



Evaluating the Cisco Security Agent

Overview

This section is meant for users evaluating the Cisco Security Agent 4.5. It is designed to help you understand the product design goals for the Cisco Security Agent 4.5, as well as how to use the Cisco Security Agent to establish a proactive security solution for your mission-critical business applications.

This evaluator's section is recommended for any user who wishes to install the Cisco Security Agent for the purpose of reviewing and evaluating the product. It contains a recommended set of tests that allows evaluators to put the product through its paces, by running a broad range of real-world tests and observing the output. Additional configuration information is available (the CSA MC Install Guide and the CSA MC User Guide) for a more wide scale deployment.

After reading through this appendix, you should be familiar with the capabilities offered by the Cisco Security Agent, and have "hands on" experience installing, configuring, and managing the product.

This section contains the following topics.

- [Evaluation Instructions, page B-2](#)
- [Test 1: Agents are managed from the same console as Firewalls, Network IDS, and VPN devices, page B-2](#)

- [Test 2: Protection is proactive—Stops the unknown attack, page B-15](#)
- [Test 3. The Cisco Security Agent is easy to customize, page B-22](#)
- [Test 4. The Cisco Security Agent runs on UNIX as well as Windows, page B-32](#)
- [Summary, page B-38](#)
- [Frequently Asked Questions, page B-39](#)
- [Java code for backdoor program, page B-43](#)

Evaluation Instructions

This section of the document describes five distinct and concrete steps that you can take to see the power and ease of use that the Cisco Security Agent offers. Each of these steps stands by itself, but build sequentially on one another. Performing each of these steps in sequence will provide a very complete, in-depth analysis and demonstration of the product.

Test 1: Agents are managed from the same console as Firewalls, Network IDS, and VPN devices

The Cisco Security Agent is managed from the Cisco Works platform (specifically, the VPN and Security Management application). This application also lets you manage PIX Firewalls, Cisco VPN concentrators, and Cisco Secure IDS appliances and blades. Centralizing security management functions lets you get more capability from the same set of trained administrators.

Other resources you might want to get:

- *Installing Management Center for Cisco Security Agents*, found on the product CD.
- *Using Management Center for Cisco Security Agents*, found on the product CD.

What you need to do this test:

- A computer to install CiscoWorks Common Services and CSA MC software on. CSA MC is supported on Microsoft Windows 2000 server, with Service Pack 3 installed. Note that if only a few agents will be installed during these tests, it can be installed on Microsoft Windows 2000 Professional.
- One or more computers to install Cisco Security Agent agents on. For this test, it is recommended that Microsoft Windows server or desktop computers are used (either Windows NT 4.0, Windows 2000, or Windows XP). We will test UNIX agents in a later step.
- A computer that you can use to attack the Cisco Security Agent protected systems. You may or may not want to do this, but this document offers some recommendations on tools you can use to attack your computers. Note that some tools run under Windows, while others run under Linux. You may need two computers, or one that will boot either Operating System.
- IP communications between these systems. Communications between agents and servers use secure HTTP (port 443). It is helpful to have access to the Internet to download exploits to exercise the Cisco Security Agent's protection as well.
- An evaluation software license key from Cisco. You will need to get an evaluation key from Cisco to install agents to protect systems.

Things to keep in mind:

- The computer on which you are installing the CSA MC software should be placed in a physically secure, locked down location with restricted access.
- Do not install any software on the CSA MC system that is not required by the product itself.
- You must have administrator privileges on the system on which you are installing the CSA MC to perform the installation.
- The CSA MC system must have a static IP address.

Step 1. Install the Management Center for Cisco Security Agents

Place the CiscoWorks (VPN and Security Management) CD in the CD drive of the computer that will be CSA MC. If you have CD autorun enabled in the computer, the installation start screen will display automatically. You must install both CiscoWorks **Common Services** and **Managing Cisco Security Agents**.

The Managing Cisco Security Agents installation will prompt you for standard information like directories to use to install the software. It will also install and configure the database (Microsoft SQL Server Desktop Engine).

**Tip**

If you already have Microsoft SQL Server 2000 with Service Pack 3 installed on the CSA MC computer, CSA MC will configure and use that, rather than install the desktop engine. While CSA MC will happily use SQL Server 2000, SQL Server version 7 is not supported. The CSA MC installation will abort, and tell you that you need to uninstall SQL 7.

**Caution**

You have to be logged on as administrator to install the agent kits.

**Caution**

If you don't have a key, agents will not register with the Manager. You can request an evaluation key which will ensure that agents work normally.

Step 2. Install Agent Kits on the computers you want to protect

Now you need to log into the Windows desktop or server computer(s) that you want to protect. The easiest way to install agent kits is to use a web browser to directly retrieve the kit from CSA MC. Since CSA MC uses a web server for remote access, and since the agent kits are small, you can easily download and install the appropriate kit on the remote computer.

In your browser, type the URL of the CSA MC computer, e.g. `https://myCSA_MC.example.com/csamc/kits`. If you do not have DNS configured, you will need to type CSA MC's IP address in the browser URL window, e.g. `http://10.0.1.17`. You can use either Internet Explorer or Netscape browsers to do this. Note that this is a secure web page—using an "https" URL.

Figure B-1 shows you the Agent Kits web page. While you could choose to log in to CSA MC via CiscoWorks via the main web page (<http://myCSAMC.example.com>), let's leave that for later. Cisco Security Agent allows you to protect agent computers without even logging in, so let's go straight to installing agent software. Click on the button that says "Agent Kits".

Figure B-1 *Getting Agent Kits Page*



There are three types of agent kits that are automatically generated during the CSA MC installation:

- Servers. The security policy for these kits is optimized for server-class machines, running server-class applications. These kits provide the protection that you would want on most, if not all, of your servers. There is a different agent kit for UNIX and Windows servers—UNIX agent kits are on the top of the screen, and Windows agent kits are on the bottom. You can choose to install the agent kit in IDS Mode (it will alert you but will not block any actions), by clicking on the "IDS Mode Server" kit.

- Desktops. This is the agent kit that is optimized to protect Windows desktop computers running desktop applications. These kits provide the protection that you would want on most, if not all, of your desktops. Note that this agent kit is offered only for Windows computers.
- Cisco Works VMS. This agent kit is optimized to protect the CiscoWorks VPN and Security Management System (VMS), which hosts CSA MC. Cisco strongly recommends that you install this agent kit on every CSA MC.

Right now, you'll want to click on the appropriate agent kit (server or desktop) that you want to install. It's fine to run the install straight from CSA MC, or you can save the agent kit locally and run it from the local hard drive. Note that no interaction is required during the installation—the agent automatically does all local setup, and automatically registers itself with CSA MC.

**Tip**

There's no need to distribute encryption keys to the agents. The agent kit contains everything that the agent needs to securely communicate with CSA MC. Since it uses the SSL encryption capability that your Operating System contains, you probably have strong encryption installed already.

**Tip**

When you're done installing the agent kit, you'll have to reboot. The reason for this is that Cisco Security Agent hooks many different locations in the kernel. You'll only need to do this once. If you like, you can change this so that agent kit installation does not require a reboot.

Congratulations! A Cisco Security Agent is now protecting your computer. Now, let's test it. The best way to test it is to attack it.

Step 3. Attack your system

The proof of any security product is in how well it protects. The Cisco Security Agent is designed to provide unmatched protection "out of the box", without any configuration being required. However, testing this requires that you actually try to attack the computer.

