



CHAPTER 8

Using Application Classes

Overview

Access control rules are application-centric. The application classes, those shipped with CSA MC and the ones you configure yourself, are the key to the rules you build as part of your security policies.

This chapter explains the application classes shipped with CSA MC and provides instructions for creating new static and dynamically defined application classes.

This section contains the following topics.

- [About Application Classes, page 8-2](#)
- [Processes Created by Application Classes, page 8-2](#)
- [Removing Processes from Application Classes, page 8-2](#)
- [Shell Scripts and Application Classes, page 8-3](#)
- [Built-in Application Classes, page 8-4](#)
- [Built-in Configurable Application Classes, page 8-7](#)
- [Configuring Static Application Classes, page 8-8](#)
- [Dynamic Application Classes, page 8-12](#)
- [Defining Dynamic Classes, page 8-13](#)
- [Configuring Dynamic Application Classes, page 8-14](#)
- [Configure an Application-Builder Rule, page 8-17](#)
- [Configure a Rule Using a Dynamic Application Class, page 8-21](#)

- [Create New Application Classes from Rule Pages](#), page 8-22
- [Application Class Management](#), page 8-23

About Application Classes

When you create rules, you must decide which applications are performing the operations you are allowing or denying as part of the rule. Once you know this, you configure the application as an "application class" in CSA MC and select it as part of your rule.

Application classes are groupings of application executable files that you combine under one name, generally as part of a File Set Variable, see [File Sets](#), page 9-10. For example, you can enter `netscape.exe` and `iexplore.exe` under the heading of Web Browsers. Then you can select Web Browsers in the application field for your rule and apply restrictions to the actions that both Netscape and Internet Explorer can perform on specified resources.

Processes Created by Application Classes

When applications are invoked, they often spawn other processes as part of the action they are performing. Therefore, when you create an application class, CSA MC gives you the option of including or excluding child processes created by the original applications you define as part of the application class (see [page 8-8](#) for details).

Removing Processes from Application Classes

Processes are part of a configured application class when they are running on the system. When the process stops running, the CSA MC application classification for that process also ends. Should the process begin again, it may or may not fall into the same application class depending on the process's behavior and on the definition of the application class. Therefore, all application classifications are ephemeral and are constantly being re-evaluated and classified on the system.

The application class configuration page lets you control how long a process maintains a certain application classification. In general, you do not have to specify a time frame. You should only put a time limit on an application

classification if you are configuring rules that require it for a particular reason. For example, you may want to create special process start rules for an application. The classification of the process could be configured to time out once the system is finished booting.

Shell Scripts and Application Classes

On UNIX systems, the agent allows control over shell scripts which satisfy both of the following conditions:

- the script begins with an interpreter string (e.g., `#!/bin/bash`)
- the script is executed directly on a command line, e.g., `"$foo.sh"`.

Therefore, if you have an application class "foo.sh", a process satisfying the above conditions becomes a member of that application class.

Note that a shell may be launched by various methods which do not meet those conditions, e.g., `"$. foo.sh"`, or `"$ cat foo.sh | /bin/sh"`. Note also that if you happen to have an application class for a script's interpreter -- say, `/bin/bash` -- when you invoke the script, the process becomes a member of the `/bin/bash` application class.

If a user has write access to the disk, and can execute commands, then using the name of a shell script in a rule to DENY actions may not make sense. For example, denying access by `foo.sh` to modify `/etc/hosts` does not improve the protection of `/etc/hosts` as the user could just run `'vi /etc/hosts'`. It would make more sense to deny everything access to a file, and then permit known good scripts access to that file.



Note

In general, with scripts such as perl scripts, the agent's ability to place the script in a configured application class depends upon whether the interpreter executes the script (via `exec`) or simply reads it. In the first case, the agent does recognize the script. In the latter case, it cannot.



Caution

If the user can copy a script (or re-implement it) to a file of their choice, then any Deny rules would be avoided.

