CHAPTER 5

Configuring Cisco Trust Agent

Cisco Trust Agent (CTA) may be configured by making changes to its ctad.ini file. The tasks in this chapter describe configuring CTA's communication of posture data to the Cisco Secure Access Control Server as well as the communication of that posture to the client. To configure CTA Logging, see Chapter 6, "Cisco Trust Agent Event Logging".

This chapter contains the following sections:

The ctad.ini Configuration File

The ctad.ini configuration file is a plain text file that contains the parameters for the Cisco Trust Agent's communication settings, user notifications, certificate filtering rules, and the scripting interface feature.

Some parameters are shared by Linux, Mac OS X, and Windows environments while other are used only for a particular operating system. See the "Sample Windows ctad.ini File" section on page 5-28, "Sample Linux ctad.ini File" section on page 5-32, and "Sample Mac OS X ctad.ini File" section on page 5-37 for examples of ctad.ini files.

The template ctad.ini file is named ctad-temp.ini and is installed during the Linux, Mac OS X, and Windows installations. The ctad-temp.ini file for Linux and Mac OS X are stored in the /etc/opt/CiscoTrustAgent/ directory. The ctad-temp.ini file for Windows is stored in the \Program Files\Cisco Systems\CiscoTrustAgent\ directory.

Editing the ctad.ini Configuration File

In order to edit the ctad.ini file you must have administrative privileges.

Step 1 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 2 If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini".

Step 3 Open the ctad.ini file in a plain text editor.

Step 4 Locate the section you want to edit.

Step 5 Delete the semicolon (;) in front of the parameter you want to edit.

Step 6 Adhering to the value ranges defined in the ctad.ini file, change the value of the parameter.

Step 7 Save the ctad.ini file and close the file.

Step 8 Activate changes in the CTA environment:

- If your changes were to the [main], [UserNotifies] or [ServerCertDNVerification] sections of the ctad.ini file, the new values will be re-read when they are needed.
- If your changes were to the [EAPoUDP] section, reboot the system on which CTA runs.

ctad.ini Configuration Parameters

Table 5-1 explains the sections and parameters used in the ctad.ini file by Linux, Mac OS X, and Windows operating systems.

Table 5-1 ctad.ini Configuration Parameters

Keyword	Description	Operating System(s)
[main]	Heading for main section of ctad.ini file.	Linux, Mac OS X, Windows
EnableVFT	The "EnableVFT" parameter indicates if Validation-Flag TLV is enabled on the version of IOS running on the Network Access Device (NAD). "EnableVFT" enables CTA to operate with the NAD wether it has support for Validation-Flag TLV or not. Default Value: 0 Range of Values: 0, 1 0 = IOS does not support Validation-Flag TLV 1 = IOS does support Validation-Flag TLV	Linux, Mac OS X, Windows

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
PPInterfaceType	The PPInterfaceType parameter describes how CTA gathers posture plugin information. Default Value: Block Range of Values: Block, NonBlockConcurrent, NonBlockSerial • Block: CTA will request posture information from one plug-in at a time. It will not request information from the next plugin until it has received the posture credentials from the current plug-in. • NonBlockConcurrent: CTA requests posture information from all posture plugins simultaneously. The operation of gathering posture credentials ends when all the posture information is returned or the PPWaitTimeout value is reached, whichever is sooner. • NonBlockSerial: CTA requests posture information from one plug-in at a time. It will not request information from the next plugin until it has received the posture credentials from the current plug-in. The operation of gathering posture credentials ends when all the posture information is returned or the PPWaitTimeout value is reached, whichever is sooner.	Linux, Mac OS X, Windows
PPWaitTimeout	The PPWaitTimeout is only applicable if the PPInterfaceType is set to NonBlockConcurrent or NonBlockSerial. The parameter defines the maximum time allowed, in seconds, to complete the processing of all plugin queries. Should this timer expire while waiting for responses, CTA will send all of the data that it received thus far in the exchange. Default value: 5 seconds Range of values: 1 - 300 seconds	Linux, Mac OS X, Windows

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
PPMsgSize	The PPMsgSize parameter allows the administrator to modify the maximum message size that a posture plugin can send from 1k to a maximum value of 6k. Default value: 1024 bytes Range of values: 1024 – 6144 bytes	Linux, Mac OS X, Windows
*PluginName*_PPMsgSize	An application-specific posture plugin message size may be added to the ctad.ini file. When added, the posture plugin this parameter references will provide a posture message of this size and it will ignore the value provided by PPMsgSize. See Configuring an Application-Specific Posture Plug-in Message Size, page 5-17. Range of values: 1024 – 6144 bytes	Linux, Mac OS X, Windows
SQTimer	The status query timer (SQTimer) parameter defines how often CTA queries the posture plugins to detect a change in their status. Default value: 300 seconds Range of values: 5 - 4294967295 seconds	Linux, Mac OS X, Windows
[EAPoUDP]	Section head for Cisco Trust Agent using EAP over UDP protocol to communicate with Cisco Secure ACS.	Linux, Mac OS X Windows
LocalPort	The LocalPort parameter defines the port on which the NAD initiates posture validation with CTA. Changing this value requires changes to the NAD configuration. Default value: 21862 Rang of values: 1 - 65550	Linux, Mac OS X, Windows

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
MaxSession	CTA supports one established session per NAD. But CTA can support concurrent sessions with multiple NADs. MaxSession is the number of sessions you allow CTA to support. Default value: 8 Range of values: 1 - 256	Linux, Mac OS X, Windows
SessionIdleTimeout	The SessionIdleTimeout parameter defines the number of seconds an EAP over UDP session can remain idle before timing out. Default value: 3600 Range of values: 60 - 172800	Linux, Mac OS X, Windows
BootTimeUDPExemptions	The BootTimeUDPExemptions parameter alters the Windows Firewall policy and enable the CTA to receive packets when the Windows XP SP2 and SP3-based computer is starting. Default value: 1 Range of values: 0, 1 0 = Windows Firewall Boot Time Exemptions are disabled. 1 = Windows Firewall Boot Time Exemptions are enabled. **Note** Use of the BootTimeUDPexemptions parameter is relevant only when used in conjunction with Microsoft's hot fix for Windows XP (KB17730)	Windows
[UserNotifies]	The [UserNotifes] section defines the behavior of pop-up windows containing messages sent from ACS to CTA.	Linux, Mac OS X, Windows