There are many tools available on the Internet that you can find to test your security. [Table B-1](#) lists a small selection of reputable sites that we believe offer high caliber tools for this purpose.

Each of these tools performs a different task, and is used for a different purpose. The following sections show the purpose for each of the tools (i.e. what the tool does when the Cisco Security Agent is not protecting the target), and the expected outcome when the Cisco Security Agent is protecting the target.

Step 3a. Scanning with nmap

nmap is a tool used to identify which devices are present on a network, and what Operating System and services they are running—indeed, the name stands for Network Mapper. nmap works by sending a series of network probes to the target; the fact that the target responds identifies that it is there (and also which ports it is running), and the pattern of error messages returned identifies the OS. nmap is surprisingly accurate in identifying targets. It is frequently used at the initial stage of an attack or investigation, to determine what systems might respond to an attacker's exploits.

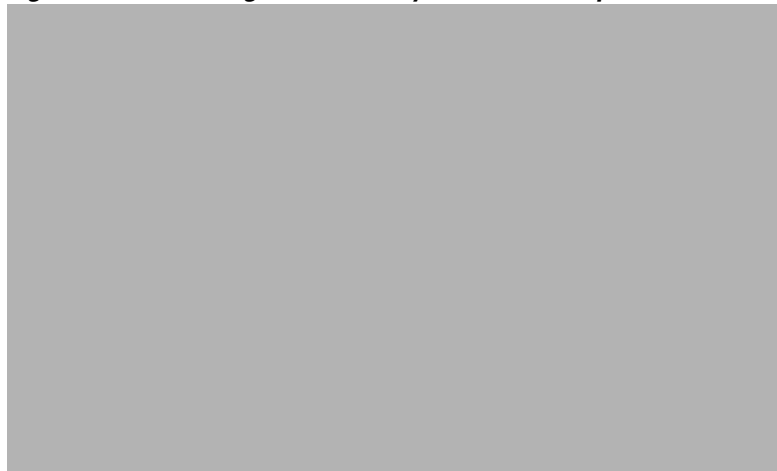
Table B-1 *Tools you can use to test Cisco Security Agent protection*

Tool	Site	Comments
nmap	http://www.insecure.org/	Most sophisticated network mapping and discovery tool. Does an excellent job of identifying the Operating System of the target device. Very commonly encountered.
Nessus	http://www.nessus.org/	A free, Linux-based, Open Source vulnerability scanner. Contains a very large list of current exploits for both UNIX and Windows systems.
Windump	http://windump.polito.it/	High quality, free network packet and password capture tool. Windows version of UNIX tcpdump.
Etherpeek	http://www.wildpackets.com/	Good commercial packet sniffer.
Silentlog	http://packetstorm.decepticons.org/Win/SilentLog.zip	Keystroke logger with source code.

Tool	Site	Comments
Pwdump2	http://razor.bindview.com/tools/files/pwdump2.zip	Allows encrypted password hashes to be dumped, even if the Windows 2000 system is protected (running SYSKEY).
Firehole	http://keir.net/firehole.html	Personal Firewall testing tool that uses DLL Injection.
netcat	http://www.atstake.com/research/tools/	Among other features, can act as a remote login server on any port.
Command Shell	%Systemroot%\system32\cmd.exe (Windows) /bin/sh (UNIX)	Lets you run commands from a command line.

Expected outcome of nmap scans against Cisco Security Agent-protected systems: nmap is unable to identify the target operating system of systems running the default server or default desktop policies. nmap scans will appear to hang while its security tests timeout. nmap scans against systems not protected by Cisco Security Agent will report results very quickly. [Figure B-2](#) shows a screenshot of an nmap scan against a Windows system protected by Cisco Security Agent.

Figure B-2 Scanning a Windows system with nmap



**Caution**

Disabling the Cloak System option (enabled by default in the Default Server and Default Desktop policies) will allow nmap to gather much more information.

Step 3b. Scanning with Nessus

Nessus is an Open Source vulnerability scanner that runs on a Linux computer. It makes network connections to remote systems and runs many hundreds of security tests against them. Nessus is one of the few scanners that relies extensively on exploits to determine vulnerability—it typically does not rely on any "banner" information passed back from the remote service (e.g. "Sendmail 8.8.2 ready"), but instead actually tries to break into the service.

Nessus is well regarded in the security industry, and is frequently updated with new tests. It is frequently used to perform network security audits.

Expected outcome of Nessus scans against Cisco Security Agent-protected systems: Nessus will not detect any vulnerabilities in systems running the default server or default desktop policies. Nessus status screens will be blank, and its reports will show no vulnerability information. Nessus scans against systems not protected by Cisco Security Agent will typically report large numbers of "holes", "warnings", or "notes".

**Caution**

Disabling the Cloak System option (enabled by default in the Default Server and Default Desktop policies) will allow Nessus to gather much more information.

Step 3c. Capturing packets with Windump or dsniff

Packet capture programs are sometimes referred to as packet "sniffers", after the popular Sniffer™ product sold by Network Associates. These utilities are used for a wide range of uses, from legitimate network troubleshooting to the much more shady password theft.

Windump is a Windows version of the popular tcpdump packet capture and analysis utility. Since it uses the tcpdump data format, add-on utilities that process tcpdump logs will also process Windump logs. This makes Windump a useful and popular packet capture utility. Dsniff is another popular packet capture program.

Expected outcome of running Windump on Cisco Security Agent-protected systems: Cisco Security Agent detects applications that perform unusual interactions with the NIC (more specifically, with the NDIS interface in Windows or with streams in UNIX). These events will be intercepted either at boot time (Windows) or in real-time (UNIX).

**Tip**

Since Windump interfaces with the NIC at boot time, you'll have to reboot the computer after installing Windump if you want to test Cisco Security Agent's capabilities.

Step 3d. Capturing keystrokes with Silentlog

Silentlog is a "keystroke logger" program that silently captures all keyboard input and logs them to a file. Attackers often install keystroke loggers to capture passwords entered by users. Many Trojan horse programs include keystroke logging as a feature.

Expected outcome of running SilentLog on Cisco Security Agent-protected systems: Cisco Security Agent will generate a message saying that an attempt is being made to capture all keystrokes. The user will be able to select "Yes" (allow this action to take place), "No" (block this action but let the program continue to run), or "Terminate" (terminate the application that caused this event). By default, Cisco Security Agent will terminate the application trying to capture keystrokes if no selection is made within 5 minutes. [Figure B-3](#) shows this message box.

Figure B-3 User queried about keyboard sniffing



**Tip**

You can install SilentLog anywhere, on any drive, under any name. You could, for example, install it under the name IEXPLORE.EXE (the name used by the Internet Explorer web browser). What is important to the Cisco Security Agent is not the name, but the action that the application takes.

**Tip**

Some programs trap keyboard input as part of their normal operations. The AOL Instant Messenger and Yahoo Messenger instant messaging clients are two examples of this.

Step 3e. Hijacking applications via overwriting memory or via DLL Injection

A popular attack program that tries to steal passwords from the Windows registry is PWDUMP2. PWDUMP2 tries to overwrite a table used by the Local Security Authority subsystem (lsass.exe) to grant itself privilege to access the passwords. These hashes are then analyzed by password cracking routines like Crack, l0phtcrack, or John the Ripper.

