CHAPTER 7

Posture Plugins

Posture plugins are the means by which Cisco Trust Agent (CTA) retrieves posture credentials from NAC-compliant applications installed on a client.

Typically, two files comprise a posture plugin. A posture plugin for Windows consists minimally of a dynamic link library ".dll" file and an information ".inf" file. A posture plugin for Linux consists of a shared object ".so" file and an information ".inf" file.

Posture plugins gather posture credentials from NAC-compliant applications. Posture credentials are information about an application that determines the trust the network should have in the security of that application. Posture credentials may include these kinds of attributes: application name, application version, application release date, or proprietary application settings or configurations. Posture credentials can be different for each NAC-compliant application.

Once the posture plugin gathers the posture credentials, it sends them to the CTA. CTA sends the posture credentials to Cisco Secure Access Control Server (ACS) which determines a posture token for each application that provided credentials and an overall posture token for the entire client. The posture token is communicated to the client in a posture notification message.

In the Network Admission Control (NAC) environment, the value of a client's posture token determines the level of network access the client is allowed. When we refer to a client's "posture," we are referring to the value of the client's posture token.

See "Initial Posture Validation Process" section on page 1-2 and "Posture Revalidation Process" section on page 1-4 for a description of the workflows and conditions surrounding these events.

This chapter contains the following sections:

Also see the "Configuring Posture Plugins" section on page 5-13 for information about configuring posture plugins.

Types of Posture Plugins Installed by Default

When CTA is installed these posture plugins are installed by default.

- Host Posture Plugin
- Cisco Trust Agent Posture Plugin

The CTA Scripting Posture Plugin posture plugin is installed if the scripting interface was installed.

Other plugins may also be installed, such as the posture plugin for Cisco Security Agent or a partner in the Network Admission Control program. These plugins are not discussed in this chapter.

Host Posture Plugin

The Host Posture Plugin retrieves basic information about the host and returns it to the ACS.

The default location of the Windows host posture plugin is the \Program Files\Common Files\PostureAgent\Plugins directory. The plugin consists of two files: CiscoHostPP.dll and CiscoHostPP.inf.

The default location of the host posture plugin on Linux and Mac OS X operating systems is the /opt/PostureAgent/Plugins directory. The plugin consists of two files: CiscoHostPP_unix.inf and ciscohostpp.so.

These are the attributes that are returned by the host posture plugin:

Table 7-1 Host Posture Plugin Attributes

Condition name in Posture Validation Policies page of ACS	Attribute Description	Used by these operating system
Cisco:Host:ServicePacks	Windows service packs that have been installed on the client	Windows
Cisco:Host:HotFixes	Windows hot fixes that have been installed on the client	Windows
Cisco:Host:HostFQDN	Machine name	Windows
Cisco:Host:Package	Version and string version of rpm packages	Linux and Mac OS X
Cisco:Host:MACAddress	Machine's MAC address	Linux, Mac OS X, Windows

Note The Host posture plugin only reports Windows operating system hotfixes. It does not report application hotfixes.

MAC Address Information Returned by Host Posture Plugin

The Host Posture Plugin reports basic information about the client running CTA to the ACS. With the release of CTA 2.1, the Host Posture Plugin can now return the MAC address of the client running CTA, provided that the MacAddress attribute has been added to the Posture-Validation Attribute Definition File employed by the ACS CSUtil database utility. (For more information about the ACS CSUtil database utility and the Posture-Validation Attribute Definition File, see the *User Guide for Cisco Secure ACS for Windows Server.*)

The attribute information for MACAddress is below.

```
[attr#n]
vendor-id=9
vendor-name=Cisco
application-id=2
application-name=Host
attribute-id=00009
attribute-name=MACAddress
attribute-profile=in
```

```
attribute-type=string
```

The plugin will return all the MAC addresses available on the client running CTA and combine them into one string; the MAC addresses will be separated by pipes (|). For example, a wireless network card and a wired network card will each return a MAC address.

If you are defining a posture validation rule in ACS based on only one of these MAC addresses, the posture attribute should "contain" the MAC address you are verifying rather than "equal" or "start with" the MAC address you are verifying.

Package Information Retrieved by Host Posture Plugin for Linux Platforms

In addition to the information defined by the host posture plugin attributes, the host posture plugin for Linux and Mac OS X platforms allows you to retrieve the version number of certain packages pre-defined in ACS.

The version number of the package may be expressed in one of these forms:

- A string.
- A representation of the version number in the form of x.x.x.x. In this case, the version numbers is converted to a 4-octet version number using an algorithm.