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
SysModal	SysModal specifies whether the user notification dialog boxes are modal or not. If the notification dialog boxes are modal, the user must close the notification dialog box to continue working. Also, if the dialog boxes are modal, and there is more than one notification, only the last notification appears. Default value: 1 Range of Windows values: 0, 1 0 = Modal dialog boxes are disabled. 1 = Modal dialog boxes are enabled.	Windows
UserActionDelayTimeout	If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message. Setting this parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. Default value: 25 seconds Range of values: 0 - 4294967295 seconds	Linux, Mac OS X, Windows
EnableNotifies	The EnableNotifies parameter enables or disables user notifications. If the EnableNotifies parameter is enabled, a clickable URL may also be presented in the pop-up browser window that contains the posture result. The end user can click the URL link to go to a browser page that contains additional information provided by the ACS administrator. This parameter applies to logged-in users. Default value: 1 Range of values 0, 1 0 = User notifications are disabled. 1 = User notifications are enabled.	Linux, Mac OS X, Windows

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
MsgTimeout	The MsgTimeout parameter specifies how long, in seconds, user notification dialog boxes are displayed. Default value: 300 Range of values: 30 - 4294967 Special value: 0 (disables the timeout)	Linux, Mac OS X, Windows
EnableLogonNotifies	Enables or disables user notification received before the user is logged on. Default value: 1 Range of values: 0, 1 0 = User notifications received before the user is logged on are discarded. 1 = User notifications received before the user is logged on are saved and displayed to the user when they log on.	Linux, Mac OS X, Windows
LogonMsgTimeout	Specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies enabled. Default value: 86400 Range of values: 30 - 4294967 Special value: 0 (disables the timeout)	Linux, Mac OS X, Windows
DisplayType	The DisplayType parameter determines if messages sent from ACS to CTA are displayed in a graphic user interface or in a terminal window. Default value: gui Range of Linux values: term, gui Range of Mac OS X values: gui	Linux

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
TermFont	The TermFont parameter sets the font in which to display the terminal screen text. Use the default value. The TermFont value for Asian Languages will be implemented in a future release. Default value: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1 TermFont entry below for Asian languages: TermFont=-misc-zysong18030-medium-r-normal--0-0-0-0-c-0-iso10646-1	Linux
ClearOldNotification	The ClearOldNotification parameter clears or saves notification messages. Default value: 1 Range of values: 0,1 0 = Notification messages are saved. 1 = CTA clears the old notification message before displaying the new window.	Linux, Mac OS X
BrowserPath	The BrowserPath parameter specifies the full path of the browser on Linux systems. • For Red Hat Enterprise Linux v3 (Enterprise, Advanced, Workstation) use this path: **/usr/bin/mozilla** • For Red Hat Enterprise Linux v4 (Enterprise, Advanced, Workstation) use this path: **/usr/bin/firefox**	Linux

Table 5-1 ctad.ini Configuration Parameters (continued)

Keyword	Description	Operating System(s)
[ServerCertDNVerification]	The [ServerCertDNVerification] section contains configurable parameters for distinguished name (DN) matching. When using CA certificates to validate your Cisco Secure ACS server certificate, you can implement additional security using distinguished name matching. See Configuring Certificate DN Matching, page 5-27 for more information on these parameters.	Linux, Mac OS X, Windows
TotalRules	The TotalRules parameter defines the number of DN matching rules that follow it. If the number of rules is greater than 1 the rules are connected by an OR statement. Default value: (none) Range of values: 0 - 64 Special value: 0 (Disables DN matching)	Linux, Mac OS X, Windows
RuleX	The RuleX parameters are DN matching rules, where X is the index for the rule. These are examples: Rule1=CN*"Server", ISSUER-CN*"Finance" Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"	Linux, Mac OS X, Windows
[Scripting_Interface]	The parameters in the [Scripting_Interface] section define the scripting interface	Linux, Mac OS X, Windows
delta_stale	The delta_stale parameter specifies how long, in minutes, before the posture database record is deemed outdated. Default value: 43200 Range of values: 1-5256000 Special value: 0 (the database will never expire)	Linux, Mac OS X, Windows

Configuring EAP over UDP Communication

CTA can communicate with a router or switch using the Extensible Authentication Protocol over User Datagram Protocol (EAP over UDP). When CTA uses EAP over UDP to communicate with a router, this is referred to as the NAC L3 IP method. When CTA uses EAP over UDP to communicate with a switch, this is referred to as the NAC L2 IP method. In these cases, you can configure the NAC L3 IP and NAC L2 IP communication in the [EAPoUDP] section of the ctad.ini configuration file.

These configurations are optional. You are not required to change the default values of these parameters in order for CTA to run properly after installation.

Note On Windows systems, CTA can also communicate with a switch, through a supplicant such as the Cisco Secure Services Client using the IEEE 802.1x protocol. That communication is not configurable in the ctad.ini file.

To configure EAP over UDP communication, use the Editing the ctad.ini Configuration File, page 5-3 procedure, reference the sample ctad.ini files, and reference the ctad.ini Configuration Parameters, page 5-4.

You can configure the following communication settings for CTA:

- LocalPort — By default, CTA listens on UDP port 21862. If you change this setting, you need to configure your network access device to initiate the posture validation process on the new port number.
- MaxSession — CTA only supports one established session per network access device, but can support sessions from multiple, different network access devices at the same time. CTA can support up to 255 sessions at one time.
- SessionIdleTimeout — This specifies the number of seconds an EAPoUDP session can remain idle before timing out.
- BootTimeUDPExemptions — The BootTimeUDPExemptions parameter alters the Windows Firewall policy and enables CTA to receive packets when the Windows XP SP2 or SP3-based computer is starting.

  By enabling BootTimeUDPExemptions you alter the Windows XP Firewall setting by adding our local EAPoUDP port to the Windows XP Firewall boot time UDP exemptions policy. This enables CTA to communicate with ACS over the network.

