Configure and Identify Secure Endpoint Exclusions

Contents

- Introduction
- Disclaimer
- Overview
- What are Exclusions
  - Cisco-Maintained Exclusions
  - Custom Exclusions
- Types of Exclusions
  - Process Exclusions
    - MacOS and Linux
    - Windows
  - Threat Exclusions
  - Path Exclusions
    - Partial Path Matches (Windows-only)
  - File Extension Exclusions
  - Wildcard Exclusions
    - Windows
  - Executable Exclusions (Windows-only)
  - IOC Exclusions (Windows-only)
- CSIDL and KNOWNFOLDERID (Windows-only)
- Prepare Connector for Exclusion Tuning
- Identify Exclusions
  - MacOS and Linux
    - Creating Process Exclusions
    - Creating Path, File Extension, and Wildcard Exclusions
    - Behavioral Protection Engine
  - Windows
- Creating Exclusion Rules in the Secure Endpoint Console
- Best Practices
  - Not Recommended Exclusions
- Related Information

Introduction

This document describes exclusions, how to identify exclusions, and the best practices for creating exclusions on the Cisco Secure Endpoint.

Disclaimer

The information in this document is based on Windows, Linux and macOS operating systems.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Overview

After reading this document, you must understand:

- What an exclusion is and the different types of exclusions available for Cisco Secure Endpoint.
- How to prepare your connector for exclusion tuning.
- How to identify potentially strong exclusions.
- How to create new exclusions in the Cisco Secure Endpoint Console.
- What the best practices are for creating exclusions.

What are Exclusions

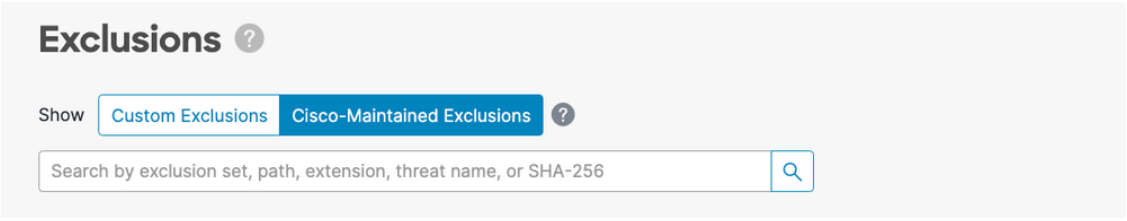
An exclusion set is a list of directories, file extensions, file paths, processes, threat names, applications, or indicators of compromise that you do not want the connector to scan or convict. Exclusions need to be carefully crafted to ensure a balance of performance and security on a machine when endpoint protection such as Secure Endpoint is enabled. This article describes exclusions for Secure Endpoint Cloud, TETRA, SPP, and MAP.

Every environment is unique as well as the entity which controls it, varying from stringent to open policies. As such, exclusions must be uniquely tailored to each situation.

Exclusions can be categorized in two ways, Cisco-Maintained Exclusions and Custom Exclusions.

Cisco-Maintained Exclusions

Cisco-Maintained Exclusions are exclusions that have been created based on research and have undergone rigorous testing on commonly used operating systems, programs, and other security software. These exclusions can be displayed by selecting **Cisco-Maintained Exclusions** in the Secure Endpoint Console on the **Exclusions** page:

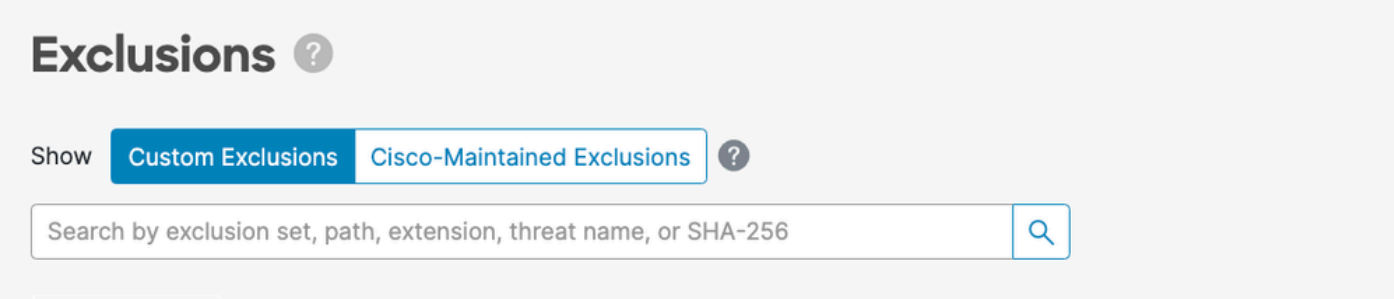


Cisco monitors recommended exclusion lists published by Anti-Virus (AV) vendors and updates the Cisco-Maintained Exclusions to include the recommended exclusions.

Note: Some AV vendors do not publish their recommended exclusions. In this case, you need to reach out to the AV vendor to request a list of recommended exclusions and then open a Support Case to get the Cisco-Maintained Exclusions updated.

Custom Exclusions

Custom Exclusions are exclusions that have been created by a user for a custom use case on an endpoint. These exclusions can be displayed by selecting **Custom Exclusions** in the Secure Endpoint Console on the **Exclusions** page.



