Product Guide

McAfee Active Response 1.1.0

For use with McAfee ePolicy Orchestrator

Contents

Preface

This guide provides the information you need to work with your McAfee product.

Contents

About this guide

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

Audience

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

The information in this guide is intended primarily for:

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

Conventions

This guide uses these typographical conventions and icons.

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

Find product documentation

On the **ServicePortal**, you can find information about a released product, including product documentation, technical articles, and more.

Task

1 Go to the **ServicePortal** at https://support.mcafee.com and click the **Knowledge Center** tab.

2 In the **Knowledge Base** pane under **Content Source**, click **Product Documentation**.

3 Select a product and version, then click **Search** to display a list of documents.

1 Introduction

McAfee® Active Response is a threat detection and response tool. It provides real-time information about endpoints on your network.

Every day, focused and persistent attacks threaten enterprise networks. Being aware of system status and quickly responding to security incidents is key. Active Response is the tool to handle these incidents.

Through early detection of suspicious activity or by detecting indicators of prior attacks, corporate IT security staff can deal effectively with security breaches. Active Response provides real-time visibility of system status, discovers anomalous events, detects indicators of compromise, and acts on compromised systems.

Contents

What you can do with McAfee Active Response

McAfee Active Response discovers, detects, and responds to previously unseen threats.

Active Response offers real-time visibility of endpoint data and immediate operation on endpoint systems. Out of the box, Active Response provides built-in data collectors, triggers, and reactions to get started right away. Also, incident responders can easily introduce custom content for specific usage. These powerful features increase system management capabilities while reducing time and cost.

Discover

Use Active Response to look for incidents. Its search and data collectors produce actionable information by exploring data.

- Discover weaknesses in your network endpoints.
- Prepare for planned protection activities.
- Identify network data flows and patterns.
- Learn what to include in security policies.

Detect

Use Active Response to detect threats when systems are compromised. Its triggers and reactions catch threatening events on the spot, and react immediately.

- Monitor network systems for your custom indicators of compromise.
- Catch known threats automatically, and react accordingly.
- Identify security needs for servers based on system's network flow.

Respond

Use Active Response to stop threats when they are detected. You can take immediate action on affected endpoints.

- Contain compromising events by acting on endpoints remotely at once.
- Minimize impact by automatically reacting to detected threats.
- Build code to run on compromised systems.

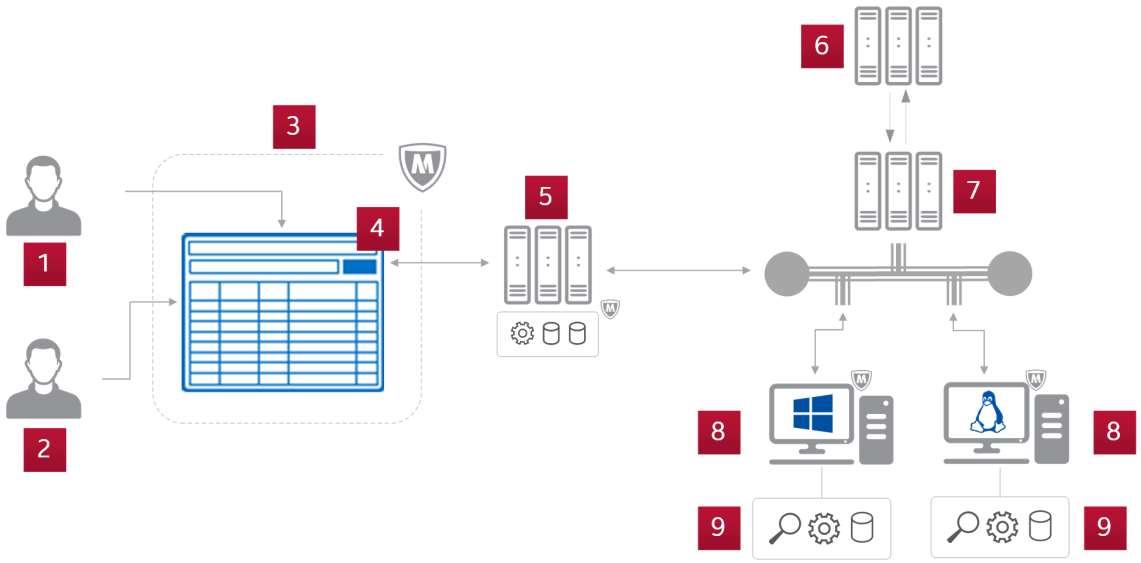
See also

Searching endpoint data on page 25
Collecting endpoint data on page 30
Reacting to incidents on page 41
Catching threats on page 43

Active Response components

McAfee Active Response is composed of three major components: the service, a set of extensions, and endpoint clients.