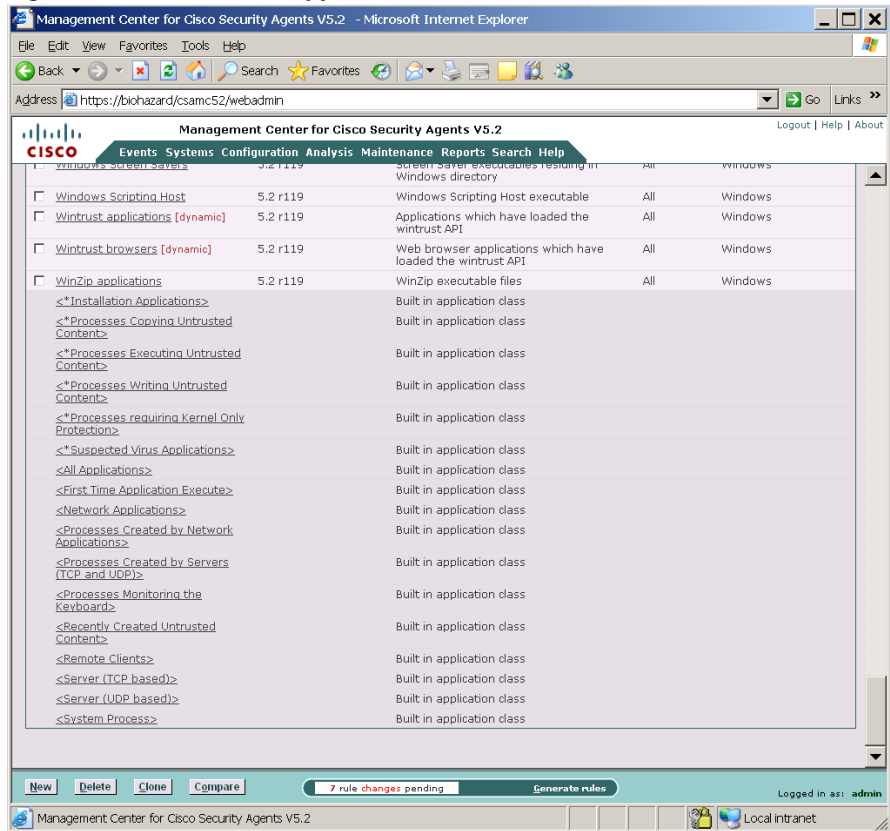
**Note**

On Windows, when writing rules for script application classes, you can create the rule for either the script itself or for the interpreter. (Scripts are handled by script interpreters.) If you write the rule for the interpreter, it will include the script handled by that interpreter.

Built-in Application Classes

CSA MC ships with several built-in application classes. Those application classes appear inside brackets (see [Figure 8-1](#)) in the rule application class selection list boxes. Some built-in class are also marked with asterisks. When there is an asterisk present, that indicates that the built-in class is configurable. See [Built-in Configurable Application Classes, page 8-7](#). You can view all application classes in the Application Class list page. Access this page from **Configuration>Applications** in the CSA MC menu bar.

Figure 8-1 Built-in Application Classes



Some included application classes are:

- **First Time Application Execute**—This includes the first invocation of any application which has never been observed to execute on the system.
- **Network Applications**—A network application would include any process that connects as a client or accepts a connection as a server and has in some manner accessed the network. The process would fall into this network application class after it has accessed the network. (This does not include applications that communicate only with other applications on the same system.)

- Processes created by Network Applications—This includes any process that is launched by a network application. For example, one network process may create another process that attempts to download code. This is one way viruses are propagated.
- Processes created by Servers (TCP and UDP)—This includes any TCP or UDP process invoked by a server (falling into the categories detailed in the two following bullet points).
- Server (TCP based)—This application class includes all processes that have accepted an inter-box connection on a non-ephemeral port.
- Server (UDP based)—This application class includes all processes that have accepted an inter-box connection on a non-ephemeral port.
- Processes Monitoring the Keyboard—This includes all processes which continuously monitor keystrokes over an extended period of time.
- Processes with elevated privileges—This application class is only available for UNIX rule types. It includes processes that have elevated user privileges for users other than root, such as ping. Using such processes is a common way to attempt a system break-in. Note that this elevated privilege designation does not apply to processes when the user is logged in as root.
- Recently created untrusted content—This includes executables that are newly created by <Processes writing untrusted content> and are immediately invoked.
- Remote clients—When a remote machine accesses resources over the network that are protected locally by an agent, the agent sees the remote access attempt as coming from a "remote application." The actual application that is used to open the resource in question cannot be determined on the local system. All remote access attempts are seen by the local system as being invoked by a remote application.

Therefore, if you are writing rules for a machine that other machines can access over the network, you must include <All Applications> or <Remote clients> as your application class. Otherwise, the rule will not work as expected in regard to remote access to those resources.

- System Process (available only in Network Access Control rules)—Using this application class, you can control network access for the operating system itself (as opposed to applications running on the operating system).

**Caution**

Any application class that you define does not include the system process. If you want to include the system process in a rule, you must select the included, built-in <All applications> or <System process> classes.