Another attack method involves tricking another application into executing your code. Common software routines are stored in collections called libraries. Microsoft Windows provides a built-in ability to load these libraries as required—this capability is called *Dynamically Linked Libraries*, and is abbreviated DLL. One form of attack is to insert (or "inject") a new and malicious DLL into a running application. This attack is called "DLL Injection."

Another application that uses DLL injection is FIREHOLE, which is used to test whether personal firewall applications can "leak"—whether applications can make unauthorized outgoing connections.

Expected outcome of running PWDump2 or Firehole on Cisco Security Agent-protected systems: The Cisco Security Agent detects applications that try to overwrite a different application's memory, or that inject code into other running applications. As with our keyboard sniffing example, the user will see a pop-up box that identifies the application and asks whether the user wants to let this activity occur, block the application, or terminate the offending application. [Figure B-4](#) shows this dialog box. Note that while [Figure B-4](#) shows the result of running PWDump2, you would see a similar message if you used Firehole.

**Tip**

There are very few legitimate programs that use this technique. If you see activity like this—especially from downloaded programs—you should be extremely suspicious.

Figure B-4 User queried about DLL injection activity



Step 3f. Replacing critical portions of the Operating System (rootkit attacks)

One of the classic attacks against UNIX systems (dating from the 1980s or before) was to replace critical Operating System (OS) binaries. For example, the routine used by most UNIX systems to authenticate users during login is the program `/bin/login`. Replacing this program with a different one that stores the passwords in a secret file is the classic subversion attack against UNIX.

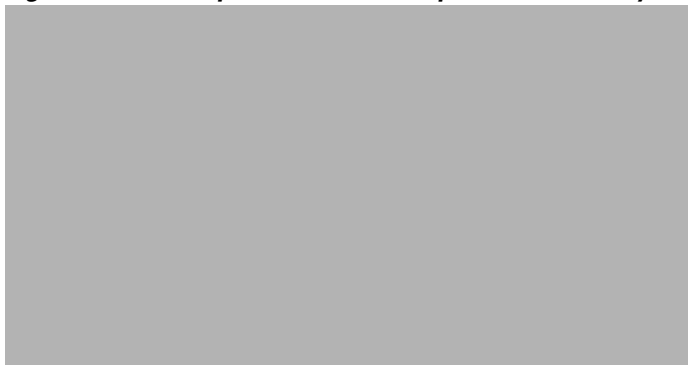
Windows stores many of these routines in the `SYSTEMROOT`, which is typically found in the directory `C:\WINNT\SYSTEM32`. Not only are there executable routines here, but DLL libraries as well.

Many malicious programs attempt to modify or replace these programs. For our example, you can use a command shell (MS DOS Prompt) to manually replace or copy files. You can run a command shell from the start menu by selecting "Run" and typing "cmd.exe" in the dialog box. In the command shell, type "CD %SYSTEMROOT%" and then "cd system32" to change to your OS directory.

All executables and libraries in this directory are protected. For testing purposes, we will use a relatively unimportant one, XCOPY. Rather than deleting it, we will simply copy it to another filename. At the command prompt, type "copy xcopy.exe xcopy1.exe". When you see the query message, choose "No".

Expected outcome of OS replacement attacks against Cisco Security Agent-protected systems: You will see a query popup asking whether this activity is desired. As with other queries of this type, you can allow, deny, or terminate the activity. [Figure B-5](#) shows this query.

Figure B-5 User queried about OS replacement activity



Caution

In this example, we tried a trivial modification of the OS. You can try to actually delete critical files like SYSTEM.EXE or LSASS.EXE, and the Cisco Security Agent will protect you. However, you should make sure that Cisco Security Agent is in protection mode (in other words, you are not in "IDS Mode"—sometimes called "Test Mode"), and that you are running the default protection policy (either Default Servers or Default Desktops). If you have disabled the Cisco Security Agent, deleting or replacing these files will damage the Operating System.



Tip

There are times when users will want to add executables or DLLs to the system32 directory—for example, when installing device drivers. Query messages will allow users to install drivers, but prevent silent malicious installation of attack programs. That is why the query box asks if the user is installing software. Query

messages are configurable by the security administrator—if you don't want to give your users the ability to make security decisions, these rules can be configured to silently block the activity.

Step 3g. Unknown servers listening on high-number ports (backdoors)

A popular way to be able to connect to other systems is via an application listening (acting as a network server) on hard-to-find high number ports. If the attacker knows that he can connect to port 53,962, it is unlikely that most security defenses will be watching for this type of attack. Backdoor programs like this are included in most Trojan Horse applications.

netcat is a utility that can be used for many purposes, such as connecting to applications across the network and sending arbitrary data streams to it. It also provides the ability to listen (act as a server) on any port that you like (we've chosen port 23000 in the example below).

```
nc -l -p 23000 -t -e cmd.exe
```

Attackers would start a netcat listener on a high level port, and then use netcat (or Telnet) on a remote system to connect to the listener:

```
telnet target.ip.address 23000
```

Expected outcome of unknown servers running on Cisco Security Agent-protected systems: The remote login session will be unable to connect to the netcat listener. The attacker may see a "Connection refused" message.



Caution

Connecting to yourself—for example, "telnet localhost 23000"—will not trigger Cisco Security Agent. As it turns out, many applications use this internal localhost address to communicate. Cisco Security Agent does not block traffic that is both generated and received locally. You have to try to break into the netcat backdoor remotely.

Key Point: Notice that until you actually run some attacks, the Cisco Security Agent is very quiet. This is by intention: if your security system requires a lot of attention when you are not being attacked, all it is doing is making more work for you. The Cisco Security Agent is designed not only to stop attacks, but also to provide a very low number of alerts for you to manage.

Test 1 Summary

We saw several key points in this test:

1. Agents are managed from the same console as Firewalls, Network IDS, and VPN devices. The same Management platform can consolidate management of several security products.
2. The Cisco Security Agent protects against a large number of attacks with the default policies. There is no need to customize security policies to get protection.
3. The Cisco Security Agent does not generate a large number of alerts. Unless you attack it, it will not give you many alerts that you have to manage.

Test 2: Protection is proactive—Stops the unknown attack

The Cisco Security Agent differs substantially from traditional Host-based security products, in that it contains no signatures. Rather, it focuses on behavior, and blocks malicious or undesired behavior. By focusing on behavior, new and previously unknown attacks can be detected and blocked.

No new tests will be performed in this section—instead, we will analyze what happened in the tests we just ran, and determine that Cisco Security Agent protection will proactively stop undesired behavior.

What you need to do this test:

- The CSA MC used in the previous test. The Cisco Security Agent agent(s) reported all malicious activity to CSA MC. We will analyze these alerts in this test.
- A computer to connect to CSA MC. The CSA MC user interface is provided in a Web Browser. You need a computer with a browser to access the CSA MC's user interface. Both Internet Explorer (5.x or higher) and Netscape (6.2) are supported. Note that you can use a browser on CSA MC itself to connect to its user interface.

Step 1. Connect to CSA MC's URL using the web browser

Using your browser, connect to CSA MC. If you are using a browser on the CSA MC computer, you can use the URL `http://localhost`. If you are using a browser on another computer, you will need to use a URL with CSA MC's domain name or IP address.

CSA MC runs under the Cisco Works management framework. This allows you to manage all of your Cisco security products from a single platform, using the same access method. For example, you can configure and monitor both your Cisco Security Agent host-based protection as well as your Cisco Secure IDS network-based protection from the same Cisco Works server.

