eTrust® Audit and eTrust® SCC

Reference Guide

r8 SP2

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.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Product References

This document references the following CA products:

- eTrust® Security Command Center (eTrust SCC)
- eTrust® Audit (eTrust Audit)
- *e*Trust® Antivirus (eTrust AV)
- Unicenter® Event Management Console (Unicenter EVT)

Contents

Chapter 1: Introduction 13

Chapter 2: Windows Services 15

Chapter 3: UNIX Daemons 27

Chapter 4: Configuration Files 39

##

##

##

Chapter 9: Firewall Considerations 205

Chapter 10: Windows NT Security-related Event IDs 207

Chapter 11: Importing, Exporting, and Converting Policies 229

Chapter 12: Audit Event Taxonomy and Data 237

Chapter 13: Post Collection Utility 255

Chapter 14: iRouter: The Bridge between iRecorder and the Audit Router 291

Appendix B: Encup Utility 343

Appendix C: Digital Certificates 347

Appendix D: Disaster Recovery 351

Appendix E: Using the eTSAPISend Program 355

Appendix F: Status and Maintenance Utilities 361

Chapter 1: Introduction

This section contains the following topics:

Using This Guide

This guide presents a variety of topics that describe technical features of eTrust® Audit. While many users might never have reason to review the topics in this guide, others will need to consult it to perform additional configuration changes to their environments.

Chapter 2: Windows Services

This section contains the following topics:

Introduction

eTrust Audit installs several services on Windows systems. These services enable the information flow between Audit components by collecting, reading, and forwarding information from all sources in the system. This chapter describes the Audit services.

Windows Service Names

The next topics describe the following eTrust Audit services on Windows, and the commands to control them:

eTrust Audit Software Component	Windows Service Name
The eTrust Audit Action Manager	eTrust Audit Action Manager (acactmgr.exe)
The eTrust Audit Distribution Agent	eTrust Audit Distribution Agent (acdistagn.exe)
The eTrust Audit Distribution Server	eTrust Audit Distribution Server (acdistsrv.exe)
The eTrust Audit Log Router	eTrust Audit Log Router (aclogrd.exe)
The eTrust Audit Portmap service	eTrust Audit Portmap (portmap.exe)
The eTrust Audit Collector	eTrust Audit Collector (aclogrcd.exe)

The eTrust Audit Redirector	eTrust Audit Redirector (selogrd.exe)
iTechnology iGateway	iTechnology iGateway (iGateway.exe)

Commands to Control the Services

You can control the services using the Windows Control Panel or the Service Control Manager application. You can also control them from a command prompt. The eTrust Audit services for Windows reside in the following location:

```
install_dir\bin
```

where install_dir is the directory in which you installed eTrust Audit. Unless you add this directory to your PATH statement, you must issue the commands from this directory.

Command Syntax

The following command syntax applies to all eTrust Audit daemons:

programname options

where *programname* is the name of the daemon. The options are described in the topic that follows.

Command Parameters

The following list describes the available command parameters on Windows systems:

-help

Displays these syntax options.

-debug

Starts the service in foreground mode; that is, it routes debug messages to the console (STDOUT). For example, the following command starts the service and routes the output to the console:

```
service -debug
```

You can use the following options:

-trace options

Starts a trace of the service. For example, the following command starts the service and routes debug messages to the console and a file named errors.txt:

```
service -debug -trace -dest1 STDOUT -dest2 errors.txt
```

See the description of the -trace option later in the list.

-install

Installs the service. You can use the following options:

-user name

Lets you specify the name of a user authorized to install a service on the system. You should combine the -user and -pwd options as follows:

```
service -install -user user01 -pwd password
```

-pwd password

Lets you specify the password of a user authorized to install a service on the system. You should combine the -user and -pwd options as follows:

```
servicename -install -user user01 -pwd password
```

-trace options

Starts a trace of the service after installing it. For example, the following command installs the service and routes debug messages to the console:

```
service -install -trace -dest1 STDOUT
```

See the description of the -trace option later in the list.

-remove

Removes the service from the registry and from the Windows Service Control Manager.

You can use the following options:

-trace options

Starts a trace of the service while removing it. For example, the following command uninstalls the service and routes debug messages to a file named errors.txt:

```
service -remove -trace -dest1 errors.txt
```

Additionally, you can use the redirect symbol, >, as follows to open a console an direct the output to a file:

```
service -remove -trace -dest1 STDOUT > errors.txt
```

See the description of the -trace option later in the list.

-start

Starts the service in background mode; that is, without a console.

You can use the following options:

-trace options

Starts a trace of the service while starting it. For example, the following command starts the service and routes debug messages to a file named errors.txt:

```
service -start -trace -dest1 errors.txt
```

See the description of the -trace option later in the list.

-stop

Stops the service.

You can use the following options:

-trace options

Starts a trace of the service while stopping it. For example, the following command stops the service and routes debug messages to a file named errors.txt:

```
service -stop -trace -dest1 errors.txt
```

See the description of the -trace option later in the list.

The -trace option applies to all parameters, except -help, as follows:

-trace options

Turns on trace mode, which routes trace-level messages of a specified level to the destination. You can specify the following trace options:

-dbglvl n

Sets the debug level. n is the level from 1 to 5, 1 providing the least amount of debug information and 5 providing the most details. If you do not specify a value, 1 is the default.

-dest1 dest

Sets the primary output destination to display the debugging information to the console. dest can be one of the following:

- STDOUT - Routes messages to the console.
- STDERR - Routes messages to the console or to wherever you have redirected STDERR.
- filename - The name of file where you want the service to write the debug output.

-dest2 dest

Sets a secondary output destination to display the debugging information. dest can be one of the following:

- STDOUT - Routes messages to the console.
- STDERR - Routes messages to the console or to wherever you have redirected STDERR.
- filename - The name of file where you want the service to write the debug output.

Action Manager Service (acactmgr)

The eTrust Audit Action Manager service, acactmgr, reads events from queues where actions were placed by the Router and performs the specified actions defined for each event. The queues have parameters such as maximum action time, maximum file number and so on. These parameters affect the performance of the Action Manager.

For information about the Action Manager, actions, and configuration files, see About Available Actions (see page 42).

Registry Keys

Several registry keys contain values that affect the functioning of the eTrust Audit Action Manager service. They are as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\xxxQueue\Queue Parameters

For information on the values in these keys, see Windows Registry Entries (see page 93).

Distribution Agent Service (acdistagn)

The eTrust Audit Distribution Agent service, acdistagn, receives policy files or message parsing (MP) files from the Policy Manager through eTrust Audit Distribution Server service running on Windows or Solaris 10 systems. The Distribution Agent service also removes old policy or MP files if instructed by the Distribution Server service.

The distribution agent service changes auditing requirements according to the policy it receives. The service notifies the Router to update the policy to get new rules.

Registry Keys

When you install an eTrust Audit Client, you specify the name of the host where the Policy Manager runs. This is the only Policy Manager host recognized by the distribution agent service. It will reject attempts to update the policy from other hosts. However, you can add more servers to be recognized as trusted servers by editing the TrustedServers key of the Distribution Agent Service. This key is found in the following registry entry:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent.

Note: Configuring multiple Trusted Servers can cause policy to be overwritten unintentionally. Although the Distribution Agent Service can accept policies from multiple Policy Managers, the last policy that is distributed to it is applied regardless to the Policy manager from which it came.

Note for r8 SP2: In r8 SP2, the Distribution Agent service can send end-point validation status to the Policy Manager hosts (configured in the TrustedServers key). If this key contains a list of servers separated by commas, the Distribution Agent Service tries to send the end-point status to each of the servers in the list. End-point validation status contains the list of policies and MPs received and an indication of whether these are tampered or not.

The distribution agent service uses TCP/IP port 8025. You can change that port by using the registry and adding a special port, as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Ports\ DistributionPort

Note: The distribution port 8025 is registered with IANA and is now a well-known TCP port reserved for the Audit Distribution Agent service.

See Windows Registry Entries (see page 93) for more information.

Distribution Server Service (acdistsrv)

The eTrust Audit Distribution Server service, acdistsrv.exe, distributes the policy files among the Clients. It must run on the same system where the Policy Manager is located.

Registry Keys

After you instruct the Policy Manager to distribute the policy, the relevant commands reach the distribution queue. The distribution server reads the distribution queue, selects from the compiled policy files or MP files, processes them, and sends them to the distribution agents according to the commands.

The distribution server tries to connect to the distribution agent as follows:

- The distribution server opens as many threads as possible up to a configurable number, called MaxThreadNumber (default 10), and connects to as many distribution agents as possible.
- If the connection succeeds, the agent starts receiving configuration files. After the transmission operation terminates successfully, the distribution log of the Policy Manager is updated.
- If the connection attempt fails (or in case the initial connection succeeds but afterwards a failure occurs), the transmission command is delayed. In the mean time, the distribution server does not wait but connects directly to the next agent in the list. After a pre-defined period (the default value is 24 hours) of failed connection trials, the distribution server terminates the transmission attempts. In any case, the distribution log of the Policy Manager is updated.

The key of the Distribution Server is found under:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server

The distribution server service uses TCP/IP port 8026. You can change that port by using the registry and adding a special port, as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Ports\ DistributionControlPort

Note: The distribution control port 8026 is registered with IANA and is now a well-known TCP port reserved for Audit Distribution Server service.

See Windows Registry Entries (see page 93) for more information.

Log Router Service (aclogrd)

The eTrust Audit Log Router service, aclogrd, receives events from a number of different sources. It handles received events according to the filters specified in the router configuration file. It then routes them to the queue files with the associated actions and targets. The Router service should be registered by the eTrust Audit Portmap service so that it can start only if the portmap is running.

See Router Configuration File (see page 46) for more information.

Registry Keys

Several registry keys contain values that affect the functioning of the eTrust Audit Log Router service. They are as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\AllowRemoteProgram

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\AllowRemoteFile

For information on the values in these keys, see Windows Registry Entries (see page 93).

Collector Service (aclogrcd)

The eTrust Audit Collector service, aclogrcd, receives information from the Action Manager services on systems where an eTrust Audit Client is running, and writes it to the event database. The Collector service should be registered by the Portmap service so that it can start only if the portmap is running.

Registry Keys

Several registry keys contain values that affect the functioning of the eTrust Audit Collector. They are as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Data Server\Database

For information on the values in these keys, see Windows Registry Entries (see page 93).

Alternative Audit Collector

This feature allows eTrust Audit Action Manager to automatically send events to an alternative Audit Collector if the primary Collector does not respond.

User Perspective

Audit Collector is an application used to save events in the Audit DB. Several Audit Recorders send events to the Collector(s) via Router and Action Manager. If for some reason data cannot be sent to the Collector, or Collector failed to pass them to the DB, the Action Manager event queue can overflow along with data lost. To avoid this, the user can define two Collector hosts, primary and secondary. A typical rule Collect is:

```
Rule Collect
Action Collector; host1; comment
```

The same rule can be extended by the following method:

```
Rule Collect
Action Collector, 'host1, host2'; comment
```

The Action Manager always begins to send data to the primary Collector, which is the first host defined in the list ‘host1, host2’. If the sending fails, the Action Manager attempts to send data to the second defined host, host2. If both Collectors are unavailable, this combined destination is then marked as ‘bad destination’. After RetryDelay (a queue parameter) when queued data is sent again, or after SwitchTimeout expires, the Action Manager attempts to send to the primary Collector again, and switch to the secondary one if the primary is unreachable.

Parameter SwitchTimeout is optional and can be defined under the Registry HKLM\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Actions\collector\Parameters. Its default value is 7200 sec.

Note: In Audit r8, the Alternate Collector can also be configured through the Policy Manager.

Redirector Service (SeLogRd)

The eTrust Audit Redirector service, SeLogRd.exe, reads the local audit file created by eTrust Access Control and forwards it to the router. The local audit file contains eTrust Access Control events originating on the local machine.

You control Redirector service by editing the configuration file, logroute.cfg. For details, see Redirector Configuration File (see page 45).

Registry Keys

Several registry keys contain values that affect the functioning of the eTrust Audit Redirector. They are as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAudit\
Client\Redirector

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\SeOS\logmgr

eTrust Audit limits the size of the audit files. As a result, when events are generated faster than they can be forwarded—for example, a router service is not running, or too many events are being generated during a peak situation—it is possible to lose data.

You can guarantee delivery of records to the router by making changes to the values in the registry. You can permit the files to exceed their prescribed maximum size by setting the option to overwrite backup files to 0.

For information on the values in these keys, see Windows Registry Entries (see page 93).

SNMP Recorder Service (snmprec)

The eTrust Audit SNMP Recorder service, snmprec, traps SNMP messages sent to a machine and then passes them onto the default router. By default, the default router is the local host.

Registry Keys

Several registry keys contain values that affect the functioning of the eTrust Audit SNMP Recorder. They are as follows:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Recorders\SNMP recorder

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Recorders\DefaultRouter

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Recorders\SNMP Recorder\cfg\snmptd_rec.mp

For information on the values in these keys, see Windows Registry Entries (see page 93).

Portmap Service

The eTrust Audit Portmap service, portmap.exe, manages a table of correspondences between ports (logical communications channels) and the services registered at them. It provides a standard way for a client to look up the TCP/IP or UDP port number of an RPC program supported by the server. This service runs on any Windows host on which an eTrust Audit component is installed.

Note: For Windows, eTrust Audit installs the Sun RPC portmapper.

Chapter 3: UNIX Daemons

This section contains the following topics:

Introduction

eTrust Audit installs several daemons on UNIX systems. These daemons enable the information flow between Audit components by collecting, reading, and forwarding information from all sources in the system. This chapter describes the Audit daemons.

UNIX Daemon Names

The eTrust Audit daemons on UNIX platforms include the following:

eTrust Audit Software Component	UNIX Daemon Name
The eTrust Audit Action Manager	acactmgr
The eTrust Audit Distribution Agent	acdistagn
The eTrust Audit Collector	aclogrcd
The eTrust Audit Log Router	aclogrd
The eTrust Audit Generic Recorder	acrecorderd
The eTrust Audit Distribution Server	acdistsrv (Solaris 10)

Commands to Control the Daemons

You can issue commands to control the eTrust Audit daemons as follows:

1. Login as root.
2. Using either the Bourne or Korn shells, use the steps in the topic for your UNIX platform.
3. Depending on your UNIX platform, do the following:

   Solaris

   From the shell prompt, enter the following command:

   ```
   /etc/rc2.d/S77servicename
   ```

   AIX

   From the shell prompt, follow these steps:

   a. Enter the following command to set environment variables in preparation for starting the eTrust Audit daemons:

      ```
      ./opt/CA/eTrustAudit/bin/ac_set_env.sh
      ```

   b. Enter the a command as follows:

      ```
      /opt/CA/eTrustAudit/bin/servicename -start
      ```

   HP-UX

   From the shell prompt enter the following command:

   ```
   /sbin/rc2.d/S770servicename
   ```

   Tru64 and Linux

   From the shell prompt enter the following command:

   ```
   /opt/CA/eTrustAudit/bin/servicename -start
   ```

   where servicename is one of the listed daemon names in UNIX Daemon Names (see page 27).

   For iGateway, enter the following command:

   ```
   /opt/CA/SharedComponents/iTechnology/S99igateway -start
   ```

   Note: If you update eTrust Audit from 1.5SP3 to r8, the services are located in /usr/eaudit/bin.

   If you update eTrust Audit to r8SP2, the services are located in install_dir/bin. The value of install_dir is the location of the installed eTrust Audit component, entered during installation. By default, install_dir is /opt/CA/eTrustAudit.

Command Syntax

The following command syntax applies to all eTrust Audit daemons portmap:

```
daemon options
```

where *daemon* is the name of the daemon. The options are described in the topic that follows.

Command Parameters

The following list describes the available command parameters on UNIX machines:

-help

Displays these syntax options.

-debug

Starts the service in foreground mode; that is, it routes debug messages to the console (STDOUT). For example, the following command starts the service and routes the output to the console:

```
daemon -debug
```

You can use the following options:

-trace options

Starts a trace of the service. For example, the following command starts the service and routes debug messages to the console and a file named errors.txt:

```
daemon -debug -trace dest1 STDOUT dest2 errors.txt
```

See the description of the -trace option later in the list.

-install

Installs the service.

You can use the following options:

-user name

Lets you specify the name of a user authorized to install a service on the system. You should combine the -user and -pwd options as follows:

```
daemon -install -user user01 -pwd password
```

-pwd password

Lets you specify the password of a user authorized to install a service on the system. You should combine the -user and -pwd options as follows:

```
servicename -install -user user01 -pwd password
```

-trace options

Starts a trace of the service after installing it. For example, the following command installs the service and routes debug messages to the console:

```
daemon -install -trace dest1 STDOUT
```

See the description of the -trace option later in the list.

-remove

Removes the daemon from the UNIX process startup list.

You can use the following options:

-trace options

Starts a trace of the service while removing it. For example, the following command uninstalls the service and routes debug messages to a file named errors.txt:

```
daemon -stop -trace dest1 errors.txt
```

Additionally, you can use the redirect symbol, >, as follows to open a console an direct the output to a file:

```
daemon -stop -trace dest1 STDOUT > errors.txt
```

See the description of the -trace option later in the list.

-start

Starts the service in background mode; that is, without a console.

You can use the following options:

-trace options

Starts a trace of the service while starting it. For example, the following command starts the service and routes debug messages to a file named errors.txt:

```
daemon -start -trace dest1 errors.txt
```

See the description of the -trace option later in the list.

-stop

Stops the service.

You can use the following options:

-trace options

Starts a trace of the service while stopping it. For example, the following command stops the service and routes debug messages to a file named errors.txt:

```
daemon -stop -trace dest1 errors.txt
```

See the description of the -trace option later in the list.

The -trace option applies to all parameters, except -help, as follows:

-trace options

Turns on trace mode, which routes trace-level messages of a specified level to the destination. You can specify the following trace options:

-dbglvl n

Sets the debug level. n is the level from 1 to 5, 1 providing the least amount of debug information and 5 providing the most details. If you do not specify a value, 1 is the default.

-dest1 dest

Sets the primary output destination to display the debugging information to the console. dest can be one of the following:

- STDOUT - Routes messages to the console.
- STDERR - Routes messages to the console or to wherever you have redirected STDERR.
- filename - The name of file where you want the service to write the debug output.

-dest2 dest

Sets a secondary output destination to display the debugging information. dest can be one of the following:

- STDOUT - Routes messages to the console.
- STDERR - Routes messages to the console or to wherever you have redirected STDERR.
- filename - The name of file where you want the service to write the debug output.

Action Manager Daemon (acactmgr)

The eTrust Audit Action Manager service, acactmgr, reads events from queues where actions were placed by the Router and performs the specified actions defined for each event. The queues have parameters such as maximum action time, maximum file number and so on. These parameters affect the performance of the Action Manager.

For information about the Action Manager, actions, and configuration files, se (see page 42)e About Available Actions.

eaudit.ini File Entries

The daemon running on UNIX is controlled by the following entries in the eaudit.ini file:

```
Client\Router\Queue Manager\Queues
Client\Router\Queue Manager\Queues\xxxQueue\Queue Parameters
```

The eaudit.ini file is located in *install_dir*/ini/, where *install_dir* is the directory where you installed eTrust Audit. For details about the eaudit.ini file, see UNIX INI Files (see page 155).

Distribution Agent Daemon (acdistagn)

The eTrust Audit Distribution Agent service, acdistagn, receives policy files or message parsing (MP) files from the Policy Manager through eTrust Audit Distribution Server service running on Windows or Solaris 10 systems. The Distribution Agent service also removes old policy or MP files if instructed by the Distribution Server service.

The distribution agent service changes auditing requirements according to the policy it receives. The service notifies the Router to update the policy to get new rules.

eaudit.ini File Entries

When you install an eTrust Audit Client, you specify the name of the host where the Policy Manager runs. This is the only host recognized by the distribution agent service. It will reject attempts to update the policy from other hosts. However, you can add more servers to be recognized as trusted servers by editing the TrustedServers key of the Distribution Agent Service. This key is found in the following eaudit.ini entry:

```
Client\Management Agent
```

Note: Configuring multiple Trusted Servers can cause policy to be overwritten unintentionally. Although the Distribution Agent Service can accept policies from multiple Policy Managers, the last policy that is distributed to it is applied regardless to which Policy manager it came from.

Note for r8 SP2: In r8 SP2, the Distribution Agent service can send end-point validation status to the Policy Manager hosts (configured in the TrustedServers key). If this key contains a list of servers separated by comma, the Distribution Agent Service will try to send the end-point status to each of the servers in the list. End-point validation status contains the list of policies and MPs received and an indication of whether these are tampered or not.

The distribution agent service uses TCP/IP port 8025. You can change that port by editing the following eaudit.ini entry:

```
Ports\DistributionPort
```

Note: The distribution port 8025 is registered with IANA and is now a *well-known* TCP port reserved for the Audit Distribution Agent service.

The eaudit.ini file is located in install_dir/ini/, where install_dir is the directory where you installed eTrust Audit. For details about the euudit.ini file, see UNIX INI Files (see page 155).

Distribution Server Service (acdistsrv)

The eTrust Audit Distribution Server service, acdistsrv, distributes the policy files and MP files among the clients. It must run on the same system where the Policy Manager is located.

eaudit.ini File Entries

After you instruct the Policy Manager to distribute the policy, the relevant commands reach the distribution queue. The distribution server reads the distribution queue, selects from the compiled policy files or MP files, processes them, and sends them to the distribution agents according to the commands.

The distribution server tries to connect to the distribution agent as follows:

- The distribution server opens as many threads as possible up to a configurable number, called MaxThreadNumber (default 10), and connects to as many distribution agents as possible.
- If the connection succeeds, the agent starts receiving configuration files. After the transmission operation terminates successfully, the distribution log of the Policy Manager is updated.
- If the connection attempt fails (or in case the initial connection succeeds but afterwards a failure occurs), the transmission command is delayed. In the meantime, the distribution server does not wait but connects directly to the next agent in the list. After a pre-defined period (the default value is 24 hours) of failed connection trials, the distribution server terminates the transmission attempts. In any case, the distribution log of the Policy Manager is updated.

The Distribution Server's eaudit.ini file entries can be found in:

```
Policy Manager\Distribution Server
```

The distribution server service uses TCP/IP port 8026. You can change that port by editing the following eaudit.ini entry:

```
Ports\DistributionControlPort
```

Note: The distribution control port 8026 is registered with IANA and is now a *well-known* TCP port reserved for the Audit Distribution Server service.

The eaudit.ini file is located in install_dir/ini/, where install_dir is the directory where you installed eTrust Audit. For details about eaudit.ini, see UNIX INI Files (see page 155).

Log Router Daemon (aclogrd)

The eTrust Audit Log Router daemon, aclogrd receives events from a number of different sources. It handles received events according to the filters specified in the router configuration file.

For details, see UNIX INI Files (see page 155).

eAudit.ini File Entries

The daemon running on UNIX is controlled by the following entries in eAudit.ini:

```
Client\Router
```

The file is located in *install_dir*/ini/, where install_dir is the directory where you installed eTrust Audit.

Collector Daemon (aclogrcd)

The eTrust Audit Collector daemon, aclogrcd, receives information from the eTrust Audit Action Manager daemons on systems where an eTrust Audit Client is running, and writes that information to the event database.

For details, see UNIX INI Files (see page 155).

eaudit.ini File Entries

The daemon running on UNIX is controlled by the following entries in eaudit.ini:

```
Data Server\Database
Data Server\Collector
```

The eaudit.ini file is located in install_dir/ini/, where install_dir is the directory where you installed eTrust Audit.

Generic Recorder Daemon (acrecorderd)

The eTrust Audit Generic Recorder daemon, acrecorderd, reads the logs created by UNIX operating system, by third-party applications running on the UNIX station, or both, and sends them to the Audit Router daemon, aclogrd, for further handling by eTrust Audit.

You can edit the recorder configuration file, recorder.ini, to specify which events are to be recorded. For details, see UNIX INI Files (see page 155).

Recorder.ini File Entries

The daemon running on UNIX is controlled by the following entries in recorder.ini:

Recorder_Modules

The file is located in *install_dir*/ini/, where *install_dir* is the directory where you installed eTrust Audit.

SNMP Recorder Daemon (snmprec)

The eTrust Audit SNMP Recorder service, snmprec, traps SNMP messages sent to a machine and then passes them onto the default router. By default, the default router is the local host.

For details about eaudit.ini, see UNIX INI Files (see page 155).

eaudit.ini File Entries

The daemon running on UNIX is controlled by the following entries in eaudit.ini:

Client\Recorders\SNMP Recorder

The eaudit.ini file is located in *install_dir*/ini/, where *install_dir* is the directory where you installed eTrust Audit.

Chapter 4: Configuration Files

This section contains the following topics:

UNIX INI and Configuration Files Introduction

The Client components on UNIX systems are controlled by entries in the following .ini files:

- eaudit.ini
- recorder.ini

The files are located in eTrustAudit_root/ini/, where eTrustAudit_root is the directory where you installed eTrust Audit.

The eTrust Audit Router reads the .cfg (policy) files that contain filters that are made up of rules, and actions and targets. Using these rule (policies) the log router, aclogrd, filters the forwarded events and discards some of them.

The .cfg files are located in directories that are specific to the operating system type. For Windows systems, the default directory is \eTrust Audit\cfg. For UNIX systems, the default directory is /opt/CA/eTrustAudit/cfg.

About Queues

The events the log router receives from the recorders are written into queues. These queues are specified as follows:

For Windows systems

The queues are located in directories specified in the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\DirectoryName

For UNIX systems

The queues are located according to specifications in the following section of the .ini file:

Client\Router\Queue Manager\Queues

The three predefined queues are as follows, however, you can define your own queues:

- Default
- AlertQueue
- CollectionQueue

About Queue Rules

The queue to which the router writes depend on the rules defined in the Queue Rules key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\AlertQueue\Queue Rules

The form of queue rules differs depending on the operating system.

