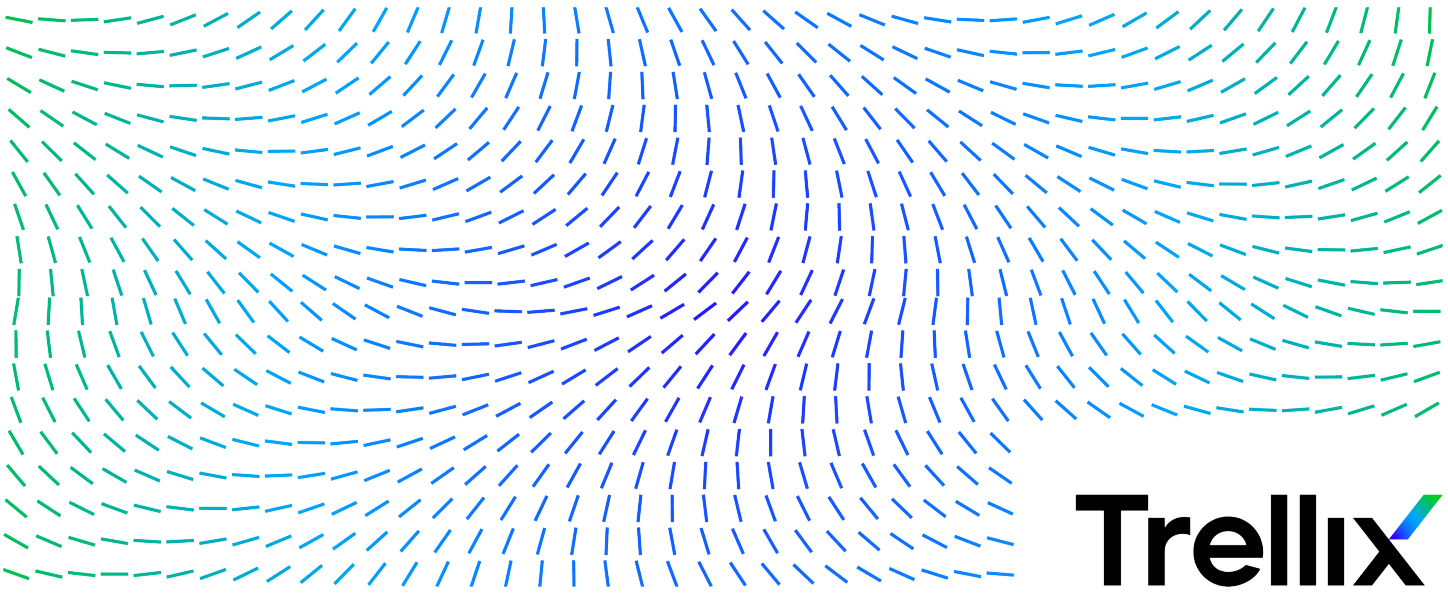
McAfee Active Response 2.4.x Product Guide



Trellix

Contents

Product overview

Overview

McAfee® Active Response is an endpoint detection and response tool that finds and responds to advanced threats.

Some of the main benefits of using Active Response include:

- Early detection of suspicious activity or indication of prior attacks
- Quick and effective way to deal with security breaches
- Reduce resources needed to detect risks from unknown processes running on endpoints
- Collect information about potentially malicious processes
- Act on shared threat intelligence with simplified workflows

With Active Response, you can take quick corrective actions to remediate a threat, and adapt protection measures against future attacks.

Active Response brings together McAfee® Threat Intelligence Exchange (TIE) and McAfee® Data Exchange Layer (DXL). Together they provide global threat information with locally collected, customer-specific intelligence that can be shared, allowing multiple security solutions to operate as one.

Active Response, Threat Intelligence Exchange, and Data Exchange Layer function together to narrow the gap from encounter to containment for advanced targeted attacks from days, weeks, or months down to seconds.

Key features

Active Response displays potential threats ranked by risk, so you can investigate, correct, and adapt with a single-click action.

Use near real-time searches and hunting flows based on collectors, triggers, and reactions. Collectors and reactions can be customized and used with the defaults. The key features of Active Response help detect threats and offer continued protection.

- **Collectors** — Collectors enable users to search and analyze data regarding critical breach or potential attack from endpoints. You can prioritize the high-risk potential threats based on behavior to focus your investigation on the most important threats. Monitor your environment with customizable collectors that search for indicators of attack that are active or dormant, but also that might have been deleted. You can also search for live and historical threat data to determine the full scope of an attack.
- **Triggers** — Triggers are set of instructions used to continuously monitor a critical event, and its change of state in the system. A trigger set beforehand initiates an action, generating an event that can execute responses.
- **Reactions** — Reactions are based on triggers and provide preconfigured and customizable actions, enabling you to search and eliminate threats. You can automate reactions based on triggers and act on multiple endpoints remotely at the same time. Use triggers and reactions to detect threatening events and react immediately.

  Reactions are preconfigured to initiate actions on search findings such as:

- Files deleted from the system by file hash (MD5 and SHA-1)
- Hosts that are actively connected to an IP address or have connected to an IP address in the past
- A non-PE based malicious file that is not accessed on the system or
- A malicious PDF on a system where it was copied to the file system but not opened

You can also customize collectors and reactions for adapting threat investigation and detection flows.

- **Centralized Management** — McAfee® ePolicy Orchestrator® (McAfee® ePO™) provides a single-console environment for comprehensive management and automation. You can use the Threat Workspace to see potential threats on endpoints, where they started, and how they moved through the environment, and their activities over time.

  Remediation actions can be set from the Threat Workspace with a single-click. For example, you can stop a running process on a single endpoint, or remove a threat and block it from recurring in the environment. You can also filter behavior conditions of potential threats. Adapt protection settings to automatically block persistent attacks. The analysis of existing threats helps you learn what to include in security policies.

- **Integrated Security Architecture** — Active Response uses DXL to streamline communication with other products such as TIE, McAfee® Advanced Threat Defense, Endpoint Security, and McAfee® Enterprise Security Manager (McAfee ESM).

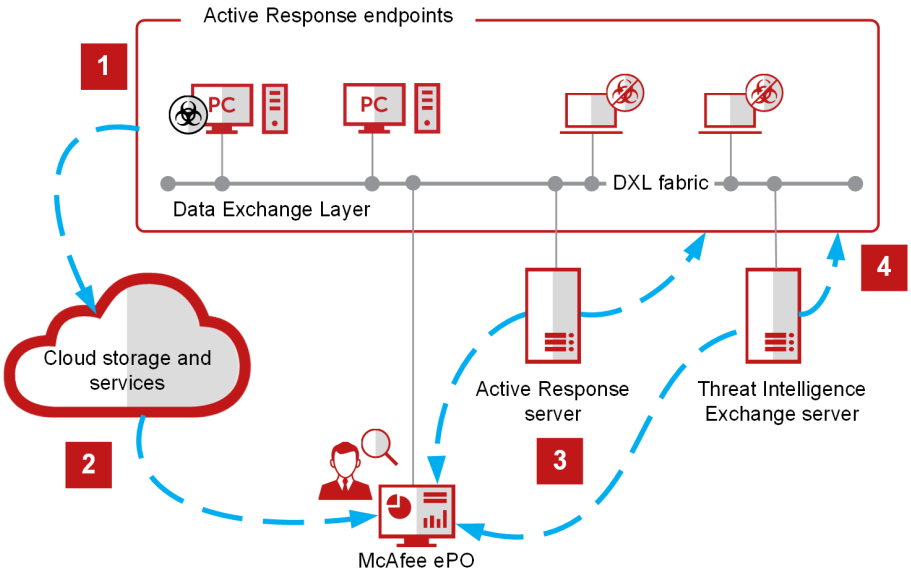
How it works

Active Response is composed of a cloud service, a server, a set of extensions, and endpoint clients.

1. The Active Response client, which runs on managed endpoints, includes a Trace module that scans and captures data about potential threats (processes) on the managed endpoints. This data is then sent to cloud storage via a DXL broker plugin. The Trace module is available on Microsoft Windows and macOS (Beta release) systems only. It enables:
   a. Continuous collection of potential threat information.
   b. Responses to information queries from the Active Response server.
   c. Execution of remediation actions on specific threats.
2. The Active Response **Threat Workspace**, installed as a McAfee® ePolicy Orchestrator® (McAfee® ePO™) extension directly retrieves the data stored in the cloud and enables visualization of threats that are seen across the endpoints. Active Response has two main extensions managed by McAfee ePO.
   a. **Threat Workspace** — Enables the visualization of potential threat information gathered from the endpoints. In-depth investigation of a potential threat is performed in the **Threat Workspace**, with additional information retrieved on-demand from the endpoints by the Active Response server. You can remediate a potential threat from the **Threat Workspace**, and the remediation actions take effect immediately on the endpoints.
   b. **Active Response Search** — Enables real-time searches over the endpoints. It also provides the ability to save searches, create custom collectors, and define triggers and reactions.

   The potential threat information from the endpoints is stored in the cloud (up to 90 days of endpoint data). Aggregation of endpoint data in the cloud provides the overall health status of the enterprise.
3. In-depth investigation of a threat is performed in the **Threat Workspace**, with additional information retrieved on-demand from the endpoints by the Active Response server. The Active Response server is the central coordinator of the Active Response solution. It communicates with the Active Response client running on managed endpoints to collect data and execute remediation actions.

4. The Threat Intelligence Exchange server provides reputation information and helps to investigate threats. You can override a reputation setting in the **Threat Workspace**, and that setting is sent to the TIE server and updated throughout your environment. You can also block future recurrences of a threat by changing the reputation of a process to Make Known Malicious, which is updated in the TIE server.



Content packages

There are two content packages that are installed with Active Response, one for built-in collectors and the other for trace and detection rules. For more information about updating content packages, see the *McAfee Active Response Installation Guide*.

Using Active Response

Using the Threat Workspace

The Threat Workspace is where you can see all potential threats on managed endpoints and respond to them.

This is where you can detect and remediate potential threats in one place. Actions performed on potential threats are immediately made available to all managed endpoints in the environment. The **Threat Workspace** includes several parts where you can view and react to potential threats. The workflow moves from left to right.

Note

Only Microsoft Windows systems information is included on the **Threat Workspace**. Dates displayed throughout Active Response are based on the timezone setting in the user's browser.
On Active Response 2.4.0, tracing is supported on macOS (Beta) and Linux.

If configuration issues need attention, the **Health Status Alert** window appears when you open the Workspace. Click the link to open the **Health Status** page.

Threat Workspace bar

View threats and their severity levels, navigate to the **Remediation History** page, refresh the **Threat Workspace**, filter events over a period of time, or configure a cloud account.

Feature	Description
Total Threats	When a process executes on a managed endpoint, its behavior is traced. Based on the detected behavior, the process is categorized and assigned a severity level, but ultimately you decide whether it is a potential threat, and what to do about it. The severity level of the root process increases when the threat level of its child processes increases, or the accumulated threat levels of its child processes increase. The severity levels are: • **High Risk** — The process appears to be a high risk of being a threat and must be immediately investigated and remediated. • **Suspicious** — The process appears suspicious and must be investigated and remediated. • **Monitored** — The risk for the process cannot be determined. Active Response continues to monitor the process and changes its status based on behavior and further analysis. You can filter the view of the **Potential Threats** list by clicking a threat severity level, such as **High Risk** to display only those events. Click **Total Threats** to display all potential threats.

Feature	Description
Quarantined Hosts	This link opens the **Active Response Search** page with a query to search for and output a list of quarantined hosts. Select one or more hosts, and from the **Actions** drop-down list select **Execute Reaction** → **UnquarantineHost** to end the quarantine.
Remediation History	View the history of remediated events.
Refresh	Refresh the Workspace to redisplay content.
Last range of time	Select a time period to filter the information displayed in the Workspace. This filter is based on last seen threats and not first seen. This means that if an old threat keeps reappearing, it is not filtered. The time period can range from the last 4 hours up to last 90 days.
Configuration	Register a new cloud account, log on to your account, reset your password, or select a cloud storage geolocation. Geolocation is the geographic location or data residence of the cloud storage and service that complies with the regulatory requirements of the country or region in where it resides. ⚠ **Caution:** Switching between different geolocations is not supported or recommended, because you might lose data. This setting is meant to be permanent.

Potential Threats window

A potential threat is identified in the Workspace by the hash and first seen name of the root process.

Potential Threats are a set of processes involved with behaviors Active Response considers a threat. A root process originates the tracing of all these processes.

The root process and related child processes are displayed as one entry and the root process's severity level includes the highest severity level of its child processes.

Active Response supports detecting and remediating non-PE file-based PowerShell scripts and non-PE file-less PowerShell command-line executions. PowerShell script and command-line events appear in the **Potential Threats** list, naming the PowerShell interpreter. A PowerShell event is triggered and displayed in the Trace timeline if the PowerShell interpreter executes a script (.ps1) or a command line.

Note

Non-PE threats are of three types: command-line, file-based, and PowerShell.

The **Event Details** page lists details about the process with advanced searching capability, and displays the entire command-line string to copy and paste to your clipboard. To search for command-line events, use the advanced searching links on the **Event Details** page. For file-less command-line events, the MD5 hash is generated from the command-line content, so if the hash is used in a search query, results are empty.

Use the **Host Actions** menu to remediate these events on endpoints:

- **Dismiss** — Removes the potential threat from the list.
- **Stop** — Stops the instance of the PowerShell interpreter that is running the script.
- **Stop and Remove** — Stops the instance of the PowerShell interpreter running the script and removes the script.

Potential threats are identified with a severity level, **High Risk** (red), **Suspicious** (orange), or **Monitored** (yellow). The potential threats displayed are based on the selected time frame.

Feature	Description
Search box	Filter potential threats by a keyword, such as its process name and host name.
By behavior risk	Sort the potential threats in ascending or descending order.
Number of hosts affected (monitor icon)	The number of hosts that the potential threat has impacted (in the present and past). The filter is based on the configured age limit. If a threat appears before the specified time, that threat is not displayed.
Age (clock icon)	The time that has passed since the potential threat's file was first seen.

If a process in the list is unique to your environment and is not a threat, you can set its reputation to **Known Trusted**. A **Known Trusted** process can appear in the list if it exhibits suspicious behavior. Endpoint Security protection products block threats whose enterprise reputation is **Known Malicious**, so these threats are not displayed in the **Potential Threats** list.

When you select a potential threat from the list, details about the threat are displayed in **Affected Hosts**, **Trace** timeline, **Event Details**, and **Reputation** windows. You can drill deeper into the details by clicking links in the **Event Details** window. Based on your investigation, you can execute actions from the **Host Actions** or **Global Actions** menus to remediate the threat.

Affected Hosts window

The **Affected Hosts** window lists details about each host affected by the potential threat. These can be hosts where the threat is running or has run in the past. This information comes from the Threat Intelligence or Endpoint Security Adaptive Threat Protection module on the endpoint.

You can select one or more hosts and apply an action from the **Host Actions** menu.

Feature	Description
Stop process	• Stops the selected potential threat's process tree (root and child processes) currently running on the selected hosts. • Stops the instance of the PowerShell interpreter that is running the script.
Stop and remove	• Stops the selected potential threat's process tree (root and child processes) and removes it from the selected hosts. For Active Response clients 2.1 and later, it performs a Validation and Trust Protection (VTP) check on the endpoints before removing the file. • Stops the instance of the PowerShell interpreter running the script and removes the script from the endpoint.
Advanced mode	**Advanced Mode** is available for **Stop process** and **Stop and remove** actions. • You can apply separate remediation actions to multiple child processes. For example, you can apply **Stop and remove** to a child process and apply **Stop instance** to a different child process. • Click **Show also Trusted processes** to show all trusted child processes tied to the potential threat.
Dismiss	Removes the potential threat from the **Potential Threats** list. This action does not impact the affected host or update the TIE reputation database. The potential threat reappears in the workspace if there is new activity. You can search for and review past dismiss actions by selecting **Menu** → **User Management** → **Audit Log** or clicking **Remediation History** in the **Threat Workspace** bar.
Quarantine	Isolates the host from the network while retaining connectivity to McAfee products, blocking all network communication of non-McAfee trusted processes. You can continue to run searches on this host with McAfee products to investigate the potential threat. A message notifies the user of the quarantine/end quarantine action and the quarantine is maintained between rebooting and shutting down. The **Quarantine** reaction requires that the **NetworkFlow** plug-in is enabled. A virus icon to the left of the host's name indicates the host is quarantined. If the host's connection status displays **Offline**, you can still apply the quarantine. When the host comes back online, the quarantine action takes effect.

Feature	Description
	You can edit the quarantine/end quarantine messages from the **Menu** → **Policy** → **Policy Catalog** → **Network Flow** tab. Select **Display Message on Quarantine actions** and write your custom notification messages in the text boxes below. You can also view quarantined hosts in different ways: • Click **Quarantined Hosts** in the **Threat Workspace** bar to run a search for quarantined hosts and view the output. From the **Actions** drop-down list, select **Execute reaction** → **Unquarantine host** to end the quarantine. To see the status change immediately, select **Menu** → **System Tree** → **Wake Up Agents** and enable **Force complete policy and task update**. • View a host's quarantine status as an Active Response property on the McAfee ePO server. Select **Menu** → **Systems** → **System Tree** to view the **Endpoint Quarantined** column. To view more details about this quarantined endpoint, select its name and click the **Products** tab. Select **Active Response** and scroll down to view the **Active Response Properties** → **Endpoint Quarantined** status of true or false (true = quarantined). • Quarantine events are captured in the **Threat Events Log** and **Audit Log**.
End Quarantine	Removes the quarantine and reconnects the host to the network.

