



# CHAPTER 4

## Building Policies

---

### Overview

The policies you create on the Management Center for Cisco Security Agents should reflect a well-planned, enterprise-wide security policy. If your corporate enterprise already has security guidelines in place, the rules that form your policies should reflect those guidelines.

It is important that you spend time charting out your security needs in advance rather than attempting to backfill holes as they are discovered. Because both networks and network security are dynamic entities, it is expected that you will need to adjust policies to meet the changing and growing needs of your enterprise. A well thought-out security plan is certain to save you time in the end.

This section contains the following topics.

- [Developing a Security Policy, page 4-2](#)
- [Combining Policies, page 4-5](#)
- [Making a Policy Mandatory, page 4-6](#)
- [Building Policies and Rule Modules, page 4-7](#)
- [Configure a Policy, page 4-7](#)
- [Attaching Rule Modules to Policies, page 4-10](#)
- [Attaching Policies to Groups, page 4-11](#)
- [Overall Policy Methodology, page 4-14](#)
- [Analyzing Applications, page 4-14](#)

- [Configuring Policies—The Methodology, page 4-15](#)
- [General Server Policy, page 4-17](#)
- [Sample Web Server Policy, page 4-18](#)
- [Combined General Server and Sample Web Server Policies, page 4-20](#)

## Developing a Security Policy

If you are crafting your own policies, please refer to [Overall Policy Methodology, page 4-14](#) for information.



### Caution

---

To maintain the integrity of the preconfigured policies shipped with CSA MC, it is recommended that you do not change them. If you are using preconfigured policies but want to edit them slightly due to your own site's needs, you should instead clone the policy in question or create a new policy and add it to the group. Note that each pre-configured rule, rule module, policy, and group page has data in the expandable **+Detailed** description field explaining the item in question. Read the information in these fields to learn about the items described and to determine if the item in question meets your needs for usage.

---

A corporate security policy should temper business concerns with security concerns. It should allow the user community to access required resources, while protecting that community from the dangers those resources can introduce. To achieve this goal, it is crucial to have a carefully planned network security policy in place to safeguard valuable organizational resources and information.

Before configuring your policies, it is important to understand exactly what network resources and services you want to protect and what threats you are most concerned about. The first step in planning a security policy is identifying the resources your user community requires to do business. That could include specific applications, protocols, network servers and web servers. Collect this information and use it to design the main features of your policy.

## Providing Safe Access to Required Resources

As you determine the network resources that are required by your user community, you can identify some of the threats posed against those resources. For example, while putting together a security plan, you might find it beneficial to limit access to some resources based on various parameters such as traffic direction and allowed file types.