Windows

The form of a queue rule is a follows:

rule_name

as registry value name and

action; target name

as value data (for a specific target)

OR

rule_name

as registry value name and

action;

as value data (for all targets)

UNIX

The form of a queue rule is a follows:

rule_name = action\; target name

for a specific target

OR

rule_name = action\;

for all targets

Example

For example, if the .cfg file contains a rule with the action, Collector, the records are written to the collection queue, because this queue, as defined by the Queue rules, includes the rule Collector.

You can add rules in the registry, to customize your settings. By default, actions for which you have no defined rules are directed to the default queue. If you want actions to be directed to the alert or collection queue, you must add a rule. In the following example, the 'file' rule was add to define that actions of this type be directed to this queue, and not to Default directory.

```
myrule = file\;
```

Any event that has some attached actions may be placed in a several queues.

About Available Actions

The Action Manager uses the saved events to determine which action to take for a specific event.

Available Action Parameters

You can control how The Action Manager reacts to events received from the queue, by using the available actions. The following entries describe the available Action Manager settings.

Note: For actions in which you can enter a host name or an alternate host name, you can enter more than one name in that field as long as the names are separated by commas.

Collector

Specifies a Collector host to receive the events from the queue. You can also define an alternate host to receive events from the Action Manager if there is no response from the system defined in the Host Name field.

E-Mail

Defines an e-mail address to receive events. The e-mail address must be in the following format:

username@organization.com

External Program

Defines an executable or batch file that the Action Manager runs when an event is received. Specify the fully-qualified program name and any parameters that are needed.

File

Defines a file to which the Action Manager writes events. Specify the fully-qualified file name.

Route

Defines a remote host to receive router events. Specify the host name of the router you want to handle these events.

Screen

Specifies a host screen to receive events. The Action Manager sends events to the screen on the chosen host.

Note: The screen action is for Windows systems only.

Security Monitor

Specifies the Security Monitor host you want to receive events. You can also define an alternate host to receive events from the Action Manager if there is no response from the system defined in the Host Name field.

SNMP

Specifies a host to receive events sent from the queue by the SNMP protocol.

Unicenter

Defines a Unicenter host to receive events. The Action Manager sends events to the local Unicenter agent (installed on the eTrust Audit host that performs the action) for forwarding to the Unicenter Event Management Console on the host you specify. You can also specify needed parameters.

Note: Status codes from eTrust Access Control are translated to their generic equivalents. In the Unicenter Event Management Console, events display color codes and status icons. The Unicenter Event Management Agent must be installed on the host where the Action Manager runs.

eSCC Status Monitor

Sets the status of a corresponding eTrust World object. Specify the product name and any optional parameters.

Performing Available Actions on Remote Servers

You can configure available actions so that they are performed on a remote server, using the options in the group box provided on each action's properties panel. The remote action tells the Action Manager to move records from a queue to a remote router and performs any action on this remote host.

To create a remote action

1. Access the Policy Manager and select or create a policy.
2. Access an existing rule or create a new rule.
3. Navigate through the Rule Wizard or Edit Rule dialogs until you can edit or define an action.

   A list of actions displays in the Browse Actions pane.
4. Select an action from the list.

   A properties dialog for that action displays.
5. Select an action and then click New in the Action List pane.

   The properties dialog displays and offers options for configuring that action. A group box displays at the bottom of the pane for defining remote server names by specific name or by AN group.
6. Enter the host name and other information for the server, or the remote server, as needed.
7. Save your work, and distribute the policy.

Additional Action Parameters

You can specify parameters for the command. When you run a batch file, it contains the same parameters as a program. It is the responsibility of the program to parse the additional parameters.

path

You must specify the name of the program or batch file as follows:

- Use the full path name
- Ensure that the program file is in the directory defined by the %path% environment variable

If the program is located in the directory defined by the system environment variable, PATH, or in the directory install_dir\bin, you can omit the path. You cannot use quotation marks, so the path statement cannot include directories with spaces in their names.

timeout

You can specify an optional timeout period in seconds. The default timeout is 30 seconds. If the program has not exited when the timeout expires, it is terminated.

When you run a program or a batch file, the following occurs:

- The event is written into a file located in the TEMP directory (currently %TEMP%)
- The program itself gets the file name and the directory path.

Note: Using your API, you can open the file, retrieve the appropriate information, and run your software accordingly.

Redirector Configuration File

The redirector configuration file tells what should be sent where. By default, everything is sent to the router (local or remote).

While running, the redirector periodically reconfigures itself according to the contents of the redirector configuration file.

For SeLogRd, the configuration file is logroute.cfg, located in the install_dir\etc directory.

Router Configuration File

The router filters events and decides what action should be performed on these events according to configuration files.

The table that follows provides a brief overview of the statements and some sample rule statements:

Statements	Example	Description
Rule	select_NT (name of rule)	Every rule must start with the word Rule and have at least one action or one Do group.
Action	Monitor;localhost (target name)	Defines the action associated with the event. Possible actions include: monitor, file, Collector, and so on.
Include int	Log ~"^NT"	Include int is the internal language command, Include. Log ~"^NT" is the condition for including the event.
Exclude int	Log ~"^Oracle"	Exclude int is the internal language command, Exclude. Log ~"^Oracle" is the condition for excluding the event.
Do group	group_NT	Can be used for activating another group of rules. The statement enables implementing a nesting of rules.
Group	group_NT	Contains a list of rules.
Do Int Define	$Host_%Location %_Count Value(1)	This defines an internal integer variable that has the value of 1. Whatever is between % (such as %location%) is replaced by embedded text. In this case, it would be whatever value location is.
	$Host_%Location %_Count exists	Test for the existence of the variable $Host_%Location%_FailedCount
Incr	Host_%Location %_Count	Increments the internally defined variable
Decr	Host_%Location %_Count	Decrements the internally defined variable
Integer:	$Host_%Location %_FailedCount equal to 3	Declares that a variable or an SAPI field is an integer.

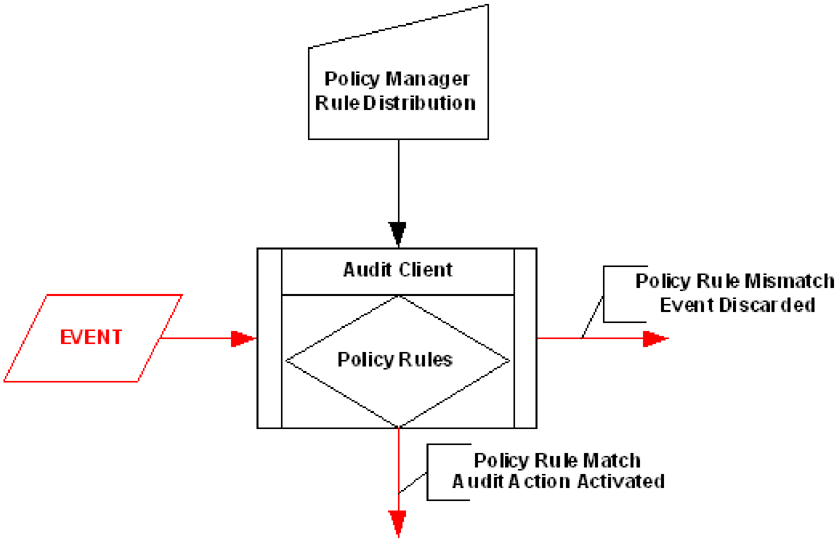
Statements	Example	Description
Do Int Define	$AlertEvent Src("eTrust Policy Manager") Type("Alert")	Defines a variable. It can be used to generate a new event.
Do Int Set	$AlertEvent.User User	Sets the value of User in the generated event by copying the value contained in the token User, which is found in the event currently filtered.
Do Int Delete	$AlertEvent	Deletes the generated events.
Do Int NewEvent	$AlertEvent	Generates a new event.

Chapter 5: Internal Language for Rule Filtering

This section contains the following topics:

Introduction

When you use the eTrust Audit Policy Manager User Interface, a policy can be defined to manipulate events received by the Router which is part of the Client component. You can view a policy as a collection of hierarchical rules written using the internal policy language. A Policy rule is similar to a script of commands that are triggered by the events received by the Router component of the Audit Client.



Each event that passes through the Client (Router) is tested against the policy rules that are created and distributed by the Policy Manager. Policy rules are defined as sequences of Include (Select) or Exclude (Ignore) filters, and each event must pass through these filters before the event is selected for Audit Action destinations.

You must have a policy rule to assign an Audit Action to an event or a group of events. If no policy rule applies to an event, the event is ignored by the Router. In other words, a policy rule with appropriate Audit Actions is required for events to be displayed in the Security Monitor, for the event to be stored in the Collector database, and so forth.

eTrust Audit provides several ways of creating a policy rule:

- Use the predefined rules provided with the Policy Manager for various third-party products supported by Audit. These rules are useful in creating simple policy rules to select or ignore specific types of events from a specific source. To setup and configure the predefined rules, you must copy the rule and paste it into the new policy. Then you define an Action for the new policy rule. To use predefined rules, no Audit policy language skills are required.
- Another set of predefined rule templates are provided to create more complex policy rules. These templates are available through the web-based Policy Manager user interface provided to setup and configure these rules which include:
  - Simple rules to select and ignore events from a source.
  - Complex data reduction rules with summary events from one or many sources.
  - Simple and complex correlation of events from one or many sources.

To use the predefined rule templates, no Audit policy language skills are required.

- You can also create a new policy rule manually by using the Policy Wizard provided with the web-based Policy Manager. No special skills in policy language syntax are required. This wizard allows you to create policy rules of the same complexity as described under the predefined policy rule templates.
- Finally, you can modify a rule (using a text editor) created through any of the above methods or write a new rule manually in a text file and then copy it into the Policy Manager rule editor. If you use this approach, you can customize the existing rules or create a new rule of any complexity. This is an important and useful method to create new rules. This approach requires a good understanding of the policy language syntax and how policy rules are applied by the Router.

The following sections explain policy rule language syntax as if you want to create a new policy rule using a text editor.

Rule Definition and Distribution

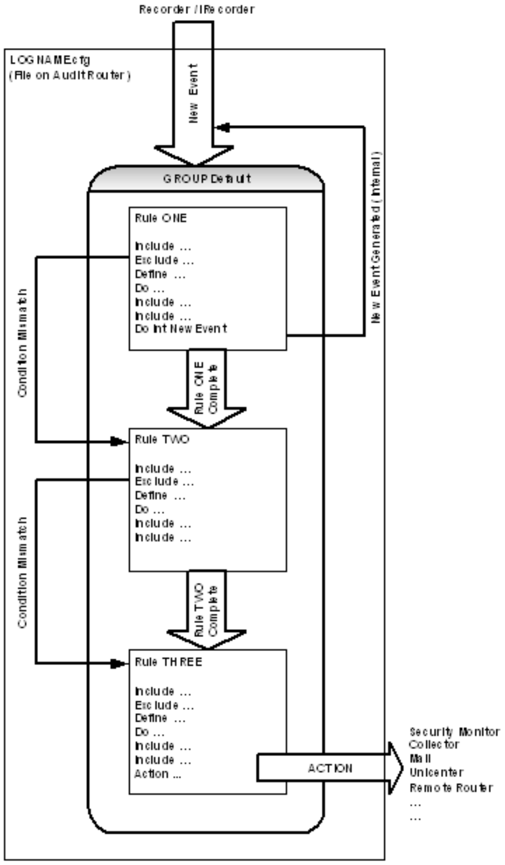
The Audit Policy Manager provides the following mechanism to define and distribute rules to the appropriate Router host:

1. Create a group of Policy rules, called a folder, for a group of audit nodes (ANs) where the eTrust Audit Client is running. In this step, you need to create the Policy rules and associate these rules with actions.
2. Create a group of ANs where events from various Audit Recorders/iRecorders are expected. A physical host running the eTrust Audit Client component can be associated with many ANs, which is a construct associated with Logname or source of events.
3. Associate the Policy Folder with one AN group and activate the policies defined on all hosts in that group. The activation process results in distribution of Policy rule files on the host running the eTrust Audit Router component.

All rules in a Policy are consolidated into a .cfg file created on the eTrust Audit Router host defined in the Audit Node Group. For more information on how to accomplish these tasks, refer to the *eTrust Audit and eTrust Security Command Center Implementation Guide* or the Audit Adminstrator online help.

Rule File Structure

A Rule File is a collection of Rules organized into named Groups. Each new event is processed sequentially through the defined Rules and filters. In case of a mismatch condition, processing of the event starts from the beginning of the next defined Rule. During the processing of an event, a Rule may be programmed to generate a new event. The new event starts the processing of the Rules from the beginning and works its way through as any other new event.



Rule Execution

It is important to note that the Rules are triggered by events. In other words, if there are no matching events, the Rule file remains inactive. A Rule is invoked, from the beginning, for each event. As long as conditions defined in the rule match, the event continues through the rule. In case of a mismatch, the event is dropped from the current rule and starts processing from the beginning of the next Rule. Once the event has gone through all the defined Rules, further processing for the event stops.

Basic Rules

A Rule, in its simplest form, is a condition to select certain events and associate the selected events with an Action. To select a group of events, you need to specify a condition that uniquely identifies the group of events. One of the most common conditions is to select on the basis of the Logname of the event. The Logname field uniquely defines the type and origin of each event. Since each Audit event is guaranteed to have the Logname field, it is a convenient way to select events for processing. For example, to send all events from a CCure Badge system to Audit Security Monitor, this can be defined as follows:

```
Include Int Logname == "CCURE Badge"
Action Monitor; localhost
```

The Include statement selects all events in which the Logname field of the event is equal to the string "CCURE Badge". If this is the only applicable rule in the Audit Router, only CCure events are routed to the local Audit Security Monitor while all other events are ignored.

If you replace the Include with an Exclude statement, the results are completely opposite that of an Include statement.

```
Exclude Int Logname == "CCURE Badge"
Action Monitor; localhost
```

In this case, all events in which the Logname is equal to the string "CCURE Badge" are excluded from the local Security Monitor destination. In other words, all events that do not contain the "CCURE Badge" string in the Logname field are sent to local Audit Security Monitor.

Note: The Int in the Include/Exclude statements stands for "Internal" Audit Policy Language and is part of the syntax for all Policy Language commands.

It is important to note that Audit supports two types of rules:

- Logname specific rules (all Logname/AN Types except Generic)
- Generic rules (Logname/AN Type set to Generic)

Logname specific rules means that the rule was created under a Policy with an AN Type other than Generic. These policies already contain a high-level or implied filter that only processes events with a Logname field (Log) that matches the AN Type.

Generic rules are able to process ANY events that the Audit Router receives.

Rule Language Commands

The following sections describe the functional areas of the policy rule language:

Event Selection

Include and Exclude commands provide a mechanism to select and filter events in a Rule. The syntax of these commands is as follows:

```
Include Int Condition

Exclude Int Condition

Include Int Condition1, Condition2, …

Exclude Int Condition1, Condition2, …
```

If an event triggers an Include statement in a Group of Rules, the Include condition is evaluated. If the condition evaluates to True, the event is selected for further processing in the Rule. If multiple Include statements are specified consecutively, the evaluation logic of the Include statement differs from when a single Include statement is present.

The Include condition is a binary operation resulting in either True or False. Multiple conditions, separated by commas, results in logical AND between conditions. Conditions combined by logical OR multiple Include statements must be specified consecutively. For example,

```
Include Int Condition1

Include Int Condition2

Include Int Condition3

Include Int Condition4
```

In the above example, if any of the above conditions evaluates to True, the event is processed.

If an event encounters an Exclude statement in a Group of Rules, the Exclude condition is evaluated. If the condition evaluates to True, the event is ignored and no further processing is done in the current Rule. If the Exclude condition evaluates to False, the event is selected and further processing continues in the current Rule.

The Exclude condition is a binary operation resulting in either True or False. Multiple conditions separated by commas result in logical AND between conditions. Conditions combined by logical OR multiple Exclude statements must be specified consecutively.

Examples

- Select all events from Snort recorder in which the source IP is 10.10.10.1 and the Snort ID is 'nnn':

  ```
  Include Int Log == "Snort", SrcIP == "10.10.10.1", NID == "nnn"
  ```

- Select all events from Snort recorder in which the source IP is 10.10.10.1 or the Snort ID is 'nnn':

  ```
  Include Int Log == "Snort", SrcIP == "10.10.10.1"

  Include Int Log == "Snort", NID == "nnn"
  ```

- Select all MS IIS events in which cs_uri_stem field is set to any of the following:
  /scripts/root.exe , /MSADC/root.exe, /c/winnt/system32/cmd.exe, or /d/winnt/system32/cmd.exe

  You can specify the include statement as follows:

  ```
  Exclude Int Log != "MS IIS"

  Include Int cs_uri_stem == "/scripts/root.exe"

  Include Int cs_uri_stem == "/MSADC/root.exe"

  Include Int cs_uri_stem == "/c/winnt/system32/cmd.exe"

  Include Int cs_uri_stem == "/d/winnt/system32/cmd.exe"
  ```

- Select all events in which the Logname does not begin with the string "CCure Badge"

  ```
  Exclude Int Log ~"^CCURE Badge"
  ```

In this case, the Exclude command ignores the event if the Logname contains the token "CCure Badge". The commands following the Exclude command are triggered only if the Logname does not match "CCure Badge". Explanation of the '~' operator and the regular expression symbol '^' is provided in the next section.

Event Selection Conditions

Syntax of the Condition in an Include and Exclude command can be described as follows:

type: Operand Operator Value

Case Insensitive: Operand Operator StringValue

ci: Operand Operator StringValue

where *type* can be any one of the following:

Integer

Timestamp

Time

Date

Note: Ci or Case Insensitive can be used when string type of comparison (default) is desired. The Operand identifier is always case-insensitive. Ci or Case Insensitive is used only when the Value must be compared in case-insensitive mode. Condition without Ci or Case Insensitive is compared as case-sensitive.

Operand can be one of the following:

Constant (Integer, String)

Event Field Name

Variable Name

Variable Property Name

Operator can be one of the following three types:

1. String or Regular Expression Comparison

   If no type is specified, the comparison defaults to case-sensitive String or Regular Expression type. To make the comparison case-insensitive, use ci: or Case Insensitive:. All regular expressions or strings must be in double quotes.

   Supported Operators for regular expression comparisons are as follows:

   EQUAL TO or ==

   DIFFERENT THAN or !=

   Supported Operators for regular expression comparisons are as follows:

   MATCHES or~

   PART OF

2. Integer, Timestamp, Time, or Date Comparisons

   For these types of comparisons, type must be specified. Supported Operators are as follows:

   EQUAL TO or ==

   DIFFERENT THAN or !=

   GREATER THAN or >

   GREATER OR EQUAL TO or >=

   LESS THAN or <

   LESS OR EQUAL TO or <=

3. Variable or Event Field Existence

   This is a special Operator that can be used detect the existence of a variable created elsewhere. This Operator requires only the variable Operand without any Value.

   Variable EXISTS

   It returns True if the specified variable has been defined and has not expired.

   Event Field EXISTS

   It returns True if the correctly processed event contains specified field.

Note: To negate or void the result of any comparison, a NOT can be used before the Operand of the Condition.

Operand in a condition can be any field name from the current event to be compared with the value of the field in the event. By default, all comparisons are of string type, and the Value is compared in case-insensitive mode. To compare other modes, override the comparison with the type: which can be Integer, Timestamp, Time, or Date. The following table provides various formats associated with these data types:

Data Type	**Permissible Formats**
Integer	Integer value
Timestamp	11 pm 11:23:00 11:23:30 pm 23:30

Data Type	Permissible Formats
Date	August 2, 2000 2 Aug, 2000 2 Aug, 00

The following table shows various supported Operators with examples to highlight the possible format of the Operand. Please note that identifier names starting with a $ or _$ are Audit variables.

Operator	Examples (Conditions)
MATCHES or ~	1. Log ~"^ CCURE Badge" 2. Log MATCHES "NT.*" 3. Src MATCHES "Printer"
EQUAL TO or ==	1. SrcIP EQUAL TO "10.10.10.1" 2. $Counter == 3
DIFFERENT THAN or ! =	1. _$FailedLogCount ! = 3 2. DstIP ! = "1.1.1.1"
PART OF	1. User PART OF "John Smith" 2. NID PART OF "123098333"
GREATER THAN or >	1. _$Counter > 2 2. _$Counter GREATER THAN 2
GREATER OR EQUAL TO or > =	1. _$FailedLogCount > = 3 2. _$Counter GREATER OR EQUAL TO 3
LESS THAN or <	1. VirusCount < 10 2. _$Counter LESS THAN 3
LESS or EQUAL TO or <=	1. VirusCount <=10 2. _$Counter LESS OR EQUAL TO 3

Include/Exclude Condition Examples

1. Select events with event field InfectionType set to File and the variable HostCount is less than 3:

   ```
   Include Int

   InfectionType == "File"

   _$HostCount exists,

   Integer: _$HostCount < 3
   ```

2. Select events in which the event field VirusName is not part of the strings in the variable VirusNames:

   ```
   Include Int

   VirusName Not Part of _$VirusNames
   ```

3. Select all failed Firewall events with Taxonomy field set to

   ```
   Network Security.Frewall.*.F

   Include Int

   Taxonomy ~ "^Network

   Security\.Firewall\..*\.F$"
   ```

Regular Expressions

Include and Exclude Conditions may contain Regular Expressions of various complexities. Specifically, the Value part of the Condition may be a UNIX type of Regular Expression. A brief overview of the important regular expression symbols (also called metacharacters) is as follows:

Regular Expression Symbol	**Description or Action**
.	Match exactly one character
*	Zero or more repetitions of the preceding character
^	Match from the beginning of the pattern
$	Match with the end of the pattern
\	Escape the following character
[ ]	Match one from the set within [ ]
[^]	Match all except from the set within [ ]
\{ \}	Match a range of instances from within { }

When using square brackets ([]) in Regular Expressions, consider the following:

- Do not use the backslash character to escape square brackets if you are using them as a range operator. When you use square brackets as a range operator, any character in the range, excluding the square brackets are matched against the specified string.

  Example: [0-9]

  Any digit from 0 to 9 is searched through the specified string.

- Use the backslash character to escape square brackets if you are using them as literal characters. When you use square brackets as literal characters (with the backslash), every character within the square brackets, including the brackets, is compared with the specified string.

  Example: \[0-9]\

  The series of 4 characters [, 0, -, 9,], in this order are searched through the specified string.

Regular Expression Examples

Regular Expression Examples	**Matching Patterns**
"NT*"	NT, NTT, NTTT, NTTTT, etc.
"NT.*"	Any number of characters following NT. Specifically, it matches NT-Security, NT-Application, NT-System, etc.
"^\."	A pattern that begins with a dot
"^[^.]"	A pattern that does not begin with a dot
"^NT.*"	The pattern NT.* must be the beginning of the word
"^NT$"	The pattern NT must be the complete word
"d[i,o]g"	dig or dog
"[a-zA-Z]"	Any letter
"[^0-9A-Za-z]"	Any symbol (not a letter or a number)
"[A-Z]*"	Zero or more uppercase letters
"[A-Z].*"	An uppercase letter followed by zero or more characters
"[A-Z][A-Z]*"	One or more uppercase letters

Variables

With variable support, the eTrust Audit Policy Language provides a powerful mechanism to count or flag events within a time range. Using a variable, you can set up a timer in your Rules and invoke additional commands when the timer expires. Using various thresholds on these counters and timers, additional alert or trigger events can be generated to indicate special conditions. Variable manipulation commands allow you to generate a suspicious event after a certain number of repetitions (within a specified time) of an event which in itself might not be suspicious.

Variable Names

Variable names must begin with $ or _$ followed by a letter (‘A’ – ‘Z’, ‘a’ – ‘z’). The remainder of the symbols in the name can be a letter (‘A’ – ‘Z’, ‘a’ – ‘z’), underscore, or digits (0 – 9). There is no restriction on the number of characters in a name, but it is recommended that you keep the name size within 64 characters. A percent symbol ‘%’ can be used in the name to create Dynamic Variables (as described below). Spaces or other non-letter symbols are not allowed in variable names. Variable names are case-insensitive.

Variable Scope

Variables can have local or global scope. Local scope limits the visibility of the variable to the policy in which it is defined. This means that local variables, defined with _$ prefix, are visible in all rules of a policy in which the local variables are defined.

Global Variables

Global variables (defined with a leading $ in the name) are visible across all policies and folders on a Router host. In other words, if variable names are not selected carefully, two policies written by two different groups could inadvertently overwrite each other's variables. With global variables, you must be very careful in naming your variables. To minimize name conflicts, you should use long and descriptive names for global variables. It is also recommended that you create a global variable only when your rule logic requires that the global state of events must be maintained across all policies and folder on the Audit Router.

Local Variables

Internally, all variables are global to the Audit Router engine. Audit Policy Language provides a mechanism to tag the variables internally to create a local scope of variables. To prevent inadvertent definition of duplicate variables names, the Audit Router engine prefixes all local variables (defined with a leading _$ in the name) with the Rule Filename (on the Router) and the Group of Rules in which a variable is defined. For example, if the following definition is created in FLOG Group under NT-POLICY:

```
Do Int Define _$Count_%User%_Logins
```

Audit Router internally changes the variable name to:

```
$Rules-NT:FLOG:Count_%User%_Logins
```

The Rules filename created on the Router host will be named as Rules-NT. The tag Rules-NT:FLOG: ensures that the variable's visibility is limited to the Group of Rules in which it is defined. The variable name change is internal to the Audit Router and is not visible to the user when editing the Rule using an editor or Policy Manager. With internally generated events, Audit Viewer may display changed variable names in the event detail.

When creating local variables, make sure that the variable name is unique within the group of Rules in a policy.

Variable Creation

Variables can be created by using the Do command. As with other commands, the Do command can only be triggered by an event. For example:

```
Do Int Define $Counter Value(1)
```

```
Do Int Define _$Flag Value("Red")
```

The above two commands create a global variable Counter and a local variable Flag. A variable is created with a default property called Value.

If you create a variable without specifying any properties or expiration criteria, then the variable created is a copy of the event that caused the creation of the variable. This means that the variable created contains all the fields(properties) and values that the original event contained. This is useful for creating variables that are used as notification events later so that you do not have to manually define all the properties as copies of the original event. For example:

```
Do Int Define $Counter Value(1)
```

creates a global variable containing all the fields (properties) and values that the event which caused this creation contained.