Here is an example of how CTA returns package version number using each format: Assume there is a posture validation rule from ACS that requests the version number of OpenSSL. When requested as a string, the version number would be returned as a combination of numbers and letters, such as 0.9.7a. When requested as a 4-octet number, the version number that would be return is 0.9.7.97.

Creating posture validation rules on ACS that request package version information as numbers rather than strings allows you to apply operators in the ACS rule such as "greater than or equal to," "greater than," "less than," or "less than or equal to." These operators do not apply to strings. The string format is beneficial when too much information is lost when using the numeric format. For example, a package might have a version "rockie-mnt-rel-Feb", in this case the converted numeric version is reported as "0.0.0.0". In this case, the string version of the package is more meaningful.

When using the Host Posture Plugin on Linux operating systems to retrieve package information, the string value of a package version can be determined in advance of making the ACS rule by running the “rpm -q package-name” command, for example, “rpm -q openssl”. The 4-octet value of a package version is determined from the same output of the “rpm -q package-name” command.

Package Information Retrieved by Host Posture Plugin for Mac OS X Platforms

For Mac OS X, there are two types of applications that are of concern to CTA: system applications which have receipts in /Library/Receipts/ and user applications which are installed in /Applications directory.

System applications are identified by the first level folder name under /Library/Receipts, like "Danish.pkg", "X11SDK.pkg". User applications are identified by the application name under /Applications directory as displayed in Finder. For example, “Firefox”, “DVD\ Player”.

The applications located in the subfolders of /Applications directory can also be queried, in these cases the package name looks like the relative path to /Applications. For example, "Utilities/Disk\ Utility", "Zinio/Zinio\ Reader".

Note White spaces in package names must be escaped with backslash (“\”).

The version information of system applications is parsed out of the Contents/version.plist file under the package's directory under the /Library/Receipts directory. Version information is in the form of "a.b.c.d". The first three fields of version are from the CFBundleShortVersionString key, and the fourth field is from SourceVersion key. For user application packages, the version information is retrieved from the Info.plist file under the Contents/ directory in the application’s directory. We first look for the value of CFBundleShortVersionString key. If this key is not present we will return the value of CFBundleVersion key. If both keys are missing no information will be returned for the package.

Cisco Trust Agent Posture Plugin

By default, CTA installs and registers a posture plugin that provides information about itself. The CTA posture plugin returns information such as CTA’s name, version number, and the name of the operating system on which it runs.

The default location of the Windows CTA posture plugin is the \Program Files\Common Files\PostureAgent\Plugins directory. The plugin consists of two files: ctapp.dll and ctapp.inf.

The default location for the CTA posture plugin on Linux and Mac OS X operating systems is the /opt/PostureAgent/Plugins directory. The plugin consists of two files: ctapp.so and ctapp_unix.inf.

The CTA plugin returns many of the same attributes for all operating systems. You can use the information in table 7-2 when defining rules for the default Cisco Trust Agent plugin in ACS:

Table 7-2 CTA Posture Plugin Attributes and Definitions

Condition name in Posture Validation Policies page of ACS	Attribute Description	Used by this operating system
Cisco:PA: Application-Posture-Assessment	This is the value of the Application-posture token from CTA.	Linux, Mac OS X, Windows
Cisco:PA:Kernel-Version	Kernel version, which is the same as the output of the “uname -r” command.	Linux, Mac OS X,
Cisco:PA:MachinePostureState	Contains the running status of the machine. Linux and Mac OS X return these values: • Booting (ACS value = 1) • Running (ACS value = 2) Windows platforms return these values: • Booting (ACS value = 1) • Running (ACS value = 2) • Logged in (ACS value = 3 See Machine Posture State, page 7-8 for more information on the use of this attribute.	Linux, Mac OS X, Windows.
Cisco:PA:OS-Release	A string that contains the OS Kernel name, version, and hardware platform.	Linux, Mac OS X,