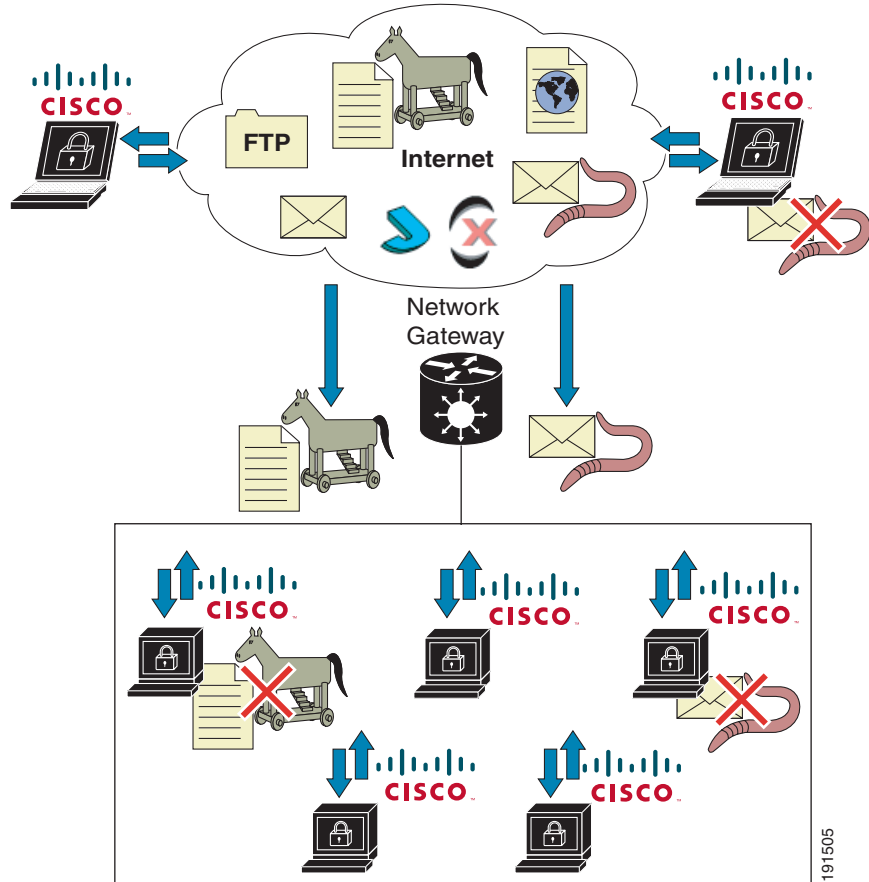
Upon examining past breaches of security, you could determine that email attachments and Internet file downloads pose the greatest threat to your network. In this case, you would want to develop policies to diminish the danger of accessing these particular resources. Your security plan should then incorporate policies for commonly used services such as Web, email, and instant messengers.

You could take a couple of approaches to enforcing your security plan depending upon the immediacy of any perceived threats and your basic corporate philosophy toward security. Both approaches are equally valid. On the one hand, you might choose to allow most activities and selectively add targeted restrictions. This would be a more permissive security model. This approach facilitates uptime, but may be less secure. Conversely, you could decide to shut everything down and then slowly add targeted permissions. This approach is far more restrictive and some legitimate requests could be rejected, but this may be suitable for highly secured environments. You could use both approaches for different groups.

As your security plan evolves, you can refine your policies, making them more or less granular to keep pace with your user community's needs. Your network system security depends on your implementing security policies carefully, and checking to see that they work as intended.

Figure 4-1

## Protecting Information



Formulate a policy to protect systems from common email worms and Trojans. Once these attacks infiltrate your network and propagate to the user community, a well-defined policy can identify errant system actions and stop an attack before it can damage mission-critical information.

# Combining Policies

You can attach multiple rule modules to single policies and you can attach multiple policies to a single group. Moreover, a host can belong to multiple groups and inherit policies from all of them. For example, a desktop can belong to the Desktop-All types group and inherit the Systems-Test Mode policy. It can also belong to the All group through which it receives the Remote Systems Policy.

When more than one policy is associated with a host, the rules modules in the individual policies are merged as though they were all defined within a single policy. In particular, the rules in the policy are ordered in the same sequence as they would be within a single module. See the section on [Rules: Action Options and Precedence](#), page 5-38 for priority order information.

[Figure 4-2](#) displays the relationship between host, group, policy, and rule module configuration items. In the diagram, you can see that the policy level is the common ground by which host groups acquire the rules that make up their security policy.