Variable Expiration

All variables created during processing of an event have a limited time to live. By default, all variables expire after one hour. In the above example, variables Counter and Flag expire in an hour. During the creation of a variable, you can specify a different expiration time (in seconds). Variables can be deleted before the expiration time of the variable.

```
Do Int Define $Counter ExpireIn (300)
```

```
Do Int Define $Flag ExpireIn (30)
```

In the above example, Counter expires in 5 minutes and Flag expires 30 seconds after creation. You can also specify that a variable should be deleted only if it is not modified within a certain time interval:

```
Do Int Define $Counter ExpireSinceLastModified (600)
```

```
Do Int Define _$Flag ExpireIn (45)
```

In the above example, the variable Counter expires only if it not modified for 10 minutes, and the variable Flag will be deleted after 45 seconds regardless of its modification state.

Variable expiration feature can be used in setting up timers. For example, you can create a variable that expires in 300 seconds (5 minutes) and then check the existence of this variable (whenever an event triggers the variable check). If the variable does not exist, you can be sure that 5 minutes have elapsed since the creation of the variable.

Another option, called Notify, in variable expiration generates an event whenever a variable expires. This option is explained in the New Event Generation Section.

Variable Properties

Variables can be associated with an arbitrary set of properties.

```
Do Int Define _$Counter Value(1) User("John")
```

In the above example, the variable Counter is associated with two properties: Value which is set to 1 (integer) and User which is set to "John" (string). Value property is also the default property of all variables.

When you initialize or assign a value to a variable, you are actually initializing or setting one of the properties of the variable. As shown in the above examples, variable initialization can be done when you define the variable. For example, an event can be used to define a variable CMD and initialize its NID, User and Result properties:

```
Do Int Define _$CMD NID (420) User ("JDoe") Result ("F")
```

After an event triggers definition and initialization of a variable, the variable can be manipulated in several ways:

An event can set a new value for the variable properties:

```
Do Int Set _$CMD NID (425) User ("JSmith") Result ("S")
```

You can also use the 'dot notation' to set the values:

```
Do Int Set _$CMD.NID 420

Do Int Set _$CMD.User "JDoe"

Do Int Set _$CMD.Result "F"
```

An event can increment or decrement the variable property:

```
Do Int Inc _$CMD.NID
```

Or

```
Do Int Dec _$CMD.NID
```

If no property is specified, the default property Value is assumed:

```
Do Int Inc _$CMD
```

Or

```
Do Int Dec _$CMD
```

An event can add or subtract the variable property:

```
Do Int Add _$CMD.NID 3
```

Or

```
Do Int Subtract _$CMD.NID 1
```

If no property is specified, the default property Value is assumed:

```
Do Int Add _$CMD 3
```

Or

```
Do Int Subtract _$CMD 1
```

String Concatenation in Variables

The Add operator described above can also be used with variable properties containing string data. With string data, the Add operator concatenates the two strings. For example,

```
Do Int Define _$Var "String"
```

```
Do Int Add _$Var " and String"
```

The above two commands result in the _$Var containing "String and String"

You can also concatenate string contents of a variable with the string contents of another variable.

```
Do Int Define _$SCMD1 "That's"
```

```
Do Int Define _$SCMD2 "all folks!"
```

```
Do Int Add _$SCMD1 _$SCMD2
```

The above commands result in the variable _$SCMD2 containing "That's all folks!"

To concatenate string contents of a field in the current event, simply use the field name:

```
Do Int Define _$TODAY "Today's date is "
```

```
Do Int Add _$TODAY "May 28, 2004"
```

The above commands result in the variable _$TODAY containing "Today's date is May 28, 2004"

In the above examples, the default property (Value) of the variable is used. You can also use other properties in string concatenation:

```
Do Int Define _$ME Name ("John")
```

```
Do Int Add _$ME.Name "Doe"
```

Variable Deletion

An event can trigger the deletion of a variable:

```
Do Int Delete $ME
```

You can also delete a specific property of the variable:

```
Do Int Define $Var Prop ("Property")
```

```
Do Int Delete $Var.Prop
```

It is recommended that you delete variables when no longer needed to release system resources.

Note: Be careful when using Reserved Keywords (see topic Reserved Keywords for list) as property names. Properties with the same name as Reserved Keywords cannot be deleted separately.

Using Variables in Filter Rules

You can use variables in Include and Exclude conditions just as you would use other operands in conditions as described above.

```
Include Int Integer: $Count equal to 3
```

To use a property of the variable in the condition, use the variable property notation:

```
Include Int Integer: _$Count.Value equal to 3
```

You can also query whether a variable was defined, as follows:

```
Include Int $Flag exists
```

If the variable is defined, and it is not expired, the condition returns true.

Dynamic Variable Names

You can define variables that incorporate field values from the event record into the variable name. For example:

```
Do Int Define _$LoginCount%Location% Value (1)
```

%Location% is translated when an event triggers the execution of the rule, and %Location% is replaced by the value of the Location field in the event. Dynamic variables can be used to associate a variable to a specific value in an event. This feature is very useful in counting specific events within a specific period of time.

Dynamic variable names can include more than one token, as in the following case:

```
Do Int Define _$Count_%User%_%Location%_FailedLogins
```

In the above example, value of User and Location in the event replaces %User% and %Location% respectively.

Actions

Events that do not trigger an Action command are discarded by the Router. The Action command simply specifies the final destination (such as the Security Monitor, the Collector database, and so forth) for the event. If you want to discard or ignore events, you can simply not associate an Action with those events. This mechanism provides a simple way to detect and count the events and then forward a few selected events to the desired destinations.

Note: It is recommended that you use the SCRIPT_ACTION macro (explained in next section), rather then using the the Action argument directly.

The syntax for an Action command is as follows:

```
Action ActionArg; Target TargetArg
```

Each ActionArg has its own appropriate TargetArg.

Action Argument	Target Parameter	Description
Collector	Hostname or IP address	Sends the record to the collector service on the specified host
mail	recipient@domain	Sends e-mail to the specified account
File	Full pathname	Appends event details to the file with the specified pathname
Route	Hostname or IP address	Forwards the message, before filtering, to a router on another host
Screen	Hostname or IP address	Sends event details as a screen popup to a host or (if logged in) user
Monitor	Hostname or IP address	Sends event to the Security Monitor console running on the specified host
SNMP	Hostname or IP address	Sends the event to the SNMP server on the specified host
Unicenter	Hostname or IP address	Sends the event to the EM console on the specified host
eSCC Status Monitor	Product Name and Status	Sets status of corresponding eTrust World object
External Program	Program Name (Full Path)	For each event, invokes the external program

Action on a Remote Router

You can invoke an Action on a remote Router with the following syntax:

```
Action Remote hostname; ActionArg; TargetArg
```

Whenever the client station is not configured to perform the action specified, you must use the Action Remote command to set up remote execution of the desired Action. The ActionArg and TargetArg parameters are the same as in the Action command described above.

Parameter	Description
hostname	The host where the Action should take place
ActionArg	See the Action command, above
TargetArg	See the Action command, above

When an Action Remote command is triggered by an event, the event is sent to the remote host and the Action Manager running the remote host is instructed to activate the desired Action. Action Remote command is a useful command for the following reasons:

1. The Targets of the desired Action may not be known to the Action Manager running locally. Also, the Target application may not be running locally, for example, sending events to Unicenter, requires that the Unicenter EM must be available locally to Audit Action Manager. In the case where Unicenter EM is not installed locally, we must use Action Remote command.
2. Action Remote command usually results in better performance than the standard Action command described above. Improved performance is more significant when the Action is targeted to the Collector database.

Script Action Macro

SCRIPT_ACTION macro is not part of the Internal Language. It is used in complex rules in the Policy Manager. If an Action is completely specified in a Rule, the Rule becomes a specific Rule that may be valid only for the Router on which it is defined. To make the Audit Policy Rules easily portable, you can specify the SCRIPT_ACTION Macro wherever an Action needs to be specified. For example,

```
Include Int Log == "NT-*"

SCRIPT_ACTION
```

The SCRIPT_ACTION macro is automatically replaced by whatever Action or Actions are defined using the Audit Rule Wizard. If Audit Rules are developed with the SCRIPT_ACTION macro, the rules are ported or distributed to other Audit Routers with any change.

Note: The External Program and eSCC Status Monitor actions can accept Optional Parameters.

External Program Optional Parameters:

[arguments] [;TimeOut] [Description]

The following variants are supported with a space between Time Out and Description:

```
Action Program;ProgramName
Action Program;ProgramName; Description
Action Program; ProgramName; Arg1 Arg2 Arg3 Arg_i; Description
```

In the last expression it is possible to specify the Time Out as part of the Description

For example:

60

or

60 After space we can write description.

To define a Time Out, other than the default, at least one argument should be specified.

Action Program creates a process named ProgramName. The following arguments will be as the user defined Arg1 Arg2 Arg3 Arg_i

Do not specify any arguments to get a temporary filename for the file that contains the event.

The Arguments can be literal strings or tokens from the event. Action Program finds the token in the event and adds its value as an argument. If there is no token or value action prog, add a NULL pointer, '$' to represent a Token. Literal strings containing a space need to be enclosed in quotation marks if they should be passed as a single parameter to the Program.

- Example 1:

  prog.exe $User $Location (action prog add the value of the tokens user and location)

- Example 2:

  proc.exe user $User logged to $Location on $Date

- Example 3:

  proc.exe $User $Location $Date "Failed Authentication" Warning; 120

The process would be terminated by the action if it was not finished before the Time Out expired.

The action will use a default Time Out of 30 seconds if it is not defined exactly or if it is greater than 5 minutes.

eSCC Status Monitor Optional Parameters:

The eSCC Status Monitor takes 2 parameters: {StatusObject},{Newstate}

Example:

Policy Change,1

The eSCC Status Monitor action can only run on routers that have eSCC installed on them. So when setting up a policy to use this action, you may want to run it on a remote node.

New Event Generation

Audit Policy Language allows you to define and generate new events or alerts whenever selected events trigger event generation rules. New events are useful in counting events and timing sequence of events. New events can be marked with special values and informational text and can be forwarded to any Action target.

New Event Definition

A variable can be viewed as an event identifier and the variable properties as event fields. To define an event, you simply define a variable and set its properties to the corresponding field values. For example, to define an event called NimdaEvent, define the variable and its properties:

```
Do Int Define $NimdaEvent
  Src("Nimda")
  Log(Log)
  Category(Category)
  Date(Date)
  Location(Location)
  Status(Status)
  Severity(Severity)
  c_ip(c_ip)
  s_ip(s_ip)
  s_port(s_port)
  TimeZone(TimeZone)
  UriStem(cs_uri_stem)
  Count($ClientIP_%c_ip%_Count)
  Info("Count contains the number of times a similar event was     suppressed in the last 15 minutes")
```

In the above example, the Src field is set to the string "Nimda", the Count field is set to the value of the variable _$ClientIP_%c_ip%_Count, and all other fields are set to the corresponding values of the fields in the event that triggered the Define statement.

As described earlier, another and simpler way to accomplish the same would be to Define the variable without any properties. This results in all fields of the current event being copied to the variable as properties:

```
Do Int Define $NimdaEvent
```

After the event fields are copied, you could reset specific fields or create new fields as follows:

```
Do Int Set _$NimdaEvent Src("Nimda")

Do Int Set _$NimdaEvent Count(_$ClientIP_%c_ip%_Count)

Do Int Set _$NimdaEvent Info("Count contains the number of times a similar event was suppressed in the last 15 minutes")
```

New Event Generation

After you Define an event as a variable, with all its fields as properties, you can generate the new event using the NewEvent command. As with all other commands, the NewEvent command can be triggered only by an event:

```
Do Int NewEvent _$NimdaEvent
```

The NewEvent command generates the new event with its fields set to the properties of the _$NimdaEvent variable.

It is important to note that the new event generated by the NewEvent command starts its life as if it were generated by an external source. This means that the new event triggers all relevant rules including the one that generated the event. This behavior has the following implications:

1. When creating rules, you must handle new events generated by your rules. This can be easily accomplished by setting a unique field in the event that can be detected through a filter. For example, in the NimdaEvent, the Src property is uniquely set to "Nimda". This can be used in the following rule segment to select the newly generated event and route it to the desired target:

```
Include Int

_$NimdaEvent Exists, Src == "Nimda"

Do Int Delete _$NimdaEvent

SCRIPT_ACTION
```

2. If you do not protect the rule that created the new event from the new event itself, you could create an infinite loop condition. Without proper protection, the rule that created the event can be triggered by the new event again and again. This condition can be avoided by adding an Exclude command at the beginning of the rule as follows:

```
Exclude Int Src == "Nimda"
```

   You may also need to protect rules other than the one that generated the event. If you do not protect other rules, the new event could trigger commands that could result in unnecessary variable creation, counting errors, and other side effects.

Event Generation on Variable Expiration

Since all Policy Language commands are triggered by events, there may be a situation where a counter variable may expire before reaching its threshold value. For example, if a counter variable may be set to trigger an alert after 10 virus detected events. It is possible to have 9 virus detected events but before we get the 10th event, the counter variable has expired.

You can define a variable with Notify option to create an alert whenever the variable expires. Router creates artificial notification event about variable expiration if relevant property Notify was defined in variable.

```
Do Int Define _$Var Value(1) Notify(1)
```

Notification event contains all variable properties as fields of the notification event. In addition, following new field is added to the notification event:

Expired_Variable   Set to the expired variable name

Other fields added to the notification event include Date, Log and Type fields if they are not defined by user in the variable.

Note that if Notify is set to zero, no notification event is generated on variable expiration.

```
Do Int Define _$Var Value(1) Notify(1)
```

There is optional registry (ini file) DWORD value EventOnExpire that may be added to the registry key (ini file section) Router to turn-off the Notify option. Default value of EventOnExpire is 0, i.e. the option to send notification event is disabled. The option is enabled, if EventOnExpire is 1.

Rules and Groups

A Rule consists of a number of commands to filter or select incoming events and route the selected events to another Rule, a Group, or an external Audit Action. One or more Rules can be collected into a Group and one or more Groups can be collected into a rule file. Rules, Groups, and Rule Files are identified by unique names.

Group

A Group must be uniquely named within a policy by using the following characters:

'A' - 'Z', 'a' - 'z', '0' - '9'

Space, Underscore, Minus, Plus, Apostrophe

A Group can be named using the following syntax:

Group GroupName

A Rule File contains one Default Group and may contain one or more additional named Groups. Scope of a Group begins from the start of the Group and continues until a new Group is encountered. New events always start from the Default Group and invoke the commands in the first Rule in the Group and continue until all Rules in the Group are traversed. Current event does not automatically enter a new Group. A named Group can be called from a Rule as follows:

```
Do Group AnotherGroupName
```

With a call to another Group, the current event starts execution of commands in the Rules from the new Group. Once all the Rules in a Group are traversed, current event stops further execution, even if there are more Groups yet to be traversed. In other words, the current event does not enter a named Group unless the Group is explicitly called and after all the Rules in a Group are traversed. The current event does not return back to the Group or the Rule it was called from. To force the current event back to the Default Group, you can use Group command without any group name:

```
Do Group
```

Rule

A Rule must be uniquely named within a policy by using the following characters:

'A' - 'Z', 'a' - 'z', '0' - '9'

Space, Underscore, Minus, Plus, Apostrophe

A Rule can be created and named using the following syntax:

```
Rule RuleName
```

The Scope of a Rule begins from the first command in the Rule and ends when a new Rule or Group is encountered. At the end of a Rule, the current event automatically enters the scope of the next Rule listed in the Group and continues trigger commands in the next Rule. With the exception of the Default Group, a Group must be explicitly called from a Rule for the current event to execute Rules in the new Group. A Group may contain one or more named Rules. Normally, the current event starts executing commands from the beginning of the Rule and continues into the next Rule listed within the current Group. A named Rule can be explicitly called from a Rule as follows:

```
Do Rule AnotherRuleName
```

With a call to another Rule, the current event starts execution of commands from the new Rule.

Group Parameters

You can write more abstract Rules by passing parameters to a Group that can be used in the Rules to replace constant values, event fields, variables, and so forth, dynamically. Group Parameters are passed to the Group as a copy of the parameters, or *by value*. An identifier starting with a % symbol is a parameter and can be used anywhere a variable can be used in a Rule.

Declaring Parameters

You must declare parameters immediately after the Group definition statement using the following syntax:

```
Args Int %Parm1, %Parm2, %Parm3
```

For example, a Group can be declared with %Usr as a parameter and then use the parameter in a Rule as follows:

```
Group UserHandler

Args Int %Usr

Rule RuleName

Include Int EffectiveUser equal to %Usr
```

Argument declaration is optional and can be done after a Group name but before the name of first Rule of the Group.

Calling a Group with Parameters

As described before, syntax for calling a Group without parameters is:

```
Do group GroupName
```

To call a Group with parameters (transmit real values to a Group with parameter declaration), all parameters should be described after the group name in Do Group command. For example,

```
Do Group GroupName("Value1", Value2, $Var1, $Var1.Prop, Field3)
```

The above call to GroupName may be followed by the following declaration:

```
Group GroupName

Args Int %Parm1, %Parm2, %Parm3, %Parm4, %Parm5
```

In the above call to GroupName, the parameters are interpreted as follows:

Parameter	Description
"Value1"	String "Value1" copied to %Parm1
Value2	A non-string constant value or event field copied to %Parm2

Parameter	Description
_$Var1	Value property of the variable _$Var1 copied to _$Parm3
_$Var1.Prop	Prop property of the variable _$Var1 copied to %Parm4
Field3	A non-string constant value or event field copied to %Parm5

If the number of parameters and the list in the Args do not match, the parameters are matched from left to right, and the extra parameters are ignored.

You may specify default values in the Args list using the equal sign '=' to handle missing parameters. For example,

```
Args Int %Param3 = User
```

Another example with default values in Args list:

```
Group GroupName
Args Int %Param1, %Param2, %Param3 = User, %Param4 = _$Var2, %Param5 = "string", %Param6 = 7

Rule RuleName
Include Int EffectiveUser equal to %Param3
```

Reserved Words

The following words have specific meanings in the filter language:

ADD

AM

AT

CASE

CI

CS

DATE_YACC

DAY

DECR

DECREMENT

DEFINE

DELETE

DELETE_YACC

DIFFERENT

DY

EQUAL

EXISTS

FATAL_ERROR

GREATER

GROUP

INCR

INCREMENT

INSENSITIVE

INTEGER

LESS

MATCHES

MONTH

NAME

NEWEVENT

NOT

NUMBER

OF

OR

PART

PM

REL_OP

RULE

SCAN_ERROR

SENSITIVE

SET

STRING

STRING_CONST

SUB

SUBTRACT

THAN

TIME

TIMESTAMP

TO

VARIABLE

YR

The names of the months (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, and DEC) and short names of the weekdays (SUN, MON, TUE, WED, THU, FRI, SAT) are also reserved.

Complex Data Reduction Example

This data reduction example shows how we can use the Audit policy language rules to reduce event data from Nimda virus activity to a manageable level. The problem is that even when Microsoft IIS web server is fully-protected from the Nimda virus, Nimda activity from infected Clients can create a huge number of events on the IIS server. Security administrators monitoring the status of the web server can be easily overwhelmed by the number of events generated. In a corporate intranet environment, the security administrators may want to keep track of the Nimda virus activity on the clients and take corrective actions. In an infected intranet, it would be hard to keep track of hundreds of clients generating thousands of events. Nimda virus data reduction rules allow you to reduce the noise and monitor the infected Clients effectively.

The main idea is always to send the first Nimda activity event from a Client but then to count and ignore all events from that client until 10 minutes have elapsed. After 10 minutes, the next event from this Client should trigger a summary event with the count included in the summary event. With this algorithm, a security administrator monitoring events on the Security Monitor can easily manage the virus activity on the web server. A dataflow diagram is provided at the end of this example to show the logic of various Rules and Groups.

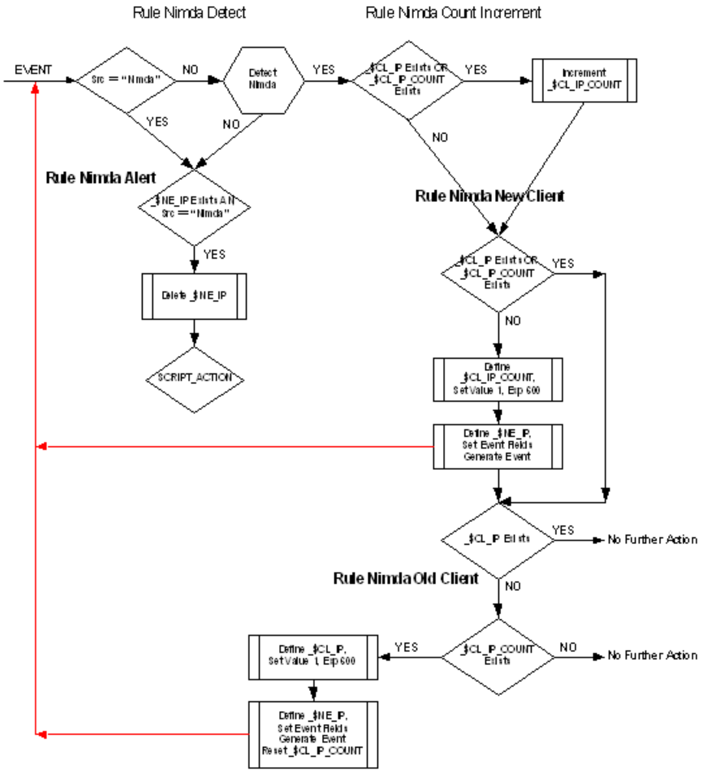
```
;<COMPLEX>
; Define a Rule called Nimda Detect in the Default Group.
; This Rule ignores all events with Src field set the
; literal "Nimda". All such events are self generated.
; When NIMDA virus is attempting to infect a protected
; machine, IIS generates events with uri_stem field set to
; various files and folders that NIMDA is trying to access.
; The Include statements in the Rule are combined with
; logical OR to detect Nimda virus if any of these files
; or folders are accessed. Once Nimda is detected, the Rule
; calls a Group Nimda Check where rest of the processing is
; completed.
Rule Nimda Detect

    Exclude Int Src == "Nimda"

    Include Int cs_uri_stem == "/scripts/root.exe"

    Include Int cs_uri_stem == "/MSADC/root.exe"

    Include Int cs_uri_stem == "/c/winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/d/winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem ==
    "/msadc/..%5c../..%5c../..%5c/..^C1^\../..^C1^\../..^C1^\../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..^C1^\../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..%5c../winnt/system32/cmd.exe"

    Include Int cs_uri_stem == "/scripts/..%2f../winnt/system32/cmd.exe"
```

```
Do Group Nimda Check
; Define another Rule: Nimda Alert. This marks the end of
; of the Rule: Nimda Detect.
; Nimda Alert Rule processes only those events that were
; internally generated in another Rule defined later. Note
; that the internally generated events are ignored by all
; other Rules. This Rule simply detects the internally
; generated events and invokes the SCRIPT_ACTION macro
; to route these events to whatever Action is defined.
Rule Nimda Alert
Include Int
  _$NimdaEvent_%c_ip%  exists,
  Src == "Nimda"
Do Int Delete _$NimdaEvent_%c_ip%
SCRIPT_ACTION
; Define the Group: Nimda Check. This marks the end of the
; Default Group and start of a new named Group. This Group
; processes events when Nimda is detected.
Group Nimda Check
; Define a new Rule in the Nimda Check Group. This Rule
; checks the existence of the dynamic variables and
; increments the appropriate client counter to mark that
; Nimda virus activity is detected from a Client. Note that
; the detected event is ignored after incrementing the
; counter.
Rule Nimda Count Increment
Include Int
          _$ClientIP_%c_ip% exists,
          _$ClientIP_%c_ip%_Count exists
Do Int Incr _$ClientIP_%c_ip%_Count
```

```
; Define a new Rule to process Nimda activity from a client
; for the first time. This Rule is triggered only when an
; event is detected from a client for the first time. The
; event initializes a counter variable and a timer
; variable. Since this is the first event from this client
; the Rule generates an internal event to send an alert.
Rule Nimda New Client
Exclude Int _$ClientIP_%c_ip% exists
Exclude Int _$ClientIP_%c_ip%_Count exists
Do Int Define _$ClientIP_%c_ip%_Count Value(1) ExpireIn(86400)
Do Int Define _$ClientIP_%c_ip% Value(1) ExpireIn(600)
Do Int Define _$NimdaEvent_%c_ip%
Do Int Set _$NimdaEvent_%c_ip% Src ("Nimda")
Do Int NewEvent _$NimdaEvent_%c_ip%
; Define a new Rule to process Nimda activity from a client
; for which we have seen Nimda activity event before.
; This Rule is triggered only when the
; event detected from a client is not for the first time
; and 10 minutes (600 seconds) have elapsed since the
; first event from this client was detected. The Rule
; re-initializes the timer variable and generates an
; internal event with the number of Nimda activity
; events detected from this client.
Rule Nimda Old Client

    Exclude Int _$ClientIP_%c_ip% exists

    Include Int _$ClientIP_%c_ip%_Count exists

    Do Int Define _$ClientIP_%c_ip% Value(1) ExpireIn(600)

    Do Int Define _$NimdaEvent_%c_ip%

    Do Int Set _$NimdaEvent_%c_ip% Src ("Nimda")

    Do Int Set _$NimdaEvent_%c_ip% Count(_$ClientIP_%c_ip%_Count)

    Do Int Set _$NimdaEvent_%c_ip% Info("Count contains the number of times a similar event was suppressed in the last 15 minutes")

    Do Int Set _$ClientIP_%c_ip%_Count Value(1)

    Do Int NewEvent _$NimdaEvent_%c_ip%
```

Note:	_$CL_IP	->_$ClientIP_%c_ip%
	_$CL_IP_COUNT	->_$ClientIP_%c_ip%_Count
	_$NE_IP	->_$NimdaEvent_%c_ip%

Event Flow Diagram


Rule Nimda Detect
Rule Nimda Count Increment
EVENT
Src == "Nimda"
NO
Detect Nimda
YES
_$CL_IP Exists OR _$CL_IP_COUNT Exists
YES
Increment _$CL_IP_COUNT
YES
NO
NO
Rule Nimda Alert
_$NE_IP Exists AN Src == "Nimda"
YES
Delete _$NE_IP
SCRIPT_ACTION
Rule Nimda New Client
_$CL_IP Exists OR _$CL_IP_COUNT Exists
YES
NO
Define _$CL_IP_COUNT, Set Value 1, Exp 600
Define _$NE_IP, Set Event Fields Generate Event
_$CL_IP Exists
YES
No Further Action
Rule Nimda Old Client
NO
Define _$CL_IP, Set Value 1, Exp 600
YES
_$CL_IP_COUNT Exists
NO
No Further Action
Define _$NE_IP, Set Event Fields Generate Event Reset _$CL_IP_COUNT

Chapter 6: Encryption Options

This section contains the following topics:

Introduction

eTrust Audit r8 supports AES and DES encryption and uses various encryption methods for different purposes. Outgoing messages are encrypted using a single preferred encryption method, while incoming messages are decrypted according to the encryption method by which they were encrypted. It is possible to configure the outgoing encryption method during installation, and it is possible to configure all encryption settings after installation.