Trace window

The **Trace** window shows details about where the potential threat started on a particular host, what other processes it started, and how those processes moved through the hosts in your environment over time for each endpoint.

Parts of the Trace window

Select a potential threat in the list to view details in the **Trace** window. At the top of the window (from left to right) are icons used for:

- Filtering the threats by severity and event types. You can use the **Filter** drop-down list to filter by severity filters such as **High risk**, **Suspicious**, **Monitored** and **Other**. You can also filter by event types such as **Processes**, **Files**, **Registry Keys**, **Network Connections**, and **Process Reputation change**.
    - If the potential threat on the affected host is a non-Windows system, the **Registry Keys** checkbox is grayed out.
    - If certain event types are not present on the Trace timeline, those event filter checkboxes are grayed out.
- Refreshing the trace information.
- Exporting the trace information to a JSON or CSV file. The output for CSV file includes actor fields to easily identify host activities (**actorSha256**, **actorPid**, **actorImageName**, **actorFullPath**, **actorCmdLine**).
- Displaying a sequential view of all the events or processes executed in the endpoints. This view does not display the time at which these events or processes are executed.

- Displaying a time view of the trace chart. You can view when a specific event or process is executed. A navigation pane along the top of the trace chart shows activity spikes, and enables you to select the time frame to view in the trace chart. Use the mouse scroll wheel to zoom in or out of a particular view.
- Changing the view from chart to table. In table view, you can sort and search the columns of information. Clicking the event displays the **Event Details** window. Table view lists each event from the Trace chart in a table row categorized by **Event type**, **Actor**, **Date**, and **Summary**. Sort the information by clicking the column heading and use the search box to filter on a keyword. Click the event row in the table to display the **Event Details** window.
- Expanding the trace chart to full-screen view.

Identifying events

Icons on the Trace timeline represent the type of event being traced (processes, files, registry keys, network connections, and process reputation change). Click an event to display the details and investigate the suspicious activity.

- Injected files are syringe icons.
- Blocked events are displayed with a slash through the icon.
- A numbered badge on the event icon indicates multiple instances of the same event.
- An icon with a downward arrow in a box indicates a process reputation change. Hover your mouse over the icon to see a quick summary of the change in reputation and the reason for the change.
- Clicking an event on the timeline opens the **Event Details** window.

Investigating events

The **Event Details** window lists general information such as event type, time stamp, behavior observed, and processes blocked by another product. Specific details are grouped into categories: **Process Reputation**, **Process**, **File**, **Registry**, **Network**, and **User**. Investigate deeper into the event by clicking a link in a category to perform advanced searching. You can use **Host Actions** to stop, stop and remove, or dismiss a process from the host that you're investigating.

Event Details actions

Category	Link item	Action
Process Reputation info	Reason	• **Find untrusted modules loaded by this process in this host.**
Process info	Name	• **Find process on this host** • **Find process on all hosts** • **Find network flow of this process in this host** • **Find network flow of this process in all hosts** • **Find modules loaded by this process in this host** • **Find modules loaded by this process in all hosts**
	Command line (copy icon)	• Copies the command-line path.

Category	Link item	Action
	Command line (path)	• **Find process command line on this host** • **Find process command line on all hosts**
File info	Name Path MD5 SHA1 SHA256	• **Find file on this host** • **Find file on all hosts**
Registry info	Key path Key value name	• **Find this key in all hosts** • **Find registry key on all hosts**
Network info	Destination IP address	• **Find network flow to this IP on this host** • **Find network flow to this socket**
User info	User name	• **Find process for this user on this host** • **Find process for this user on all hosts** • **Check if this user is logged on to other hosts**
Injector process info	Name Path MD5 SHA1 SHA256	• **Find file on this host** • **Find file on all hosts**
Injector user info	User name	• **Find process for this user on this host** • **Find process for this user on all hosts** • **Check if this user is logged on to other hosts**

Copy the URL of a trace position to share or bookmark

Capture an instance of the Threat Workspace to share or bookmark using the URL.

- Select a potential threat and copy the URL of the Threat Workspace or you can specifically target an event's position in the Trace timeline and copy the URL. You can send the URL to a teammate for further investigation or collaboration. The recipient of the URL must have access to that endpoint.
- Bookmark the trace event to investigate further later.
- An error message is displayed if the URL has invalid data.
- The URL defaults to the original workspace view if the information is no longer there because the potential threat was remediated or if the threat is not seen for 90 days.

Enabling the trace plug-in for macOS endpoints

Turn on trace events for macOS endpoints by enabling the plug-in.

Before you begin

Verify macOS endpoints meet the minimum Active Response requirements.

The trace plug-in for macOS is disabled by default.

Note

Trace for macOS is in Beta release.

Task

1. Log on to McAfee ePO as administrator.
2. Select **Menu** → **Policy** → **Policy Catalog**, and the **Trace** tab.
3. Select **Enable Plug-in for macOS Endpoints (Beta)**.
   Enable Plug-in is used for Windows only and does not affect the macOS plug-in setting.

Reputation window

The **Reputation** window displays detailed information about the file that generated the threat.

The window is divided into cards that display threat levels from different reputation servers configured in the Active Response environment. You can compare threat levels from multiple reputation sources to analyze potential threats within the Active Response workspace.

TIE Reputation card

The first card displays file reputation information retrieved from the TIE server:

- Reputation (for example, Unknown)
- First seen name (for example, DATAGENERATOR.EXE)
- Age (length of time the potential threat has been in the environment)
- Last Seen (time stamp of the last seen threat event)
- Prevalence (number of hosts affected)

Use the **Global Actions** menu to assign a reputation to the potential threat across all hosts in the environment.

Feature	Description	
Make Known Malicious	Changes the reputation of the selected process to **Known Malicious** and updates the reputation information in the TIE database. The entire process tree is killed and the root process is cleaned on systems that use TIE policies that block malicious files. The process continues to display on the Threat Workspace until the action is successfully completed on all affected hosts. If the parent process was killed and no longer exists, any remaining child processes spawned are also killed, removing the entire process tree.	
Make Known Trusted	Changes the reputation of the selected process to **Known Trusted** and updates the reputation information in the TIE database.	
	Advanced mode	**Advanced Mode** is available for **Stop instance** and **Make Known Trusted** actions. View more details about the potential threat's process tree, such as its MD5 hash and prevalence. • You can apply different reputations to multiple child processes. For example, you can apply **Make Known Malicious** to a child process and apply **Make Known Trusted** to a different child process. • Also, you can take remediation action on any of the child processes by selecting **Stop instance**. • Click **Show also Trusted processes** to show all trusted child processes tied to the potential threat.
Dismiss	Removes the potential threat from the **Potential Threats** list. This action does not impact the affected host or update the TIE reputation database. The potential threat reappears in the workspace if there is new activity. You can search for and review past dismiss actions by selecting **Menu** → **User Management** → **Audit Log** or clicking **Remediation History** in the **Threat Workspace** bar.	

Reputations card

The **Reputations** card displays the reputation sources configured with Active Response and displays their respective reputations of the potential threat. See the respective product guides for threat level definitions.

Abbreviation	Product name
TIE	McAfee® Threat Intelligence Exchange server
GTI	McAfee® Global Threat Intelligence™ (McAfee GTI) server
ATD	McAfee® Advanced Threat Defense server

Sandbox Result card

The **Sandbox Result** card displays the threat level or conviction of an uploaded sample analyzed by McAfee® Advanced Threat Defense. Sandbox results are displayed in the Active Response Workspace for Advanced Threat Defense (version 4.4.x and later). It supports a standalone server or the primary server of a cluster of physical or virtual servers.

The sample submissions are automatically uploaded by TIE to Advanced Threat Defense for analysis. You can also manually upload and submit the sample to the Advanced Threat Defense server for analyzing. The results are displayed in the card. Download the report to view details such as the file's data, hash details, and environment.

From the drop-down list, select **Download full report** or **Download IOCs**.

- The full report downloads a threat analysis in .pdf format.
- The IOC (Indicators of Compromise) downloads a malware summary in .xml format.

See the Advanced Threat Defense product guide for integration and analysis methods.

File Details card

File Details lists the potential threat's first seen name and hash details. **Latent** displays the number of hosts affected by this potential threat.

Investigate and remediate a potential threat

You can view the list of potential threats and easily see data about what the threat is, how long it has been in your environment, and which host systems are affected. You can then remediate the potential threat without having to open another window or product.

Note

If an endpoint is offline, its current state does not match the information available in the Workspace. When the endpoint comes back online, the remediation action is executed or the action is no longer effective because the potential threat was resolved by other means.

Task

1. Select **Menu** → **Systems** → **Active Response Workspace**.
2. Select the type of potential threats you want to see, for example **High Risk**, **Suspicious** or **Monitored**.
   The **Potential Threats** list shows all threat processes of that type. A process can appear as a potential threat, but ultimately you decide whether it is, and what to do about it.
   To find a specific threat, select **Total** to see all potential threats and use the search box to filter. You can search for a process name, file hash, IP address, or registry key.
3. Select a threat from the **Potential Threats** list.
   The information about that potential threat is displayed on the **Threat Workspace**.
   - The **Affected Hosts** lists the detailed information about each host affected by the potential threat. These can be hosts where the threat is running or has run in the past. This information comes from the Threat Intelligence module or Endpoint Security Adaptive Threat Protection module on the endpoint.
   - The **Reputation** and **Event Details** windows show detailed information about the file that generated the threat.
   - The **Trace** information shows details about where the potential threat started on a particular host, what other processes it started, and how those processes moved through the hosts in your environment. You can filter what is visible on the **Trace** timeline by selecting or deselecting the checkboxes for process reputation changes, processes, files, registry keys, and network connections.
4. Select **Host Actions** or **Global Actions** to take action on the potential threat.
   - **To stop or dismiss processes on one or more selected hosts** — Select one or more hosts in the **Affected Hosts** list, then select **Host Actions**. You can either stop the process tree currently running and leave it on the host (**Stop process**), stop the process tree and delete it from the host (**Stop and remove**), or remove only the selected host from the list (**Dismiss**). If you stop a process and leave it on the host, you can restart it later. Click **Advanced mode** to apply separate actions to root and multiple child processes.
   - **To quarantine a host in your environment** — Select a host in the **Affected Hosts** list and select **Host Actions**. Select **Quarantine** to block non-McAfee trusted products and only allow access to McAfee products. **End Quarantine** removes the quarantine action.
   - **To perform an action from the Trace timeline** — Select an event icon for the process you want to stop, then select **Host Actions**. You can either stop the process tree currently running and leave it on the selected host, stop the process tree and delete it from the host, or remove only the selected host from the list.
   - **To perform a global action on one or more potential threats** — From the **Reputation** window of the selected potential threat, select **Global Actions** to change the potential threat processes' reputation (**Make Known Malicious**/**Make Known Trusted**) or remove the potential threat from the list (**Dismiss**). The new reputation setting is updated and saved in the TIE database. The process is either blocked or allowed to run on managed endpoints throughout your environment, depending on the TIE policy configurations. Click **Advanced mode** to apply different reputations to root and multiple child processes.

When you perform an action on a process, a progress indicator appears next to the threat in the **Potential Threats** list, showing that the action is in process. Go to the **Remediation History** page to see details about the action.

5. Manage plug-ins and their features to isolate issues on endpoints. Enable or disable a plug-in and its features on the host in the **Policy Catalog** to reduce the functionality of Active Response. To view the plug-in status of this endpoint, select the endpoint's name and click the **Products** tab. Select **Active Response** and scroll down to view the **Active Response Features** table.

View threat remediation history

When an action is taken on a threat process in the **Threat Workspace**, a remediation item is created. You can view the remediation actions that were taken on specific threats, regardless of who initiated them.

Task

1. Click the **Remediation History** link at the top of the **Threat Workspace**, or select **Menu** → **Reporting** → **Remediation History**.
   The **Remediation History** page shows the threat processes that have been remediated. The information includes the action taken, the number of host systems affected by the remediation, and other details about the threat process.
2. Select an action to see its details.
   - **Make Known Malicious** or **Make Known Trusted** shows the current TIE reputation information for the file.
   - **Stop process** or **Stop and remove** shows details about the threat, including where it was running, the McAfee Agent GUID, and event information.
   - **Dismiss Threat** displays the process name, MD5, and remediation time stamp.
   - **Dismiss Host** displays the process name, MD5, remediation time stamp, and number of hosts affected.
3. Select the **Impacted Hosts** number to display the systems where the threat process was running when it was remediated, including information about the IP address and operating system. Selecting a system opens the **Systems Information** page listing the **System Properties**.

Delete threat remediation history

Use a server task to delete threat remediation history information.

Server tasks are configurable actions that run on McAfee ePO at scheduled time or intervals. You can create a server task to delete remediation entries older than a specific date.

Task

1. Select **Menu** → **Automation** → **Server Tasks**, then click **New Task**.
2. Give the task an appropriate name, and decide whether the task has a **Schedule** status. If you want the task to run automatically at set intervals, click **Enabled**, then click **Next**.

3. From the **Actions** drop-down, click **Purge Remediation History**. Specify how old a remediation record must be before it's purged, then click **Next**.
4. Choose the schedule type (the frequency), start date, end date, and schedule time to run the task.
   The **Summary** page appears.
5. Click **Save** to save the task.

Results

The new task appears in the **Server Tasks** list.

Searching endpoint data

Active Response searches data on your managed endpoints in real time.

Note

When online endpoints do not respond due to network issues, all searches time out automatically after a configurable amount of time.

The search box understands simple syntax to combine collectors and build powerful search expressions and filters. A search expression consists of two parts:

- A projection of at least one collector. The collector name specifies the data that Active Response returns. The projection lists the output fields that appear as columns in the **Search results** table. If no output fields are specified, the default output fields are presented.
- A filter applied to the values in the output fields, optionally. Filters specify conditions to match in returned data. Only data that matches the filter appear in the **Search results** table.

Simple search expression

Get all records returned by the **Processes** collector.

```
Processes
```

Search expression with projected fields

Get the name, SHA-1, and MD5 values for all records returned by the **Processes** collector.

```
Processes name,sha1,md5
```

Search expression with filtered values

Get the name, SHA-1, and MD5 values from the **Processes** collector, for processes files that have the ".exe" extension.

```
Processes name, sha1, md5 where Processes name contains ".exe"
```

Get the name of the processes running on an endpoint with the particular IP address.

```
Processes name where HostInfo ip_address equals 10.112.241.202
```

Search expression with multiple collectors in the projection

Get the name and path of process files that currently spawn more than five threads.

```
Processes name and Files dir where Processes threadCount greater than 5
```

System Tree restrictions to search results

When you run a search expression, not every endpoint on the DXL fabric replies with results. Results come only from those endpoints where your McAfee ePO administrator has granted access to you. For example, suppose that you have access to endpoints in China and don't have access to endpoints in Poland. When you run a search expression, only endpoints in China reply with results.

These access restrictions are set on the **System tree** sections of the **Permission Sets** that apply to your McAfee ePO user.

Use the search box

Write search expressions to navigate results.

Task

1. Select **Menu** → **Systems** → **Active Response Search**.
2. In the **Search** box, enter a search expression.
3. Click **Search** to start collecting data from managed endpoints.

   Note

   If **Search** is disabled, check for errors in the search expression.

   - Click **Cancel** to stop an ongoing search.
   - Click **Save search** to store the search expression in the **Searches** tab of the **Active Response Catalog**.

Get the names and IDs of processes that execute 10 or more threads:

```
Processes name, id where Processes threadCount greater equal than 10
```

Save a search expression

You can save any number of expressions in the **Searches** tab of the **Active Response Catalog**.

Task

1. Select **Menu** → **Systems** → **Active Response Search**.
2. In the **Search** box, type a search expression.
3. Click **Save search**.
4. Enter a name and description for the search expression. This information appears as details in the **Searches** tab of the **Active Response Catalog**.
5. Click **OK**.

What to do next

Option definitions

Section	Option	Definition
Search box The search box provides auto-completion to create search expressions.	**Search**	Click to access the **Searches** catalog and execute a saved search expression.
	Magnifying glass	Executes the search expression, retrieving results from managed endpoints. When the *magnifying glass* turns into an 'x', click it to stop the ongoing search.
	Save	Opens editing options for the saved search expression currently on the search box. • **Save** — Saves the changes made to the saved search expression. • **Save as** — Saves the changes made to the search expression with a new name. • **Discard Changes** — Discards changes made to the saved search expression.

Section	Option	Definition
	Save search	Saves the current search expression on the **Searches** catalog.
Search Results filter	**Quick filter**	Enter a term to filter result rows. Click **Apply** to perform the search.
	Apply	Applies the quick filter search to the result rows.
	Clear	Removes the **Quick filter** that was applied.
Search Results rows	Each header name represents a collector output, selected or implied by the search expression's projection.	• Click a header name once to sort rows by that column, in ascendent order. • Click a header name twice to sort rows by that column, in descendent order. • Click a header name for a third time to undo sorting and return to the default ordering of rows.
	Count	The number of managed endpoints that matched the values in each result row.
Search Results footer	**Search status**	Shows the current state of the search box.
	Systems responding	Shows how many endpoints replied to the search over the number of managed endpoints visible to Active Response.
	Errors	Shows the number of errors produced by the current ongoing search. Click **Errors** to see details.
	Save search	Opens the **Save Search** dialog to save the search expression entered in the search box. Saved search expressions must have a name and, optionally, a description.
Actions	**Show Related Systems**	Shows which managed endpoints relate to selected result rows.
	Execute Reaction	Runs a reaction on the managed endpoints that relate to selected result rows.

