eTrust® Access Control

Documentation Addendum

r8 SP1

Fourth Edition

This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the documentation for their own internal use, and may make one copy of the related software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the product are permitted to have access to such copies.

The right to print copies of the documentation and to make a copy of the related software is limited to the period during which the applicable license for the Product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in the Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Product References

This document references the following CA products:

- eTrust AC
- CA Single Sign-On (eTrust SSO)
- eTrust® Web Access Control (eTrust Web AC)
- CA Top Secret®
- CA ACF2™
- CA Audit
- CA Unicenter Network and Systems Management (CA NSM)
- Unicenter® Network and Systems Management (Unicenter TNG)
- Unicenter® Software Delivery

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Change History

August 2007 CR (Fourth Edition)

- Changed the name of the HP-UX native package customization script to customize_eac_depot (was customize_eac_pkg).

  This change was introduced in the June CR.

- HP-UX native package installation format is called Software Distributor-UX (SD-UX) format package.

  In earlier versions this format was incorrectly referred to as SysV format package.

- Added a chapter called Bypassing Drivers on Windows.

  This chapter provides information about defining a bypass for drivers on Windows. The functionality was added in the March 2007 CR but was not documented.

- Added a chapter called Disabling Network Interception on Windows.

  This chapter provides information about preventing the network interception hook from initializing at boot time on Windows.

April 2007 CR (Third Edition)

- Added a section called AIX Native Package Installation.

  This section provides information about installp format packages (bff files), for native installation on AIX, that was introduced in this CR.

March 2007 CR (Second Edition)

- Added a chapter called Auditing Access Control Activity for Windows.

  This chapter provides information about improved auditing capabilities that were introduced in this CR. These new capabilities include full auditing, Audit Only mode, improved auditing performance, additional statistics and data, and transparent SID to account name resolution.

February 2007 CR

- Published first version.

  This first version provides information about enhancements that were introduced in this CR, as follows:

  - Outgoing connections bypass on UNIX.
  - sesu support of native Linux options.
  - Software Distributor-UX (SD-UX) packages, for native installation on HP-UX.

Contents

Chapter 1: Introduction

This section contains the following topics:

About this Guide

This guide describes enhancements that were introduced to eTrust AC after r8 SP1, through a cumulative release (CR). It is structured to complement the standard documentation set with each chapter covering a new enhancement and containing either new or replacement topics that are for specific guides in the existing set.

Note: As the functionality is rolled into the next release of eTrust AC, the content in this guide will be integrated into the regular documentation set for that release.

Who Should Use this Guide

This guide was written for security and system administrators who are installing or using an eTrust AC update (CR) containing a new enhancement.

Documentation Conventions

The eTrust AC documentation uses the following conventions:

Format	**Meaning**
`Mono-spaced font`	Code or program output.
Italic	Emphasis or a new term.
Bold	Text that you must type exactly as shown.
A forward slash (/)	Platform independent directory separator used to describe UNIX and Windows paths.

The documentation also uses the following special conventions when explaining command syntax and user input (in a mono-spaced font):

Format	**Meaning**
Italic	Information that you must supply.
Between square brackets ([])	Optional operands.
Between braces ({})	Set of mandatory operands.
Choices separated by pipe (|).	Separates alternative operands (choose one). For example, the following means *either* a user name *or* a group name: `{`*`username`*`|`*`groupname`*`}`
...	Indicates that the preceding item or group of items can be repeated.
<u>Underline</u>	Default values.
A backslash at end of line preceded by a space (\)	Sometimes a command does not fit on a single line in this guide. In these cases, a space followed by a backslash (\) at the end of a line indicates that the command continues on the following line. **Note:** Avoid copying the backslash character and omit the line break. These are not part of the actual command syntax.

Example: Command Notation Conventions

The following code illustrates how command conventions are used in this guide:

```
ruler className [props({all|{propertyName1[,propertyName2]...})]
```

In this example:

- The command name (ruler) is shown in regular mono-spaced font as it must be typed as shown.
- The *className* option is in italic as it is a placeholder for a class name (for example, USER).
- You can run the command without the second part enclosed in square brackets, which signifies optional operands.
- When using the optional parameter (props), you can choose the keyword *all* or, specify one or more property names separated by a comma.

Chapter 2: Bypassing Outgoing Connections on UNIX

This section contains the following new topics for the Release Summary:

This section contains the following replacement topics for the Administrator Guide:

Outgoing Connections Bypass on UNIX

eTrust AC on UNIX lets you bypass outgoing network connection events (in addition to the existing bypass for incoming connections). You can specify ports on which outgoing network connections can be established without eTrust AC authorization checks. Bypassing these ports reduces system load and speeds event processing. Bypassed connection events are not logged in the audit and trace files.

Note: eTrust AC lets you bypass the network connection event only; not any subsequent events that use the network connection (for example, opening a file).

The ports you want to bypass for outgoing connections are defined using the bypass_outgoing_TCPIP configuration setting in the [seosd] section of the seos.ini file.

Important! When you upgrade an older AIX installation, eTrust AC populates the bypass_outgoing_TCPIP configuration setting with the value you have for the bypassing incoming connections configuration setting (bypass_TCPIP).

bypass_outgoing_TCPIP

Defines a comma-separated list of ports for which seos_syscall will not pass outgoing connection events to seosd.

Default: Token not set

Bypass Ports for Network Activity

To specify that all connection events (inbound *and* outbound) related to specific TCP/IP ports can be established without eTrust AC authorization, you can define a bypass for these ports. Bypassing these ports reduces system load and speeds event processing. Bypassed connection events are not logged in the audit and trace files.

Note: eTrust AC lets you bypass the network connection event only; not any subsequent events that use the network connection (for example, opening a file).

Trusted inbound connections are specified separately from outbound connections:

- To bypass *incoming* connections, modify the *bypass_TCPIP* configuration setting in the [seosd] section of the seos.ini file.
- To bypass *outgoing* connections, modify the *bypass_outgoing_TCPIP* configuration setting in the [seosd] section of the seos.ini file.

Note: For more information about the seos.ini initialization file, updating tokens, and affecting changes, see the *Reference Guide*.

Example: Bypass incoming Telnet events

If you set the bypass_TCPIP configuration setting to 23 (the Telnet port), the audit and trace files no longer log the network event when you Telnet *to* that workstation. Events related to other services, such as ssh, login, and FTP, and subsequent events that use the network connection (for example, opening a file), will still be logged.

Example: Bypass outgoing FTP events

If you set the bypass_outgoing_TCPIP configuration setting to 21 (the FTP port), the audit and trace files no longer log the network event when you FTP *from* that workstation. Events related to other services, such as ssh, login, and Telnet, and subsequent events that use the network connection (for example, opening a file), will still be logged.

Chapter 3: Substituting Users Safely with eTrust AC

This section contains the following new topics for the Release Summary:

This section contains the following replacement topics for the Administrator Guide:

This section contains the following replacement topics for the Utilities Guide:

sesu Enhancements to Support Native Linux Options

eTrust AC lets you use native options with the sesu utility on Linux operating systems with version RHEL 4 (AS & ES), SLES 9, or SLES 10. The supported options include native su *-l*, *-s*, and the cross-UNIX - and *-c* options.