Types of Exclusions

Process Exclusions

Process exclusions allow admins to exclude processes from supported engines. The engines that support Process exclusions on each platform are outlined in this table:

Operating System	Engine			
	File Scan	System Process Protection	Malicious Activity Protection	Behavioral Protection
Windows	✓	✓	✓	✓
Linux	✓	✗	✗	✓
macOS	✓	✗	✗	✓

MacOS and Linux

You must provide an absolute path when creating a Process exclusion, you can also provide an optional user. If you specify both a path and user then both conditions must be met for the process to be excluded. If you do not specify a user, then the process exclusion applies to all users.

Note: On macOS and Linux, Process exclusions apply to all engines.

Process Wildcards:

Secure Endpoint Linux and macOS connectors support using a wildcard within the Process exclusion. This allows for broader coverage with less exclusions but can also be dangerous if too much is left undefined.

Caution: You must only use the wildcard to cover the minimum number of characters required to provide the needed exclusion.

Use of Process Wildcard for macOS and Linux:

- The wildcard is represented using a single asterisk character (*).
- The wildcard can be used in place of a single character or a full directory.

- Placing the wildcard at the beginning of the path is considered invalid.
- The wildcard works between two defined characters, slashes or alphanumerics.

Examples:

Exclusion	Expected Result
/Library/Java/JavaVirtualMachines/*/java	Excludes java within all subfolders of JavaVirtualMachines
/Library/Jibber/j*bber	Excludes the process for jabber, jibber, jobber, etc

Windows

You can provide an absolute path and/or a SHA-256 of the process executable when creating a Process exclusion. If you specify both a path and SHA-256 then both conditions must be met for the process to be excluded.

On Windows, you can also use the CSIDL or KNOWNFOLDERID within the path to create Process exclusions.



Caution: Child processes created by an excluded process are not excluded default. To exclude additional processes when creating a Process exclusion, select **Apply to Child Processes**.

Limitations:

- If the file size of the process is greater than the maximum scan file size set in your policy, then the SHA-256 of the process is not computed and the exclusion does not work. Use a path-based Process exclusion for files larger than the maximum scan file size.
- The Windows connector imposes a limit of 500 process exclusions across all process exclusion types.
    - Process exclusions are only honored up to the limit, starting from the top of the process exclusions list in policy.xml.
    - Every Windows policy has a Process exclusion for sfc.exe, which counts against the process exclusions limit:

```
<item>3|0||CSIDL_Secure Endpoint_VERSION\sfc.exe|48|</item>
```



Note: On Windows, Process exclusions are applied per engine. If the same exclusion must be applied to multiple engines, then the Process exclusion must be duplicated in this case for each applicable engine.

Process Wildcards:

Secure Endpoint Windows connectors support using a wildcard within the Process exclusion. This allows for broader coverage with less exclusions but can also be dangerous if too much is left undefined.



Caution: You must only use the wildcard to cover the minimum number of characters required to provide the needed exclusion.

Use of Process Wildcard for Windows:

- The wildcard is represented using a single asterisk character (*) or a double asterisk (**)
- Single asterisk wildcard (*):
  - The wildcard can be used in place of a single character or a full directory.
  - Placing the wildcard at the beginning of the path is considered invalid.
  - The wildcard works between two defined characters, slashes or alphanumerics.
  - Placing the wildcard at the end of a path excludes all processes in that directory but not subdirectories.
- Double asterisk wildcard (**):
  - Can only be placed at the end of a path.
  - Placing the wildcard at the end of a path excludes all processes in that directory and all processes in subdirectories.
  - This allows for a much larger exclusion set with minimal input but also leaves a very large security hole for visibility. Use this feature with extreme caution.

Examples:

Exclusion	Expected Result
C:\Windows*\Tiworker.exe	Excludes all Tiworker.exe processes found in the subdirectories of Windows
C:\Windows\P*t.exe	Excludes Pot.exe, Pat.exe, P1t.exe, etc
C:\Windows*chickens.exe	Excludes all processes in Windows directory ending in chickens.exe
C:*	Excludes all processes in the C: drive but not in the subdirectories
C:**	Excludes every process on the C: drive

Threat Exclusions

Threat exclusions let you exclude a particular threat name from triggering events. You must only ever use a Threat exclusion if you are certain that the events are the result of a false-positive detection. In this case, use the exact threat name from the event as your Threat exclusion. Be aware that if you use this type of exclusion even a true-positive detection of the threat name is not detected, quarantined, or generate an event.



Note: Threat exclusions are case insensitive. Example:W32.Zombies.NotAVirus andw32.zombies.notavirus both match the same threat name.



Warning: Do not exclude threats unless a thorough investigation has confirmed the threat name to be false-positive. Threats excluded no longer populate the events tab for review and audit.

Path Exclusions

Path exclusions are the most frequently used, as application conflicts typically involve the exclusion of a directory. You can create a path exclusion using an absolute path. On Windows, you can also use the CSIDIL or KNOWNFOLDERID to create path exclusions.