Section	Option	Definition
		• **Reaction** — select a reaction from the drop-down list. These reactions are the ones stored in the **Active Response Catalog**. • **Argument** — if the reaction takes arguments, enter values for each argument in each text box. **Attention:** Reaction arguments may be already mapped to collector output fields. These mappings are configured for each reaction in the **Active Response Catalog**.
	Export all	Exports all rows in the **Search Results** table to a CSV file. **Tip:** Remember to encode the export file to UTF-8 with BOM if there are non-ANSI values in the data.

Use a saved search expression

Quickly start an Active Response search from a previously saved search expression.

Before you begin

A search expression must be saved in the **Active Response Catalog** to complete this task.

Task

1. Select **Menu** → **Active Response Catalog** → **Searches**.
2. Click the name of the search expression that you want to run.

Tip

To import, export or delete saved search expressions, use **Actions** in the **Searches** tab of the **Active Response Catalog**.

What to do next

Option definitions

Section	Option	Definition
Filter options	**Show selected rows**	Displays only the rows you selected.
Actions	**Choose Columns**	Opens the **Choose Columns** page to select the columns that are displayed in the catalog table.
	Delete	Removes selected objects from the **Searches** tab. ⚠ **Caution:** This action cannot be undone.
	Export	Exports the selected search expressions to a file, in JSON format. Use this action to back up search expressions or share them with other users.
	Export Table	Allows you to export this table.

Search syntax

Use this detailed example to create powerful, real-time searches.

Get the names and IDs of processes that execute 10 or more threads.

```
Processes name, id where Processes threadcount greater than 10
```

Projection

The projection clause specifies which columns to show in the search results table. This example shows only two columns: process `name` and `id`.

```
Processes name, id
```

Term	Name	Description
`Processes`	Collector name	Specifies the search capabilities and output fields of the specific collector. In the example, the collector for running processes is selected.
`name`, `id`	Collector output fields	Selects an output field from the collector. In the projection, the output field represents a column in the result table.

Filter

The filter clause specifies conditions to match in the returned data. Only data that matches the filter appear in the search results table. In this example, only processes that execute 10 or more threads match the filter.

```
where Processes threadCount greater equal than 10
```

Term	Name	Description
`where`	Filter keyword	The keyword that starts a filter clause.
`Processes`	Collector name	Specifies the search capabilities and output fields of the specific collector. In the example, the collector for running processes is selected.
`threadcount`	Collector output field	Specifies which data must be matched against the condition output field from the collector.
`greater equal than`	Comparison operator	The operator that defines the condition to match. Different operators are available for different literal types.
`10`	Literal	A literal value.

Logical operators

Operator	Used in	Usage	Description

`and`	Projections and filters	`Projection:`
		`Processes name and Files dir`
		
		`Filter:`
		`where Processes name starts with "abc" and Processes threadCount equals 5`
			In a projection, `and` selects output fields from different collectors. In a filter, it displays a result record if both the first condition `and` the second condition are true.

`or`	Filters	`where Processes name starts with "abc" or Processes name starts with "xyz"`	Displays a result record if either the first condition `or` the second condition are true.
`not`	Filters	`where Processes name not starts with "abc"`	Negates a comparison operator, so that the condition returns `true` if the comparison is `false`, or returns `false` if the comparison is `true`.

Comparison operators

Data type	Operator	Usage
Timestamp	`before`	`where Files last_access before "2014-12-31"`
	`after`	`where Files last_access after "2014-12-31"`
Number	`equals`	`where Files size equals 1024`
	`greater than`	`where Files size greater than 1024`
	`greater equal than`	`where Files size greater equal than 1024`
	`less than`	`where Files size less than 1024`

Data type	Operator	Usage
	`less equal than`	`where Files size less equal than 1024`
String **Note:** All string comparisons are case insensitive.	`equals`	`where Files name equals "abc"`
	`contains`	`where Files name contains "abc"`
	`starts with`	`where Files name starts with "abc"`
	`ends with`	`where Files name ends with "abc"`
String **Note:** Search using Perl regular expression syntax.	`matches`	`where Files name matches "^file_[0-9]+.exe$"`
IP **Note:** Filtering by IPv4 omits IPv6 results and, likewise, filtering by IPv6 omits IPv4 results.	`equals`	`where NetworkFlow src_ip equals 10.250.45.15`
	`contains`	`where NetworkFlow src_ip contains 10.250.0.0/24`

Literals

When searching for a path, you must enter an additional \ character in directory paths, for example, `Users\\Administrator\\Documents`. When searching for a value that includes a double quotation mark, use the \ character before the quotation, for example, `Files where File name contains \"`.

Type	Sample values
Timestamp	`"2014"`, `"2014-12"`, `"2014-12-31"`
Number	`123`, `123.45`

Type	Sample values
IP	`10.250.45.15`, `10.250.45.15/24`, `2001:0DB8::1428:57ab`, `2001:0DB8::1428:57ab/96`
String	`"aString123"`, `"This is another string"`, `"quotes\"in\"string"`
Win Registry String	`"My Computer\\HKEY_LOCAL_MACHINE\\HARDWARE\\VIDEO"`, `"0x00000001"`

Collecting endpoint data

Active Response collects real-time data from managed endpoints. Active Response collectors are components that run on managed endpoints, executed by search expressions.

Collectors specify what data to collect from managed endpoints, and how to report it back to Active Response. There are two main types of collectors.

- Built-in — Active Response provides these collectors by default, available after installation.
- Custom — You create these collectors to gather specific data.

Built-in collectors

Autoruns collector

Returns the autorun entries for a Windows managed endpoint.

Collector output

Field	Type	Description
entry_timestamp	String	Last modification time of the autorun entry location. It can be a folder, registry key, or scheduled tasks.
entry_location	String	The entry location of the autorun entry. It can be a folder, registry key, or scheduled tasks.
entry	String	The name of the entry. It can be the name of the shortcut, registry key name, or task full name.
enabled	String	Indicates whether the entry is active or not.

Field	Type	Description
category	String	Specifies the category of the entry. The possible values are: Logon, Explorer, Internet Explorer, Tasks, Codecs, Image Hijack, AppInit Dlls, Known DLLs, Boot Execute, WinLogon, Print Monitors, LSA Providers, Network Providers, Office)
Profile	String	Profile indicates the user mode under which we discovered the entry. The possible values are: Logged-on user name, System-wide.
Description	String	The description member of the version data for the file associated with the entry.
Publisher	String	The publisher member of the version data for the file associated with the entry.
image_path	String	The file path for the file associated with the entry.
Version	String	The version member of the version data for the file associated with the entry.
launch_string	String	The command line with which the file was launched for this entry.

Supported versions

Windows	Linux	macOS
2.4	n/a	n/a

Show autorun entries for Task Scheduler

```
AutoRun where AutoRun entry_location equals "Task Scheduler"
```

CommandLineHistory collector

Returns the command-line history from managed Linux and macOS endpoints.

Collector output

Field	Type	Description
user	String	The user who runs the command.
ID	Number	The incremental execution sequence number (number 1 is the first command executed).
CommandLine	String	The command executed.

Important

The history of the command line and the number depend on the previous configuration available on each endpoint.

Supported versions

Windows	Linux	macOS
n/a	2.0 and later	2.2 and later

Show history of the usage of the service command

```
CommandLineHistory where CommandLineHistory command_line contains "service"
```

CurrentFlow collector

The **CurrentFlow** collector gathers real-time data on the network flow from managed endpoints.

Collector output

Field	Type	Description
local_ip	IPv4 or IPv6 address	IP address of the source of the packet. Supports CIDR block notation.
local_port	Number	Port number originating the packet.
remote_ip	IPv4 or IPv6 address	IP address of the destination of the packet. Supports CIDR block notation.

Field	Type	Description
remote_port	Number	Port number receiving the packet.
status	String	The status of the TCP transaction (not available in UDP transactions).
process_id	Number	The originating process's ID.
user	String	The user that owns the originating process.
user_id	String	The user ID of the process owning the socket.
proto	String	The packet's protocol: TCP or UDP.
md5	String	The MD5 hash code for the source process.
sha1	String	The SHA-1 hash code for the source process.
sha256	String	The SHA-256 hash code for the source process.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show process image names for current flow originating on CIDR block 10.250.45.0/24 and targeting endpoint 10.0.0.2.

```
CurrentFlow process_id where CurrentFlow local_ip contains 10.250.45.0/24 and CurrentFlow remote_ip equals
10.0.0.2
```

DisksAndPartitions collector

Collects information of disks and partitions.

Collector output

Field	Type	Description
disk	String	Numeric index of the physical disk.
model	String	Model of the physical disk.
disk_size	String	Size of the physical disk.
logical_sector	String	Size of the logical sector in bytes. **Note:** On Windows, only NTFS partitions are supported.
physical_sector	String	Size of the physical sector in bytes.
virtual_loc	String	Virtual location of the physical device. (Only for Linux)
disk_flags	String	Flags of the disk. (Only for Linux)
partition	String	Numeric index of the partition on a physical disk.
volume	String	Volume of the partition or location where it is mounted.
partition_size	String	Size of the partition.
partition_freespace	String	Free space available in the partition.
file_system	String	Name of the file system.
type	String	Type of physical device. For example, fixed hard disk media, removable disk media. (Only for Windows)
partition_flags	String	Flags of the partition.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

Show the models of physical disks connected to endpoint "john-pc"

```
DisksAndPartitions model where HostInfo hostname equals "john-pc"
```

DNSCache collector

The **DNSCache** collector shows DNS information on endpoint local cache.

Collector output

Field	Type	Description
hostname	String	The host name.
ipaddress	String	The IP address for the host.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show DNS information for host "ping.alot.com"

```
DNSCache where DNSCache hostname equals "ping.alot.com"
```

EnvironmentVariables collector

This collector returns information about system environment variables, current user, and volatile and process variables.

Collector output

Field	Type	Description
username	String	The owner of the process that is running on the environment where this variable is set.
process_id	Number	ID given by operating system to the process.
name	String	The variable's name.
value	String	Value set on the variable.

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

Show the PATH environment variable set on endpoint 192.168.0.5

```
EnvironmentVariables where EnvironmentVariables name equals "PATH" and HostInfo ip_address equals 192.168.0.5
```

Files collector

The **Files** collector gathers data about managed endpoints' file systems.

Collector output

Field	Type	Description
name	String	The file name.
dir	String	The directory path where the file is located.

Field	Type	Description

		Important: When matching directories with the `equals` operator, a trailing path separator is needed.
		Windows example:
		`dir equals "C:\\Program Files\\"`
		Linux example:
		`dir equals "/bin/"`
		macOS example:
		`dir equals "/bin/"`

full_name	String	The fully qualified file name, including its path.
size	Number	File size in bytes.
last_write	Timestamp	The last time the operating system wrote the file.
md5	String	The file's content, in MD5 format.
sha1	String	The file's content, in SHA-1 format.
sha256	String	The file's content, in SHA-256 format.
created_at	Timestamp	Time stamp when the file was created.
deleted_at	Timestamp	Time stamp when the file was deleted.
status	String	Shows `current` for files that are currently on the file system, or `deleted` for files that were removed from the file system.
create_process_pid	Number	Process ID of the process that created the file.

Field	Type	Description
create_process_sha256	String	SHA-256 hash of the process that created the file.
create_process_full_path	String	Full path of the process that created the file.
modify_process_pid	Number	Process ID of the process that modified the file.
modify_process_sha256	String	SHA-256 hash of the process that modified the file.
modify_process_full_path	String	Full path of the process that modified the file.
delete_process_pid	Number	Process ID of the process that deleted the file.
delete_process_sha256	String	SHA-256 hash of the process that deleted the file.
delete_process_full_path	String	Full path of the process that deleted the file.
create_user_domain	String	Domain name of the user executing the process that created the file.
create_user_name	String	Name of the user executing the process that created the file.
create_user_id	String	ID of the user executing the process that created the file.
modify_user_domain	String	Domain name of the user executing the process that modified the file.
modify_user_name	String	Name of the user executing the process that modified the file.
modify_user_id	String	ID of the user executing the process that modified the file.
delete_user_domain	String	Domain name of the user executing the process that deleted the file.
delete_user_name	String	Name of the user executing the process that deleted the file.
delete_user_id	String	ID of the user executing the process that deleted the file.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show files in the C:\Windows\Boot\DVD\EVE\ path.

```
Files where Files dir equals "c:\\windows\\boot\\dvd\\efi\\"
```

File hashing

To provide information about file systems, Active Response must first complete the file hashing process to record file system metadata in its databases.

Active Response hashes only non-removable file systems.

- On Windows, Active Response hashes only media that return `DRIVE_FIXED` after calling the `GetDriveTypeA` function.
- On Linux, Active Response hashing ignores all paths that return `RM = 1, TYPE = part, MOUNTPOINT != ""` after running the command `lsblk -o RM,TYPE,MOUNTPOINT -r`.
- On macOS, Active Response hashing ignores all paths that return the command `diskutil info -all` and are marked "Removable Drive".

Restrictions

Some restrictions apply to what files are returned by the collector.

- Only endpoints where the user has **System Tree** permissions reply with results.
- Only files that are note excluded by ignore policies appear in search results.
- Depending on the database size limit set on file hashing policies, information about files deleted before the past 30 days might not appear in search results.

HostEntries collector

The **HostEntries** collector shows the IP addresses and host names from hosts file on Windows, Linux, and macOS endpoints.

Collector output

Field	Type	Description
ipaddress	IP	An IP address set in the hosts file.

Field	Type	Description
hostname	String	The host name mapping for the IP address.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Find endpoints whose hosts file configures access to www.malware.com.

```
HostEntries where HostEntries hostname equals "www.malware.com"
```

HostInfo collector

The **HostInfo** collector shows an endpoint's host name, physical IP address, and operating system version.

Collector output

Field	Type	Description
hostname	String	The endpoint's host name.
ip_address	IP	The endpoint's first physical IP address
os	String	The endpoint's operating system version.
connection_status	String	Displays the endpoint's status of **Quarantined** or **Online**.
platform	String	The endpoint's operating system (Windows, macOS, Linux).

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Find all endpoints with Windows operating system.

```
HostInfo where HostInfo os contains "Windows"
```

InstalledCertificates collector

Returns information about installed certificates.

Collector output

Field	Type	Description
issued_to	String	The subject field identifies the entity associated with the public key stored in the subject public key field.
issued_by	String	Identifies the entity that has signed and issued the certificate.
expiration_date	Time stamp	Indicates the expiration date of the certificate.
purposes	String	The key usage extension defines the purpose (for example, encipherment, signature, and certificate signing) of the key obtained in the certificate. The usage restriction might be employed when a key that could be sent for more than one operation is to be restricted.
purposes_extended	String	This extension indicates one or more purposes for which the certified public key might be used, in addition to or in place of the basic purposes indicated in the key usage extension. In general, this extension appears only in end entity certificates. This field is optional. (**Extended Key Usage** on Linux and **Enhanced Key Usage** on Windows).

Field	Type	Description
friendly_name	String	Displays a more friendly name of the certificate. (Only on Windows)

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

Important

The signing certificates are located in specific locations for each operating system. Linux file certificates are ca-bundle.crt and ca-bundle.trust.crl located at /etc/pki/tls/certs. Windows certificates must be registered in the drivers at Certs:. For macOS, only certificates installed on the system or user keychain are shown.

Show the installed certificates issued by Intel

```
where installed_certificates issued_by contains "Intel"
```

InstalledDrivers collector

The **InstalledDrivers** collector shows details about drivers installed on managed endpoints.

Collector output

Field	Type	Description
displayname	String	The display name for the driver.
description	String	A description for the driver.
last_modified_date	Timestamp	A date-time value indicating when the driver was last modified.
name	String	A short name that uniquely identifies the driver.
servicetype	String	The type of service provided to calling processes.

Field	Type	Description
startmode	String	The driver start-up mode. • Boot — the driver is started by the operating system loader. • System — the driver is started by the operating system. • Automatic — the driver starts automatically at system start-up. • Manual — the driver starts by the service control manager. The service control manager also allows to manually start the driver. • Disabled — the driver can no longer be started.
state	String	The current state of the driver.
path	String	The fully qualified path to the driver file.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show drivers which are disabled on endpoints.

```
InstalledDrivers where InstalledDrivers state equals "disabled"
```

InstalledUpdates collector

The **InstalledUpdates** collector gathers data about installed updates, hotfixes, and security updates on Windows endpoints.

Collector output

Field	Type	Description
description	String	The description for the update package.
hotfix_id	String	Microsoft knowledge base identifier for the update package.

Field	Type	Description
install_date	Timestamp	The date when the package was installed.
installed_by	String	The user name that performed the installation, qualified by its namespace.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show which hotfix packages were installed by bad_user.

```
InstalledUpdates where InstalledUpdates description equals "Hotfix" and InstalledUpdates installed_by
contains "bad_user"
```

InteractiveSessions collector

The **InteractiveSessions** collector gathers information about live interactive sessions on endpoint systems.