This figure shows the major Active Response components.



The major components are:

1. **Content developer** — This user creates and manages content in Active Response. Content includes custom data collectors, reactions, and trigger settings.
2. **Incident responder** — Using content on Active Response, this user discovers, detects,b and responds to threatening events.
3. **McAfee® ePolicy Orchestrator® (McAfee ePO™)** — Active Response is built as an extension to McAfee ePO.
4. **Active Response extensions** — The user interface presents pages to run and save searches, create custom collectors, and define triggers and reactions. This interface also handles communication between ePolicy Orchestrator and Active Response service. The server extension manages internal metadata.
5. **Active Response server** — This is the brain of Active Response. The service provides the back-end functionality of Active Response.
6. **Active Response aggregator** — This optional component pre-processes communication between DXL brokers and the Active Response server. It reduces network bandwidth usage and allows for DXL fabric escalation.
7. **McAfee® Data Exchange Layer (DXL)** — DXL brokers and clients are the communication channel for Active Response operation.
8. **Managed endpoints** — Active Response supports both Windows and Linux endpoints.
9. **Active Response client** — The client is a McAfee® Agent plug-in that executes collectors, monitors triggers, and fires reactions on managed endpoints.

2 Installation

Active Response installation comprises these parts: Active Response and Data Exchange Layer extensions, the Active Response service, and Active Response and Data Exchange Layer endpoint clients.

Privacy notice

Active Response collects information from the network, such as user names, system names, IP addresses, and audit data. Access to this information is available in Active Response pages within 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.

This version of Active Response is not compatible with previous BETA versions. Remove all Active Response BETA versions from your environment before installing this version.

Installation components

Install each component in this order:

1. Data Exchange Layer extensions
   - DXL Broker Management
   - DXL Client for ePolicy Orchestrator
   - DXL Client Management
2. Active Response extensions
   - User interface extension: `mar-ui`
   - Server extension: `mar-server`
   - Client extension: `mar-client`
   - Product license: `mar-license`
   - Help extension: `mar-help`
3. Active Response server
4. Active Response aggregator
5. DXL client on managed systems
6. Active Response clients for Windows and Linux managed systems

Contents

Requirements

For a successful installation, check these minimum requirements are met before installing Active Response components.

Required server hardware

The server can be installed on a virtual machine if necessary.

The minimum recommended hardware for the Active Response server is:

- 4 Intel® Xeon® CPU X5675, 3.07 GHz
- 8 GB RAM
- 120 GB solid-state disk

Required service infrastructure

- McAfee® ePolicy Orchestrator® (McAfee ePO™) 5.1.1 or later
- McAfee® Agent 5.0 extension or later
- McAfee® Data Exchange Layer (DXL) 2.0.0.430 broker or later

Supported web browsers for user interface

Web browser	Version
Internet Explorer	9 or later
Chrome	17 or later
Firefox	10.0 or later

Required client infrastructure

- McAfee Agent 5.0.2.132 or later for Windows 10 endpoints
- McAfee Agent 5.0.0.2620 or later for Windows earlier than Windows 10
- McAfee Agent 5.0.0.2710 or later for Linux endpoints
- Data Exchange Layer 2.0.0.430 clients or later on all managed endpoints

For sizing information, see *Performance details* in the Active Response Product Guide.

Supported client operating systems

Operating system	Version	Architecture
Windows 10 Enterprise	Base	32-bit and 64-bit
Windows 8.0	Base	32-bit and 64-bit
Windows 8.1 Enterprise	Base, U1	32-bit and 64-bit

Operating system	Version	Architecture
Windows 2012 Server	Base, R2, U1	64-bit
Windows 2008 R2 Enterprise	SP1	64-bit
Windows 2008 R2 Standard	SP1	64-bit
Windows 7 Enterprise	Up to SP1	32-bit and 64-bit
Windows 7 Professional	Up to SP1	32-bit and 64-bit
CentOS	6.5	32-bit
RedHat	6.5	32-bit

See also

Install extensions on page 13
Install the service on page 14
Activate product license on page 21
Install clients on page 15

Install extensions

Install Active Response and Data Exchange Layer extensions.

For details about installing software using McAfee ePO, see the *McAfee ePolicy Orchestrator Installation Guide*.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Log on to ePolicy Orchestrator as an administrator.
2. Select **Menu** | **Software** | **Software Manager**.
3. Download Active Response and Data Exchange Layer extension .zip files, and save them to a temporary directory.
4. Select **Menu** | **Software** | **Extensions**. For each extension:
   a. Click **Install Extension**.
   b. Click **Browse** to locate and select an extension in the temporary folder.
   c. On the summary page, click **OK** to start the installation.