Click the "Login" button, and type the user name and password that you defined when you installed CSA MC. Note that your browser may notify you that it does not recognize CSA MC's web server certificate. This is normal, since CSA MC creates a unique certificate when it is installed. You can import the cert into your browser, but for now, it is fine to ignore this.

After selecting "VPN/Security Management Solution," "Management Center," and "Security Agent" in CiscoWorks, you will see the CSA MC Status Summary screen shown in [Figure B-6](#).

Figure B-6 CSA MC Status Summary Screen

Step 2. Open the Event Log

The color coded Bar Graph contains a quick summary of the number and severity of the alerts that CSA MC has received. As we can see, in this example, there are several high severity events. Clicking on the red portion of the bar takes us directly to the Event Log screen that provides a detailed breakdown of the events for each Severity Level, as shown in [Figure B-7](#).

The Event Log shows all of the alerts that were received by CSA MC. Some of these may be pop-up query messages that the user saw; others may relate to events that end users never saw. Note that if the alert is from a user query message, the alert will specify which action the user took (Allow, Deny, or Terminate).

Figure B-7 The Event Log Screen

Look at the alert generated when we ran PWDUMP2. You'll see an alert that says something like the following:

The user was queried when the program 'C:\Documents and Settings doty\Data\Products\PWDump2\pwdump2\pwdump2.exe' (as user TEDOTY-W2Kedoty) tried to modify the memory in program 'C:\WINNT\system32\lsass.exe'. This is normally only done by debuggers. The user chose 'Terminate'.

[\[Details\]](#) [\[Rule 1579\]](#) [\[Wizard\]](#)

Key Point: Notice that the activity was denied. The rule did not alert us about an attack that possibly was under way and that we might want to look into. Rather, the attack was intercepted and terminated. The alert is interesting for the administrator only in a historical sense—there is no need to have a room full of security operators watching consoles, looking for things to turn red. Even if the agent cannot communicate with the console, the system will be protected with the last policy that the agent received.

**Tip**

There are not 1579 rules that ship with the Cisco Security Agent product. Rules are renumbered for each product release to ensure that rules are always unique (in other words, to ensure that you are always running the latest rule). Typically, the default policies for servers and desktops contain 30 to 40 rules.

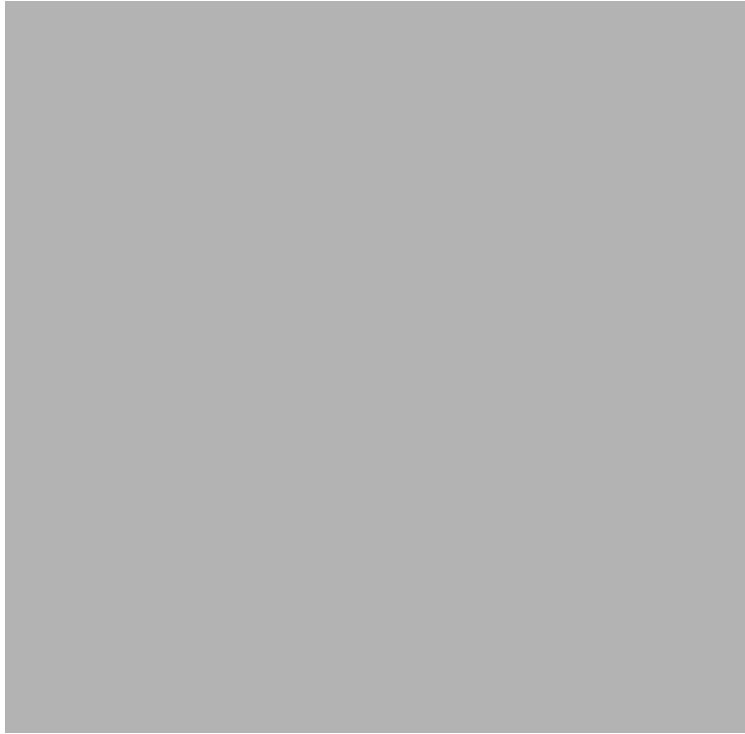
Step 3. Click through to the rule that caused this alert

Now let's look at why we couldn't copy the XCOPY.EXE program. You'll see an alert that says something like the following:

The process 'C:\WINNT\System32\cmd.exe' (as user CISCO-MAdoty) tried to open/create the file 'C:\WINNT\system32\xcopy1.exe' and the user was queried. The user responded by choosing 'No'.

[\[Details\]](#) [\[Rule 1613\]](#) [\[Wizard\]](#)

Let's take a look at rule 1613, which controls this activity. Click on the link to the rule, which will display the actual rule (shown in [Figure B-8](#)).

Figure B-8 The Rule Screen

Note that we can tell several things from the rule screen:

- The user will be queried as to whether the activity is allowed or not.
- If the user doesn't make a selection, the default action will be that the activity is blocked.
- The rule will trigger when any application trying to write system executables, system libraries, or drivers.

**Tip**

Note that we have a couple pre-defined variables here: System Executables and System libs and drivers. Cisco has predefined these to relate to the specific files of these types for the various Operating Systems that the Cisco Security Agent supports. These variables can be examined and modified if desired, but there is typically no need to do so.

Key Point: What is important to note here is that most of the rules that we've caused to trigger are very generic in nature. Rather than a signature to look for a particular executable file, the rules look for types of activity that are not desired. What is important is not that cmd.exe (or a particular version of cmd.exe) that tried to modify a system executable; as you can see from the rule, any application trying to overwrite the Operating System will trigger exactly the same reaction. The application could be a web browser that is executing malicious mobile code, or an email client program that executes a malicious attachment, or a web server attacked by a remote exploit. Since all applications are effected by this rule, the system is protected against all manner of attacks against the Operating System binaries.

**Tip**

One result of this is that there is typically little or no need to have a specific security policy for an individual application. The default policies for desktops and servers provide broadly-based security coverage that applies to all applications that run on the system.

This ability to block attacks based on behavior, rather than signatures, is the reason that we say that Cisco Security Agent is Proactive.

Test 2 Summary

We saw several key points in this test:

- The Cisco Security Agent stops attacks. All Cisco Security Agent alerts are of the form "I saw this activity which violates your policy, so I stopped it". By the time the security administrator sees the alert on CSA MC, the activity has already been stopped. The Cisco Security Agent does not provide Intrusion Detection, it provides Intrusion Prevention.
- The Cisco Security Agent protects against attacks that it has never seen before, because the protective rules target malicious activity, not particular offending programs or network traffic. Any application that behaves in a way proscribed by the rules will be detected, and the activity stopped. Even applications that have never been seen before—or perhaps not even created when the rule was defined—will be controlled.

Test 3. The Cisco Security Agent is easy to customize

Now that we've seen the "out of the box" proactive protection that the Cisco Security Agent can provide, let's see what is required to tune the default server and desktop policies to fit your local security environment. Since the rules are easily configured via a "point and click" interface, this will be relatively straightforward.



Tip

You will typically only do this to adapt the default policies to your environment. While every environment is a little different, there should be no reason to change many rules to get an exact fit. There is also little need for most people to create custom policies, or policies for most applications—the broad protection provided by the default policies will protect all applications running on a system.

What you need to do this test:

- The CSA MC used in the previous test. We will modify some of the rules that caused alerts in the first test.
- A computer to connect to CSA MC. Remember, you can use a browser on CSA MC itself to connect to its user interface.
- A computer running the Cisco Security Agent to test the customized policy. We will repeat Step 3f from Test 1. It's best to use the same computer that you used in Test 1, but any computer running the same policy will work, too.