Note Use of the BootTimeUDPexemptions parameter is relevant only when used in conjunction with Microsoft's hot fix for Windows XP (KB17730)

Configuring Posture Plugins

These are the aspects of posture plugin behavior that are configurable:

- The interaction between CTA and posture plugins for the collection of posture data, transferring notifications, and status updates.
- The message size a posture plugin provides to CTA.
- Reporting behavior of legacy posture plugins.

Configuring CTA and Posture Plugin Interaction

CTA and the posture plugins interact for the transfer of posture data, posture notifications, and status updates. The PPInterfaceType and PPWaitTimeout parameters are used together to determine how CTA interacts with the plugins and how long the interaction with all plugins lasts. The procedure below describes how to set the values of PPInterfaceType and PPWaitTimeout.

Step 1 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 2 If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini".

Step 3 Locate the [Main] section in the ctad.ini file.

Step 4 Delete the semicolon (;) in front of the **PPInterfaceType** and set the parameter to Block, NonBlockSerial, or NonBlockConcurrent. See the description of these values in the "ctad.ini Configuration Parameters" section on page 5-4.

Step 5 Delete the semicolon (;) in front of the **PPWaitTimeout** parameter and set the PPWaitTimeout period to the number of seconds you require for all posture plugins to return their posture information.

Note The maximum setting for PPWaitTimeout cannot be more than the maximum time that the network access device allows the host to respond to posture requests. If PPWaitTimeout is set at too high a value, the entire posture request will timeout and posture will be re-requested.

Step 6 Save and close the ctad.ini file.

See Example 5-1 for a description of how these parameters interact during a posture request.

Example 5-1 Interaction of PPInterfaceType and PPWaitTimeout parameters

The examples that follow describe the effect of the values of PPInterfaceType and PPWaitTimeout during a request for posture information. The interaction between CTA and the plugin for the transfer of a notification or a status update would also follow the same workflow.

For these examples, assume that these are the posture plugins on the client that return posture credentials and that they take the stated amount of time to return those posture credentials to CTA:

Posture Plugin	Time to collect posture credentials and send them to CTA
CiscoHostPP	0.5 seconds
CTAPP	0.5 seconds
AntivirusPP	3.0 seconds
ApplicationPP	2.0 seconds

Scenario 1—PPInterfaceType is set to “Block” and PPWaitTimeout is set to 5 seconds.

Because PPInterfaceType is set to “Block” one plugin must collect its credential information before the next plugin can collect its posture credential information. Once all the posture information is collected it is sent to CTA.

In this case, it would take 6 seconds for the full amount of posture credentials to be collected before being sent to CTA. The PPWaitTimeout has no effect on the collection of this data because that parameter is only relevant when PPInterfaceType is set to either NonBlockConcurrent or NonBlockSerial.

Scenario 2—PPInetrfaceType is set to NonBlockConcurrent and PPWaitTimeout is set to 5 seconds.

Because PPInterfaceType is set to "NonBlockConcurrent" all the plugins can collect their posture data simultaneously.

In this example, it would take 3 seconds for all the posture credential information to be collected. The PPWaitTimeout parameter is greater than 3 seconds, so it has no effect on the transaction.

Scenario 3—PPInterfaceType is set to NonBlockSerial and PPWaitTimeout is set to 5 seconds.

Because PPInetrfaceType is set to NonBlockSerial the plugins collect their posture data one at a time. In this scenario, the following would occur:

1. CTA requests posture credentials from CiscoHostPP.
2. CiscoHostPP collects its posture credentials.
3. CTA request posture credentials from CTAPP.
4. CTAPP collects its posture credentials.
5. CTA requests posture credentials from AntivirusPP.
6. AntivirusPP collects its posture credentials.
7. CTA requests posture credentials from ApplicationPP.
8. ApplicationPP begins collecting its posture credentials.
9. The PPWaitTimeout expires before ApplicationPP can complete collecting posture credentials.
10. CTA sends the posture credentials for CiscoHostPP, CTAPP, and AntivirusPP to the ACS.
11. CTA considers ApplicationPP an "unresponsive plugin" and will not request its posture credentials again until ApplicationPP supplies its posture credentials for the last request. When ApplicationPP does reply with its posture credentials it is no longer considered an "unresponsive plugin."

Tip These scenarios describe how the posture plugin sends posture credentials to CTA. Once CTA has the posture credentials, it forwards them to ACS. ACS then evaluates the posture credentials against the posture validation network access profile.

In the specific case of scenario 3, if the posture credentials of ApplicationPP are required in order for the client to be granted network access, then access may not be granted because ApplicationPP did not have enough time to report its credentials to CTA and CTA did not have the credentials to send them to ACS.

Configuring the Default Posture Plug-in Message Size

By default, plug-ins are permitted to provide 1024 bytes of information to CTA. This number can be increased to allow all plug-ins to provide up to 6KB of information. However, limiting the size of the posture message to as close to 1KB of information as possible allows more applications to provide a posture message and optimizes the reporting time of all the posture messages. The PPMsgSize parameter in the ctad.ini sets the posture plugin message size for all plugins.

You can also set an application-specific posture plugin message size that is different than the default PPMsgSize value for an application that has its own plugin. See "Configuring an Application-Specific Posture Plug-in Message Size" section on page 5-17 for more information.

The maximum amount of posture data that CTA can send to ACS at one time is 16KB.

Note If there is a Symantec posture plugin installed on the client, the ctad.ini file must be configured in one of two ways:

- PPMsgSize must be set to 1024 bytes.
- The Symantec posture plugin must use an application-specific posture plugin set to 1024 bytes

Without using one of these configurations posture plugin messages from any posture plugin will not be transferred to CTA.

Step 1 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 2 If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini."

Step 3 Locate the [Main] section in the ctad.ini file.

Step 4 Delete the semicolon (;) in front of the **PPMsgSize**.

Step 5 Increase the **PPMsgSize** to up to 6144 bytes (6KB).

Step 6 Save and close the ctad.ini file.

