221b4016f62d63a26d3ee9at`

View authorized and banned binaries

List all authorized and banned binaries on a protected system.

Run this command at the command prompt:

```
sadmin auth -l
```

This command lists all authorized and banned binaries on a protected system. Also, it lists the binaries that are added as updaters.

Remove authorized or banned binaries

Remove authorized or banned binaries on a system to restrict the authorized binaries to execute and to remove the ban rules from the binaries that are banned.

You can remove the authorized or banned binaries using two methods:

Syntax	Description
`sadmin auth -r <checksumvalue>`	To specify the checksum value of the binary to be removed. For example: `sadmin auth –r 803291bcc5aa45a0221b4016f62d63a26d3ee4af`
`sadmin auth -f`	To flush all authorized or banned binaries. This command removes all binaries that are authorized or banned on a system.

Configure execution of installers

Manage the installation and uninstallation of software packages on a protected Windows system.

When Application Control is running in Enabled mode, it performs these actions.

- Windows optional components are blocked from installation or uninstallation except for few optional components for which the updater rules are defined.
- Installation by the **install** option, which appears when you right-click on an `.INF` file is blocked when the pkg-ctrl-inf feature is enabled.
- Installation of `.INF` files using certain exported functions from `setupapi.dll` or `advpack.dll` are blocked.

This table describes the associated features, listed in the features list:

Feature	Description
pkg-ctrl	Controls installation and uninstallation of all MSI-based installers. By default, this feature is enabled and can be managed by using the `sadmin features enable/disable pkg-ctrl` command. Restart the system after enabling or disabling this feature.
pkg-ctrl-inf	Prevents installation and uninstallation of all INF-based installers. By default, this feature is disabled and can be managed by using the `sadmin features enable/disable pkg-ctrl-inf` command. This is a subfeature of the pkg-ctrl feature. The pkg-ctrl-inf feature can be enabled only if the pkg-ctrl feature (parent feature) is enabled. Similarly, the pkg-ctrl-inf feature can be disabled only if the pkg-ctrl feature is disabled. You can install or uninstall Windows optional components and INF-based installers only in Update mode.

Any unauthorized attempt to install or uninstall a software package is prevented and an event is generated.

Add installers

To install software packages on a protected system, you can override the pkg-ctrl and pkg-ctrl-inf features and install the software packages.

After the pkg-ctrl feature is enabled, you cannot install software packages using standard commercial installers. However, you can use one of these methods to install software packages:

- Installer is added as an updater by using the *sadmin updaters* command.
- Installation/uninstallation of the software in the Update mode.
- Installer is marked as a signed binary.
- Authorize an installer by checksum and configure as an updater using the `sadmin auth -a -u -c <checksumvalue>` command.

Using this command overrides the impact of this feature on the executable.

Understand limitations

Understand limitations of configuring the execution of installers using the pkg-ctrl feature.

- When you uninstall a Windows optional component particularly, games software, **Add or Remove Programs** screen in Windows shows that the component is no longer installed. However, the component remains installed and is executable.
- When you click the **Next** or **Cancel** button on the **Windows Components Wizard** window, even without making any changes to the selected components, the following error message appears:

  ```
  McAfee Solidifier Prevented package modification by 'windows
  optional component manager' by user: <user_name>
  ```

- Utilities, such as WinDriver tools (`wdreg.exe`) can bypass the pkg-ctrl-inf feature and install `.INF` files.
- Optional Windows components can be installed using standard Windows tools such as **secedit** and **gpupdate**. By default, installation or uninstallation from these tools is not prevented.

- After installing fax services from **Add or Remove Programs**, fax services are installed but several write-protection errors related to `spoolsv.exe` are observed in the event viewer. However, the fax services work fine even with these errors. This is a specific case and happens when `rundll32.exe` is added as an updater.
- Application executables such as VNC server and client might not be able to execute. On running these applications, the following event is generated in the event log.

```
McAfee Solidifier prevented package modification by '<executable-name>' by user: <username>.
```

Using trusted directories

You can override the protection applied to a system using trusted directories. After you add directories as trusted directories, systems can run any software present in these directories.

On the Windows platform, Application Control tracks files and blocks the execution of binaries and scripts on the network directories. Application Control also supports tracking files on the Server Message Block (SMB) mount points. This feature is identified as *network-tracking* in the features list. By default, this feature is enabled and prevents the execution of binaries and scripts on network directories.

When this feature is disabled, execution of scripts on network directories is allowed. However, execution of binaries on network directories is not allowed. Also, write-protecting or read-protecting components on a network directory is not in effect.

What are trusted directories

On a protected system, you can add directories (local or network share) as trusted directories to run any software present in these directories. Trusted directories are identified by their Universal Naming Convention (UNC) path.

When do I add trusted directories

If you maintain shared folders containing installers for licensed applications on the internal network in your organization, add trusted directories for such network shares.

When enabled, Application Control tracks files over network shares and blocks their execution until the network share is added as a trusted share. Application Control also prevents protected systems from executing any code residing on a network share.

Additionally, if needed, you can also allow the software located at the UNC path to install software on the protected systems. For example, when logging on to a Domain Controller from a protected system, you will need to define `\\domain-name\SYSVOL` as a trusted directory (to allow execution of scripts and binaries).

Add trusted directories

Add directories as trusted directories to run any software present in these directories on a protected system. You can specify the absolute or relative path (on Linux only) to one or more directories. You can also specify paths to the directories located on network shares.

This table describes adding trusted directories.

Syntax	Description
`sadmin trusted –i <pathname1...pathnameN>`	Specify one or more paths to the directories to be added as trusted directories. You can also specify paths of the directories located on network shares. For example: • `sadmin trusted –i C:\Documents and Settings\admin\Desktop\McAfee` (for Windows) • `sadmin trusted –i \\192.168.0.1\documents` • `sadmin trusted –i /etc/security` (for Linux) For more information on specifying directory path, see the *Follow the guidelines to specify directory path* section.
`sadmin trusted –u <pathname1...pathnameN>`	Specify one or more paths to the directories to be added as trusted directories. This command adds all binaries and scripts present in the directories as updaters. You can also specify paths to the directories located on network shares. For example: • `sadmin trusted –u C:\Documents and Settings\admin\Desktop\McAfee` (for Windows) • `sadmin trusted –u \\192.168.0.1\documents` You can also add a trusted volume by specifying a volume name with this command to include all the binaries and scripts present in the specified volume as updaters. Use the `sadmin trusted –i -u <volumename>` command to specify the volume name.

Follow the guidelines to specify directory path

Network file system is supported on the Windows and Linux platforms. You can specify directory paths on a mounted network file system using these methods.

On the Windows platform:

Syntax	Description
`sadmin trusted –i \\server-name \share-name`	Specify the server name that has a network share. Also, specify the name of the network share.
`sadmin trusted –i \\server-name`	Specify the server name.
`sadmin trusted –i \\*`	Use this method to specify all network shares by all servers.

On the Linux platform, the network file system is mounted and then the local mount point is mentioned in trusted list. For example, A whitelisted server "A" can run a remote file `/mnt/ps` located on server "B" only if it has been added as a trusted share on server "A". To execute any file on the network share, establish the network share as a trusted share using the following command:

```
# sadmin trusted -i /mnt
```

Paths can include the wildcard character (*). However, it can only represent one complete path component. Here are a few examples.

- On the Windows platform, using \abc*\def is allowed while \abc*.doc, \abc*.*, or \abc\doc.* is not allowed.
- On the LINUX platform, using /abc/*/def is allowed while /abc/*.sh, /abc/*.*, or /abc/doc.* is not allowed.

List trusted directories

View the list of directories that are added as trusted directories on the system.

Run this command at the command prompt.

```
sadmin trusted -l
```

This command lists all trusted directories added on the system.

Exclude specific directories from the list of trusted directories

Exclude specific directories from the list of directories that you added as trusted directories on the system.

Run this command at the command prompt.

```
sadmin trusted –e <pathname1...pathnameN>
```

Use this command to specify one or more paths to the directories to be excluded from the list of trusted directories.

For example:

- `sadmin trusted –e C:\Documents and Settings\admin\Desktop\McAfee` (for Windows)
- `sadmin trusted –e \\192.168.0.1\documents`
- `sadmin trusted –e /etc/security` (for LINUX)

Remove trusted directories

Remove trusted directories to restrict those directories to run any software present in them.

You can remove the trusted directories using two methods described in this table.

Syntax	Description
`sadmin trusted –r <pathname1...pathnameN>`	To specify one or more paths to the directories to be removed as trusted directories. For example: • `sadmin trusted –r C:\Documents and Settings\admin \Desktop\McAfee` (for Windows) • `sadmin trusted –r \\192.168.0.1\documents` • `sadmin trusted –r /etc/security` (for Linux)
`sadmin trusted –f`	To flush all rules for trusted directories. If you specify this argument, all rules for the trusted directories are removed from the system.

Using trusted users

You can add users as updaters to allow users to perform update operations on a protected system. If you provide the updater permissions to a user, the user is defined as a trusted user.

This is allowed on the Windows platform only.

What are trusted users

Trusted user is an authorized Windows user with updater permissions to dynamically add to the whitelist. For example, add the administrator as a trusted user to allow the administrator to install or update any software. While adding the user information, you may also provide the domain details.

When do I add trusted users

Add specific users as trusted users when they are required to perform update operations on the protected system.

Of all the strategies available to allow changes to protected systems, this is the least preferred one because it offers minimal security. We suggest that you define trusted users judiciously because after a trusted user is added, there are no restrictions on what the user can modify or run on the system.

Add trusted users

Add trusted users to allow them to perform update operations on a protected system.

Run this command at the command prompt.

```
sadmin updaters add –u <username>
```

This table lists the supported arguments, descriptions, and examples.

<table>
<tr><th>Argument</th><th>Description</th></tr>
<tr><td>-u</td><td>Specify the –u argument to add a user as a trusted user (Windows). All update operations by the specified user name are allowed.
The table lists the type of user names that can be added as trusted users:
<table>
<tr><th>User Name</th><th>Example</th></tr>
<tr><td>Simple name</td><td>john_smith
<code>sadmin updaters add –u john_smith</code></td></tr>
<tr><td>Domain name</td><td>username@domain name
<code>sadmin updaters add –u john_smith@mycompany.com</code></td></tr>
<tr><td>Hierarchical domain name</td><td>domain name\user name.
sadmin updaters add –u mydomain\john_smith</td></tr>
</table>
If you specify a simple name, users with this name in all domains are added as updaters.
When you specify the –u argument, other arguments supported for sadmin updaters add command, such as -l, -p, -d, and –n are not applicable.
For more information on updaters, see the Using updaters section.</td></tr>
<tr><td>-t</td><td>Specify the –t argument to add a user with a tag name as an updater. Tag name is an identification label which will be present in the logs for all files processed by this rule.
<code>sadmin updaters add –t <tagname> –u <username></code>
<code>sadmin updaters add –t McAfee001 -u john_smith</code></td></tr>
</table>

List trusted users

List trusted users to view the list of all users who have updater permissions on the system.

Run this command at the command prompt.

```
sadmin updaters list
```

This command lists all trusted users and other components defined as updaters on the system.

Remove trusted users

When you remove a user as a trusted user, the updater permissions assigned to that user are removed.

Run this command at the command prompt.

```
sadmin updaters remove -u <username>
```

For example, `sadmin updaters remove -u john_smith`

After using this command, restart the system to remove updater permissions from the users.

Allow ActiveX controls to run

Typically, certain websites and programs require ActiveX controls to be installed on systems. By default, Application Control prevents the installation of ActiveX controls on a protected Windows system and the **ACTX_INSTALL_PREVENTED** event is generated.

Install and run ActiveX controls on a protected system using the ActiveX feature. This feature is enabled by default and available only on Windows operating system. Only the Internet Explorer browser is supported for ActiveX control installations. Installation of ActiveX controls is supported only for the Internet Explorer (32-bit) application. Simultaneous installation of ActiveX controls using multiple tabs of Internet Explorer is not supported.

How can I allow ActiveX controls

On a protected system, you can install and run ActiveX controls required for a website by adding the certificate of the website to Application Control certificate store.

Run this command at the command prompt to add the certificates.

```
sadmin cert add <certificatefilename>
```

For more information on adding certificates, see the *Using certificates* section.

Block execution of ActiveX controls

Uninstall ActiveX controls required for a website by removing the certificate of the website from the Application Control certificate store.

You can block the execution of allowed ActiveX controls using two methods.

- Block the execution of ActiveX control that was previously allowed (but not installed on the system).

  Run this command to remove the certificate from the Application Control certificate store.

  ```
  sadmin cert remove <certificatefilename>
  ```

  If ActiveX control is not installed on the system, removing the website's certificate blocks the execution of ActiveX control.

- Block the execution of ActiveX control when the certificate is added to the Application Control certificate store and ActiveX is already installed on the system.

  If the certificate is added to the Application Control certificate store and ActiveX is already installed on the system, follow these steps to deny the execution of ActiveX control:

  1. Run this command to remove the website's certificate from the Application Control certificate store.

     ```
     sadmin cert remove <certificatefilename>
     ```

  2. Remove the installed ActiveX control from **Add or Remove Programs** (Windows 2003, 2008, and XP) or **Programs and Features** (Windows Vista and 7).

Disable the ActiveX feature

Disable the ActiveX feature to stop running ActiveX controls.