   If an older version is already installed, the extension is updated with the newer version.

See also

Requirements on page 12

Install the service

Active Response service is provided as an .iso image, packaging a McAfee Linux OS instance.

Task

1 Log on to McAfee ePO as an administrator.

2 Select **Menu** | **Software** | **Software Manager** and download the Active Response service ISO file.

3 Start the system where Active Response service will be installed, making sure that the it boots from the Active Response service ISO image. McAfee® Linux Operating System (MLOS) and all necessary packages are installed automatically after the system starts.

4 When installation finishes, restart the system. Make sure that it starts from the installed system, not from the .iso image.

5 Configure the Active Response service.

a Read the License Agreement and enter `Y` to accept its terms.

b Set a root password and confirm it.

c Create an operational account. You can use this account to connect through `ssh` to the system, and use `su` to obtain root permissions.

d Select the main network interface for the system. This interface connects Active Response service to McAfee ePO and the Data Exchange Layer.

e Configure the network interface.

- Type `D` for DHCP configuration.
- Type `M` to manually set the network addresses.

f Set a host name and domain name for the system.

g (Optional) Enable IPv6 routing.

h Set the time server for the system.

i (Optional) Set proxy variables. `http_proxy` and `https_proxy` definitions are comma-separated lists of host names or IP addresses. `no_proxy` definition is a comma-separated list of host names, domains, or IP addresses.

Proxy settings are for operating system administration only. Active Response does not use proxies for communication with McAfee ePO or network endpoints.

j Configure McAfee Agent to set up the connection to McAfee ePO.

k Select which services must run on the system.

- **DXL Broker** — installs a Data Exchange Layer broker.
- **AR Server** — installs the Active Response service.

l Set the DXL broker communication port.

6 Log on to McAfee ePO as an administrator.

7 Select **Menu** | **Configuration** | **Registered Servers**.

8 Add and configure a new server for the Active Response service.

You can only add the Active Response service to one McAfee ePO. Active Response does not support multiple McAfee ePO deployments.

a In **Server type**, select **Active Response Server**.

b In **Name**, enter a name for your Active Response server.

c In **Active Response Server Location**, enter `https://{server URL}}/mar/api`.

For details about setting up McAfee ePO servers, see the *McAfee ePolicy Orchestrator Product Guide*.

See also

Requirements on page 12

Install aggregators

You are not required to install an aggregator to use Active Response. However, aggregators improve escalation in the number of managed endpoints and reduce DXL bandwidth usage. We recommend that you install an aggregator on each system in your fabric that runs a DXL broker alone.

Before you begin

Install Active Response aggregators on DXL broker systems in your fabric.

These brokers must not have a DXL client deployed on the system. This means that aggregators can't be installed on Active Response or McAfee® Threat Intelligence Exchange (TIE) server systems.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1 Log on to ePolicy Orchestrator as an administrator.

2 Select **Menu** | **Software** | **Software Manager** and check in the Active Response Aggregator package.

3 Select **Menu** | **Software** | **Product Deployment**, then click **New Deployment**.

4 In the **Package** drop-down list, select the Active Response Aggregator.

5 Click **Select Systems** and choose the DXL broker where to install the aggregator.

6 Select **Run Immediately** and click **Save** to start deployment.

Manage Active Response clients

Use these tasks to manage clients on managed endpoints.

For details about installing software using McAfee ePO, see the *McAfee ePolicy Orchestrator Installation Guide*.

Install clients

Active Response clients are ready to function immediately after installation and configuration.

For details about product features, usage, and best practices, click **?** or **Help**.

Task

1 Log on to ePolicy Orchestrator as an administrator.

2 Select **Menu** | **Software** | **Software Manager** and check in the Active Response and Data Exchange Layer client packages.

3 Deploy Data Exchange Layer clients.

a Select **Menu** | **Software** | **Product Deployment**, then click **New Deployment**.

b Select the Data Exchange Layer client software package.

c Click **Select Systems** to select which endpoints to manage with Active Response.

d Select **Run Immediately** and click **Save** to start deployment.

If an older version is already installed, the Data Exchange Layer client is updated with the newer version.

4 Deploy Active Response clients.

During deployment on Windows systems, Active Response disables Microsoft Protection Service momentarily to complete installation. Endpoint users might see a warning that this service has been disabled. When installation is complete, Microsoft Protection Service is restored and the warning can be ignored.

a Select **Menu** | **Software** | **Product Deployment**, then click **New Deployment**.

b Select the Active Response client software package for Windows or Linux.