Step 1. Connect to the CSA MC's URL using the web browser, and go to the Event Log screen

Using your browser, connect to CSA MC. Log in with the username and password you entered when installing the CiscoWorks software, and open up CSA MC from the "VPN/Security Management Solution".

If you didn't start there on login, select "Event Log" from the "Monitor" menu. You should see a screen that looks like the one we saw in [Figure B-7](#). You should see an alert from the time you tried to copy XCOPY.EXE to a different name in the system32 directory.

In the last test, we just looked at the rule that stopped this. Now let's modify the rule.

Step 2. Jump to the rule

This step is just like step 3 in the previous test. To view the rule, click on the link to the rule in the alert. We saw this rule earlier in [Figure B-8](#).

Let's assume that our environment is a little different. We do not want users to be able to install drivers or update the Operating System—perhaps our IT organization takes care of all driver and hotfix installation. Because the default policy allows the user to decide whether to do this (via the query pop-up messages), the default doesn't quite fit the way we work.

Let's change the rule to prohibit the action, rather than querying the user. Click the pull-down menu for the rule action and select "Deny", as shown in [Figure B-9](#). We need to save the rule by clicking "Save" on the lower left hand side of the window. Congratulations—you've adapted a rule to your environment.

Figure B-9 Customizing a Rule



Tip

Notice that we could have also chosen "High Priority Deny", which would have accomplished the same thing.

Step 3. Deploy the rule

While we've successfully changed the rule, we need to get it distributed to the agents. Updating large numbers of agents can be a daunting challenge with many security products. Fortunately, the Cisco Security Agent makes this easy.

First, we need to regenerate the rule. This causes CSA MC to build a policy update package for every agent that uses the particular rule. Click "Generate", which you find in the middle of the bottom of the current screen. What you'll see is a screen that describes the change you are about to make. This is to give you a chance to change your mind before deploying changes to your running policies, and to verify that all the changes are what you want (you could have several rule changes generated all at the same time). Let's assume that we do in fact want to make this change. Go ahead and click "Generate" again.

You will see a few messages about what the CSA MC does when it generates the rule. There really isn't anything you need to do here, and this usually only takes a few seconds.

That's it! CSA MC has built updates for all agents that need them. As soon as the agent polls in, it will automatically get the modified policy and start enforcing it. Your agent will poll in after the default interval of 10 minutes. You can hurry this along by clicking on the red flag icon in the task bar of the system that is running the agent.



This opens up the local Cisco Security Agent GUI. Selecting the "Advanced" tab shows you the window in [Figure B-10](#).

Figure B-10 Using the Cisco Security Agent local UI to poll



Clicking "Poll" causes the agent to poll immediately. Now you should have the updated policy running on your agent.

Key Point: Updating large numbers of agents has traditionally been the hardest part of managing security. While security policies tend to change slowly, each change typically required a considerable amount of "Leg Work" to deploy. The Cisco Security Agent is designed to completely eliminate policy update work—once rules are generated, the policy deploys itself.

Step 4. Repeat the attack from Test 1, Step 3f

This step was where we tried to copy an executable into the system32 directory. That time, we saw a pop-up message asking whether this was OK. This time, all we get is an "Access is denied" message in our command shell. CSA MC has a new alert, but there was no end-user query.



Tip

You will notice that the task bar flag icon on the system where the attack took place is now waving. This is to attract the attention of the end user, who can view the local alerts (but not modify the security policy). If you want, you can configure a group of agents so that the flag icon is hidden from the end user, and no pop-up query messages are generated. Since some organizations prefer not to make security products visible to users, this is easily configured.

Step 5. The Tuning Wizard helps automate policy tuning, and the optional "Profiler" product investigates alerts

CSA MC contains a tuning wizard that automates the process of adapting Cisco Security Agent policies to local environments. The optional Cisco Security Agent Profiler product also will investigate application behavior to investigate alerts that you do not understand.



Tip

Profiler will also create entirely new policies that control applications, based on how the application behaves in your environment. You can use this not only to investigate unknown activity that you see, but to create custom security policies for high-value applications.

**Caution**

Profiler is a separately license component of CSA MC. You don't have to do anything to get the Profiler installed—it is automatically installed with CSA MC. You do have to install a license key to enable Profiler functionality.

You will find that many of the alerts in the Event Log have a "click-through" option: "[Wizard]". The wizard helps you customize policies to allow events that are normal or expected in your environment. It walks you through the process of deciding if this is normal, whether it is normal for one system or more systems, and how you would like the policy to be updated. It analyzes your current policies and makes recommendations as to the most sensible manner of making the update. At the end of the process, it makes the appropriate changes for you. [Figure B-11](#) shows an alert with a link to the wizard.

Figure B-11 An alert with a link to the Profiler policy wizard



Let's take a closer look at this alert. It seems that a VPN client wants to accept incoming network connections from the VPN concentrator. This is blocked by the Default Desktop policy, which does not allow desktop applications to act as network servers. Users haven't complained that the application isn't working correctly, but it is hard to know precisely what the impact to the application might be. Let's tune the policy to allow this behavior.

Figure B-12 The Tuning Wizard

Clicking the **Wizard** link brings up the first screen of the wizard, shown in [Figure B-12](#). The wizard first provides the administrator with a decision point, since there are three mutually exclusive ways to tune your security policy to handle this type of event:

1. If this application is known to be safe in this environment—and a VPN client communicating with a VPN concentrator is almost certainly safe—you want to update your policy to let this activity occur in the future. If you select this option, the wizard guides you through the process of modifying the policy to allow this activity. By allowing behavior that you know to be safe to occur, you will eliminate future occurrences of this event.
2. If the application is trying to do something that you don't want to occur, but that you don't want to be notified about, you want to update your policy to disable logging/alerting for this event. For example, if a (possibly mis-configured) print server keeps trying to connect to one of your other servers, an agent may block and alert this activity. Assuming that this is activity you want blocked, you will find your event log filling up with a large number of these alerts, as the mis-configured server repeatedly tries to connect. Having the rule keep blocking the activity, but disabling logging, may be appropriate for situations like this.
3. If you don't know whether the action is safe, the wizard can set up a Profiler analysis job to observe the application and report on what it does. This will give you enough information to decide whether to allow the activity or to keep blocking it.

After taking a closer look at the VPN application, let's assume that this activity is low risk. We can also anticipate that this is something that is likely to occur frequently, and that not blocking this activity may result in more efficient operation. If there is a vulnerability in the VPN application—for example, perhaps a buffer overflow—the Default Desktop policy will still offer protection from attack. Therefore, let's use the wizard to change the policy. We will make sure that we've selected the radio button labeled "create an exception rule". Clicking the **Next** button brings up the next screen of the Wizard, shown in [Figure B-13](#).

Figure B-13 *The Policy Tuning Wizard*



This screen shows the action that was blocked, and gives a description of the rule that blocked it. If you want, you can click through to the rule itself like we did in Test 2, but for now, let's just click **Next** to go to the next screen, shown in [Figure B-14](#).

Figure B-14 Deciding how to update the policy

The wizard now shows your options on how to update the policy. One way is to actually change the policy module—the building block used by one or more policies—that contains this rule (in this case, the "Inbound Port Blocking Module"). Notice, however, that in this case the wizard recommends creating an "Exception" policy that will be included in the groups of systems that are affected. The group affected here (in other words, the group that contains the agent that generated the alert) is the Default Desktops group.