Run this command to stop running the ActiveX feature.

```
sadmin features disable activex
```

You don't need to restart the system after enabling or disabling this feature.

Configure interpreters to allow execution of additional scripts

Application Control allows you to enable the execution control for scripts. You can establish custom associations between file-extensions and the interpreters that interpret the content of such files.

By default, Application Control supports the standard interpreters, and script files that are integrated with Windows operating system such as, batch files (.bat), command interpreter (.cmd), script files (.vbs), System files (.sys), Power shell files (.ps1), and Command files (.com).

Add interpreters

Add interpreters and scripts to allow the execution of additional scripts that you want to add to the whitelist. After adding the interpreters and scripts, the scripts are added to the whitelist and allowed to execute on the system.

Task	Description

Add interpreters	Run this command at the command prompt.
	`sadmin scripts add extension interpreter1 [interpreter2]...`
	When you establish an association, these files become the supported file types and they need to be whitelisted. Files having these extension can be only executed by these interpreters. For example:
	`sadmin scripts add .vbs wscript.exe cscript.exe`
	This command enables Application Control to enforce that only `wscript.exe` and `cscript.exe` can execute any `.vbs` script. The execution becomes effective immediately for all new interpreter instances that are initiated after running this command. Another interpreter can be added later to augment this list as seen in this example.
	`sadmin scripts add .vbs zscript.exe`
	If you attempt to add an interpreter that already exists on this list, no action is taken.

Enable execution control for '16Bit' binaries	Application Control supports a special tag '16Bit' as a synthetic extension for the 16-bit binaries. To enable the execution control for the 16-bit binaries, execute these commands. • `sadmin scripts add 16Bit wowexec.exe` • `sadmin scripts add 16Bit ntvdm.exe`

List interpreters

List the interpreters and scripts that are authorized to run when added to the whitelist.

Run this command at the command prompt.

```
sadmin scripts list
```

Sample output appears like this:

```
.ps1     "powershell.exe"
.bat     "cmd.exe"
.cmd     "cmd.exe"
.pif     "ntvdm.exe"
.sys     "ntvdm.exe"
.vbe     "cscript.exe" "wscript.exe"
16Bit    "ntvdm.exe" "wowexec.exe"
.vbs     "cscript.exe" "wscript.exe"
.exe     "ntvdm.exe"
```

Remove interpreters

Remove the interpreters for scripts on which execution control is not required.

Run this command at the command prompt.

```
sadmin scripts remove extension [interpreter1 [interpreter2]]...
```

- If you do not mention any interpreter, this command removes the extension for the entire list.
- Files having the extension for which the rule has been recently disabled remains in the whitelist until you run the `sadmin check -r` command or remove the files from the whitelist.

5 Configuring memory-protection techniques

Application Control offers multiple memory-protection techniques on the Windows platform to prevent zero-day attacks and to make sure the integrity of the running process executables and DLLs.

These techniques provide extra protection over the protection offered by native Windows features or signature-based buffer overflow protection products. These techniques are available on all Windows operating systems. At a high-level, the memory-protection techniques prevents from these types of exploits.

- Buffer overflow followed by direct code execution.
- Buffer overflow followed by indirect code execution using Return-Oriented Programming (ROP).

For a detailed and updated list of the exploits prevented by the memory-protection techniques, subscribe to McAfee Global Threat Intelligence (McAfee GTI) services security advisories.

Contents

Memory-protection techniques

Memory-protection techniques prevent malicious code execution and unauthorized attempts to gain control of a system through buffer overflow.

This table describes the memory-protection techniques with the supported operating systems, default states, and events.

<table>
<tr><th>Technique</th><th>Description</th></tr>
<tr><td>CASP - Critical Address Space Protection (mp-casp)</td><td>Renders code that is running from the non-code area. Code that is running from the non-code area is an abnormal event that usually happens due to a buffer overflow being exploited.

CASP is different from the Data Execution Prevention (DEP) feature available on the 64-bit Windows platforms.

The DEP feature prevents the code in a non-code area from executing (usually with the help of hardware). CASP allows such code to execute but disallows such code from making any meaningful API calls such as CreateProcess(), DeleteFile(), and others. Any meaningful exploit code will try to invoke at least one of these APIs and because CASP blocks them, the exploit fails to do any damage.

CASP technique is identified as mp-casp in the features list. Use the <code>sadmin features</code> command to view identifiers of the supported features.

You can bypass or restore CASP on executables. Also, you can list or flush the executables that are bypassed by CASP. For more information, see the Configure CASP section.

Supported operating systems: 32-bit — Windows 2003, Windows 2008, Windows XP, Windows XPE, WEPOS, Pos Ready 2009, WES 2009, Windows Vista, Windows 7, and Windows 7 Embedded

Default state: Enabled

Event generated: PROCESS HIJACKED</td></tr>
<tr><td>NX - No execute (mp-nx)</td><td>Uses the DEP feature to protect processes against exploits that try to execute code from writable memory area (stack/heap). In addition to native DEP, NX provides granular bypass capability and raises violation events.

Windows DEP is a memory-protection technique that prevents code from being run from a non-executable memory region. In most cases, code running from the non-executable memory region is an abnormal event. This scenario mostly occurs when a buffer overflow happens and the malicious exploit is attempting to execute code from these non-executable memory regions. DEP is available on 64-bit Windows platforms.

NX technique is identified as mp-nx in the features list. Use the <code>sadmin features</code> command to view identifiers of the supported features.

You can bypass or restore NX on executables. NX is only applicable for WoW64 (or 32-bit) processes. Also, you can list or flush the executables that are bypassed by NX. For more information, see the Configure NX section.

Supported operating systems: 64-bit — Windows XP, Windows 2003, Windows 2008, Windows 2008 R2, Windows Vista, Windows 7, and Windows 7 Embedded

This feature is not available on the IA64 architecture.

Default status: Enabled

Event generated: NX_VIOLATION_DETECTED</td></tr>
</table>

Technique	Description	
VASR - Virtual Address Space Randomization [mp-vasr (subfeatures: mp-vasr-rebasing, mp-vasr-relocation, mp-vasr-randomization)]	Although VASR is similar to the Address Space Layout Randomization (ASLR) technique available on the Windows platform, VASR is more than just ASLR. Windows ASLR randomizes the addresses where modules are loaded to help prevent an attacker from leveraging data from predictable locations. The problem with ASLR is that all modules have to use a compile-time flag to opt into this technique. VASR is available on obsolete Windows operating systems that do not support ASLR. The aim of this technique is that the malicious code that expects useful functions or data to be at fixed addresses does not find the functions or data. VASR stops ROP-based attacks by adopting the following approach: 1 Stack or heap randomization — Randomize the location of stack or heap in each process (mp-vasr-rebasing, mp-vasr-randomization). 2 Code relocation — Randomize the location of code in memory (mp-vasr-relocation). If an exploit tries to work with fixed addresses, the associated process might crash. If an application crashes while the mp-vasr-relocation feature is enabled, disable this feature and run the application again. Disabling this feature can enable the application to run again, if it has crashed. No event is generated. VASR technique is identified as mp-vasr in the features list. Use the sadmin features command to view identifiers of the supported features. You can bypass or restore VASR on the executables and DLLs. Also, you can list or flush the executables and DLLs that are protected by VASR (only for mp-vasr-rebasing). For more information, see the *Configure VASR* section.	
	Supported operating systems	• 32-bit — Windows XP and Windows 2003 • 64-bit — Windows XP and Windows 2003
	Default state	Disabled
	Event generated	No event is generated
Forced DLL Relocation (mp-vasr-forced-relocation)	Forces relocation of those Dynamic Link Libraries (DLLs) that have opted out of Windows' native ASLR feature. Certain malware rely on these DLLs that are always getting loaded at the same and known addresses. By relocating such DLLs, these attacks are prevented. Forced DLL Relocation technique is identified as mp-vasr-forced-relocation in the features list. Use the sadmin features command to view all identifiers of the supported features. You can bypass or restore Forced DLL Relocation on executables. List or flush the executables that are bypassed by Forced DLL Relocation. Also, you can bypass a DLL module that is loaded for the specified process. For more information, see the *Configure Forced DLL relocation* section.	
	Supported operating systems	Available on the Windows Vista (32- and 64-bit), Windows 7 (32- and 64-bit), Windows 2008 (both 32- and 64-bit), and Windows 2008 R2 (64 bit) operating system.
	Default state	Enabled
	Event generated	VASR_VIOLATION_DETECTED

Occasionally, some applications (as part of their day-to-day processing) might run code in an atypical way and hence can be prevented from running by the memory-protection techniques.

Contact McAfee support for information on other deprecated memory-protection techniques such as Mangling and Decoying.

Configure CASP

To protect the code in a non-code area from making any API calls, configure rules to add executables to CASP.

Task	Syntax	Description
Bypass executables from CASP	`sadmin attr add -c <filename1 ... filenameN>`	Specify one or more executables on which CASP must be bypassed. For example: `sadmin attr add -c alg.exe`
Restore CASP on executables	`sadmin attr remove -c <filename1 ... filenameN>`	Specify one or more executable on which CASP must be restored. For example: `sadmin attr remove -c alg.exe`
List the executables that are bypassed from CASP	`sadmin attr list -c`	Lists all the executables that are bypassed from CASP. For example: `sadmin attr list -c`
Flush the CASP bypass rules from all executables.	`sadmin attr flush -c`	Removes all the executables that are bypassed from CASP. For example: `sadmin attr flush -c`

Configure NX

To protect processes against exploits that try to execute code from writable memory area, configure rules to add executables to NX. This technique prevents code from being run from a non-executable memory region.

Task	Syntax	Description
Bypass executables from NX	`sadmin attr add -n <filename1 ... filenameN>`	Specify one or more executables on which NX must be bypassed. For example: `sadmin attr add -n alg.exe`
Restore NX on executables	`sadmin attr remove -n <filename1 ... filenameN>`	Specify one or more executables on which NX must be restored. For example: `sadmin attr remove -n alg.exe`

Task	Syntax	Description
List the executables that are bypassed from NX	`sadmin attr list –n`	Lists all executables that are bypassed from NX. For example: `sadmin attr list –n`
Flush NX bypass rules from all executables	`sadmin attr flush –n`	Removes the NX bypass rules from all executables. For example: `sadmin attr flush –n`

Configure VASR

VASR technique consists of mp-vasr-rebasing, mp-vasr-relocation, and mp-vasr-randomization subfeatures. You can configure any of these subfeatures to use VASR.

Configure rebasing

To prevent an attacker from leveraging data from predictable locations, configure rules to add one or more DLLs to rebasing. This technique makes sure rebasing of the addresses where modules are loaded.

Task	Syntax	Description
Add DLLs for Rebasing	`sadmin attr add –e <filename1 ... filenameN>`	Specify one or more DLLs on which rebasing their preferred base address is required. For example: `sadmin attr add –e devmgr.dll`
Remove Rebasing on DLLs	`sadmin attr remove –e <filename1 ... filenameN>`	Specify one or more DLLs on which Rebasing must be removed. For example: `sadmin attr remove –e devmgr.dll`
List the DLLs that are added to Rebasing	`sadmin attr list –e`	Lists all DLLs that are added to Rebasing. For example: `sadmin attr list –e`
Flush Rebasing rules from all DLLs	`sadmin attr flush –e`	Removes Rebasing rules from all DLLs. For example: `sadmin attr flush –e`

Configure relocation

To prevent ROP-based attacks using code relocation, configure rules to add one or more DLLs to relocation. This technique randomizes the location of code in memory.

Task	Syntax	Description
Bypass DLLs from mp-vasr- relocation	`sadmin attr add –r <filename1 ... filenameN>`	Specify one or more DLLs on which you bypass Relocation. For example: `sadmin attr add –r console.dll`
Restore Relocation on DLLs	`sadmin attr remove –r <filename1 ... filenameN>`	Specify one or more DLLs on which you restore Relocation. For example: `sadmin attr remove –r console.dll`

Task	Syntax	Description
List the DLLs that are bypassed from Relocation	`sadmin attr list -r`	Lists all DLLs that are bypassed from Relocation. For example: `sadmin attr list -r`
Flush Relocation rules from all DLLs	`sadmin attr flush -r`	Removes Relocation rules from all DLLs. For example: `sadmin attr flush -r`

Configure randomization

To prevent ROP-based attacks using stack or heap randomization, configure rules to add one or more DLLs to randomization. This technique randomizes the location of stack or heap in each process.

Task	Syntax	Description
Bypass executables from Randomization	`sadmin attr add -d <filename1 ... filenameN>`	Specify one or more executables on which you bypass Randomization. For example: `sadmin attr add -d alg.exe`
Restore Randomization on executables	`sadmin attr remove -d <filename1 ... filenameN>`	Specify one or more executables on which you restore Randomization. For example: `sadmin attr remove -d alg.exe`
List the executables that are bypassed from Randomization	`sadmin attr list -d`	Lists all executables that are bypassed from Randomization. For example: `sadmin attr list -d`
Flush Randomization rules from all executables	`sadmin attr flush -d`	Removes the Randomization rules from all executables. For example: `sadmin attr flush -d`

Configure forced DLL relocation

Certain malware rely on the DLLs that have opted out of Windows' ASLR feature and are always getting loaded at the same and known addresses. To prevent from such malware, configure rules to add one or more executables to forced DLL relocation. This technique forces relocation of those DLLs that have opted out of Windows' native ASLR feature.