Built-in Configurable Application Classes

The Management Center for Cisco Security Agents also ships with built-in application classes that are built by policy rules. These application classes appear inside brackets with asterisks before them (*) in the rule application class selection list boxes (see [Figure 8-1](#)). This means that you should only use them in conjunction with a rule module that dictates the parameters that causes processes to become classified as one of these application types. CSA MC ships with pre-configured policies to define these classes. You can change these policies, if necessary.

- **Installation Applications**—This includes processes installing software.
- **Processes Executing Untrusted Content**—This includes any downloaded executable or any process that is interpreting downloaded content.
- **Processing requiring Kernel Only Protection**—This is intended to remediate interoperability issues with CSA's user component and other third party software products. Processes in this class will not enforce COM component checks and some buffer overflow checks.
- **Processes requiring OS Stack Execution Protection**—This application class is only available for UNIX rule types. This is intended to enable native Solaris operating system stack execution protection emulation. This enables additional buffer overflow protection.
- **Processes Writing Untrusted Content**—This is intended to identify processes that write executables which need to be treated as untrusted and tracked. e.g., This could identify a network application that downloads an executable and saves it to disk. The process is the network application and the untrusted content is the downloaded executable.
- **Suspected Virus Applications**:—This application class includes processes dynamically defined as being suspect by specified, exhibited behavior. Being classified as belonging to this application causes a quarantine message to be sent to CSA MC.

- **Third Party Security Applications:**—This application class is used to mark other security products which may attempt to control similar resources as CSA. This application class provides certain built-in permissions to facilitate interoperability and system stability.

Preserving Application Process Classes

You should be aware that all application process classes are preserved when your policies are changed if those processes (application classes) are used in an existing policy. For example, processes that have been classified by CSA MC as descendents or as network applications are preserved if the application classes that included them are changed in any way.

On policy changes, process name-based application classes are re-evaluated. Old application class memberships are not lost, only new memberships are gained.

Configuring Static Application Classes

Access control rules are application-centric. Meaning that when you write your rules, you should understand that the application(s) you select are really the heart of each rule. In your file, network, registry, and COM rules, you are controlling what applications can do to the files, addresses, registry keys, and COM components you specify. So, when you begin creating rules, think in terms of the applications your enterprise as a whole uses and the manner in which you want to limit an application's ability to perform undesired actions.

See also [Built-in Application Classes, page 8-4](#).

To create an application class, do the following:

-
- Step 1** Move the mouse over **Configuration** in the menu bar and select **Applications>Application Classes** (Windows or UNIX) from the drop-down list that appears. The list of existing Application classes is displayed. CSA MC ships with several pre-configured applications. Some Application classes appear within brackets. These are built-in CSA MC application classes and you cannot edit them.
 - Step 2** Click the **New** button to create a new application class. This takes you to the application class configuration view (see [Figure 8-2](#)).

- Step 3** Enter a **Name** for the application class you are creating. It is important to use a descriptive name that you can easily recognize in the application selection list that appears in the rule views.
- Step 4** Enter a **Description** for your application class. This description becomes visible in the application class list view.
- Step 5** **Operating System**—When you create an application class, you must select to either create a UNIX or a Windows application class. Your application class is then designated for all UNIX or all Windows platforms. Optionally, you select to target an operating system more narrowly by selecting a specific UNIX or Windows operating system from the **Target** pulldown menu.
- Step 6** If your lists of application classes are growing too long in rule pages, clicking the **Display only in Show All mode** checkbox on an application class page causes that item to no longer appear in list pages and selection lists. This feature works in conjunction with Admin Preference settings. You must go the Admin Preferences page to make the item reappear. See [Configuring Role-Based Administration, page 2-5](#).
- Step 7** Under **Add process to application class**, for a static application class, do the following:

Leave the default **when created from one of the following executables** radio button selected. Then enter the executable file names (one per line) for the applications you are grouping together in this application class.

See [Configuring Dynamic Application Classes, page 8-14](#) for details on that feature.



Note You can enter preconfigured File Set variables in the executables edit field by clicking the **Insert File Set** link. To learn more about File Sets, see [File Sets, page 9-10](#).

- Step 8** **Remove process from application class**—Select the checkbox beside **After** and enter a time frame in **seconds** to configure an application classification which expires after a period of time. Only use this feature if you have a rule that requires it. In general, you do not have to specify a time-out for application classifications. See [Removing Processes from Application Classes, page 8-2](#) for details.

For UNIX application classes, you have the additional option of selecting the **When session association is voided** checkbox. Selecting this checkbox causes the application classification to be removed when a process disassociates itself

from the current TTY session. For example, when an application class exists for applications descended from "superuser", you might not want the process to continue having the application class of the superuser shell.