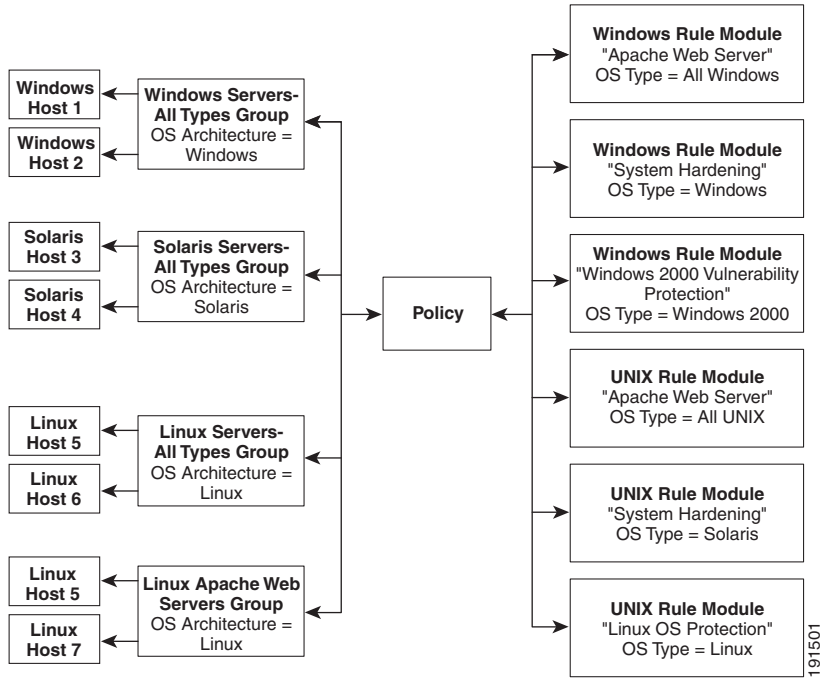
---

**Note**

You can view merged policy rules at both the group and host levels.

---

Figure 4-2 Host, Group, Policy, Rule Module Associations



## Making a Policy Mandatory

CSA MC provides three auto-enrollment architectural groups (Windows, Solaris, Linux) that are mandatory for all hosts of a given OS architecture. By providing group auto-enrollment for hosts, any policies you attach to these groups also become mandatory by association. You might want to use these mandatory groups to apply policies which prevent some critical service from being inadvertently banned. For example, you could attach policies to prevent DNS or DHCP from being disabled by an overly restrictive rule.

# Building Policies and Rule Modules

When you configure a policy, you are combining multiple rule modules under a common name. That policy name is then attached to a group of hosts and it uses the rules that comprise the policy to control the actions that are allowed and denied on those hosts. You can have several different types of rules in a rule module and consequently within one policy.

The policy level is the common ground by which host groups acquire the rules that make up their security policy. If you are configuring your own policies, you should begin by understanding the purpose of your policy and how you must build your rule modules to meet your needs. It's recommended that you build your policies from the top down. In other words, configure items in the following manner:

- a. Decide what purpose the policy serves.
- b. Understand what tasks the rule modules that comprise your policy must accomplish.
- c. Decide what rule types you must configure to accomplish the tasks you've isolated.

## Configure a Policy

Generally, when you configure a policy, you are combining multiple rule modules under a common name. That policy name is then attached to a group of hosts and it uses the rules that comprise the policy to control the actions that are allowed and denied on those hosts. You can have several different types of rules in a rule module and consequently within one policy.

The policy level is the common ground by which host groups acquire the rules that make up their security policy. You can attach rule modules of differing architectures to the same policy. This way, you can configure task-specific, self-contained, inclusive policies across all supported architectures (Windows, Solaris, Linux) for software that is supported on all platforms.