Configuring an Application-Specific Posture Plug-in Message Size

The posture plugin message size may be customized for any posture plugin. The default posture plugin message size for all plugins is equal to the value of the PPMsgSize parameter in the ctad.ini file. If that default value is not appropriate for a specific posture plug-in, you can specify an application-specific PPMsgSize parameter value.

The application-specific PPMsgSize parameter is added to the ctad.ini file in the [main] section and it uses this naming convention: ***PluginName*_PPMsgSize,** where ***PluginName*** is the name of plugin as specified in the posture plugin's information file.

For example, assume you need to set XYZApplication's unique posture plugin message size to 4096 bytes. Suppose XYZApplication posture plugin's information file defines its posture plugin .dll file name like this:

```
[main]
PluginName=XYZApplicationPP.dll
```

In this example, the name of the new PPMsgSize parameter for XYZApplication you would create would be **XYZApplicationPP_PPMsgSize**.

Note If there is a Symantec posture plugin installed on the client, the ctad.ini file must be configured in one of two ways to maintain the transfer of posture plugin messages from any posture plugin to CTA:

- PPMsgSize must be set to 1024 byte
- The Symantec posture plugin must use an application-specific posture plugin, set to 1024 bytes.

Step 1 In the \Program Files\Common Files\PostureAgent\Plugins or /opt/PostureAgent/Plugins directory, find the .inf file for the posture plugin that requires an application-specific PPMsgSize.

Step 2 Make note of the plugin name in the PluginName field of .inf file.

Step 3 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the \Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 4 If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini."

Step 5 Locate the [main] section in the ctad.ini file.

Step 6 Add a new parameter to the ctad.ini file following the naming convention described previously in this section. The minimum and maximum value for all PPMsgSize parameters remains 1024 bytes and 6144 bytes respectively.

Step 7 Save and close the ctad.ini file.

Configuring PPMsgSize for Host Posture Plugin

If PPMsgSize is less than 4096 bytes and there is no application-specific PPMsgSize parameter set for the host posture plugin, then CTA internally sets the value of CiscoHostPlugin_PPMsgSize at 4096 bytes.

If you create a specific CiscoHostPP_PPMsgSize parameter for the Host posture plugin, it must be set between 4096 bytes and 6144 bytes.

Configuring PPMsgSize for Symantec Posture Plugin

If there is a Symantec posture plugin installed on the client, PPMsgSize must be set to 1024 bytes or the Symantec posture plugin must use an application-specific posture plugin set to 1024 bytes to maintain the transfer of posture plugin messages from any posture plugin to CTA.

Configuring Asynchronous Posture Status Query

CTA can be configured to query posture plugins at regular intervals to determine if there has been a change to their application's status. If a posture plugin alerts CTA that there has been a change in posture status, CTA alerts the network access device which triggers a re-posturing of the host. This is called "asynchronous posture status query." This feature is available on NAC L2 802.1x networks. To use this feature the Cisco Secure Services Client must installed on the client along with CTA.

Asynchronous posture status query can not be used on NAC L2 IP or NAC L3 IP networks. To configure the regular interval at which CTA queries the resident plugins for posture status, perform the following procedure:

Step 1 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 2 If there is a ctad.ini file in the directory, you can edit that file directly. If there is only a ctad-temp.ini file, make a copy of it and save the copy as "ctad.ini".

Step 3 Locate the [Main] section in the ctad.ini file.

Step 4 Delete the semicolon (;) in front of the **SQTimer** parameter.

Step 5 Change the value of SQTimer to reflect the interval at which you want CTA to query posture plugins for a change in posture status.

Configuring User Notifications

User notifications report the “posture” of the host. The messages are sent from Cisco Secure Access Control Server (ACS) to Cisco Trust Agent (CTA). The notifications are displayed as pop-up messages on the desktop, or login screen, of the system on which CTA is installed. These notifications are the only interactive end-user functionality of CTA.

User notifications are configured in the [UserNotifies] section of the ctad.ini configuration file. Any changes made to the [UserNotifies] section of the ctad.ini configuration file are detected and implemented by CTA the next time a notification is received.

The Windows, Linux, and Mac OS X user notification configurations are optional. You do not need to change the default configuration of user notifications in order for CTA to run properly. After reading Configuring Windows User Notifications or Configuring Linux User Notifications, follow the Editing the ctad.ini Configuration File, page 5-3 procedure to make the appropriate changes to the ctad.ini file. Use the sample ctad.ini files and the section on ctad.ini Configuration Parameters, page 5-4 as references.

Configuring Windows User Notifications

To configure user notifications, use the Editing the ctad.ini Configuration File, page 5-3 procedure, reference the sample ctad.ini files, and reference the ctad.ini Configuration Parameters, page 5-4.

You can configure the following notification properties on Windows platforms:

- Where the notifications appear.
  - If the **EnableNotifies** parameter is enabled then user notifications are displayed on the desktop, after the user has logged in.
  - If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.
- How long the notification dialog box displays before it closes automatically.
  - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.

  - The **LogonMsgTimeout** Specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies is enabled.

- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message.
- **SysModal** window behavior is enabled by default. The behavior requires a user to close a notification dialog box to continue working. You can change this behavior by creating a ctad.ini file and disabling the parameter in the file.

Note If the user notification dialog box is set to system modal, and there is more than one notification message, only the newest message appears. Responding to the message closes all of the message dialog boxes.

Configuring Linux User Notifications

To configure user notifications, use the Editing the ctad.ini Configuration File, page 5-3 procedure, reference the sample Linux ctad.ini file, and reference the ctad.ini Configuration Parameters, page 5-4.

You can configure the following notification properties:

- If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.

  If a GUI is not installed with the Linux operating system, these notifications are written to a message file in the /var/opt/CiscoTrustAgent/msg directory.
- How long the notification dialog box displays before it closes automatically.
  - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.
  - The **LogonMsgTimeout** specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies enabled.