Step 9 When applications are invoked, they often spawn other processes as part of the action they are performing. When you create an application class, select one of the following radio buttons to determine when processes spawned by the applications in the application class are also included.

- Only this process
- This process and all its descendents
- Only descendents of this process

(Creating an application class for "Only descendents of this process" is useful when making exceptions to a rule that is written for the main process itself. For example, you can write a rule allowing IIS to talk on the network, but create another rule denying descendents of the IIS process from talking on the network.)

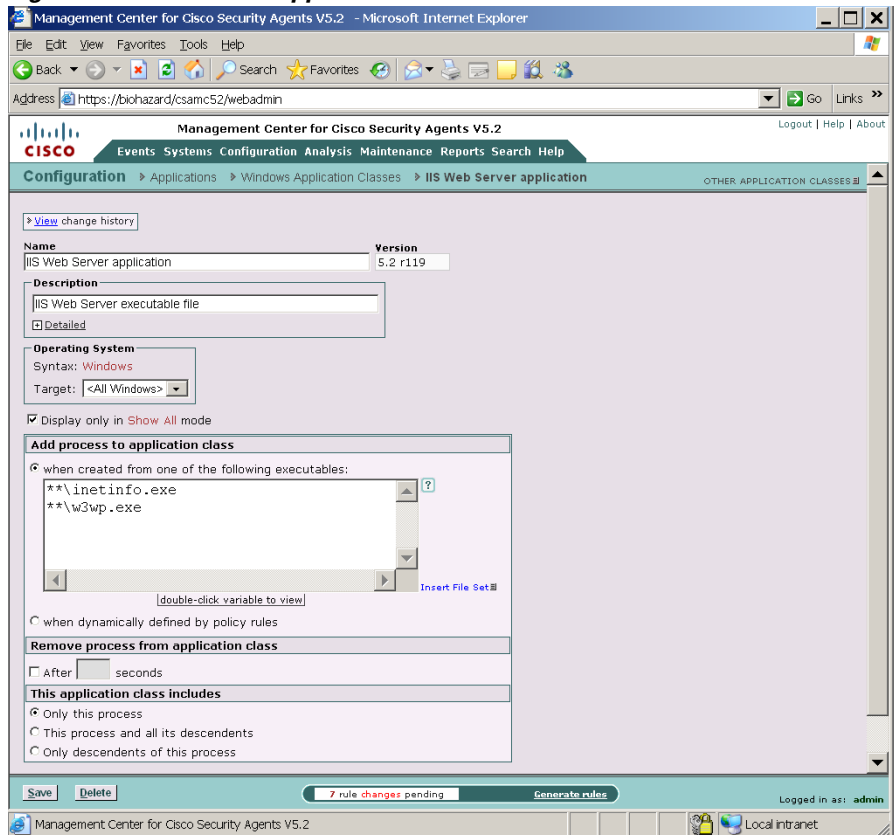
Step 10 When you are finished, click the **Save** button. This application class name, IIS Web Server application, now appears in the application list view and in the application selection fields for rule configurations. When you select it in a rule, you are indicating all the executables that comprise it.



Note You can use the Compare button in the Application Class list view to compare and merge similar application classes. See [Comparing Configurations, page 5-28](#) for details on using the Compare tool.

See [Dynamic Application Classes, page 8-12](#) for information on that application class type.

Figure 8-2 Static Application Class



19152

Dynamic Application Classes

The configurable application classes described in the previous pages are considered static application classes. Basically, in a static application class, a process is added to the class based on the name of its executable file (or the process name). Alternatively, you can build an application class based on an application's behavior rather than by a specific application executable name. This would be a dynamic application class defined by process behavior on a system. There are already built-in dynamically defined application classes in CSA MC. For example, the <Processes executing untrusted content> application class is a "built-in" dynamically configured class.

One example of an instance in which you might need a dynamic application class would be if you are writing rules for email clients but you do not know all the different email applications that are being used throughout your corporate network. In this case, you could use a dynamic application class. Any process appearing to act as a client for SMTP (you can use whatever criteria you decide to define what an email application is) would fall into a dynamic email application class that could be used in rules quarantining dangerous email messages.

Building Classes as Rule Consequences

You can also build a dynamic application class as a consequence of rules triggering. This way, for example, you can configure a query user rule in which a process is added to an application class as a result of a specific user response (yes, no, terminate). For example, you can build a "suspected virus" application class based on the end user being queried when untrusted content arrives on the desktop and the Terminate button on a query box is clicked to disallow it. But if the user clicked Yes to allow it, the process would not be added to the suspected virus application class.