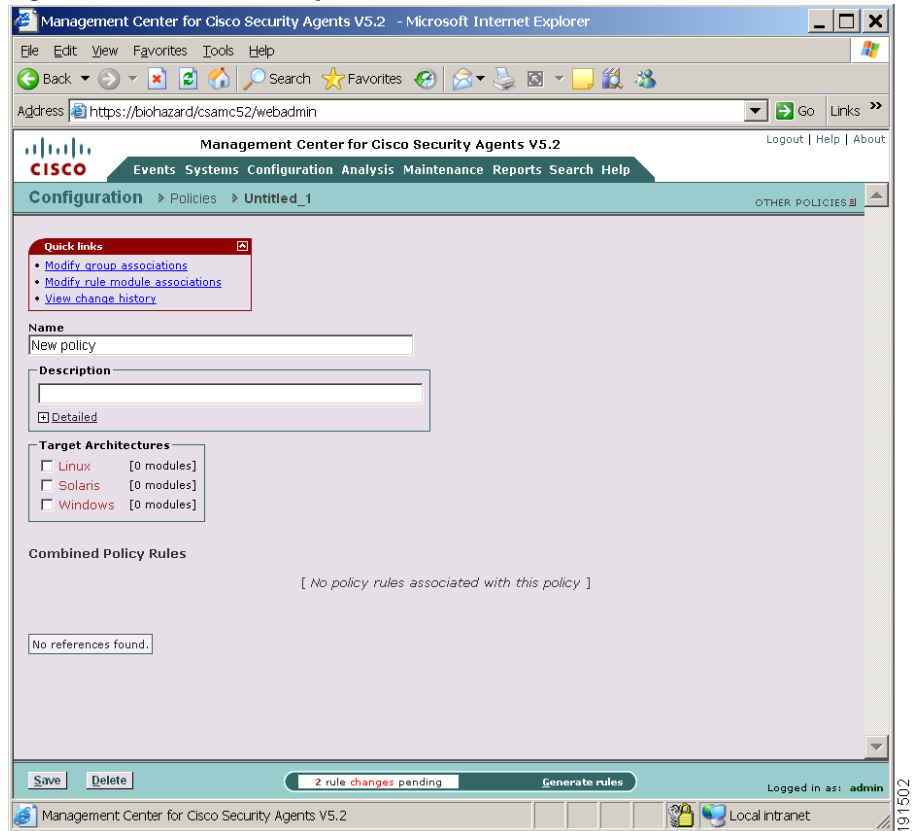
**Note**

Management Center for Cisco Security Agents ships with preconfigured policies you can use if they meet your initial needs. If you use a preconfigured policy, you do not have to create your own policy as detailed in the following pages.

To configure a policy, do the following.

- 
- Step 1** Move the mouse over **Configuration** in the menu bar of CSA MC and select **Policies** from the drop-down menu that appears. The policy list view appears.
- Step 2** Click the **New** button to create a new policy entry. This takes you to the policy configuration page.
- Step 3** In the available policy configuration fields, enter the following information:
- **Name**—This is a unique name for this policy grouping of rule modules. Names are case insensitive, must start with an alphabetic character, can be up to 64 characters long and can include alphanumeric characters, spaces, and underscores.
  - **Description**—This is an optional line of text that is displayed in the list view and helps you to identify this particular policy.
- Step 4** Select one or more **Target architecture** types for the policy. You can have one policy, for example - an Apache Web Server policy, and have all three architecture checkboxes selected. This way, each architecture specific rule module for Apache can be attached and deployed through one single Apache policy.
- Step 5** Click the **Save** button.
- This policy is empty until you attach configured rule modules to it.

Figure 4-3 New Policy



191502

# Attaching Rule Modules to Policies

When you configure a rule module, you are combining access control rules and/or tagging and monitoring rules under a common name. That rule module name is then attached to a policy. That policy uses the rules that comprise the module to control the actions that are allowed and denied on hosts. See [Configuring Rule Modules, page 5-5](#).

CSA MC gives you the option of attaching a rule module to a policy using the **Modify policy associations** link in the Rule Module configuration page or attaching a policy to a rule module using the **Modify rule module associations** link in the Policy list view page.

To attach a rule module or rule modules to an existing policy using the **Modify policy associations** link in the rule module configuration page, do the following.

- 
- Step 1** Attach a rule module to a particular policy by accessing that rule module's edit view. From **Configuration** in the menu bar, click on **Rule Modules** for the OS type you want to access the list view for those modules.
  - Step 2** From the rule module list view, click the link for the rule module you want to attach to a policy. This brings you to that rule module's edit view.
  - Step 3** From the edit view, click the **Modify policy associations** link. This takes you to a page containing swap boxes. See [Figure 4-4](#). The left box contains the policies the rule module is not attached to. The right box contains policies that the rule module is attached to.
  - Step 4** To add this rule module to an existing policy, select the rule module in the left box and click the **Add** button. The selected rule module moves to the right box and is now attached to the policy.