During installation, the user may configure the initial outgoing encryption method using the appropriate dialog, which appears for every Audit installation package. If the installation is done on a clean machine, it prompts the user to use AES 256 bit, AES 128-bit, or DES 56-bit. If the installation is upgrading from an existing product that uses an encryption that is different from AES, it prompts the user to either keep any existing encryption, or switch to AES. If the existing product is already using AES, the installation prompts the user to stay with AES to switch back to DES.

By default, the information Audit sends from station to station is encrypted using 56-bit DES encryption. You can change your encryption key, switch to a different encryption cipher, or turn off encryption. Whatever you do about encryption, you should do the same thing at every station where Audit is installed.

Note: Audit r8 loads both encryption types for compatibility. The encryption method setup from install only applies to outgoing events. Either encryption can be used when decrypting incoming events. Un-encrypted information is accepted from all sources, regardless of the encryption setting.

Managing Encyption on Windows

After installation, you can configure the encryption settings using the Registry Editor. The following registry key holds the encryption settings:

HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrust Audit\Encryption

The Ciphers registry value in this key holds a comma-separated list of cipher names or paths. All ciphers in the list are used to decrypt incoming messages. Each element in the list can be a name of a cipher DLL with or without a full path. The '.dll' extension need not be added to any element. If a fully-qualified path is specified, eTrust Audit uses the cipher from that path.

Otherwise, the specified path is considered relative to the eTrust Audit DLL path on that machine, which is the \bin directory under the eTrust Audit installation directory, by default. In addition, the *install_dir*\bin\adcipher.dll file is the cipher used for outgoing encryption.

To change the preferred ciphers for decryption of incoming messages

1. Modify the contents of the Ciphers registry value.
2. Restart all eTrust Audit services and applications, including iGateway.

To change the preferred cipher for encryption of outgoing messages

1. Stop all eTrust Audit services including iGateway and close all eTrust Audit applications.
2. Overwrite the *install_dir*\bin\adcipher.dll file with the new cipher library.
3. Restart the services and applications.

Managing Encryption on UNIX

After installation, it is possible to configure the encryption settings by editing configuration file eaudit.ini. The following entry holds the settings:

```
Encryption
{
  Ciphers =
}
```

The Ciphers contains a comma separated list of cipher names or paths. The cipher adcipher.so (.sl for HPUX, and .o for AIX) directory /usr/lib is a symbolic link to shared library in the eTrust Audit dll directory and used to encrypt outgoing messages. All ciphers in the list are used to decrypt incoming messages. Each element in the list can be a name of a cipher shared library with or without a full path. The library extension need not be added to any element. If a fully-qualified path is specified, eTrust Audit will use the cipher from that path. Otherwise, the specified path is considered relative to the eTrust Audit dll path on that machine, which is the lib directory under the installation directory of eTrust Audit by default.

To change the preferred Ciphers for decryption of incoming messages:

1. Modify the contents of Ciphers configuration value.
2. Restart all eTrust Audit daemons including iGateway.

To change the preferred Ciphers for encryption of outgoing messages:

1. Modify soft link /usr/lib/adcipher.so (or .sl, or .o).
2. Restart all eTrust Audit daemons including iGateway.

Changing Your Encryption Key

You can change the encryption key at any time, and you can switch back to the default key at any time. But whenever you change the key at any station, you must make the same change at all stations.

Note: You must make the encryption change manually at each station. There is no method to automatically distribute the change to each station in your eTrust Audit environment.

eTrust Audit generates new keys using the MD5 hashing function. They can be based on a file or string of any size.

To change the encryption key, follow these steps:

1. Stop the eTrust Audit services and Security Monitor, if installed.
2. From the command line, use the **setkey** utility.
   - On Windows systems: setkey is located in the *install_dir*\bin directory (where *install_dir* is the directory in which you installed eTrust Audit).
   - On UNIX systems: setkey is located in the *install_dir*/bin directory.
3. Restart the services and Security Monitor.

Note: After you run **setkey** , you should also run the encup utility to re-encrypt all encrypted values (username/password).

setkey Command Options

You can use the following options for the setkey command:

-c

Clears the user key and sets a default key.

-f[e] filename

Specifies the contents of filename as the basis for the new encryption key. If the file is not in the current directory, you can include an absolute or relative pathname.

If you use -fe, the file is then deleted. If you use -f, the file remains.

-help

Displays these syntax options.

-k newkey

Installs newkey as the basis for the new encryption key.

Turning Off Encryption

Perform these steps to turn encryption on or off according to your operating system.

To turn encryption off in Windows:

- Delete the *install_dir*\bin\adcipher.dll file.

To turn encryption on in Windows:

- Copy the file, *install_dir*\bin\Des56bit.dll to *install_dir*\bin\adcipher.dll.

To turn encryption off in UNIX:

- Delete the /usr/lib/adcipher.so file.

To turn encryption on in UNIX:

- Create link/usr/lib/adcipher.si file to install_dir/bin/Des56bit library.

Note: The username/password is encrypted using strong encryption. After you delete the adcipher.dll, you must run the encup utility again to encrypt the username/password using an internal *wick* encryption.

Chapter 7: Windows Registry Entries

This section contains the following topics:

Introduction

The Windows registry entries for eTrust Audit control many facets of how the software operates on Windows systems. The keys described in this section are located under the following registry entry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit
```

Opening the Windows Registry

The Windows registry contains key that control various features in eTrust Audit. The root level key is as follows:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit
```

To open the Windows registry to view or modify its contents, follow these steps:

Open a command prompt session.

1. Enter the regedit or regedt32 command.
2. Expand the tree items for the HKEY_LOCAL_MACHINE, SOFTWARE, ComputerAssociates, and finally the eTrust Audit branch to view the registry keys described in the topics that follow.

   Note: The topics that follow describe only those key values that you can modify.

Ports

eTrust Audit maintains information about the ports it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Ports

Normally, eTrust Audit uses one of its default ports or uses portmapper to dynamically assign a port. eTrust Audit uses the values of these keys under the following conditions:

- The default port is busy
- The service cannot get the dynamic port from the portmapper

Under normal circumstances, you would not have any reason to modify these values. However, if a port is being used by another application or service or you need to route events through a firewall, you must modify the values for these keys.

The key values are as follows:

MonitorPort

The data value specified for the MonitorPort key is used by the Action Manager to route events to the Security Monitor and by the Security Monitor to receive events.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

RouterPort

The data value specified for the RouterPort key is used by the router and redirector.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

RouterSapiPort

The data value specified for the RouterSapiPort key is used by the UNIX Recorder, the Recorder, the Generic NT Recorder, the Check Point Firewall-1 Recorder, and applications that use SAPI, and is used by the router.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

MonitorSapiPort

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

CollectorSapiPort

The data value specified for the CollectorPort key is used by the Action Manager to route events to the Collector and by the Collector to receive events using SAPI protocol.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

CollectorPort

The data value specified for the CollectorPort key is used by the Action Manager to route events to the Collector and by the Collector to receive events.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by portmapper.

DistributionPort

The data value specified for the DistributionPort key is used by the Distribution Agent to receive policies and MP files from the Distribution Server..

Type

String Value, TCP/IP, bi-directional

Data

Specify the number of the port to be used. The default is 8025, which is an official IANA-registered TCP port.

DistributionControlPort (r8SP2)

The data value specified for the DistributionControlPort key is used by the distribution server to receive end-point status from the distribution agent.

Type

String Value, TCP, bi-directional

Data

Specify the number of the port to be used. The default is 8026, which is an official IANA-registered TCP port.

SNMPRecorderPort

The data value specified for the SNMPRecorderPort key is used by the SNMP recorder.

Type

String Value, one-directional (incoming)

Data

Specify the number of the port to be used. The default is 162.

SNMPTrapPort

The data value specified for the SNMPTrapPort key is used by the Action Manager to route actions defined as Action SNMP to the router.

Type

String Value, one-directional (outgoing)

Data

Specify the number of the port to be used. The default is 162.

Note: The Windows SNMP service also uses port 162. If you need to use the SNMP recorder, you must disable the Windows SNMP service or assign another data value for the SNMPRecorderPort and SNMPTrapPort keys.

RPC

eTrust Audit maintains information about the name of the program used to map ports on the system. It uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\RPC

Under normal circumstances, you would not have any reason to modify these values. If you are using a different program to map ports other than portmap, you must change the data value.

The key values are as follows:

PortmapName

The data value specified for the PortmapName key is used to identify the name of the program used to map RPC ports.

Type

String Value

Data

Specify the name of the RPC port map program. The default is portmap.exe. If you do not know the program name, leave this value empty.

Messages

eTrust Audit maintains information about the name of the file where it stores messages under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Messages

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MessageFile

The data value specified for the MessageFile key is used to identify the name and location of the message file.

Type

String Value

Data

Specify the name of the message file, including its full path. The default is *install_dir*\Messages\message.txt (/Messages/message.txt on Unix)

Severity

eTrust Audit maintains information about the name of the targets where messages are to be sent under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociatesAudit\Messages\Severity

Under normal circumstances, you would not have any reason to modify these values.

The key values are described in the topics that follow.

Fatal

Targets

The data value specified for the Targets key is used to identify the targets where fatal messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Critical

Targets

The data value specified for the Targets key is used to identify the targets where critical messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Error

Targets

The data value specified for the Targets key is used to identify the targets where error messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Warning

Targets

The data value specified for the Targets key is used to identify the targets where warning messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Info

Targets

The data value specified for the Targets key is used to identify the targets where info messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Targets

eTrust Audit maintains information about targets under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Messages\Targets

The key values are described in the topics that follow.

Monitor

eTrust Audit maintains information about the self-monitor target to use to send its own notification messages under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Messages\Targets\Monitor

Host

The data value specified for the Host key is used to identify the host where messages are to be sent.

Type

String Value

Data

Specify the name of the host. The default value is localhost.

MonitorPort

The data value specified for the MonitorPort key is used to identify the port used by the Security Monitor.

Type

String Value

Data

Specify the number of the port. By default, the port is dynamically assigned by portmapper.

Mail

eTrust Audit maintains information about the mail server to use to send email under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Mail

If you specify the name of the mail server at installation, you would not have any reason to modify these values, unless you wanted to change the name of the mail server or change the name of the user sending the mail.

The key values are as follows:

ServerType

The data value specified for the ServerType key is used to identify the type of mail server.

Type

String Value

Data

Specify the name of the type of mail server. The default is SMTP. You cannot change this value.

MailServer

The data value specified for the MailServer key is used to identify the host name of the mail server.

Type

String Value

Data

Specify the name of the mail server. The default is mailsrv or the name you specified at installation time.

Sender

The data value specified for the Sender key is the mail address of the account from which mail is sent.

Type

String Value

Data

Specify the name of the sender from which mail is sent. The default value is Administrator. For certain SMTP servers, the value of Sender must represent an existing mail account, with the format name@domain.

Client\SeOS\logmgr

eTrust Audit maintains information about the audit and error log files for eTrust Audit under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\SeOS\logmgr

Under normal circumstances, you would not have any reason to modify these values. However, there might be times when you must increase the value of the audit_size parameter, such as during periods of peak use.

The key values are as follows:

audit_back

The data value specified for the audit_back key is used to identify the name of the backup file for the local audit file. When the local audit file reaches the size specified by the audit_size parameter, it is given this name and the old file with this name is discarded.

Type

String Value

Data

Specify the name of the audit backup file, including path. The default is install_dir\dat\log\seos_audit.bak.

audit_log

The data value specified for the audit_log key is used to identify the name of the local audit file. The recorder service writes to the file named here, and the redirector service reads from it.

Type

String Value

Data

Specify the name of the local audit file, including path. The default is install_dir\dat\log\seos.audit.

audit_size

The data value specified for the audit_size key is used to identify the maximum size, in KB, for the local audit file.

Type

DWORD Value

Data

Specify the size of the local audit file. The default is 3000, for 3000 KB.

error_back

The data value specified for the error_back key is used to identify the name of a file used internally by eTrust Audit.

Type

String Value

Data

Specify the name of the error log backup file, including path. The default is install_dir\dat\log\seos_error.bak.

error_log

The data value specified for the error_log key is used to identify the name of a file used internally by eTrust Audit.

Type

String Value

Data

Specify the name of the error log file, including path. The default is install_dir\dat\log\seos.error.

Recorders

eTrust Audit maintains information about the audit and error log files for eTrust Audit under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Recorders

By default, the recorder sends messages to the router on the system where it is installed. However, as you begin to deploy eTrust Audit throughout your enterprise, you can change this value to send events to dedicated routers. You identify these dedicated router systems by changing the value of DefaultRouter from localhost to the host name or IP address of the dedicated router system.

The key values are as follows:

DefaultRouter

The data value specified for the DefaultRouter key is used to identify the host name or IP address of the computer that runs the eTrust Audit Router.

Type

String Value

Data

Specify the name of the host that runs the eTrust Audit Router. The default is localhost.

NT Recorder

eTrust Audit maintains information about the files used by the recorder under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Recorders\NT Recorder

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

DataFile

The data value specified for the DataFile key is used to identify the name of the file used by the recorder internally.

Type

String Value

Data

Specify the name of the file, including path. The default is install_dir\dat\recorders\selogrec.dat. You should not change this location.

FilterFile

The data value specified for the FilterFile key is used to identify the name of the recorder configuration file.

Type

String Value

Data

Specify the name of the recorder configuration file, including path. The default is install_dir\dat\recorders\selogrec.cfg.

SearchStringsFile

The data value specified for the SearchStringsFile key is used to identify the name of a file that the recorder service uses internally.

Type

String Value

Data

Specify the name of the search strings file, including path. The default is install_dir\dat\recorders\selogrec.str. You should not change this location.

SkipImportLogs

The data value specified for the SkipImportLogs key is used to identify whether to import earlier Windows NT audit logs.

Type

DWORD Value

Data

This value is generated during setup. Specify 1 or 0. When set to 1, the recorder will start to send only new events.

Interval

The data value specified for the Interval key is used to identify the time the recorder service suspends (sleeps) without writing any data from the event log.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 10 seconds. This value is optional.

MaxSeqNoSleep

The data value specified for the MaxSeqNoSleep key is used to identify the maximum number of records written before sleeping.

Type

DWORD Value

Data

The maximum number of records before sleeping. The default value is 50. This value is optional.

SNMP Recorder

eTrust Audit maintains information about the mapping file used by the SNMP recorder to parse events under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Recorders\SNMP Recorder
```

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MPFile

The data value specified for the MPFile key is used to identify the name of the mapping file used by the SNMP recorder to parse events.

Type

String Value

Data

Specify the name of the mapping file, including path. The default is install_dir\cfg\snmptd_rec.mp.

Redirector

eTrust Audit maintains information used by the redirector under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Redirector

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

DataFile

The data value specified for the DataFile key is used to identify the name of a file used by the redirector internally.

Type

String Value

Data

Specify the name of the internal file used by the redirector, including path. The default is install_dir\dat\logroute.dat. You should not change this location.

MailSubject

The data value specified for the MailSubject key is used to identify the subject line for eTrust Audit outgoing email.

Type

String Value

Data

Specify the subject line of an email sent by eTrust Audit. The default is Notification from eTrust Audit.

RouteFile

The data value specified for the RouteFile key is used to identify the name of the redirector configuration file.

Type

String Value

Data

Specify the name of the redirector configuration file. The default value is install_dir\etc\ logroute.cfg.

SendTimeout

The data value specified for the SendTimeout key is used to identify the time the redirector waits for confirmation from the router before resending a message. If the timeout period is too short, the same message might appear in the database several times.

Type

DWORD Value

Data

Specify the time in seconds the redirector waits for confirmation from the router before sending a message. The default value is 25 seconds. Setting this value is optional.

Interval

The data value specified for the Interval key is used to identify the time the redirector service sleeps without writing any data from the event log.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 5 seconds. Setting this value is optional.

MaxSeqNoSleep

The data value specified for the MaxSeqNoSleep key is used to identify the maximum number of records sent before sleeping.

Type

DWORD Value

Data

The maximum number of records sent before sleeping. The default value is 50. Setting this value is optional.

SpeedBackup

The data value specified for the SpeedBackup key affects the values of Interval and MaxSeqNoSleep, previously mentioned. This value affects only if the Redirector reads from the eTrust Audit backup file. The value of MaxSeqNoSleep is multiplied by the value of SpeedBackup to give an effective value. The value of Interval is divided by the value of SpeedBackup to give an effective value. The effective value has a set minimum of 1 second.

Type

DWORD Value

Data

The default value is 2. Setting this value is optional.

ChangeLogFactor

The data value specified for the ChangeLogFactor key is used to identify the number of sleep periods before retrying failed targets.

Type

DWORD Value

Data

The number of sleep periods before the redirector retries failed targets. The default value is 3. Setting this value is optional.

SavePeriod

The data value specified for the SavePeriod key is used to identify the time before the current position of the redirector service in seos.audit is stored in logroute.dat.

Type

DWORD Value

Data

The time in minutes before the current position of the redirector service in seos_audit is stored in logroute.dat. The default value is 10 minutes. Setting this value is optional.

OverWriteBackup

The data value specified for the OverWriteBackup key is used to identify whether the redirector closes the backup file during sleep periods so that it can be erased.

Type

DWORD Value

Data

Specify 1 or 0. When set to 1, the redirector service closes the backup file during sleep periods, allowing it to be erased.

Router

eTrust Audit maintains information used by the router under the following keys:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\AllowRemoteProgram

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\AllowRemoteFile

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

RulesDirectory

The data value specified for the RulesDirectory key is used to identify the directory where routers configuration files are located.

Type

String Value

Data

Specify the name of the directory where the router configuration files are located. The default is install_dir\cfg\.

RulesExtension

The data value specified for the RulesExtension key is used to identify the extension for router configuration files.

Type

String Value

Data

Specify the extension for router configuration files. The default is .cfg. Setting this value is optional.

AllowRemoteProgram

Discards events contained in request to execute action 'remote program'.

Type

String Value

Data

This parameter is optional. Value is 0 or 1. Default value is 0.

AllowRemoteFile

Discards events contained in request to execute action 'remote file'.

Type

String Value

Data

This parameter is optional. Value is 0 or 1. Default value is 0.

Router\Queue Manager\Queues

eTrust Audit maintains information about the queues used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

DirectoryName

The data value specified for DirectoryName is used to identify the directory where queues are located.

Type

String Value

Data

Specify the name of the directory where the queues are located. The default is install_dir\dat\Queue\route (or /dat/Queue/route on Unix).

AlertQueue Rules

eTrust Audit maintains information about the alert queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\AlertQueue\Queue Rules

Note: The rule name (value name) is unimportant, so you can change it. The Data section indicates which action and which target the action reaches to be performed from this queue. In case the target is not indicated, it means that only the action is of importance.

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

monitor

The data value specified for the monitor key is used to identify the name of the action and target, separated by a semicolon.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "monitor; "

snmp

The data value specified for the snmp key is used to identify the name of the action and target, separated by a semicolon.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "snmp; "

screen

The data value specified for the screen key is used to identify the name of the action and target, separated by a semicolon.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "screen; "

AlertQueue Parameters

eTrust Audit maintains information about the alert queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\AlertQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

SwitchTimeout

The data value specified for SwitchTimeout is used for action “collector” and “monitor” cases to define alternate destinations.

Type

DWORD Value

Data

The default value is 2 hours.

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

CollectionQueue Rules

eTrust Audit maintains information about the collection queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\CollectionQueue\Queue Rules

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

Name

The data value specified for the Name key is used to identify the name of the collector.

Type

String Value

Data

Specify the name of the collector. The default value is “collector;”. There is no reason to change this value.

CollectionQueue Parameters

eTrust Audit maintains information about the collection queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\CollectionQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

DefaultQueue Rules

eTrust Audit maintains information about the collection queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\Default\Queue Rules

Under normal circumstances, you would not have any reason to modify these values.

The default key has no key rules; it gets all the rules of the other keys.

DefaultQueue Parameters

eTrust Audit maintains information about the collection queue rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Queues\Default\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

Router Queue Actions

eTrust Audit maintains information about the actions specified rules used by the router under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Router\Queue Manager\Actions

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

file

The file action routes events to a file in ASCII text format. It has no parameters you should change.

monitor

The monitor action routes events to the security monitor. It has no parameters you should change.

collector

The collector action routes events to the collector database. It has no parameters you should change.

mail

The mail action routes messages to a designated SMTP mail server and onto an email address.

The mail parameters are as follows:

MailSubject

The data value specified for the MailSubject key is used to identify the subject line for eTrust Audit mail. Specify the text you want to appear in the subject line of email sent by eTrust Audit. The default is "Notification from eTrust Audit."

screen

The screen action routes events to an NT screen session.

remote

The remote action routes events to an action manager on the host named in the action where it is executed without filtering.

route

The route action sends events to the host named in the action where it reviewed by the router on that system and executed according to any filters that apply on that system.

snmp

The snmp action sends SNMP traps to the host named in the action.

program

The program action executes a command on the host named in the action on the local host.

unicenter

The unicenter action routes events to the Event Management Console on the host named in the action.

The key values are as follows:

UnicenterHome

The data value specified for the UnicenterHome key is used to identify the location of the Unicenter installation. Specify the location of the Unicenter installation.

Management Agent

eTrust Audit maintains information about the which systems are trusted policy servers and parameters related to policy distribution under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent

When you install eTrust Audit, you identify the name of a trusted policy server. By changing the value of the TrustedServers key, you can add more servers to identify other policy servers.

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

TrustedServers

The data value specified for the TrustedServers key is used to identify one or more policy servers.

Type

String Value

Data

Specify the host names or IP addresses of one or more policy servers, separated by commas.

Parameters

eTrust Audit maintains information about the how policy management runs under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\Parameters

Under normal circumstances, you would not have any reason to modify these values.

All the following keys are optional:

AuditStatusDir

The data value specified for the AuditStatusDir key is used to identify the directory where end-point status files are located.

Type

String Value

Data

Specify the directory name. The default value is install_dir\dat\status.

iGatewayStatusDir

The data value specified for the iGatewayStatusDir key is used to identify the directory where iRecorder status files are located.

Type

String Value

Data

Specify the directory name. The default value is nodestatus, which is located under <igateway_install_dir>. Value can be absolute path or relative path. The relative path starts from the <igateway_install_dir>

TrustedServers

Specify the host names or IP addresses of one or more policy servers, separated by commas.

TmpPolicyDir

The data value specified for the TmpPolicyDir key is used to identify the directory where temporary policy files are stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir\dat\tmp\agent_tmp_policies (or /dat/tmp/agent_tmp_policies on Unix).

ConnectionTimeout

The data value specified for the ConnectionTimeout key is used to identify the number of seconds after which a connection between a policy server and distribution agent is closed.

Type

DWORD Value

Data

Specify the number of seconds after which the connection is broken. The default value is 600 seconds.

ReceiveTimeout

The data value specified for the ReceiveTimeout key is used to identify an internal parameter for the TCP session.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 10 seconds.

SendTimeout

The data value specified for the SendTimeout key is used to identify an internal parameter for the TCP session.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 10 seconds.

DistributionTimeout

The data value specified for the DistributionTimeout key is used to identify the time from the start of the TCP session until the agent receives the policy.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 800 seconds.

AN Types

eTrust Audit maintains information about the types of event logs defined to it under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types

Under normal circumstances, you would not have any reason to modify these values through the registry.

Note: All the following event log sources have a parameters section that contains no values.

Apache

eTrust Audit maintains information about the library used by the Apache AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\Apache

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process Apache events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

Default

eTrust Audit maintains information about the library used by the Default AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\Default

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process Default events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

eTrust Access Control

eTrust Audit maintains information about the library used by the eTrust Access Control AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\eTrust Access Control

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process eTrust Access Control events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

Netscape

eTrust Audit maintains information about the library used by the Netscape AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\Netscape

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process Netscape events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

NT

eTrust Audit maintains information about the library used by the NT AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\NT

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process NT events.

Type

String Value

Data

Specify the name of the library. The default value is TALR.

Oracle

eTrust Audit maintains information about the library used by the Oracle AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\Oracle

Under normal circumstances, you would not have any reason to modify these values through the registry.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process Oracle events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

UNIX

eTrust Audit maintains information about the library used by the UNIX AN type under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Client\Management Agent\AN Types\UNIX

Under normal circumstances, you would not have any reason to modify these values through the registry. You define and modify these using the Policy Manager GUI.

LibraryName

The data value specified for the LibraryName key is used to identify the library used to process UNIX events.

Type

String Value

Data

Specify the name of the library. The default value is TGNR.

Policy Manager

The keys in the topics that follow apply to the Policy Manager.

In order to distribute policies (or Message Parsing files in r8SP2), the Policy Manager must extract data from the database, compile, and place the compiled objects in a staging area where the Distribution Server retrieves them for distribution to the eTrust Audit Clients. The location of this staging area is stored in the following value:

PolicyDir

Specify the directory of the staging area where compiled policies (and MPs in r8SP2) are stored.

Database

eTrust Audit maintains information about the database used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Database

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

DSN

The data value specified for the DSN key is used to identify the name of the data source.

Type

String Value

Data

Specify the data source name. The default value is eAuditPMDB.

UserName

The data value specified for the UserName key is used to identify the name of the user under whose name changes can be made to the database.

Type

Binary Value

Data

Specify the user name. The value is encrypted.

Password

The data value specified for the Password key is used to identify the password of the user under whose name changes can be made to the database.

Type

Binary Value

Data

Specify the password. The value is encrypted.

Distribution Log

eTrust Audit maintains information about the log file used to store messages about the success or failure of policy distribution used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Log

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxLogSize

The data value specified for the MaxLogSize key is used to identify the number of records to be stored in the log.

Type

DWORD Value

Data

Specify the number of records. The default value is 10000.

MaxTimeOut

The data value specified for the MaxTimeOut key is used to identify the maximum time (in seconds) the distribution server waits to write to the database. After this period ends without success, an error is recorded in the machine event log.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 60.

DelPartSize

The data value specified for the DelPartSize key is used to identify the number of records to erase when the value of MaxLogSize is reached.

Type

DWORD Value

Data

Specify the number of records to be erased. The default is 500.

Distribution Server

eTrust Audit maintains information about the output directory used by the distribution server under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server

Under normal circumstances, you would not have any reason to modify these values.

The key values are described in the topics that follow.

Distribution Server Threads

The eTrust Audit Distribution Server uses threads to push a policy to audit nodes (ANs). eTrust Audit maintains information about the number of parallel distributions of the same policy to eTrust Audit Clients under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server
"MaxThreadNumber"=dword:00000040
```

Note: Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxThreadNumber

(Configurable through the Audit Administrator user interface.) Defines the number of parallel distributions of the same policy to Clients.