On Linux 64-bit systems, compatible 32-bit libraries must be installed on endpoints for Active Response to work properly.

c Click **Select Systems** to select which endpoints to manage with Active Response.

d Select **Run Immediately** and click **Save** to start deployment.

If an older version is already installed, the Active Response client is updated with the newer version.

After deploying the Active Response clients, make sure to configure the appropriate McAfee ePO policies.

See also

Requirements on page 12
Client configuration on page 23

Uninstall clients

Remove Active Response clients from endpoints.

For details about uninstalling software using ePolicy Orchestrator, see the *McAfee ePolicy Orchestrator Installation Guide*.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1 Log on to McAfee ePO as an administrator.

2 Select **Menus** | **System Section** | **System Tree**.

3 From the **Systems** tab, select the endpoints that are to be uninstalled. Then select **Actions** | **Agent** | **Run Client Task Now**.

4 Start a new client task to uninstall Active Response clients.

a Under **Product**, select **McAfee Agent**.

b Under **Task Type**, select **Product Deployment**.

c Under **Task Name**, select **Create New Task**.

d In **Target platforms**, select **Windows** or **Linux**.

e In **Products and components**, select the Active Response client package.

Select the latest Active Response package version if you have more than one version in your **Master Repository**.

f In the **Action** drop-down list, select **Remove**.

5 Click **Run Task Now**.

Install content packages

Install content packages to get new collectors and reactions, or new versions of existing built-in collectors and reactions.

New versions of collectors and reactions in the content package might turn some of your saved searches and triggers unusable. This only happens if the update changes a built-in collector output fields, or if the update changes built-in reaction arguments. Check the *Active Response Content Package Release Notes* for information on changes to collectors and reactions introduced by a content package.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1 Log on to ePolicy Orchestrator as an administrator.

2 Select **Menu** | **Software** | **Software Manager** and check in the desired version of Active Response content package.

After the package checks into the **Master Repository** it is installed automatically.

3 Upgrade

A complete upgrade installs new Active Response service, extensions and client packages.

To minimize down-time during the upgrade process, install components in this order:

- Active Response service: `ActiveResponseServer-{version}.zip`
- Active Response extensions: `mar-extensions-{version}.zip`
- DXL and Active Response clients on managed systems

Contents

Upgrade the service

Manage Active Response service update packages on McAfee ePO **Software Manager**.

For details about installing software using McAfee ePO, see the *McAfee ePolicy Orchestrator Installation Guide*.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Log on to ePolicy Orchestrator as an administrator.
2. Select **Menu** | **Software** | **Software Manager** and check in the Active Response Server package.
3. Deploy the update package.
   a. Select **Menu** | **Software** | **Product Deployment**, then click **New Deployment**.
   b. In the **Package** drop-down list, select the service update package.
   c. Click **Select Systems** to select the Active Response server in your network.
   d. Select **Run Immediately** and click **Save** to start deployment.

Once the update package is installed, see *Upgrade extensions* to continue.

Upgrade extensions

Upgrade Active Response extensions on McAfee ePO.

Before you begin

Active Response service of the same later version must be installed.

For details about installing software using McAfee ePO, see the *McAfee ePolicy Orchestrator Installation Guide*.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Log on to ePolicy Orchestrator as an administrator.
2. Select **Menu** | **Software** | **Software Manager** and download the Active Response Extensions .zip file. Save it to a temporary directory.
3. Select **Menu** | **Software** | **Extensions** and click **Install Extension**.
4. Click **Browse** to locate and select the extension in the temporary folder.
5. On the summary page, click **OK** to start the installation.

Once that the extensions are installed, see *Upgrade clients* to continue.

Upgrade clients

Install a later Active Response client version on managed systems to upgrade clients.

You can upgrade Active Response clients while they are online. As soon as the new version is installed, clients respond to the Active Response service.

To complete the upgrade, follow the instructions in the *Install clients* section.

4 Configuration

Configure Active Response extensions, service, and clients from McAfee ePO.

Contents

Activate product license

Active Response components are available to you as trial versions for a limited time. Follow these steps to continue using Active Response after the trial period.

Before you begin

Active Response extensions and service must be installed. These task is to license a previously installed Active Response deployment.