**Note**

---

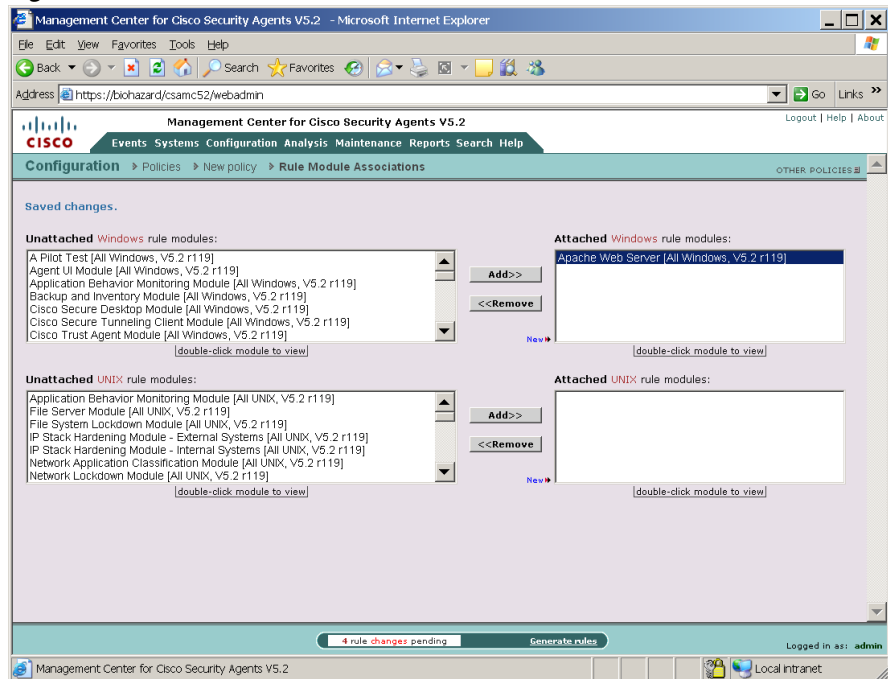
You can attach rule modules of differing architectures to the same policy. This way, you can configure task-specific, self-contained, inclusive policies across all supported architectures for software that is supported on all platforms. For example, Apache is a web server software product that supports Windows, Linux, and Solaris platforms. You can attach three OS specific rule modules for Apache to one policy and only need to maintain that one Apache policy.

---

**Caution**

In order to deploy rule module to hosts, you must remember to attach the policy that the rule module is associated with to a group.

**Figure 4-4 Rule Module Associations**



191500

## Attaching Policies to Groups

When you configure a policy, you are combining configured rule modules under a common name. That policy name is then attached to a group of hosts and it uses the rules that comprise the policy to control the actions that are allowed and denied on those hosts. See [Configuring Rule Modules, page 5-5](#).

CSA MC gives you the option of attaching a policy to a group using the **Modify policy associations** link in the Group configuration page or attaching a group to a policy using the **Modify group associations** link in the Policy list view page.

(You can use the **Modify policy associations** link to attach multiple policies to a group and use the **Modify group association** link to attach one policy to multiple groups.)

To attach a policy or policies to an existing group using the **Modify policy associations** link in the Group configuration page, do the following.

- 
- Step 1** Attach a policy to a particular group by accessing that group's edit view. From **Systems** in the menu bar, click on **Groups** to access the group's list view.
  - Step 2** From the group list view, click the link for the group you want to attach a policy to. This brings you to that group's edit view.
  - Step 3** From the edit view, click the **Modify policy associations** link. This takes you to a page containing swap boxes (see [Figure 4-5](#)). The left box contains the policies not attached to this group. The right box contains policies that are attached to this group.
  - Step 4** To add an existing policy to this group, select the policy in the left box and click the **Add** button. The selected policy moves to the right box and is now attached to the group.