Task	Syntax	Description
Bypass executables from Forced DLL relocation	`sadmin attr add -v <filename1 ... filenameN>`	Specify one or more protected components on which you bypass Forced DLL Relocation. For example: `sadmin attr add -v AcroRD32.exe`
Restore Forced DLL relocation on executables	`sadmin attr remove -v <filename1 ... filenameN>`	Specify one or more components on which you restore Forced DLL Relocation. For example: `sadmin attr remove -v AcroRD32.exe`

Task	Syntax	Description
List the executables that are bypassed from Forced DLL relocation	`sadmin attr list –v`	Lists all components that are bypassed from Forced DLL Relocation. For example: `sadmin attr list –v`
Flush Forced DLL relocation rules from all executables	`sadmin attr list –v`	Removes Forced DLL relocation rules from all executables. For example: `sadmin attr flush –v`
Bypass a DLL module that is loaded for a specific process	`sadmin attr add -o module=<DLLmodulename> -v <processname>`	Bypass the DLL module name for a process. For example: `sadmin attr add –o module= wuauserv.dll -v svchost.exe`

6 Maintaining your systems

Using Application Control features, you can perform tasks to maintain and manage the systems in your environment.

Contents

View product status and version

View Application Control status for product status details, such as operational mode, operational mode after restart, whitelist status. For managed configuration of the product, you can also see connectivity with McAfee ePO.

You can also view the Application Control version to see details of the installed product and the copyright information.

Task

1 Complete these steps to review Application Control status.

   a Type this command to view Application Control status. include `[Volume]` to view details of a single system. `sadmin status [volume]`

   b A message similar to this example displays the system details. The following table describes the fields and their meaning.

```
McAfee Solidifier:           Disabled
McAfee Solidifier on reboot: Disabled

ePO Managed:    No
Local CLI access: Recovered

[fstype]       [status]       [driver status]    [volume]
* NTFS        Solidified         Unattached           C:\
```

Status detail	Description
McAfee Solidifier	Specifies the operational mode of Application Control.
McAfee Solidifier on reboot	Specifies the operational mode of Application Control after system restart.
ePO Managed	Displays the connectivity status of Application Control with McAfee ePO. In standalone configuration of the product, this status is *No*.
Local CLI access	Displays the *lockdown* or *recovered* status of the local CLI. In standalone configuration of the product, this status is *Recovered*.
fstype	Displays the supported file systems for a volume.
status	Displays the current whitelist status for all the supported volumes on a system. If a volume name is specified, only the whitelist status for that volume is displayed.
driver status	Displays whether the Application Control driver is loaded on a volume. If the driver is loaded on a volume, status is *attached*; otherwise the status is *unattached*.
volume	Displays the volume names.

2 Type this command to view version and copyright details of application control product installed on the system `sadmin version`.

Manage the whitelist

An important part of system maintenance is managing the whitelist. You can perform various tasks to manage the whitelist.

Tasks

- *Configure the whitelist thread priority* on page 59
  Configure the whitelist thread priority before creating the initial whitelist on the Windows operating system. The whitelist thread priority determines the usage of system resources and the time required to create the whitelist.
- *Add components to the whitelist* on page 59
  Add new components to the initial whitelist to allow their execution on a protected system.
- *List the whitelisted components* on page 60
  View a list of all whitelisted files, directories, and drives/volumes on the system.
- *List the non-whitelisted components* on page 61
  View a list of all non-whitelisted files, directories, and volumes on the system.
- *Check and update the status of whitelisted components* on page 61
  Compare the current whitelist status and checksum values of whitelisted files, directories, and volumes with the status and values stored in the whitelist. If they are not current, you can update the whitelist and fix inconsistencies.
- *Remove components from the whitelist* on page 61
  Remove files, directories, or volumes from the whitelist, if needed.

Configure the whitelist thread priority

Configure the whitelist thread priority before creating the initial whitelist on the Windows operating system. The whitelist thread priority determines the usage of system resources and the time required to create the whitelist.

Task

1 Run this command at the command prompt.

```
sadmin config set SoPriority=<value>
```

2 Specify the `SoPriority` value.

This value determines the thread priority. By default, the thread runs on low priority (value of 0) and if you do not specify the thread priority, Application Control considers the default priority to create the whitelist.

This table describes the `SoPriority` values that you can specify.

Value	Priority
0	Low
1	Medium
2	High

The `SoPriority` value that you specify should be based on your preference. For example, if you want to create the whitelist in less time, you can specify the high value. However, doing this utilizes more system resources and can cause performance impact on the system. Similarly, if you specify the low value, Application Control takes more time to create the whitelist but causes minimal performance impact on the system. McAfee recommends you to specify the low value. This ensures that Application Control causes minimal performance impact on the systems.

Add components to the whitelist

Add new components to the initial whitelist to allow their execution on a protected system.

After the initial whitelist is created, execution is blocked for the components that are not included in the whitelist. If needed, add additional components to the whitelist.

Run this command at the command prompt.

```
sadmin solidify [<arguments> <components>]
```

Task

1 Specify the components as file names, directory names, or volume names.

This table describes the components that you can specify to be added to the whitelist.

Component	Description
File name	Adds files to the whitelist. For example, `sadmin solidify filename1 ... filenameN`
Directory name	Adds all supported files (recursively) under specified directories to the whitelist. For example, `sadmin solidify directoryname1 ... directorynameN`
Volume name	Adds all supported files (recursively) under specified system volumes to the whitelist. For example, `sadmin solidify volumename1 ... volumenameN`

2 Optionally, specify more supported arguments with this command.

For example,

```
sadmin solidify [ –q | –v ] filename1 ... filenameN | directoryname1 ...
directorynameN | volumename1 ... volumenameN
```

Here are the arguments descriptions:

- The `-q` argument displays only the error messages.
- The `-v` argument displays all messages.

List the whitelisted components

View a list of all whitelisted files, directories, and drives/volumes on the system.

Run this command at the command prompt.

```
sadmin list-solidified
```

This command lists all whitelisted components. You can narrow the results by specifying the names of files, directories and drive/volumes by adding these arguments to the command.

Syntax	Description
`sadmin list-solidified filename1 ... filenameN`	Lists all whitelisted files from the specified list of files. If only one file name is specified, this command shows the file name only if the file is whitelisted. Specify a set of files to list the whitelisted files from that file set.
`sadmin list-solidified directoryname1...directorynameN`	Lists all whitelisted files present in the specified directories.
`sadmin list-solidified volumename1...volumenameN`	Lists all the whitelisted files present in the specified drives/volumes.
`sadmin list-solidified [ -l ] filename1 ... filenameN | directoryname1...directorynameN | volumename1...volumenameN`	Lists details about the files, such as file type, file path, and file checksum.

List the non-whitelisted components

View a list of all non-whitelisted files, directories, and volumes on the system.

Run this command at the command prompt.

```
sadmin list-unsolidified
```

This command lists all the non-whitelisted components. You can narrow the results by specifying the names of files, directories and drive/volumes by adding these arguments to the command.

Syntax	Description
`sadmin list-unsolidified filename1 ... filenameN`	Lists all non-whitelisted files from the specified list of files. If only one file name is specified, this command shows the file only if the file is non-whitelisted. Specify a set of files to list non-whitelisted files from that file set.
`sadmin list-unsolidified directoryname1...directorynameN`	Lists all non-whitelisted files present in the specified directories. Specify directory names with this command to list all the non-whitelisted files in the specified directories.
`sadmin list-unsolidified volumename1...volumenameN`	Lists all non-whitelisted files present in specified volumes. Specify volume names with this command to list all non-whitelisted files in the specified volumes.

Check and update the status of whitelisted components

Compare the current whitelist status and checksum values of whitelisted files, directories, and volumes with the status and values stored in the whitelist. If they are not current, you can update the whitelist and fix inconsistencies.

If the components in the whitelist are modified or removed and the whitelist is not updated, the execution of these components is blocked by Application Control. This results in inconsistencies in the whitelist.

Run this command at the command prompt.

```
sadmin check [ -r ] file | directory | volume
```

This command checks the whitelist for consistency and compares the current whitelist status and checksum values of the components with the whitelist status and checksum values stored in the whitelist.

You can narrow the results by specifying the names of files, directories and drive/volumes by adding these arguments to the command.

You can also specify the `-r` argument with this command. This argument fixes all inconsistencies by updating the whitelist with the latest checksum values of the components and adds the components to the whitelist, if the components are not already present. If you do not specify a component, inconsistencies in all supported drives/volumes are fixed.

Remove components from the whitelist

Remove files, directories, or volumes from the whitelist, if needed.

Run this command at the command prompt.

```
sadmin unsolidify
```

This command removes all components from the whitelist. When you remove components from the whitelist, they are no longer protected by Application Control.

You can narrow the results by specifying the names of files, directories and drive/volumes by adding these arguments to the command.

We recommend that you do not remove a system drive or volume from the whitelist.

Syntax	Description
`sadmin unsolidify filename1 ... filenameN`	Removes one or more files from the whitelist.
`sadmin unsolidify directoryname1 ... directorynameN`	Removes all supported files in one or more directories from the whitelist.
`sadmin unsolidify volumename1 ... volumenameN`	Removes all the supported files in one or more system volumes from the whitelist.
`sadmin unsolidify [ –v ] filename1 ... filenameN | directoryname1 ... directorynameN | volumename1 ... volumenameN`	Displays all messages.

Advance exclusion filters

You can use a combination of conditions to define advanced filters to exclude reporting of changes.

For example, you might want to monitor the changes made to the `tomcat.log` file by all programs except the `tomcat.exe` program. To achieve this, define an advanced filter to exclude all changes made to the log file by `tomcat.exe`. This means that you receive only events when the log file is changed by other (non-owner) programs.

In this case, the defined filter is similar to exclude all events where `filename` is `<log-file>` and `program name` is `<owner-program>`. Use AEFs to prune routine system-generated change events that are not relevant for your monitoring or auditing needs.

Several applications, particularly the web browser, maintain the application state in registry keys and regularly update several registry keys. For example, the ESENT setting is routinely modified by the Windows Explorer application and it generates the Registry Key Modified event. These state changes are regular and need not be monitored or reported. Defining AEFs allows you to eliminate any events that are not necessary for compliance, and ensures the event list includes only meaningful notifications.

Add AEFs

Limit the notifications you receive by adding an advanced filter that excludes changes made to specified components.

Run this command at the command prompt.

```
sadmin aef add [component <condition> value]
```

Specify the component, condition, and value with this command.

Component	Value	Description
File	File path	`sadmin aef add [file <condition> PATH]`
Registry key	Registry path	`sadmin aef add [reg <condition> PATH]`
Process	Process path	`sadmin aef add [process <condition> PATH]`

Component	Value	Description
User	User name	`sadmin aef add [user <condition> USER-NAME]`
Event	Event name	`sadmin aef add [event equals EVENT_NAME]`
Multiple components	Supported values for the specified components	Specify the *and* operator to include multiple components to the filter rule. For example: `sadmin aef add [file <condition> PATH] and [reg <condition> PATH] and [process <condition> PATH] and [user <condition> USER-NAME] and [event equals EVENT_NAME]`

Specify one or more conditions with the components to add AEFs. The filter rule is based on the specified conditions.

Condition	Description
equals	Add all components with the specified name. Only this condition is valid to add events as AEFs. For example: `sadmin aef add file equals C:\Program Files\Microsoft Download Manager \MSDownloadManager.exe`
begins	Add all components whose paths begin with the specified characters. For example: `sadmin aef add file begins C:\Program Files\Adobe`
ends	Add all components whose paths end with the specified characters. For example: `sadmin aef add file ends rtf`
contains	Add all components whose paths contain the specified characters. For example: `sadmin aef add process contains svchost.exe`
doesnt_contain	Add all components whose paths do not contain the specified characters. For example: `sadmin aef add reg doesnt_contain CurrentControlSet`

List AEFs

List AEFs to review all added AEFs with the specified conditions.

Run this command at the command prompt.

```
sadmin aef list
```

This command lists all AEFs with the specified conditions that are added to a system.

Remove AEFs

Remove AEFs to include the excluded notifications for the changes made to the specified components. After removing the AEFs, you receive events for all changes made to the excluded components. However, this results in inclusion of non-meaningful events to the events list.

You can remove one, multiple, or all AEFs using these methods.

Task

1 Run this command to remove one or multiple AEFs.

```
sadmin aef remove [component <condition> value]
```

2 Run this command to remove all AEFs.

```
sadmin aef flush
```

Manage product features

When Application Control is installed on the system, the product features are in their default status. You may need to change the status to allow configuration changes.

Tasks

- *Review features* on page 64
  Review the list of all Application Control features and their status (enabled or disabled) on your system.
- *Enable or disable features* on page 66
  Enable features that are currently disabled on the system to protect the system. Also, you can disable features that are enabled on the system to remove the applied protection. After disabling a feature, the system will no longer be protected by that feature.

Review features

Review the list of all Application Control features and their status (enabled or disabled) on your system.

Run this command at the command prompt.

```
sadmin features list
```

The features list is displayed on the screen.

Starting from the Application Control 6.0.0 release, the features list has been minimized to show only the features that require modifications regularly.