Removing Processes from Classes

You can also use a dynamic "remove process" capability in conjunction with dynamically adding a process. For example, you can dynamically add a process to a "suspicious web server descendents" class if a web server spawns a process. Then, if that spawned process attempts to read a script from a normally accessed directory, you can decide this isn't a dangerous process and have the process

removed from the class after the attempt. But if the spawned process attempts to read a script from a directory it should not be accessing, the process should remain in the suspicious web server descendents class.

Defining Dynamic Classes

**Note**

A dynamically defined application class can be used in any rule where a static application class can be used.

Define a dynamic application class by doing the following:

- Create a new application class and select the **Processes dynamically defined by policy rules** radio button. (Do not enter any process names in the Application class page edit field.)
- Configure an application-builder rule to define your dynamic application class.

**Note**

Configuring the dynamic application class is only the first step. It does not become populated by processes until it is selected in a rule that will be used to define it.

For example, create a new File access control rule and select **Add process to application class** from the pulldown list as the rule action. Then choose the name of the dynamic application class (created in the first bullet point) from the pulldown list. Configure the remaining rule parameters. This rule type takes precedence over all others in the policy, but it does not override other rules in the policy the way allow, deny, and query rules do when triggered.

- Configure another rule to control the actions of this dynamic application class. As processes are added to this dynamic application class, those same processes will be used in all other rules in which the dynamic class is selected.

The following section provides an example of defining and using a dynamic application class in a policy

Configuring Dynamic Application Classes

Continuing to use the email client example, we will create an application class that will be dynamically populated by email client applications. You might want to do this if you are writing rules to protect email applications, but you do not know what email applications are being used across your network. Using this dynamic class, rules will restrict email clients based on detected behavior, such as using SMTP to access an email server, rather than by explicitly defining email application executables.

To create a dynamic application class, do the following:

-
- Step 1** Move the mouse over **Configuration** in the menu bar and select **Applications>Application Classes** (Windows or UNIX) from the drop-down list that appears. The list of existing Application classes is displayed.
 - Step 2** Click the **New** button to create a new application class. This takes you to the application class configuration view (see [Figure 8-3](#)).
 - Step 3** Enter a **Name** for the dynamic application class you are creating. It is important to use a descriptive name that you can easily recognize in the application selection lists that appear in the rule views.
For this example, we will create a new dynamic class called *Email clients_dynamic*. We will use this class to determine what email client applications are running on systems. Then we will add this dynamic class to an existing email quarantine rule.
 - Step 4** Enter a **Description** for your application class.
 - Step 5** If your lists of application classes are growing too long in rule pages, clicking the **Display only in Show All mode** checkbox on an application class page causes that item to no longer appear in list pages and selection lists. This feature works in conjunction with Admin Preference settings. You must go the Admin Preferences page to make the item reappear. See [Configuring Role-Based Administration, page 2-5](#).
 - Step 6** Under **Add process to application class**, for a dynamic application class, do the following:
 - Select the **when dynamically defined by policy rules** radio button. (Do not enter any process names in the edit field.)

**Tip**

When you use a dynamic application class in rules to define it, those “Defining rules” for the particular application class are accessible by clicking the link beside the **when dynamically defined by policy rules** radio button.

Step 7 Remove process from application class: Select the checkbox beside **After** and enter a time frame in **seconds** to configure an application classification which expires after a period of time. Only use this feature if you have a rule that requires it. In general, you do not have to specify a time-out for application classifications. See [Removing Processes from Application Classes, page 8-2](#) for details.

For UNIX application classes, you have the additional option of selecting the **When session association is voided** checkbox. Selecting this checkbox causes the application classification to be removed when a process disassociates itself from the current TTY session. For example, when an application class exists for applications descended from "superuser", you might not want the process to continue having the application class of the superuser shell.

Step 8 When applications are invoked, they often spawn other processes as part of the action they are performing. When you create a dynamic application class, you can select one of the following radio buttons (just as you can when you create a static application class) to determine when processes spawned by the applications in the dynamic application class are also included.