For example, to exclude an AV application in the Program Files directory on Windows, the exclusion path could be any of these paths:

```
C:\Program Files\MyAntivirusAppDirectory
CSIDL_PROGRAM_FILES\MyAntivirusAppDirectory
FOLDERID_ProgramFiles\MyAntivirusAppDirectory
```



Note: Path Exclusions are recursive and exclude all sub-directories as well.

Partial Path Matches (Windows-only)

If a trailing slash is not provided in the Path exclusion then the Windows connector does a partial match on paths.



Note: Mac and Linux do not support partial path matches.

For example, if you apply the these path exclusions on Windows:

```
C:\Program Files
C:\test
```

Then all of these paths are excluded:

```
C:\Program Files
C:\Program Files (x86)
C:\test
C:\test123
```

Changing the exclusion from C:\test to C:\test\ prevents C:\test123 from being excluded.

File Extension Exclusions

File Extension exclusions allow the exclusion of all files with a certain extension.

Key points:

- Expected input in the Secure Endpoint Console is .extension
- The Secure Endpoint Console automatically prepends a period to the file extension if none was added.
- Extensions are case insensitive.

For example, in order to exclude all Microsoft Access database files, you can create this exclusion:

```
.MDB
```



Note: Standard file extension exclusions are available in the default list, it is not recommended to delete these exclusions, doing so can cause performance changes on your endpoint.

Wildcard Exclusions

Wildcard exclusions are the same as Path or File Extension exclusions except that you can use an asterisk character (*) to represent a wildcard within the path or extension.

For example, if you wanted to exclude your virtual machines on macOS from being scanned you can enter this Path exclusion:

```
/Users/johndoe/Documents/Virtual Machines/
```

However, this exclusion only works for one user, so instead replace the username in the path with an asterisk and create a Wildcard exclusion instead to exclude this directory for all users:

```
/Users/*/Documents/Virtual Machines/
```



Caution: Wildcard exclusions do not stop at path separators, this can lead to unintended exclusions. For example, C:*\test excludesC:\sample\test as well as C:\1\test** orC:\sample\test123.



Warning: Beginning an exclusion with an asterisk character can cause major performance issues. Remove or change all exclusions that begin with an asterisk character to mitigate CPU impact.

Windows

When creating Wildcard exclusions on Windows, there is an option to **Apply to all drive letters**. Selecting

this option applies the Wildcard exclusion to all mounted drives.



If you were to manually craft the same exclusion you would need to prepend it with ^[A-Za-z], for example:

```
^[A-Za-z]\testpath
```

In both examples, the C:\testpath and D:\testpath are excluded.

The Secure Endpoint Console automatically generates the ^[A-Za-z] when **Apply to all drive letters** is selected for wildcard exclusions.

Executable Exclusions (Windows-only)

Executable exclusions only apply to Windows connectors with Exploit Prevention enabled. An Executable exclusion excludes certain executables from being protected by Exploit Prevention. You must only exclude an executable from Exploit Prevention if you are experiencing problems or performance issues.

You can check the list of Protected Processes and exclude any from protection by specifying its executable name in the application exclusion field. Executable exclusions must match the executable name exactly in the format name.exe. Wildcards are not supported.

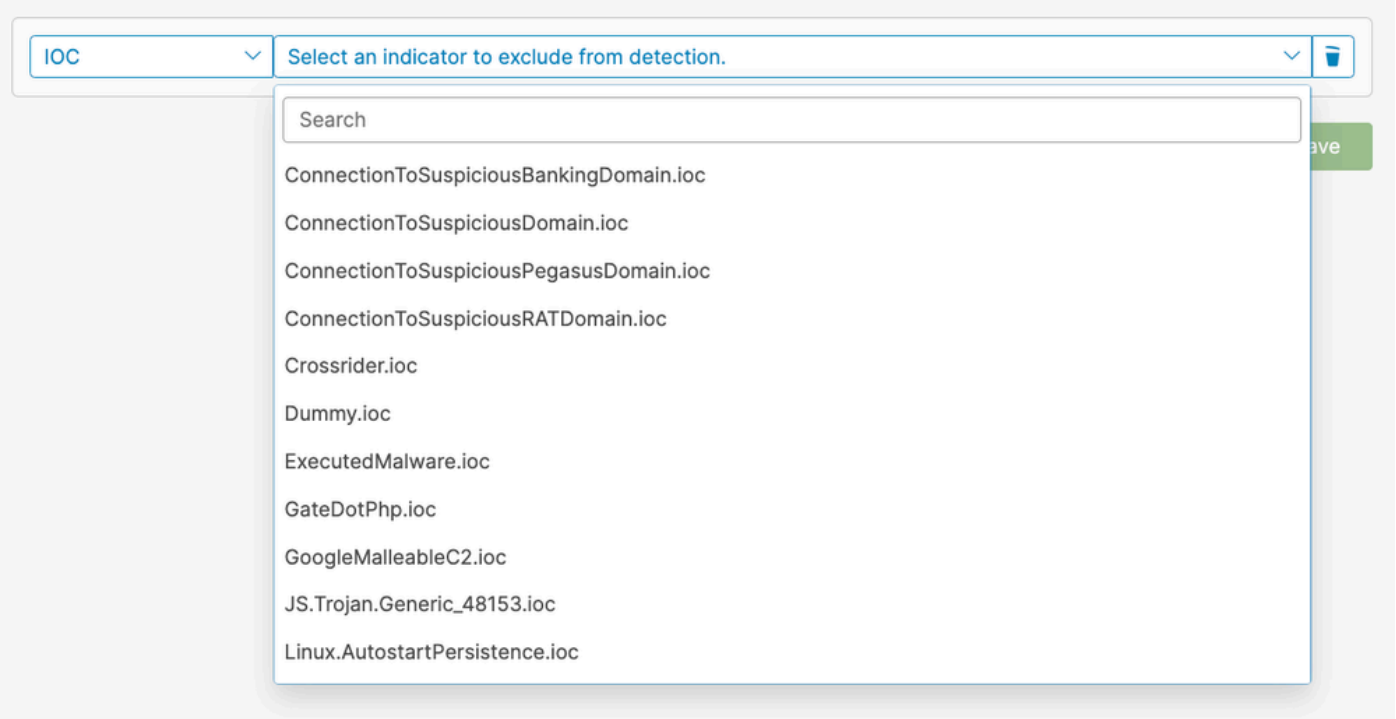
Note: Only applications can be excluded using Executable exclusions via Secure Endpoint Console. Any exclusions related to DLLs require opening a support case for an exclusion to be created.