Condition name in Posture Validation Policies page of ACS	Attribute Description	Used by this operating system
Cisco:PA:OS-Type	Contains the name of the operating system running on the client. Linux platforms return these names: • Red Hat Enterprise Linux WS • Red Hat Enterprise Linux ES • Red Hat Enterprise Linux AS Mac OS X returns these names: • Mac OS Panther • Mac OS Tiger Windows platforms return these names: • Windows Server 2003 Enterprise Edition • Windows Server 2003 Web Edition • Windows Server 2003 Standard Edition • Windows XP Home Edition • Windows XP Professional • Windows 2000 Advanced Server • Windows 2000 Server	Linux, Mac OS X, Windows
Cisco:PA:OS-Version	The operating system version number in the format *major.minor.sustaining.build.*	Linux, Mac OS X, Windows
Cisco:PA:PA-Name	Contains the name of the posture agent running on the client. In this case, the name would be Cisco Trust Agent.	Linux, Mac OS X, Windows
Cisco:PA:PA-Version	The Cisco Trust Agent version number in the format *major.minor.sustaining.build.*	Linux, Mac OS X, Windows

Condition name in Posture Validation Policies page of ACS	Attribute Description	Used by this operating system
Cisco:PA: System-Posture-Assessment	This indicates the overall posture token of a client on which CTA runs.	Linux, Mac OS X, Windows
Cisco:PA:User-Notification	Contains an informational message that the Cisco Trust Agent displays to the user on request from the ACS server.	Linux, Mac OS X, Windows

Machine Posture State

Machine posture state is provided by CTA to inform ACS about the status of the machine when it boots. One of the following states can be reported:

- Booting
- Running
- Logged in (Not supported on Linux)

Booting is the initial state when the machine is started. The state is set to running when all services are started. On Linux, running is the final state because there is no simple way to determine if a user has logged in. On Windows, when a user has logged in, the logged in state is available. The machine posture state is set once a user has logged into the machine. If the user logs out of the machine, the state is set back to the running state until another user logs in.

This state is used with ACS to determine the posture of a machine based on the rules set up for different policies. For example, the following policy could be set on ACS.

Rule	Posture Token
Antivirus Enabled = TRUE	Healthy
Antivirus Installed and Cisco Trust Agent Machine State = Booting	Transitional
Antivirus Installed = FALSE	Quarantine

CTA Scripting Posture Plugin

The ctascriptPP retrieves the posture credentials requested by a third party script.

A binary posture plugin consists of two parts, a dynamic link library (.dll) file for Windows systems, or a shared object (.so) file for Linux systems, and a .inf file. The .inf file points to the .dll (or .so) file that retrieves the posture credentials. Typically, these are pairs of files; one .inf file is associated with one .dll or .so file.

A script can be substituted for a binary posture plugin. However, the script still needs the .dll (or .so) file and the .inf file to retrieve the posture credentials. The .inf file supplied by the third party must always point to the ctascriptPP file. In the case of a script, many unique .inf files point to one .dll (or .so) and that is the ctascriptPP file.

See Chapter 11, “Using the Scripting Interface” for more information on the CTA scripting interface.

Plugin Installation and Upgrade

Each NAC-compliant application is responsible for installing its own posture plugin on end systems.

Plugins for Windows environments are installed in this directory:

\Program Files\Common Files\PostureAgent\Plugins\Install

Plugins for Linux and Mac OS X environments are installed in this directory:

/opt/PostureAgent/Plugins/install

When CTA receives a posture request, it scans the PostureAgnt\Plugins\Install directory for new or updated posture plugins. If there are new or updated posture plugins in the PostureAgnt\Plugins\Install directory, CTA performs one of the following actions:

- If the .dll (Windows) or the .so (Linux and Mac OS X) plugin **does not exist** in the PostureAgent\Plugins directory, CTA moves the plugin files from the PostureAgent\Plugins\Install directory to the PostureAgent\Plugins directory.

- If the .dll (Windows) or the .so (Linux and Mac OS X) plugins **does exist** in the PostureAgent\Plugins directory, then CTA checks to see if the plugin, in the PostureAgent\Plugins\Install directory, is newer than the one in the Plugins directory. CTA then moves the newer plugin to the PostureAgent\Plugins directory and overwrites the older one. If the plugin in the PostureAgent\Plugins\Install directory is older than the one in the Plugins directory, CTA deletes it, and continues to use the original plugin.
- If the plugin creates an error during registration, CTA moves the plugin to one of the following directories (if the logging is enabled, the error information is logged):

  Windows:

  ```
  \Program Files\Common Files\PostureAgent\Plugins\Quarantine
  ```

  Linux and Mac OS X:

  ```
  /opt/PostureAgent/Plugins/Quarantine
  ```

Note Quarantined plugins do not participate in posture validation.

You do not need to install CTA before other NAC-compliant applications in order for CTA to make use of their plugins. All plugins are stored in a common directory. When CTA receives a request for posture credentials, it checks the common directory for new plugins before it proceeds to retrieve the posture credentials.