Type

DWORD Value

Data

Specify the number of threads for parallel distribution of the same policy.

Default: 10

Limits: 64 (MAXIMUM_WAIT_OBJECTS)

Other key values related to the distribution process tuning are following:

ActivePolling

(Configurable through the Audit Administrator user interface.) Defines whether Policy Manager should directly poll Clients to request end-point status.

Type

DWORD Value

Data

0 or 1. 0 means no polling; the Client sends status if there is any change in the status (new policies or MPs, deleted policies or MPs, tampered policies or MPs, …). 1 means active polling.

Default: 0

NodeAutoCorrect

(Configurable through Audit Admin UI). Defines whether Policy Manager should automatically send the correct policies or MPs to an Audit Client if the latter’s end-point status does not correspond to its state stored in Policy Manager database.

Type

DWORD Value

Data

0 or 1. 0 means no autocorrect. 1 means autocorrect if there is discrepancy between the Client's end-point status and Policy Manager information for this Client.

Default: 0

MaxDistRetryNumber

(Configurable through the Audit Administrator user interface.) Defines the maximum number of retries to distribute.

Type

DWORD Value

Data

Specify the maximum number of distribution retries.

Default: 3

Distribution Server\Queue Manager\Queues

eTrust Audit maintains information about the queues used by the distribution server under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution
Server\Queue Manager\Queues
```

Under normal circumstances, you should not have any reason to modify these values.

The key values are as follows:

DirectoryName

The data value specified for DirectoryName is used to identify the directory where queues are located.

Type

String Value

Data

Specify the name of the directory where the queues are located. The default is *install_dir*\dat\Queue\distrib (or *install_dir*/dat/Queue/distrib on UNIX).

DistributionQueue Rules

eTrust Audit maintains information about the queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\DistributionQueue\Queue Rules

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

distribute

The data value specified for the distribute key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "distribute; "

remove

The data value specified for the remove key is used to identify the name of the action and target.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "remove; "

DistributionQueue Parameters

eTrust Audit maintains information about the distribution queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\DistributionQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the file in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 100 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the distribution server operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the distribution server operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the distribution server operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the distribution server operates in the queue before moving to another queue. The default value is 50 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the distribution server service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 10 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a policy again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 1800 seconds (30 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a policy can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a policy can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

PollingQueue Rules

eTrust Audit maintains information about the queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\PollingQueue\Queue Rules

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

add

The data value specified for the add key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *add;*.

change

The data value specified for the change key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *change;*.

delete

The data value specified for the delete key is used to identify the name of the action and target.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *delete;*.

PollingQueue Parameters

eTrust Audit maintains information about the default queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\PollingQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 50.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 1000 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 50 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 10 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 86400 seconds (24 hours).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 0.

Distribution Server Queue Actions

eTrust Audit maintains information about the actions specified rules used by the Distribution Server under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Actions

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

add

The add action adds a new audit node to the polling queue.

change

The change action changes an audit node in the polling queue.

delete

The delete action removes an audit node from the polling queue.

distribute

The distribute action routes policies to distribution agents.

remove

The remove action removes policies from the distribution agents.

NT-Auditing Policy

By default, the Audit tab in the NT Policy Properties dialog is available to define auditing policy on the Windows clients. You can disable the Audit option by setting the following optional DWORD value in the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Options:
AuditingPolicy = 0
```

When disabled, the Audit settings will not appear in Policy Manager and will not be distributed; prior definitions are ignored.

Data Server

The keys in the topics that follow apply to the Data Server.

Collector

eTrust Audit maintains information about the Collector used by the data server under the following key:

HKLM\Software\ComputerAssociates\eTrust Audit\Data Server\Collector

The key values are as follows:

MaxBulkInsertRows

Type

DWORD Value

Data

The maximum number of rows for bulk insert. Default: 100.

MaxBulkBufferSize

Type

DWORD Value

Data

The maximum size of the bulk insert buffer (in bytes). This parameter is reserved for future use.

BulkInsertCutoffTime

Type

DWORD Value

Data

The maximum number of seconds to wait for buffer to reach either MaxBulkInsertRows or MaxBulkBufferSize. Default: 5 seconds.

MaxDBSessions

Type

DWORD Value

Data

The maximum number of connections to open on the database. Default: 10.

Database

eTrust Audit maintains information about the database used by the data server under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Data Server\Database

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

AuditDSN

The data value specified for the AuditDSN key is used to identify the name of the data source for the database used by the Data Tools components. This values is used by the Collector service and by the Viewer and Reporter as the default database.

Type

String Value

Data

Specify the data source name. The default value is eAudit_DSN.

Note: To switch to a different database, access the ODBC Data Sources applet in the Windows NT Control Panel (or the Administrative Tools in the Control Panel, in Windows 2000 and XP) to set up a new database with the same DSN. If you want to start a new database with a new DSN, you need to match this value to it.

DSNList

The data value specified for the DSNList key is used to identify the another system DSNs for the databases used by the Viewer and the Reporter.

Type

String Value

Data

Specify the data source names, separated by commas. The default value is eAudit_DSN.

UserName

The data value specified for the UserName key is used to identify the name of the user under whose name connection can be made to the database.

Type

Binary Value

Data

Specify the user name. The value is encrypted. If no value is specified when the collector service or the Viewer starts, it is requested.

Password

The data value specified for the Password key is used to identify the password of the user under whose name connection can be made to the database.

Type

Binary Value

Data

Specify the password. The value is encrypted. If no value is specified when the collector service or the Viewer starts, it is requested.

Note: You can change the user name and the password using the Encup utility.

EntryIDRange

Type

DWORD Value

Data

The range of entryIDs to cache in the Collector. The greater the range, the fewer attempts made to the database to ask for the next entryID (stored in table 'highval'). The smaller the range, the smaller the gap in entryIDs if the collector stops. Default: 5000.

Viewer

eTrust Audit maintains information about the Viewer under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Data Server\Viewer

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

FiltersDir

The data value specified for the FiltersDir key is used to identify the location where the filter definition files are stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir\dat\filters\.

IniFile

The data value specified for the IniFile key is used to identify the location where the ini file is stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir\ini\SeAuditW.ini.

Reports

eTrust Audit maintains information about reports under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Data Server\Reports

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

ReportsDir

The data value specified for the ReportsDir key is used to identify the location where reports are stored.

Type

String Value

Data

Specify the directory. The default value is install_dir\dat\reports.

ReadyReportsDir

The data value specified for the ReadyReportsDir key is used to identify the location where saved reports are stored.

Type

String Value

Data

Specify the directory. The default value is Saved\.

TemplatesDir

The data value specified for the TemplatesDir key is used to identify the location where the report templates are stored.

Type

String Value

Data

Specify the directory. The default value is Templates\.

MailSubject

The data value specified for the MailSubject key is used to identify the subject line of email notifications about report completion.

Type

String Value

Data

Specify the text for the subject line. The default value is “Notification from eTrust Audit Report Generator.”

MailBody

The data value specified for the MailBody key is used to identify the body text in email notifications about report completion.

Type

String Value

Data

Specify the body text. The default value is “Report has been created successfully. You can view the report using the eTrust Audit Reporter.”

DataSourcesFile

Type

String Value

Data

Specify the location of the file containing access information to different Collector databases. The location can be specified relative to the eTrust Audit install_dir or as an absolute path. Default: etc\RVConfiguration.xml.

JobLoggersFile

Type

String Value

Data

Specify the location of the file containing the Reporter’s Java logger to log Reporter activities. The location can be specified relative to the eTrust Audit install_dir or as an absolute path. Default: etc\JobLoggers.xml.

Security Monitor

eTrust Audit maintains information about the Security Monitor under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Monitors\Security Monitor

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

EventData

The data value specified for the EventData key is used to identify the name of a file to which the currently displayed events are written each time you close the Security Monitor. When you next open the Security Monitor, the contents of the file are displayed and new events are added.

Type

String Value

Data

Specify the file name, including path. The default value is install_dir\etc\events.data.

IniFile

The data value specified for the IniFile key is used to identify the location where the ini file is stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir\ini\SecMonW.ini.

Chapter 8: UNIX INI Files

This section contains the following topics:

UNIX INI and Configuration Files Introduction

The Client components on UNIX systems are controlled by entries in the following .ini files:

- eaudit.ini
- recorder.ini

The files are located in eTrustAudit_root/ini/, where eTrustAudit_root is the directory where you installed eTrust Audit.

The eTrust Audit Router reads the .cfg (policy) files that contain filters that are made up of rules, and actions and targets. Using these rule (policies) the log router, aclogrd, filters the forwarded events and discards some of them.

The .cfg files are located in directories that are specific to the operating system type. For Windows systems, the default directory is \eTrust Audit\cfg. For UNIX systems, the default directory is /opt/CA/eTrustAudit/cfg.

Performing Available Actions on Remote Servers

You can configure available actions so that they are performed on a remote server, using the options in the group box provided on each action's properties panel. The remote action tells the Action Manager to move records from a queue to a remote router and performs any action on this remote host.

To create a remote action

1. Access the Policy Manager and select or create a policy.
2. Access an existing rule or create a new rule.
3. Navigate through the Rule Wizard or Edit Rule dialogs until you can edit or define an action.

   A list of actions displays in the Browse Actions pane.
4. Select an action from the list.

   A properties dialog for that action displays.
5. Select an action and then click New in the Action List pane.

   The properties dialog displays and offers options for configuring that action. A group box displays at the bottom of the pane for defining remote server names by specific name or by AN group.
6. Enter the host name and other information for the server, or the remote server, as needed.
7. Save your work, and distribute the policy.

eaudit.ini

The following topics describe sections of the initialization file that you might need to change.

Note: String values for the following entries are case-sensitive.

Data Server

The keys in the topics that follow apply to the Data Server.

Collector

eTrust Audit maintains the information about the collector used by the data server.

The values are as follows:

MaxBulkInsertRows

Type

DWORD Value

Data

The maximum number of rows for bulk insert. Default: 100.

MaxBulkBufferSize

Type

DWORD Value

Data

The maximum size of the bulk insert buffer (in bytes). This parameter is reserved for future use.

BulkInsertCutoffTime

Type

DWORD Value

Data

The maximum number of seconds to wait for buffer to reach either MaxBulkInsertRows or MaxBulkBufferSize. Default: 5 seconds.

MaxDBSessions

Type

DWORD Value

Data

The maximum number of connections to open on the database. Default: 10.

Collector\Database

eTrust Audit maintains information about the Collector Database used by the Collector service to insert events.

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

OracleHome

Type

String Value

Data

Install location of the Oracle Database.

OracleSid

Type

String Value

Data

Oracle SID of the Audit Event Database on the local host.

TwoTask

Type

String Value

Data

Oracle service name of the Audit Event Database on the remote host. Note: either TwoTask or OracleSID is used.

User

Type

Binary Value

Data

Specify the login name of the user who can connect to the Audit Event Database. The user's name is encrypted.

Password

Type

Binary Value

Data

Specify the password of the user who can connect to the Audit Event Database. The user's password is encrypted.

Database

This key is not used.

Viewer

eTrust Audit maintains information about the Viewer. Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

FiltersDir

The data value specified for the FiltersDir key is used to identify the location where the filter definition files are stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir/dat/Filters/.

Ports

Normally, eTrust Audit uses one of its default ports or uses Portmapper to assign a port dynamically. eTrust Audit uses the values of these keys under the following conditions:

- The default port is busy
- The service cannot get the dynamic port from the Portmapper

Under normal circumstances, you would not have any reason to modify these values. However, if a port is being used by another application or service or you need to route events through a firewall, you must modify or set these entry values.

The entries and their default values are as follows:

MonitorPort

The data value specified for the MonitorPort key is used by the Action Manager to route events to the Security Monitor and by the Security Monitor to receive events.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by Portmapper.

RouterPort

The data value specified for the RouterPort key is used by the Router and Redirector.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by p\Portmapper.

RouterSapiPort

The data value specified for the RouterSapiPort key is used by the UNIX Recorder, the Recorder, the Generic NT Recorder, the Check Point Firewall-1 Recorder, and applications that use SAPI, and is used by the Router.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by Portmapper.

MonitorSapiPort

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by Portmapper.

CollectorSapiPort

The data value specified for the CollectorPort key is used by the Action Manager to route events to the Collector, and by the Collector to receive events using the SAPI protocol.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by Portmapper.

CollectorPort

The data value specified for the CollectorPort key is used by the Action Manager to route events to the Collector, and by the Collector to receive events.

Type

String Value, UDP, bi-directional

Data

Specify the number of the port to be used. By default, the port is dynamically assigned by Portmapper.

DistributionPort

The data value specified for the DistributionPort key is used by the Distribution Server and the Distribution Agent.

Type

String Value, TCP/IP, bi-directional

Data

Specify the number of the port to be used. The default is 8025.

SNMPRecorderPort

The data value specified for the SNMPRecorderPort key is used by the SNMP Recorder.

Type

String Value, one-directional (incoming)

Data

Specify the number of the port to be used. The default is 162.

SNMPTrapPort

The data value specified for the SNMPTrapPort key is used by the Action Manager to route actions defined as Action SNMP to the Router.

Type

String Value, one-directional (outgoing)

Data

Specify the number of the port to be used. The default is 162.

Messages

The Message section contains entries that describe the location of the message file:

MessageFile

The data value specified for the MessageFile key is used to identify the name and location of the message file.

Type

String Value

Data

Specify the name of the message file, including its full path. The default is *install_dir*\Messages\message.txt (/Messages/message.txt on Unix)

Severity

Under this section, you specify values for the types of messages. There are several subsections with the same values: Targets (Mandatory) and SkipTimeout (Optional). Only the default SkipTimeout value differs.

The values are described in the topics that follow.

Fatal

Targets

The data value specified for the Targets key is used to identify the targets where fatal messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Critical

Targets

The data value specified for the Targets key is used to identify the targets where critical messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Error

Targets

The data value specified for the Targets key is used to identify the targets where error messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Warning

Targets

The data value specified for the Targets key is used to identify the targets where warning messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Info

Targets

The data value specified for the Targets key is used to identify the targets where info messages are to be sent.

Type

String Value

Data

Specify the name of the targets, separated by commas. The default value is Monitor,Log.

SkipTimeout

The data value specified for the SkipTimeout key is used to identify the minimum time interval between identical messages. If eTrust Audit receives two of the same message within the interval, it discards the second message.

Type

DWORD Value

Data

Specify the time interval in seconds. The default value is 0 seconds.

Monitor

eTrust Audit maintains information about the self-monitor target to use to send its own notification messages under the following entries:

Host

The data value specified for the Host key is used to identify the host where messages are to be sent.

Type

String Value

Data

Specify the name of the host. The default value is localhost.

MonitorPort

The data value specified for the MonitorPort key is used to identify the port used by the Security Monitor.

Type

String Value

Data

Specify the number of the port. By default, the port is dynamically assigned by portmapper.

Policy Manager (Solaris 10)

The keys in the topics that follow apply to the Policy Manager.

In order to distribute policies (or MPs in r8SP2), the Policy Manager must extract data from the database, compile them and place the compiled objects in a staging area where Distribution Server will pick up for distribution to Audit Clients. The location of this staging area is stored in the following value:

PolicyDir

Specify the directory of the staging area where compiled policies (and MPs in r8SP2) are stored.

Policy Manager Database

eTrust Audit maintains information about the database used by the Policy Manager under this key.

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

DSN

The data value specified for the DSN key is used to identify the name of the data source.

Type

String Value

Data

Specify the data source name. The default value is eAuditPMDB.

UserName

The data value specified for the UserName key is used to identify the name of the user under whose name changes can be made to the database.

Type

Binary Value

Data

Specify the user name. The value is encrypted.

Password

The data value specified for the Password key is used to identify the password of the user under whose name changes can be made to the database.

Type

Binary Value

Data

Specify the password. The value is encrypted.

ORACLE_HOME

The data value specified for the ORACLE_HOME key is used to identify the install location of the Oracle Database.

Type

String

Data

Specify the directory of the Oracle Database.

ODBC_HOME

The data value specified for the ODBC_HOME key is used to identify the install location of the CAI/PT ODBC drivers from CA.

Type

String

Data

Specify the directory of the CAI/PT ODBC.

Policy Manager Distribution Log

eTrust Audit maintains information about the log file used to store messages about the success or failure of policy distribution used by the Policy Manager under this key. Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxLogSize

The data value specified for the MaxLogSize key is used to identify the number of records to be stored in the log.

Type

DWORD Value

Data

Specify the number of records. The default value is 10000.

MaxTimeOut

The data value specified for the MaxTimeOut key is used to identify the maximum time (in seconds) the distribution server waits to write to the database. After this period ends without success, an error is recorded in the machine event log.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 60.

DelPartSize

The data value specified for the DelPartSize key is used to identify the number of records to erase when the value of MaxLogSize is reached.

Type

DWORD Value

Data

Specify the number of records to be erased. The default is 500.

Distribution Server

eTrust Audit maintains configuration information about the distribution server under this key. Under normal circumstances, you would not have any reason to modify these values.

The key values are described in the topics that follow.

Distribution Server Threads

The eTrust Audit Distribution Server uses threads to push a policy to AN nodes.

The key values are as follows:

MaxThreadNumber

(Configurable through the Audit Administrator user interface.) Defines the number of parallel distributions of the same policy to the eTrust Audit Clients.

Type

DWORD Value

Data

Specify the number of threads for parallel distribution of the same policy.

Default: 10

Limits: 64 (MAXIMUM_WAIT_OBJECTS)

Other key values related to the distribution process tuning are following:

ActivePolling

(Configurable through the Audit Administrator user interface.) Defines whether Policy Manager should directly poll the Clients to request end-point status

Type

DWORD Value

Data

0 or 1. 0 means no polling; the Audit Client will send status if there is any change in the status (new policies or MPs, deleted policies or MPs, tampered policies or MPs, …). 1 means active polling.

Default: 0

NodeAutoCorrect

(Configurable through the Audit Administrator user interface.) Defines whether Policy Manager should automatically send the correct policies or MPs to a Client if the latter's end-point status does not correspond to its state stored in Policy Manager database.

Type

DWORD Value

Data

0 or 1. 0 means no autocorrect. 1 means autocorrect if there is discrepancy between the Client's end-point status and Policy Manager information for this Client.

Default: 0

MaxDistRetryNumber

(Configurable through the Audit Administrator user interface.) Defines the maximum number of retries to distribute.

Type

DWORD Value

Data

Specify the maximum number of distribution retries.

Default: 3

Distribution Server\Queue Manager\Queues

eTrust Audit maintains information about the queues used by the distribution server. Under normal circumstances, you should not have any reason to modify these values.

The key values are as follows:

DirectoryName

The data value specified for DirectoryName is used to identify the directory where queues are located.

Type

String Value

Data

Specify the name of the directory where the queues are located. The default is install_dir\dat\Queue\distrib (or install_dir/dat/Queue/distrib on UNIX).

DistributionQueue Rules

eTrust Audit maintains information about the queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\DistributionQueue\Queue Rules

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

distribute

The data value specified for the distribute key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *distribute;*.

remove

The data value specified for the remove key is used to identify the name of the action and target.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *remove;*.

DistributionQueue Parameters

eTrust Audit maintains information about the distribution queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\DistributionQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the file in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 100 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the distribution server operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the distribution server operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the distribution server operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the distribution server operates in the queue before moving to another queue. The default value is 50 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the distribution server service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 10 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a policy again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 1800 seconds (30 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a policy can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a policy can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

PollingQueue Rules

eTrust Audit maintains information about the polling queue rules used by the Policy Manager under this key. Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

add

The data value specified for the add key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *add;*.

change

The data value specified for the change key is used to identify the name of the action.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *change;*.

delete

The data value specified for the delete key is used to identify the name of the action and target.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is *delete;*.

PollingQueue Parameters

eTrust Audit maintains information about the default queue rules used by the Policy Manager under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\Policy Manager\Distribution Server\Queue Manager\Queues\PollingQueue\Queue Parameters

Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 50.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 1000 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 50 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 10 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 86400 seconds (24 hours).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 0.

Distribution Server Queue Actions

eTrust Audit maintains information about the actions used by the Distribution Server under this key. Under normal circumstances, you would not have any reason to modify these values.

The key values are as follows:

add

The add action adds a new audit node to the polling queue.

change

The change action changes an audit node in the polling queue.

delete

The delete action removes an audit node from the polling queue.

distribute

The distribute action routes policies to distribution agents.

remove

The remove action removes policies from the distribution agents.

Recorders

By default, the recorder sends messages to the router on the system where it is installed. However, as you begin to deploy eTrust Audit throughout your enterprise, you can change this value to send events to dedicated routers. You identify these dedicated router systems by changing the value of DefaultRouter from localhost to the host name or IP address of the dedicated router system.

The values are as follows:

RecordersIniFile

Specify the path to the recorder .ini file. The default value is ini/recorder.ini.

DefaultRouter

Specify the host name or IP address of the computer that runs the eTrust Audit router. An empty value means use the local host.

SNMP Recorder

eTrust Audit maintains information about the mapping file used by the SNMP recorder to parse events.

The values are as follows:

MPFile

The data value specified for the MPFile key is used to identify the name of the mapping file used by the SNMP recorder to parse events.

Type

String Value

Data

Specify the name of the mapping file, including path. The default is install_dir\cfg\snmptd_rec.mp.

Router

eTrust Audit maintains information used by the router.

The values are as follows:

RulesDirectory

The data value specified for the RulesDirectory key is used to identify the directory where routers configuration files are located.

Type

String Value

Data

Specify the name of the directory where the router configuration files are located. The default is install_dir\cfg\.

RulesExtension

The data value specified for the RulesExtension key is used to identify the extension for router configuration files.

Type

String Value

Data

Specify the extension for router configuration files. The default is .cfg. Setting this value is optional.

Queue Manager

eTrust Audit maintains information about the queues used by the router. The values are as follows:

DirectoryName

The data value specified for DirectoryName is used to identify the directory where queues are located.

Type

String Value

Data

Specify the name of the directory where the queues are located. The default is install_dir\dat\Queue\route (or /dat/Queue/route on Unix).

Queues\AlertQueue

eTrust Audit maintains information about the alert queue rules. The values are as follows:

monitor

The data value specified for the monitor key is used to identify the name of the action and target, separated by a semicolon.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "monitor; "

snmp

The data value specified for the snmp key is used to identify the name of the action and target, separated by a semicolon.

Type

String Value

Data

Specify the name of the action and the target separated by a semicolon. The default value is "snmp; "

AlertQueue Parameters

eTrust Audit maintains information about the alert queue rules used by the router. The values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to loose any record.

Setting this value is optional. The default value is 1.

Collection Queue

eTrust Audit maintains information about the collection queue rules used by the router. The values are as follows:

Collector

Specify the name of the collector. The default value is “collector".

CollectionQueue Parameters

eTrust Audit maintains information about the collection queue rules used by the router. The values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

Default Queue

The default section has no rules; it gets all the rules of the other subsections.

DefaultQueue Parameters

eTrust Audit maintains information about the default queue rules used by the router. The values are as follows:

MaxFileNum

The data value specified for the MaxFileNum key is used to identify the maximum number of files in the queues.

Type

DWORD Value

Data

Specify the number of files in the queues. The default value is 10.

MaxFileSize

The data value specified for the MaxFileSize key is used to identify the size of the files in the queue.

Type

DWORD Value

Data

Specify the size of the file in the queue in KB. The default value is 500 KB.

MaxActionTime

The data value specified for the MaxActionTime key is used to identify the maximum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the maximum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 500 milliseconds.

MinActionTime

The data value specified for the MinActionTime key is used to identify the minimum time the action manager operates in the queue before moving to another queue.

Type

DWORD Value

Data

Specify the minimum number of milliseconds the action manager operates in the queue before moving to another queue. The default value is 20 milliseconds.

SleepTime

The data value specified for the SleepTime key is used to identify the time the action manager service sleeps without writing any data from the queue.

Type

DWORD Value

Data

The sleep interval in seconds. The default value is 3 seconds.

RetryDelay

The data value specified for the RetryDelay key is used to identify the amount of time that passes before trying to transmit a message again.

Type

DWORD Value

Data

The retry interval in seconds. The default value is 600 seconds (10 minutes).

MaxLifeTime

The data value specified for the MaxLifeTime key is used to identify the maximal time a message can be in the queue before it is erased.

Type

DWORD Value

Data

The maximum time in seconds a message can be in the queue before it is erased. The default value is 86400 seconds (24 hours).

DeleteOldFiles

The data value specified for the DeleteOldFiles key is used to identify the whether the oldest queue file should be deleted if the number of MaxFileNum is reached.