Finding the correct exclusions for Exploit Prevention is a far more intensive process than any other exclusion type and requires extensive testing to minimize any detrimental security holes.

IOC Exclusions (Windows-only)

IOC exclusions allow you to exclude Cloud Indications of Compromise. This can be useful if you have a custom or internal application that can not be signed and causes certain IOCs to trigger frequently. Secure Endpoint Console provides a list of indicators to choose from for IOC exclusions.

You can select which indicators to exclude via a dropdown:



Note: If you exclude a high or critical severity IOC you lose visibility into it and could leave your organization at risk. You must only exclude these IOCs if you experience a large number of false-positive detections for it.

CSIDL and KNOWNFOLDERID (Windows-only)

CSIDL and KNOWNFOLDERID values are accepted and encouraged when writing path and process exclusions for Windows. CSIDL/KNOWNFOLDERID values are useful for creating process and path exclusions for environments that use alternate drive letters.

There are limitations that need to be considered when CSIDL/KNOWNFOLDERID is used. If your environment installs programs on more than one drive letter, the CSIDL/KNOWNFOLDERID value only refers to the drive marked as the default or known installation location.

For example, if the OS is installed on C:\ but the installation path for Microsoft SQL was manually changed to `D:\`, the CSIDL/KNOWNFOLDERID based exclusion in the maintained exclusion list does not apply to that path. This means one exclusion must be entered for each path or process exclusion not located on the C:\ drive as the use of CSIDL/KNOWNFOLDERID does not map it.

Refer to this Windows documentation for more information:

- CSIDL
- KNOWNFOLDERID

Note: KNOWNFOLDERID is only supported in Windows connector 8.1.7 and later. Earlier versions of the Windows connector use CSIDL values.

Note: The KNOWNFOLDERID values are case sensitive. For example, you must use the value **FOLDERID_ProgramFiles** and not the invalid **valueFolderID_programfiles**.

Prepare Connector for Exclusion Tuning

To prepare your connector for exclusion tuning, you need to:

1. Set up a policy and group to run in Debug mode.
2. Run the computers in the new Debug group as per normal business operations, allow time to obtain sufficient connector log data.
3. Generate diagnostic data on the connector to use for identifying exclusions.

Refer to these documents for instructions on enabling Debug Mode and gathering diagnostic data on different operating systems:

- Cisco Secure Endpoint Connector for Mac Diagnostic Data Collection
- Cisco Secure Endpoint Connector for Linux Diagnostic Data Collection
- Analyze AMP Diagnostic Bundle for High CPU (Windows)

Identify Exclusions

MacOS and Linux

The diagnostic data generated in Debug mode provides two files that are useful for creating exclusions: fileops.txt and execs.txt. The fileops.txt file is useful for creating Path/File Extension/Wildcard Exclusions and the execs.txt file is useful for creating Process Exclusions.

Creating Process Exclusions

The execs.txt file lists the executable paths that triggered Secure Endpoint to perform a file scan. Each path has an associated count that indicates how many times it was scanned and the list is sorted in descending order. You can use this list to determine processes with a high volume of execute events and then use the process path to craft exclusions. However, it is not recommended to exclude general utility programs (such as /usr/bin/grep) or interpreters (such as /usr/bin/ruby).

If a general utility program or interpreter is generating a high volume of file scans, you can do some more investigating to try and craft more targeted exclusions:

1. Exclude the parent process: determine what application is executing the process (suc as find the parent process that is executing grep) and exclude this parent process. This must be done, if and only if, the parent process can be safely made into a process exclusion. If the parent exclusion applies to children, then the calls to any children from the parent process are also excluded.
2. Exclude the process for a given user: determine what user is executing the process. If the process is being executed at high volume by a specific user, you can exclude the process for just that specific user (such as if a process is being called at a high volume by user root,you can exclude the process, but only for specified user root, this allows Secure Endpoint to monitor executes of a given process by any user that is not root).