Collector output

Field	Type	Description
userid	String	The user name that is logged into the session.
name	String	The user's full name. **Note:** This field is not reported for non-local users.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show interactive sessions for user 'owilde'

```
InteractiveSessions where InteractiveSessions userid equals "owilde"
```

Note

On Windows endpoints, information of past sessions may appear in the results if they belonged to accounts from different domains that have the same userid as the currently active one.

LoadedModules collector

Shows the loaded modules of running processes.

You can run a search query to display all loaded modules of a process to investigate and perform actions such as:

- Determine if a process is compromised.
- Reconfigure reputations for a proper process reputation calculation.
- Change the reputation of a loaded module.
- Display modules injected from other processes.

Collector output

Field	Type	Description
process_id	Number	The process's system identifier.
process_name	String	The name of the running process.
process_imagepath	String	Path to the process's image name.
module_name	String	The name of the module.
module_imagepath	String	Path to the module's image name.

Field	Type	Description
module_reputation	String	The module's reputation name and level (range) defined by TIE or ATP. • Known Trusted — [99,100] — This is a trusted file • Most Likely Trusted — [71,85] — Almost certainly a trusted file • Might Be Trusted — [51,70] — Appears to be a benign file • Unknown — [31,50] — Cannot make a determination at this time • Might Be Malicious — [16,30] — Appears to be a suspicious file • Most Likely Malicious — [14,15] — Almost certainly a malicious file • Known Malicious — 1 — This is a malicious file • Not Set — 0 — No reputation has been specified
module_sha1	String	The SHA-1 hash code for the module.
module_sha2	String	The SHA-256 hash code for the module.
module_md5	String	The MD5 hash code for the module.

Supported versions

Windows	Linux	macOS
2.3 and later	2.3 and later	2.3 and later

Show names of modules with process ID of 71, running on host "osx-elcapitan-01".

```
LoadedModules where HostInfo hostname equals "osx-elcapitan-01" and LoadedModules id equals 71
```

LocalGroups collector

The **LocalGroups** collector gathers data on local system groups. Access Directory groups are not returned.

Collector output

Field	Type	Description
groupname	String	The name of the group.
groupdomain	String	The domain name of the local group.
groupdescription	String	The description of the local group.
islocal	String	Confirms that the group is stored locally on the endpoint.
sid	String	The security identifier for the group.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show local groups under the "corp.sensitive" domain.

```
LocalGroups where LocalGroups groupdomain contains "corp.sensitive"
```

LoggedInUsers collector

The **LoggedInUsers** collector gathers data about users logged into managed systems.

Collector output

Field	Type	Description
id	String	The user ID set by the operating system.
userdomain	String	The domain to which the user belongs.

Field	Type	Description
username	String	The logon user name.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show users logged under the "RISK" domain

```
LoggedInUsers where LoggedInUsers userdomain equals "RISK"
```

NetworkFlow collector

The **NetworkFlow** collector gathers historical data on network usage from managed endpoints.

Collector output

Field	Type	Description
src_ip	IPv4 or IPv6 address	IP address of the source of the packet. Supports CIDR block notation.
src_port	Number	Port number originating the packet.
dst_ip	IPv4 or IPv6 address	IP address of the destination of the packet. Supports CIDR block notation.
dst_port	Number	Port number receiving the packet.
time	Date	Date and time when the packet was collected.
status	String	The status of the TCP transaction (not available in UDP transactions).

Field	Type	Description
		Important: The TCP status must be interpreted as follows: • On a TCP connection open operation, the `CONNECTED` value means that the source endpoint sent a `SYN` message and received an `ACK`, `SYN` message from the remote server. • On a TCP connection close operation, the `CLOSED` value means that the source endpoint sent a `SYN` message and received an `ACK`, `FIN` message from the destination server. • The final `ACK` message is ignored on both open and close operations.
process	String	The originating process image name.
process_id	Number	The originating process ID.
user	String	The user that owns the originating process.
user_id	String	The user ID of the process owning the socket.
proto	String	The packet's protocol: TCP or UDP.
direction	String	Specifies whether the packet came `in` to the managed endpoint, or was sent `out` of the endpoint.
ip_class	Number	Specifies the IP class used for the transaction: • IPv4 returns `0` • IPv6 returns `1` • Unknown returns `2`
seq_number	Number	TCP transaction sequence number (not available in UDP transactions).
src_mac	String	MAC address of originating endpoint.
dst_mac	String	MAC address of destination endpoint (Linux only).
md5	String	The MD5 hash code for the source process.

Field	Type	Description
sha1	String	The SHA-1 hash code for the source process.
sha256	String	The SHA-256 hash code for the source process. (Only for macOS and Windows)

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show process IDs and image names for network flow originating on CIDR block 10.250.45.0/24 and targeting endpoint 10.0.0.2.

```
NetworkFlow process, process_id where NetworkFlow src_ip contains 10.250.45.0/24 and NetworkFlow dst_ip
equals 10.0.0.2
```

NetworkInterfaces collector

The **NetworkInterfaces** collector lists network interfaces on managed endpoints.

Collector output

Field	Type	Description
bssid	String	The BSSID to which the interface is connected.
displayname	String	The interface's short name on the operating system.
gwipaddress	IP	The IP address of the gateway to which the interface is connected.
gwmacaddress	String	The MAC address of the gateway to which the interface is connected.
ipaddress	IP	The interface's IP address.
ipprefix	Number	The IP prefix for the interface's IP address.

Field	Type	Description
macaddress	String	The interface's MAC address.
name	String	The interface's name.
ssid	String	The SSID to which the interface is connected.
type	String	The interface's type.
wifisecurity	String	The WiFi security algorithm used by the interface on the current connection.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

NetworkSessions collector

The **NetworkSessions** collector gets information of currently open network sessions on the endpoint.

Collector output

Field	Type	Description
computer	String	IP or host name of remote endpoint.
user	String	User logged on to host through the network session.
client	String	Remote session command provider. (Only on Windows.)
file	String	Path of local resource being accessed by client. (Only on Windows.)
idletime	String	Time since last session activity. (Only on Windows.)

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	n/a

Show which shared resources are being accessed by user name "owilde"

```
NetworkSessions where NetworkSessions user equals "owilde"
```

NetworkShares collector

The **NetworkShares** collector finds network shared paths accessible from each managed endpoint.

Collector output

Field	Type	Description
name	String	Name of shared resource.
description	String	Description of shared resource set either by the user or by default.
path	String	Local path to the resource.

Important

When Samba service is started, only resources configured at /etc/samba/smb.conf are returned by the collector. It obtains information of the Network File System (NFS) from file /etc/samba/smb.conf.

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

Show which paths on endpoint "owilde-office" are being shared

```
NetworkShares path where HostEntries hostname equals "owilde-office"
```

ProcessHistory collector

The **ProcessHistory** collector displays the status, create time, and terminated time of any running or terminated processes.

Note

The **ProcessHistory** collector does not retain the collected information between reboots and can store only up to 2000 terminated process events. When the number of running process events exceed the limit of 2000, the stored terminated process events are purged in order to store the current running process events.

Collector output

Field	Type	Description
name	String	The name of the running process.
id	Number	The process system identifier.
threadcount	Number	The number of active threads spawned by the process.
parentid	Number	The system identifier for the process that spawned the current process.
parentname	String	The name of the process that spawned the current process.
parentimagepath	String	The full path of the parent process.
file_reputation	String	The process reputation's name and level (range) defined by TIE or ATP. • Known Trusted — [99,100] — This is a trusted file • Most Likely Trusted — [71,85] — Almost certainly a trusted file • Might Be Trusted — [51,70] — Appears to be a benign file • Unknown — [31,50] — Cannot make a determination at this time • Might Be Malicious — [16,30] — Appears to be a suspicious file • Most Likely Malicious — [14,15] — Almost certainly a malicious file • Known Malicious — 1 — This is a malicious file • Not Set — 0 — No reputation has been specified

Field	Type	Description
process_reputation	String	The reputation of a running process. See **file_reputation** for range of values.
started_at	Timestamp	Time when the process started.
finished_at	Timestamp	Time when the process terminated.
content_size	Number	If the process is a PowerShell, this is the size of the script being executed.
content	String	A piece of the script; if the script is larger than 8 k, it is truncated.
content_file	String	The full path of the script if it was in a file and the PowerShell was executed with `-file` parameter (if interactive, it might include the first file read by the interpreter).
execution_mode	String	The mode that the PowerShell was executed: • Interactive — No file was introduced and the user is interacting with the console • Unknown — It was not known how it was executed • File — With the `-file` parameter and a file on it (File based execution) • Commandline — When the command is placed in the command line with the interpreter (Fileless) • Mar_child — This is a PowerShell instance launched by Active Response to execute a collector
size	Number	The amount of resident RAM used by the process.
md5	String	The MD5 hash code for the process.
sha1	String	The SHA-1 hash code for the process.
sha256	String	The SHA-256 hash code for the process.
cmdline	String	The command that started the process.
imagepath	String	Path to the process image name.

Field	Type	Description
kerneltime	Number	The process's use of kernel mode CPU time, in seconds.
usertime	Number	The process's use of user mode CPU time, in seconds.
uptime	Number	The number of seconds passed since the process started.
user	String	The user name that started the process.
user_id	String	The ID for the user that started the process.
normalized_cmdline	String	The result of using a Windows API for getting command line arguments in a standard format. This API has a special interpretation of backslash character and double quotation marks. • Ex: `-C"a"r` is a valid argument and is equal to `-Car/` • More information: https://blogs.msdn.microsoft.com/oldnewthing/20100917-00/?p=12833

Note: Content_size, content, and content_file are only available if Endpoint Security 10.6 is installed and the Antimalware Scan Interface is enabled.

Supported versions

Windows	Linux	macOS
2.3 and later	n/a	n/a

Show running and terminated processes with "powershell" in their name with their size, content, location, execution mode, and count.

```
ProcessHistory status, content_size, content, content_file, execution_mode where ProcessHistory name
contains "powershell"
```

Processes collector

The **Processes** collector gathers data on processes running on managed endpoints.

Collector output

Field	Type	Description
name	String	The name of the running process.
id	Number	The process' system identifier.
threadcount	Number	The number of active threads spawned by the process.
parentid	Number	The system identifier for the process that spawned the current process.
parentname	String	The name of the process that spawned the current process.
parentimagepath	String	The full path of the parent process.
parent_cmdline	String	The command that started the parent process.
file_reputation (reputation renamed file_reputation)	String	The process reputation's name and level (range) defined by TIE or ATP. • Known Trusted — [99,100] — This is a trusted file • Most Likely Trusted — [71,85] — Almost certainly a trusted file • Might Be Trusted — [51,70] — Appears to be a benign file • Unknown — [31,50] — Cannot make a determination at this time • Might Be Malicious — [16,30] — Appears to be a suspicious file • Most Likely Malicious — [14,15] — Almost certainly a malicious file • Known Malicious — 1 — This is a malicious file • Not Set — 0 — No reputation has been specified **Note:** When using an Active Response 2.3 server and Active Response 2.2 clients, the search will return **file_reputation** as `Not Set` and no error will be generated. We recommend you upgrade the clients to version 2.3 to see the output results.

Field	Type	Description
process_reputation	String	The reputation of a running process. See **file_reputation** for range of values.
started_at	Timestamp	Time when the process started.
finished_at	Timestamp	Time when the process terminated.
content_size	Number	If the process is a PowerShell, this is the size of the script being executed.
content	String	A piece of the script; if the script is larger than 8 k, it is truncated.
content_file	String	The full path of the script if it was in a file and the PowerShell was executed with `-file` parameter (if interactive, it may include the first file read by the interpreter).
execution_mode	String	The mode that the PowerShell was executed: • Interactive — No file was introduced and the user is interacting with the console • Unknown — It was not known how it was executed • File — With the `-file` parameter and a file on it (File based execution) • Commandline — When the command is placed in the command line with the interpreter (Fileless) • Mar_child — This is a PowerShell instance launched by Active Response to execute a collector
size	Number	The amount of resident RAM used by the process.
md5	String	The MD5 hash code for the process.
sha1	String	The SHA-1 hash code for the process.
sha256	String	The SHA-256 hash code for the process.
cmdline	String	The command that started the process.
imagepath	String	Path to the process image name.

Field	Type	Description
kerneltime	Number	The process's use of kernel mode CPU time, in seconds.
usertime	Number	The process's use of user mode CPU time, in seconds.
uptime	Number	The number of seconds passed since the process started.
user	String	The user name that started the process.
user_id	String	The ID for the user that started the process.
normalized_cmdline	String	The result of using a Windows API for getting command line arguments in a standard format. This API has a special interpretation of backslash character and double quotation marks. • Ex: `-C"a"r` is a valid argument and is equal to `-Car/` • More information: https://blogs.msdn.microsoft.com/oldnewthing/20100917-00/?p=12833 For Linux and macOS, it is the same as **cmdline**.

Note: Content_size, content, and content_file are only available if Endpoint Security 10.6 is installed and the Antimalware Scan Interface is enabled.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show processes' names and RAM size for processes that use more than 10 MB of resident RAM.

```
Processes name, size where Processes size greater than 10240
```

Show processes' process reputation on a host with IP address of 10.9.9.9

```
Processes id, name, reputation, processreputation where HostInfo ip_address equals 10.9.9.9
```

ScheduledTasks collector

The **ScheduledTasks** collector shows the status of scheduled tasks on endpoints, and also when it is scheduled to run next.

Collector output

Field	Type	Description
folder	String	The path from where the scheduled task runs. (Empty in Linux)
taskname	String	Name of task.
nextruntime	Date	Time and date when the task will run.
status	String	Current task status can be ready, disabled, setting, running, or could not start.
task_run	String	Full command line to execute tasks.
last_run	Date	Last time the task ran successfully.
username	String	Name of the user that executed the task.
schedule_on	String	See Trigger field documentation.
log_on_type	String	Security logon method required to run tasks. See Log on Type documentation. (Only for Windows)

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

Show when will the task called 'backupDaily' run next

```
ScheduledTasks taskname, nextruntime where ScheduledTasks taskname equals "backupDaily"
```

Services collector

The **Services** collector lists services installed on managed endpoints.

Collector output

Field	Type	Description
description	String	A description of the service's functionality.
name	String	A short name that uniquely identifies the service.
startuptype	String	The start-up mode. • Boot — specifies a device driver started by the operating system loader. • System — specifies a device driver started by the operating system. • Automatic — specifies a service that starts automatically at system start-up. • Manual — specifies a service started by the service control manager. • Disabled — specifies a service that can no longer be started.
status	String	The current status of the service.
user	String	The user that owns the service's process.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show services that are currently running and are set to start manually by users.

```
Services where Services status equals "Running" and Services startuptype equals "Manually"
```

Software collector

The **Software** collector lists software installed on managed endpoints.

Collector output

Field	Type	Description
displayname	String	Commonly used software name.
installdate	Timestamp	A date-time value indicating when the object was installed.
publisher	String	The name of the software's supplier.
version	String	Software version information.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show installed software provided by 'Bad Co.' publisher

```
Software where Software publisher equals "Bad Co."
```

Startup collector

The **Startup** collector shows information about start-up applications on managed endpoints.

Collector output

Field	Type	Description
caption	String	The short name set by the application.
command	String	The command line that starts the application.
description	String	The description set by the application.
name	String	The application's file name.
user	String	The user name for whom this start-up command will run.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Show applications that start up automatically for user 'owilde'

```
Startup where Startup user equals "owilde"
```

UsbConnectedStorageDevices collector

Find which users have used USB mass storage devices on managed endpoints. This collector gets details on last usage and device details.

Collector output

Field	Type	Description
vendor_id	String	Device's vendor ID.
product_id	String	Device's product ID.
serial_number	String	Device's serial number.
device_type	String	Only "USB storage" type is supported.
guid	String	ID provided by operating system. (Only on Windows)
last_connection_time	Date	Last time the device was plugged. (Only on Windows)
user_name	String	User that mounted the device. If no user was logged in when device was mounted, then the field will be empty. (Only on Windows)
last_time_used_by_user	Date	Last time the operating system touched the device.

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

Show all USB storage devices that were connected to computers with running Windows

```
UsbConnectedStorageDevices where HostInfo os contains "win"
```

UserProfiles collector

The **UserProfiles** collector gathers data about local users on Windows endpoints.

Collector output

Field	Type	Description
accountdisabled	String	`True` if the account is disabled. `False` otherwise. (1)
domain	String	The domain that holds the user. *
fullname	String	The user's full name. *
installdate	Timestamp	The creation date for the user's home folder (C:\Users\user-name). The user must log in at least once for this date to be returned.
localaccount	String	`True` if the user is stored locally on the endpoint. `False` otherwise.
lockedout	String	`True` if the user has been locked out from the endpoint. `False` otherwise. *
accountname	String	The user's account name.
sid	String	The security identifier for the user.
passwordexpires	String	`True` if the password is configured to expire. `False` otherwise. *

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

Find user accounts that have been locked out from endpoints.

```
UserProfiles where UserProfiles lockedout equals "true"
```

1 This field is not returned for non-local users.

WinRegistry collector

The **WinRegistry** collector gathers Windows registry data from endpoints.

Collector output

Field	Type	Description
keypath	Win Registry String	A path to a registry key. The path does not include the key name. Only `equals` and `starts_with` operators are valid for this output field.
keyvalue	Win Registry String	The key value name.
valuedata	Win Registry String	The data stored by the key value.
valuetype	Win Registry String	The data type of the registry data.