**Tip**

This is actually the best way to tune most rules. There are a couple of reasons why. First, by collecting these "exceptions" in separate policy modules, you can easily see how you've customized your policies. If you changed a number of existing rules in a number of policies, it would be difficult or impossible to reconstruct later. Second, when you upgrade to a new release, the default policies might change, but your exception policy modules will be preserved untouched. This means that tuning your policies by using this exception policy module approach will be easier for you to understand and maintain in the future.

For now, let's take this approach. Click **Next** to continue.

The wizard now asks whether this exception applies only to the application in question—the application that caused the rule to block the action—or whether it applies to other types of applications. Note that Cisco Security Agent has a large number of applications that it knows about (applications that use the network, etc), and these are displayed in the wizard. However, the wizard recommends that the exception be applied more strictly—only to the Blackberry application itself. Click **Next**.

The wizard will now ask you which applications this exception rule will be applied to, as shown in [Figure B-15](#). While you can choose to have a rule apply to one of the many built-in application sets (actually referred to as "classes"), or to all applications, the wizard will suggest that you make exceptions specific to the particular application causing the alert. This is the best approach for our VPN client, so we will let the wizard create a new application class for our VPN client. We're almost done, so click **Next** to go to the last step.

Figure B-15 *Selecting the appropriate applications*



The wizard now shows a summary ([Figure B-16](#)) of what you've decided to do. If you click **Finish**, the wizard builds or updates the appropriate policies. You'll have to click the **Generate Rules** button on CSA MC to update the live policies, and agents will get the updated policies the next time they poll in. If you click **Cancel** in the wizard, it discards your changes before committing them.

Figure B-16 Summary of policy changes



Tip

Profiler's analysis capability is very useful for forensics investigation. If your Network IDS reports traffic to a particular high-numbered port on a host, you can use Profiler to analyze what application is listening on that port and what it does. Profiler collects and summarizes all resource access requests that the application makes. It can even build a policy to "sandbox" the application, if you like. This ability to closely investigate mysterious applications can greatly speed up the security administrator's ability to identify undesired activity.

Key Point: While tuning policies to your environment is not a complex task, automation allows lower-level administrators to be effective in updating policies. Because the wizard guides the process, policies are updated sensibly. Using the wizard ensures that policies remain well-structured and maintainable over time.

Test #3 Summary

We saw several key points in this test:

1. The Cisco Security Agent is simple to customize. The CSA MC UI provides hotlinks to the pertinent rules, making it easy to identify which rules need modification to fit your local policy.
2. CSA MC provides a wizard to make customization even easier. The wizard helps automate more complex customization tasks, so that even junior level security administrators can quickly and sensibly adapt the Cisco Security Agent policies to their environment.

Test 4. The Cisco Security Agent runs on UNIX as well as Windows

So far, we've been doing all of our testing on Windows systems. Now let's see how the Cisco Security Agent performs in a heterogeneous Windows and UNIX environment.

What you need to do this test:

- The CSA MC used in the previous tests. We will do some testing on a Solaris system.
- A computer running Solaris. Cisco Security Agent supports Solaris 8 systems.
- A computer to connect to CSA MC. Remember, you can use a browser on CSA MC itself to connect to its user interface. You can also use a Netscape 6.2 browser on the Solaris computer to connect to CSA MC.

Step 1. Install Agent Kits on UNIX computer

You'll need to log into the Solaris computer(s) that you want to protect. The easiest way to install agent kits is to use a Netscape web browser to directly retrieve the kit from CSA MC, just like we did on Windows computers in Test 1.

**Tip**

There is a GNU utility called `wget` that will download an agent kit from CSA MC, when run from a command line. Systems that do not have web browsers, or where installation is desired to be scripted will find this utility handy. You can find a compiled `wget` Solaris 8 binary at <http://www.sunfreeware.com/sol8right.html>.

**Tip**

You can also save the Solaris agent kit on a server, and use FTP to distribute it to your target computer.

**Caution**

Unlike Windows systems, The Cisco Security Agent does not offer a "Default Desktops" agent kit for Solaris. It is assumed that all Solaris systems are servers, so only a "Default Servers" agent kit is provided by default. Note that the "IDS Mode Server" agent kit is also available for Solaris systems. As with the equivalent kit for Windows servers, this will alert, but not block activity.

In your browser, type the URL of the CSA MC computer, just like you did in Test 1. The Solaris agent kit is not an InstallShield executable—rather, it is a Solaris package. You install the agent kit just like you install any other Solaris package.

**Caution**

Just like you need to be logged in with administrator privilege to install the agent kit on Windows computers, you have to be logged in as root to install the agent kit on Solaris computers.

We'll assume that you've used a Netscape browser to download the agent kit to your Solaris system. Once it's downloaded, you need to extract the files from the tar archive:

```
# tar xf CSA-Server_V4.0.0.76-setup.tar
```

The installation will be familiar to any Solaris system administrator: you use `pkgadd` to install the agent:

```
# pkgadd -a csa/reloc/cfg/admin -d .
```

**Caution**

While pkgadd installs and configures the agent, you will need to either reboot or kill and restart running processes for the agent to begin protecting the system. The reason is that while the agent doesn't replace any portions of the Operating System, it intercepts calls to the OS in a number of places, but this interception is effected when the process starts running.

Step 2. Attack your system

Just like we did in Test 1, we need to attack the Solaris system to see how the Cisco Security Agent provides protection. You can use many of the tools discussed earlier.

Step 2a. Scanning with nmap or Nessus

Just as port scans are used against Windows targets, they are used against UNIX targets.

Expected outcome of Nessus or nmap scans against Cisco Security Agent-protected systems: Because the Solaris Operating System works very differently from Windows, the results of scanning your Solaris system are different than when you scanned your Windows system. Because the Solaris inetd process services all incoming connections (before handing them off to the appropriate service), nmap and Nessus will report that the Solaris system has a number of open ports. This will be true even if Cisco Security Agent is blocking incoming connections from the scanning host. While it may seem surprising that blocked ports are reported as open, this is an artifact of the way UNIX handles network connections—in fact, this is exactly what is seen when using other security tools like TCP Wrappers or xinetd.

**Tip**

The Cisco Security Agent will report all port scans to CSA MC. CSA MC will also correlate these reported portscan events with similar events received from other agents to detect distributed port scans (port scans where many systems are scanned at the same time).

**Tip**

Unlike TCP Wrappers, Cisco Security Agent provides centralized control of which systems will be blocked. If you like, you can even define "Hosts.Allow" and "Hosts.Deny" variable names to hold allowed or blocked systems, and enforce this globally. Unlike TCP Wrappers, you can also centrally control which applications are allowed to accept connections. For example, you could globally prohibit the use of Telnet to prevent the sending of unprotected passwords over the network.