Type

DWORD Value

Data

Specify either of the following:

- Specify 1 if you want to delete the oldest queue file when the number of files in the queue equals the number set in the MaxFileNum parameter.
- Specify 0, if you do not want to lose any record.

Setting this value is optional. The default value is 1.

DefaultQueue Actions

eTrust Audit maintains information about the actions used by the router. The values are as follows:

file

The file action routes events to a file in ASCII text format. It has no parameters you should change.

monitor

The monitor action routes events to the security monitor. It has no parameters you should change.

collector

The collector action routes events to the collector database. It has no parameters you should change.

mail

The mail action routes messages to a designated SMTP mail server and onto an email address.

The mail parameters are as follows:

MailSubject

The data value specified for the MailSubject key is used to identify the subject line for eTrust Audit mail. Specify the text you want to appear in the subject line of email sent by eTrust Audit. The default is "Notification from eTrust Audit."

screen

The screen action routes events to an NT screen session.

remote

The remote action routes events to an action manager on the host named in the action where it is executed without filtering.

route

The route action sends events to the host named in the action where it reviewed by the router on that system and executed according to any filters that apply on that system.

snmp

The snmp action sends SNMP traps to the host named in the action.

program

The program action executes a command on the host named in the action on the local host.

unicenter

The unicenter action routes events to the Event Management Console on the host named in the action.

The key values are as follows:

UnicenterHome

The data value specified for the UnicenterHome key is used to identify the location of the Unicenter installation. Specify the location of the Unicenter installation.

Management Agent

eTrust Audit maintains information about which systems are trusted policy servers and parameters related to policy distribution.

When you install eTrust Audit, you identify the name of a trust policy server. By changing the value of the TrustedServers, you can add more servers to identify other policy servers.

The values are as follows:

TrustedServers

Specify the host names or IP addresses of one or more policy servers, separated by commas.

Parameters

eTrust Audit maintains information about the how policy management runs. The values are as follows:

AuditStatusDir

The data value specified for the AuditStatusDir key is used to identify the directory where end-point status files are located.

Type

String Value

Data

Specify the directory name. The default value is install_dir/dat/status.

iGatewayStatusDir

The data value specified for the iGatewayStatusDir key is used to identify the directory where the iRecorder status files are located.

Type

String Value

Data

Specify the directory name. The default value is nodestatus, which is located under <igateway_install_dir>, by default /opt/CA/SharedComponents/iTechnology. This value can be an absolute path or a relative path. The relative path starts from the <igateway_install_dir>.

TmpPolicyDir

The data value specified for the TmpPolicyDir key is used to identify the directory where temporary policy files are stored.

Type

String Value

Data

Specify the directory name. The default value is install_dir\dat\tmp\agent_tmp_policies (or /dat/tmp/agent_tmp_policies on Unix).

ConnectionTimeout

The data value specified for the ConnectionTimeout key is used to identify the number of seconds after which a connection between a policy server and distribution agent is closed.

Type

DWORD Value

Data

Specify the number of seconds after which the connection is broken. The default value is 600 seconds.

ReceiveTimeout

The data value specified for the ReceiveTimeout key is used to identify an internal parameter for the TCP session.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 10 seconds.

SendTimeout

The data value specified for the SendTimeout key is used to identify an internal parameter for the TCP session.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 10 seconds.

DistributionTimeout

The data value specified for the DistributionTimeout key is used to identify the time from the start of the TCP session until the agent receives the policy.

Type

DWORD Value

Data

Specify the number of seconds. The default value is 800 seconds.

AN Types

eTrust Audit maintains information about the types of event logs defined to it. The values are as follows:

Apache

eTrust Audit maintains information about the library used by the Apache AN type.

LibraryName

Specify the library used to process Apache events. The default value is TGNR.

Default

eTrust Audit maintains information about the library used by the Default AN type.

LibraryName

Specify the library used to process Default events. The default value is TGNR.

eTrust Access Control

eTrust Audit maintains information about the library used by the eTrust Access Control AN type.

LibraryName

Specify the library used to process eTrust Access Control events. The default value is TGNR.

Netscape

eTrust Audit maintains information about the library used by the Netscape AN type.

LibraryName

Specify the library used to process Netscape events. The default value is TGNR.

NT

eTrust Audit maintains information about the library used by the NT AN type.

LibraryName

Specify the library used to process NT events. The default value is TALR.

Oracle

eTrust Audit maintains information about the library used by the Oracle AN type.

LibraryName

Specify the library used to process Oracle events. The default value is TGNR.

UNIX

eTrust Audit maintains information about the library used by the UNIX AN type.

LibraryName

Specify the library used to process UNIX events. The default value is TGNR.

recorder.ini

The following topics describe sections of the ini file that you might need to change.

Recorder Modules

The recorders supported by eTrust Audit in UNIX are:

- File Spooler (UNIX native recorder)
- Netscape
- Apache
- Oracle

Each recorder has its own section in the recorder.ini file, which bears its name. The topics that follow describe entries found in sections for each recorder.

Definitions

The following definitions are found in all UNIX recorders supported by eTrust Audit, except for the last definition, ORACLE_HOME, which is found only in the Oracle Recorder:

ModuleName

Specify the unique name for the Recorder module.

LibraryPrefix

Specify the prefix for the name of the Recorder module library.

Active

When specified, activates the Recorder module.

SleepInterval

Specify the time, in seconds, that the service sleeps after each record. The default value is 1.

SendInterval

Specify the time, in seconds, that the service sleeps after the value of MaxSeqNoSleep is reached. The default value is 10.

MaxSeqNoSleep

Specify the maximum number of records sent before sleeping. The default value is 50.

ORACLE_HOME

Specify where Oracle is located on the file system.

Parameters

The Parameters section is found in all UNIX recorders supported by eTrust Audit. However, there is a significant difference between Oracle and other recorders.

Non-Oracle Parameters

In all recorders except Oracle, you can find two parameters under this section.

DatFilePath

A mandatory parameter, found in all UNIX recorders supported by eTrust Audit.

MPDebug

An optional parameter and is found in all recorders except Oracle. If you specify 1, debug information for the message parser is generated.

Oracle-Only Parameters

Besides the DatFilePath parameter and MP file parameter (see the Log Data), Oracle has additional parameters, which are not found in the other recorders.

These other parameters are as follows:

DatFilePath

Specify the relative path to the .dat file as follows:

UNIX

The default value is dat/recorders/syslog.dat.

Netscape

The default value is dat/recorders/netscape.dat.

Apache

The default value is dat/recorders/apache.dat.

Oracle

The default value is dat/recorders/oracle.dat.

ORACLE_SID

Specify the Oracle SID on the local host.

TWO_TASK

Specify the Oracle service name on the remote host.

Password

Specify the password for the user that can connect to the Oracle database. The value is encrypted.

Username

Specify the name of the user that can connect to the Oracle database. The value is encrypted.

Log Data

The Log Data section describes parameters for the recorder logs. The file spooler has two logs: syslog and sulog. Other recorders have only one log that bears their name: Netscape or Apache.

Note the following:

- The only parameter here that is found also in Oracle is the MPfile parameter.
- The ConfigFile and Source parameters are found only in syslog.

The values are as follows:

LogName

Specify the recorder name: Unix, Netscape, or Apache. You should not change this value.

StartOver

If 1 is specified, eTrust Audit restarts reading the log files (ignores the .dat file). The default value is 0.

SkipCurrentLogs

Specify one of the following:

0

Sends all records from the log files.

1

Skips old records from the log files.

Mpfile

Specify the relative path to .mp file as follows:

UNIX

The default value is cfg/syslog.mp, or cfg/sulog.mp

Netscape

The default value is cfg/netscape.mp.

Apache

The default value is cfg/apache.mp.

Oracle

The default value is cfg/oracle.mp.

ConfigFile

Specify the relative path to syslog configuration file. The default value is /etc/syslog.conf.

Source

Specify one of the following:

0

Takes the log files defined in the default configuration file plus all log files found in the LogFiles section.

1

Takes the log files defined in the configuration file under the ConfigFile parameter, plus all log files found in the LogFiles section.

LogFiles

Specify a list of paths to log files from which records are to be read as follows:

Chapter 9: Firewall Considerations

This section contains the following topics:

Introduction

To enable communications between eTrust Audit components through a firewall, you must configure the Audit components on each side of the firewall to use the same open port in the firewall. For example, you might:

- Install the Security Monitor, the Router, or the Collector service on one side of a firewall
- Install the recorder and router services on the opposite side

However, if the firewall does not allow communication in the protected network, the client and the server (the redirector service, the router service, and the Collector service) must be made to agree on a specific port.

Syncing Client and Server Ports

You can ensure port agreement for proper communication through a firewall by setting the same value in the registry at the client and the server stations.

To sync the client and server ports, follow these steps:

1. At the client stations, edit the value Ports.
2. On Windows systems, edit the following registry key:
   HKEY_LOCAL_MACHINE\SOFTWARE\eTrust Audit\Ports

   For example:
   HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrust Audit\ports\MonitorPort
3. On UNIX systems, edit the value for MonitorPort in the eaudit.ini file.

   For more information on these parameters, see Ports (see page 95) or UNIX INI Files (see page 155).
4. Enter the same entry and value in the registry (or in the eaudit.ini file) at the target station.

   Note: The ports must be open in both directions.

Chapter 10: Windows NT Security-related Event IDs

This section contains the following topics:

Introduction

The following lists numerous event ID's that relate to security for each supported operating system or platform:

Event IDs

The following events are among those directly involved in security.

Event ID	Type	Description
512	Success Audit	Windows NT startup.
513	Success Audit	Windows NT shutdown.
514	Success Audit	Authentication package has been loaded. It will be used to authenticate logon attempts.
515	Success Audit	Trusted logon process has been registered. It will be trusted to submit logon requests.
516	Success Audit	Some audit messages have been discarded (full queue).
517	Success Audit	The event log was cleared. Indicates primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID.
518	Success Audit	Notification package has been loaded. It will be notified of any account or password changes.

Event ID	Type	Description
528	Success Audit	Successful logon. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
529	Failure Audit	Failed logon—unknown user name or bad password. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
530	Failure Audit	Failed logon—time restriction violation. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
531	Failure Audit	Failed logon—account disabled. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
532	Failure Audit	Failed logon—account expired. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
533	Failure Audit	Failed logon—user not permitted at this computer. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
534	Failure Audit	Failed logon—logon type not permitted for this user. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
535	Failure Audit	Failed logon—password expired. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
536	Failure Audit	Failed logon—Netlogon component not active. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.

Event ID	Type	Description
537	Failure Audit	Failed logon—unexpected error. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
538	Success Audit	Logoff. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
539	Failure Audit	Failed logon—account locked out. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
560	Success Audit	Object open. Includes the following: Object server, object type, object name, new handle ID, operation ID, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID.
561	Success Audit	Handle allocated. Includes handle, ID, operation ID, and process ID.
562	Success Audit	Handle closed. Includes handle, ID, operation ID, and process ID.
563	Success Audit	Object open for delete. Includes the following: Object server, object type, object name, new handle ID, operation ID, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID.
564	Success Audit	Object deleted. Includes object server, handle ID, and process ID.
576	Success Audit	Special privileges assigned to new logon. Includes user name, domain, login ID, and assigned privilege.
577	Success Audit	Privilege service called. Includes the following: server, service, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and privileges.

Event ID	Type	Description
578	Failure Audit	Privileged object operation. Includes the following: object server, object handle, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and privileges.
592	Success Audit	New process created. Includes the following: new process ID, image file name, creator process ID, user name, domain, logon ID.
593	Success Audit	Process exited. Includes the following: process ID, user name, domain, logon ID.
594	Success Audit	Handle duplicated. Includes the following: source handle ID, source process ID, target handle ID, target process ID.
595	Success Audit	Indirect access to an object. Includes the following: object type, object name, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and accesses.
608	Success Audit	User right assigned. Includes the following: user right, assigned to, assigned by, user name , and logon ID.
609	Success Audit	User right removed. Includes the following: user right, removed from, removed by, user name , and logon ID.
610	Success Audit	New trusted domain. Includes the following: domain name, domain ID, established by, user name , domain, and logon ID.
611	Success Audit	Removing trusted domain. Includes the following: domain name, domain ID, removed by, user name , domain, and logon ID.

Event ID	Type	Description
612	Success Audit	Audit policy change. Includes the following: new policy name, and success and failure for System, Logon/Logoff, Object Access, Privilege Use, Detailed Tracking, Policy Change, and Account Management. It also includes changed by, user name, domain name, logon ID.
624	Success Audit	User account created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
625	Success Audit	Account type changed. Includes the following: target account name, target domain, target account ID, new type, caller user name, caller logon ID.
626	Success Audit	Account enabled. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID.
627	Success Audit	Change password attempt. Includes the following: target account name, target domain, target account ID, caller user name, domain, caller logon ID, privileges.
628	Success Audit	Password set. Includes the following: target account name, target domain, target account ID, caller user name, domain, caller logon ID.
629	Success Audit	Account disabled. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID.
630	Success Audit	Account deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
631	Success Audit	Global group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
632	Success Audit	Global group member added. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
633	Success Audit	Global group member removed. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
634	Success Audit	Global group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
635	Success Audit	Local group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
636	Success Audit	Local group member added. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
637	Success Audit	Local group member removed. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
638	Success Audit	Local group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
639	Success Audit	Local group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
640	Success Audit	General account database changed. Includes the following: type of change, object type, object name, object ID, caller user name, caller domain, caller logon ID.
641	Success Audit	Global group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
642	Success Audit	User account changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
643	Success Audit	Domain policy changed. Includes the following: domain, domain ID, caller user name, caller domain, caller logon ID, privileges.
644	Success Audit	User account locked out. Includes the following: target account name, target account ID, caller machine name, caller user name, caller domain, caller logon ID.

Windows 2000 Event IDs

The following events are among those directly involved in security.

Event ID	Type	Description
512	Success Audit	Windows NT startup.
513	Success Audit	Windows NT shutdown.
514	Success Audit	Authentication package has been loaded. It will be used to authenticate logon attempts.

Event ID	Type	Description
515	Success Audit	Trusted logon process has been registered. It will be trusted to submit logon requests.
516	Success Audit	Some audit messages have been discarded (full queue).
517	Success Audit	The event log was cleared. Indicates primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID.
518	Success Audit	Notification package has been loaded. It will be notified of any account or password changes.
528	Success Audit	Successful logon. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
529	Failure Audit	Failed logon—unknown user name or bad password. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
530	Failure Audit	Failed logon—time restriction violation. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
531	Failure Audit	Failed logon—account disabled. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
532	Failure Audit	Failed logon—account expired. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
533	Failure Audit	Failed logon—user not permitted at this computer. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.

Event ID	Type	Description
534	Failure Audit	Failed logon—logon type not permitted for this user at this machine. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
535	Failure Audit	Failed logon—password expired. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
536	Failure Audit	Failed logon—Netlogon component not active. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
537	Failure Audit	Failed logon—unexpected error. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
538	Success Audit	Logoff. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
539	Failure Audit	Failed logon—account locked out. Indicates user name, domain, logon type, logon process, authentication package, and workstation name.
540	Success Audit	Successful network logon. Includes the following: user name, domain, logon, ID, logon type, logon process, authentication package, workstation name.
541	Success Audit	IKE security association established. Includes the following: mode, peer identity, filter, parameters.
542	Success Audit	IKE security association ended. Includes the following: mode--data protection, filter, inbound SPI, outbound SPI.
543	Success Audit	IKE security association ended. Includes the following: mode--key exchange, filter.

Event ID	Type	Description
544	Failure Audit	IKE security could not be established because the peer could not authenticate. The certificate trust could not be established. Includes the following: peer identity, and filter.
545	Failure Audit	IKE peer authentication failed. Includes the following: peer identity, and filter.
546	Failure Audit	IKE security could not be established because the peer sent and invalid proposal. Includes the following: mode, filter, attribute, expected value, received value.
547	Failure Audit	IKE security association negotiation failed. Includes the following: mode, filter, failure point, failure reason.
560	Success Audit	Object open. Includes the following: Object server, object type, object name, new handle ID, operation ID, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, accesses, privileges.
561	Success Audit	Handle allocated. Includes handle, ID, operation ID, and process ID.
562	Success Audit	Handle closed. Includes handle, ID, operation ID, and process ID.
563	Success Audit	Object open for delete. Includes the following: Object server, object type, object name, new handle ID, operation ID, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, accesses, privileges.
564	Success Audit	Object deleted. Includes object server, handle ID, and process ID.

Event ID	Type	Description
565	Success Audit	Object open. Includes the following: Object server, object type, object name, new handle ID, operation ID, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, accesses, privileges, properties.
566	Success Audit	Object operation. Includes the following: operation type, object type, object name, handle ID, operation ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, accesses, privileges.
576	Success Audit	Special privileges assigned to new logon. Includes user name, domain, login ID, and assigned privilege.
577	Success Audit	Privilege service called. Includes the following: server, service, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and privileges.
578	Failure Audit	Privileged object operation. Includes the following: object server, object handle, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and privileges.
592	Success Audit	New process created. Includes the following: new process ID, image file name, creator process ID, user name, domain, logon ID.
593	Success Audit	Process exited. Includes the following: process ID, user name, domain, logon ID.
594	Success Audit	Handle duplicated. Includes the following: source handle ID, source process ID, target handle ID, target process ID.

Event ID	Type	Description
595	Success Audit	Indirect access to an object. Includes the following: object type, object name, process ID, primary user name, primary domain, primary logon ID, client user name, client domain, client logon ID, and accesses.
608	Success Audit	User right assigned. Includes the following: user right, assigned to, assigned by, user name , and logon ID.
609	Success Audit	User right removed. Includes the following: user right, removed from, removed by, user name , and logon ID.
610	Success Audit	New trusted domain. Includes the following: domain name, domain ID, established by, user name , domain, and logon ID.
611	Success Audit	Removing trusted domain. Includes the following: domain name, domain ID, removed by, user name , domain, and logon ID.
612	Success Audit	Audit policy change. Includes the following: new policy name, and success and failure for System, Logon/Logoff, Object Access, Privilege Use, Detailed Tracking, Policy Change, and Account Management. It also includes changed by, user name, domain name, logon ID.
613	Success Audit	IPSec policy agent started. Includes the following: IPSec policy agent, policy source, event data.
614	Success Audit	IPSec policy agent disabled. Includes the following: IPSec policy agent, event data.
615	Success Audit	IPSec Policy Agent service. Includes event data.
616	Failure Audit	IPSec policy agent encountered a potentially serious failure. Includes event data.

Event ID	Type	Description
617	Success Audit	Kerberos policy changed. Includes changed by, user name, domain name, login ID, changes made, parameter name new and (old).
618	Success Audit	Encrypted data recovery policy changed. Includes the following: changed by, user name, domain name, logon ID, changes made parameter new and (old).
619	Success Audit	Quality of service policy changed. Includes the following: changed by, user name, domain name, logon ID, changes made parameter new and (old).
620	Success Audit	Trusted domain information modified. Includes the following: domain name, domain ID, modified by, user name, domain, logon ID.
624	Success Audit	User account created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
625	Success Audit	Account type changed. Includes the following: target account name, target domain, target account ID, new type, caller user name, caller logon ID.
626	Success Audit	Account enabled. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID.
627	Success Audit	Change password attempt. Includes the following: target account name, target domain, target account ID, caller user name, domain, caller logon ID, privileges.
628	Success Audit	User account password set. Includes the following: target account name, target domain, target account ID, caller user name, domain, caller logon ID.

Event ID	Type	Description
630	Success Audit	User account deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
631	Success Audit	Security enabled global group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
632	Success Audit	Security enabled global group member added. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
633	Success Audit	Security enabled global group member removed. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
634	Success Audit	Security enabled global group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
635	Success Audit	Security enabled local group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
636	Success Audit	Security enabled local group member added. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
637	Success Audit	Security enabled local group member removed. Includes the following: member, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
638	Success Audit	Security enabled local group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
639	Success Audit	Security enabled local group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
640	Success Audit	General account database changed. Includes the following: type of change, object type, object name, object ID, caller user name, caller domain, caller logon ID.
641	Success Audit	Security enabled global group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
642	Success Audit	User account changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
643	Success Audit	Domain policy changed. Includes the following: domain, domain ID, caller user name, caller domain, caller logon ID, privileges.
644	Success Audit	User account locked out. Includes the following: target account name, target account ID, caller machine name, caller user name, caller domain, caller logon ID.

Event ID	Type	Description
645	Success Audit	Computer account created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
646	Success Audit	Computer account changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
647	Success Audit	Computer account deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
648	Success Audit	Security disabled local group created. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
649	Success Audit	Security disabled local group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
650	Success Audit	Security disabled local group member added. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
651	Success Audit	Security disabled local group member removed. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
652	Success Audit	Security disabled local group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
653	Success Audit	Security disabled global group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
654	Success Audit	Security disabled global group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
655	Success Audit	Security disabled global group member added. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
656	Success Audit	Security disabled global group member removed. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
657	Success Audit	Security disabled global group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
658	Success Audit	Security enabled universal group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
659	Success Audit	Security enabled universal group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
660	Success Audit	Security enabled universal group member added. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
661	Success Audit	Security enabled universal group member removed. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
662	Success Audit	Security enabled universal group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
663	Success Audit	Security disabled universal group created. Includes the following: new account name, new domain, new account ID, caller user name, caller domain, caller logon ID, privileges.
664	Success Audit	Security disabled universal group changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
665	Success Audit	Security disabled universal group member added. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.

Event ID	Type	Description
666	Success Audit	Security disabled universal group member removed. Includes the following: member name, member ID, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
667	Success Audit	Security disabled universal group deleted. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
668	Success Audit	Group type changed. Includes the following: target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
669	Success Audit	Add SID history. Includes the following: source account name, source account ID, target account name, target account ID, caller user name, caller domain, caller logon ID, privileges.
670	Success Audit	Add SID history. Includes the following: source account name, target account name, target domain, target account ID, caller user name, caller domain, caller logon ID, privileges.
672	Success Audit	Authentication ticket granted. Includes the following: user name, supplied realm name, user ID, service name, service ID, ticket options, ticket encryption type, pre-authentication type, client address.
673	Success Audit	Service ticket granted. Includes the following: user name, user domain, user ID, service name, service ID, ticket options, ticket encryption type, client address.

Event ID	Type	Description
674	Success Audit	Ticket granted renewed. Includes the following: user name, user domain, user ID, service name, service ID, ticket options, ticket encryption type, client address.
675	Failure Audit	Pre-authentication failed. Includes the following: user name, user ID, service name, pre-authentication type, failure code, client address.
676	Failure Audit	Authentication ticket request failed. Includes the following: user name, supplied realm name, user ID, service name, ticket options, failure code, client address.
677	Failure Audit	Service ticket request failed. Includes the following: user name, supplied realm name, service name, ticket options, failure code, client address.
678	Success Audit	Account mapped for logon by . Includes the following: client name, mapped name.
679	Failure Audit	The name could not be mapped for logon by . Includes the following: client name, mapped name.
680	Success Audit	Account used for logon by . Includes the following: account name, workstation.
681	Failure Audit	The login to account by from workstation failed.
683	Success Audit	Session reconnected to winstation. Includes the following: user name, domain, logon ID, session name, client name, client address.
684	Success Audit	Session disconnected to winstation. Includes the following: user name, domain, logon ID, session name, client name, client address.

UNIX Event IDs

For the following sources, the various eTrust Audit recorders use an event ID of 0:

- Syslog.conf
- Sylog
- Oracle
- Netsacpe
- IPlanet
- SNMP
- Check Point Firewall-1

Windows Event IDs

For the following sources, the various eTrust Audit recorders use an event ID of 0:

- MS-IIS
- Microsoft Proxy
- Oracle
- Microsoft ISA
- SNMP
- Check Point Firewall-1

eTrust Access Control Event IDs

The following event IDs are used for eTrust Access Control 5.1 SP1 and below. These events are generated by the seaudit -t command. This list also includes event IDs for eTrust Single Signon 6.5 and lower:

Event ID	Reason
0	No request for LOG operation.
1	User logged in out-of shift with LOGSHIFT property.
2	User audit mode requires logging.
3	Resource audit mode requires logging.
4	Resource in WARNING mode.
5	Serevu utility requested logging.

Event ID	Reason
6	Network attack protection.
7	Incoming or outgoing connection (not from Log reason, but from stage code).
8	PAM support 1 failed logon.
10	A specific request to log operation.

Chapter 11: Importing, Exporting, and Converting Policies

This section contains the following topics:

Introduction

During administration of your r8 SP2 Policy Manager database, you may from time-to-time need to import or export policies. You may also need to convert Windows system PTF files to XML for use with older iRecorders and SAPI Recorders.

You can use the following utilities to convert policy files to XML, and to import and export policy files:

acptf2xml

Converts Windows PTF policy files to XML format.

acxml2pmdb

Imports XML policy files to the r8 SP2 Policy Manager database.

acpmdb2xml

Exports policy files from the r8 SP2 Policy Manager database to XML files.

Converting PTF Files to XML with the acptf2xml Utility

The acptf2xml utility converts policy template files in .ptf format to policy files in XML format. It is used only on Windows systems.

For example, if you download an iRecorder and you want to import its default policies to the r8 SP2 Policy Manager database, it may not have an XML policy file supplied with it. In that case, you would need to convert the supplied .ptf file to XML using this utility, and then import the new XML file to the Policy Manager database.

After you convert PTF files to XML, you can use the acxml2pmdb utility (see page 232) to import them to the r8 SP2 Policy Manager database.

Note: Once you have imported the default policies, you must use the Policy Manager interface to add actions to the policies and distribute them to audit nodes before you receive any events.

acptf2xml Requirements

To use the acptf2xml utility, you must install the r8 SP2 Policy Manager.

acptf2xml Options

[-help]

Retrieves the command line help for this utility.

–ptf <file>

Specifies the fully-qualified name of the PTF file that contains the policy you want to convert to XML format.

[-antid <number>]

Specifies an audit node type ID for which you want to convert policies to XML format. This parameter is a CA internal-only value used for CA templates. This value is not used by CA customers.