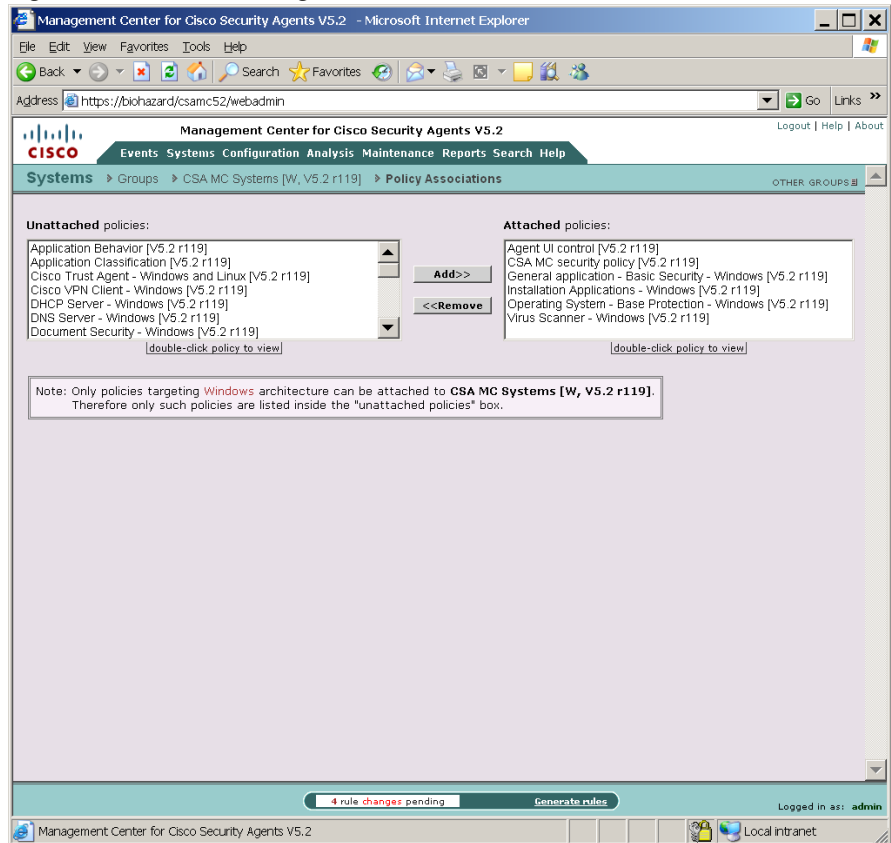
**Note**

---

To remove a policy from a group, select the policy in the right box and click the **Remove** button. It moves back to the left box. (The policy is not deleted from the database, it is just no longer applied to the group.) Although the selected policy is no longer attached to the group, this is not apparent in the GUI until you click the **Generate rules** link in the bottom frame and then the **Generate** button.

---

Figure 4-5 Attaching Policies

**Note**

You can try out policies on host systems by selecting Test Mode for a group or for a particular rule module. Selecting Test Mode and enabling logging on rules attached to "test mode" groups causes the agent to log designated denied events triggered by policies but not take any actions on those events.

*The following chapters provide detailed feature descriptions and configuration information for rule modules and rule types. The rest of this chapter provides a brief discussion on the methodology behind the system, including a high level view of the what the policies strive to accomplish.*

# Overall Policy Methodology

The policies created on the Management Center for Cisco Security Agents should reflect a well-planned, enterprise-wide security policy. If your corporate enterprise already has security guidelines in place, the rules that form your policies should reflect those guidelines.

When you begin to configure policies, there is a common methodology you can use to successfully form the rules that will provide the security and the flexibility you require.

## Analyzing Applications

The rule modules you create as part of your policies are “application-centric.” The application classes, those shipped with CSAMC and the ones you configure yourself, are the key to the rules you build as part of your security policies. Understanding how those applications work is necessary for configuring rules that adequately address the needs of a secure, yet unobtrusive, policy for that application.

There are three specific areas to consider when determining the type of security required by the application in question. There are overall *generic types of protection* that stop malicious code such as System API protection, Buffer overflow protection, and Port scan detection (Network shield rule). There are *application-specific types of protection* you can put in place to allow the application to operate normally while insulating it from any undesired access. Then there are *environment-specific types of protection* that control access to the application in question and its data over various network channels. It is the latter two, application-specific and environment-specific protection requirements, that this section concentrates on.