The functionality of these sesu options is identical to the native su command functionality.

Safe User Substitution using eTrust AC

The UNIX su command lets a user switch to another user without knowing the target user's password. It does not record who invoked the command so a user pretending to be the owner of an account is indistinguishable from the actual owner.

eTrust AC includes the sesu utility, which, is an enhanced version of the UNIX su command. You can configure sesu to prompt the user for their own password as a means of authentication, rather than prompting for the target user's password. The authorization process is based on the access rules defined in the SURROGATE class and, optionally, on the password of the user executing the command.

Unlike permission to su, permission to sesu does not depend on knowing the target user's password. Instead, it depends on permissions specified in the database; users remain accountable for their actions because their login identities are remembered.

If a user is a surrogate to one of the users in the _surrogate group, eTrust AC sends a full trace of the user's actions as the new user to the audit trail.

To protect against inadvertent use, sesu is marked in the file system so that no one can run it. The security administrator must mark the program as executable and setuid to root before you can use it.

Important! Before you use the sesu utility, define all users to the eTrust AC database and set sesu prerequisites. This prevents you from opening up the entire system to users who are not defined to eTrust AC.

How to Set Up sesu

By default, the sesu utility is marked in the file system so that no one can run it. Before you make sesu available to your users, you must set database rules to ensure it is used safely. You then need to lock the system's su utility so that users are forced to use the eTrust AC sesu utility instead.

To set up sesu, do the following:

1. Set basic user substitution rules.
2. Replace the system's su utility with the eTrust AC sesu utility.
3. Prevent users from running the system's su utility.

Note: After you complete this setup, when eTrust AC is running the system's su utility will not execute and users will be forced to use the secured sesu utility. When eTrust AC is not running, the system's su utility will work.

Set Basic User Substitution Rules

Before you start using the sesu utility, you should set up some common user substitution rules in the database. These rules prevent unknown users undesirably substituting privileged user accounts, but permit specific users and processes to perform necessary user substitution activities.

To set basic user substitution rules

1. Open a selang window.

   Note: The following instructions use selang. You can use the user interface to perform the same actions.

2. Prevent all users from substituting *root*, unless explicitly authorized, using the following command:

   ```
   nr surrogate USER.root defacc(n) own(nobody)
   ```

3. Prevent all users from substituting *root*'s group, unless explicitly authorized, using the following command:

   ```
   nr surrogate GROUP.other defacc(n) own(nobody)
   ```

   Note: On most UNIX systems root's group is either *other* or *sys*.

4. Authorize all administrators to substitute *root*, using the following command:

   ```
   auth surrogate USER.root gid(sys_admin_GID) acc(a)
   ```

   Note: By using the administrators' group *sys_admin_GID* you are authorizing all administrators. You can authorize individual administrators by using the uid option of the command.

5. Authorize all administrators to substitute root's group, using the following command:

   ```
   auth surrogate GROUP.other gid(sys_admin_GID) acc(a)
   ```

6. Prevent all users from substituting any user, unless explicitly authorized, using the following command:

   ```
   cr surrogate USER._default defacc(n) own(nobody)
   ```

7. Prevent all users from substituting any group, unless explicitly authorized, using the following command:

   ```
   cr surrogate GROUP._default defacc(n) own(nobody)
   ```

8. Authorize root to substitute any user, unless explicitly denied, using the following command:

   ```
   auth surrogate USER._default uid(root) acc(a)
   ```

   Note: You need to specifically authorize root to permit programs such as dtlogin to switch session ownership from root, the default X window owner (uid=0), to anyone else. If you do not do this, login attempts will fail because eTrust AC is blocking any user substitution activity that has not been explicitly authorized.

9. Authorize root to substitute any group, unless explicitly denied, using the following command:

   ```
   auth surrogate GROUP._default uid(root) acc(a)
   ```

10. Authorize the administrators' group to substitute to any user, unless explicitly denied, using the following command:

    ```
    auth surrogate USER._default gid(sys_admin_GID) acc(a)
    ```

11. Authorize the administrators' group to substitute any group, unless explicitly denied, using the following command:

    ```
    auth surrogate GROUP._default gid(sys_admin_GID) acc(a)
    ```

Replace the System's su Utility with the eTrust AC sesu Utility

By default, the sesu utility is marked in the file system so that no one can run it. To let users substitute other users by using the sesu utility, you must enable sesu and replace the system su with this utility.

To replace the system's su utility with the eTrust AC sesu utility

Note: You need to be root or another authorized user to perform the following steps.

1. Permit users to run the sesu utility using the following command:

   ```
   chmod +s /opt/CA/eTrustAccessControl/bin/sesu
   ```

2. Find out the location of the system's su utility using the following command:

   ```
   which su
   ```

3. Rename the system's su utility using the following command:

   ```
   mv su_dir/su su_dir/su.ORIG
   ```

   where *su_dir* is the directory where su resides.

4. Link the sesu utility to the su command:

   ```
   ln -s /opt/CA/eTrustAccessControl/bin/sesu su_dir/su
   ```

   This lets users continue to use the su command, although it now runs the sesu utility.

5. Stop eTrust AC using the following command:

   ```
   secons -s
   ```

6. Modify eTrust AC configuration settings using the following commands:

   ```
   seini -s sesu.SystemSu su_dir/su.ORIG
   seini -s sesu.UseInvokerPassword yes
   ```

   The token SystemSu is set so that sesu can call the original system su utility if eTrust AC is not running.

   The token UseInvokerPassword is set to tell eTrust AC to prompt the user for their original password instead of root's password or another user's password. The user needs to re-authenticate before the user substitution is permitted.

7. Reload eTrust AC using the following command:

   ```
   seload
   ```

Prevent Users from Running the System's su Utility

Although the sesu utility is configured, anyone can run su.ORIG (the renamed system su utility), as before, with root's or a user's password. To prevent this, use the PROGRAM class to explicitly prevent su.ORIG execution when eTrust AC is running.

Note: If you used seuidpgm during eTrust AC installation and configuration, you do not need to follow this procedure. su will not run as it has been modified (renamed to su.ORIG).

To prevent users from running the system's su utility

1. In selang, set eTrust AC to monitor the renamed su utility, using the following command:

   ```
   nr program su_dir/su.ORIG defacc(x) own(nobody)
   ```

2. Logged in as root, change file access and modification time, using the following command:

   ```
   touch su_dir/su.ORIG
   ```

   eTrust AC is watching su.ORIG and, because the file has been *touched*, will prevent it from being executed.

sesu Utility—Substitute User

Use the sesu utility to temporarily act as another user. This utility is the eTrust AC version of the UNIX su command. However, the sesu utility provides a user substitution command that does not require you to provide the password of the substituted user. The authorization process is based on the eTrust AC access rules as defined in class SURROGATE and, optionally, on the password of the user executing the command.