For details about installing software using McAfee ePO, see the *McAfee ePolicy Orchestrator Installation Guide*.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Download the product activation extension `mar-license.zip`, and save it to a temporary directory.
2. Log on to ePolicy Orchestrator as an administrator.
3. Select **Menu** | **Software** | **Extensions**.
4. Click **Install Extension**.
5. Click **Browse** to locate and select `mar-license.zip` in the temporary folder.
6. On the summary page, click **OK** to start the installation.

   Active Response is automatically activated after installing the extension.
7. On the **Registered Servers** page, check that Active Response is activated in the entry for Active Response server. Look under **MAR license** for the legend "MAR Server is licensed."

   If you see the **Activate license** button, Active Response encountered a problem connecting to Data Exchange Layer. Make sure that Data Exchange Layer is operational, then click **Activate license**.

See also

Requirements on page 12

Network ports

Active Response uses these ports for network connectivity.

Make sure your network settings are not blocking access to Active Response server and clients through these ports.

Table 4-1 Server ports

Port number	Open to	Incoming connections	Outgoing connections
443	Connect to extensions on the McAfee ePO server.	Yes	Yes
8883	Connect the DXL broker to the DXL client on the McAfee ePO server.	Yes	Yes
8081	Connect McAfee Agent to the McAfee ePO server.	Yes	Yes
22	Connect remotely through ssh to perform maintenance tasks.	Yes	Yes
123 UDP	Network Time Protocol	Yes	Yes

Table 4-2 Client ports

Port number	Open to	Incoming connections	Outgoing connections
8081	Connect McAfee Agent to a McAfee ePO server.	Yes	Yes
8883	Connect the DXL client to a DXL broker.	Yes	Yes

Service configuration

Configure how the Active Response service works.

Search execution time-to-live

Active Response search expressions execute collectors on managed endpoints. However, 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.

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. However, if you change the certificates used by McAfee ePO, use this configuration option to reset the certificate store in the Active Response server.

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 Active Response server and aggregators.

Make sure that McAfee ePO applied the appropriate tags to your systems after installation.

Client configuration

Use McAfee ePO policies to configure Active Response clients.

By setting Active Response policies in **Menu** | **Policy Catalog**, you can:

- Set the maximum number of results returned by search expressions.
- Enable endpoints to execute triggers.
- Enable **NetworkFlow** and **Files** collectors and triggers.
- Set database limits and maximum number of results returned by the **NetworkFlow** collector.
- Set database limits, maximum number of results returned, and files excluded by the **Files** collector.
- Enable system logging on managed endpoints.

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, **NetworkFlow** and **Files** collectors are disabled. All triggers are disabled too.
- **Full Visibility** — When this policy is enforced, **NetworkFlow** and **Files** collectors are enabled, but all triggers are disabled.
- **Full Monitoring** — When this policy is enforced, all collectors and triggers are enabled.

Access management

After installation, Active Response creates two 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.

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

Privacy notice

Active Response collects information from the network, such as user names, system names, IP addresses, and audit data. Access to this information is available in Active Response pages within 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.

5 Using Active Response

Use Active Response to search incidents, collect data, and trigger reactions.

Contents

Searching endpoint data

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

To avoid stressing the network, all searches time out automatically after a configurable amount of time. See *Service configuration* for more information.

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, SHA1, and MD5 values for all records returned by the **Processes** collector.

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

Search expression with filtered values

Get the name, SHA1, and MD5 values from the **Processes** collector, for processes files that have the ".exe" extension.

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

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

For details about product features, usage, and best practices, click **?** or **Help**.

1 Select **Menu** | **Systems Section** | **Active Response Search**.

2 In the **Search** box, enter a search expression.

See *Search syntax* for details about writing search expressions.

3 Click **Search** to start collecting data from managed endpoints.

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
```

See also

Search syntax on page 28
CurrentFlow collector on page 31
Files collector on page 32
HostEntries collector on page 33
HostInfo collector on page 33
InstalledUpdates collector on page 34
LocalGroups collector on page 35
NetworkFlow collector on page 35
Processes collector on page 37
UserProfiles collector on page 39
WinRegistry collector on page 39

Save a search expression

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

For details about product features, usage, and best practices, click **?** or **Help**.

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**.

See also

Search syntax on page 28

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

For details about product features, usage, and best practices, click **?** or **Help**.

1 Select **Menu** | **Active Response Catalog** | **Searches**.

2 Click the name of the search expression that you want to run.

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

See also

Search syntax on page 28

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 equal 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`
	`less equal than`	`where Files size less equal than 1024`
String 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"`
IP 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

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

Collector summary

A name and description identify each collector. 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

A collector's content specifies the code that Active Response executes on a managed operating system to collect data. See *Custom content* for information about content types and usage.

Collector output

The data returned by a collector is accessible through the collector's *output fields*. The output data fills the search results table after running a *search expression*. To create columns for the result table, a collector defines three attributes:

- **Name** — Sets a column header.
- **Type** — Specifies a data type for the values in the column. See *Literals* section in *Search syntax* for a list of available data types.
- **Show by default** — Sets the column to appear by default in the search results table.

Built-in collectors

Active Response provides several collectors, available out of the box after installation.

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.
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' 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 SHA1 hash code for the source process.

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
```