Feature	Description	Default Status	Supported Operating System
activex	Installs and runs ActiveX controls on the protected system. Only the Internet Explorer browser is supported for the ActiveX control installations. Simultaneous installation of ActiveX controls using multiple tabs of Internet Explorer is not supported.	Enabled	Windows
checksum	Compares the checksum of the file to be executed with the checksum stored in the whitelist.	Enabled	Windows and Linux

Feature	Description	Default Status	Supported Operating System
deny-read	Read-protects the specified components. When this feature is applied on components, they cannot be read. Read protection works only when Application Control is running in the Enabled mode.	Disabled	Windows and Linux
deny-write	Write-protects the specified components. When this feature is applied on the components, they are rendered as read-only, to protect your data.	Enabled	Windows and Linux
discover-updaters	Generates a list of potential updaters that can be included in the system. When running in Enabled mode, Application Control protection might prevent a legitimate application from executing (if the necessary rules are not defined). This feature tracks all such failed attempts made by authorized executable to modify protected files or run other executable files and generates a list of possible updaters that can be configured on the system to perform an update.	Enabled	Windows
integrity	Protects Application Control files and registry keys from unauthorized tampering. Allows the product code to run even when the components are not present in the whitelist. Ensures that all product components are protected. Prevents accidental or malicious removal of components from whitelist to ensure that the product does not become unusable. In update mode, this feature is disabled to facilitate product upgrades.	Enabled	Windows and Linux
mp	Protects running processes from hijacking attempts. Unauthorized code injected into a running process is trapped, halted, and logged. Attempts to gain control of the system through buffer overflow and similar exploits are rendered ineffective.	Enabled	Windows
mp-casp	Renders useless code that is running from the non-code area, which happens due to a buffer overflow being exploited on 32-bit Windows platforms.	Enabled	Windows
mp-vasr	Randomizes the location of stack or heap (Stack or heap randomization) in each process. Also, randomizes the location of code in the memory (Code relocation). VASR can stop return-oriented programming (ROP)-based attacks using stack or heap randomization and code relocation approach. VASR is similar to the Address Space Layout Randomization (ASLR) technique available on the Windows platform. However, VASR is more than just ASLR. Windows ASLR randomizes the addresses where modules are loaded to help prevent an attacker from leveraging data from predictable locations. The problem with ASLR is that all modules have to use a compile-time flag to opt into this.	Enabled	Windows Vista Windows 7
		Disabled	Windows Server 2003 Windows XP Windows Server 2008
network-tracking	Tracks files over network directories and blocks the execution of scripts over network directories. By default, this feature is enabled and prevents the execution of scripts over network directories. When this feature is disabled, execution of scripts over network directories is allowed. Also, write-protecting or read-protecting components over a network directory will not be effective.	Enabled	Windows

Feature	Description	Default Status	Supported Operating System
pkg-ctrl	Manages the installation and uninstallation of all MSI-based installers. When this feature is enabled, you cannot install software packages using standard commercial installers.	Enabled	Windows
script-auth	Prevents the execution of supported script files that are not present in the whitelist. Only whitelisted script files are allowed to execute on the system. For example, supported script files such as .bat, .cmd, .vbs (on Windows), and script files containing #! (hash bang) for supported local file systems (on Linux) are added to the whitelist and are allowed to execute.	Enabled	Windows and Linux

Enable or disable features

Enable features that are currently disabled on the system to protect the system. Also, you can disable features that are enabled on the system to remove the applied protection. After disabling a feature, the system will no longer be protected by that feature.

Task

1 Run this command to enable a feature.

```
sadmin features enable <featurename>
```

2 Run this command to disable a feature.

```
sadmin features disable <featurename>
```

Making emergency changes

Run Application Control in Update mode to perform emergency changes on a protected system.

- When the product is in effect, you can allow scheduled or emergency changes to the system and track the changes made to the system by running the product in Update mode.
- We recommend you that you use Update mode to make the changes that cannot be made when Application Control is running in Enabled mode. Whenever possible, use other preferred methods, such as trusted users, directories, certificates, checksum values, or updaters to allow changes.
- In Enabled mode, if you install new software or add new binary files, the files are not added to the whitelist or allowed to execute unless you use a trusted method to add them. However, if you install or uninstall software, or add new binary files in Update mode, all changes are tracked and added to the whitelist.
- To authorize or approve changes to the system, a change window is defined, where users and programs can make changes to the system. I Update mode allows you to schedule software and patch installations, remove or modify software, and dynamically update the whitelist. Memory-protection techniques are enabled in Update mode, so that running programs cannot be exploited.

- Application Control generates the FILE_SOLIDIFIED event for files added during Update mode, and FILE_UNSOLIDIFIED event for files deleted during Update mode. Also, when the write-protected files are modified or renamed in Update mode, corresponding update mode events, such as FILE_MODIFIED_UPDATE and FILE_RENAMED_UPDATE are generated.
- From Update mode, you can switch to Enabled or Disabled mode.

Switch to Update mode

Switch Application Control to Update mode to perform scheduled or emergency changes on a system.

Task

- If the product is in Enabled or Disabled mode, perform these steps to switch to Update mode.

  a Type the `sadmin begin-update [workflow-id [comment]]` command.

  Optionally, specify these arguments with the command.

Attribute	Description
Workflow-id	Specify a workflow ID for the current Update mode session. This is an identification ID that can be used for a Change Management or Ticketing System. If you do not provide the workflow ID, the workflow ID is set to an automatically generated string, AUTO_n, where n is a number that is incremented each time an update window is opened.
Comment	Specify a comment that describes the current Update mode session. This information can be used for a Change Management or Ticketing System.

  b Press **Enter**.

  If Application Control was in Enabled mode, it is switched to Update mode.

  If Application Control was in Disabled mode, perform these extra steps.

  - Restart the system.

    When you restart the system, the product is switched to Update mode. Restarting the system is a recommended way to switch to Update mode.

  - Restart the Application Control service.

    Alternatively, you can restart theApplication Control service to switch to Update mode. However, only limited features will be enabled after service restart. Key product features, such as memory-protection will not be enabled and to enable all features, you need to restart the system.

  The product is switched to Update mode.

Exit Update mode

Exit Update mode after making scheduled or emergency changes, patch installations, or software updates on your system.

Run this command at the command prompt.

```
sadmin end-update
```

Enable password protection

Restrict users from running critical `sadmin` commands by enabling password protection. When password protection is enabled, Application Control allows these critical commands to run only when

the user enters in the correct password. If you do not need password protection, remove the password, which allows users to run all `sadmin` commands.

Passwords are encrypted with the SHA2 hashing algorithm. To protect password details, a random number is added to the password before the hash is computed.

The SHA5012 encryption algorithm, a subset of SHA2, generates a hash of 512 bits, which protects the password from "rainbow table" attacks.

Task

1 Complete these steps to set a new password.

a Type the `sadmin passwd` command.

When you set a password, users can no longer run critical commands without providing the correct password. Only a limited set of non-critical commands can run without the password.

You can use the `-z` switch to prevent the system from prompting for the password. It can be used in all CLI commands.

b Press **Enter**.

- If you already set the password, Application Control prompts you to enter your password. Type the old password and press **Enter**. You are now asked to set the new password and retype it.
- If you didn't set the password earlier, Application Control prompts you to enter a new password. Set the new password and retype it.

2 Complete these steps to remove the password.

a Type the `sadmin passwd -d` command.

b Press **Enter**.

Review changes using events

Application Control generates events for all changes that are made to a protected component. Use events to review the changes and diagnose unauthorized execution attempts and failures on the system.

Whenever an attempt is made to access or change a protected resource, an event is generated on the system. Application Control tracks changes on the system and records events. For example, every time the attributes or contents change for a protected file, a corresponding event is generated.

Configure event sinks

Events are stored at locations called *event sinks*.

You can log events in many types of event sinks, including:

- Operating system log (oslog)
- System controller (sc)

When sc event sink is enabled, it sends the events to McAfee ePO.

- Debug output (debuglog)
- Popup (Windows only)

You can track changes that occur on the system by reviewing the events. Refer to *Application Control event list* section for a list of all the Application Control events that can be generated and their description.

See the event sinks configured to events by viewing the event sink details. If more events are needed, you can add them to a specified event sink. If you do not want the events to be logged to a specific event sink, you can stop the logging of events to that event sink.

Tasks

- *Add an event* on page 69
  Add an event by specifying both the event name and the event sink where you want to log the event.
- *View the event sink details* on page 69
  View the event sink details for all events generated on the system. You can view the associated event sinks for each event.
- *Remove an event* on page 69
  Remove an event by specifying both the event name and the event sink from where you want to remove the event.

Add an event

Add an event by specifying both the event name and the event sink where you want to log the event.

Run this command at the command prompt.

```
sadmin event sink -a <event_name> <sink_name>
```

Specify the event name and the event sink with this command. The specified event will be added to the event sink.

View the event sink details

View the event sink details for all events generated on the system. You can view the associated event sinks for each event.

Run this command at the command prompt.

```
sadmin event sink
```

Event sink details configured on the system for all events are listed.

Remove an event

Remove an event by specifying both the event name and the event sink from where you want to remove the event.

Removing an event from an event sink allows you to stop logging the event to that event sink. Perform these steps to remove an event from an event sink.

Run this command at the command prompt.

```
sadmin event sink -r <event_name> <sink_name>
```

Specify the event name and event sink with this command.

Configure the event cache size

Configure the event cache size to set the upper and lower limit of the event cache. Events are stored in the cache before being placed in the event sinks.

Application Control buffers the change events to deal with outages. By default, the buffer limit is set to 2 MB. When the buffer limit nears the threshold, an event is logged on the system log stating that the cache is about to overflow. When this buffer limit exceeds the threshold, new events are not logged until the number of events in the buffer falls below its high watermark.

Tasks

- *Set the event cache size* on page 70
  Set the event cache size to define the buffer limit for the event cache.
- *Define the upper and lower limits for the event cache* on page 70
  Set the upper and lower limits for the event cache. When the limits are set, an alert is generated to notify that the cache is about to overflow or recovered from an overflow.

Set the event cache size

Set the event cache size to define the buffer limit for the event cache.

Perform these steps to set event cache size.

Task

1 Type the `sadmin config set EventCacheSize=<value>` command.

  Specify a value for the *EventCacheSize* parameter. This value determines the event cache size.

2 Press **Enter**.

Event cache size is set to the specified value.

Define the upper and lower limits for the event cache

Set the upper and lower limits for the event cache. When the limits are set, an alert is generated to notify that the cache is about to overflow or recovered from an overflow.

Task

1 Type the `sadmin config set EventCacheWMHigh=<value>` command to set an upper limit.

  Specify a value for the *EventCacheWMHigh* parameter. The specified value for this parameter should be between 50% to 100% of the event cache size.

2 Type the `sadmin config set EventCacheWMLow=<value>` command to set an lower limit.

  Specify a value for the *EventCacheWMLow* parameter. The specified value for this parameter should be above 20% of the event cache size. The value of the low watermark level must always be less than the value of the high watermark level.

3 Press **Enter**.

View events

View events specific to Application Control on your system to track changes related to the product.

Perform these steps to view Application Control events on the Linux and Windows platforms.

Task

1 Perform these steps to view events on the Linux platform.

   a Navigate to the `/var/log/messages` directory.

   b View the Application Control events.

2 Perform these steps to view events on the Windows platform.

   a Open the **Event Viewer** application.

Platform	Navigation
Windows Server 2003 Windows XP Windows Server 2008	Select **Start** | **Run** and type `eventvwr`.
Windows Vista Windows 7	Select **Start** | **Search** and type `eventvwr`.

   b Press **Enter**.

   c Perform these steps based on your platform.

Platform	Step
Windows Server 2003 Windows XP Windows Server 2008	1 From the navigation pane, select **Application**. All application events categorized by type, date, time, source, category, event, user, and computer columns are displayed. 2 Under the **Source** column, double-click **McAfee Solidifier** event to view its description. Events are specific to Application Control and are listed by order of occurrence, with most recent first.
Windows Vista Windows 7	1 From the navigation pane, expand Window Logs and select **Application**. Application events are categorized by level, date and time, source, event ID, and task category columns are displayed. 2 Under the **Source** column, look for the **McAfee Solidifier** events. These events are specific to Application Control and are listed based on the order of occurrence. The most-recent event is listed on the top. 3 Double-click an event to view its description.

Configuring log files

Application Control generates log messages for all actions and errors related to the product. These log messages are stored in log files that are used for troubleshooting errors.

This table describes the types of log files present on the system.

Log file	Operating system	Path	Description
`solidcore.log`	Windows Server 2003 Windows Server 2008 Windows XP	`<system drive>\Documents and Settings\All users\Application Data\McAfee\Solidcore\Logs`	After the product is deployed on a system, a log file named `solidcore.log` is created in the Logs folder (Windows) or solidcore directory (Linux). This file is also known as debuglog. You can configure the `solidcore.log` file size and number of `solidcore.log` files that you need to create on the system. Configuring log files is applicable only to the `solidcore.log` file. You cannot change the configuration of any other log file.
	Windows Vista Windows 7	`<system drive>\ProgramData\McAfee\Solidcore\Logs`	
	Linux	`/var/log/mcafee/solidcore/`	
`s3diag.log` (Windows only)	Windows Server 2003 Windows Server 2008 Windows XP	`<system drive>\Documents and Settings\All users\Application Data\McAfee\Solidcore\Logs`	`s3diag.log` file stores logs for all operations performed on the supported files.
	Windows Vista Windows 7	`<system drive>\ProgramData\McAfee\Solidcore\Logs`	
`S3setup.log` (Windows)/ `solidcoreS3_install.log` (Linux)	Windows (all supported versions)	`<system drive>\Windows`	Application Control installation logs are stored in this file. If installation fails on the Linux platform, the file is stored at: `/tmp/solidcoreS3_install.log`. If installation is successful on the Linux platform, the file is stored at: `/var/log/mcafee/solidcore/solidcoreS3_install.log`
	Linux	• `/tmp/solidcoreS3_install.log` • `/var/log/mcafee/solidcore/solidcoreS3_install.log`	

Configure the log file size

Configure the default `solidcore.log` file size to allow more log messages to be stored.