The sesu utility uses the tokens in the sesu section of the seos.ini file. It also uses the following special files:

- /etc/passwd
- /etc/group
- /etc/shells

To protect against inadvertent use, sesu is marked in the file system so that no one can run it. The security administrator must mark the program as executable and setuid to root before you can use it.

Important! Before you use the sesu utility, define all users to the eTrust AC database and set sesu prerequisites. This prevents you from opening up the entire system to users who are not defined to eTrust AC.

Usage notes:

- If the eTrust AC authorization server is not found, the utility executes the system's standard su command.
- If the sesu.old_sesu configuration token is set to no, the utility executes the system's standard su command.
- If /etc/shells exists, and it does not specify the current shell, sesu does not permit substitution to root.

This utility has the following syntax:

```
sesu [-] [username] [-l] [-s shell] [-c command]
```

-

Sets the environment to that of the target user.

Note: On Linux, this is the same as using the *-l* option.

-c *command*

Executes the specified command then exits.

Enclose commands containing spaces in quotes.

-h

Displays the help for this utility.

-l

(Linux only). Specifies that the shell it opens is a login shell.

-s *shell*

(Linux only). Specifies a shell to open instead of the shell from the user's passwd entry.

The shell must be listed in the /etc/shells file.

username

Changes the ID associated with the session to the ID of the specified target user *username*.

If you do not specify a *username*, sesu default to root.

Examples

- The following command changes the UID to root. The environment remains that of the user who executed the command.

  ```
  sesu
  ```

- The following command changes the UID to root*.* The utility changes the environment to root's environment.

  ```
  sesu -
  ```

- The following command surrogates to the user John.

  ```
  sesu John
  ```

- The following command surrogates to the user Carol and executes the specified command, ls -la, from the /home/carol directory.

  ```
  sesu - Carol -c "ls -la /home/carol"
  ```

- The following command surrogates to the user Angelo, uses a bash shell and opens it as a login shell.

  ```
  sesu Angelo -l -s /bin/bash
  ```

  Note: This is valid on Linux only.

Chapter 4: Using Native Packaging

This section contains the following new topics for the Release Summary:

This section contains the following new topics for the Implementation Guide:

Native Installation Support

eTrust AC offers native package formats for installing and managing eTrust AC natively on supported operating systems. Native packages let you manage your eTrust AC installation using native package management tools. eTrust AC now supports the following new native installation formats:

- Software Distributor-UX (SD-UX) packages, for installation on HP-UX.
- installp format packages (bff files), for installation on AIX.

These native package formats are in addition to the existing RPM and Solaris package formats.

eTrust AC Native Packages

eTrust AC includes native packages for each supported native installation format. These packages let you manage eTrust AC components. The following are the packages and their descriptions:

CAeAC

Installs the core eTrust AC components. This is the main eTrust AC installation package and combines the server, client, documentation, TNG integration, API, and mfsd packages which are traditionally packaged separately.

CAeACgui

Adds the eTrust AC administration GUI component.

Package Customization

If you want to install eTrust AC with custom settings using native packaging, you need to customize the package before you install it. eTrust AC provides a customization script you can use for each native package format it supports.

Note: To customize any of the eTrust AC native packages, follow the steps in the procedure for your native package format. We recommend that you do not modify the packages manually; instead, use the script as described.

HP-UX Native Package Installation

HP-UX native packaging is provided as a set of GUI and command-line utilities that let you create, install, remove, and report on individual software packages.

Note: For more information about the HP-UX native packaging, Software Distributor-UX (SD-UX), see the HP website at http://www.hp.com. You can also refer to the man pages for swreg, swinstall, swpackage, and swverify.

Instead of a regular installation, you can use the SD-UX native packages eTrust AC provides. This lets you manage your eTrust AC installation with all your other software installations performed using the SD-UX.

Important! To uninstall eTrust AC after a package installation, you must use the *swremove* command. Do not use the uninstall_eTrustAC script.

Install eTrust AC HP-UX Native Packages

The eTrust AC Software Distributor-UX (SD-UX) native packages let you install eTrust AC on HP-UX easily.

Note: The following procedure installs eTrust AC with the default settings. Alternatively, you may want to customize the eTrust AC package (see page 22) before installing it.

To install the eTrust AC HP-UX native packages using the command line interface

1. Log in as root.

   To register and install HP-UX native packages you need permissions associated with the root account.

2. Register the package with SD-UX using the following command:

   ```
   swreg -l depot pkg_location
   ```

   where *pkg_location* is the directory where the eTrust AC package (CAeAC) is located.

3. Install the eTrust AC package using the following command:

   ```
   swinstall -s pkg_location CAeAC
   ```

   SD-UX starts installing the CAeAC package from the *pkg_location* directory.

4. (Optional) Register and Install the GUI package:

   ```
   swreg -l depot pkg_location
   swinstall -s pkg_location CAeACgui
   ```

   eTrust AC is now fully installed but not started.

Customize the eTrust AC SD-UX Format Packages

If you want to install eTrust AC with custom settings using Software Distributor-UX (SD-UX) format packaging, you need to customize the package before you install it. You customize the script by extracting the installation parameters file from the package, modifying it as required, and then loading it back into the package.

To customize the eTrust AC SD-UX format packages

1. Extract the package you want to customize to a temporary location on your file system.

   Copy the package's directory and its entire content to a read/write location on the file system, where the package can be customized as required.

2. (Optional) Copy the customize_eac_depot script file and the pre.tar file to a temporary location on your file system.

   You only need to have pre.tar file in the same directory as the script file if you want script messages in a language other than English. The pre.tar file is compressed tar file containing installation messages and the eTrust AC license agreement.

3. (Optional) Enter the following command to set the language of the installation parameters file:

   ```
   customize_eac_depot -r -l lang -d pkg_location pkg_name
   ```

   where *lang* is the language you want for the installation parameters file, *pkg_location* is the directory where you placed your package on the file system, and *pkg_name* is the name of the package.

   Note: For a list of supported languages you can specify, run *customize_eac_depot -h*. By default, the installation parameters file is in English.

4. (Optional) Enter the following command to change the installation directory:

   ```
   customize_eac_depot -i install_loc [-d pkg_location] [pkg_name]
   ```

   where *install_loc* is the location where you want eTrust AC installed.

5. Enter the following command to get the installation parameters file:

   ```
   customize_eac_depot -g -f tmp_params -d pkg_location pkg_name
   ```

   where *tmp_params* is the full path and name for the extracted installation parameters file.

6. Edit the installation parameters file to suit your installation requirements.

   This file lets you set the installation defaults for the package. For example, activate the POSTEXIT token (remove the preceding # character) and point it to post-installation script file you want to run.

7. Enter the following command to set the installation parameters in your customized package:

   ```
   customize_eac_depot -s -f tmp_params -d pkg_location pkg_name
   ```

   You can now use the package to install eTrust AC with the customized defaults.

customize_eac_depot Command—Customize an SD-UX Format Package

The customize_eac_depot command runs the eTrust AC native package customization script for SD-UX format packages.