- The **DisplayType** parameter allows you to choose between receiving messages in a terminal window or in the GUI.
- The **TermFont** parameter specifies the font that will be displayed in the terminal window.
- The **ClearOldNotifications** parameter clears or saves old notification messages. If ClearOldNotifications is enabled, an old notification is cleared before showing a new notification.
- The **BrowserPath** parameter specifies the full path to the browser CTA should use.
- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message**.**

Configuring Mac OS X User Notifications

To configure user notifications, use the Editing the ctad.ini Configuration File, page 5-3 procedure, reference the sample Mac OS X ctad.ini file, and reference the ctad.ini Configuration Parameters, page 5-4.

You can configure the following notification properties:

- If the **EnableLogonNotifies** parameter in the ctad.ini is enabled, then user notifications received before the user is logged on are saved and displayed to the user when they log on.
- How long the notification dialog box displays before it closes automatically.
  - The **MsgTimeout** parameter defines how long the message is displayed on the desktop.
  - The **LogonMsgTimeout** specifies how long, in seconds, a message is saved when no user is logged on and when EnableLogonNotifies enabled.
- The **ClearOldNotifications** parameter clears or saves old notification messages. If ClearOldNotifications is enabled, an old notification is cleared before showing a new notification.

- The **UserActionDelayTimeout** parameter allows you to delay the launch of the browser window so that the host has more time to obtain an IP address. If the browser that displays the posture message is launched before the host obtains an IP address, the browser will fail to open the URL contained in the posture message**.**
- **SysModal** window behavior is enabled by default. The behavior requires a user to close a notification dialog box to continue working. You can change this behavior by creating a ctad.ini file and disabling the parameter in the file.

Note If the user notification dialog box is set to system modal, and there is more than one notification message, only the newest message appears. Responding to the message closes all of the message dialog boxes.

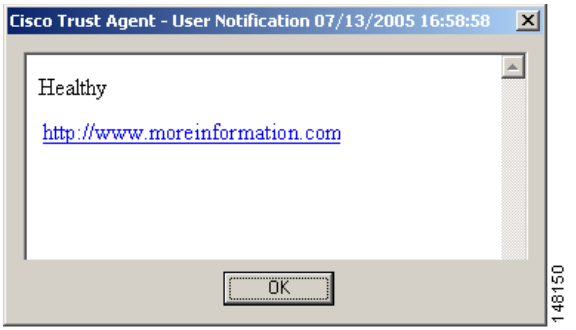
Configuring Clickable URL and Browser Auto-Launch Features

After the overall posture of the client has been determined, the Cisco Secure ACS can be configured to notify the user of that posture. Notifications may be a message in a pop-up window or the user can be directed to a particular URL.

- **Clickable URL**

  The pop-up window that notifies the user will contain a posture value, such as “Healthy” or “Quarantine.” It can also contain a clickable URL for the user to follow. For example, if the client requires remediation, the clickable URL can direct the user to an area where a particular remediation solution is available.

Figure 5-1 Clickable URL Notification Pop-up



- **Browser Auto-launch**

  Instead of receiving a pop-up window with a clickable URL, the notification can be configured to automatically open a browser window which is already pointing to the URL the user requires.

Logging Notifications

If logging is enabled, the following notification events are logged:

- Notification failures, such as a failure to create the notification dialog box.
- User notification messages.

The messages that are logged are filtered by the logging level assigned to the notification component of CTA. This logging level is specified by the CTAMsg parameter in the [LogLevel] section of the ctalogd.ini configuration file. If this parameter does not appear in the configuration file, or if the configuration file does not exist, then a default level of 3-High is used. This level allows all informational, warning, and critical messages to be written to the log file. The message severity is assigned by the component or plugin that sent the message.

For more information about logging, see Chapter 6, “Cisco Trust Agent Event Logging”.

Certificate Distinguished Name Matching

Certificate distinguished name (DN) matching is performed when CTA communicates with ACS using the NAC L3 IP and NAC L2 IP methods. Similar functionality can be obtained for NAC L2 802.1x environments by configuring Cisco Secure Services Client to validate trusted servers.

When using CA certificates to validate your ACS server certificate to CTA, you can implement additional security using DN matching to validate the certificate. This prevents other servers or processes that may be using the same root certificate from gaining a trust relationship with the host.

DN matching occurs at the end of the transport layer security (TLS) handshake, after the certificate chain is built. After CTA has received ACS's certificate, this rule is employed to ensure that the properties of the certificate fields have the expected values.

If any rule in the [ServerCertDNVerification] section of the ctad.ini file succeeds then the ACS server is authenticated to CTA.

Note There are no default settings for the [ServerCertDNVerification] section; that section in the sample file contains example information. If you use this sample file as the basis for your own ctad.ini file, delete the section or change it to suit your environment.

DN Matching Rule Syntax

Each rule can contain multiple sub-rules, separated by a comma. If a single sub-rule within a rule fails, then the entire rule fails. The maximum length for a single rule is 255 characters.

The following shows the general format for a rule:

```
Rule1=subrule, subrule, subrule, ...
```

Each sub-rule consists of a certificate subject attribute followed by a value:

```
attribute[operator]"value"
```

The following operators are supported:

- **= (equals)**—The value must exactly match the value given in the subrule.
- *** (contains)**—The value must contain the value given in the subrule.

The following DN attributes are supported:

- **CN**—Common Name
- **SN**—Surname
- **GN**—Given Name
- **N**—Unstructured Name
- **I**—Initials
- **GENQ**—Generation Qualifier
- **C**—Country
- **L**—Locality
- **SP**—Province
- **ST**—State
- **O**—Organization
- **OU**—Organization Unit
- **T**—Title
- **EA**—E-mail Address

Note You can also create sub-rules that check certificate issuer attributes by adding the ISSUER- prefix to the attributes listed above. For example, to validate the Common Name of certificate issuer, you would use the attribute ISSUER-CN.

Keeping the syntax in mind, here is an example of two rules:

```
Rule1=CN*"Server", ISSUER-CN*"Finance"
Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"
```

At the end of the TLS handshake the Subject field or Issuer field in the Cisco Secure ACS server certificate are compared to these rules. The DN matching will succeed if one or the other rules succeed.

Rule 1 indicates that the Subject field CN (Common Name) must contain “Server” and the Issuer field CN must contain “Finance.” If both fields do not have the correct information the rule fails.