Supported versions

Windows	Linux	macOS
1.1 and later	n/a	n/a

Show registry data related to Active Response installation on managed endpoints.

```
WinRegistry where WinRegistry keypath equals "hkey_local_machine\\software\\mcafee\\mar"
```

Strings in conditions and filters are case insensitive: "software" and "SOFTWARE" match the same registry entries.

Custom collectors

Custom collectors use the output of content execution to gather specific data from managed endpoints.

The collector parses content output as records of comma-separated values data. Then, it matches the fields in the records to the output fields defined for the collector, in order of appearance.

If a collector's content executes the following lines:

```
echo "value1","value2"
echo "value3","value4"
```

Active Response maps `"value1"` and `"value3"` to the first output field, and `"value2"` and `"value4"` to the second output field, like this:

Output field 1	Output field 2
value1	value2
value3	value4

Create a custom collector

Specify what data to collect from endpoints with custom collectors.

Task

1. Select **Menu** → **Systems** → **Active Response Catalog**.
2. Select the **Collectors** tab, then click **New Collector**.
3. Enter a name and description for the collector.
4. For **Windows**, **Linux**, **macOS** tabs, insert the collector's content.
   a. Use the **Type** drop-down list to select the appropriate content type.
   b. In the **Content** code editor, enter the commands or code that Active Response executes on managed endpoints.

   Note

   Add content to **Windows**, **Linux**, and **macOS** tabs to run the collector on Windows, Linux, and macOS managed endpoints.

5. Click **Add Output** or **+** to add an output field.
6. Enter a name for the field.
7. From the **Type** drop-down list, select a type for the field's data.
8. Select **Show by default** to make the output field a default field in the **Search results** table.
9. Set the **Collector Timeout** to increase or decrease the default 60-second timeout limit. Increase the timeout limit for collectors that need more time to run.
10. Click **Save** to finish.

    Tip

    If **Save** is disabled, check for problems in the form fields.

Option definitions

Section	Option	Definition
Common actions	**Edit**	Enters edit mode. This is only available for custom collectors.
	Save	Saves changes made to collector details.
	Close	Restores changes made to collector details and exits.
Collector Summary	**Name**	Use this box to set the collector name. **Note:** You cannot change a collector's name if it is used by a saved search expression.
	Description	Use this box to set the collector description. Give meaningful names and descriptions to collectors, based on the domain of the collected data, to easily find them in the **Active Response Catalog**.
Collector Content	**Windows** tab	Use this tab to define the collector's content for endpoints running Windows.
	Linux tab	Use this tab to define the collector's content for endpoints running Linux.
	macOS tab	Use this tab to define the collector's content for endpoints running macOS.
	Type	Selects a supported content type.
	Convert collector output to UTF-8 encoding	When selected, Active Response encodes all collector output in UTF-8.
	Content	Use this code box to set the collector's content.
Collector Output	**Add Output**	Adds an output field for the collector.

Section	Option	Definition
	Name	Sets a name to the output field. This is the name of the column in the **Search Results** table.
	Type	Sets a type to the values in the output field for validation.
	Show by default	If selected, the output field appears in the search results table when no output fields are specified in the search expression.
	+	Adds an output field.
	-	Deletes an output field.
Collector Timeout	**Timeout [sec]**	Increase or decrease the timeout limit. Default is 60 seconds.
Actions	**Choose Columns**	Select and configure the columns to display in the collector's tab.
	Delete	Removes the collector from the **Active Response Catalog**. **Caution:** This action cannot be undone.
	Export	Exports the current collector to a file, in JSON format. Use this action to back up objects or share them with other users.
	Export Table	Exports the collector's catalog in compressed and various file formats.

Reacting to incidents

Active Response acts on managed endpoints by executing reaction code.

Reaction summary

A reaction specifies an action to take on managed endpoints. A name and description identify the reaction. Give meaningful names and descriptions to reactions based on what effect each reaction produces so you can easily find them in the **Active Response Catalog**.

Reaction content

A reaction's content specifies the code that Active Response executes on managed endpoints.

Reaction arguments

A reaction's content supports named arguments to pass values during execution.

These fields define an argument:

- **Name** — Specifies the argument's handle
- **Type** — Specifies a data type for the argument.

Argument mappings

Reaction arguments are related to trigger and collector output fields.

When a trigger is set to run a reaction, the trigger output fields are passed as values to reaction arguments. So if a trigger returns a filename as output, this filename can be passed as a value in a reaction argument that expects a filename.

Also, you can map arguments to collector output fields. After running a search expression, you can execute a reaction on endpoints related to **Search Results**. If the reaction arguments are mapped to collector output fields used in the search expression, then Active Response knows which values to pass as arguments during reaction execution.

System Tree restrictions when applying reactions

When you apply a reaction, not every endpoint on the DXL fabric is affected. Only those endpoints where your McAfee ePO administrator has granted access to you are affected by the reaction. For example, if you have access to endpoints in China and don't have access to endpoints in Poland. When you execute a reaction, only endpoints in China are affected.

These access restrictions are set on the **System tree** sections of the **Permission Sets** that apply to your McAfee ePO user.

Built-in reactions

DeleteFolder reaction

Use this reaction to delete a selected folder or folders on the system.

Arguments

Name	Type	Description
full_path	String	The full, qualified file name, including its path.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

DeleteRegistryValue reaction

Deletes a Windows Registry value in a specified registry key path.

Attention

This reaction can only delete key values that are not protected by other software.

Arguments

Name	Type	Description
keypath	Win Registry String	The absolute path to a registry key. The path does not include the key value name.
keyvalue	Win Registry String	The key value name to erase.

Supported versions

Windows	Linux	macOS
2.0 and later	n/a	n/a

DumpProcesstoFile reaction

Use this reaction to generate a memory dump of a given process into a file.

Arguments

Name	Type	Description
pid	number	The process ID.

Supported versions

Windows	Linux	macOS
2.4	n/a	n/a

Execute Reboot OS reaction

Reboots the endpoint's operating system without warning the user.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

Execute Shutdown OS reaction

Shuts down the operating system.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

Execute User Logoff reaction

Use this reaction to log off the user by user name.

Arguments

Name	Type	Description
username	String	The user to be logged off.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

KillProcess reaction

Use this reaction to kill processes on endpoints by passing the process ID.

Arguments

Name	Type	Description
pid	Number	The process ID, set by the operating system.

Supported versions

Windows	Linux	macOS
1.1 and later	1.1 and later	2.2 and later

KillProcessByHash reaction

Use this reaction to kill processes that have a specific hash value on endpoints.

Note

If the target endpoint is offline when this reaction is executed, the reaction is saved on the Active Response server and executes when the endpoint is back online. If a specific file cannot be deleted because a process blocks it, the file is deleted when the endpoint reboots.

Arguments

Name	Type	Description
MD5	String	The process's MD5 value.

Name	Type	Description
SHA1	String	The process's SHA-1 value.
SHA256	String	The process's SHA-256 value.
pid	Number	The process ID.

Supported versions

Windows	Linux	macOS
2.1 and later	n/a	2.2 and later

KillProcessByName reaction

Use this reaction to kill a process by its name.

Arguments

Name	Type	Description
name	String	The name of the process.

Supported versions

Windows	Linux	macOS
2.1 and later	n/a	2.2 and later

KillProcessByPath reaction

Use this reaction to kill a process by its path.

Arguments

Name	Type	Description
full_name	String	The process's full path.
pid	Number	The process ID.

Supported versions

Windows	Linux	macOS
2.1 and later	n/a	2.2 and later

KillProcessTree reaction

Use this reaction to kill a process and its subprocesses.

Arguments

Name	Type	Description
pid	Number	The process ID, set by the operating system.

Supported versions

Windows	Linux	macOS
2.1 and later	n/a	2.2 and later

QuarantineHost / UnquarantineHost reaction

Starts and ends the quarantine of the host from the network while retaining connectivity to McAfee products.

Supported versions

Windows	Linux	macOS
2.3 and later	n/a	2.3 and later

RemoveFile reaction

Use this reaction to delete files from endpoint filesystems.

Note

If the target endpoint is offline when this reaction is executed, the reaction is saved on the Active Response server and executes when the endpoint is back online. If a specific file cannot be deleted because a process blocks it, the file is deleted when the endpoint reboots.

Arguments

Name	Type	Description
full_name	String	The fully qualified file name, including its path.

Supported versions

Windows	Linux	macOS
2.0 and later	2.0 and later	2.2 and later

RemoveFileSafe reaction

This reaction performs a Validation and Trust Protection (VTP) check on endpoints before removing a file.

This feature is for Active Response 2.1 and later clients only.

Note

If the target endpoint is offline when this reaction is executed, the reaction is saved on the Active Response server and executes when the endpoint is back online. If a specific file cannot be deleted because a process blocks it, the file is deleted when the endpoint reboots.

Arguments

Name	Type	Description
full_name	String	The fully qualified file name, including its path.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

ScheduleReboot reaction

Use this reaction to schedule a system reboot for a certain day and time.

Arguments

Name	Type	Description
reboot_time	Timestamp	The specified time for a system scheduled reboot.

Supported versions

Windows	Linux	macOS
2.1 and later	2.1 and later	2.2 and later

StopAndRemoveContent reaction

Terminate the interpreter instance running the script and remove the script without removing the interpreter.

Arguments

Name	Type	Description
pid	Number	The process ID.
full_name	String	The process's full path.

Supported versions

Windows	Linux	macOS
2.3 and later	n/a	2.3 and later

Create a custom reaction (Windows, Linux, macOS)

Reactions execute custom content on managed endpoints.

Task

1. Select **Menu** → **Systems** → **Active Response Catalog**.
2. Select the **Reactions** tab, then click **New Reaction**.
3. Enter a name and description for the reaction.
4. Enter the reaction's content.
   a. Use the **Type** drop-down list to select the appropriate content type.
   b. In the **Content** code editor, enter the commands or code that Active Response executes on managed endpoints.

   Note

   Add content to **Windows, Linux**, and **macOS** tabs so that the reaction applies to Windows, Linux, macOS managed endpoints.

5. Click **Add Argument** or **+** to add an argument.
   a. Enter a name for the argument.

      Important

      An argument's name must match the name given in the reaction's content between `{{` and `}}`.

   b. From the **Type** drop-down list, select a type for the argument values.
   c. Click **Set Collector Mapping** to map the reaction argument to specific collector output fields.

6. Set the **Reaction Timeout** to increase or decrease the default 60-second timeout limit. Increase the timeout limit for collectors that need more time to run.
7. Click **Save** to finish.

Note

If **Save** is disabled, check for problems in the form fields.

Option definitions

Section	Option	Definition
Common actions	**Edit**	Enters edit mode.
	Save	Saves changes made to reaction details.
	Close	Restores changes made to reaction details and exits.
	New Reaction	Creates a reaction.
	Import	Imports reactions into **Active Response Catalog**. Use this action to restore reactions from a backup file.
Reaction Summary	**Name**	Sets the reaction name. **Note:** You cannot change a reaction name if it is used by a trigger.
	Description	Sets the reaction description.
Reaction Content	**Windows** tab	Defines the reaction content for endpoints running Windows.
	Linux tab	Defines the reaction content for endpoints running Linux.
	macOS tab	Defines the reaction content for endpoints running macOS.
	Type	Selects a supported content type.

Section	Option	Definition
	Content	Sets the reaction content.
Reaction Arguments	**Add Argument**	Adds an argument for the reaction. After clicking **Add Argument**, use the empty text box to set a name and the drop-down list to set a type for the argument.
	+	Adds an argument.
	-	Deletes an argument.
	Set Collector Mapping	Sets a mapping from the argument to a collector's output field.
	View Collector Mapping	Shows the current argument mapping.
	Change collector Mapping	Changes the current argument mapping.
Reaction Timeout	**Timeout [sec]**	Increase or decrease the timeout limit. Default is 60 seconds.
Actions	**Choose Columns**	Opens the **Choose Columns** page allowing you to select the columns that are displayed in the catalog table.
	Delete	Removes the reaction from the **Active Response Catalog**. ⚠ **Caution:** This action cannot be undone.
	Export	Exports the current collector to a file, in JSON format. Use this action to back up objects or share them with other users.
	Export Table	Exports the complete table.

Section	Option	Definition
	Show Related Triggers	Opens the **Triggers** catalog, showing only triggers that execute the current reaction.

Apply a reaction

Execute reactions from the **Search Results** table.

Caution

Reactions applied on endpoints cannot be undone. Continue with care.

Task

1. Select **Menu** → **Systems** → **Active Response Search**, then run a search expression.
2. When results appear in the **Search Results** table, select the rows you want to target.

   Important

   Remember that a single row might reference more than one managed endpoint, expressed in the **count** column. In that case, the reaction is applied to all endpoints referenced by the row.

3. Click **Actions** → **Apply Reaction**.
4. Select a reaction from the drop-down list. If the reaction takes arguments, insert values for each argument.

   Note

   Some arguments might be mapped to the collector output fields used in the search expression. The values returned by such output fields are passed to the mapped arguments.

5. Click **Yes** to confirm.

What to do next

When you execute a reaction with Active Response, an Event ID entry (36625) is added to the McAfee ePO **Threat Event Log**. From **Menu** → **Reporting** → **Threat Event Log**, select the event and click **Show Target Systems** to learn more details about the endpoint.

Catching threats

Active Response triggers track system activity to detect possible threats. They can be set to catch specific events on managed endpoints and react immediately.

Based on Active Response data collection capabilities, triggers catch events in managed endpoints and execute reactions.

Trigger summary and configuration

A name and description identify a trigger. Triggers can be *enabled* or *disabled*.

- Enabled triggers are set and active on managed endpoints, listening to events. Even if the endpoint goes offline, the trigger is still enabled and operational.
- Disabled triggers are stored in the **Triggers** catalog for future use, but do not listen to events on managed endpoints.

Also, triggers select an **Event Severity**. This is the level of urgency that is reported in the McAfee ePO **Threat Event Log** when the trigger is fired.

Detection

A trigger's detection settings specify what fires the trigger. Triggers have a type. Each trigger type listens to different events and returns different output fields. For example, the `Files` trigger type listens to `Created`, `Modified`, and `Deleted` events on files. It returns the file's `name`, `size`, `last_access`, `md5`, and `sha1`.

Optionally, triggers can specify a condition that must be met for the trigger to be fired. For example, a `Files` type trigger can be set to catch `Modified` events only in files with a specific name or size.

Reaction

When a trigger fires, it can execute a reaction. The reaction is selected from the **Reactions** catalog.

If the reaction takes arguments, they can be matched to the trigger type's output fields. This matching means that when the trigger fires, its output passes as arguments to the reaction. For example, a reaction that deletes files can take the file name to delete as an argument. When the trigger catches an event in a file, it can pass the file name to the reaction, and that particular file is deleted.

System Tree restrictions to setting triggers

When you enable a trigger, it is not set on every endpoint of the DXL fabric. You can enable triggers only on those endpoints where the McAfee ePO administrator has granted you access. For example, if you have access to endpoints in China and don't have access to endpoints in Poland. When you run a search expression, only those endpoints from China will reply.

Also, only users that have access to the same endpoints that you have can modify your triggers on those endpoints. Users who don't have access to an endpoint where you have set a trigger can't modify your trigger.

These access restrictions are set on the **System tree** sections of the **Permission Sets** that apply to your McAfee ePO user.

Create a trigger

Triggers are set on managed endpoints to catch and react to specific events. Active Response provides different trigger types to catch events on managed endpoints.

Option definitions

Section	Option	Definition
Common actions	**New Trigger**	Creates a trigger.
	Import	Imports triggers into **Active Response Catalog**. Use this action to restore triggers from a backup file.
	Edit	Enters edit mode.
	Save	Saves changes made to trigger details.
	Close	Restores changes made to trigger details and exits.
Filter options	**Show selected rows**	Displays only the rows you selected.
Actions	**Choose Columns**	Opens the **Choose Columns** page allowing you to select the columns that are displayed in the catalog table.
	Delete	Removes selected triggers from **Active Response Catalog**. ⚠ **Caution:** This action cannot be undone.
	Disable	Disables the selected triggers.
	Enable	Enables the selected triggers.
	Export	Exports the selected triggers to a file, in JSON format. Use this action to back up triggers or share them with other users.
	Export Table	Exports the complete table.
Trigger Summary	**Name**	Use this box to set the trigger name.

Section	Option	Definition
	Description	Use this box to set the trigger description.
Trigger Configuration	**Status**	• Select **Enabled** so that the trigger is active in managed endpoints. • Select **Disabled** so that the trigger is active in managed endpoints.
	Event Severity	Selects the severity level that appears in McAfee ePO **Threat Event Log** when the trigger is activated on an endpoint.
Detection	**Type**	Selects one of the supported trigger types.
	Event	Selects the event that this trigger catches.
	Trigger outputs	Shows the output fields supported by the trigger type.
	Condition	Sets a condition that must be met to fire the trigger. **Note:** The condition expression uses the same syntax as the filters in search expressions.
Reaction	**Name**	Selects the reaction that is executed when the trigger fires.
	Arguments	Maps each reaction argument to a trigger output field. **Note:** When the trigger is fired, the values in the mapped output fields are passed to the reaction as arguments.

Task

1. Select **Menu** → **Systems** → **Active Response Catalog**.
2. Select the **Triggers** tab, then click **New Trigger**.
3. Enter a name and description for the trigger.
4. Set the status to **Enabled** if you want the trigger immediately set on managed endpoints. Else, set it to **Disabled**.