Important! To customize a package, the package must be in a read/write directory on your file system. When you copy the package, you must make sure that file attributes for the entire directory structure of the package are preserved or the packaging tools will consider the package corrupt.

Note: For localized script messages, you need to have pre.tar file in the same directory as the script file.

```
customize_eac_depot -r [-l lang] [-d pkg_location] [pkg_name]
customize_eac_depot -i install_loc [-d pkg_location] [pkg_name]
customize_eac_depot -s -f tmp_params [-d pkg_location] [pkg_name]
customize_eac_depot -g [-f tmp_params] [-d pkg_location] [pkg_name]
```

pkg_name

(Optional) The name of the eTrust AC package you want to customize. If you do not specify a package, the script defaults to the main eTrust AC package (CAeAC).

-d *pkg_location*

(Optional) Specifies the directory where you placed your package on the file system. If you do not specify a directory where the package is located, the script defaults to /var/spool/pkg.

-f *tmp_params*

Specifies the full path and name of the installation parameters file to create or retrieve information from.

If you use the -g option but do not specify the -f option, the installation parameters are directed to the standard output (stdout).

-g

Gets the installation parameters file and places it in the file specified by the -f option.

-h

Displays command usage and what languages the installation parameters file is available in.

Note: You can find the list of languages eTrust AC itself supports in the comment for the LANGUAGE token in the installation parameters file. Set the installation language using this token.

-i *install_loc*

Sets the installation directory for the package to *install_loc*.

-l *lang*

Sets the language of the installation parameters file to *lang*. You can only specify the -l option when using the -r option.

Note: For a list of supported languages you can specify, run *customize_eac_depot -h*. By default, the installation parameters file is in English.

-r

Resets the package to use default values as set in the original package.

-s

Sets the specified package to use inputs from the customized installation parameters file specified by the -f option.

Uninstall eTrust AC HP-UX Packages

To uninstall an eTrust AC HP-UX package installation, you need to uninstall the eTrust AC packages in the reverse order of their installation.

To uninstall eTrust AC packages

1. Uninstall the last eTrust AC HP-UX package you installed.

   For example, if you installed the GUI package, uninstall this first.

   ```
   swremove CAeACgui
   ```

2. Uninstall the main eTrust AC package.

   ```
   swremove CAeAC
   ```

AIX Native Package Installation

AIX native packaging is provided as a set of GUI and command-line utilities that let you manage individual software packages.

Instead of a regular installation, you can use the AIX native packages eTrust AC provides. This lets you manage your eTrust AC installation with all your other software installations performed using the AIX installp.

Note: While some AIX versions support several package formats (installp, SysV, RPM), eTrust AC only provide the AIX native package format, installp.

Important! To uninstall eTrust AC after a package installation, you must use the *installp* command. Do not use the uninstall_eTrustAC script.

Install eTrust AC AIX Native Packages

The eTrust AC AIX native packages let you install eTrust AC on AIX easily.

Note: The following procedure installs eTrust AC with the default settings. Alternatively, you may want to customize the eTrust AC package (see page 27) before installing it.

To install the eTrust AC AIX native packages using the command line interface

1. Log in as root.

   To register and install AIX native packages, you need permissions associated with the root account.

2. (Optional) Record the level (version) of the package that you want to install:

   ```
   installp -l -d pkg_location
   ```

   where *pkg_location* is the directory where the eTrust AC package (CAeAC) is located.

   For each package in *pkg_location*, AIX lists the level of the package.

   Note: For more information about the AIX native packaging installation options, refer to the man pages for installp.

3. Install the eTrust AC package using the following command:

   ```
   installp -ac -d pkg_location CAeAC [pkg_level]
   ```

   where *pkg_level* is the level number of the package you recorded earlier.

   AIX starts installing the CAeAC package from the *pkg_location* directory.

4. (Optional) Install the GUI package:

   ```
   installp -ac -d pkg_location CAeACgui [pkg_level]
   ```

   eTrust AC is now fully installed but not started.

Customize the eTrust AC bff Native Package Files

If you want to install eTrust AC with custom settings using installp format native packaging (bff files), you need to customize the package before you install it. You customize the script by extracting the installation parameters file from the package, modifying it as required, and then loading it back into the package.

Note: To build a custom eTrust AC AIX native installation package, you must have bos.adt.insttools installed on your computer.

To customize the eTrust AC bff files

1. Copy the package you want to customize (a bff file) to a temporary location on your file system.

   In the read/write location on the file system, you can customize the package as required.

   Important! This location needs to have space that is at least twice the size of the package, so that it can hold temporary repackaging files.

2. (Optional) Copy the customize_eac_bff script file and the pre.tar file to a temporary location on your file system.

   You only need to have pre.tar file in the same directory as the script file if you want script messages in a language other than English. The pre.tar file is compressed tar file containing installation messages and the eTrust AC license agreement.

3. (Optional) Enter the following command to set the language of the installation parameters file:

   ```
   customize_eac_bff -r -l lang -d pkg_location pkg_name
   ```

   where *lang* is the language you want for the installation parameters file, *pkg_location* is the directory where you placed your package, and *pkg_name* is the name of the bff package file.

   Note: For a list of supported languages you can specify, run *customize_eac_pkg -h*. By default, the installation parameters file is in English.

4. (Optional) Enter the following command to change the installation directory:

   ```
   customize_eac_bff -i install_loc [-d pkg_location] [pkg_name]
   ```

   where *install_loc* is the location where you want eTrust AC installed.

5. Enter the following command to get the installation parameters file:

   ```
   customize_eac_bff -g -f tmp_params -d pkg_location pkg_name
   ```

   where *tmp_params* is the full path and name for the extracted installation parameters file.

6. Edit the installation parameters file to suit your installation requirements.

   This file lets you set the installation defaults for the package. For example, activate the POSTEXIT token (remove the preceding # character) and point it to post-installation script file you want to run.

7. Enter the following command to set the installation parameters in your customized package:

   ```
   customize_eac_bff -s -f tmp_params -d pkg_location pkg_name
   ```

   You can now use the package to install eTrust AC with the customized defaults.

customize_eac_bff Command—Customize a bff Native Package File

The customize_eac_bff command runs the eTrust AC native package customization script for bff native package files.

The script works on any of the available eTrust AC native packages for AIX. To customize a package, the package must be in a read/write directory on your file system.

Important! This location should have space to contain at least twice the size of the package for intermediate repackaging results.

Note: For localized script messages, you need to have pre.tar file in the same directory as the script file.

```
customize_eac_bff -r [-d pkg_location] [-l lang] pkg_name
customize_eac_bff -i install_loc [-d pkg_location] pkg_name
customize_eac_bff -s -f tmp_params [-d pkg_location] pkg_name
customize_eac_bff -g [-f tmp_params] [-d pkg_location] pkg_name
```

pkg_name

The name of the eTrust AC package (bff file) you want to customize.

-d *pkg_location*

(Optional) Specifies the directory where you placed your package on the file system. If you do not specify a directory where the package is located, the script defaults to /var/spool/pkg.

-f *tmp_params*

(Optional) Specifies the full path and name of the installation parameters file to create or retrieve information from.