Example output of execs.txt:

```
33 /usr/bin/bash
23 /usr/bin/gawk
21 /usr/bin/wc
21 /usr/bin/sleep
21 /usr/bin/ls
19 /usr/bin/pidof
17 /usr/bin/sed
14 /usr/bin/date
13 /usr/libexec/gdb
13 /usr/bin/iconv
11 /usr/bin/cat
10 /usr/bin/systemctl
 9 /usr/bin/pgrep
 9 /usr/bin/kmod
 7 /usr/bin/rm
 6 /usr/lib/systemd/systemd-cgroups-agent
 6 /usr/bin/rpm
 4 /usr/bin/tr
 4 /usr/bin/sort
 4 /usr/bin/find
```

Creating Path, File Extension, and Wildcard Exclusions

The fileops.txt file lists the paths where file create, modify, and rename activities triggered Secure Endpoint to perform file scans. Each path has an associated count that indicates how many times it was scanned and the list is sorted in descending order. One way to get started with Path exclusions is finding the most frequently scanned file and folder paths from fileops.txt and then consider creating rules for those paths. While a high count does not necessarily mean the path must be excluded (such as a directory that stores e-mails can be scanned often but must not be excluded), the list provides a starting point to identify exclusion candidates.

Example output of fileops.txt:

```
31 /Users/eugene/Library/Cookies/Cookies.binarycookies
24 /Users/eugene/.zhistory
9 /Users/eugene/.vim/.temp/viminfo
9 /Library/Application Support/Apple/ParentalControls/Users/eugene/2018/05/10-usage.data
5 /Users/eugene/Library/Cookies/HSTS.plist
5 /Users/eugene/.vim/.temp/viminfo.tmp
4 /Users/eugene/Library/Metadata/CoreSpotlight/index.spotlightV3/tmp.spotlight.state
3 /Users/eugene/Library/WebKit/com.apple.Safari/WebsiteData/ResourceLoadStatistics/full_browsing_sessio
3 /Library/Logs/Cisco/supporttool.log
2 /private/var/db/locationd/clients.plist
2 /Users/eugene/Desktop/.DS_Store
2 /Users/eugene/.dropbox/instance1/config.dbx
2 /Users/eugene/.DS_Store
2 /Library/Catacomb/DD94912/biolockout.cat
2 /.fseventsd/000000000029d66b
1 /private/var/db/locationd/.dat.nosync0063.arg4tq
```

A good rule of thumb is that anything with a log or journal file extension must be considered a suitable exclusion candidate.

Behavioral Protection Engine

The Behavioral Protection engine was introduced in Linux connector version 1.22.0 and in macOS connector version 1.24.0; starting with these versions, the connector can detect overwhelmingly high system activity and then raise fault 18.

Process exclusions are applied to all engines and file scans. Apply process exclusions to very active benign processes in order to remediate this fault. Generated by the Debug Mode diagnostic data, the top.txt file can be used to determine the most active processes on the system. Please refer to the Secure Endpoint Mac/Linux Connector Fault 18 guidance for detailed remediation steps.

In addition, process exclusions can silence false-positive behavioral protection detections from benign software. For false-positive detections in the Secure Endpoint Console, the process can be excluded to improve reporting.

Windows

Windows operation system is more complicated, more exclusion options are available due to the parent and child processes. This indicates that deeper review is required to identify the files which had been accessed, but also the programs which generated them.

Please refer to this Windows Tuning Tool from Cisco Security's GitHub page to obtain more details about how to analyze and optimize Windows performance with Secure Endpoint.

Creating Exclusion Rules in the Secure Endpoint Console

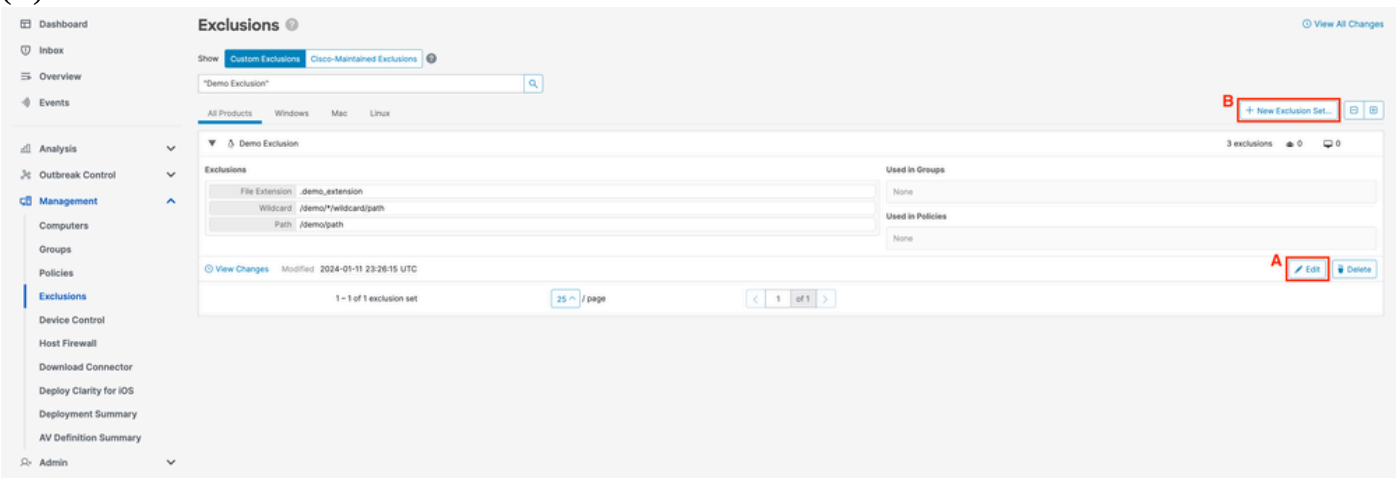
⚠ **Caution**: Always understand the files and processes before writing an exclusion to avoid security vulnerabilities on the endpoint.