5. From the **Trigger Type** drop-down list, select a type for the trigger.
6. From the **Event** drop-down list, select the event to catch.
7. In the **Condition** text box, enter a condition to meet when catching events.
8. From the **Reaction Name** drop-down list, select a reaction.

   Important

   Be careful that the reaction you select doesn't re-create the condition that sets off the trigger. An infinite loop happens if your trigger sets off, it executes a reaction which in turn sets your trigger off again, and so on.

9. In the **Arguments** table, use the drop-down lists in the **Trigger Output** column to map output fields to reaction arguments.
10. Click **Save** to finish.

    Note

    If **Save** is disabled, check for problems in the form fields.

Files trigger

The **Files** trigger listens to events on managed endpoints' file systems.

Events

Event	Description
FileCreated	A matching file is created on a target endpoint.
FileModified	A matching file is changed on a target endpoint.
FileDeleted	A matching file is deleted on a target endpoint.

Output fields

Field	Type	Description
name	String	The file name.
dir	String	The directory path where the file is located.

Field	Type	Description

		Important: When matching directories with the `equals` operator, a trailing path separator is needed.
		Windows example:
		`dir equals "C:\\Program Files\\"`
		Linux example:
		`dir equals "/bin/"`
		macOS example:
		`dir equals "/bin/"`

full_name	String	The fully qualified file name, including path.
size	Number	File size in bytes.
last_write	Date	The last time the operating system wrote the file.
md5	String	The file's content, in MD5 format.
sha1	String	The file's content, in SHA-1 format.
created_at	Date	Time stamp when the file was created.
deleted_at	Date	Time stamp when the file was deleted.
status	String	Shows `current` for files that are currently on the file system, or `deleted` for files that were removed from the file system.

Match *.exe files with SHA-1 hash `97eb5a5b721e28f9696729d14ef9d4076c9b4e2e`

```
name ends with '.exe' and sha1 equals '97eb5a5b721e28f9696729d14ef9d4076c9b4e2e'
```

Tip

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name.

File creation and hashing timing

When a file is created on a managed endpoint, Active Response starts hashing the file and fires the **FileCreated** event. But if the file is large enough, the event might be caught before the hashing process finishes. In this situation, an incomplete MD5 or SHA-1 hash of the file is reported with the event.

Triggers set to catch files over **FileCreated** events based on an MD5 or SHA-1 hash can fail under this condition: when a file large is created, Active Response reports an incomplete file hash. Because the trigger condition is set to match the file hash, this trigger is not executed.

However, when the hashing process finishes, the complete file hash is created. Then, a **FileModfied** event is caught, reporting the complete hash. To avoid this condition, you are encouraged to create two triggers: one for the **FileCreated** event and another one for the **FileModfied** event. Set both triggers to match the complete file hash.

Network trigger

The **Network** trigger listens to events on network flow to or from managed endpoints.

Connection events

McAfee Active Response catches these events on Windows and Linux systems.

Event	Description
ConnectionOpen	A connection is opened.
ConnectionClose	A connection is closed.

Connection output fields

Field	Type	Description
src_ip	IPv4 or IPv6 address	IP address of the source of the packet. Supports CIDR block notation.
src_port	Number	Port number originating the packet.
dst_ip	IPv4 or IPv6 address	IP address of the destination of the packet. Supports CIDR block notation.
dst_port	Number	Port number receiving the packet.
time	Date	Date and time when the packet was collected.
status	String	The status of the TCP transaction. (Not available in UDP transactions.) **Important:** The TCP status must be interpreted as follows: • On a TCP connection open operation, the `CONNECTED` value means that the source endpoint sent a `SYN` message and received an `ACK`, `SYN` message from the remote server. • On a TCP connection close operation, the `CLOSED` value means that the source endpoint sent a `SYN` message and received an `ACK`, `FIN` message from the destination server. • The final `ACK` message is ignored on both open and close operations.
process	String	The originating process's image name.
process_id	Number	The originating process ID.
user	String	The user who owns the originating process.
user_id	String	The user ID of the process owning the socket.
proto	String	The packet's protocol: TCP or UDP.

Field	Type	Description
flags	String	One of TCP flags `ACK`, `SYN`, `RST`, `FIN`.
direction	String	Specifies whether the packet came `in` to the managed endpoint, or was sent `out` of the endpoint.
ip_class	Number	Specifies whether IPv4 (`0`) or IPv6 (`1`) was used for the transaction.
seq_number	Number	TCP transaction sequence number (not available in UDP transactions).
src_mac	String	MAC address of originating endpoint.
dst_mac	String	MAC address of destination endpoint (Linux only).
md5	String	The MD5 hash code for the source process.
sha1	String	The SHA-1 hash code for the source process.
sha256	String	The SHA-256 hash code for the source process.

Match network flow originating on CIDR block 10.250.45.255/24 and targeting endpoint 10.0.0.2 on port 22.

```
src_ip contains 10.250.45.255/24 and dst_ip equals 10.0.0.2 and dst_port 22
```

Tip

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name.

Port events

McAfee Active Response only catches these events on Windows managed endpoints.

Event	Description
PortOpened (Windows only)	A port is opened for listening.

Event	Description
PortClosed (Windows only)	A port is closed.

Port output fields

Field	Type	Description
src_port	Number	Port number originating the packet.
user	String	The user who owns the originating process.
user_id	String	The user ID of the process owning the socket.
proto	String	The packet's protocol: TCP or UDP.
md5	String	The MD5 hash code for the source process.
sha1	String	The SHA-1 hash code for the source process.
sha256	String	The SHA-256 hash code for the source process.

Match network flow originating on port 22 by the system administrator.

```
src_port equals 22 and user equals "NT AUTHORITY\\SYSTEM"
```

Processes trigger

The **Processes** trigger listens to events on running processes.

Events

Event	Description
ProcessCreated	A matching process is created on an endpoint.

Event	Description
ProcessTerminated	A matching process is terminated on an endpoint.

Output fields

Field	Type	Description
name	String	The name of the running process.
id	Number	The process's system identifier.
parentId	Number	The system identifier for the process that spawned the current process.
parentimagepath	String	The location path of the parent process.
parentname	String	The name of the process that spawned the current process.
md5	String	The MD5 hash code for the process.
sha1	String	The SHA-1 hash code for the process.
sha256	String	The SHA-256 hash code for the source process.
cmdline	String	The command that started the process.
imagepath	String	Path to the process's image name.
user	String	The user name that started the process.
user_id	String	The ID for the user that started the process.

Match processes started by user "blackhat" with the SHA-1 hash:

97eb5a5b721e28f9696729d14ef9d4076c9b4e2e

```
user equals 'blackhat' and sha1 equals '97eb5a5b721e28f9696729d14ef9d4076c9b4e2e'
```

Tip

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name.

WinRegistry trigger

The **WinRegistry** trigger listens to changes on Windows Registry keys.

Events

Event	Description
ValueCreatedOrModified	Key value created or value data changed.
ValueDeleted	Key value deleted or renamed.

Output fields

Field	Type	Description
keypath	Win Registry String	**Mandatory**. A path to a registry key. The path does not include the key name. If the value is not a valid registry path, the trigger can't be saved. **Tip:** Only `equals` and `starts_with` operators are valid for this output field.
keyvalue	Win Registry String	The key value name.
valuedata	Win Registry String	The data stored by the key value. **Tip:** All values must be expressed as `REG_DWORD` values.
valuetype	Win Registry String	The data type of the registry data.

Catch when a registry value is modified in the registry key path for Active Response configuration.

```
WinRegistry keypath equals "HKLM\\software\\mcafee\\mar" and WinRegistry keyvalue equals "szVersion"
```

Tip

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name.

Adding custom content

Custom content specifies code or scripts that Active Response clients execute on managed endpoints.

This content lives inside the custom collectors and reactions that you create:

- Content written for a collector prints Comma-Separated Value (CSV) records to standard output.
- Content written for a reaction can take values passed as arguments to the operations executed on endpoints.

Best practices

- Collectors should not access resources external to the endpoint, because a collector might run on several thousand endpoints at the same time. Accessing an external resource could cause a Denial of Service attack. An example is an endpoint querying the Active Directory server for non-local user information. If there are several thousand endpoints, this could cause the Active Directory server to crash because of the large number of requests.
- Each collector has a comment at the beginning, specifying its purpose and the copyright.
- Collectors should support unicode characters. Test fields with special characters:
    - If a collector returns the content of a file, and if the file contains unicode characters, verify the output is displayed correctly.
    - If a collector returns the name of a file, and if the file contains unicode characters, verify the output is displayed correctly.
    - If a collector returns a registry key or value, and if the registry key or value contains unicode characters, verify the output is displayed correctly.
    - If a collector returns a command's output, and if the output contains unicode characters, verify the output is displayed correctly.
- Commands executed work independently of the endpoint's configuration.
    - The endpoint might be in a language other than English which affects paths, file names, and commands' output.
    - The endpoint might (not) have a specific software package installed. Use only commands that are default for the operating system.
- Collectors have a default timeout of 60 seconds. Considering the time spent in communication between the endpoint and the Active Response Server, ideally the scripts should be finished in less than 30 seconds.
- A static analysis of the collectors returns no errors and a minimum number of warnings.

Limitations

On Windows, commands that require access to `STDIN` or the desktop fail to execute because Active Response runs on endpoints as a non-interactive service.

Content output

During content execution, Active Response gathers from standard output all lines produced by custom content.

This means that your content must print to standard output only those lines to be parsed as comma-separated value (CSV) records. Consider the following examples.

Content with incorrect data

This simple content executes the `PS` command on a managed endpoint.

```
ps
```

This is a sample output for the command:

```
  PID    PPID    PGID     WINPID   TTY    UID        STIME     COMMAND
 1440   18908    1440      11236  pty2    2831382   14:40:33  /usr/bin/sh
19184    2128   19184      11640  pty3    2831382   17:16:00  /usr/bin/ps
13708       1   19200      13708  ?       2831382   14:43:33  /usr/bin/dbus-launch
16196    1440    1440      12284  pty2    2831382   14:43:33  /usr/bin/xinit
  808       1     808        808  ?       2831382   14:43:33  /usr/bin/dbus-daemon
```

Because the command output's first line contains a header, the following CSV document is constructed:

```
PID,PPID,PGID,WINPID,TTY,UID,STIME,COMMAND
1440,18908,1440,11236,pty2,2831382,14:40:33,/usr/bin/sh
19184,2128,19184,11640,pty3,2831382,17:16:00,/usr/bin/ps
...
```

Active Response incorrectly interprets the first line in the CSV document as being valid data.

Removing incorrect data from output

Contrast this example to *Content with incorrect data*. This content executes the `ps` command, but removes the header line.

```
ps | tail -n +2
```

This is a sample output for the command:

```
 1440   18908    1440      11236  pty2    2831382   14:40:33  /usr/bin/sh
19184    2128   19184      11640  pty3    2831382   17:16:00  /usr/bin/ps
13708       1   19200      13708  ?       2831382   14:43:33  /usr/bin/dbus-launch
16196    1440    1440      12284  pty2    2831382   14:43:33  /usr/bin/xinit
  808       1     808        808  ?       2831382   14:43:33  /usr/bin/dbus-daemon
```

Then, a CSV document with only valid data is constructed:

```
1440,18908,1440,11236,pty2,2831382,14:40:33,/usr/bin/sh
19184,2128,19184,11640,pty3,2831382,17:16:00,/usr/bin/ps
...
```

CSV value escaping

These characters must be escaped in content output to avoid problems when executing collectors and reactions:

```
' \ , [space]
```

To escape one of these characters in content output, place them between double quotes (" and ").

Tip

To escape the double quotes character, use a slash. To escape the slash character, use another slash.

For example:

```
"escaped [space]"
"escaped ,"
"escaped ' "
"escaped \"quotes\" "
"escaped \\"
```

Value strings encoding

All values printed to standard output must be encoded as UTF-8 characters. Using any other encoding can produce characters that break the execution of the collector, producing incorrect output values or no output values at all.

When creating content for collectors, you have the option to encode content output to UTF-8 automatically. If your search results contain broken character encodings, try encoding your custom collector content in UTF-8, or enabling the **Convert collector output to UTF-8 encoding** option from the collector details page.

Timestamp output fields

If your custom collector specifies an output field of type **Timestamp**, you must make sure that the time stamp is generated in full when the content is executed. A complete time stamp includes both date and time values.

Example	Description
`2015-01-09 08:43:25`	This time stamp is complete.
`2015-01-09`	Incomplete: missing time value.
`2015-01`	Incomplete: missing day and time values.
`08:43:25`	Incomplete: missing date value.

Content arguments

During content execution, Active Response can pass values as arguments to be expanded in the content.

Arguments are specified in the content by placing the argument name between `{{` and `}}`.

Content with arguments

In this example content, two arguments are defined: `{{dir_glob}}` and `{{file_glob}}`.

```
for file in {{dir_glob}}/{{file_glob}}.exe; do rm $file; done
```

This content is suitable for a reaction that deletes all files in specific directories, with known file names, ending with the .exe extension. When this content is executed on a managed endpoint, Active Response can expand the argument names with values passed by, for example, a trigger.

Content types

Operating system commands

This content type executes a system command in a managed endpoint.

Caution

Only reference operating system commands and libraries from a trusted source in Active Response custom content.

Linux system command

Show the endpoint's system time.

```
date +%T
```

Windows system command

Show the endpoint's system time.

```
time /t
```

Windows echo display rules

When executing Windows operating system commands, Active Response follows these display rules for the `echo` command.

- The first space after the command name is ignored.
- Trailing spaces in message are ignored.
- Functions and variables not enclosed between back quotes (`) are evaluated.
- To include special characters like `< | >`, enclose them in double quotes (") or back quotes. You can also precede them with the ASCII escape character, or use the `/X` option of the `SETDOS` command.
- To display `%`, you can alternately use two `%` marks for each one to be displayed: `%%`
- To display trailing spaces, either enclose them in back quotes, or append a pair of back quotes behind them.
- The ASCII `NUL` character cannot be included.
- If `stdout` is the console, after displaying content on the current line, the cursor moves to the beginning of the next line.
- If `stdout` is a file, the `CR LF` sequence is appended to the content.
- To display a blank line, use one of these forms:

```
echo `` (two consecutive back quotes)
```

```
echo. (special syntax for compatibility with CMD)
```

Bash scripts

This content type executes a Bash script.

Caution

Only reference operating system commands and libraries from a trusted source in Active Response custom content.

Best practices

- By default, scripts are run with the bash shell. All scripts need to be prefaced with `#!/bin/bash` so if you copy the script from the McAfee ePO interface to a local file, the operating system knows which shell to use.
- File names in Linux can contain almost any character (even newlines). Take this into consideration when parsing a list of files.
- Double check that all your assumptions are valid for all supported distributions. For example, Red Hat version 6 uses `SysV` to manage services, whereas Red Hat version 7 uses `systemd`, but also keeps backward compatibility with `SysV` services. Because of this, a script made for Red Hat 6 runs on Red Hat 7, but might not return all required information. In this case, it would output only the `SysV` services, omitting the `systemd` ones.
- Much of the information collected is inside configuration files (`/etc/hosts`, `/etc/hostname`). Because these files allow commented lines (starting with #), these can be ignored when parsing.

Show interactive users logged on endpoints.

```
#!/bin/bash
#
# Copyright (C) 2015 McAfee, Inc. All Rights Reserved.
#
if [ `w | awk '{ if( NR>2 ) print $3, $1 }' | grep -E ^\: | wc -l` != 0 ]; then
    w | awk '{ if( NR>2 ) print $3, $1 }' | grep -E ^\: | awk '{ print $2 }';
else
    echo "No interactive users found"
fi
```

PowerShell scripts

This content type executes a PowerShell script.

Caution

Only reference operating system commands and libraries from a trusted source in Active Response custom content.

Best practices

- By default, the output of a PowerShell script is reflowed to fit the PowerShell window's default size. Newlines are inserted where needed so that the output fits the window's width. To work around this, preface every script with the following:

```
$pshost = get-host

$pswindow = $pshost.ui.rawui

$newsize = $pswindow.buffersize

$newsize.height = 3000

$newsize.width = 3000

$pswindow.buffersize = $newsize
```

- By default, output is not unicode. Preface every script with:

```
$OutputEncoding = New-Object -typename System.Text.UTF8Encoding
```

- WMI objects provide a convenient interface to access information about the endpoint. However, it can be slow. If the same information can be obtained easily by other means, choose that over WMI.
- WMI objects can access the registry, files, or even network resources to get the information requested. Check the documentation in detail before using them, to avoid unwanted accesses.
-

All commands must support PowerShell 2.0 or greater.

Return information about endpoint system information.

```
#
# Copyright (C) 2015 McAfee, Inc. All Rights Reserved.
#
# Summary       : This script lists endpoint system information
#
$PhysicalMemory = (get-wmiObject -class win32_ComputerSystem).TotalPhysicalMemory
$LocalTime = get-wmiObject -class win32_LocalTime
$OperatingSystem = get-wmiObject -class win32_OperatingSystem
$Processor = get-wmiObject -class win32_Processor
$TimeAndDate = get-date

$o = new-object PSObject
$o | add-member NoteProperty PhysicalMemory $PhysicalMemory
$o | add-member NoteProperty LocalTime $LocalTime
$o | add-member NoteProperty OperatingSystem $OperatingSystem
$o | add-member NoteProperty Processor $Processor
$o | add-member NoteProperty TimeAndDate $TimeAndDate

$p = $o | ConvertTo-CSV -NoTypeInformation | select -Skip 1

$p = $p.replace('\', '\\')
$p
```

Visual Basic scripts

This content type executes a Visual Basic script.