**Tip**

If you use the Solaris system as the scanning computer, the Cisco Security Agent prevents the execution of both nmap and Nessus. Since both nmap and Nessus send many non-standard packets, they need to communicate directly with the network interface. The Cisco Security Agent prohibits this, so these applications will not run on a system protected by the Cisco Security Agent. If you try, you will get a message similar to the following generated by nmap (note how nmap is confused, and thinks that you are scanning localhost):

```
Starting nmap V. 2.54BETA28 ( www.insecure.org/nmap/ )
pcap_open_live: /dev/eri: Permission denied
There are several possible reasons for this, depending on your
operating system:
LINUX: If you are getting Socket type not supported, try modprobe
af_packet or recompile your kernel with SOCK_PACKET enabled.
*BSD: If you are getting device not configured, you need to
recompile your kernel with Berkeley Packet Filter support. If
you are getting No such file or directory, try creating the
device (eg cd /dev; MAKEDEV <device>; or use mknod).
SOLARIS: If you are trying to scan localhost and getting
'/dev/lo0: No such file or directory', complain to Sun. I don't
think Solaris can support advanced localhost scans. You can
probably use "-P0 -sT localhost" though.
```

Step 2b. Capturing packets with snoop

One of the first things that an attacker tries once he gains access to a UNIX system is to start a packet sniffer to capture passwords that are sent across a network. While there are many sniffer programs available, they all cause the NIC to change

into "promiscuous" mode (i.e. receive and decode all packets seen on the network, not just those addressed to the system itself). For the purpose of this test, we've used snoop, a packet sniffer which comes bundled with Solaris.

Figure B-17 Capturing packets with a packet sniffer



Expected outcome of packet sniffing against Cisco Security Agent-protected systems: The Cisco Security Agent Default Server policy prevents the NIC from changing into promiscuous mode. Different programs will handle the "access denied" OS signal differently, but [Figure B-17](#) shows the result when snoop is run. Note that the user trying to run snoop is root. Root normally has unrestricted access to all system resources; this example shows that even the root account can be prevented from performing dangerous tasks.

Step 2c. Installing a backdoor on a high-numbered port

One of the most popular utilities for running backdoor services is netcat. As we saw earlier, you can use netcat to run a backdoor Telnet-like service on any port you like with the following command (this example causes netcat to listen on port 23000):

```
nc -l -p 23000 -t -e /bin/sh
```

Unfortunately for our testing purposes (but good for security), we've seen that netcat won't even run on a Solaris system protected by the Cisco Security Agent, because it tries to hook directly into the NIC streams interface. Therefore, we have to use a different program to test the Cisco Security Agent. We chose to write a small Java program to do this, because Java is widely available (especially on Solaris systems). Also, many people know how to write small Java programs, so the number of potential attackers who can do this is large.

Expected outcome of running backdoor programs against Cisco Security Agent-protected systems: While the program may start up and believe that it is listening for incoming connections, (Figure B-18 shows us running a backdoor on port 21337), the Cisco Security Agent Default Server policy will prevent the remote client from connecting to the backdoor program.

Figure B-18 *Running a backdoor program*



Tip

The Java source code for this backdoor is listed at the end of this appendix.

Step 2d. Overwriting the Operating System (rootkit attacks)

As we said earlier, replacing critical Operating System (OS) binaries is one of the classic computer attacks. Any command shell (/bin/sh, /bin/csh) will let you try to replace critical system binaries.

Expected outcome of running backdoor programs against Cisco Security Agent-protected systems: As shown in Figure B-19, we attempted to overwrite the command shell program /bin/sh with one of our own, /tmp/evilsh. As you can see from the example, the Cisco Security Agent Default Server policy prevented this even though we were logged in as root.

Figure B-19 *Trying to Trojan the Operating System binaries*



Key Point: The Cisco Security Agent offers the ability to secure UNIX and Windows systems, whether they are desktops (Windows) or servers (Windows or UNIX). This simplifies training, reduces the number of console systems that are required to manage security, and eliminates the need for data consolidation between products.

Test 4 Summary

We saw several key points in this test:

1. The Cisco Security Agent protects UNIX systems. Just like we did with Windows, we deployed agents that protected a Solaris system.
2. The Cisco Security Agent manages UNIX and Windows agents from a single console. CSA MC lets you control both Windows and UNIX systems from a single management point.

Summary

Congratulations! There's a lot to see in any new product, and we've seen a lot during our evaluation of the Cisco Security Agent. While any introduction can only cover the highlights, we've seen several key points:

1. Agents are managed from the same console as Firewalls, Network IDS, and VPN devices. The same management platform can consolidate management of several security products.
2. The Cisco Security Agent protects against a large number of attacks with the default policies. There is no need to customize security policies to get protection, or even to log into CSA MC.
3. The Cisco Security Agent does not generate a large number of alerts. Unless you attack it, it will not give you many alerts that you have to manage.
4. The Cisco Security Agent stops attacks. All Cisco Security Agent alerts are of the form "I saw this activity which violates the policy you specified, so I stopped it". By the time the security administrator sees the alert on CSA MC, the activity has already been stopped. The Cisco Security Agent does not provide Intrusion Detection, it provides Intrusion Prevention.

5. The Cisco Security Agent protects against attacks that it has never seen before, because the protective rules target malicious activity, not particular offending programs or network traffic. Any application that behaves in a way proscribed by the rules will be detected, and the activity stopped. Even applications that have never been seen before—or perhaps not even created when the rule was defined—will be controlled.
6. The Cisco Security Agent is simple to customize. The CSA MC GUI provides hotlinks to the pertinent rules, making it easy to identify which rules need modification to fit your local policy.
7. CSA MC provides a wizard to make customization even easier. The wizard helps automate more complex customization tasks, so that even junior level security administrators can quickly and sensibly adapt the Cisco Security Agent policies to their environment.
8. The Cisco Security Agent protects UNIX systems. Just like we did with Windows, we deployed agents that protected a Solaris system.
9. The Cisco Security Agent manages UNIX and Windows agents from a single console. CSA MC lets you control both Windows and UNIX systems from a single management point.

Frequently Asked Questions

How does the Cisco Security Agent integrate with the kernel of my Windows system?

The Cisco Security Agent uses the same interfaces that are currently used by anti-virus and personal firewall software manufacturers. These interfaces include file system filter drivers and the Network Driver Interface Specification (NDIS).

What is the performance impact incurred by running the Cisco Security Agent?

Because the Cisco Security Agent does not scan packets for content (unlike AV and IDS products) the performance impact is less than 1-2%.

Do I need to replace any system programs?

Because the Cisco Security Agent intercepts system calls at the operating system level, there is no need to replace any system programs.

What is the impact to Cisco Security Agent when a new service pack is released?

Like all software companies, the Cisco will perform a thorough QA process to maintain currency with the latest Microsoft issued patches and service packs that will interact with the Cisco Security Agent software.

What is the impact of the Cisco Security Agent to newly installed executables on a system?

Provided that the new software does not change the behavior of existing the Cisco Security Agent protected applications, there is no expected impact.

Do I need to develop policies in order to start using the Cisco Security Agent?

The Cisco Security Agent ships with 'out of the box' agent kits and policies for default desktop and server systems. This makes the Cisco Security Agent immediately useful for securing your environment. Most organizations start with these policies, because they are designed to stop malicious activity. It also ships with a number of default policies for popular Microsoft applications such as IIS, SQL Server, and Office, for use by organizations that want to implement policy-based control over application behavior. These can be supplemented by new, user-defined policies as a the Cisco Security Agent implementation proceeds in scope.

What administrative overhead accompanies the Cisco Security Agent policy development and deployment?

The Cisco Security Agent application-centric policies are easy to deploy because they don't need to be assigned to the user population, only to hosts; they can be simply evaluated without impacting production environments, through the use of test mode; and they can be disseminated simply through the enterprise by using host groups and agent polling.

What Sort Of Correlation Does the Cisco Security Agent Provide?