Complete these steps to create a new exclusion rule using the Secure Endpoint Console:

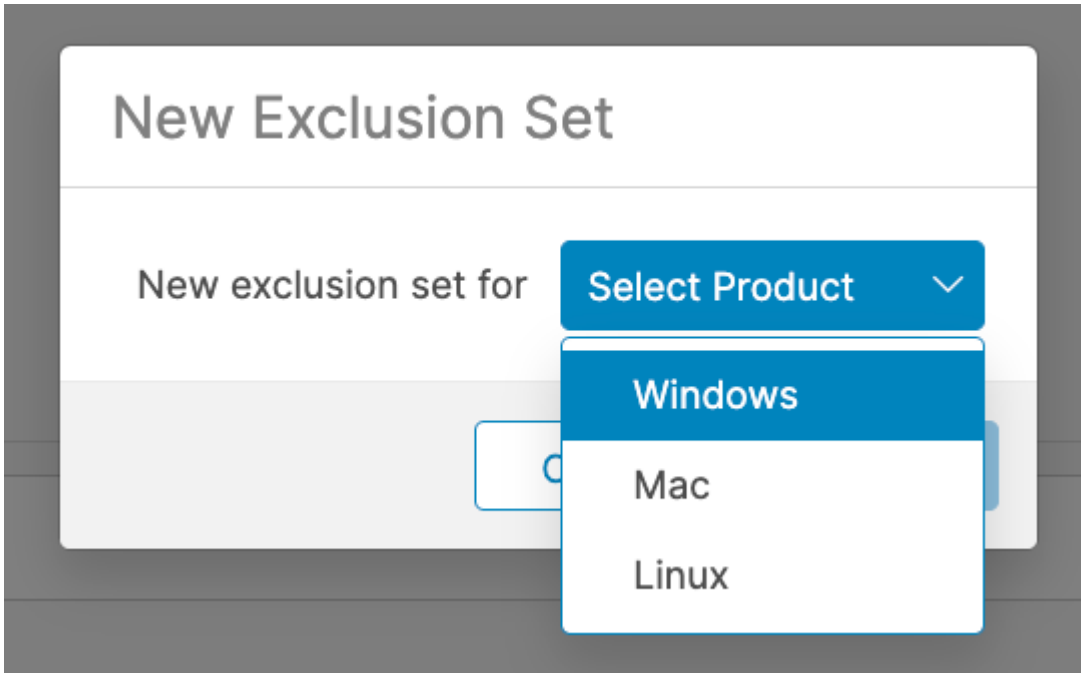
1. In the Secure Endpoint Console, navigate to the **Policies** page by selecting **Management** > **Exclusions**. Either:

(A) locate the exclusion set you wish to modify and click **Edit**, or

(B) click **+ New Exclusion Set...**.