Rule 2 indicates that the Subject field CN must equal “Finance posture Cert”, the Organization Unit (OU) field must contain “Finance”, and the Issuer CN field must contain “ACME”. If all of these subrules do not have the correct information, the rule fails.

Configuring Certificate DN Matching

Step 1 Locate the ctad.ini file or the ctad-temp.ini template file on the host.

- For Linux and Mac OS X operating systems, navigate to the /etc/opt/CiscoTrustAgent directory.
- For Windows Operating systems, navigate to the Program Files\CiscoSystems\CiscoTrustAgent directory.

Step 2 If there is a ctad.ini file in the directory, you can edit that file directly. If there is **only** a ctad-temp.ini file, make a copy of it and save the copy as “ctad.ini”.

Step 3 Locate the [ServerCertDNVerification] section in the ctad.ini file.

Step 4 Delete the semicolon (;) in front of the TotalRules parameter and equate the parameter to the number of rules you are going to write.

Step 5 Following the example in the ctad.ini file create a new line for each rule. Begin each line with “RuleX=” where X is the number of the rule.

Step 6 Write each rule following the “DN Matching Rule Syntax” section on page 5-25.

Step 7 Save and close the file. The new rules will be implemented the next time DN matching occurs.

Configuring the Scripting Interface

The delta_stale parameter in the [ScriptingInterface] section of the ctad.ini file is the only configurable parameter for the Scripting Interface. This parameter defines the time, in minutes, before the posture data file is considered out of date.

To configure the CTA Scripting Interface, use the Editing the ctad.ini Configuration File, page 5-3 procedure, reference the sample ctad.ini files, and reference the “ctad.ini Configuration Parameters” section on page 5-4.

Sample Windows ctad.ini File

```
; ***************************************
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location for the ctad.ini file is
; the Program Files\CiscoSystems\CiscoTrustAgent\
; directory.
;
; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; ***************************************

; ***************************************
[main]

;The EnableVFT parameter indicates if Validation-Flag TLV
;   is enabled on the version of IOS running on the Network Access
;   Device (NAD). "EnableVFT" enables CTA to operate with the NAD
;   wether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
;   0 = IOS does not support Validation-Flag TLV
;   1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
;   plugin information.
;   Default value: Block
;   Range of values: Block, NonBlockConcurrent, NonBlockSerial
:   Block = CTA will request posture information from one plug-in at a time.
;   NonBockConcurrent = Request posture information from all posture plugins
;   simultaneously.
;   NonBlockSerial = Requests posture information from posture plugins one at
;   a time and waits for either the return of the posture credentials
;   or the end of the PPWaitTimeout value before
;   requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

```
;The PPWaitTimeout parameter represents the maximum time allowed,
;   in seconds, to complete the processing of all plug-ins.
;   Default value: 5 seconds
;   Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
;   the maximum message size that a posture plugin can send
;   from 1k to a maximum value of 6k.
;   Default value: 1024 bytes
;   Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines
;   how often CTA queries the posture plugins to detect a change
;   in their status.
;   Default value: 300 seconds
;   Range of values: 5 - 4294967925 seconds
;SQTimer=300

;The [EAPoUDP] section defines the communication settings between CTA
;   and a Layer 3 Network Access Device (NAD), such as a router.
[EAPoUDP]

;The LocalPort parameter defines the port on which the NAD initiates
;   posture validation with CTA. Changing this value requires
;   changes to the NAD configuration.
;   Default value: 21862
;   Rang of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But CTA can support
;   concurrent sessions with multiple NADs. MaxSessions is
;   the number of sessions you allow CTA to support.
;   Default value: 8
;   Range of values: 1 - 256
;MaxSession=8

;The SessionIdleTimeout parameter defines the number of seconds
;   an EAP over UDP session can remain idle before timing out.
;   Default value: 3600
;   Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The BootTimeUDPExemptions parameter alters the Windows Firewall
;   policy and enables CTA to receive packets when the
;   Windows XP SP2 or SP3-based computer is starting.
```

```
;   Default value: 1
;   Range of values: 0, 1
;   0 = Windows Firewall Boot Time Exemptions are disabled.
;   1 = Windows Firewall Boot Time Exemptions are enabled.
;BootTimeUDPExemptions=1

;The [UserNotifes] section defines the behavior of pop-up windows
;   containing messages sent from ACS to CTA
[UserNotifies]

;SysModal specifies whether the user notification dialog boxes are
;   modal or not. If the notification dialog boxes are modal,
;   the user must close the notification dialog box to continue
;   working. Also, if the dialog boxes are modal, and there is
;   more than one notification, only the last notification appears.
;   Default value: 1
;   Range of values: 0, 1
;   0 = Modal dialog boxes are disabled.
;   1 = Modal dialog boxes are enabled.
;SysModal=1

;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
;   the browser window so that the host has more time to obtain an IP address.
;   If the browser that displays the posture message is launched before the host
;   obtains an IP address, the browser will fail to open the URL contained in the
;   posture message.
;   Default value: 25 seconds
;   Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25

;The EnableNotifies parameter enables or disables user
;   notifications. This parameter applies to logged-in users.
;   Default value: 1
;   Range of values 0, 1
;   0 = User notifications are disabled.
;   1 = User notifications are enabled.
;EnableNotifies=1

;The MsgTimeout parameter specifies how long, in seconds,
;   user notification dialog boxes are displayed.
;   Default value: 300
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;MsgTimeout=300

;The EnableLogonNotifies parameter enables or disables user
;   notifications received before the user is logged on.
;   Default value: 1
```

```
;   Range of values: 0, 1
;   0 = User notifications received before the user is logged on are
;   discarded.
;   1 = User notifications received before the user is logged on are
;   saved and displayed to the user when the log on.
;EnableLogonNotifies=1

;The LogonMsgTimeout Specifies how long, in seconds, a message
;   is saved when no user is logged on and when
;   EnableLogonNotifies is enabled.
;   Default value: 86400
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;   LogonMsgTimeout=86400

;The [ServerCertDNVerification] section contains configurable
;   parameters for distinguished name (DN) matching. When
;   using CA certificates to validate your Cisco Secure ACS
;   server certificate, you can implement additional security
;   using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
;   that follow it. If the number of rules is greater than 1
;   the rules are connected by an OR statement.
;   Default value: (none)
;   Range of values: 0 - 64
;   Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
;   for the rule.
;   NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
;   WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER-CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"

;The parameters in the [Scripting_Interface] section define the
;   scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
;   the posture database record is deemed outdated.
;   Default value: 43200
;   Range of values: 1-5256000
;   Special value: 0 (the database will never expire)
;delta_stale=43200
; ***************************************
```

Sample Linux ctad.ini File

```
; ************************************
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location of the ctad.ini file is the
; /etc/opt/CiscoTrustAgent/ directory.

; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; ************************************

; ************************************
[main]
;The “EnableVFT” parameter indicates if Validation-Flag TLV
;   is enabled on the version of IOS running on the Network Access
;   Device (NAD). “EnableVFT” enables CTA to operate with the NAD
;   wether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
;   0 = IOS does not support Validation-Flag TLV
;   1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
;   plugin information.
;   Default value: Block
;   Range of values: Block, NonBlockConcurrent, NonBlockSerial
:   Block = CTA will request posture information from one plug-in at a time.
;   NonBockConcurrent = Request posture information from all posture plugins
;   simultaneously.
;   NonBlockSerial = Requests posture information from posture plugins one at
;   a time and waits for either the return of the posture credentials
;   or the end of the PPWaitTimeout value before
;   requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

```
;The PPWaitTimeout parameter represents the maximum time allowed,
;   in seconds, to complete the processing of all plug-ins.
;   Default value: 5 seconds
;   Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
;   the maximum message size that a posture plugin can send
;   from 1k to a maximum value of 6k.
;   Default value: 1024 bytes
;   Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines how often CTA queries the posture
;   plugins to detect a change in their status.
;   Default value: 300 seconds
;   Range of values: 5 - 4294967925 seconds
;SQTimer=300

; The [EAPoUDP] section defines the communication settings
;   between CTA and the Network Access Device (NAD).
[EAPoUDP]

;The LocalPort defines the port on which the NAD initiates posture
;   validation with CTA. Changing this value requires changes
;   to the NAD configuration.
;   Default value: 21862
;   Rang of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But can support
;   concurrent sessions with multiple NADs. MaxSessions is
;   the number of sessions you allow CTA to support.
;   Default value: 8
;   Range of values: 1 - 256
;MaxSession=8

;SessionIdleTimeout defines the number of seconds an EAP over UDP
;   session can remain idle before timing out.
;   Default value: 3600
;   Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The [UserNotifes] section defines the behavior of pop-up windows
;   containing messages sent from ACS to CTA
[UserNotifies]
```

```
;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
;   the browser window so that the host has more time to obtain an IP address.
;   If the browser that displays the posture message is launched before the host
;   obtains an IP address, the browser will fail to open the URL contained in the
;   posture message.
;   Default value: 25 seconds
;   Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25

;The EnableNotifies parameter enables or disables user
;   notifications. This parameter applies to logged-in users.
;   Default value: 1
;   Range of values 0, 1
;   0 = User notifications are disabled.
;   1 = User notifications are enabled.
;EnableNotifies=1

;The MsgTimeout parameter specifies how long, in seconds,
;   user notification dialog boxes are displayed.
;   Default value: 300
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;MsgTimeout=300

;The EnableLogonNotifies parameter enables or disables user
;   notifications received before the user is logged on.
;   Default value: 1
;   Range of values: 0, 1
;   0 = User notifications received before the user is logged on are
;   discarded.
;   1 = User notifications received before the user is logged on are
;   saved and displayed to the user when the log on.
;EnableLogonNotifies=1

;The LogonMsgTimeout Specifies how long, in seconds, a message
;   is saved when no user is logged on and when
;   EnableLogonNotifies is enabled.
;   Default value: 86400
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;   LogonMsgTimeout=86400

;The DisplayType parameter determines if messages sent from
;   ACS to CTA are displayed in a graphic user interface or
;   in a terminal window.
;   Default value: gui
;   Range of values: term, gui
;DisplayType=gui
```

```
;The TermFont parameter sets the font in which to display
;   the terminal screen text. Use the default value until
;   Cisco Secure ACS can forward localized messages to CTA.
;   Default value: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1
;TermFont=-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10636-1
;TermFont entry below for Asian languages
;TermFont=-misc-zysong18030-medium-r-normal--0-0-0-0-c-0-iso10646-1

; The BrowserPath parameter specifies the full path of the browser on Linux systems.
;   For Red Hat Enterprise Linux v3 (Enterprise, Advanced, Workstation)
;   use this path: /usr/bin/mozilla
;   For Red Hat Enterprise Linux v4 (Enterprise, Advanced, Workstation)
;   use this path: /usr/bin/firefox
; BrowserPath=

;The ClearOldNotification parameter clears or saves notification
;   messages.
; Default value: 1
; Range of Values:
;   0 = Notification messages are saved.
;   1 = CTA clears the old notification message before displaying
;       the new window.
;ClearOldNotification=1

;The [ServerCertDNVerification] section contains configurable
;   parameters for distinguished name (DN) matching. When
;   using CA certificates to validate your Cisco Secure ACS
;   server certificate, you can implement additional security
;   using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
;   that follow it. If the number of rules is greater than 1
;   the rules are connected by an OR statement.
;   Default value: (none)
;   Range of values: 0 - 64
;   Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
;   for the rule.
;   NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
;   WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER-CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"
```

```
;The parameters in the [Scripting_Interface] section define the
;   scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
;   the posture database record is deemed outdated.
;   Default value: 43200
;   Range of values: 1-5256000
;   Special value: 0 (the database will never expire)
;delta_stale=43200
```