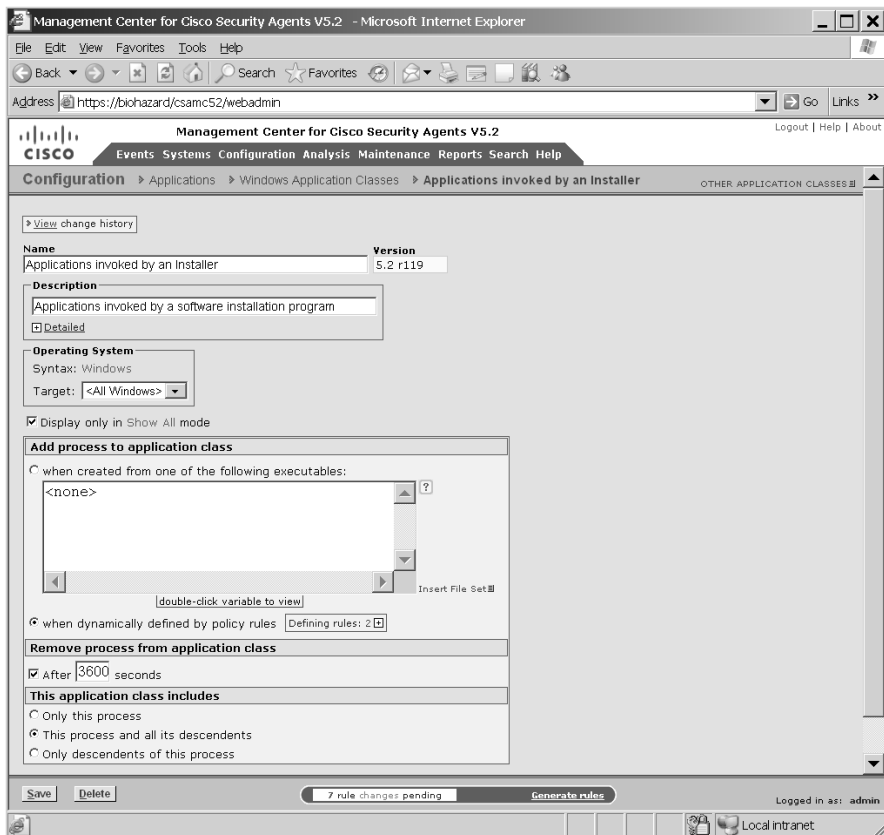
For this example, we will leave the default, Only this process, selected.

- Only this process
- This process and all its descendents
- Only descendents of this process

Step 9 When you are finished, click the **Save** button. This dynamic application class name now appears in the pulldown list beside the **Add to application class** radio button in access control rules and in all application selection fields.

Next we will use this dynamic class in an application-builder rule that will define the class.

Figure 8-3 Dynamic Application Class



191560

Configure an Application-Builder Rule

In this example, we are going to use a Network access control rule to define our dynamic application class. You can use any access control rule type as your application-builder rule. We are adding this rule to the Desktop Module that ships with CSA MC. (Remember, your dynamic application class is not populated with applications until an application-builder rule is triggered by the process's behavior and added to the class.)



Note

Defining dynamic application classes from the Application control rule is a bit different than creating them from other rule types. Because this rule has two application class fields, you can choose to add the current application to the dynamic class or choose to add the new application that is invoked by the first application to the dynamic class.



Caution

Dynamic application class process membership is temporary and is based on a running process meeting the criteria in the application-builder rule. When the process is no longer running on a system, it is no longer included in the dynamic class.

To prevent errors or unexpected behavior, you should avoid selecting the dynamic application class for a rule within a policy that does not also include the corresponding application-builder rule. Both the application-builder rule and the subsequent rule(s) that use the dynamic application class should co-exist within the same policy—although this is not required.

Step 1

To configure the application-builder rule which will dynamically create a new Email client class, access the Desktop rule module and click the **Modify rules** link.



Note

This is only an example. This is not intended to recommend that you add this rule to the Desktop module. This simply shows you what you can do if you do not know all the Email clients being used on systems across your network.

Step 2 Click **Add rule** and select **Network access control**.

Step 3 In the Network access control rule, configure the following (see [Figure 8-4](#)):

- Enter a description
- Select the **Add process to application class** radio button. Select the dynamic application class, *Email clients Dynamic*, from the corresponding pulldown list.



Note This rule type takes precedence over all other types but it does not override them. The only action of this rule is to build the application class for any subsequent rules within the policy that make use of it.