When analyzing an application for the purpose of writing a policy, consider the following questions.

- What resources does the application own (file, network, and registry resources)?
- Can the application access other resources?
- Can other applications access this application’s resources?

- How is the application administered? (e.g. configuration tools used, accessed locally or remotely)
- Does the application interact with other applications as part of its normal operation?
- Does the application spawn processes and if so, what resources do those processes access?
- What application-based rules vs. environmental rules are necessary?

Determining the answers to these types of questions will help you target the resources you want to control as part of your policy for protecting the application.

For example, asking the questions above when analyzing how a Web server application operates would first lead you to determine which files are installed and used by the Web server application itself. What network resources are accessed and what registry keys are owned by the application? How is the Web server administered? Are html files FTP'ed to the server or is Front Page used locally on the system? These are questions targeted at producing application specific rules for a policy.

You would also note how the Web server is used and who can access it. Is it an intranet or Internet server? Does it act as a standalone server or does it access other resources? If there are forms users fill out on the Web server, does it use a backend SQL Server to store data? If so, which applications must be able to communicate with each other and what services, other than HTTP, are required for this communication? These are questions targeted at producing environment specific rules for a policy.

Ultimately, you want your policy to secure both the application and the environment it operates within.

## Configuring Policies—The Methodology

Once you understand how an application works, you can begin forming a policy to protect it. There are three general areas you want to address for each resource you are protecting. By addressing the security needs of these three areas, you can configure a well-formed policy to protect the resources you are targeting.

When building a policy to protect a designated resource, refer to the following steps to help you address each resource area.

---

**Step 1** Protect the application and its resources (binary files, directories, registry keys, etc.).

You must prevent writing to the application executables themselves. This maintains the integrity of the executable. The only time the executable should change is if you're upgrading the application.

This type of rule would prevent a Trojan from naming itself "Netscape.exe" to disguise itself as the real Netscape executable.

Restrict access to specified data by other applications. For server policies, you'll want to protect information in certain directories on the server in question, allowing restricted access to specific files and blocking all outside access to other files.

In order to correctly formulate this rule, you must examine what other applications (if any) need to access the application data. This type of rule would protect another application from retrieving sensitive data from a server, such as credit card information or a password file.

Restrict access to sensitive application-specific registry keys. You want to allow the specific application to write to its own registry keys, but prevent all other applications from writing to those registry keys.

**Step 2** Restrict the application processes. Understand what resources the application needs and write restrictions to lockdown the application and not compromise the system.

Dictate what the applications in question can and cannot do. Likely, you'll want specific applications to write only to their own file types. To restrict an application, you must look at the files the application needs to read from and write to and then restrict it to only those files. This type of rule would prevent a buffer overrun from compromising a running application, and damaging other components on the system.

When applications are invoked, they often spawn other processes as part of the action they are performing. It may be desirable to place different restrictions on spawned processes. Therefore, when you analyze an application in preparation for writing rules, CSA MC gives you the option of including or excluding child processes created by the original application. You can also restrict the child processes of an application and create a rule to address only those processes.

**Step 3** Provide permissions, as required, to allow the application to function.

For example, if an application requires network connectivity, you should specify what required network services must be enabled. Components that are “network visible” are especially vulnerable to attacks. It is important to control what these network-accessible applications (and their spawned processes) can do.

## General Server Policy

The General Server policy described here uses the applicable steps mentioned previously to secure common server resources. This is a generic server policy that can be applied to any server. Depending on the type of server you’re protecting, you’ll want to apply this General Server policy and then create an additional policy, which more specifically targets the resources you want protected, to augment this general one. Here is an overview of a General Server policy.

**Table 4-1**      **General Server Policy**

<b>Rule Type</b>	<b>Description</b>
File access control	Allow, all applications read system dll’s
Network access control	Deny, lockdown network access client
Network access control	Deny, lockdown network access server
File access control	Deny, protect system executables
Network shield	Detect network port scans, detect and protect against network SYN flood attacks
System API control	Detect and terminate potential application Trojans and viruses.