Sample Mac OS X ctad.ini File

```
; ***************************************
; CTAD.INI FILE DEFINITION
; This file defines communication parameters between a
; Network Access Device (NAD) and Cisco Trust Agent (CTA).
; It also defines variables for notifications and certificate
; filtering rules.
;
; This file can be edited with a plain text editor and used
; in a custom installation of CTA.
;
; The default location of the ctad.ini file is the
; /etc/opt/CiscoTrustAgent/ directory.

; GENERAL EDITING INSTRUCTIONS
; To "turn on" a parameter in a section, delete the ; before
; the ParameterName.
; To "turn off" a parameter, type a ; before the ParameterName.
; ***************************************

; ***************************************
[main]
;The “EnableVFT” parameter indicates if Validation-Flag TLV
;   is enabled on the version of IOS running on the Network Access
;   Device (NAD). “EnableVFT” enables CTA to operate with the NAD
;   wether it has support for Validation-Flag TLV or not.
;Default Value: 0
;Range of Values: 0, 1
;   0 = IOS does not support Validation-Flag TLV
;   1 = IOS does support Validation-Flag TLV
;EnableVFT=0

;The PPInterfaceType parameter describes how CTA gathers posture
;   plugin information.
;   Default value: Block
;   Range of values: Block, NonBlockConcurrent, NonBlockSerial
:   Block = CTA will request posture information from one plug-in at a time.
;   NonBockConcurrent = Request posture information from all posture plugins
;   simultaneously.
;   NonBlockSerial = Requests posture information from posture plugins one at
;   a time and waits for either the return of the posture credentials
;   or the end of the PPWaitTimeout value before
;   requesting posture credentials from the next posture plugin.
;PPInterfaceType=Block
```

```
;The PPWaitTimeout parameter represents the maximum time allowed,
;   in seconds, to complete the processing of all plug-ins.
;   Default value: 5 seconds
;   Range of values: 1 - 300 seconds
;PPWaitTimeout=5

; The PPMsgSize parameter allows the administrator to modify
;   the maximum message size that a posture plugin can send
;   from 1k to a maximum value of 6k.
;   Default value: 1024 bytes
;   Range of values: 1024 - 6144 bytes
;PPMsgSize=1024

;The SQTimer (status query timer) parameter defines how often CTA queries the posture
;   plugins to detect a change in their status.
;   Default value: 300 seconds
;   Range of values: 5 - 4294967925 seconds
;SQTimer=300

; The [EAPoUDP] section defines the communication settings
;   between CTA and the Network Access Device (NAD).
[EAPoUDP]

;The LocalPort defines the port on which the NAD initiates posture
;   validation with CTA. Changing this value requires changes
;   to the NAD configuration.
;   Default value: 21862
;   Rang of values: 1 - 65550
;LocalPort=21862

;CTA supports one established session per NAD. But can support
;   concurrent sessions with multiple NADs. MaxSessions is
;   the number of sessions you allow CTA to support.
;   Default value: 8
;   Range of values: 1 - 256
;MaxSession=8

;SessionIdleTimeout defines the number of seconds an EAP over UDP
;   session can remain idle before timing out.
;   Default value: 3600
;   Range of values: 60 - 172800
;SessionIdleTimeout=3600

;The [UserNotifes] section defines the behavior of pop-up windows
;   containing messages sent from ACS to CTA
[UserNotifies]
```

```
;Setting this UserActionDelayTimeout parameter allows you to delay the launch of
;   the browser window so that the host has more time to obtain an IP address.
;   If the browser that displays the posture message is launched before the host
;   obtains an IP address, the browser will fail to open the URL contained in the
;   posture message.
;   Default value: 25 seconds
;   Range of values: 0 - 4294967295 seconds
;UserActionDelayTimeout=25

;The EnableNotifies parameter enables or disables user
;   notifications. This parameter applies to logged-in users.
;   Default value: 1
;   Range of values 0, 1
;   0 = User notifications are disabled.
;   1 = User notifications are enabled.
;EnableNotifies=1

;The MsgTimeout parameter specifies how long, in seconds,
;   user notification dialog boxes are displayed.
;   Default value: 300
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;MsgTimeout=300

;The EnableLogonNotifies parameter enables or disables user
;   notifications received before the user is logged on.
;   Default value: 1
;   Range of values: 0, 1
;   0 = User notifications received before the user is logged on are
;   discarded.
;   1 = User notifications received before the user is logged on are
;   saved and displayed to the user when the log on.
;EnableLogonNotifies=1

;The LogonMsgTimeout Specifies how long, in seconds, a message
;   is saved when no user is logged on and when
;   EnableLogonNotifies is enabled.
;   Default value: 86400
;   Range of values: 30 - 4294967
;   Special value: 0 (disables the timeout)
;   LogonMsgTimeout=86400

;The ClearOldNotification parameter clears or saves notification
;   messages.
; Default value: 1
; Range of Values:
;   0 = Notification messages are saved.
;   1 = CTA clears the old notification message before displaying
```

```
;       the new window.
;ClearOldNotification=1

;The [ServerCertDNVerification] section contains configurable
;   parameters for distinguished name (DN) matching. When
;   using CA certificates to validate your Cisco Secure ACS
;   server certificate, you can implement additional security
;   using distinguished name matching.
[ServerCertDNVerification]

;The TotalRules parameter defines the number of DN matching rules
;   that follow it. If the number of rules is greater than 1
;   the rules are connected by an OR statement.
;   Default value: (none)
;   Range of values: 0 - 64
;   Special values: 0 (Disables DN matching)
;TotalRules=2

;The RuleX parameters are DN matching rules, where X is the index
;   for the rule.
;   NOTE: THE RULES BELOW ARE EXAMPLES ONLY. DO NOT USE THEM
;   WITHOUT MODIFYING THEM TO SUIT YOUR ENVIRONMENT.
;Rule1=CN*"Server", ISSUER-CN*"Finance"
;Rule2=CN="Finance posture Cert", OU*"Finance", ISSUER-CN*"ACME"

;The parameters in the [Scripting_Interface] section define the
;   scripting interface behavior.
[Scripting_Interface]

; The delta_stale parameter specifies how long, in minutes, before
;   the posture database record is deemed outdated.
;   Default value: 43200
;   Range of values: 1-5256000
;   Special value: 0 (the database will never expire)
;delta_stale=43200
```