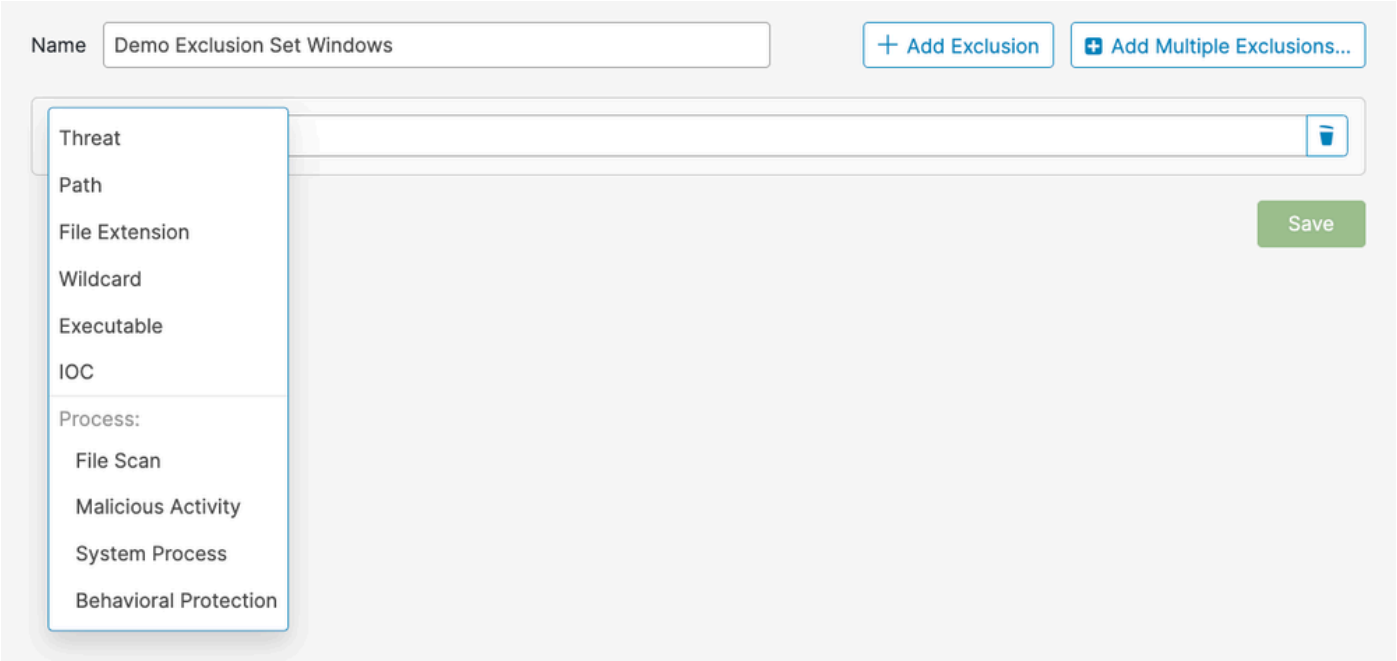


1. In the **New Exclusion Set** popup, select an **operating system** to create the exclusion set for. **Click Create**.

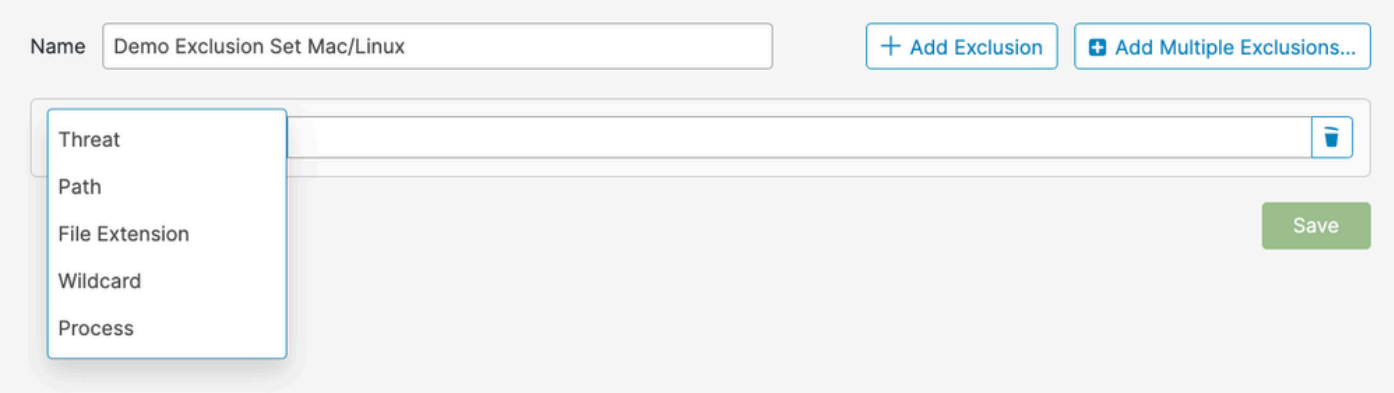


2. You are redirected to a **New Exclusion Set** page. Click **+ Add Exclusion** and select the **exclusion type** from the **Select Type** dropdown.

Windows:



Mac/Linux:



1. Fill out the **required fields** for the selected exclusion type.
2. Repeat steps 2 and 3 to add more rules, or click **Save** to save the exclusion set.

Best Practices

⚠ **Caution**: Use caution when creating exclusions as they reduce the level of protection provided by Cisco Secure Endpoint. Excluded files are not hashed, scanned or available in the cache or cloud, activity is not monitored, and information is missing from Backend Engines, Device Trajectory and Advanced Analysis.

Exclusions must only be used in targeted instances such as compatibility issues with specific applications or performance issues that cannot otherwise be improved.

Some best practices to adhere to when creating exclusions are:

- Only create exclusions for proven issues.
  - Do not assume an exclusion is necessary unless it has been proven to have been a problem that cannot be addressed otherwise.
  - Performance issues, false positives, or application compatibility issues must be thoroughly investigated and mitigated before applying an exclusion.
- Prefer Process exclusions over Path/File Extension/Wildcard exclusions.
  - Process exclusions provide a more direct way to exclude benign software activities than using a combination of Path, File Extension and Wildcard exclusions to achive the same result.
  - It is recommended to replace replace Path, File Extension and Wildcard exclusions that target program executables with corresponding Process exclusions when possible.
- Avoid broad exclusions.
  - Do not exclude large portions of the endpoint, such as the entire C drive.
  - Use the fully qualified path to the file instead of just the filename.
  - Use Device Trajectory, Secure Endpoint Diagnostics Data, and the Windows Tuning Tool to investigate and determine specific exclusions.
- Avoid overusing wildcard exclusions.
  - Be careful when creating exclusions with wildcards. Use more specific exclusions when possible.
  - Use the minimum amount of wildcards in an exclusion; only the folders that are truly variable must use a wildcard.
- Avoid excluding General Utility Programs and Interpreters.
  - It is not recommended to exclude general utility programs or interpreters.
  - If you do need to exclude a exclude general utility programs or interpreters then provide a