[-policyid <number>]

Specifies the policy number for a predefined policy that you want to convert to XML format. This parameter is a CA internal-only value used for CA templates. This value is not used by CA customers.

[-caver <version>]

Specifies the version of a predefined policy that you want to export. This parameter is used for CA templates only.

-xml <file>

Specifies the fully-qualified name of the XML file to which the PTF version of the policy will be exported.

acptf2xml Examples

The following are sample commands to help you use this utility:

To convert a user-defined PTF file to XML

```
acptf2xml -ptf MyPolicy.ptf -xml MyPolicy.xml
```

To convert a predefined CA template policy PTF to XML

```
acptf2xml -ptf CAPolicy.ptf -anid 2345 -policyid 23 -caver 1 -xml CAPolicy.xml
```

Importing Policies with the acxml2pmdb Utility

This utility imports policy XML files into the Policy Manager database. You can use this utility on both Windows and Solaris systems.

Some iRecorder packages contain XML policy files for import to the Policy Manager database. If you download an iRecorder that does not have an XML policy file, you can use the acptf2xml utility (see page 230) to convert the existing .ptf file to SML for import.

ac_xml2pmdb Requirements

The following requirements must be met to use ac_xml2pmdb:

- The location of the Java files must be in the path.
- The r8 SP2 Policy Manager must be installed.

acxml2pmdb Options

[-help]

Retrieves command line help for this utility.

–sql | -ora

Specifies whether a Microsoft SQL Server or Oracle database is in use.

-orahome <directory>

Specifies the fully-qualified Oracle home path. This parameter is used only for Oracle databases.

-srv <host>

Specifies the host name of the server where the Oracle database resides. This parameter is used only for Oracle databases.

-port <port>

Specifies the port number on which the Oracle database listens. This optional parameter is used only for Oracle databases. The default port value is 1521.

-sid <sid>

Specifies the SID for the Oracle database. This parameter is used only for Oracle databases.

-dsn <dsn>

Specifies the database service name for the Microsoft SQL Server database. This parameter is used only for SQL Server databases.

-usr <user>

Specifies a valid user name for connection to the database.

-pwd <password>

Specifies the password associated with the user name in the -usr parameter.

-xml <file>

Specifies the name of the XML file that you want to import to the Policy Manager database.

acxml2pmdb Examples

The following are sample commands to help you use this utility:

Solaris

```
./acxml2pmdb -ora -orahome /etc/OraHome -srv OracleServer1 -port 1533 -sid ORA_PMDB
 -usr Admin -pwd Admin -xml MyPolicy.xml
```

Windows

```
acxml2pmdb -sql -dsn MySQLDSN -usr Admin -pwd Admin -xml Mypolicy.xml
```

Exporting Policies with the acpmdb2xml Utility

This utility exports policies from the Policy Manager database to a policy file in XML format. You can use this utility on both Windows and Solaris systems.

acpmdb2xml Requirements

The following requirements must be met to use acpmdb2xml:

- The location of the Java files must be in the path.
- The r8 SP2 Policy Manager must be installed.

acpmdb2xml Options

[-help]

Retrieves command line help for this utility.

–sql | -ora

Specifies whether a Microsoft SQL Server or Oracle database is in use.

-orahome <directory>

Specifies the fully-qualified Oracle home path. This parameter is used only for Oracle databases.

-srv <host>

Specifies the host name of the server where the Oracle database resides. This parameter is used only for Oracle databases.

-port <port>

Specifies the port number on which the Oracle database listens. This optional parameter is used only for Oracle databases. The default port value is 1521.

-sid <sid>

Specifies the SID for the Oracle database. This parameter is used only for Oracle databases.

-dsn <dsn>

Specifies the database service name for the Microsoft SQL Server database. This parameter is used only for SQL Server databases.

-usr <user>

Specifies a valid user name for connection to the database.

-pwd <password>

Specifies the password associated with the user name in the -usr parameter.

-xml <file>

Specifies the name of the XML file that you want to import to the Policy Manager database.

-folder <name>

Specifies the policy folder name that contains the policy you want to export.

-policy <name>

Specifies the name of the policy that you want to to export.

-antype <name>

Specifies the Audit Node types for the exported policy.

-ver <version>

Specifies which version of a policy you want to export.

[-caver <version>]

Specifies the version of a predefined policy that you want to export. This parameter is used for CA templates only.

-xml <file>

Specifies the fully-qualified name of the XML file to which the policy will be exported.

acpmdb2xml Examples

The following are sample commands to help you use this utility:

Solaris

```
./acpmdb2xml -ora -orahome /etc/OraHome -srv OracleServer1 -port 1533 -sid ORA_PMDB
 -usr Admin -pwd Admin -folder FolderName -policy PolicyName -antype ANType
 -ver 1 -xml MyPolicy.xml
```

Windows

```
acpmdb2xml -sql -dsn MySQLDSN -usr Admin -pwd Admin -folder FolderName
 -policy PolicyName -antype ANType -ver 1 -xml Mypolicy.xml
```

Chapter 12: Audit Event Taxonomy and Data

This section contains the following topics:

Introduction

The following sections present an introduction to Audit taxonomy and data model and describe an Audit view of events.

Taxonomy represents a general classification of events that covers several products of different brands. Data model further characterizes an event to allow finer analysis. These characteristics, or event attributes, are called event fields, including taxonomy. These fields comprise the following types:

- Common fields to all events, including taxonomy
- Normalized fields common to events in a particular taxonomy
- Product-specific fields that are specific to the system or application that generates the events

For more details on taxonomy, and data models of specific type of events, please refer to the corresponding recorder reference guides.

Data models are also available online in the Audit Administrator web interface to assist in writing Audit policy rules. To access the data model online, use the Rule Builder wizard in the Policy Manager tab of the Audit Administrator. Data models of active iRecorders in Audit deployment can also be queried through the iRecorder Management tab of the Audit Administrator.

Audit Taxonomy

Audit collects various kinds of events ranging from Physical Security to Data Access, Network Security, …, and from different brands of products. Audit Taxonomy classifies events that allows you to select events based not on the product name of a source of events, but rather on criteria that spans products and brands. This selection permits correlation of events from different products without the need to know each product firsthand. Following is the criteria one may specify for a specific event type:

- General category. For example: all Network Security events.
- Class of observer systems. For example: all Network Security events from Firewall systems. In Audit taxonomy dot notation, this is translated to Network Security.Firewall.*.*.*
- Action that gives rise to the event. For example: all Login events, or more specifically all OS events from System Access (reporting system) that result from a Login action, or all Firewall events that result from a Login action. In Audit taxonomy dot notation (see below), these are translated to:
    - *.*.Login.*.* for all Login events
    - OS.System Access.Login.*.* for OS events resulting from Login action
    - *.Firewall.Login.*.* for Firewall events resulting from Login action
- Result of the action, whether it’s Successful or Failed.
- Severity level of the event, whether it’s Informational, Warning, Critical, Fatal or no severity at all.

These criteria are the basis of Audit taxonomy. Audit taxonomy has five (5) components represented in the following dot notation:
SystemCategory.System.Action.Result.Severity

The table below lists the three components of taxonomy, SystemCategory, System and Action, as currently known to Audit.

Note: Computer Associates, Inc. constantly refines Audit taxonomy as new sources of events are considered and new types of events are collected by Audit. Please refer to the applicable Recorder Reference Guide for up-to-date information on the specific taxonomy and data model used for the Recorder of interest.

SYSTEM CATEGORY	SYSTEM	ACTION	Description

SYSTEM CATEGORY	SYSTEM	ACTION	Description
Anomaly Detection	Event Analysis	Login, Configure, Report...	Systems that detect threats or security breaches through statistical analysis of events from an Event Management system.
Data Access	DBMS, Storage, Backup System, Directory Server	DDLAction, DMLAction, Create, Alter, Drop, Select, AddLogin, Login, …	Products or devices that guarantee continuous access to enterprise data.
Distributed Computing	Application Server, Messaging, Collaboration	Start, Stop, RunApp, MessageDeliver, ReportReceipt, …	Middleware applications that offer an environment to run enterprise computing services such as EJB, database accesses, transaction processing, messaging, …
Event Management	Manager, Server, Agent	LogEvent	Systems that collect, classify, normalize and analyze events for the purpose of auditing activities from networks, hosts, devices, databases, operating systems, system applications and other sources capable of logging events as log files, reports, journalling databases, snmp traps or through special log mechanism.
Host IDS	Manager, Server, Agent	Detect Attack, Suspect Activity, ..	Products that have the ability to detect change in the status of a host and send alerts or take appropriate actions to protect the host.
Host Security	AntiVirus (AV), AccessControl, ContentControl, PersonalFirewall, VPN Client	Clean, Delete, Move, Quarantine, Scan, Update, …	Products or devices that protect a host from non-authorized accesses (viruses, denial-of-service, hacker, …).

SYSTEM CATEGORY	SYSTEM	ACTION	Description
Identity Management	PKI, AuthServer, User Management	Configure, Create User, Delete User, Grant Certificate, Deny Certificate, Expire Certificate, Query Object	Systems that manages users accessing a system or network, users credentials and rules for authentication and authorization.
Network Access	WebServer, Proxy, Search Engine, Content Management, Domain Server, FtpServer, SmtpServer	Get, Post, Put, Load Service, Publish	Software components that manage specific protocols, e.g. http, ftp, mail, search engine.
Network IDS	Manager, Server, Agent	Detect Attack, Suspect Activity, Create, Update, Configure, Delete, …	Products or systems that have the ability to detect change in the status of the network and send alerts or take appropriate actions to protect the network.
Network Management	Routing, Monitoring, Policy Management, Network Mapper	Notify, Reset, Inventory, Alert, …	Systems that implement NOC (network operating center) requirements: defining network topology, adding new nodes, creating routes, monitoring network, establishing policies with actions such as alerts, paging, emails.

SYSTEM CATEGORY	SYSTEM	ACTION	Description
Network Security	Firewall, VPN Gateway, Network Device, ContentControl Server, AntiVirus (AV) Gateway, AccessControl Gateway	Accept, Authorize, Deny, Drop, Permit, Login, Logout, …	Products, devices or appliances that protect a network or hosts sitting on that network, either by blocking certain traffic or re-routing them elsewhere.
NT Application	<NT-Event Source Field>		Windows NT Event in the NT-Application event log.
NT-System	<NT-Event Source Field>		Windows NT Event in the NT-System event log
NT-Security	<NT-Event Source Field>	Login, Logout, AddUser, AddGroup, ChangePolicy, …	Windows NT Event in the NT-Security event log
NT-...	<NT-Event Source Field>		Windows NT Event in other NT event log
OS	Account Management, System Access, System/Application, Object Access, Audit, General	Login, Logout, SwitchUser, Finger, ChangePassword, Run, Restart, Shutdown, Automount, Mount, Access, Share, Quota, Create, Delete, Rename, OpenLog, CloseLog, Syslog	Events reported by Operating Systems (OS) such as Unix, Linux, ...

SYSTEM CATEGORY	SYSTEM	ACTION	Description
Physical Security	Person, Asset, Status, CommandControl, Guard, Video	CheckIn, CheckOut, Enter, Exit, Access, Login, Logout	Products or Systems that provide security to a geographical zone through the use of physical devices or persons (doors, safes, card reader, fences, assets, guards, …)
Policy Compliance	Manager, Server, Agent	Run_<Check_Id>, Result_<Check_Id>	Products that check versions or configurations in the target system to detect deviation from a pre-established baseline, or detect vulnerabilities from security advisories from CERT, Computer Associates, Microsoft.
Vulnerability Assessment	Manager, Server, Agent	DetectVulnerability, ScanStart, ScanStop, …	Products that simulate attacks to discover security holes or vulnerabilities in the target system or looks for breaches as an hacker would, e.g. password world-writable, …
Vulnerability Management	Manager, Server, Agent	AssetDiscover, AssetDetectVulnerability, …	In addition to Policy Compliance capability, these types of products manage assets and provide remediation suggestions for detected vulnerabilities.

The Result component of Taxonomy has one of the following values:

‘S’ - successful

‘F’ - failed

The Severity component of Taxonomy has one of the following values:

‘N’ - No severity

‘I’ - Informational

‘W’ - Warning

‘C’ - Critical

‘F’ - Fatal

As we have seen, Taxonomy allows us to select easily a set of events of the same type for viewing or further analysis such as correlation. For more specific details concerning an event, you need a data model. This is the subject of the next section.

Data Model

Events are as diverse as a log message, a database record, a hardware device status or a network packet. This diversity comes from different sources reporting events in different formats and with different information content. For example, a CheckPoint firewall reports events as an OPSEC packet, while a Cisco Pix firewall can report events as a syslog message.

When an event is captured by an Audit Recorder, one important processing performed by the latter is to parse the event into attributes and map these attributes to event fields. The totality of event fields is called a data model. A Data model has three (3) different types of event fields as explained below.

As varied as they are, events have a set of attributes common to all events. For example, the event timestamp, location, Audit taxonomy,… They are Audit Common Event Fields.

Audit Taxonomy groups events from different sources, different products and different brands into classes of events. Within each class, there are similar characteristics that are formalized as Audit Normalized Event Fields.

Finally, each product has its own set of attributes that are specific to the product itself. They are called Audit Product-Specific Event Fields.

Audit Common Event Fields

The set of common fields that each event must have are:

Date – Event timestamp. The time when the event is occurred or captured. The Date field is stored in number of seconds since January 1, 1970 at 00:00:00 UTC. Default: The time when the recorder receives the event.

TimeZone – Local time zone expressed in number of seconds. This number is negative if the time zone is East of UTC time zone and positive if it is West of UTC time zone. Default: The timezone of the recorder's host.

Taxonomy – Audit event classification system. Default: General.General.Unknown.N.I

Src – This field defines the origination point of the event. For example, the name of the subsystem that creates the event: Disk, NETLOGON, telnetd, ftpd, or the pathname of the log file. Default: no default.

Log – Logical name of the auditing device. This is a classification of event used by Audit 1.5 or earlier, based on the product. For example: NT-System, NT-Application, NT-Security, Unix, eTrust AC, Oracle, MSSQL, CheckPoint FW-1, … Default: no default.

Location – Host name or IP address or Windows UNC of the SOURCE host. Default: The location of the host running the recorder.

RecorderHost – Host name or IP address of the Recorder host. Default: The location of the host running the recorder.

AuditRouter – Host name of the first Audit Router that sends events.

Audit Normalized Event Fields

Normalized Event Fields depend on taxonomy or more specifically to the pair (SystemCategory, System) components of taxonomy.

Normalized Event Fields are like Common Event Fields. Events that fall in a certain taxonomy must have Normalized Event Fields.

Each Recorder Reference Guide documents in detail the adopted taxonomy and normalized event fields.

Here are some frequently used normalized event fields:

Status – Event status. Possible values:

> ‘S’ – Success
>
> ‘F’ – Failed
>
> ‘D’ – Denied (this status is maintained for backward compatibility with Audit 1.5 and earlier)
>
> Default: S

Severity – The severity level defined as follows:

> ‘0’ – Informational
>
> ‘1’ – Warning
>
> ‘2’ – Critical
>
> ‘3’ – Fatal
>
> Default: 0

Category – Audit 1.5 extensible classification of events modeled after NT security events. There are 11 pre-defined categories:

> Category 1: System Access
>
> Category 2: Account Management
>
> Category 3: Object Access
>
> Category 4: Policy Management
>
> Category 5: Security System Status
>
> Category 6: Network
>
> Category 7: Detailed Tracking
>
> Category 8: Physical Security
>
> Category 9: System \ Application
>
> Category 10: Administration

Category 11: General

Other categories can be defined at will.

Default: no default

Audit r8 maintained this field for backward compatibility with Audit 1.5 or earlier versions.

Note: This field is not related to the Taxonomy's SystemCategory component.

Type – This field is set at the Audit Recorder or Audit Router level to differentiate between Alert event and non-Alert event. Specifically, if the field is set to "Alert", the event is sent as an Alert event. By default, any event generated by a policy rule is an "Alert" event unless set otherwise. This field is used to derive the EVENTTYPE database field, which appears as an icon under the Type column of the Audit Viewer and the Security Monitor screen. See section "Correspondence between Event Field and Audit Database Field".

NID – This field typically contains the event code from NT. However, it can be used for other events as well. Default: 0

User – The user that generates the event. This field is frequently specified when Action component of Taxonomy is Login.

LogF – Pathname of the log file.

AuditRoute – List hostname of all routers that received the event (format: host1->host2-> …). Note the first router that received the event is recorded in AuditRouter common event field.

Check_Id – Vulnerability identification according to some popular vulnerability databases, such as CVE, CAN, CERT, or custome id.

Check_Desc – Description of the vulnerability.

SrcHost – Host name of the source host in a peer-to-peer communication, such as client-server communication.

SrcIP – IP address of SrcHost

SrcPort – TCP or UDP port

DstHost – Host name of the destination host in a peer-to-peer communication.

DstIP – IP address of DstHost

DstPort – TCP or UDP port

Protocol – IP protocol

Term – terminal port, tty device used in interactive communication

EffectiveUser – Remote or Real user behind a remote login such as telnet or executing a command on behalf of User (who generates the event).

Info – a free form text describing the event.

Audit Product-Specific Event Fields

These fields are specific to the product, device and appliance that generates the event. The Recorder Reference Guide details these fields.

The following rules must be used when using product-specific event fields:

1. Naming Convention

   Field names must be chosen to be database-friendly since Audit will store events in its database.

   Following is the list of characters that can be used in field names:

   - Lowercase Latin alphabetic letters
   - Uppercase Latin alphabetic letters
   - Digits: 0 through 9
   - Underscore '_'

   Field name must begin with an alphabetic letter.

2. Conflict avoidance with keywords and existing database fields

   To avoid conflict with existing database fields and keywords, it is strongly recommended to prefix your fields with an abbreviated product name, for example, ePC_User, ePC_Status.

Correspondence Between Event Field and Database Field

In Audit, there is no one-to-one correspondence between event field and database field for 2 primary reasons:

- Events are far richer than a set of limited, finite database fields.
- Some event field name, such as Date, cannot be used as a database column name since it is a reserved keyword.

Audit r8 database defines the following twelve (12) fields:

ENTRYID

DOMAINNAME

USERNAME

EVENTTYPE

LOGNAME

TIMSTAMP

SOURCE

COMPUTERNAME

EVENTID

EVENTCATEGORY

SEARCHSTRINGS

MSGTEXT

The distinction between event fields and database fields in Audit is quite simple:

- Event fields are used to map event's attributes. They are also used in policy rules.
- Database fields are visible through the Audit Viewer, Reporter, Security Monitor and Audit Administrator's Visualizer queries. They are used in building filter criteria.
- Database fields should never be used in mapping event fields. Audit automatically converts event fields into database fields.

The following table describes the correspondence between the two fields:

Event Field ('-' means no equivalent)	Database field	Description

Event Field ('-' means no equivalent)	Database field	Description
Date	TIMSTAMP	Time when the event occurs or is captured. The Date field is stored in number of seconds since January 1, 1970 at 00:00:00 UTC. The TIMSTAMP field has the same value as Date field; however, it is stored in the database using the datetime type of the underlying database engine.
Log	LOGNAME	Logical name of the auditing device. This is a classification of event used by Audit 1.5 or earlier, based on the product. For example: NT-System, NT-Application, NT-Security, Unix, eTrust AC, Oracle, MSSQL, Check Point FW-1, …
Location	COMPUTERNAME DOMAINNAME	Host name or IP address of the SOURCE host. If Location is a fully qualified domain name of format hostname.domain.domain… or Location is a UNC name of format \\NTDomain\Computername, COMPUTERNAME is assigned the value of hostname or Computername, and DOMAINNAME will have NTDomain or domain.domain….

Event Field ('-' means no equivalent)	Database field	Description
Src	SOURCE	This field defines the origination point of the event. For example, the name of the subsystem that creates the event: Disk, NETLOGON, telnetd, ftpd, or the pathname of the log file.
Category	EVENTCATEGORY	Audit 1.5 extensible classification of events modeled after NT security events. Note: this field is not related to the SystemCategory component of Audit taxonomy
NID	EVENTID	This field typically contains the event code from NT. however, it can be used for other events as well.
User	USERNAME	The user that generates the event.
-	EVENTTYPE	This field is derived from Type, EVENTCATEGORY, Status and Severity fields of an event. The value of this field is represented as an icon in Security Monitor and Audit Viewer. See EVENTTYPE section below.

Event Field ('-' means no equivalent)	Database field	Description
-	MSGTEXT	Free-form text field with a big size (typically from 4K to 2G bytes). The data type for this field is CLOB (Oracle), ntext (SQL and Access). This field contains all event fields that do not have a correspondence with one of the 12 database fields for Audit. The content of this field is a multi-line entry with following two possible line formats:1. free-form text with no event field. This format is used to store NT event description. It is maintained only for backward compatibility with earlier versions of Audit. This format becomes obsolete in future versions of Audit.2. field-value pair format as follows:event_field_name<spaces or TABs><TAB>: event_field_valueDue to its size, MSGTEXT is not suitable for searching. In fact searching for an event-specific information through MSGTEXT is very time consuming because DBMS will scan through the whole Audit database

Event Field ('-' means no equivalent)	Database field	Description
-	SEARCHSTRINGS	Contains selected keywords from MSGTEXT. It contains notably the Taxonomy value. This is also a text field but with limited length. Its main use is for faster search and query without having to search through MSGTEXT.
-	ENTRYID	Unique identifier for the specific row.

EVENTTYPE

EVENTTYPE is a database field with value derived from Type, Category, Status and Severity event fields.

To recall:

Type – This field is set at the Audit Recorder or Audit Router level to differentiate between Alert event and non-Alert event. Specifically, if the field is set to "Alert", the event will be sent as an Alert event. By default, any event generated by a policy rule is an "Alert" event unless set otherwise. This field is used to derive the EVENTTYPE field, which appears as an icon under the Type column of the Audit Viewer and the Security Monitor screen.

Category – Audit 1.5 extensible classification of events modeled after NT security events. There are 11 pre-defined categories:

Category 12: System Access

Category 13: Account Management

Category 14: Object Access

Category 15: Policy Management

Category 16: Security System Status

Category 17: Network

Category 18: Detailed Tracking

Category 19: Physical Security

Category 20: System \ Application

Category 21: Administration

Category 22: General

Status – Event status. Possible values:

'**S**' – Success

'**F**' – Failed

'**D**' – Denied (this status is maintained for backward compatibility with Audit 1.5 and earlier)

Default: S

Severity – The severity level defined as follows:

'**0**' – Informational

'**1**' – Warning

'**2**' – Critical

'**3**' – Fatal

EVENTTYPE is computed as follows:

Alert Type:

EVENTTYPE	ICON	SEVERITY
Alert Information	Exclamation mark in blue triangle	0
Alert Warning	Exclamation mark in yellow triangle	1
Alert Critical	Exclamation mark in pink triangle	2
Alert Fatal	Exclamation mark in red triangle	3

Non-Alert Type with event in Category 1 through Category 4:

EVENTTYPE	ICON	STATUS	SEVERITY
Success	Yellow key icon	'S'	No severity or severity = 0
Warning	Exclamation mark in yellow circle	'S'	1
Failure	Key lock icon	'F'	No severity or severity = 0 or 1
Error	Stop sign (red hexagonal icon)	'S', 'F', or 'D'	Severity = 2 or 3

Non-Alert Type with event in Category 5 or above:

EVENTTYPE	ICON	STATUS	SEVERITY
Success	Yellow key icon	'S'	No severity
Information	Letter 'i' in blue circle	'S', 'F' or 'D'	0
Warning	Exclamation mark in yellow circle	'S', 'F' or 'D'	1
Failure	Key lock icon	'F' or 'D'	No severity
Error	Stop sign (red hexagonal icon)	'S', 'F' or 'D'	Severity - 2 or 3

Chapter 13: Post Collection Utility

This section contains the following topics:

Introduction

The Post Collection Utility (PCU) is a component of the eTrust Audit Data Tools. The PCU provides a set of post-collection utilities to enhance your use of the data stored in the Collector database. These utilities let you do the following:

- *Load Policies* that let you expand eTrust Audit event data into individual entries
- *View Policies* that let you build logical views so any expanded data is selectable
- *Tamper Policies* that detect event tampering by digitally signing and verifying signatures on collected events
- *Prune Policies* that let you manage the size and contents of the collector database
- *Sign Policies* that let you handle incoming events by initial digital signatures

The Post Collection Utility consists of the following components:

- A web-based user interface for defining policies and viewing status
- A back-end service which implements the policies on the Collector database

Note: In earlier releases, the PCU was called the "Asynchronous Reporting, Information Extraction, and Signing" program or ARIES. Some of the files still use that designation.

Post Collection Utility Concepts

eTrust Audit Database uses one main table as the primary storage component of its event collection architecture. This design of the Collector Database offers many advantages as well as some limitations.

Enhancement to eTrust Audit Collector Database

Advantages

- Very fast insertion of events into the database.
- Storage of virtually any type of event data in the database. A very large database column, MSGTEXT, will contain any event's attributes that cannot go into Audit's predefined fields.
- Fast access to events based on indexed predefined fields with the rest of event definition viewable through the free form MSGTEXT field.

Limitations

- No direct access to event-specific information contained in MSGTEXT field using standard SQL language or off-the-shelf reporting tools.
- No pruning capability
- No protection against data tampering

Solution

eTrust Audit Post Collection Utility provides the following components to solve the above problems:

- A Post Collection Reporting system, which expands the MSGTEXT field of the database into discrete columns to facilitate report generation.
- A database pruning system to prune old data using various selection criteria.
- A signing mechanism using digital signature technology to keep events from being tampered with.
- A verification process to detect tampering.

Data Flow between PCU and eTrust Audit Components

Post Collection Utility consists of the following components:

- An iTechnology-based iSponsor known as the PCU Engine that provides the back-end service support.
- An iTechnology-based Spindle known as the PCU User Interface provides a web-based administrative interface.

The PCU Engine corresponds with the Collector database through an ODBC connection. Two additional PCU tables called the AuditExtendString table and the AuditSign table assist in analyzing events contained in the Collector database's SEOSDATA table. Views allow you to organize data as you need to see it.