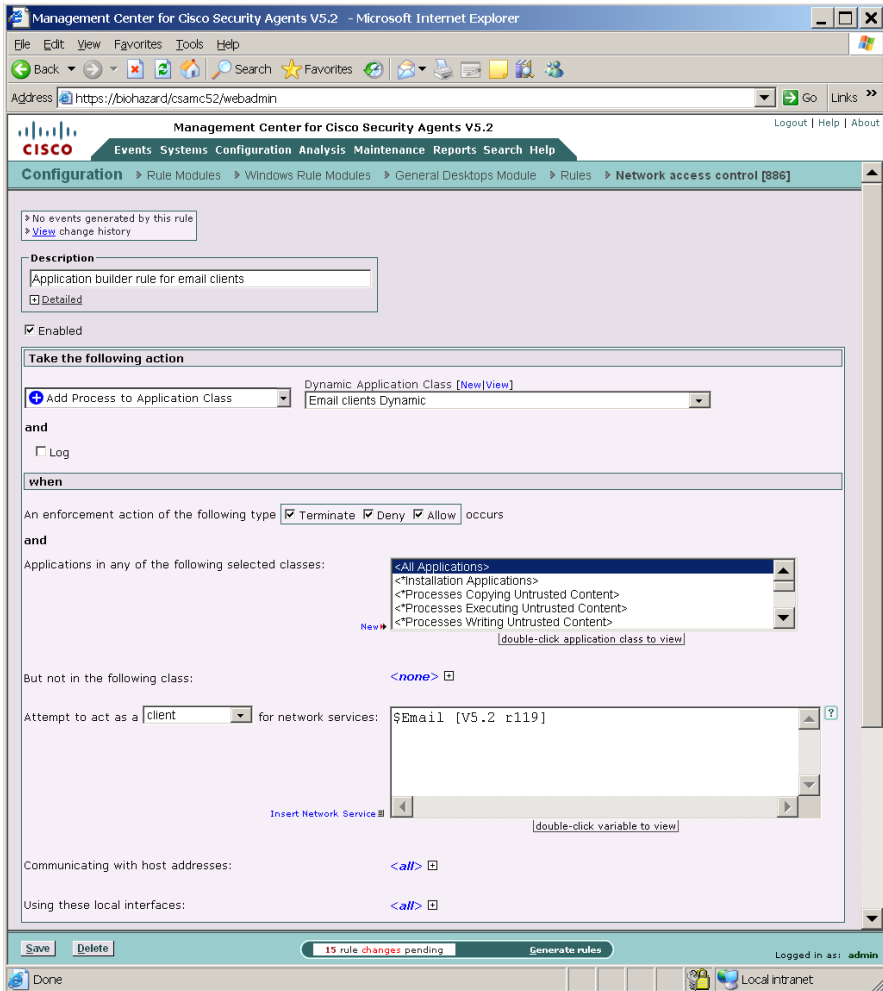
- **When—An enforcement action of the following type occurs.** Optionally, you can select one more of the available checkboxes. (Terminate, Deny, Allow) All entries are selected by default meaning that the tag will apply when the request is made regardless of the action that occurs. All actions apply. If you make a specific selection here, you are determining to create your dynamic application class based on that action occurring when the request is made (perhaps via another configured rule). You should note that all resource requests always result in either an allow, deny, or terminate occurring. Even if there is no rule governing the resource, for example, the implicit action is allow. See [Building Classes as Rule Consequences](#), page 8-12.
- Leave the default, **<All Applications>**, selected in the Application class field. This way, all applications that trigger the rule have the potential of being added to the dynamic class. You could select another application class here if you only want specific applications to fall into the dynamic class.
- Select **client** from the pulldown list and select the pre-configured variable, \$Email, from the list of configured Network services.
- Leave the default of 0.0.0.0-255.255.255.255 entered in the host addresses field.
- Leave the default of 0.0.0.0-255.255.255.255 entered in the Use these local addresses field.

Step 4 Click **Save**.

Now, based on the application-builder rule we've just configured, any application which uses the network services, SMTP, POP3, IMAP3 or IMAP2 as a client to access any system on the network, will fall into the Email clients_dynamic application class.

Next we will select this dynamic application class in a rule within this same policy.