process user (macOS/Linux only).
    - For example, avoid writing exclusions that include python, java, ruby, bash, and sh.
- Avoid duplicate exclusions.
    - Before creating an exclusion, check if the exclusion already exists either in the Custom Exclusions or the Cisco-Maintained Exclusions.
    - Removing duplicate exclusions improves performance and reduces operational management of exclusions.
    - Ensure the path specified in a Process exclusion is not covered by a Path/File Extension/Wildcard exclusion.
- Avoid excluding processes known to be commonly used in malware attacks.
    - See Not Recommended Exclusions for more details.
- .Remove stale exclusions.
    - Regularly review and audit your exclusion list and keep a record of why certain exclusions were added.
- Remove exclusions on compromise.
    - Exclusions must be removed a connector is compromised in order to regain optimum security and visibility.
    - Automated Actions can be used to apply more secure policies to connectors post-infection. If a connector becomes compromised, it must be moved to a group that contains a policy without any exclusions to ensure the highest level of protection is applied.
    - Refer to Identify Conditions to Trigger Automated Actions in Secure Endpoint for more details on how to proactively setup the "Move Computer to Group upon Compromise" Automated Action.
- Increase protection on excluded items.
    - When exclusions are absolutely necessary, consider what mitigating tactics can be taken such as enabling write protection to add some layers of protection for the excluded items.
- Create exclusions intelligently.
    - Optimize rules by choosing the highest level parent process that uniquely identifies the application to exclude and use the Apply to Child Process option to minimize the number of rules.
- Never exclude the startup process.
    - The startup process (launchd on macOS, init or systemd on Linux) is responsible for starting all other processes on the system and is at the top of the process hierarchy.
    - Excluding the startup process, and all its children processes, would effectively disable Secure Endpoint monitoring.
- Specify the process user when possible (macOS/Linux only).
    - If the user field is left blank, the exclusion applies to any process running the specified program.
    - While an exclusion that applies to any user is more flexible, this broad scope could unintentionally exclude activity that must be monitored.
    - Specifying the user is especially important for rules which apply to shared programs such as runtime engines (for example, java) and script interpreters (for example, bash, python).
    - Specifying the user limits scope and directs Secure Endpoint to ignore specific instances while monitoring other instances.

Not Recommended Exclusions

Although it is impossible to know every possible attack vector that an adversary can use, there are some core attack vectors that must be monitored. To maintain good security posture and visibility, these exclusions are not recommended:

AcroRd32.exe
addinprocess.exe
addinprocess32.exe
addinutil.exe

bash.exe
bginfo.exe
bitsadmin.exe
cdb.exe
csi.exe
dbghost.exe
dbgsvc.exe
dnx.exe
dotnet.exe
excel.exe
fsi.exe
fsiAnyCpu.exe
iexplore.exe
java.exe
kd.exe
lxssmanager.dll
msbuild.exe
mshta.exe
ntkd.exe
ntsd.exe
outlook.exe
psexec.exe
powerpnt.exe
powershell.exe
rcsi.exe
svchost.exe
schtasks.exe
system.management.automation.dll
windbg.exe
winword.exe
wmic.exe
wuauclt.exe
.7z
.bat
.bin
.cab
.cmd
.com
.cpl
.dll
.exe
.fla
.gif
.gz
.hta
.inf
.java

.jar
.job
.jpeg
.jpg
.js
.ko
.ko.gz
.msi
.ocx
.png
.ps1
.py
.rar
.reg
.scr
.sys
.tar
.tmp
.url
.vbe
.vbs
.wsf
.zip
bash
java
python
python3
sh
zsh
/
/bin
/sbin
/usr/lib
C:
C:\
C:*
D:\
D:*
C:\Program Files\Java
C:\Temp\
C:\Temp*
C:\Users\
C:\Users*
C:\Windows\Prefetch
C:\Windows\Prefetch\
C:\Windows\Prefetch*
C:\Windows\System32\Spool

C:\Windows\System32\CatRoot2
C:\Windows\Temp
C:\Windows\Temp\
C:\Windows\Temp*
C:\Program Files\<company name>\
C:\Program Files (x86)\<company name>\
C:\Users\<UserProfileName>\AppData\Local\Temp\
C:\Users\<UserProfileName>\AppData\LocalLow\Temp\



Note: This is not an exhaustive list of exclusions to avoid, but it provides insight into the core attack vectors. Maintaining visibility into these paths, file extensions, and processes is crucial.

Related Information

- Technical Support & Documentation - Cisco Systems
- Cisco Secure Endpoint - TechNotes
- Cisco Secure Endpoint - User Guide
- Troubleshoot Exploit Prevention in Secure Endpoint
- Identify Conditions to Trigger Automated Actions in Secure Endpoint