See also

Use the search box on page 26

DNSCache collector

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

Table 5-1 Collector output

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

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

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

Files collector

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

Table 5-2 Collector output

Field	Type	Description
name	String	The file name.
dir	String	The directory path where the file lives. 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/'`
full_name	String	The fully qualified file name, including its 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 SHA1 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.

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`.

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.

See also

Use the search box on page 26

HostEntries collector

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

Table 5-3 Collector output

Field	Type	Description
ipaddress	IP	An IP address set in the `hosts` file.
hostname	String	The host name mapping for the IP address.

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

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

See also

Use the search box on page 26

HostInfo collector

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

Table 5-4 Collector output

Field	Type	Description
hostname	String	The endpoint's host name.
ip_address	IP	The endpoint's first IP address
os	String	The endpoint's operating system version.

Find all endpoints with Windows operating system.

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

See also

Use the search box on page 26

InstalledDrivers collector

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

Table 5-5 Collector output

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

Table 5-5 Collector output *(continued)*

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. • 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.

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.

Table 5-6 Collector output

Field	Type	Description
description	String	The description for the update package.
hotfix_id	String	Microsoft knowledge base identifier for the update package.
install_date	Timestamp	The date when the package was installed.
installed_by	String	The user name that performed the installation, qualified by its namespace.

Show which hotfix packages where installed by bad_user.

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

See also

Use the search box on page 26

InteractiveSessions collector

The **InteractiveSessions** collector gathers information about ongoing interactive sessions on managed systems.

Table 5-7 Collector output

Field	Type	Description
userid	String	The log-in username.
name	String	The user's full name.

Show interactive sessions for user 'owilde'

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

LocalGroups collector

The **LocalGroups** collector gathers data on local system groups.

Table 5-8 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.

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

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

See also

Use the search box on page 26

LoggedInUsers collector

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

Table 5-9 Collector output

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

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.

Table 5-10 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.

Table 5-10 Collector output *(continued)*

Field	Type	Description
status	String	The status of the TCP transaction (not available in UDP transactions). 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.
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 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.
sha1	String	The SHA1 hash code for the source process.

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
```

See also

Use the search box on page 26

NetworkInterfaces collector

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

Table 5-11 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.
macaddress	String	The interface's MAC address.
name	String	The interfaces 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.

Processes collector

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

Table 5-12 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.
size	Number	The amount of resident RAM used by the process.
md5	String	The MD5 hash code for the process.
sha1	String	The SHA1 hash code for the process.
cmdline	String	The command that started the process.
imagepath	String	Path to the process' image name.
kerneltime	Number	The process' use of kernel mode CPU time, in seconds.
usertime	Number	The process' 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.

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
```

See also

Use the search box on page 26

Services collector

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

Table 5-13 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.

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.

Table 5-14 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.

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.

Table 5-15 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.

Show applications that start up automatically for user 'owilde'

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

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.
domain	String	The domain that holds the user.
fullname	String	The user's full name.
installdate	Timestamp	The date when the user was created.
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.
group	String	The group that contains the user account.

Find user accounts that have been locked out from endpoints.

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

See also

Use the search box on page 26

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.

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.

See also

Use the search box on page 26

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

See also

Create a custom collector on page 40

Create a custom collector

Specify what data to collect from endpoints with custom collectors.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Select **Menu** | **Systems Section** | **Active Response Catalog**.
2. Select the **Collectors** tab, then click **New Collector**.
3. Enter a name and description for the collector.
4. For either or both the **Windows** and **Linux** 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.

   Add content to both **Windows** and **Linux** tabs to run the collector on both Windows and Linux 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 Click **Save** to finish.

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

See also

Custom collectors on page 40
Adding custom content on page 49
Content output on page 49
Content types on page 52

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. This way you can find reactions easily in the **Active Response Catalog**.

Reaction content

A reaction's content specifies the code that Active Response executes on managed endpoints. See *Custom content* for information about content types and usage.

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. See *Literals* section in *Search syntax* for a list of available data types.

When a trigger is set to run a reaction, the trigger output fields are passed as values to the reaction content using arguments.

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, suppose that 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

Active Response provides several reactions, available out of the box after installation.