Run this command at the command prompt.

```
sadmin config set LogFileSize=<size>
```

Specify a value for the *LogFileSize* parameter in kilobytes. The specified value signifies the log file size. The default file size is 2048 KB. When the `solidcore.log` file size reaches 2048 KB, this log file is renamed to `solidcore.log.1` and a new log file `solidcore.log` is created where the new log messages are logged. You can change the default file size as needed.

Configure the number of log files

Set the maximum number of files that are needed on the system.

Run this command at the command prompt.

```
sadmin config set LogFileNum=<value>
```

Specify a value for the *LogFileNum* parameter. By default, the parameter value is set to 4 to allow you to create maximum five log files, namely `solidcore.log`, `solidcore.log.1`, up to `solidcore.log.4`. There is no limit for the maximum number of log files you can create on the system. The logs are placed in chronological order, with solid core, log always having the most recent messages.

Configuring log files is applicable only to the `solidcore.log` file. You cannot change the configuration of `s3diag.log` file.

Runtime environment of the system

Review the run-time environment and system configuration using the ScAnalyzer utility. When you install the product, this utility automatically checks whether the host system satisfies the prerequisites to install the product.

The ScAnalyzer checks the system for :

- Operating system version
- Service pack level
- Processor and memory configuration
- Installed applications
- Installed hotfixes
- Installed services
- System devices
- Running processes
- Open network ports
- Incompatible applications (for Windows)

When you execute ScAnalyzer on the Windows platform, it also compares the software installed on the system with an internal prepackaged checklist to create a file `scanalysis.bat`. This batch file contains the whitelist customization rules for the installed applications to run smoothly.

Run ScAnalyzer

Run ScAnalyzer on the Windows and Linux platforms to get details of the run-time environment and system configuration.

Perform these steps to run ScAnalyzer.

Task

1 Navigate to this location.

- On Windows: `C:\Program Files\McAfee\Solidcore\Tools\ScAnalyzer.`
- On Linux: `/usr/local/mcafee/solidcore/tools/scanalyzer/`

These are the default installation paths for this utility on the Windows and Linux platforms.

2 Run this command.

- On Windows: `scanalyzer`
- On Linux: `# ./scanalyzer.sh`

You can specify more parameters with this command as described in this table.

Parameter	Description
-h	Displays help for using ScAnalyzer.
-v	Displays ScAnalyzer version.
[-c <checklist>]	Detects if any application in the checklist is installed on the system. (Windows only)
-d	Displays the difference in running services, processes, and open ports in two separate ScAnalyzer reports. For Linux use command -d<rep1 rep2>.
-o <output file>	Writes output to the output file. If no file is specified, output is printed to screen (Windows) or console (Linux).
-s <scan_file>	Detects if any of the applications in the checklist is present in the ScAnalyzer report. (Windows only)
-q	Runs the ScAnalyzer in quiet mode.
-n	Prevents time stamp to be added to output file name.

The ScAnalyzer report is generated.

Review the ScAnalyzer report

Review the ScAnalyzer report to view details of your system configuration.

After running the command, the ScAnalyzer utility generates report in a data file. This report is known as ScAnalyzer report and contains details of your system configuration.

Perform these steps to review ScAnalyzer report.

Task

- Navigate to this path:
  - On Windows: `<System Drive>\Program Files\McAfee\Solidcore\Data`

    The file name is `scan_<machine_name>_<date>_<time>.txt`.
  - On Linux: `/usr/local/mcafee/solidcore/tools/scanalyzer/data`

    The file name is `report-<machine_name>-<date>_<time>.`

    Check these items during the manual review of the ScAnalyzer report.
    - Operating system version and the service pack level for the supported version.
    - Hotfixes needed to install the product.
    - Anti-virus software, which might update the code during execution. Check the ScAnalyzer output for these applications and change the system configuration to add them as updaters.

Managing mass deployments and system upgrades

Export the system configuration to a configuration file and deploy the file to multiple systems at once.

The configuration file stores all configuration parameters for a system in a standard format. Examples of configuration items are event cache size, SO priority, log file size, and log file path. You can add, delete, or change the configuration parameter values of the exported files and set new values.

You can also import the modified configuration file to enable new parameters and upgrade your system configuration. Importing the configuration file is allowed on the same system or on other systems but the system images should be identical.

For some system-specific rules are displayed. Modifying them is not straightforward and not recommended. Such parameters include **Monitoring Rules List**, **Capability Rules List**, **Bypass List**, and **Updaters List**.

View the existing configuration parameters

View all existing configuration parameters on your system.

Run this command at the command prompt.

```
sadmin config show
```

Application Control lists the configuration parameters items. For example:

- On the Windows platform:

```
  CustomerConfig                 30 (0x1e)
  ssLangId                       Default
  CustomizedEventCacheSize       1000 (0x3e8)
  EventCacheSize                 2 (0x2)
  EventCacheWMHigh               90 (0x5a)
  EventCacheWMLow                70 (0x46)
  FailSafeConf                   0 (0x0)
* FeaturesEnabled                2233943118021567 (0x7efc269ff83bf)
* FeaturesEnabledOnReboot        2233943118021567 (0x7efc269ff83bf)
* FeaturesInstalled              3659168154103807 (0xcfffe79ffafff)
* FileAttrCTrack                 5024 (0x13a0)
* FileDenyReadOptions            1024 (0x400)
* FileDenyWriteOptions           4831 (0x12df)
  FileDiffMaxSize                10 (0xa)
  FipsMode                       0 (0x0)
* LockdownStatus                 0 (0x0)
  LogFileNum                     4 (0x4)
* LogFilePath                    C:\Documents and Settings\All Users\Application Data
\McAfee\Solidcore\Logs
  LogFileSize                    2048 (0x800)
* RTEMode                        0 (0x0)
* RTEModeOnReboot                0 (0x0)
  SoPriority                     0 (0x0)
* WorkFlowId                     None
  ObservationLogsRotationMillis60000 (0xea60)
```

- On the Linux platform:

```
  CustomerConfig               0 (0x0)
  EventCacheSize               2 (0x2)
  EventCacheWMHigh             90 (0x5a)
  EventCacheWMLow              70 (0x46)
  FailSafeConf                 0 (0x0)
* FeaturesEnabled              47269939391728575 (0xa7efc269ff8bbf)
* FeaturesEnabledOnReboot      47269939391728575 (0xa7efc269ff8bbf)
* FeaturesInstalled            48695164427808767 (0xacfffe79ffafff)
* FileAttrCTrack               5024 (0x13a0)
* FileDenyReadOptions          1024 (0x400)
* FileDenyWriteOptions         4831 (0x12df)
  FileDiffMaxSize               10 (0xa)
* FipsMode                    0 (0x0)
* LockdownStatus                0 (0x0)
  LogFileNum                   4 (0x4)
* LogFilePath                  /var/log/mcafee/solidcore
  LogFileSize                  2048 (0x800)
* RTEMode                      1 (0x1)
* RTEModeOnReboot              1 (0x1)
* WorkFlowId                   UPDATE_MODE: AUTO_26
```

* Entries cannot be configured using the command line interface.

Export configuration settings

Export configuration settings to a file to allow deployment of configuration settings to other systems.

Run this command at the command prompt.

```
sadmin config export filename
```

Here `filename` is the target file in which the configuration is to be exported

Import configuration settings

Import configuration settings from a configuration file to deploy the same settings on your system.

Task

1 Switch Application Control to Disabled or Update mode.

2 Restart the system.

3 Import Application Control configuration from a file using this command.

```
sadmin config import [ -a ] filename
```

Use the `-a` argument to append the configuration values. Default behavior is to replace the configuration values.

4 Switch Application Control to Enabled mode and restart the system.

Change configuration parameters

Change the default value to a new value within the permitted range.

Task

1 Type the `sadmin config set NAME=VALUE` command.

NAME signifies the configuration parameter name. VALUE refers to the new value for this configuration parameter.

See this table for default values and the value range allowed for the configurable parameters.

Parameter	Default value	Value range
EventCacheSize	2 (0x2)	> 0 and < MAX_INT32
EventCacheWMHigh	90 (0x5a)	(> 50 and < 100) & (> EventCacheWMLow)
EventCacheWMLow	70 (0x46)	(>20 and < EventCacheWMHigh)
FailSafeConf	0 (0x0)	0 or 1
FipsMode	0 (0x0)	0 or 1
LogFileNum	4 (0x4)	>= 0 and <= MAX_INT
LogFileSize	2048 (0x800)	>= 0 and <= MAX_INT

2 Press **Enter**.

Disable Application Control

Switch to Disabled mode to deactivate the features of Application Control.

Task

1. Type the `sadmin disable` command.
2. Press **Enter**.
3. Restart the system.

7 Troubleshooting

Use this information to identify and troubleshoot issues when you run Application Control.

If the issues are still not resolved after following the troubleshooting steps, collect the required information and contact McAfee Support. See *Collect prerequisite information before contacting McAfee Support* .

Contents

Collecting information before contacting McAfee Support

Collecting specific information before you contact McAfee Support helps McAfee better understand the problem.
If an Application Control-related issue is not resolved, after trying all the suggested troubleshooting steps collect a recent set of GatherInfo logs and details of the system and issue before contacting McAfee Support.

Collect GatherInfo logs

GatherInfo is a utility that collects information related to log files, inventory, product version, and system state, which are needed for troubleshooting.
This utility is shipped with the product and is available in the product installation directory. Collect the most recent set of logs generated using the GatherInfo utility.

The default installation directory depends on the operating system:

- Windows — `<System drive>\Program Files\McAfee\Solidcore\Tools\GatherInfo`
- Linux — `/usr/local/mcafee/solidcore/tools/gatherinfo`

Task

- Type these GatherInfo commands on a Windows or Linux system.
  - For Windows, type `Gatherinfo`

    GatherInfo generates the `gatherinfo.zip` file in the current working directory. The logs in this file are used by McAfee Support to identify issues.
  - For Linux, type `# ./gatherinfo.sh`

    GatherInfo generates the `gatherinfo-<machine_name>-<date>_<time>.tar.gz` file in the present working directory. The logs in this file can be used by McAfee Support to identify issues.

  Optionally, specify these arguments as described in this table.

Windows	Linux	Description
`-h`	`-h` or `--help`	Displays help for using GatherInfo.
`-v`	`-v` or `--version`	Displays version of GatherInfo.
`-q`	`-q`	Gathers logs in quiet mode.
`-x`		Excludes security logs collection.
	`-c` or `--core <core-file>`	Traces previous logs for the specified core file. Specify a core file with this argument to get details of previously generated logs.
	`-n`	Excludes timestamp from the output file name and no timestamp is added.

Collecting system and issue details

Collect the system and issue details before contacting McAfee Support. This helps McAfee Support understand and recreate the issue for diagnostics.

Required detail	Description
Problem description	Describe the problem in detail.
Diagnostics	Collect recent set of log files generated using the GatherInfo utility. See *Collect GatherInfo logs*.
Error messages	Observe and note the error messages. See *Troubleshoot command line interface error messages*.
System image	Create a system image that helps McAfee Support to recreate the problem for diagnostics. See the KnowledgeBase article KB60323 to create a system image.

<table>
<tr><th>Required detail</th><th>Description</th></tr>
<tr><td>Complete memory dump</td><td>Collect complete memory dump in the case of a system crash. Perform these steps to create complete memory dump.
<table>
<tr>
<td>On Linux:
1 Press Alt+SysRq+c on the keyboard.
2 Restart the system.
3 Navigate to the path <code>/var/crash</code>, where the crash dump is generated.
For detailed information on generating crash dump on Linux, refer to the KnowledgeBase article KB66568.</td>
<td>On Windows:
1 Right-click on My Computer.
2 Click Properties.
3 Navigate to Advanced | Startup and Recovery.
4 Click Settings.
You see options to create either a Small, Kernel, or Complete memory dump, and where to save the file (default is: <code>%SystemRoot%\MEMORY.DMP</code>)
5 Select either Kernel memory dump or Complete memory dump, and save your settings.
McAfee Support cannot use a Small memory dump (64 KB) for any purpose.
The next time Windows has a blue screen error, the file <code>%SystemRoot%\MEMORY.DMP</code> will be created.
6 Send <code>%SystemRoot%\MEMORY.DMP</code> to McAfee Support in a zip file.
McAfee recommends using the Microsoft <code>Dumpchk.exe</code> utility before you send the memory dump file for analysis. Dumpchk is a command line utility you can use to verify that a memory dump file has been created properly and is not corrupt. Download <code>Dumpchk.exe</code> from the Microsoft website at: http://support.microsoft.com/kb/156280
For detailed information on generating crash dump on Linux, refer to the KnowledgeBase article KB56023.</td>
</tr>
</table>
</td></tr>
</table>

Startup failure

Troubleshoot startup failure on a Windows system using the safe mode startup option. Safe mode uses a minimum set of device drivers and services to start Windows.