Note that the rules in these tables are ordered (top to bottom) according to their priority. High priority deny rules take precedence over all others. Allow rules take precedence over deny rules. This General Server policy now locks down the server machine protecting the system directory and protecting network access.

## Sample Web Server Policy

Once you have a general server policy to protect basic server resources, you can write a policy that actually targets the resources used by the particular server application you want to protect. For the purposes of this example, the application is a Web server. The executable is “WEB.EXE.”

This targeted server policy builds on the General-Server policy restrictions, allowing the services required for WEB.EXE to operate securely. Once we explain the components of this policy, we will combine both the General-Server and Sample Web Server policies and implement them together to provide the overall protection the Web server application requires.

**Table 4-2** *Sample Web Server Policy*

<b>Rule Type</b>	<b>Description</b>
File access control	High Priority Deny, protect Web server data
File access control	Allow, let WEB.EXE write to temp files and log files
Network access control	Allow, let WEB.EXE talk to network
File access control	Query user, protect Web server directories from others
Registry access control	Deny, protect sensitive Web server keys
File access control	Deny, prevent WEB.EXE all file write access

Here is how the methodology detailed in the first section of this document was applied to the creation of this policy. The Description, appearing in italics below, given for each rule in the Web server policy table is listed here with the “methodology” step that applies to it.

---

**Step 1** Protect the application executables and data.

*Protect Web server directories from others:* Here we have denied all applications from writing to the directories that contain the Web server application executables. *Protect Web server data:* This rule prevents anyone from writing to html files and defacing web pages. *Protect sensitive Web server keys:* This would protect, for example, keys controlling user authentication settings.

**Step 2** Restrict the application processes.

For a general purpose policy, you want to protect the system from the application in question. Therefore, you can allow the application (ex. WEB.EXE) to read all system files, but restrict writes to system files. (If you are concerned about the application reading certain system files, you can restrict reads to those files specifically, if necessary.)

*Prevent WEB.EXE all file write access:* This rule denies the Web server application access to all files on the system.

*Let WEB.EXE write to temp files and log files:* This rule allows the Web server application to write to temp and log files used by the application.

Note that restricting access to a resource should always be done in the policy that owns that resource.

**Step 3** Provide permissions as required.

*Let the WEB.EXE talk to network:* This allows WEB.EXE to act as a server for the http service.

## Combined General Server and Sample Web Server Policies

To fully protect the Web server, we apply our base General-Server policy and our targeted Sample Web Server policy to the agent running on the Web server system. When applied to the Web server, the combined policies work as displayed in the table below (in order of rule precedence).

**Table 4-3 Combined Policies**

Rule Type	Description
File access control	High Priority Deny, protect Web server data
File access control	Allow, let WEB.EXE write to temp files and log files
File access control	Allow, all applications read system dll's
Network access control	Allow, let WEB.EXE talk to network
File access control	Query user, protect Web server directories from others
Network access control	Deny, lockdown network access client
Network access control	Deny, lockdown network access server
Registry access control	Deny, protect sensitive Web server keys
File access control	Deny, prevent WEB.EXE all file write access
File access control	Deny, protect system executables
Network shield	Detect network port scans, detect and protect against network SYN flood attacks
System API control	Detect and terminate potential application Trojans viruses.

## Reference

“Vulnerable applications” defined in various rules are network-aware applications. These application types are much more vulnerable than others. They are as follows:

- TCP and UDP servers and processes created by them are vulnerable because they are susceptible to buffer overflow attacks.

- Processes that read downloaded content are vulnerable because they may be interpreting and taking action based on downloaded data.
- Remote clients are applications running on another machine and are therefore vulnerable because CSA does not know what these applications are when they attempt to access resources.
- Removable media, in some cases, is categorized as vulnerable. This includes media accessed from CD-ROM, floppy, USB drives, or any other peripheral device.