Caution

Only reference operating system commands and libraries from a trusted source in Active Response custom content.

Return information about local users on Windows endpoints.

```
'
' Copyright (C) 2015 McAfee, Inc. All Rights Reserved.
'
' Summary       : This script will list all local user
'                 information, to include group memberships.
'
Option Explicit

' ************************************************
' Declare all variables
' ************************************************

Dim strComputer
Dim varUseWmi, varRunWmiQuery, varWmiValue
Dim colGroups
Dim objGroup, objUser

' ************************************************
' Call WMI to gather Windows
' user account information.
' ************************************************

strComputer = "."

set varUseWmi = GetObject("winmgmts:\\.\root\cimv2")
set varRunWmiQuery = varUseWmi.ExecQuery("Select * from Win32_UserAccount")

' ************************************************
' List all groups for each user, and put into an
' array.
'
' Next, echo back all of the user info, to include
' the group.
' ************************************************

For Each varWmiValue In varRunWmiQuery
Set colGroups = GetObject("WinNT://" & strComputer)
colGroups.Filter = Array("group")
        For Each objGroup In colGroups
                For Each objUser In objGroup.Members
                        If objUser.name = varWmiValue.Name Then
                                Wscript.Echo varWmiValue.Disabled & "," & varWmiValue.Domain & "," &
varWmiValue.FullName & "," & varWmiValue.InstallDate & "," & varWmiValue.LocalAccount & "," &
varWmiValue.Lockout & "," & varWmiValue.Name & "," & varWmiValue.SID & "," & varWmiValue.PasswordExpires &
"," & objGroup.Name
                        End If
                Next
        Next
Next
```

Python 2.7 scripts

This content type executes a Python 2.7 script.

Caution

Do not create Python custom content unless you are sure that the Python interpreter on endpoints is installed in a system-protected location.

Return information about routes.

```
#
# Copyright (C) 2015 McAfee, Inc. All Rights Reserved.
#
import subprocess
process = subprocess.Popen("route PRINT -4", stdin=subprocess.PIPE, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, shell=True)
output, error = process.communicate()
process = False
import re
map_list = []

for x in output.split('\r'):
    if "Metric" in x:
        process = True
        continue
    if process:
        data = re.sub('\s+', ' ',x).strip().split(" ")
        if len(data)>=3:
            print( ",".join(data))
```

Managing access

After installation, Active Response creates permission sets to manage access to its resources.

- **Group Active Response Editor** — Allows access to all features and resources. Most importantly, this permission set allows users to create, edit, and delete collectors, triggers, and reactions. Set this permission set for users that need to:
    - Create custom content.
    - Set triggers to automatically catch events on endpoints and execute reactions.
    - Back up or share custom content with other McAfee ePO instances.
- **Group Active Response Responder** — Allows access to Active Response Search. It also allows users to see the content and configuration of collectors, triggers, and reactions, but not to edit or delete them. Set this permission set for users that need to:
    - Actively monitor endpoints for indicators of compromise.
    - Quickly execute reactions from Active Response Search results.
- **Group Active Response Responder Workspace Monitor** — Allows access to the **Threat Workspace** and Active Response Search functions. It allows users to see threat behavior activity, and to execute searches to investigate a potential threat but not take remediation actions. Set this permission for users that need to:
    - Actively monitor endpoints for indicators of compromise.
    - Inform incident responders who can remediate a possible threat.
- **Group Active Response Workspace Responder** — Allows full access to the **Threat Workspace** and Active Response Search functions. It allows users to see threat behavior activity, execute searches to investigate a potential threat and take immediate action through the **Threat Workspace**, or automate tasks on endpoints through triggers and reactions. Set this permission for users that need to:
    - Actively monitor endpoints for indicators of compromise.
    - Take immediate action on endpoints using the **Threat Workspace**.
    - Quickly execute reactions from search results.

- Create custom content.
- Set triggers to automatically catch events on endpoints and execute reactions.
- Back up or share custom content with other McAfee ePO instances.

You can also customize access management by creating your own permission sets.

Privacy information and Active Response

Active Response collects information from managed endpoints, such as user names, system names, and IP addresses. It also includes process activity such as modified registry entries, files created, and established network connections. Access to this information is available in Active Response pages in McAfee ePO. Make sure that access to these pages is authorized and appropriately managed.

McAfee ePO restrictions to the **System Tree** through access management configuration do not prevent Active Response users from receiving information from systems outside their authorized segment of the **System Tree**. Make sure that Active Response users are qualified and trained to appropriately handle private information from your users' systems.

McAfee also collects data that is not personally identifiable to further enhance threat intelligence, but cannot search the data or trace it back to a specific organization. For more information, review the License Agreement.

Active Response Permission Sets

Manage access to Active Response objects.

Collectors section

Option	Definition
No permissions	Blocks access to collectors in the **Active Response Catalog**.
View, use, and export	Allows `read` access to collectors.
Create, edit, delete, and import	Allows `write` access to collectors.

Reactions section

Option	Definition
No permissions	Blocks access to reactions in the **Active Response Catalog**.
View, use, and export	Allows `read` access to reactions.

Option	Definition
Create, edit, delete, and import	Allows `write` access to reactions.
Apply reactions over search results.	Allows to execute reactions over results from an Active Response search expression.

Triggers section

Option	Definition
No permissions	Blocks access to triggers in the **Active Response Catalog**.
View, use, and export	Allows `read` access to triggers.
Create, edit, delete, and import	Allows `write` access to triggers.

Saved searches section

Option	Definition
No permissions	Blocks access to saved search expression in the **Active Response Catalog**.
View, use, and export	Allows `read` access to saved search expressions.
Create, edit, delete, and import	Allows `write` access to saved search expressions.

Recommendations for configuring clients

Use McAfee ePO policies to configure Active Response clients.

Using policies, you can:

- Set the maximum number of results returned by search expressions.
- Enable endpoints to execute triggers.
- Enable **Network Flow** and **File Hashing** collectors and triggers.
- Enable the Trace plug-in on the endpoint. This is required to see potential threat activity in the **Threat Workspace**.

- Set database limits and maximum number of results returned by the **Network Flow** collector. For Network Flow in Windows, traffic can be excluded for specific processes. This is done using the complete process path.
- Set database limits, maximum number of results returned, and files excluded by the **File Hashing** collector.
    - You can also exclude entire paths and extensions by policy.
    - File Hashing "Hash Strategy" determines how many endpoint resources are dedicated for hashing. For example, setting the default to **Low** reduces performance impact (resource consumption), but makes the hashing period longer.
- Set database and data limits for the **Trace** collector.
- Enable system logging on managed endpoints.
- Enable data folder protection. When selected, you cannot read the files in C:\ProgramData\McAfee\MAR\data. Deselect it to read the logs and config files.

Preset McAfee ePO policies

After installing Active Response, the following McAfee ePO policies are available in the **Policy Catalog**:

- **McAfee Default** — This is the policy enforced by default after installation. When this policy is enforced, **Network Flow** and **Trace** collectors are enabled. **Triggers** and **File Hashing** are disabled.
- **Full Visibility** — When this policy is enforced, **NetworkFlow**, **File Hashing**, and **Trace** collectors are enabled. **Triggers** are disabled.
- **Full Monitoring** — When this policy is enforced, all collectors and triggers are enabled.

Create an Active Response policy

Add custom Active Response policies to the **Policy Catalog**.

Task

1. Select **Menu** → **Policy** → **Policy Catalog**.
2. From the **Product** list, select **Active Response**.
3. Select **New Policy**, or select an existing policy and select **Duplicate**.
4. Enter a name and a brief description for the new policy, then click **OK**.
5. Complete the fields on the **Policy Catalog** page for the options you want to apply to the policy.

What to do next

After you create a policy, assign it to managed systems to configure the Active Response clients on those systems. See the McAfee ePO documentation for information about assigning policies.

Configure a policy

Change the plug-in settings on managed endpoints.

Task

1. Log on to McAfee ePO as an administrator.
2. Select **Menu** → **Policy** → **Policy Catalog**, then select **Full Visibility**.
3. Select a tab and adjust the settings for the policy.

General policy configuration

Configure Active Response on managed endpoints.

Option	Definition
Max merge retrieve	Sets the maximum number of results returned by Active Response search expressions. The maximum limit is 512.
Enable triggers	Enables endpoints to execute triggers.
Enable data folder protection	Data folder isn't readable. Only log files are readable when they are enabled. If they are enabled, files in %PROGRAMDATA%\McAfee\MAR\data are readable.
Enable Unattended Content Updates	Enables endpoints to update content without an explicit deployment task.
Unattended Content Updates Timeout	Displays the update frequency in minutes used by endpoint when unattended content updates are enabled.

Network Flow policy configuration

Configure the **Network Flow** plug-in on managed endpoints.

Option definitions

Option	Definition
Enable Plug-in	Enables the **NetworkFlow** collector and trigger capabilities on the endpoint.
Enable Network Sniffing	Enables you to configure your network-related activities to a granular level. This option is enabled by default. If this option is disabled, certain network activities that require

Option	Definition
	higher resource consumption such as network triggers and network flow collectors are disabled.
Max database size (MB)	The size limit for the **NetworkFlow** collector database on endpoints. When this limit is reached, oldest records are discarded from the database.
Max database size %	The percentage of endpoint storage that the **NetworkFlow** collector database can use.
Max rows retrieve	Maximum number of result rows returned by the **NetworkFlow** collector. The maximum number of rows retrieved is 512.
Collect TCP/UDP System process information (Windows only)	Collects TCP/UDP connection information generated by system processes on endpoints running Windows. It determines if an application tried to connect to a particular host or IP address. Disabling this option collects only user space application connections. **Note:** If you do not need to collect TCP/UDP information, do not enable this feature.
Ignore Process for collection of TCP/UDP information - (Use ';' as separator) (Windows only)	Excludes processes from the **NetworkFlow** monitoring to reduce resource consumption and performance impact in systems that are web servers.
Display Message on Quarantine actions	Sends notifications to endpoints when the endpoints are quarantined.
Quarantine Endpoint notification message	The default notification message that is sent to an endpoint when it is quarantined. This message can be customized.
Remove Quarantine Endpoint notification message	The default notification message that is sent when the quarantine status is removed from an endpoint. This message can be customized.

File Hashing policy configuration

Configure the **File Hashing** plug-in on managed endpoints.

Option definitions

Option	Definition
Enable Plug-in	Enables the **Files** collector and trigger capabilities on the endpoint.
Max database size (MB)	The size limit for the **Files** collector database on endpoints. If the value exceeds the **Max database size (MB)**, then Active Response tries to get as close as possible to the maximum storage target by removing information about deleted files from the database.
Max database size %	The maximum storage target size that the **File Hashing** plug-in database tries to keep. If the value exceeds the **Max database size %** value, then Active Response tries to get as close as possible to the maximum storage target by removing information about deleted files from the database.
Max rows retrieve	Maximum number of result rows returned by the **Files** collector.
Hash Strategy	Selects the priority of the file hashing process. A lower priority uses less endpoint resources. • **Low** — This is the recommended setting. Uses the least endpoint resources. • **Normal** — The operating system alone decides priority. • **Medium** — Incremented priority. • **High** — Top priority. • **Auto** — Switches between **Low** and **Normal** strategies depending on endpoint usage.
Pause on battery	Pause file hashing when the endpoint is running on battery to preserve battery life.
Seconds to delay hashing after boot	The time in seconds to delay the file hashing process in the Active Response clients after booting the system. The endpoints will start the warm-up procedure for the file hashing component only after the specified delay time is over. The default value is 120 seconds.
Ignore Files on Windows	A list of file paths to ignore. File paths must be complete, indicating full path, file name, and extension. For example, C:\PAGEFILE.SYS. **Note:** Unless there is a reason to ignore a specific file, only ignore those files that are repeatedly opened and modified.

Option	Definition
	You can use these system variables to specify ignored files and paths: • `%systemdrive%` • `%profilesdirectory%` • `%windir%` • `%commonprogramfiles%` • `%commonprogramfiles(x86)%` • `%comspec%` • `%homedrive%` • `%programdata%` • `%programfiles%` • `%programfiles(x86)%` • `%public%` • `%systemroot%` • `%temp%` • `%tmp%` Active Response only expands system user variables. Use of variables that point to user paths is not recommended. Avoid using the following variables: • `%allusersprofile%` • `%appdata%` • `%homepath%` • `%localappdata%` • `%userprofile%` Under certain circumstances, endpoint users with administrator level access can change the destination folders of system variables. **Tip:** If you must use system variables, you can create an Active Response trigger that catches changes to system variables in the Windows registry. The trigger must be of **WinRegistry** type and watch for changes on this registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment.
Ignore Extensions on Windows	A list of file extensions to ignore. Place a '.' before each extension, separated by semicolons. For example, .swap;.o;.temp;

Option	Definition
	Note: Unless there is a reason to ignore a specific extension, only ignore those extensions that are repeatedly opened and modified.
Ignore Paths on Windows	Paths and folder names to be ignored. You can specify folder paths in full or partially, using '*' to replace the beginning of a path, or entering '**' to ignore a folder regardless of how many subdirectories it appears. For example: • *bar excludes folders bar and foobar anywhere in the file system. • *foo\bar excludes the bar folder inside the foo and snafoo directories, anywhere in the file system. • C:\foo\bar\ excludes only the folder bar inside C:\foo directory. • C:**\bar\ excludes one or more subdirectories where bar is found. For example, ignores c:\foo\bar, c:\snafoo\bar, and c:\snafoo\foo\bar. **Warning:** When entering absolute paths, the trailing '\' is mandatory. Removing a path from the ignored list makes the path available for file hashing only after the endpoint system is restarted. Until then, only newly created files are hashed at previously ignored paths. Windows operating system exclusions must start with a Microsoft "FOLDERID_" prefix. For details, see Microsoft documentation "knownfolderid" for complete supported list. Directory paths are case sensitive in ignore paths on Windows for Active Response 2.4.x and below. **Note:** Unless there is a reason to ignore a specific path, only ignore those paths that are repeatedly accessed.
Ignore Files on Linux	A list of file paths to ignore. File paths must be complete, indicating full path, file name, and extension. For example, /etc/cma.d/lpc.conf; You can use the following system variables to specify ignored files and paths: • `$BASH_VERSION` • `$HOSTNAME` • `$CDPATH` • `$HISTFILE` • `$HISTFILESIZE`

Option	Definition
	• $HISTSIZE • $HOME • $IFS • $LANG • $PATH • $PS1 • $TMOUT • $TERM • $SHELL • $DISPLAY • $EDITOR
Ignore Extensions on Linux	A list of file extensions to ignore. Place a '.' before each extension, separated by semicolons. For example, .swp;.o;.temp;
Ignore Paths on Linux	Paths and folder names to be ignored. You can specify folder paths in full or partially, using '*' to replace the beginning of a path. For example: • *bar excludes folders bar and foobar anywhere in the file system. • */bar excludes the bar folder anywhere in the file system. • /foo/bar excludes only the folder bar inside the /foo directory. Removing a path from the ignored list makes the path completely available for file hashing only after the endpoint system is restarted. Until then, only newly created files are hashed at previously ignored paths.
Max File Size for Hashing (Mb)	The maximum file size that Active Response hashes. Files above this limit are excluded from file hashing.
Ignore Files on macOS	A list of file paths to ignore. You can use ';' as separator.
Ignore Extensions on macOS	A list of file extensions to ignore. You can use ';' as separator.

Option	Definition
Ignore Paths on macOS	Paths and folder names to be ignored. You can use ';' as separator.

Defining the maximum thresholds for network database and file hash database

The maximum size of your network and file hash databases affects the network flow, file hash, and trace events. You can use Active Response to maintain the size of your network and file hash databases based on a configured value.

Managing network database

In a network database, when the database size exceeds the **Max database size (MB)** specified in the **Network Flow** policy, older rows are deleted. This cleanup task is performed every hour, by default. Between a cleanup task and the next cleanup, the database size can increase.

Increase in network activity on an endpoint can also increase the database size rapidly between cleanups. For web servers, you might already know about the size increase in advance. The increase can be due to a process that generates a large amount of data when connected to the network. In such scenarios, to keep the database size from exceeding the limit, you can add the process name to **Ignore Process for collection of TCP/UDP information - (Use ';' as separator)(Windows only)**.

Managing file hash database

A file hash database stores the list of files and attributes in the system. It contains existing files and some deleted files. When the database size exceeds **Max database size (MB)** specified in the **File Hashing** policy, Active Response starts removing the rows for the deleted files. But the rows for the existing files aren't removed.

If an endpoint has numerous files, the file hash database size appears more than the limit specified in the policy. Active Response can't remove the entries in the database as these entries can be linked to a file in the file system. This is because Active Response doesn't have visibility whether the entries belong to a deleted file. The limit specified in the file hash database is an indication to start removing the deleted files that are in the database. To prevent the database size from exceeding the limit, you can add these extensions or paths to the ignored lists in the policy:

- **Ignore Files on Windows - (Use ';' as separator)**
- **Ignore Extensions on Windows (Use ';' as separator)**
- **Ignore Paths on Windows (Use ';' as separator)**

Example 1

Assume that **Max database size (MB)** for File Hashing policy is set to 60 MB. The entries for the current files (that exist in the file system) occupy 50 MB. The remaining 10 MB in the database are for storing the entries for deleted files in the file system.