RemoveFile reaction

Use this reaction to delete files from endpoint filesystems.

Table 5-16 Arguments

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

KillProcess reaction

Use this reaction to kill proccess running on endpoints by passing the process' ID.

Table 5-17 Arguments

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

Create a custom reaction

Reactions execute custom content on managed endpoints.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Select **Menu** | **Systems Section** | **Active Response Catalog**.
2. Select the **Reactions** tab, then click **New Reaction**.
3. Enter a name and description for the reaction.
4. For either or both the **Windows** and **Linux** tabs, insert 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.

   Add content to both **Windows** and **Linux** tabs so that the reaction applies both to Windows and Linux managed endpoints.
5. Use the **Type** drop-down list to select the appropriate content type.
6. Click **Add Argument** or **+** to add an argument. An argument's name must match the name given between `{{` and `}}` in the reaction's content.
7. Enter a name for the argument.
8. From the **Type** drop-down list, select a type for the argument values.
9. Click **Save** to finish.

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

See also

Content arguments on page 51

Apply a reaction

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

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

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1 Select **Menu** | **Systems Section** | **Active Response Search**, then run a search expression.

2 When results appear in the **Search Results** table, select the rows you want to target.

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.

5 Click **Yes** to confirm.

See also

Built-in reactions on page 42

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 fire 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.

See *Trigger types* for detailes on each type of trigger.

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. Only those endpoints where your McAfee ePO administrator has granted access to you can set the trigger. 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.

Also, only users that have access to the same endpoints that you have can modify your triggers on those endpoints. In other words, users that 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.

Task

For details about product features, usage, and best practices, click **?** or **Help**.

1. Select **Menu** | **Systems Section** | **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.

   Be careful that the reaction you select doesn't recreate the condition that sets the trigger off. An infinite loop happens if when 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.

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

See also

Reacting to incidents on page 41

Trigger types

Active Response provides different trigger types to catch events on managed endpoints.

See also

Reacting to incidents on page 41

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 lives. 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/'`
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 SHA1 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 SHA1 hash `97eb5a5b721e28f9696729d14ef9d4076c9b4e2e`

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

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name. See *Search syntax* for more information.

File creation and hashing race condition

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 SHA1 hash of the file is reported with the event.

Triggers set to catch files over **FileCreated** events based on an MD5 or SHA1 hash can fail under this race condition: when a file large enough 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.) 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 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.
flags	String	One of TCP flags `ACK`, `SYN`, `RST`, `FIN`.

Field	Type	Description
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 SHA1 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
```

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name. See *Search syntax* for more information.

Port events

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

Event	Description
PortOpened (Windows only)	A port is opened for listening.
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 SHA1 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.
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' 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.
size	Number	The amount of resident RAM used by the process.
md5	String	The MD5 hash code for the process.
sha1	String	The SHA1 hash code for the process.
cmdline	String	The command that started the process.
imagepath	String	Path to the process' image name.
kerneltime	Number	The process' use of kernel mode CPU time, in seconds.
usertime	Number	The process' 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.

Match processes started by user "blackhat" with the SHA1 hash:
97eb5a5b721e28f9696729d14ef9d4076c9b4e2e

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

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name. See *Search syntax* for more information.

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. 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. All values must be expressed as `REG_DWORD` values.
valuetype	Win Registry String	The data type of the registry data.

Catch when the DisableAllTriggers key is set to 1 in the registry key path for Active Response configuration.

```
keypath equals "hkey_local_machine\software\mcafee\mar" and keyvalue
"DisableAllTriggers" and valuedata equals "1"
```

A trigger condition is like an Active Response search expression filter without the `where` keyword or the collector name. See *Search syntax* for more information.

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.

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.

See also

Create a custom collector on page 40
Create a custom reaction on page 42

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 ").

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.

See also

Create a custom collector on page 40

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.

See also

Create a custom reaction on page 42

Content types

Active Response supports several content types.

See also

Create a custom collector on page 40

Operating system commands

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

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)
```

See also

Create a custom collector on page 40
Create a custom reaction on page 42

Bash scripts

This content type executes a Bash script.

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

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
```

See also

Create a custom collector on page 40
Create a custom reaction on page 42

PowerShell scripts

This content type executes a PowerShell script.

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 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.

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
```

See also

Create a custom collector on page 40
Create a custom reaction on page 42

Python 2.7 scripts

This content type executes a Python 2.7 script.