If you do not specify a file when using the -g option, the installation parameters are directed to the standard output (stdout).

-g

Gets the installation parameters file and places it in the file specified by the -f option.

-h

Displays command usage and what languages the installation parameters file is available in.

Note: You can find the list of languages eTrust AC itself supports in the comment for the LANGUAGE token in the installation parameters file. Set the installation language using this token.

-i *install_loc*

Sets the installation directory for the package to *install_loc*.

-l *lang*

Sets the language of the installation parameters file to *lang*. You can only specify the -l option when using the -r option.

Note: For a list of supported languages you can specify, run *customize_eac_bff -h*. By default, the installation parameters file is in English.

-r

Resets the package to use default values as in the original package.

-s

Sets the specified package to use inputs from the customized installation parameters file specified by the -f option.

Uninstall eTrust AC AIX Packages

To uninstall an eTrust AC AIX package installation, you need to uninstall the eTrust AC packages in the reverse order of their installation.

To uninstall eTrust AC packages

1. Uninstall the last eTrust AC AIX package you installed.

   For example, if you installed the GUI package, uninstall this first.

   ```
   installp -u CAeACgui
   ```

2. Uninstall the main eTrust AC package.

   ```
   installp -u CAeAC
   ```

Chapter 5: Auditing Access Control Activity for Windows

This section contains the following new topics for the Release Summary:

This section contains the following new topics for the Administrator Guide:

This section contains the following replacement topics for the Administrator Guide:

Auditing Improvements for Windows

eTrust AC on Windows offers improved auditing capabilities, efficiency, and transparency:

Full auditing

Full auditing provides audit records for 100% of the following intercepted events:

- File access (FILE class).
- Program execution (PROGRAM class).
- Registry access (REGKEY class).
- Impersonation control (SURROGATE class).
- Network control (CONNECT, TCP, HOST, GHOST, HOSTNET, and HOSTNP classes).
- Log in (TERMINAL class)
- Process termination (PROCESS class)

Full auditing is enabled by default when you upgrade eTrust AC.

Important! Depending on the rules you have in the database, the number of audit events that eTrust AC records to the log file could significantly increase as a result of this feature. We recommend that you review your audit log file size and backup settings.

Audit Only mode

Audit Only mode lets you audit all intercepted events without checking for authorization.

You can use this mode to monitor users' access without checking access rules. eTrust AC does not process the request for an authorization result which means quicker processing times.

Improved auditing performance

Improved auditing performance reduces processing times by using cached audit events.

eTrust AC uses these audit events to write audit records without checking each event for authorization rules.

Additional statistics and data

Additional statistics and data provide useful information for analyzing and troubleshooting eTrust AC behavior.

You can use the secons utility to view information about the eTrust AC engine and the auditing interface.

Transparent SID to account name resolution

An eTrust AC registry entry controls the amount of time eTrust AC spends trying to resolve an SID (security identifier) into an account name.

Records are added to the Event Viewer if eTrust AC could not resolve an SID. These specific messages help you trace the problem. If necessary, you can adjust the SID-to-account resolving timeout.

eTrust AC Registry Settings

eTrust AC registry entries control eTrust AC behavior and functionality. eTrust AC creates its registry entries under the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAccessControl

New Registry Entries for Auditing

New eTrust AC registry entries let you control the improved auditing functionality, while some old entries are obsolete. This list contains all the new and obsolete entries:

SeOSD Key

AuditCollectorInterfaceName

Defines the pipe name which functions as an audit interface between the audit collector component (within seosd) and the different clients of the audit collector (kernel).

Default: AuditCollector

AuditServerCacheSize

Defines the size of the audit cache, in number of entries.

Default: 1024

DefLookupThreads

Defines the number of threads eTrust AC can use to resolve SIDs into account names.

Default: 5

DefLookupTimeout

Defines the timeout, in milliseconds, before eTrust AC stops trying to resolve an SID into an account name.

Default: 2000

GeneralInterceptionMode

Specifies whether to use Full Enforcement mode (0) or Audit Only mode (1).

Default: 0

FsiDrv Key

BatchOplockStatus

Specifies whether to disable batch oplocks (opportunistic locks of an entire file). When disabled, the driver collects 100% of audit information for file access but performance decreases. A non-zero value keeps batch oplocks operating regularly (enabled), potentially limiting file access audit records.

Note: You must reload the driver to affect a new setting. Unload the driver (net stop seosdrv) after you stop eTrust AC (secons -s).

Default: 0

FileCacheRefreshPeriod

Obsolete.

MaxAuditRecordLimit

Defines the audit queue limit. When the queue length exceeds this limit, eTrust AC artificially slows down threads that generate audit events so that it can read the queue and write to the log file faster than new items are added to the queue.

Note: When new items are added to the queue faster eTrust AC can read process them, the system's memory may be exhausted.

Default: 200

MaxTimeoutLimit

Defines the number of consecutive timeouts that eTrust AC detects before it triggers a driver bypass. Once reached, the driver stops sending authorization requests to the authorization engine until the engine indicates that it is ready to process events.

A value of zero disables this bypass.

Default: 5

RegistryCacheRefreshPeriod

Obsolete.

New Registry Key for the Event Log

eTrust AC creates a new registry key for registering seosdrv.sys driver as an event source for the event log:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\SeosDrv
```

To control auditing functionality, eTrust AC creates the following new registry entries under this key:

EventMessageFile

Defines the pathname to the seosdrv.sys driver.

Default: %SystemRoot%\System32\drivers\seosdrv.sys

TypesSupported

A standard Windows entry used to define the bitmask of supported event types.

Default: 7

Events Interception

eTrust AC intercepts an event if:

- The appropriate class is active
- A rule anticipating this event exists in the database

For example, you can use the following generic rule to audit all file accesses to files that reside in c:\data\payroll:

```
newres FILE c:\data\payroll\*
```

You also need to make sure that the FILE class is active (the default).

Types of Intercepted Events

eTrust AC intercepts two types of events:

Interception Events

An *interception event* is an event that has been encountered for the first time and for which the kernel cache has no authorization or audit information. Information from an interception event is cached as part of the process for future use by an audit event.

Audit Events

An *audit event* is an event for which the kernel cache has enough information to process for auditing purposes; it is also known as a *cached intercepted event*. An audit event is the result of an interception event being cached.

Interception Modes

Based on the interception mode, eTrust AC intercepts, checks for authorization, and logs audit records of access request events. eTrust AC has four modes of interception:

- Full Enforcement mode

  Full Enforcement mode is the normal operation mode for eTrust AC. In this mode, eTrust AC intercepts events and enforces the access rules written to the database.

- Audit Only mode

  Audit Only mode records *all intercepted* events without checking or enforcing access rules.

- No Interception mode

  No Interception mode disables eTrust AC event interception. In this mode, eTrust AC does not intercept events or enforce access rules.

Note: Warning mode is not an interception mode; it works in Full Enforcement mode only and is designed for short term use during implementation. For more information on Warning mode, see the *Administrator Guide*.

Audit Only Mode

Audit Only mode records *all intercepted* events without checking or enforcing access rules. Use this mode to collect data for compliance requirements or regulations. In Audit Only mode, eTrust AC intercepts the events and writes an audit event but does not process the request for an authorization result and does not enforce rules. As a result, eTrust AC permits all access requests it intercepts. This means that the authorization result recorded in the audit log for all events is *P* (permitted).

The following restrictions apply to Audit Only mode:

- No audit records are sent to Unicenter.

  In Audit Only mode all events are permitted (*P*). Permitted events are not sent to Unicenter.
- The audit properties of the resource and the user are *not* taken into consideration.

  Audit Only mode records *all intercepted* events regardless of resource- or user-specific settings.
- The audit filter does not get used.

  Audit Only mode bypasses the authorization process, which is when the filter file is read.

Set Up Audit Only Mode

Audit Only mode records *all intercepted* events without checking or enforcing access rules. Use this mode to collect data for compliance requirements or regulations.

To set up audit only mode, set the SeOSD\GeneralInterceptionMode eTrust AC registry entry to 1.

Important! If you use Audit Only mode, make sure you have enough disk space for the audit logs and that the size limit of the audit log is large enough. You should also consider options for audit log backup (see page 43).

What eTrust AC Audits

For security auditing, eTrust AC keeps audit records for intercepted events according to the audit rules defined in the database and the enforcement mode it operates in. The records in the audit log accumulate according to these audit rules.

Full auditing provides audit records for all intercepted events of any of the following:

- File access (FILE class).
- Program execution (PROGRAM class).
- Registry access (REGKEY class).
- Impersonation control (SURROGATE class).
- Network control (CONNECT, TCP, HOST, GHOST, HOSTNET, and HOSTNP classes).
- Log in (TERMINAL class)

  Note: Intercepted login events are not cached; they always follow the auditing process for interception events.
- Process termination (PROCESS class)

The decision whether to log an event depends on the eTrust AC interception mode.

What eTrust AC Audits in Full Enforcement Mode

In Full Enforcement mode (regular operation), eTrust AC logs events as follows:

- If Warning mode is turned *off* for the intercepted resource, eTrust AC enforces rules and logs the events based on the *audit* property of the resource or user.

Audit Property	**Events Logged**
ALL	*All*
SUCCESS	Access permitted
FAIL	Access denied

- If Warning mode is turned *on* for the intercepted resource, a record is written to the audit log if an access request violates an access rule (if the rules were enforced, the request would have failed). The audit record mentions that the violation was permitted because Warning mode is in effect.

  Rules are not enforced in this mode.

What eTrust AC Audits in Audit Only Mode

In Audit Only mode, eTrust AC does not process requests for authorization or enforce rules. *All intercepted* login events for the accessor and *all intercepted* events for resources protected by eTrust AC are logged, regardless of whether access failed or succeeded.

The Auditing Process

To configure eTrust AC for your auditing requirements, you must first understand how auditing works. Auditing lets you keep track of access requests (events) that eTrust AC intercepted. You can use this data to meet with compliance requirements, to analyze and refine your access rules for your security requirements, or to monitor access requests.

The process eTrust AC follows to record audit events in the log depends on the type of event it intercepts:

- Interception events

  Note: Intercepted login events (TERMINAL class) are not cached; they always follow the auditing process for interception events.

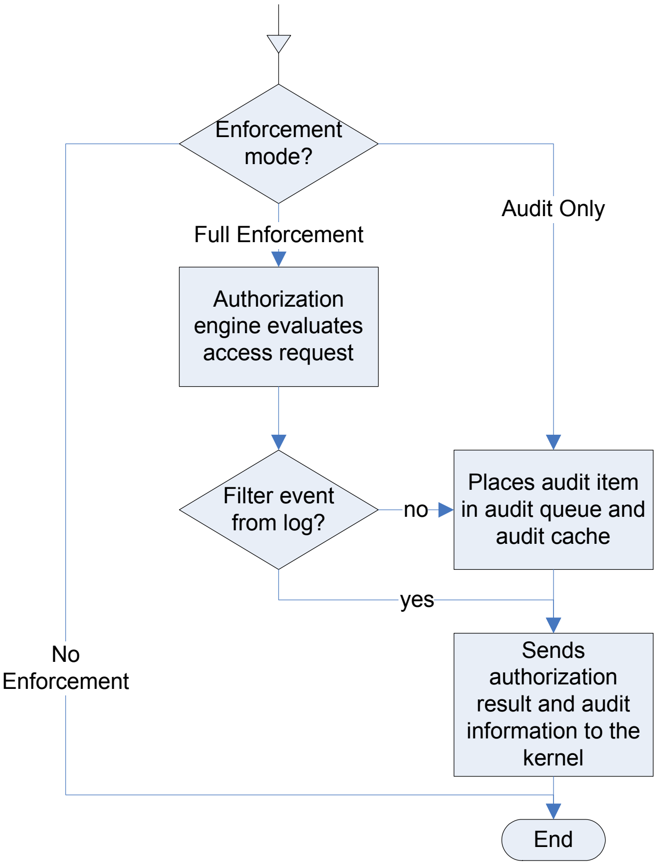
- Audit events

Note: eTrust AC intercepts an event only if the appropriate class is active, and the database contains a rule anticipating this event.

How Auditing Works for Interception Events

An *interception event* is an event that has been encountered for the first time and for which the kernel cache has no authorization or audit information.

To log audit records, eTrust AC performs the following actions and causes these effects for an interception event:


Enforcement mode?
Full Enforcement
Audit Only
No Enforcement
Authorization engine evaluates access request
Filter event from log?
no
yes