This table describes the issue and symptom.

Category	Description
Issue	Windows system does not start up.
Symptom	Starting a system takes more than the usual time.

When you run the system in safe mode, only the basic files and drivers necessary to run Windows are started. The word **Safe Mode** appears at the corners of the screen. If an existing problem does not reappear when you start in safe mode, you can eliminate the default settings and basic device drivers as possible causes.

Task

1 Restart your computer and press the **F8** key repeatedly on your keyboard.

The **Boot Menu** appears.

2 Select an option when the **Windows Advanced Options** menu appears, then press **Enter**.

3 When the **Boot Menu** appears again and **Safe Mode** appears in the corner, select the installation you want to start, then press **Enter**.

For more details, see the Microsoft knowledge base article KB315222

Application Control does not run in safe mode.

Self-modifying driver issues

Prevent a blue screen failure due to self-modifying drivers.

Category	Description
Symptom	The system stops responding.
Issue	Blue screen failure because of self-modifying drivers.

When loaded on a system, certain drivers can modify their images on system drives. An example for such driver is `clkdrv.sys`, the crypt key driver. If such drivers are added to the whitelist during initial configuration, Application Control does not allow them to load on the systems and self-modification is not allowed. This might lead to a blue screen.

We recommend that the self-modifying drivers must always be authorized to execute on the system. You should authorize the self-modifying drivers by their name. Authorizing such drivers by name does not block the driver from loading on to the system.

Task

1 Identify the self-modifying drivers on your system.

If you cannot identify the self-modifying drivers, proceed to step 3.

2 To authorize the self-modifying drivers by their name, run the `sadmin attr add –a <filename>` command for each identified self-modifying driver and provide a driver name.

3 Restart the system.

If the system is not able to restart normally, proceed to the next step.

4 Collect all the required information contact McAfee Support.

System crashes

Diagnose system-crash issues to recover a system that has Application Control installed on it.

On Windows platform, when the system crashes, a blue screen is observed with a bug check number.

On Linux platform, in case of a system crash, the system might stop responding to any command.

System crash on Windows

Diagnose to recover a system that crashes with blue screen.

This table describes the issue and symptom.

Category	Description
Crash type	System crash (blue screen).
Symptom	System shows a blue screen with a bug check number.

Task

1 Collect this required information.

   a Note the bug check number and all parameters displayed on the screen.

   Make sure that the **Automatically restart** option is deselected under **System properties | Advanced | Startup and Recovery**, while trying to reproduce the bug check. Otherwise, the system restarts automatically and you will not be able note the bug check details, when the bug check occurs.

   b Collect a complete memory dump.

   See the *Collect the system and issue details* section.

2 Start up the system in safe mode.

   a Press the **F8** key while the system is booting up.

   b Select **Safe mode with networking**.

3 Prevent the system from going into a restart loop by deselecting **Automatically restart** on the **StartUp and Recovery** screen.

   For detailed instructions, see this KnowledgeBase article.

   http://support.microsoft.com/kb/307973.

4 To disable Application Control, type the command `scsrvc -d` in the Application Control command line interface, then type the command `sadmin disable` .

5 Restart the system.

6 If the issue is not resolved, collect all required information and contact McAfee Support.

Corrupt whitelist on Windows

Diagnose to recover a system that crashes because of corrupt whitelist.

Category	Description
Crash type	The whitelist for a drive is corrupt.
Symptom	The system shows a blue screen with this error (bug check and parameters). 0xE0100010 (0X00000010, 0X00000000, 0X00000000, 0X00000000)

Task

1 Complete these steps to collect the initial required information.

   a Turn off the system, then turn on the system.

   b Run this command to verify the whitelist status for a corrupt drive.

   `sadmin status`

   The corrupt drive shows the status as `Corrupt` as seen in this output.

```
McAfee Solidifier:           Disabled
McAfee Solidifier on reboot: Disabled

ePO Managed:    No
Local CLI access: Recovered

[fstype]       [status]       [driver status]    [volume]
* NTFS        Corrupt           Unattached           C:\
FAT 32        Solidified        Unattached           E:\
```

   c Run this command at the command prompt.

   `sadmin enable`

   The system displays an error message and Application Control cannot be enabled.

2 Delete the corrupt whitelist for the drive using the following command:

`sadmin clean <drive>`

3 Restart Application Control service using these commands.

- `net stop scsrvc`
- `net start scsrvc`

4 Whitelist the drive again using the `sadmin so <volume name>` command.

5 If the issue is not resolved, collect all the required information and contact McAfee Support.

System crash on Linux

Diagnose to recover a system that stops responding to any command.
This table describes the issue and symptom.

Category	Description
Crash type	Linux system crashes
Symptom	The system might stop responding to any command.

Task

1 Start up the system in the single user mode.

2 Open the Application Control configuration file located at `/etc/mcafee/solidcore/solidcore.conf`.

3 Change the value of parameter **RTEModeOnReboot** to be 0x0.

4 Run the Application Control service manually (`/<install-dir>/mcafee/solidcore/scripts/scsrvc -d`).

   This starts Application Control in the Disabled mode.

5 If required, remove the Application Control package.

6 If the issue is not resolved. collect all the required information and contact McAfee Support.

Active Directory issues

Diagnose to recover a system that is unable to execute logon scripts from AD (via Group Policy). This issue is applicable only on the Windows platform.

This table describes the issue and symptom.

Category	Description
Issue	Not able to execute logon scripts from Active Directory (via Group Policy).
Symptom	The system shows this error message. `McAfee Solidifier prevented unauthorized execution of '\Device\LanmanRedirector \<Domain Controller host_name>\sysvol\<Domain name>\Policies\ {<unique_policy_name>} \User\Scripts\Logon\<script_name>' by process <process_name> (Process Id: PID, User: user_name)`

Task

1 Create a whitelist for the Domain Controller using the `sadmin solidify` command.

There is no need to whitelist any path related to `sysvol` manually because all Application Control supported files are automatically whitelisted on the system.

2 Perform one of these steps.

- Add the `ntfrs.exe` file as an authorized updater to automatically update all `sysvol` volumes in all domain controllers in a local domain using this command.

  `sadmin updaters add -t AD ntfrs.exe`

- Add the `dfsrs.exe` file as an authorized updater to automatically update all `sysvol` volumes among all domain controllers in a local domain using this command.

  `sadmin updaters add -t AD dfsrs.exe`

3 Add the `sysvol` network path as a trusted path for each domain controller in the local domain and for all domain controllers (self and peers) using this command.

`sadmin trusted -i \\<DC_DNS_NAME>\SYSVOL`

If a child domain of the root domain is also present in AD cluster, one trusted rule for each domain controller (self and peers) in the child domain must be added to each domain controller.

For example, if `sales.mycompany.com` is a child domain of `mycompany.com`, it has its own three domain controllers named `cdc1.sales.mycompany.com`, `cdc2.sales.mycompany.com`, and `cdc3.sales.mycompany.com` respectively. In this scenario, three trusted rules need to be added for all three domain controllers of the child domains for proper functioning of Group Policy in the child domain. This is described in these commands.

- `sadmin trusted -i`
  `\\cdc1.sales.mycompany.com\SYSVOL`
- `sadmin trusted -i`
  `\\cdc2.sales.mycompany.com\SYSVOL`
- `sadmin trusted -i`
  `\\cdc3.sales.mycompany.com\SYSVOL`

You can add and execute any file in the trusted path, independent of the Application Control status (Enabled or Disabled). You can execute unauthorized (or non-whitelisted) code only from the trusted network path and not from the local system path. The existing files that are whitelisted on local system are still protected and cannot be modified or deleted from the network path.

4 Run the `sadmin ls` command with the actual file paths to list the status of the whitelisted files.
`sadmin ls C:\WINDOWS\SYSVOL\domain\Policies`

5 If the issue is not resolved, collect all the required information and contact McAfee Support.

Application installation failure

Troubleshoot to successfully install an application that fails to install.

Category	Description
Issue	Application installation fails.
Symptom	The system displays an error message related to installation failure.

Task

1 Verify that the application installs in Update mode.

2 Configure the installer as an updater or trusted configuration. See *Add updaters*.

3 Check if the application installs with pkg-ctrl feature disabled.
If the application installs with pkg-ctrl feature disabled, enable pkg-ctrl and proceed to *step a*.

a Increase the log file size or number of log files. See *Configuring log files*.
If necessary, you can increase the log file size and number of log files both.

b Run the `sadmin loglevel enable pst info` command.

c Re-install the application.
If the issue persists, proceed to the next step.

d Run the `sadmin loglevel disable pst info` command.

4 Check if the application installs with memory-protection feature disabled.

If the application installs with memory-protection feature disabled, enable memory-protection and proceed to *step a*.

a Increase the log file size or number of log files. See *Configuring log files*.

If necessary, you can increase the log file size and number of log files both.

b Run the `sadmin loglevel enable pst info` command.

c Re-install the application.

If the issue persists, proceed to the next step.

d Run the `sadmin loglevel disable pst info` command.

e Collect all the required information and contact McAfee Support.

Application execution failure

Troubleshoot to successfully execute an application that fails to execute.

This table describes the issue and symptom.

Category	Description
Issue	Application execution fails.
Symptom	The application is not allowed to run and the system shows an error message.

Task

1 Check if the application is allowed to run in Update mode.

2 Identify components to be added as updaters or trusted configuration.

3 Configure the identified components as updaters or trusted configuration. See *Add updaters*.

4 Perform these steps to check if the application executes when the memory-protection feature is disabled.

If the application executes when the memory-protection feature is disabled, enable memory-protection and proceed to *step a*.

a Increase the log file size or number of log files. See *Configuring log files*.

If necessary, you can increase the log file size and number of log files both.

b Run the `sadmin loglevel enable pst info` command.

c Run the application again.

If the issue persists, proceed to the next step.

d Run the `sadmin loglevel disable pst info` command.

5 Check if the application executes when the script-auth feature is disabled.

If the application executes when the script-auth feature is disabled, enable the script-auth feature and proceed to *step a*.

a Increase the log file size or number of log files. See *Configuring log files*.

If necessary, you can increase the log file size and number of log files both.

b Run the `sadmin loglevel enable pst info` command.

c Run the application again.

If the issue persists, proceed to the next step.

d Run the `sadmin loglevel disable pst info` command.

e Collect all the required information and contact McAfee Support.

Application performance

Diagnose to recover an application that is running low on performance or stops responding while running.

Category	Description
Issue	Application stops responding while running.
Symptom	Application does not run properly and low on performance.

Perform these steps to diagnose application performance-related issue.

Task

1 Check if the application is running properly in Update mode.

If the issue resolves in Update mode, perform extra steps to diagnose what caused the problem to occur.

a Run the `sadmin features disable checksum` command.

b Collect all required information and contact McAfee Support.

2 Identify components to be added as updaters or trusted configuration.

3 Configure the identified components as updaters or trusted configuration. See *Add updaters*.

4 Check if the application runs properly when the memory-protection feature is disabled.

If the application runs properly when the memory-protection feature is disabled, enable memory-protection and proceed to *step a*.

a Increase the log file size or number of log files. See *Configure log files*.

If necessary, you can increase the log file size and number of log files both.

b Run the `sadmin loglevel enable pst info` command.

c Run the application again.

If the issue persists, proceed to the next step.

d Run the `sadmin loglevel disable pst info` command.

e Collect all required information and contact McAfee Support.

System hang issues

Diagnose to recover a system that hangs (stops responding) while running.

This table describes the issue and symptom.

Category	Description
Issue	System stops responding while running.
Symptom	System does not respond to input from the keyboard or mouse.

Task

1 Turn-off the system.

2 Start up the system in safe mode.

By default, Application Control protection is not available in safe mode.

3 Collect complete memory dump. For detailed instructions on collecting complete memory dump, see the *Collect the system and issue details* section.

4 Issue the `scsrvc -d` command at the command prompt.

5 Open a new Application Control CLI window keeping the previous CLI window still running on the system.

6 Perform these steps on the new CLI window.

a Run the `begin-update` command.

b Restart the system to switch to Update mode.

c Check if the system is running properly in Update mode.

7 Set manual crash dump settings on the system and crash the system. For detailed instructions, see the KnowledgeBase article http://support.microsoft.com/kb/927069.

8 Start up the system in safe mode.

9 Disable Application Control by running the `sadmin disable` command.

10 Issue the `scsrvc -d` command at the command prompt.

11 If the issue persists, collect all the required information and contact McAfee Support.

System performance issues

Diagnose to recover a system that is running low on performance or slows down while running.
This table describes the issue and symptom.

Category	Description
Issue	System slows down while running.
Symptom	System does not work properly and low on performance.

Task

1 Check if the system is running properly in the Update mode.

2 Identify components to be added as updaters or trusted configuration.

3 Configure the identified components as updaters or trusted configuration. See *Add updaters*.

4 Check if the system runs properly when the memory-protection feature is disabled.

If the system runs properly when the memory-protection feature is disabled, enable memory-protection and proceed to *step a*.

a Run the `sadmin loglevel enable pst info` command.

b If the issue persists, run the `sadmin loglevel disable pst info` command.

c Collect all the required information and contact McAfee Support.

Application Control installation failure

Troubleshoot to successfully install Application Control that fails to install on a system.

This table describes the issue and symptom.