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 local users on Windows endpoints.

```
#
# 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))
```

See also

Create a custom collector on page 40
Create a custom reaction on page 42

Backing up and sharing content

You can export Active Response content to a file in JSON format. Use the exported file to restore content after product upgrade or to share your collectors, triggers, and reactions with other Active Response installations.

To export and import content, look for **Export**, **Export all**, and **Import** in **Active Response Catalog**.

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.

Table 5-18 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.
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 libraries. The Active Response client is corrupted.	Redeploy Active Response client on endpoint.
4	MAR_E_INTERNAL_ERROR	Failed during process boot. The Active Response client is corrupted.	Redeploy Active Response client on endpoint.

Table 5-18 Generic errors *(continued)*

Code	Name	Description	Workaround
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. See *Search syntax* for more information.
7	MAR_E_MISSING_ARGUMENT	Failed to create McAfee ePO events.	Check Active Response server is operational.
8	MAR_E_INVALID_ARGUMENT	A collector or trigger failed to create McAfee ePO events due to an unsupported event ID.	Check Active Response server is operational.
9	MAR_E_REQUEST_TIMEOUT	A collector took too long to return results.	Reduce the execution time of your custom collectors.
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.

Table 5-19 Runtime plug-in errors

Code	Name	Description	Workaround
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 is empty or the query is missing the 'content' field.	Check content of collector or reaction.
259	MAR_E_MISSING_SCRIPT_ENGINE	A collector o 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.
261	MAR_E_SCRIPT_ENGINE_UNSUPPORTED	The Active Response server version is newer than the client package on the endpoint.	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.
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.

Table 5-20 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. See *Search syntax* for more information.
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.

Table 5-21 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. See *Search syntax* for more information.
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.
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.

Table 5-22 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.
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.

Table 5-23 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. See *Search syntax* for more information.
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.

6 Performance details

Determine the level of system core, memory, and storage for the best Active Response clients deployment.

What is measured

Active Response has performed tests on different endpoint-class systems to determine hardware requirements for client deployment. Simulations were used to gather reference metrics.

The following items were analyzed:

- File hashing warm-up — This is the initial file system hashing. It is performed before the **Files** collector or triggers can be executed on the endpoint.
- Search expression execution — This is a random Active Response search expression including the **Files**, **NetworkFlow**, and **Processes** collectors.
- Idle — This is a baseline, consisting of Active Response installed on the endpoint but executing no searches or triggers, just standing by, once the file hashing warm-up process has completed.

The following tests were ran for each of the analyzed items:

- Software installation — Rapidly installs and uninstalls applications on the test system.
- Microsoft Office usage — Opens an existing Word, Excel, and PowerPoint document, performs different user operations, then closes the applications. The test simulates keyboard and mouse input, copying and pasting text and graphics, scrolling, calculations, and full-screen presentations.
- Idle CPU usage — Checks the CPU load during idle times.
- Web browser — Opens the default web browser and navigates to a series of webpages hosted on a local IIS server.
- Cygwin compilation — starts a processor intensive task by compiling the open source Ghostscript project using Cygwin on a windows system.

Products tested

The following products were tested:

- Active Response client, extension and service 1.0.0.366
- McAfee Agent 5.0.0.2620
- McAfee ePO 5.1.1
- Data Exchange Layer 2.0.0.405
- McAfee® VirusScan® Enterprise 8.8.0.1445

Hardware profiles

The performance tests where executed on two different hardware profiles.

Profile		A	B
CPU	**Make**	Intel core-i7 2600K	Intel core-i7 2600K
	Cores	4	4
	Threads	8	4
	Hyper threading	Yes	No
	Cache	16 KB L1, 256 KB L2, 8 MB L3	16 KB L1, 256 KB L2, 8 MB L3
Storage	**Total**	1 TB	1 TB
	Used	~80 GB	~80 GB
	RPM	7200	7200
	Solid-state drive	No	No
Memory		4 GB DDR3 RAM	2 GB DDR3 RAM
Operating system		Microsoft Windows 7 Enterprise Service Pack 1 64 bits	Microsoft Windows 7 Enterprise Service Pack 1 64 bits

Results

These conclusions were drawn from running the performance tests:

- When Active Response is in an idle state, endpoint performance is not affected in a noticeable way.
- When Active Response is executing the file hashing warm-up process, other tasks that require disk access can be affected. During this warm-up process, Active Response requires between 5 and 15% of CPU time.
- When Active Response has triggers set and enabled, endpoint performance can be reduced for specific tasks. For example, software installations might slow down. However, this depends on the amount of triggers set. Overall, the execution of triggers consumes up to 4% of CPU time.
- Active Response searches do not affect endpoint performance significantly.

Index

M

N

P

R

S

T

U

W