Places audit item in audit queue and audit cache
Sends authorization result and audit information to the kernel
End

- In No Enforcement mode, events are not intercepted or audited.
- In Full Enforcement mode, eTrust AC does the following:

  1. The authorization engine places an audit item based on the authorization result in the audit queue and in the audit cache.

     eTrust AC writes an audit item only if the audit property for the resource or accessor is set to audit the resulting event and the audit filter file is not set to filter this event.

  2. The authorization engine returns an informative answer on the authorization result and the audit related information to the kernel.

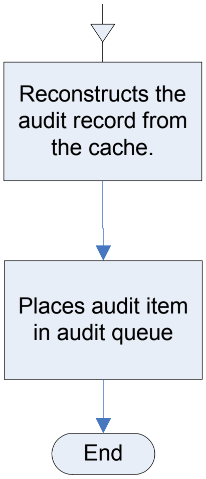
- In Audit Only mode, eTrust AC does not process the request for authorization. Audit information is always written, regardless of the audit property of the resource and user.

  The authorization result in this mode is always *P* (permitted).

Note: Intercepted login events (TERMINAL class) are not cached; the authorization engine always writes audit records for these events.

How Auditing Works for Audit Events

An *audit event* is an event for which the kernel cache has enough information to process for auditing purposes; it is also known as a *cached intercepted event*. An audit event is the result of an interception event being cached.


Reconstructs the audit record from the cache.
Places audit item in audit queue
End

Once the kernel notifies eTrust AC about the cached interception event, eTrust AC performs the following actions to log the audit event:

1. Reconstructs the audit data using the audit cache out of the information sent by the kernel.
2. Puts the audit item in the audit queue.

Kernel and Audit Caches

The kernel cache contains data about previously intercepted events. Such cached intercepted events (audit events) are identified by the kernel, which, then sends them to eTrust AC for processing. Essentially, eTrust AC uses the kernel cache to intercept events that follow the same pattern as a previously intercepted event.

The audit cache contains data that lets eTrust AC reconstruct reoccurring audit records and send them to the audit queue without needing to follow the authorization process. This means that intercepted events, for which enough information already exists in the cache (audit events), are processed quickly and added to the audit queue. The authorization engine provides the data that is stored in the kernel and audit caches from the result of the initial event it intercepted (the interception event).

Cache Reset

eTrust AC clears both the kernel and audit caches in the following cases:

- Database changes

  eTrust AC clears the entire cache when database information changes. New or modified access rules make an existing cache potentially inaccurate.

- Time checkpoint reached

  eTrust AC clears the entire cache when a time checkpoint affects an authorization result for any event. At the time that a DAYTIME restriction property or a HOLIDAY class record changes, the authorization result may change too and the cache becomes potentially inaccurate.

- PROGRAM resource change

  eTrust AC clears the entire cache when the watchdog identifies that a PROGRAM resource has changed and become un-trusted. An un-trusted program affects the result of an authorization request regarding that program. This makes the cache potentially inaccurate.

- Audit cache filling

  eTrust AC clears 10% of cache items (the least recently used items) when the audit cache fills up.

Once the cache is cleared, information from new interception events is needed to refill the cache and let eTrust AC intercept an audit event.

Audit Log Backup

eTrust AC lets you automatically backup the audit log file for archiving.

The name of the audit log backup file is set in the logmgr\audit_back eTrust AC registry entry.

You can use the following methods for backing up the audit log file:

- Size-triggered backups
- Date-triggered backups

The method and settings you choose for backing up your audit log file should depend on:

- Whether you need backup copies of the log file
- How much auditing data is likely to be generated in your environment
- System performance issues (for example, larger audit log files increase processing time)

Set the Size at which the Audit Log will be Backed Up Automatically

You can set a limit on the size of the audit log file. When the file reaches the defined size, eTrust AC automatically creates a backup copy of the file. This means that the file is automatically backed up regularly.

To set the size at which the audit log will be backed up automatically, set the maximum size you require, in KB, in the logmgr\audit_size eTrust AC registry entry.

Note: You can define the name of the backup file by setting the logmgr\audit_back eTrust AC registry entry.

Important! If the logmgr/BackUp_Date eTrust AC registry entry is set to yes (no is the default), each size-triggered backup copy of the audit log is suffixed with a timestamp. In all other cases, including when date-triggered backups are configured, each backup copy *overwrites* the previously written backup copy.

Example: Set automatic backup of audit log file when it reaches 5 MB

This example shows you how you set your audit log file to be backed up when it reaches 5 MB (5120 KB). To do this, set the logmgr\audit_size eTrust AC registry entry to **5120**.

When the audit log file reaches 5 MB, eTrust AC will create a backup copy of the file, named seos.audit.bak by default, and clear the log.

Example: Set automatic backup of audit log file when it reaches 1 MB with a custom name and a timestamp

This example shows you how you set your audit log file to be backed up when it reaches 1 MB (1024 KB), using a custom name for the backup file and adding a timestamp to the name.

To do this, set the following eTrust AC registry entries as shown:

- logmgr\audit_size=1024
- logmgr\audit_back=log\ac_audit.old
- logmgr\BackUp_Date=yes

When the audit log file reaches 1 MB, eTrust AC will create a backup copy of the file, and clear the log. The name of the backup log file name will be: ac_audit.old.*timestamp*, where *timestamp* is the date and time in the format DD-Mon-YYYY.hhmmss. For example:

```
ac_audit.old.06-Feb-2007.144330
```

Set the Time Interval at which the Audit Log will be Backed Up Automatically

You can define a time interval (daily, weekly, or monthly) at which eTrust AC automatically creates a backup copy of the audit log file.

To set the time interval at which the audit log is backed up automatically, set the interval in the logmgr\BackUp_Date eTrust AC registry entry. The interval can be one of the following:

daily

Backs up the audit log file once a day.

weekly

Backs up the audit log file once a week.

monthly

Backs up the audit log file once a month.

Note: You can define the name of the backup file by setting the logmgr\audit_back eTrust AC registry entry.

Important! If the audit log reaches the size limit defined in the logmgr\audit_size eTrust AC registry entry before the backup interval is reached, eTrust AC creates a backup copy of the file without a timestamp. Each such backup copy can potentially overwrite any previous copy.

Example: Set a daily backup of the audit log file

This example shows you how you set your audit log file to be backed up daily. To do this, set the logmgr\BackUp_Date eTrust AC registry to **daily**.

Once a day eTrust AC creates a backup copy of the file, and clear the log. The backup log file name has the *.timestamp*, suffix, where *timestamp* is the date and time in the format DD-Mon-YYYY.hhmmss. For example:

```
seos.audit.bak.06-Feb-2007.144330
```

eTrust AC Run-time Data (secons -i)

The secons utility displays eTrust AC run-time statistics and internal counters when you use the *-i* option. Use this statistical system behavior information to find out:

- How many events were triggered for each interception type.
- How effective each kernel cache is, by comparing the number of cached events against the number of fully authorized events.

Note: To reset the counters to zero, type secons -i -reset.

Following are descriptions of information that is not self-explanatory:

Database run-time data

Displays the number of classes, objects, and properties in the eTrust AC database, the ID of the last created class, object, and property, and the number of property values.

Use this information to evaluate the size of the database. The more objects and properties you use, the bigger the database is.

Kernel run-time data

Displays for each of the kernel caches (file, registry, and surrogate) their creation time, size, and efficiency. Efficiency is shown as the number of audit events out of the total number of events. The events that are not audit events are interception events that follow the authorization process.

Use this information to evaluate the need for, and the efficiency of, each kernel cache.

Kernel audit information

Displays the current kernel audit queue size, the maximum size it reached and the time at which it reached the maximum size.

Use this information to evaluate the audit queue behavior. You should make sure that the audit queue does not exceed the maximum allocated queue size, which is set in the FsiDrv\MaxAuditRecordLimit eTrust AC registry entry. When this limit is reached, eTrust AC generates audit events more slowly so that it can process the queue.

User mode enforcement run-time data

Displays information for intercepted file, registry, impersonation, and outbound network connection events in Full Enforcement mode. You can find out about the number of events being authorized by the authorization engine and the maximum and average time an authorization process took to complete for each class.

Use this information to troubleshoot problems in a live production system. It provides you with some valuable initial data without you needing to shut down eTrust AC.

User mode audit run-time data

Displays information for audit events (cached intercepted event*)*.

Use this information to monitor audit cache queue behavior. If the maximum audit queue consistently increases, make sure that eTrust AC can write to the audit log file. eTrust AC may not be able to write to the file if the system has run out of disk space, or it does not have native access permissions to file.

Note: It is fine for the audit queue to increase during periods of increased activity. However, the queue size should decrease once the load is normal again.

Audit Log Troubleshooting

This section lets you troubleshoot problems you may encounter with the audit log.

SID Resolution Failed (Event Viewer Warning)

Valid on Windows

Symptom:

When I view the Application log of the Windows Event Viewer, I find a Warning event from eTrust AC that says that resolving a specific SID into an account name has failed.

Solution:

A *security identifier (SID)* is a numeric value that identifies a user or group. Each entry in the system access control list (SACL) has an SID that identifies the user or group for whom access is allowed, denied, or audited.

This warning appears when the operating system was not able to convert the SID into an account name. Make sure that the problematic system and its corresponding domain controller are configured correctly for SID resolution.

SID Resolution Times Out (Event Viewer Warning)

Valid on Windows

Symptom:

When I view the Application log of the Windows Event Viewer, I find a Warning event from eTrust AC that says that resolving a specific SID into an account name has timed out.

Solution:

A *security identifier (SID)* is a numeric value that identifies a user or group. Each entry in the system access control list (SACL) has an SID that identifies the user or group for whom access is allowed, denied, or audited.

This warning appears when the operating system was not able to convert the SID into an account name within the defined timeout. Make sure that the:

- Problematic system and its corresponding domain controller are configured correctly for SID resolution.
- Network settings are configured correctly.

You can also increase the timeout by changing the SeOSD\DefLookupTimeout eTrust AC registry entry.

Note: Increasing the SID resolution timeout may downgrade eTrust AC performance.

Process Short Names Appear in the Audit Log

Valid on Windows

Symptom:

When I view the eTrust AC audit log, some records list process short names (8.3 format). For example, if a process named C:\Program File\MyVeryLongProcessName.exe tries to write to a protected file, this process may appear in the audit log as MYVERY~1.EXE.

Solution:

Such records appear when a cache reset occurs after an event is sent to the audit queue, but before it is handled by the audit collector. In this situation, the information the audit collector has cannot be synchronized and it has to reconstruct the audit event.

Run the secons utility with the -i option to evaluate and optimize audit cache performance. Also, minimize cache resets (see page 42) in your production environment.

Chapter 6: Bypassing Drivers on Windows

This section contains the following new topics for the Release Summary:

This section contains the following new topics for the Administrator Guide:

Driver Bypass on Windows

eTrust AC on Windows lets you bypass drivers for file access activity. You can specify drivers that can open other files without eTrust AC authorization checks. Bypassing these drivers prevents potential deadlocks between eTrust AC authorization checks and the drivers that need to access files as part of their routine operations, such as antivirus product drivers.

Note: A bypass for a current version of Trend Micro™ PC-cillin Antivirus is hard-coded into eTrust AC.

Important! This change means that the two registry entries, EnableTMBypass and TMDriverName, under the key HKLM\SYSTEM\CurrentControlSet\SeosDrv\Parameters, are obsolete.

The number of drivers you want to bypass is defined in the FsiDrv key of the eTrust AC registry in the BypassDriversCount entry value. For each driver, you need to create a registry entry named DriverName_*drvNumber* whose value is the name of the driver you want to bypass.

BypassDriversCount

Defines how many drivers you want to add to your bypass list.

Type: REG_DWORD

Default: 0

DriverName_*drvNumber*

Defines the name of a driver that you want to bypass (for example, thisdrv.sys).

drvNumber is a number from 0 through BypassDriversCount - 1. You need to create one registry entry for each driver you want to bypass and make sure that BypassDriversCount specifies the number of drivers you defined.

Type: REG_SZ

Limit: 49 characters.

Bypass Drivers

To specify that some drivers can operate without needing to submit operations for eTrust AC authorization checks, define a bypass for these drivers. For example, define a bypass for your antivirus program driver so that it can open files for scanning without eTrust AC authorization checks. Without the bypass, the driver might cause a deadlock with eTrust AC.

To bypass drivers

1. Set the BypassDriversCount registry entry value to the number of drivers you want to define a bypass for.

   You can find this entry in the FsiDrv key of the eTrust AC registry.

   Note: You must stop eTrust AC before you can change eTrust AC registry entries.

2. For each driver you want to bypass:

   a. Create a registry entry of type REG_SZ named DriverName_*drvNumber*

      The first entry should be DriverName_0 and the last DriverName_*X*, where *X* is BypassDriversCount - 1.

   b. Edit each DriverName_*drvNumber* entry so that its value is the name of the program driver you want to bypass.

      The value should be the name of the driver only (for example, thisdrv.sys).

3. Restart eTrust AC.

   eTrust AC reloads and bypasses the drivers you defined in the registry.

Example: Bypass Drivers to Resolve Compatibility Issues

This example resolves a compatibility issue an antivirus product has with eTrust AC by defining the antivirus drivers (avDriverA.sys and avDriverB.sys) for bypass. You set registry entries for driver bypass in the eTrust AC registry tree under the FsiDrv key:

```
HKLM\SOFTWARE\ComputerAssociates\eTrustAccessControl\FsiDrv
```

Set the registry entries as follows:

Name	Type	Data
BypassDriversCount	REG_DWORD	2
DriverName_0	REG_SZ	avDriverA.sys
DriverName_1	REG_SZ	avDriverB.sys

The BypassDriversCount registry entry value of 2 tells eTrust AC to look for two drivers to bypass. Each DriverName_*drvNumber* registry entry value defines a driver to bypass.

Chapter 7: Disabling Network Interception on Windows

This section contains the following new topics for the Release Summary:

Network Interception Changes on Windows (see page 53)

This section contains the following new topics for the Administrator Guide:

Disable Network Interception (see page 54)

Network Interception Changes on Windows

eTrust AC lets you disable network interception hooking so that it does not load at all at boot time. In previous versions of eTrust AC, network interception was always loaded at boot time and you had to disable network classes to disable actual interception of the specific network activity.

The network interception eTrust AC boot time hook state is defined in the *DisableNetworkInterception* entry under the key HKLM\SYSTEM\CurrentControlSet\Services\drveng\Parameters.

Note: You must restart the computer for the change to the registry entry to take effect.

DisableNetworkInterception

Specifies whether network interception hooking is disabled (relevant functions are not initialized at boot time).

Values: 1 (disabled)

Type: REG_DWORD

Note: If this registry entry does not exist (the default), or is set to any value other than 1, network interception is initialized at boot time.

Disable Network Interception

Even when the network classes are not enabled and you are not using those classes to intercept network activity, network interception functions load at boot time, affecting performance. To improve performance, you can disable network interception from loading at boot time.

To disable network interception

1. Create a registry entry named DisableNetworkInterception of type REG_DWORD and set its value to 1.

   This entry needs to exist in the following registry key:

   HKLM\SYSTEM\CurrentControlSet\Services\drveng\Parameters

2. Reboot the computer.

   eTrust AC reloads without initializing network interception.