Figure 8-4 Application-Builder Rule



191547

Configure a Rule Using a Dynamic Application Class

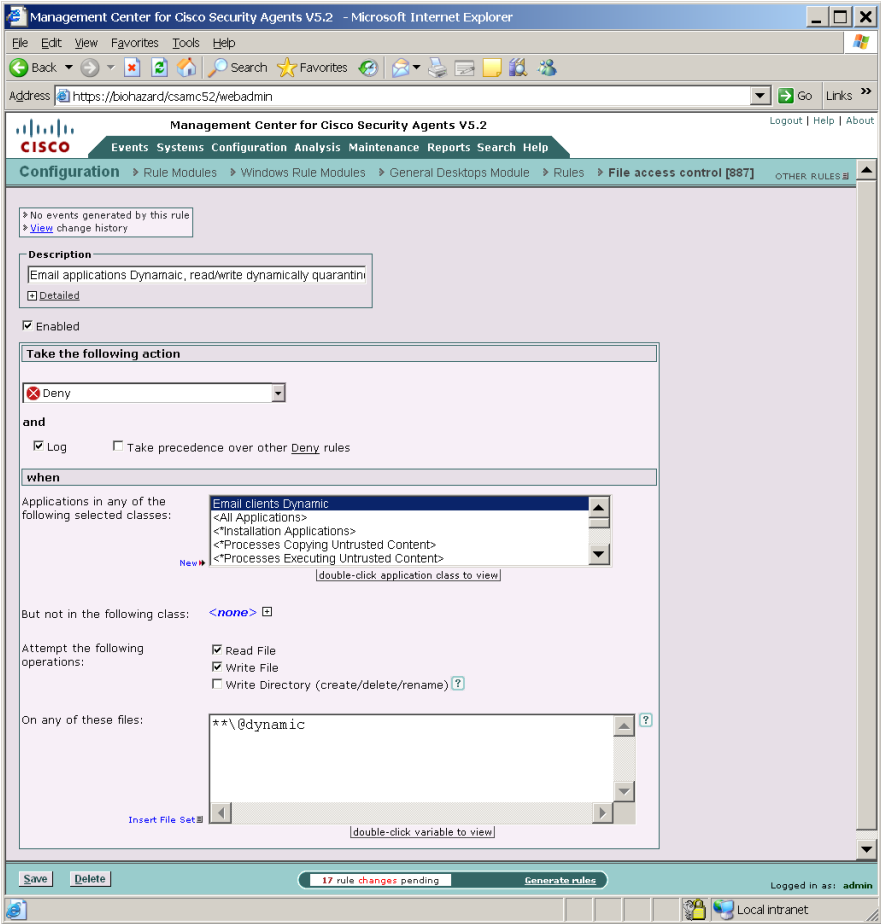
In this example, we are going to use a File access control rule to control the actions of a dynamic application class.

-
- Step 1** Configure this rule in the same manner in which you configure any other rule. For this example, once again access the Desktop Module policy (This is the policy already containing our application-builder rule.) and click the **Modify rules** link.
- Step 2** Click **Add rule** and select **File access control**.
- Step 3** In the File access control rule, configure the following (see [Figure 8-5](#)):
- Enter a description
 - Select the **Deny** radio button.
 - Select the dynamic application class, **Email clients Dynamic**, in the Application class list box.
 - Select the **read file** and **write file** checkboxes.
 - Enter **@dynamic** in the files field.
- Step 4** Click **Save**.

This rule will prevent any email application that falls into the selected dynamic email client class from reading or writing any dangerous, quarantined files.

Create New Application Classes from Rule Pages

Figure 8-5 Rule with Dynamically Defined Application



191551

Create New Application Classes from Rule Pages

You can create a new application class from a rule page and have that application class be available to the rule you’re currently configuring and to all other rules as well.

From the rule page, click the **New** link beside the Application class selection field to access configuration window. Configure your new application class and click **Save**. It is now available for selection in the rule page.

Also available for Application classes from the rule page, is the ability to view the configuration parameters for a selected application class. Double-click an application class in the rule page to view its configuration page.

Application Class Management

The Application Class Management page (available from the **Configuration** option in the menu bar) allows you to pare down the application class selection fields in the rule pages and in the Analysis feature pages. If you have a long list of application classes and you only want to view specific classes in rule configuration pages or only view them in the rule pages or only view them for analysis, you can choose to have application classes appear or not appear in features you select.

Note that selecting certain application classes to not appear in certain products does not delete those application classes. They will still appear in the main Application Class list page. They simply will not appear in the application class selection fields in the feature in question. By default, all application classes appear in all application class fields in all feature sets.

To enable or disable an application for general configuration or for analysis purposes, do the following:

Step 1 Move the mouse over **Configuration** in the menu bar and select **Applications>Application Class Management** from the drop-down list that appears. In the Application Class Management page (see [Figure 8-6](#)) there are swap box fields for CSA MC and for Application Behavior Investigation and Application Deployment Investigation.

The application classes appearing in the white swap box(es) (the bottom swapbox for each category) are enabled for the feature in question. Those appearing in the gray swap box(es) (the top swapbox for each category) will not appear in the feature in question.

Step 2 Select an application class and click the up arrow or down arrow buttons to move the selected class to the other swap box. This action enables or disables the application for the product. (It does not delete the application class.)



Note

Use the "Show [All, UNIX, Windows] application classes that apply to [<All features>, Management Center for Cisco Security Agents, Application Behavior Investigation, Application Deployment Investigation]" to narrow the application class categories to specific product components.

Figure 8-6 Application Class Management Window