Category	Description
Issue	Application Control fails to install on a Windows or Linux system.
Symptom	System shows an error message related to installation failure. Installation can fail in these scenarios. • Installing Application Control on unsupported operating systems. • Blacklisted applications are installed on the system (for Windows). • System does not meet the memory or disk space requirement to install Application Control. • Trying to upgrade, when Application Control upgrade is not supported.

Task

1 Contact McAfee Support, if you don't have another system with the same operating system and Application Control installed on it.

2 Perform these steps on another system that has the same operating system with Application Control installed on it.

a Run `scanalyzer` on the system. For more information, see *Review the run-time environment of a system*.

A report file is generated with a warning if any pre-requisite to install Application Control is missing.

b Ensure that your system meets the requirements as per the report file generated by ScAnalyzer.

If your system does not meet the requirements, you cannot install Application Control.

Updater privileges issues

Diagnose to provide updater privileges to processes that are configured as updaters but do not have updater privileges.

This table describes the issue and symptom.

Category	Description
Issue	Processes that are configured as updaters do not have the updater privileges.
Symptom	Updater processes are not behaving as updaters.

Task

1 Check if the process configured as an updater is having updater privileges.

   a Ensure the process is running.

   b Run the `sadmin xray` command to check the output and process configuration.

   Running the `sadmin xray` command shows if the process added as an updater and running using the updater privileges or not.

   c If the process is not using updater privileges restart the process.

   d Check if the process has updater privileges.

2 If the process still does not have updater privileges, collect all required information and contact McAfee Support.

Events flooding

Diagnose to filter similar type of events or undesired events that are generated and cause flooding of the event list.

This table describes the issue and symptom.

Category	Description
Issue	Event list is flooded with similar type of events.
Symptom	Similar types of events or undesired events are generated.

Task

1 Filter undesired events using advanced exclusion filters. See *Add AEFs*.

2 If the issue is not resolved, collect all required information and contact McAfee Support.

Using error messages

Troubleshoot to resolve an error by always keeping a note of the error message related to that particular error.

You can find the error messages at these locations on your system.

- Console window
- Application Control command line interface
- Popup window from the operating system or an application (for Windows)
- Event viewer
- In the `/var/log/messages` file (for Linux)

These error messages provide valuable insight to people who investigate the problem further. If there are multiple error messages that look similar but are not identical, you should record the details of each error message because that can be helpful to provide the context related to the problem.

Command line interface error messages

Troubleshoot to fix common errors messages that appear on the CLI.

Error message	Solution
When an invalid volume name is used with a command, such as `sadmin solidify` that accepts volume name as a parameter. For example, `sadmin solidify J:` This message is displayed on the CLI: Volume "Volume name:" does not exist.	Use a correct volume name with the command.
When `sadmin <Command Name>` command is run from non-administrative account, it fails to connect to the Application Control service and the following message is displayed on the CLI: Access Denied. Administrator permissions are needed to use the selected options. Use an administrator command prompt to complete these tasks.	Run CLI as an administrator.
If you run the `sadmin solidify` command and specify an improper volume name, such as non-alphabetic characters or colon ':' is missing after the volume name, then the following message is displayed on the CLI: The Path "`C:\Program Files\McAfee\Solidcore\<Volume_Name>`" does not exist or cannot be accessed.	Use a proper supported volume name with the command.
While using the commands, such as `sadmin updaters`, if you specify more arguments than the supported number of arguments for that particular command. For example `sadmin updaters add -u <user name> -p <binary name>`. This message is displayed on the CLI: Too many arguments. Please type sadmin help <Command Name> for help.	Use the supported number of arguments with the commands.
While using the commands, such as `sadmin updaters`, if you specify arguments but don't specify the value for arguments. For example, `sadmin updaters add -u` This message is displayed on the CLI: Not enough arguments. Please type `sadmin help <Command Name>` for help.	Specify the value such as user name, file name or tag name for all the arguments you use with the command.
While using the commands, such as `sadmin write-protect`, if you use an invalid argument. For example, `sadmin write-protect -k` This message is displayed on the CLI: Invalid option "<Argument Name>". Please type `sadmin help <Command Name>` for help.	Use the correct and supported argument with the command.
If you run the `sadmin solidify` command in a CLI and open another CLI and run any other command, such as `sadmin status` then the following message is displayed on the CLI: Another CLI is already connected.	Do not run commands on another CLI when whitelisting is in progress.

Legitimate failures and error messages

Certain legitimate failures can occur when Application Control is running in Enabled mode on a system. Error messages corresponding to the legitimate failures are also generated. However, such error messages are legitimate and reflect that Application Control is preventing unauthorized operations.

For example, Application Control ensures that a component, such as binary, script, or installer package, can execute only if it is present in the whitelist. If a component is present in the whitelist, it is allowed to execute normally. Otherwise, Application Control prevents its execution and shows a corresponding error message on the system. Events are generated in all such scenarios.

Error messages generated for binary and script files

When attempts are made to execute the binary or script files not present in the whitelist, corresponding error messages are generated. Review the error messages when such attempts are made to understand the errors.

This table describes error messages displayed when you attempt to execute the binary and script files not present in the whitelist.

Attempt	Description
Attempt to execute an `.exe` file not present in the whitelist.	When an attempt is made to execute a program not present in the whitelist, such as `putty.exe` from a supported volume, the operation fails and a pop-up window displays this message. `Windows cannot access the specified device, path, or file. You may not have the appropriate permissions to access the item.` If you execute `putty.exe` from the command prompt, this message appears. Access is denied.
Attempt to execute a `.vbs` script file not present in the whitelist.	If you double-click a `.vbs` script file not present in the whitelist, its execution fails and the Windows Script Host displays a pop-up window that shows: `Loading script "C:\shared\AUTH\AUTH.vbs" failed (Access is denied)` An event is also generated. For more information on events, see *Review changes using events*.
Attempt to execute an ELF binary file not present in the whitelist. (Linux).	When an attempt is made to execute an ELF binary file, such as foo2bar2 that is not present in the whitelist, the operation fails. If foo2bar2 file is executed from the command prompt, this message is displayed. `Permission denied.`
Attempt to execute a #! (hash-bang) script not present in the whitelist. (Linux).	When you try to execute a #! script not present in the whitelist, its execution fails and this message is displayed. `bad interpreter: Permission denied`

Error messages generated for installer packages

When attempts are made to execute the installer packages not present in the whitelist, corresponding error messages are generated. Review the error messages when such attempts are made to understand the errors.

This table describes error messages displayed when you run installer packages that are not present in the whitelist.

Attempt	Description
Attempt to run an MSI-based installer.	When an attempt is made to install an MSI-based installer, such as `Ica32Pkg.msi`, the operation fails. This error message is displayed in a pop-up window. The system administrator has set policies to prevent the installation. An event is generated that displays Application Control has prevented the execution of unauthorized code. The event appears if package-control feature is enabled.
Attempt to uninstall an MSI-based installer.	When an attempt is made to uninstall a MSI-based package, such as `Ica32Pkg.msi`, the operation fails. This error message is displayed in a pop-up window. This installation is forbidden by system policy. Contact your system administrator. An event is generated that displays Application Control has prevented the execution of unauthorized code. In some cases, you might not be able to uninstall an application (that was installed using an MSI-based installer) using the Add or Remove Programs feature. To remove such applications, execute <installer>.msi file to uninstall the application.
Attempt to install or uninstall Windows optional components.	When an attempt is made to install or uninstall Windows optional components from Add or Remove Programs, the operation fails and an event is generated. The event shows that Application Control has prevented the execution of unauthorized code. The event appears if package-control feature is enabled.
Attempt to run an INF-based installer.	When an attempt is made to install an INF-based installer, such as `mmdriver.inf` by right-clicking on the installer, the operation fails and an event is generated. The event shows that Application Control has prevented the execution of unauthorized code. The event appears if package-control feature is enabled.

Error messages generated while tampering the whitelisted components

During normal usage, whitelisted program files cannot be modified, renamed, or deleted, even with administrator rights. When attempts are made to tamper the whitelisted components, corresponding

error messages are generated. Review the error messages when such attempts are made to understand the errors.

Any attempt to modify a whitelisted file is prevented and an access denied error is generated. Also, Application Control does not allow you to make changes to its registry files and protects its registry files from any changes being made to them in Enabled mode.

This table describes the error messages that are displayed when an attempt is made to tamper with whitelisted files and registry keys.

Attempt	Description

Attempt to rename a file present in the whitelist.	The rename operation fails and a pop-up window displays the following message:
	`Cannot rename <filename>: Access is denied.`
	`Make sure the disk is not full or write-protected and that the file is not currently in use.`
	An event is also generated that shows that Application Control has prevented the rename operation. Also, an error message is displayed in the Windows Event Viewer.

Attempt to move a file present in the whitelist.	**On Windows platform**:
	The move operation fails and a pop-up window displays the following message:
	`Cannot rename <filename>: Access is denied.`
	`Make sure the disk is not full or write-protected and that the file is not currently in use.`
	An event is also generated that shows that Application Control has prevented the move operation. Also, an error message is displayed in the Windows Event Viewer.
	On Linux platform:
	The move operation fails and this message is displayed.
	`mv: cannot move 'filename' to 'filename1': Permission denied.`

Attempt to delete a file present in the whitelist.	**On Windows platform:**
	The delete operation fails and a pop-up window displays this message.
	`Cannot delete <filename>: Access is denied. Make sure the disk is not full or write-protected and that the file is not currently in use.`
	An event is also generated that shows that Application Control has prevented the delete operation. Also, an error message is displayed in the Windows Event Viewer.
	On Linux platform:
	The remove operation fails and this message is displayed.
	`rm: cannot remove 'filename': Permission denied.`

Attempt	Description

Attempt to overwrite a file present in the whitelist.	**On Windows platform:**
	The overwrite operation fails and a pop-up window displays this message.
	`Cannot copy <filename>: Access is denied.`
	`Make sure the disk is not full or write-protected and that the file is not currently in use.`
	An event is also generated that shows that Application Control has prevented the overwrite operation. Also, an error message is displayed in the Windows Event Viewer.
	On Linux platform:
	Overwrite operation fails and this message is displayed.
	`cp: cannot create regular file 'filename': Permission denied.`

Attempt to add alternate stream for a file present in the whitelist.	The operation fails and this message is displayed on the CLI. `Access is denied.` An event is also generated.
Attempt to delete alternate stream for a file present in the whitelist.	The operation fails but no message is displayed. However, an event is generated in the Windows Event Viewer.
Attempt to rename the Application Control specific registry keys.	The operation fails and a pop-up displays this message. `The Registry Editor cannot rename 'registry key name'. Error while renaming value.`
Attempt to delete a registry key.	The operation fails and a pop-up displays this message. `Cannot delete Parameters: Error while deleting key.`

Bypass rules for files and scripts

Define specific rules to bypass files and scripts from the write-protection and script-auth features using the process context file operations technique.

Some applications (as part of their day-to-day processing) run code in an atypical way and hence are prevented from running. To allow such applications to run, define appropriate bypass rules. Bypassing a file should be the last-resort to allow an application to run and should be used judiciously.

Add bypass rules for files and scripts

Add bypass rules for files and scripts to allow the non-whitelisted scripts to execute on the system and bypass the script-auth feature. Also, adding this rule bypasses the deny-write feature and a whitelisted file added to this rule is not be write-protected. However, this rule doesn't bypass the deny-exec feature.

Add bypass rules using the `sadmin attr add` command and specify the required arguments.

The command syntax is `sadmin attr add -o <parent_file> -p <file>`.

Task

1 Run the `sadmin attr add -o <parent_file> -p <file>` command.

Specify a file name with the command to bypass the file from process context file operations technique.

Optionally, use the `-o` argument with this command to specify the DLL module name for a specified process or a file name. On the Linux platform, use this argument to specify the parent program.

2 Press **Enter**.

3 Optionally, run the `sadmin attr list -p` command.

View the list of all the files that are bypassed using this command.

The bypass rule is added.

Remove bypass rules for files and scripts

Remove bypass rules for files and scripts to restrict the non-whitelisted scripts to execute on the system. Also, when this rule is removed, deny-write and script-auth features are effective again.

There are two methods to remove bypass rules.

- Remove bypass rules from a specified file or script.

  Bypass rules are removed only from the specified file or script.

  The command syntax is `sadmin attr remove -p <file>`.

- Flush all bypass rules.

  Removes all bypass rules added for files and scripts.

  The command syntax is `sadmin attr flush -p`.

Task

1 Remove bypass rules from a specific file or script.

a Type the `sadmin attr remove -p <file>` command.

Specify the file or script name.

b Press **Enter**.

Bypass rules for the specified file or script are removed.

2 Flush all bypass rules.

a Type the `sadmin attr flush -p` command.

b Press **Enter**.

All bypass rules added for files and scripts are removed.

Skip rules for path components

Define skip rules on the Windows platform to skip specific path components from multiple Application Control features and the Windows Alternate Data Stream (ADS) feature. When you skip a path component, all files in that path are skipped.

When the product is successfully deployed on a system and running in Enabled mode, all directories and subdirectories present on the system are added to the whitelist and protected by Application Control features.

There can be certain files present in a path that require frequent modification or deletion operations but because of the applied protection by Application Control features, these operations are not allowed. You might want to perform operations on the files that are protected by Application Control features. In such cases, skip specific path components from features to allow operations that are blocked by those features. Also, you can skip specific path components from the Windows ADS feature.