Example 2

Assume that **Max database size (MB)** for File Hashing policy is set to 60 MB. The entries for the current files (that exist in the file system) occupy 200 MB because the endpoint has numerous files. As a result, there is no space reserved for the deleted files. But, Active Response retains the entries for deleted files up to 30 days.

Max database size (MB) and Max database size %

There are two options for configuring the network and file hash database size thresholds: **Max database size (MB)** and **Max database size %**. If **Max database size (MB)** is set to a size in MB, Active Response uses it as the defined threshold. If **Max database size (MB)** is set to 0 (zero), Active Response uses the percentage threshold. This percentage is calculated based on the disk, where Active Response is installed, for example,**C:**.

You can also disable a feature if you don't want a specific capability in the server. For example, if you don't want the network sniffing feature, you can disable the **Enable Network Sniffing** option.

Trace policy configuration

Configure the **Trace** plug-in on managed endpoints. Trace information is sent to Active Response and displayed on the **Threat Workspace** page. This information helps you determine how threats and potential threats move through your environment.

Option definitions

Option	Definition
Enable Plug-in	Enables the **Trace** plug-in on the Microsoft Windows endpoint.
Enable Plug-in for macOS Endpoints (Beta)	Enables the **Trace** plug-in on the macOS endpoint. This plug-in is disabled by default.
Report Internal Reputation Failures	When enabled, the Active Response client reports a McAfee ePO threat event due to internal connection issues for the Trace plug-in. This plug-in is disabled by default.
Max database size (MB)	The size limit for the **Trace** database on the endpoint. When this limit is reached, oldest records are discarded from the database.
Real time notification	Sends trace events as they occur.

Option	Definition
Interval to send data	If **Real time notification** is disabled, specify how often (in seconds) to send trace event notifications. The default value is 30 seconds.
Log Level	Specify the level of information to include in the **Trace** log file. Selecting an option includes that information and all options below it. For example, selecting the **Trace** log level includes trace, info, warning, and error information. Use the lowest log level whenever possible, or disable this feature if you do not need log information. • **Debug** — Detailed debug information. • **Trace** — Contains most variable value dumps. This can be too verbose to debug problems on production systems. • **Info** — Logs messages that highlight the progress of the **Trace** plug-in. • **Warning** — Information about potentially harmful situations. • **Error** — Error events that might prevent the **Trace** plug-in from running. The **Trace** Log file is located at %PROGRAMDATA%\McAfee\Mar\data\ts_events*.log
Configuration	This text box is used to enter configuration code that can enable or disable **Trace** features. Enter information in this text box only when working with a Technical Support representative.

Logger policy configuration

Configure how endpoints log Active Response client service events.

Option definitions

Option	Definition
Logger format	Select **None** to stop Active Response from creating logs. Select **File** to create logs in files. Logs are stored in the following locations: • Windows: %systemdrive%\ProgramData\McAfee\Mar\data\marlog.log • Linux: /var/McAfee/Mar/data/marlog.log • macOS: /var/McAfee/Mar/data/marlog.log
Level	• **Debug** — Logs fine-grained events useful to identify problems with Active Response client execution. • **Trace** — Logs almost all variable value dumps. This can be too verbose to debug problems on production systems. • **Info** — The default level. It logs messages that highlight the progress of the Active Response client service.

Option	Definition
	• **Warning** — Logs potentially harmful situations. • **Error** — Logs errors that cause Active Response client service to abort. **Note:** Use the lowest log level whenever possible, or disable this feature if you do not need log information.
Buffer Size	The number of messages that the logger holds in a buffer before saving to the log file. If set to 1, Active Response updates the log file immediately after each message is generated.

Configuring Active Response service

Configure how the Active Response service works. Use the Active Response option in the McAfee ePO **Server Settings** page.

Search execution time-to-live

Active Response search expressions execute collectors on managed endpoints. Because endpoints might come online or offline during the execution of a collector, Active Response can't know when all endpoints that could answer have already answered. This configuration tells Active Response to stop expecting search results after a certain time has passed.

Active Response server options

Option	Definition
Search time-to-live	The timeout (in milliseconds) that Active Response waits since the last endpoint replied to a search expression. If another endpoint replies during this wait, the time count is restarted. Else, the search stops. Default: 15,000 ms
Search time-to-live at 50%	Defines a percentage of the value in **Search time-to-live** that applies as the new timeout wait after 50% of available endpoints have replied. Default: 33%
Search time-to-live at 90%	Defines a percentage of the value in **Search time-to-live** that applies as the new timeout wait after 90% of available endpoints have replied. Default: 7%

Option	Definition
Compatibility with Active Response 1.0 clients	When enabled, the legacy Active Response endpoint clients reply to searches, reactions, and triggers executed by an Active Response server.
Authentication	The Active Response service relies on McAfee ePO certificates to authenticate access, so that only Active Response extensions can make service requests. This configuration is set up after the installation of the Active Response service. If you change the certificates used by McAfee ePO, use this configuration option to reset the certificates in the Active Response server.

Active Response Workspace configuration

These Workspace configuration settings control what you see on the **Threat Workspace**.

Option	Description
Max Events on Trace chart (Chrome and Firefox)	The **Process instances** setting controls the number of potential threat instances that display on the trace chart.
Max Events on Trace chart (Internet Explorer)	The **Events per instance** setting controls the number of potential threat events that display on the trace chart.

McAfee collects your usage telemetry data only after you select the **I choose to share telemetry data** checkbox and agree to share data.

Server and aggregator tags

After installation, the Active Response server and aggregator systems are automatically applied with these tags:

- `MARSERVER` — Identifies the Active Response server.
- `MARAGG` — Identifies an Active Response aggregator system.
- `DXLBROKER` — Identifies both the Active Response server and the aggregators.

You can review and edit the tags applied to your systems in the McAfee ePO **System Tree**.

Configuration examples and benefits

You can improve the performance of running on Windows servers. These configuration methods can improve performance, but MVISION EDR can lose visibility into all devices. As a result, all threats may not be detected.

network flow

- From **Menu** → **Policy** → **Policy Control**, select the **Network Flow** tab and deselect **Collect TCP/UDP System process information (Windows only)**.
- Prevent from tracking and keeping a history of all connections to save disk and CPU usage. To do this, ignore the network traffic from the binary that attends to network requests. Configure this behavior through the device policy in McAfee ePO by using the full path of the binary. For example:
  - Apache server — C:\Apache24\bin\httpd.exe
  - IIS web server — C:\Windows\System32\inetsrv\w3wp.exe

file hashing

- Select the **File Hashing** tab and set the **Hash Strategy** to **Low**.
- Ignore folders where: The server logs and data is saved, the server databases are located, and the servers data backup folders are located. This prevents from tracking and keeping a history of all files created, deleted, and changed, avoiding demands on disk and CPU usage. For example:
  - Apache server — C:\Apache24\logs; C:\Apache24\htdocs
  - IIS web server — C:\inetpub\wwwroot; C:\inetpub\logs

file hashing for SQL Server

- Select the **File Hashing** tab and disable the plug-in. If the file hashing is disabled, the File Hash collector from real-time search does not return results for that endpoint. However, the file activity is monitored as part of the trace and there will be limited visibility for those files created/deleted during the retention timeframe.
- Ignore these SQL Server policy extensions: `ldf`, `mdf`, `adf`, `bak`.
- Ignore FOLDERID_ProgramFiles\Microsoft SQL Server and the backup folder.

Error codes

These error codes appear in **Active Response Search** or in Active Response client logs. Use this table to troubleshoot a problem or as reference when contacting product support.

Generic errors

Code	Name	Description	Workaround
1	MAR_E_UNKNOWN	Failed to execute a search expression, enable a trigger, or execute a reaction.	Check the custom collector content, the reaction content, or the trigger condition.

Code	Name	Description	Workaround
2	MAR_E_UNDEFINED	Failed to execute a search expression, enable a trigger, or execute a reaction.	Check the custom collector content, the reaction content, or the trigger condition.
3	MAR_E_REQUEST_FAIL_TO_BE_PLACE	Failed to access client plug-in. The Active Response client might be corrupted.	Redeploy Active Response client on endpoint.
4	MAR_E_INTERNAL_ERROR	Failed during process boot. The Active Response client might be corrupted.	Redeploy Active Response client on endpoint.
6	MAR_E_MERGE_SIZE_MAX_REACHED	The search expression produced too many results.	Add filters to reduce the number of results or remove collectors from the projection.
7	MAR_E_MISSING_ARGUMENT	Failed to create McAfee ePO events.	Check Active Response server and client versions. The server version must be equal or higher than the client one.
8	MAR_E_INVALID_ARGUMENT	A McAfee ePO event failed to create proper arguments due to an unsupported event ID.	Check Active Response server and client versions. The server version must be equal or higher than the client one.
9	MAR_E_REQUEST_TIMEOUT	A collector took too long to return results.	Reduce the execution time of your custom collectors.
10	MAR_E_PLUGIN_SHUTTING_DOWN	A plug-in is shutting down and has not yet ended.	None
11	MAR_E_UNSUPPORTED_API	An API from a different version is trying to run and is not supported.	None

Code	Name	Description	Workaround
15	MAR_E_FEATURE_DISABLED	The requested feature is disabled.	Check the Active Response policy to make sure that the feature is enabled.
160	MAR_E_GENERIC_PLUGIN_IS_DISABLED	A required Active Response plug-in is disabled on the endpoint.	Enable the plug-in in the Active Response policy enforced on the endpoint.

Runtime plug-in errors

Code	Name	Description	Workaround
256	MAR_E_RUNTIME_BASE	Active Response client failed to encode the custom collector code or failed to generate the temporary file with the custom collector code.	Check the content of the collector.
257	MAR_E_RUNTIME_FAIL	A collector or reaction failed during the execution of its content.	Check the content of the collector or reaction.
258	MAR_E_MISSING_CONTENT	Failed to execute collector or reaction due to missing content. The collector or reaction might be empty.	Check content of collector or reaction.
259	MAR_E_MISSING_SCRIPT_ENGINE	A collector or reaction content failed to be executed due to missing script engine.	Check that Python, VisualBasic, or Bash engines are available on the endpoint.
260	MAR_E_MISSING_SCRIPT_DATA	Failed to execute collector or reaction due to missing content. The content is empty or there is a problem in the Active Response server.	Check the content of collector or reaction.

Code	Name	Description	Workaround
261	MAR_E_SCRIPT_ENGINE_UNSUPPORTED	The Active Response client doesn't support the script engine that it tries to use.	Check that versions of Active Response server and clients match.
262	MAR_E_FORMAT_ERROR	Failed to parse collector output.	Check the output values in the collector content. Check the collector output field definitions.
263	MAR_E_MISSING_PYTHON_ENGINE	Python interpreter can't be found.	Install Python on the endpoint.
264	MAR_E_SHELL_IS_NOT_TRUSTED	The script interpreter doesn't match a trusted interpreter. Active Response will not execute any script using it.	None
265	MAR_E_SCRIPT_TIMED_OUT	The execution of the collector or reaction exceeds the configured timeout.	Check collector or reaction timeout configuration in the Active Response Catalog.
416	MAR_E_RUNTIME_PLUGIN_IS_DISABLED	A required Active Response plug-in is disabled on the endpoint.	Change the Active Response policy enforced on the endpoint to enable the plug-in.

NetworkFlow errors

Code	Name	Description	Workaround
513	MAR_E_NETWORK_MAX_REACHED	The **NetworkFlow** collector returned too many results.	Add filters to reduce the number of results.
514	MAR_E_NETWORK_QUARANTINE_FAIL	Failed to quarantine or unquarantine the host.	None

Code	Name	Description	Workaround
672	MAR_E_NETWORK_PLUGIN_IS_DISABLED	The **NetworkFlow** plug-in is disabled on the endpoint.	Change the Active Response policy enforced on the endpoint to enable the plug-in.

File hashing errors

Code	Name	Description	Workaround
769	MAR_E_FILE_HASHING_MAX_REACHED	The **Files** collector returned too many results.	Add filters to reduce the number of results.
770	MAR_E_FILE_HASHING_HASH_IN_PROGRESS	Active Response is hashing the file system on this endpoint.	Wait for file hashing to complete and retry your search.
771	MAR_E_FILE_HASHING_REMOVE_FILE_ERROR	An error occurred when Active Response tried to delete a file.	None
772	MAR_E_FILE_HASHING_DB_CORRUPTED	An internal database is corrupted.	None
928	MAR_E_FILE_HASHING_PLUGIN_IS_DISABLED	The **File Hashing** plug-in is disabled on the endpoint.	Change the Active Response policy enforced on the endpoint to enable the plug-in.

Processes errors

Code	Name	Description	Workaround
1025	MAR_E_AQUIRE_PROCESS	The endpoint's operating system is preventing Active Response from collecting running processes information.	Retry your search expression.

Code	Name	Description	Workaround
1026	MAR_E_SYSTEM_INFO_INVALID_PARAMETERS	The client detected invalid system information parameters.	Verify that the correct parameters are used. Set the logger level in Debug to check which parameters the client is receiving and retry.
1027	MAR_E_CANNOT_KILL_PROCESS	The client cannot kill the specified process.	Verify that the process exists and its ID is entered correctly.
1028	MAR_E_CANNOT_STOP_SERVICE	The client failed to stop the specified service.	Verify that the service exists and its ID is entered correctly.
1029	MAR_E_CANNOT_KILL_SERVICE_PROCESS	The client failed to kill the specified service.	Verify that the service exists and its ID is entered correctly.
1030	MAR_E_CANNOT_SET_SERVICE_AS_MANUAL	The client failed to set service 'Startup Type' to Manual. The process has been killed, but the service is still automatic.	Retry if the process starts again.
1031	MAR_E_CANNOT_KILL_TRUSTED_PROCESS	The client does not kill trusted processes.	None
1184	MAR_E_SYSTEM_INFO_PLUGIN_IS_DISABLED	McAfee ePO hasn't yet initialized the policies on the endpoint, so the **Processes** plug-in is disabled.	Wait for McAfee ePO to initialize policies on the endpoint and try again.

WinRegistry errors

Code	Name	Description	Workaround
1281	MAR_E_WIN_REGISTRY_MAX_REACHED	The **WinRegistry** collector returned too many results.	Add filters to reduce the number of results.
1282	MAR_E_WIN_REGISTRY_INVALID_PARAMETERS	A **WinRegistry** reaction received invalid parameters	The keypath/keyvalue specified doesn't exist. Check that the correct keypath/keyvalue is used.
1283	MAR_E_WIN_REGISTRY_ACCESS_DENIED	A **WinRegistry** reaction did not have permission to execute its task.	None
1284	MAR_E_WIN_REGISTRY_UNDEFINED_ERROR	A **WinRegistry** reaction returned an unknown error.	None
1285	MAR_E_WIN_REGISTRY_MISSING_ARGUMENT	A **WinRegistry** reaction did not receive all the parameters it was expecting.	This error is not generated in the Active Response client. Check the service or extension.
1286	MAR_E_WIN_REGISTRY_CANNOT_FIND_USER	A **WinRegistry** reaction could not find the specified user.	Check that the correct user is specified.
1287	MAR_E_WIN_REGISTRY_INVALID_KEYPATH_OPERATOR	A condition for the keypath field used an invalid operator.	The Keypath operator supports only the *equals* or *starts with* operators. Change the condition to use one of these operators.

Code	Name	Description	Workaround
1288	MAR_E_WIN_REGISTRY_KEYPATH_IS_MANDATORY	A condition was executed without using keypath as a filter.	WinRegistry queries must apply a filter related to the Keypath condition.
1440	MAR_E_WIN_REGISTRY_PLUGIN_IS_DISABLED	The **WinRegistry** plug-in is disabled on the endpoint.	Change the Active Response policy enforced on the endpoint to enable the plug-in.

Event logs

In Active Response, when a trigger or reaction executes or fails to execute, an Event ID entry is added to the McAfee ePO **Threat Event Log**.

Event IDs and descriptions

Event ID	Event name	Description
36625	Active Response Reaction Event	A reaction event is executed.
36626	Active Response Reaction Event Failure	A reaction event executed with errors.
36627	Active Response FileHashing Trigger Event	A FileHashing trigger event occurred.
36628	Active Response SystemInfo Trigger Event	A SystemInfo trigger event occurred.
36629	Active Response NetworkFlow Trigger Event	A NetworkFlow trigger event occurred.
36630	Active Response WinRegistry Trigger Event	A WinRegistry trigger event occurred.
36631	Active Response Context Trigger Event	A Context trigger event occurred.
36632	Active Response Task Event	A task is configured successfully.
36633	Active Response Task Event	A task is removed successfully.

Event ID	Event name	Description
36634	Active Response Task Event Failure	A task event executed with errors.
36635	Update successful	Content update is successful.
36636	Update failed	Content update failed.
36637	Roll back successful	Roll back of trace content is successful.
36638	Roll back failed	Roll back of trace content failed.
36639	Active Response Install Event Failure	An install event executed with errors.
36640	Active Response Trace Connection Success	A trace connection using VLAN Trunk Protocol (VTP) or RepBO is successful.
36641	Active Response Trace Internal Reputation Failure	Trace connection using VTP or RepBO failed.

COPYRIGHT

Trellix