The Cisco Security Agent utilizes real-time correlation at both the agent and the global level. This provides greater accuracy for decision-making at the agent level and enables security to be dynamically adapted across the enterprise in reaction to events that occur on distributed hosts. The Cisco Security Agent real-time correlation enables the following functionality: Correlation of network scans (ping scans and port scans) Correlation and dynamic quarantining of files based on email worm events Correlation of OS events from host event logs Correlation of events received from AV scanners (files can be added to the Cisco Security Agent dynamic quarantine list).

Do Cisco Security Agent Default Policies Help With System Hardening?

Buffer Overflow protection, port scan detection, network worm protection, and Trojan detection are pre-configured rules that you can add to your policies in the same way you add other rules. These are basic system hardening features that should be applied in most cases.

How Does the Cisco Security Agent Protect Against Buffer Overflows?

When malicious code attempts to overrun buffers, the Cisco Security Agent can detect and prevent the accessing of system functions by code executing in data or stack space. This functionality was the key to preventing notorious buffer overrun attacks like Code Red and Nimda. Note that the Cisco Security Agent provides automatic protection to all applications from both Stack and Heap overflow attacks.

How Do Cisco Security Agent Policies Protect My Applications Against SYN Floods?

SYN floods results in half open connections on the server. An abundance of half open states on a server can prevent legitimate connections from being established. The Cisco Security Agent policies help to prevent the proliferation of half open states. You should apply SYN flood protection to servers within your enterprise, keeping them up and running and able to provide resources should a SYN flood attack occur.

How Do Cisco Security Agent Policies Protect My Applications Against Port Scans?

Using port scan protection in a policy causes the intelligent agent on a system to log an event when an attempt is made to scan the system for an open port. This can warn you if someone is mapping out your system in preparation for an attack. The intelligent agent also gathers information on the source IP addresses perpetrating a scan and it reveals the source address of the latest scan attempt. If scans are detected across several machines, the Cisco Security Agent correlates against these events and generates an additional event to warn of this correlation.

How Do Cisco Security Agent Policies Protect Against The Propagation Of Network Worms?

When a worm of this type is received through email and executed by unsuspecting users, it generally attempts to send copies of itself to all entries in the email address book of the user. In doing this, the worm modifies registry keys, writes its own script files, and modifies existing files. When this

type of suspicious activity is detected, the intelligent agent queries the user, informing the user of this activity. When the user elects to stop the system action on the desktop where the worm was received, the network worm is prevented from propagating itself. The pre-configured Cisco Security Agent network worm protection rule also correlates a series of suspicious events across multiple machines.

How Do Cisco Security Agent Policies Protect My Applications Against Trojan Horses?

The included Trojan detection rule lets you enable several different types of Trojan detection. Trapping of keystrokes by network applications. (Detect and prevent applications that attempt to capture system keystrokes.) Accessing memory owned by other applications (Detect and prevent applications that attempt to interfere with the memory space of other applications.) Stealing local passwords (Detect and prevent applications that attempt to steal local system passwords.) Downloading and invoking executable file. (Detect and prevent applications that download executables and immediately attempt to execute them. This is a type of buffer overrun attack that takes the form of an email attachment executable file) Downloading and invoking ActiveX controls (Detect and prevent malicious applications acting like a Web browser).

Java code for backdoor program

The following Java application test illustrates what an attacker might use as a "back door" on a server.

Please note the following:

- This program is provided as an example only, and is not supported.
- To run the program you need to have Java Runtime Environment (JRE) installed on your machine. You can download it from <http://java.sun.com/>. Note that most Solaris builds include this by default.
- To run the executable you would execute this from the command line (note that this is case sensitive; you must type "Server" and not "server"):

```
# java Server [protocol] [port] ===>CASE SENSITIVE!, so you must
type "Server" and not "server"
```

example:

```
# java Server tcp 25000
"Whenever in doubt of the arguments, just execute it without any
arguments to get the usage statement."
```

```
/** Server.java
 * Version: 1.0
 * Author: Veronika
 * Date: 4/01/02
 */
import java.io.*;
import java.net.*;
import java.util.*;

/**
 * This program takes 2 command line arguments from the user
 * 1.Protocol to be used: TCP or UDP
 * 2.Port number: 0 to 65,535
 *
 * The server listens on a port and protocol specified by the
user listening
 * for a client connection. Once client connected, it
receives and displays
 * a text message that the client sent.
```

```

*
*/

public class Server
{
    private final static int MINPORT = 0;
    private final static int MAXPORT = 65535;

    public static void main (String[] args) throws Exception
    {
        String protocol; // protocol
        int port; // port number

        //extract and authenticate command line arguments
        try
        {
            protocol=args[0];
            port=Integer.parseInt(args[1]);

            //Check protocol name specified for invalid name
            if (protocol.equalsIgnoreCase("TCP"))
                tcpConnect(port);
            else
                if (protocol.equalsIgnoreCase("UDP"))
                    udpConnect(port);
                else
                {
                    System.out.println("Error: Invalid protocol
name used!");
                    System.out.println("Exiting...");
                    System.exit(0);
                }

            //check port number in range
            if (port < MINPORT || port > MAXPORT)
            {
                System.out.println("Error: Port number is out of
range!");
                System.out.println("Exiting...");
                System.exit(0);
            }
        } catch (Exception e)
        {

```

```
        System.out.println("Usage: Server [protocol]
[port]");
        System.out.println("Where - [protocol] TCP or UDP
only");
        System.out.println("        - [port] is the port
number from 0 to 65,535");
        System.out.println("Example: Server TCP 80 ");
        System.exit(0);
    }
} //end of main

public static void tcpConnect(int port)
{
    String message;

    try{
        //create a TCP socket for the client to 'knock' on
        ServerSocket listenSocket = new ServerSocket(port);
        System.out.print("Listening on port " + port + "
using TCP... ");
        do //loop waiting for a connection from client
        {
            Socket connectionSocket=listenSocket.accept();
            System.out.println("Received a connection from
client.");

            //establish in/out streams
            BufferedReader inFromClient=new
BufferedReader(new
InputStreamReader(connectionSocket.getInputStream()));
            DataOutputStream outToClient=new
DataOutputStream(connectionSocket.getOutputStream());

            //read the message from client
            System.out.print("Receiving a message from
client... ");
            message=inFromClient.readLine();
            System.out.println("Done.");

            //print message received from client
            System.out.print("FROM CLIENT: ");
            System.out.println(message);

            //close the connection
```

```

        System.out.print("Closing connection to client...
");
        connectionSocket.close();
        System.out.println("Done");
    } while (true);
} catch (Exception e)
{
    System.out.println("Error: " + e);
}

} //end of tcpConnect
public static void udpConnect(int port)
{
    try{
        //create a UDP socket to listen on
        DatagramSocket serverSocket = new
DatagramSocket(port);
        System.out.println("Listening on port " + port + "
using UDP.");

        byte[] receiveData = new byte[1024];
        while(true)
        {
            DatagramPacket receivePacket = new
DatagramPacket(receiveData, receiveData.length);

            //receive a packet from client
            System.out.print("Waiting for a datagram
packet...");
            serverSocket.receive(receivePacket);
            System.out.println("Received.");

            //extract message received from client
            String message = new
String(receivePacket.getData());
            System.out.println("FROM CLIENT: " + message);

        } //end of while
    } catch (Exception e)
    {
        System.out.println("Error: " + e);
    }
}
}

```

```
} // end of server class
```