Specify a path component to define skip rules and not the absolute or relative path. Application Control searches the specified path component across all volumes and applies skip rules on that particular path component present on a system. This applies skip rules on all files in that path component. For example, to define skip rules on a path `C:\WINDOWS\Debug\UserMode`, do not specify the absolute or relative path. Specify only the path component `\UserMode`. Application Control applies the skip rules on this path component across all volumes.

When you skip a path component from a feature, the path component is skipped from that feature only and the protection applied by that feature on the path component is removed. However, the path component is not removed from the whitelist. You can also define skip rules to skip path components from the whitelist. This removes path components from the whitelist.

Add skip rules for path components

Add skip rules to skip path components from Application Control features and Windows ADS feature.

Add skip rules using the `sadmin skiplist add` command and specify the required arguments.

The command syntax is `sadmin skiplist add [-a | -d | -f | -i | -r | -s | -v] PATH`.

Skip path components from Windows ADS feature

Skip path components from Windows ADS feature to remove the protection applied on the path component by the feature.

Task

1 Type the `sadmin skiplist add -a <path component>` command.

Specify a path component with this command.

2 Press **Enter**.

The skip rule is added.

Skip path components from write protection

Skip path components from the write-protection feature to remove write protection applied to all the files in that path. Also, write denied event is not observed for such paths.

Task

1 Type the `sadmin skiplist add -d <path component>` command.

Specify a path component with the command.

User mode paths and paths with volume name do not work with this command. Text added with this command is treated as complete component. For example, text should start with a slash (`/`) and end with a slash (`\`) , dot (`.`), or null character.

2 Press **Enter**.

The skip rule is added.

Skip path components from file operations

Skip path components from file operations and the script-auth feature. When you skip path components from file operations, file operations, such as creation, modification, and deletion are not protected under the write-protection feature. However, link and rename operations for the destination path are still protected by the write protection feature.

Task

1 Type the `sadmin skiplist add -f <path component>` command.

Specify a path component with the command. User mode paths and paths with volume name do not work with this command.

Text added with this command is treated as substring in a path. No events are raised and the whitelist is not updated for the skipped path components. Also, script execution control does not work for paths added with this command.

2 Press **Enter**.

The skip rule is added.

Skip path components from file operations and deny-exec

Skip path components from file operations using the ignore path list. This works similar to the `sadmin add -f` command. Also, on the Windows 64-bit platform, binaries of pe32 type are also skipped from the deny-exec feature using this skip rule. Restarting the system is necessary to enable this skip rule.

Task

1 Type the `sadmin config show` command.

2 Press **Enter**.

3 Type the `sadmin config set customerconfig=<new value>` command.

Change the value of `CustomerConfig` registry key under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swin\Parameters` to current default value or 0x80 value. You can use the Windows calculator in scientific mode to perform the logical OR operation between the current default value and 0x80 (0x80[hexadecimal] and 128 [decimal]) to get the new value that should be specified. However, the new value is effective only after system restart.

4 Press **Enter**.

5 Restart the system.

6 Type the `sadmin skiplist add -i <path component>` command.

Specify a path component with the command.

User mode paths and paths with volume name do not work with this command.

When the path components are specified on Windows 64-bit platforms, even the deny-exec feature is skipped.

7 Press **Enter**.

The skip rule is added.

Skip registry path components from write protection

Skip registry path components from the write protection feature for registry to remove the write protection applied on the registry paths.

Task

1 Type the `sadmin skiplist add -r <path component>` command.

Specify a registry path component with the command.

Text added with this command is treated as complete component. For example, text should start with a slash (`/`) and end with a slash (`\`) , dot (`.`), or null character.

2 Press **Enter**.

The skip rule is added.

Skip path components from the whitelist

Skip path components from the whitelist to remove files in the specified path components from the whitelist. After the files are removed from the whitelist, modification and deletion operations are allowed on such files. However, the files are not allowed to execute.

Task

1 Type the `sadmin skiplist add -s <path component>` command.

Specify a path component with the command. Network path names cannot be specified with this command. When a path component is specified with this command, files present in the whitelist under that path and subdirectories are removed from the whitelist. However, files generated or modified on such paths and subdirectories are added to the whitelist in unso state only regardless of the Application Control mode (Enabled mode or Update mode) or using an updater. When files are added to the whitelist in unso state, modifications to the files are allowed but execution is denied for all such files.

Volume relative rules can also be specified using `*\<vol_rel_name>`.

An asterisk (`*`) can be used to represent any one component in the path. On addition of rules with asterisk (`*`), files in that path are not removed from the whitelist, but files generated in Enabled mode, Update mode, or using an updater are added to the whitelist in unso state only. Because files are not removed from the whitelist while adding rules containing asterisk (`*`), write protection is observed for whitelisted files on such paths.

2 Press **Enter**.

The skip rule is added.

Skip volumes from Application Control

Skip volume names from attaching to Application Control. You can additionally specify the file system, such as NTFS or FAT. When you specify a volume name with this argument, Application Control is not

attached to that volume. Script-auth and deny-exec features are also not effective on the specified volume. Components in that volume are allowed to execute on the system.

Task

1 Type the `sadmin skiplist add -v <path component>` command.

Specify a path component with the command.

You can specify a path component using user mode volume names, such as `C:` and `D:`. Also, device names, such as `\device\harddiskvolume1` and file systems, such as NTFS and FAT can also be specified.

If any of the criteria for specifying the path component is met, Application Control does not attach to that volume. Hence, script-auth and deny-exec features will not work for such volumes. A restart is required for the rule to work if the drive is already attached.

2 Press **Enter**.

The skip rule is added.

List skip rules for path components

You can view the list of all the skip rules added for the path components to skip from features.

The command syntax is `sadmin skiplist list [-a | -d | -f | -i | -r | -s | -v]`.

Run this command at the command prompt.

```
skiplist list
```

You must specify arguments to view the list of skip rules applied using those arguments.

Remove skip rules for path components

Remove skip rules applied to specific path component to again protect the path component by Application Control and Windows features.

There are two methods to remove skip rules.

- Remove skip rules from a specified path component.

  Skip rules are removed only from the specified path component.

  The command syntax is `sadmin skiplist remove [-a | -d | -f | -i | -r | -s | -v] PATH`.

- Flush all skip rules.

  Removes all skip rules for the specified argument.

  The command syntax is `sadmin skiplist flush [-a | -d | -f | -i | -r | -s | -v]`.

Task

1 Remove skip rules from a specified path component.

a Type the `sadmin skiplist remove` command and specify the path component and the argument for which you want to remove the skip rules.

b Press **Enter**.

Skip rules from the specified path components are removed.

2 Flush all skip rules.

   a Type the `sadmin skiplist flush` command and specify the argument for which you want to remove the skip rules.

   b Press **Enter**.

   All skip rules for the specified argument are removed.

A FAQs

Here are answers to frequently asked questions.

How can I switch Application Control from standalone to McAfee ePO managed mode?

See these articles.

- Windows platform: KB69408
- Linux platform: KB74077

I have a McAfee Change Control product and I have upgraded it to Application Control. How does this affect my licenses?

When you add the Application Control license to upgrade from McAfee Change Control, the common features are set to the default status for Application Control.

What is the difference between log messages and events?

Events are generated for all changes made to a protected system and are stored in the event sinks. Log messages are generated for all actions and errors related to the product, stored in log files. For information, see *Review changes using events* and *Configuring log files*.

What are the supported operating systems for Application Control 6.1.0 release?

These are the supported operating systems for the Application Control 6.1.0 release.

- Windows Server 2003
- Windows XP
- Windows Server 2008
- Windows Vista
- Windows 7
- Linux

B Standalone features vs. managed features

All Application Control features supported in the standalone configuration are also supported in the McAfee ePO managed configuration. Some features are available only in the managed configuration.

Feature	Standalone configuration	Managed configuration
activex	Supported	Supported
checksum	Supported	Supported
deny-read	Supported	Supported
deny-write	Supported	Supported
discover-updaters	Supported	Supported
end-user notification	Not supported	Supported
integrity	Supported	Supported
mp	Supported	Supported
mp-casp	Supported	Supported
mp-vasr	Supported	Supported
network-tracking	Supported	Supported
pkg-ctrl	Supported	Supported
script-auth	Supported	Supported
self-approval	Not supported	Supported

C Application Control event list

Application Control specific events with the name, event ID, severity, and the description are described in this table.

Application Control Event ID	Event name	Severity	Description
19	PROCESS_TERMINATED	Major	Application Control prevented an attempt to hijack the process <string> (Process Id: <string>, User: <string>), by illegally calling the API '<string>'. The process was terminated.
20	WRITE_DENIED	Major	Application Control prevented an attempt to modify file '<string>' by process <string> (Process Id: <string>, User: <string>).
21	EXECUTION_DENIED	Major	Application Control prevented unauthorized execution of '<string>' by process <string> (Process Id: <string>, User: <string>).
29	PROCESS_TERMINATED_UNAUTH_SYSCALL	Major	Application Control prevented process <string>, being run by <string>, from making unauthorized syscall %d (return address %d). The process was terminated.
30	PROCESS_TERMINATED_UNAUTH_API	Major	Application Control prevented process <string>, being run by <string>, from making unauthorized access to API <string> (return address <string>). The process was terminated
49	REG_VALUE_WRITE_DENIED	Major	Application Control prevented an attempt to modify Registry key '<string>' with value '<string>' by process <string> (Process Id: <string>, User: <string>).
50	REG_KEY_WRITE_DENIED	Major	Application Control prevented an attempt to modify Registry key '<string>' by process <string> (Process Id: <string>, User: <string>)

Application Control Event ID	Event name	Severity	Description
51	REG_KEY_CREATED_UPDATE	Info	Application Control detected creation of registry key '<string>' by program <string> (User: <string>, Workflow Id: <string>).
52	REG_KEY_DELETED_UPDATE	Info	Application Control detected deletion of registry key '<string>' by program <string> (User: <string>, Workflow Id: <string>).
54	REG_VALUE_DELETED_UPDATE	Info	Application Control detected deletion of registry value '<string>' under key '<string>' by program <string> (User: <string>, Workflow Id: <string>).
57	OWNER_MODIFIED_UPDATE	Info	Application Control detected modification to OWNER of '<string>' by program <string> (User: <string>, Workflow Id: <string>).
61	PROCESS_HIJACKED	Major	Application Control detected an attempt to exploit process <string> from address <string>.
62	INVENTORY_CORRUPT	Critical	Application Control detected that its internal inventory for the volume <string> is corrupt. To rectify, delete the inventory and solidify the volume again.
75	FILE_CREATED_UPDATE	Info	Application Control detected creation of '<string>' by program <string> (User: <string>, Original User: <string>, Workflow Id: <string>).
76	FILE_DELETED_UPDATE	Info	Application Control detected deletion of '<string>' by program <string> (User: <string>, Original User: <string>, Workflow Id: <string>).
77	FILE_MODIFIED_UPDATE	Info	Application Control detected modification of '<string>' by program <string> (User: <string>, Original User: <string>, Workflow Id: <string>)
79	FILE_RENAMED_UPDATE	Info	Application Control detected renaming of '<string>' to '<string>' by program <string> (User: <string>, Original User: <string>, Workflow Id: <string>).

Application Control Event ID	Event name	Severity	Description
80	FILE_SOLIDIFIED	Info	<string>' was solidified which was created by program <string>(User: <string>, Workflow Id: <string>).
82	FILE_UNSOLIDIFIED	Info	<string>' was unsolidified which was deleted by program <string>(User: <string>, Workflow Id: <string>).
89	READ_DENIED	Major	Application Control prevented an attempt to read file '<string>' by process <string> (Process Id: <string>, User: <string>).
96	PKG_MODIFICATION_PREVENTED	Critical	Application Control prevented package modification by '<string>' by user: '<string>'.
97	PKG_MODIFICATION_ALLOWED_UPDATE	Info	Application Control allowed package modification by '<string>' by user: '<string>'. (Workflow Id: <string>)
98	PKG_MODIFICATION_PREVENTED_2	Critical	Application Control prevented package modification by '<string>' by user: '<string>'.
99	NX_VIOLATION_DETECTED	Critical	Application Control prevented an attempt to hijack the process '<string>' (Process Id: '<string>', User: '<string>'), by executing code from an address outside of code pages region. Faulting address '<string>'. The process was terminated.
101	REG_VALUE_MODIFIED_UPDATE	Info	Application Control detected modification to registry value '<string>' of type '<string>' under key '<string>' by program '<string>' (User: <string>, Workflow Id: <string>), with data: <string>
103	FILE_READ_UPDATE	Info	Application Control detected read for '<string>' by program <string> (User: <string>, Original User: <string>, Workflow Id: <string>)
124	INITIAL_SCAN_TASK_COMPLETED	Info	Application Control Initial Scan task is complete and Application Control is enforced on the system now.
126	ACTX_ALLOW_INSTALL	Info	Application Control allowed installation of ActiveX <string> Workflow Id: <string> by user <string>

Application Control Event ID	Event name	Severity	Description
127	ACTX_INSTALL_PREVENTED	Major	Application Control prevented installation of ActiveX <string> Workflow Id: <string> by user <string>
129	VASR_VIOLATION_DETECTED	Critical	Application Control prevented an attempt to hijack the process '<string>' (Process Id: '<string>', User: '<string>'), by executing code from non-relocatable dll '<string>'. Faulting address '<string>'. Target address '<string>'

Index