Types of Event Fields in the eTrust Audit Collector Database

The eTrust Audit Collector database (also known as the SEOSDATA table) is the repository for a wealth of information collected from many varied information sources including eTrust products, other CA products, third-party products, and end user solutions. Before information is stored in the eTrust Audit Collector database, it is tokenized into fields in order to identify the event being stored.

There are three types of event fields in eTrust Audit: required fields, recommended fields, and fields specific to other products. The following list provides each of these fields organized according to type:

Required Fields

The required fields are actual database columns, and are as follows:

ENTRYID

A numeric record ID assigned by the database. This field is indexed.

EVENTID

A numeric ID for a class of events (such as NT Event ID or other native event ID). This field is indexed.

LOGNAME

A logical name that uniquely identifies the auditing device. For example, Unix, NT-System, NT-Application, ACF2, Top Secret, or eTrust AntiVirus.

SOURCE

A string value to identify the component of the product that generated the event.

COMPUTERNAME

The host name of the system where the event originated. This field is indexed.

DOMAINNAME

The domain name of the system where the event originated.

TIMSTAMP

The date and time when the event occurred. This field is indexed.

EVENTTYPE

Field that can be set to one of the following values: Failure, Error, Success, Information, Warning. These values represent a combination of Status and Severity.

EVENTCATEGORY

Predefined names for various event groupings.

USERNAME

The name of the user who performed the audited event. This field is indexed.

SEARCHSTRING

Special keywords normally extracted from the MSGTEXT field data.

MSGTEXT

A very large field containing recommended or product-specific fields in the form of name-value pairs.

Recommended Fields Defined for the Category

Certain event categories frequently found in the auditing area are conveniently predefined in eTrust Audit, along with a set of popular field names or recommended field names. To provide consistent eTrust Audit reports and facilitate post-collection analysis activities, it is strongly recommended these fields be used in events of the same categories. These fields are added to the MSGTEXT field as name-value pairs.

Other Product Specific Fields

Other fields (non-eTrust Audit defined) can also be stored in the MSGTEXT field as name-value pairs. These fields are exclusively specific to a product.

Some required fields are indexed, allowing rapid searching using standard SQL statements. All required fields or collector database columns, including MSGTEXT, can be used as part of a SQL *where* clause for searching and locating events of interest.

Recommended fields and product-specific fields are stored in the MSGTEXT field. The MSGTEXT field is used to store whatever information that the developer of an eTrust Audit recorder (which is gathering events for eTrust Audit) decides to include in the event.

How to Install Post Collection Utility

Post Collection Utility is now bundled with the Data Tools installation.

The topics in this chapter describe installation requirements and how to install Post Collection Utility.

Prerequisites

Review the following information concerning installation prerequisites before you install Post Collection Utility:

ODBC Prerequisites

If you install Post Collection Utility on the same host where the eTrust AuditCollector database is installed, you might not need to create a new DSN to access the local eTrust Audit Collector database. You can select the eAudit_DSN during the installation of Post Collection Utility.

However, if you install Post Collection Utility on another system, you must configure MS ODBC to create a system Data Source Name (DSN) to access the eTrust Audit Collector Database.

Database Space Prerequisites

On the host where eTrust Audit Collector database is installed, you need additional disk space for the AuditExtendString table. The size of this table could be as large as the SEOSDATA table.

System Prerequisites

The host where Post Collection Utility is installed should meet the following requirements:

- X86 PC running Windows 2000 with Service Pack 3; or Windows XP Professional with SP1, Windows 2003
- MS Internet Explorer 6.0 SP1 or higher
- 512 MB RAM
- Microsoft ODBC

You can install Post Collection Utility on the same Windows host where the eTrust Audit Data Tools components reside, or on the eTrust Policy Manager host.

Verify Post Collection Utility Installation

To verify that Post Collection Utility is properly installed

1. Verify that the following files are present in the \Program Files\CA\SharedComponents\iTechnology directory:
   - HIDSV.pol
   - HSECURITYV.pol
   - NETACCESSV.pol
   - NETIDSV.pol
   - NETSECURITYV.pol
   - NTSECV.pol
   - igateway.exe
   - igateway.conf
   - igCert.p12
   - icontrol.dll
   - icontrol.conf
   - aries.dll
   - aries.conf
   - Aries
   - Aries_ctrl.log
   - AriesAdmin.dll
   - ariesSpindle.dll
   - ariestool.exe
   - cleanStart.pol
   - FilterLoad.pol
   - LoadPCM.pol
   - Pruneaccess.pol
   - PrunePCM.pol
   - ReloadAll.pol
   - ViewAll.pol
   - ViewExclude.pol
   - ViewInclude.pol
   - ViewPCM.pol
   - ViewPCM_Acct.pol

- ViewPCM_BASELINE.pol
- ViewPCM_Files.pol
- ViewPCM_GrpSumm.pol
- ViewPCM_OutOfPol.pol
- ViewPCM_PrivAcct.pol
- ViewPCM_PWD.pol
- ViewPCM_SecAdv.pol
- ViewPCM_SummAud
- ViewPCM_SummMon
- ViewPCM_UnproSys.pol
- ViewPCM_WinServ.pol
- VMV.pol

2. Start the Windows Services (Computer Management Services) applet and verify that the following service is running:

   - iTechnology – iGateway 4.1

   If all of these files and services are found, the installation of the PCU is successful.

You can now start the PCU and define Load and View policies.

Start and Setup Post Collection Utility

The following topics describe how to start the Post Collection Utility web interface and create a job.

Start Post Collection Utility

You must start the Post Collection Utility before you can run any of the utilities to create the policies.

To start the PCU

1. Access the Post Collection Utility using the Audit Administrator user interface:

   https://localhost:5250/spin/auditadmin

   Replace *localhost* with the name of the system on which the Post Collection Utility is installed.

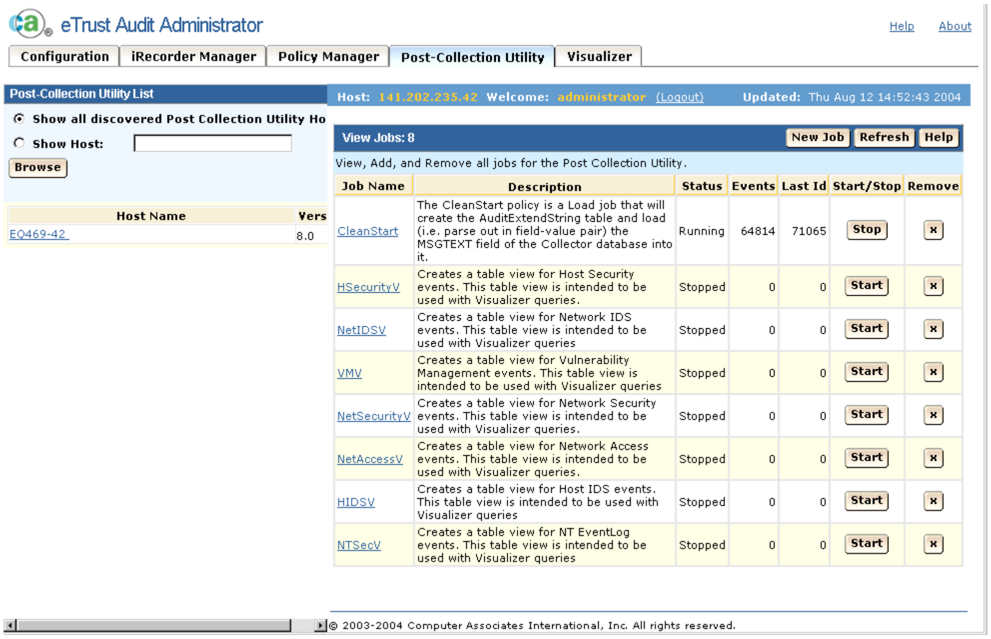
2. Click Login in the upper left to open the Login window.

   The Login window appears.

3. Enter your user name, password and host in the applicable text boxes, and click Login.

   Note: The user must exist on your local server or the domain server, and have Administrator privileges. An account without Administrator privileges has only viewing capabilities.

   You are now logged into the Post Collection Utility through Audit Administrator. Select the Post Collection Utility tab, and the following window appears showing the current job status:



● Show all discovered Post Collection Utility Ho

○ Show Host:

Host Name	Vers
EQ469-42	8.0

Host: 141.202.235.42 Welcome: administrator (Logout) Updated: Thu Aug 12 14:52:43 2004

View Jobs: 8

View, Add, and Remove all jobs for the Post Collection Utility.

Job Name	Description	Status	Events	Last Id	Start/Stop	Remove
CleanStart	The CleanStart policy is a Load job that will create the AuditExtendString table and load (i.e. parse out in field-value pair) the MSGTEXT field of the Collector database into it.	Running	64814	71065	Stop	×
HSecurityV	Creates a table view for Host Security events. This table view is intended to be used with Visualizer queries.	Stopped	0	0	Start	×
NetIDSV	Creates a table view for Network IDS events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×
VMV	Creates a table view for Vulnerability Management events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×
NetSecurityV	Creates a table view for Network Security events. This table view is intended to be used with Visualizer queries.	Stopped	0	0	Start	×
NetAccessV	Creates a table view for Network Access events. This table view is intended to be used with Visualizer queries.	Stopped	0	0	Start	×
HIDSV	Creates a table view for Host IDS events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×
NTSecV	Creates a table view for NT EventLog events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×

© 2003-2004 Computer Associates International, Inc. All rights reserved.

You are now ready to create or monitor your jobs.

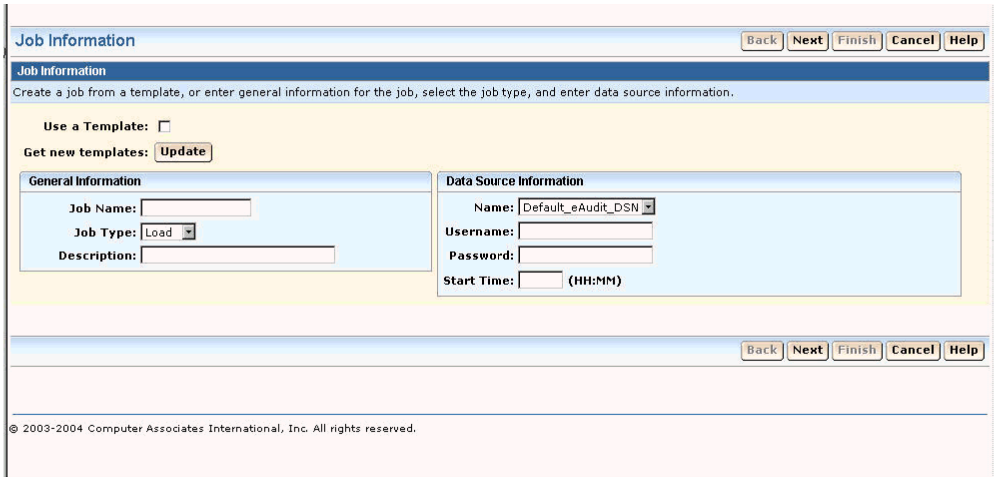
Create Jobs to Run Utilities

All Post Collection Utility utilities are defined as jobs. Therefore, to run any of the utilities, you must create a job.

To create a job from the Post Collection Utility List window, follow these steps:

1. Click New Job.

   The Post Collection Utility List Job Information window appears:



Job Information

Create a job from a template, or enter general information for the job, select the job type, and enter data source information.

Use a Template: □

Get new templates: Update

General Information

Job Name:

Job Type: Load

Description:

Data Source Information

Name: Default_eAudit_DSN

Username:

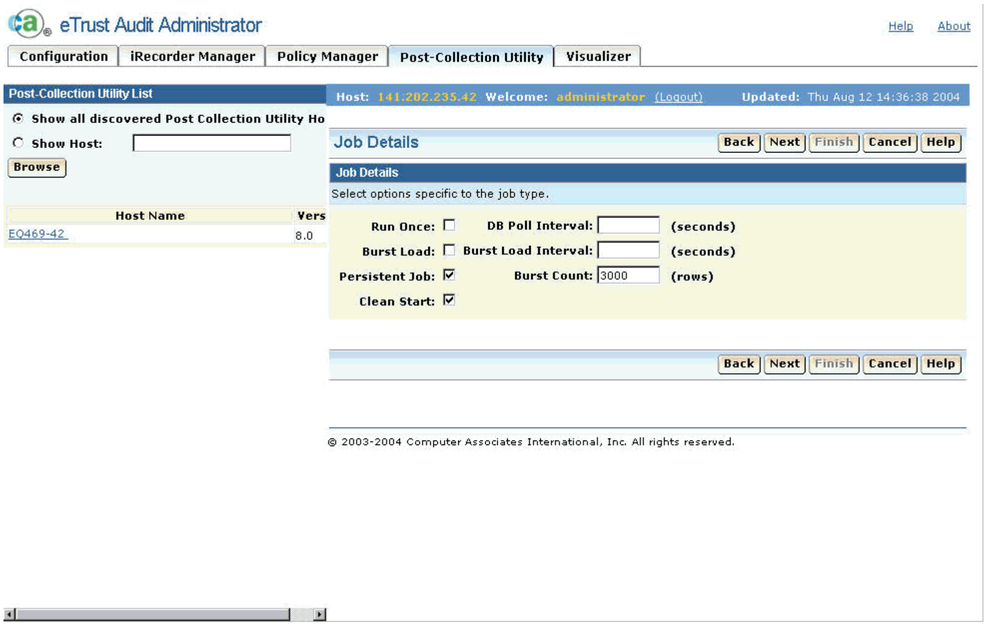
Password:

Start Time: (HH:MM)

© 2003-2004 Computer Associates International, Inc. All rights reserved.

2. Complete the fields on the window, and then click Next.

   The following window appears:



Post-Collection Utility List

● Show all discovered Post Collection Utility Ho

○ Show Host:

Host Name	Vers
EQ469-42	8.0

Host: 141.202.235.42 Welcome: administrator (Logout) Updated: Thu Aug 12 14:36:38 2004

Job Details

Select options specific to the job type.

Run Once: □ DB Poll Interval: (seconds)

Burst Load: □ Burst Load Interval: (seconds)

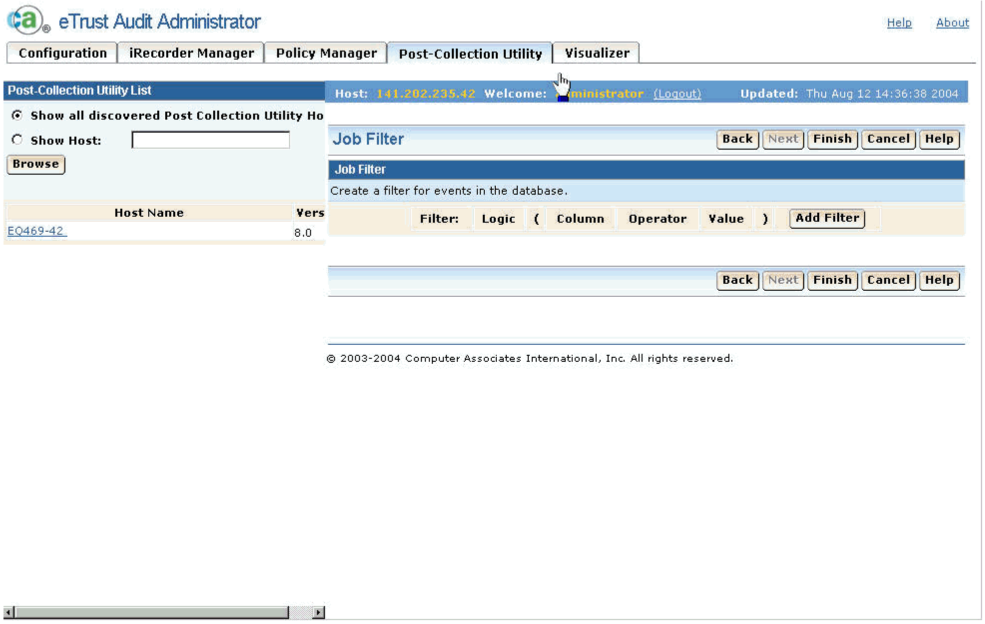
Persistent Job: ☑ Burst Count: 3000 (rows)

Clean Start: ☑

© 2003-2004 Computer Associates International, Inc. All rights reserved.

3. Check the applicable boxes, then click Next.

   The following window appears:



● Show all discovered Post Collection Utility Ho

○ Show Host:

Host Name	Vers
EQ469-42	8.0

Host: 141.202.235.42 Welcome: ministrator (Logout) Updated: Thu Aug 12 14:36:38 2004

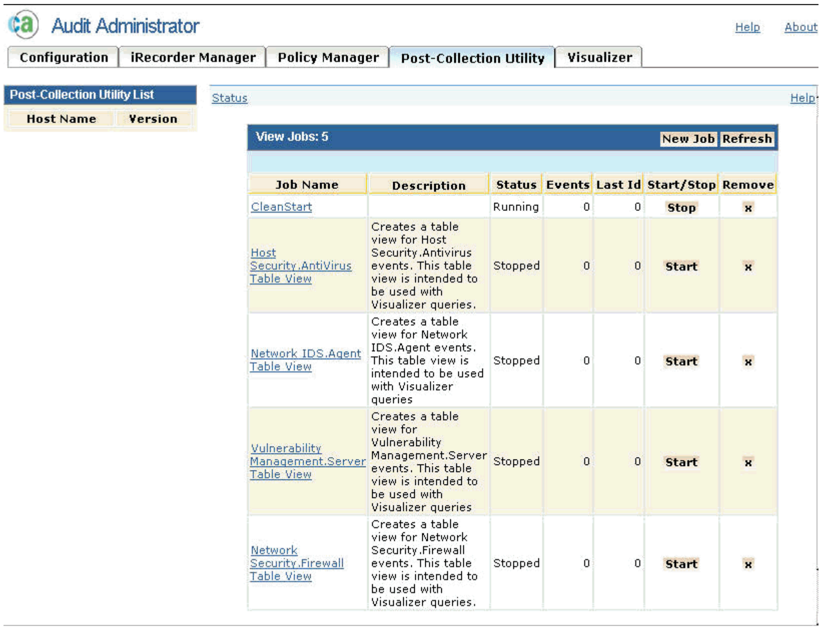
Job Filter

Create a filter for events in the database.

Filter: Logic (Column Operator Value)

© 2003-2004 Computer Associates International, Inc. All rights reserved.

4. Click Finish, and the Post Collection Utility List Status is updated with the new job.



View Jobs: 5

Job Name	Description	Status	Events	Last Id	Start/Stop	Remove
CleanStart		Running	0	0	Stop	×
Host Security.AntiVirus Table View	Creates a table view for Host Security.Antivirus events. This table view is intended to be used with Visualizer queries.	Stopped	0	0	Start	×
Network IDS.Agent Table View	Creates a table view for Network IDS.Agent events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×
Vulnerability Management.Server Table View	Creates a table view for Vulnerability Management.Server events. This table view is intended to be used with Visualizer queries	Stopped	0	0	Start	×
Network Security.Firewall Table View	Creates a table view for Network Security.Firewall events. This table view is intended to be used with Visualizer queries.	Stopped	0	0	Start	×

Note: Depending on the size of your database and the options you selected, it might take several minutes to load the events.

When your job is completed, click Refresh, and the status changes to Stopped.

See the following sections for information on using the various Post Collection Utility utilities.

Post Collection Utility Job Window Field Descriptions

You must use the Define a New Post Collection Utility Job window to create any Post Collection Utility job and use the utilities.

Note: When you schedule a job, it does not start its run automatically. You must click the Start button to start the job, and then it runs at the scheduled time.

The fields on the window are as follows:

Use predefined policies

Check this box to display a drop-down list of predefined policies installed with Post Collection Utility. If you select a policy from the list, some of the fields in the rest of the window are populated with values defined in the selected policy. For example, if you choose FilterLoad.pol, Load is specified as the value of the Type field, indicating the type of job that you want to run.

Name

Specifies a unique name for the job.

Description

Specifies additional information about the purpose of the job.

Type

Specifies the type of job that you want to run. Valid values are as follows:

Load

View

Prune

Sign

Verify

Data Source User Name

Specifies a user ID with authorized access to the eTrust Audit Collector database. When you choose a data source name, the value of this field is populated with a default value. You can enter a new value if the user ID provided does not meet your site specifications.

Data Source Password

Specifies the password for the Data Source User Name. No default value is provided. You must specify the password. Asterisk characters are used to mask the password you are entering.

Initial Start Time (HH:MM)

Specifies the time (in hours:minutes) when you want the job to start. Other fields on the dialog let you specify whether the job will run only once or at regular intervals. If you leave this field empty, you can only start the job from the Status window.

Note: Depending on the size of your eTrust Audit Collector database and the criteria you specify for a filter, you might want to set the value of Initial Start Time to off hours when there is less impact on the database performance.

Run Once?

Check this box if you want this job to run only once. After it is finished, the job exits.

DB Poll Interval in Seconds

Specifies the interval (in seconds) before Post Collection Utility checks specific tables for new rows.

Note: This field cannot be edited if the RunOnce field is checked.

The specific tables depend on the type of job you have specified in the Type field:

Load

The Post Collection Utility Engine checks the SEOSDATA table for new events and loads the AuditExtendString table with the new events that match the filter criteria.

View

The Post Collection Utility Engine checks the AuditExtendString table for any new data that match the filter criteria (added by a load job), including new field names and events.

Prune

The Post Collection Utility Engine checks the SEOSDATA table and the AuditExtendString table and prunes events that match the filter criteria.

Sign

The Post Collection Utility Engine checks the SEOSDATA table and signs the events based on the last ID processed in a previous job.

Verify

The Post Collection Utility Engine checks the SEOSDATA table and verifies that all the signed events have not been tampered with since the last verify job.

Persistent Job?

Check this box to save the job's configuration parameters. When a job is marked persistent, the job is automatically loaded and run by Post Collection Utility every time the machine is rebooted or the iGateway service is restarted.

Clean Target Table Before Start?

Check this box to specify that you want to drop the AuditExtendString table and create a new one before the job runs.

Burst Load?

Check this box to start Post Collection Utility in burst mode, causing Post Collection Utility to access the database only for the finite interval specified in the Burst Load Interval field.

Burst Load Interval (in seconds)

Specify an interval (in seconds) that limits the amount of time Post Collection Utility is working in the database. Using less time will take Post Collection Utility longer to finish the process, but other processes can better share the eTrust Audit database. Using more time will make the Post Collection Utility process the priority. The default value is one-half the value specified in the DB Poll Interval field.

Filter: Logic (Col Oper Val) Add Filter

Click the Filter button to specify criteria to identify the data that you want the job to process. After you click Add Filter, you can specify the filter criteria using the following:

logic

Specifies additional logic when you add more than one filter. Select AND or OR from the drop-down.

left parentheses

Specifies the number of left parentheses you want to use.

Col

Type the name of one of the required columns from the SEOSDATA table.

Oper

Select one of the following operators from the drop-down:

EQUAL

NEQ (not equal)

LESS (less than)

LEQ (less than or equal to)

GREATER (greater than)

GREATEQ (greater than or equal to)

LIKE

NOTLIKE

WITHINSET

NOTINSET

Val

Type the value of the column.

right parentheses

Specifies the number of right parentheses you want to use.

Cancel Button

Click Cancel to close the window without generating the XML to run the job.

Post Collection Utility and Policies

The Post Collection Utility utilities work with the event data after it has been collected and stored in the database. You run the utilities using the Post Collection Utility web interface. The utilities are as follows:

- Load Policies (see page 271) let you expand eTrust Audit event data into individual entries.
- View Policies (see page 273) let you build logical views so all expanded data is selectable.
- Tamper Policies (see page 276) let you detect event tampering by digitally signing and verifying signatures on collected events.
- Prune Policies (see page 275) let you manage the size of the collector database.

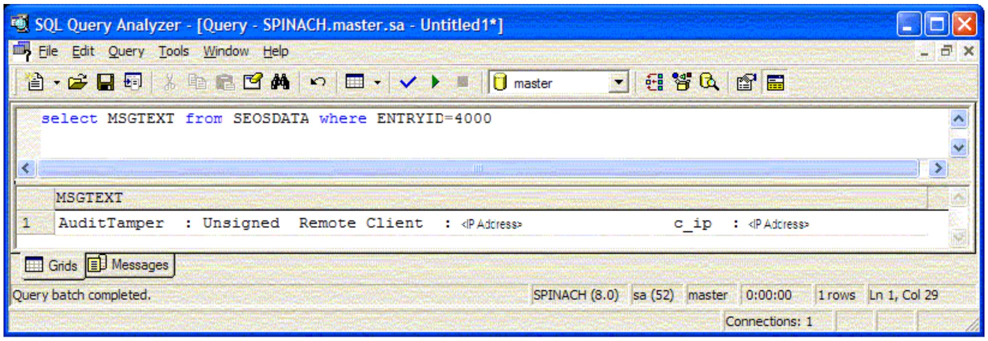
Load Policies

Load policies expand the MSGTEXT field into individual entries. These entries can be converted in database columns through views and then are used in normal SQL queries or customized reports through external reporting tools.

After you click Create on the Define a new Post Collection Utility Job window, Post Collection Utility constructs a Load job as follows:

- Gives it the name and description specified in Name and Description so that you can view its status.
- Starts at the time you specified for Initial Start Time, or if you did not specify an initial start time, when you start the job manually.
- Connects to the ODBC data source (using Data Source Name, Type, User Name, and Password)
- Selects each SEOSDATA event that is greater than the last processed ID and that matches the criteria specified in the filters.
- Keeps track of last SEOSDATA.ENTRYID that was processed for future jobs.
- Parses the MSGTEXT field into multiple entries, consisting of the following:
  - The ENTRYID of the event, specified in SEOSDATA.entryid
  - The column name from the MSGTEXT field
  - The value from the MSGTEXT field
- Adds each entry as a row into the new AuditExtendString table.
- If running in burst mode, waits for the Burst Load Interval, then commits the Select operation.
- Waits for the DB Poll Interval and then runs the same job again.

The following is a sample of a MSGTEXT column in the SEOSDATA table:


SQL Query Analyzer - [Query - SPINACH.master.sa - Untitled1*]
File
Edit
Query
Tools
Window
Help
master
Grids
Messages

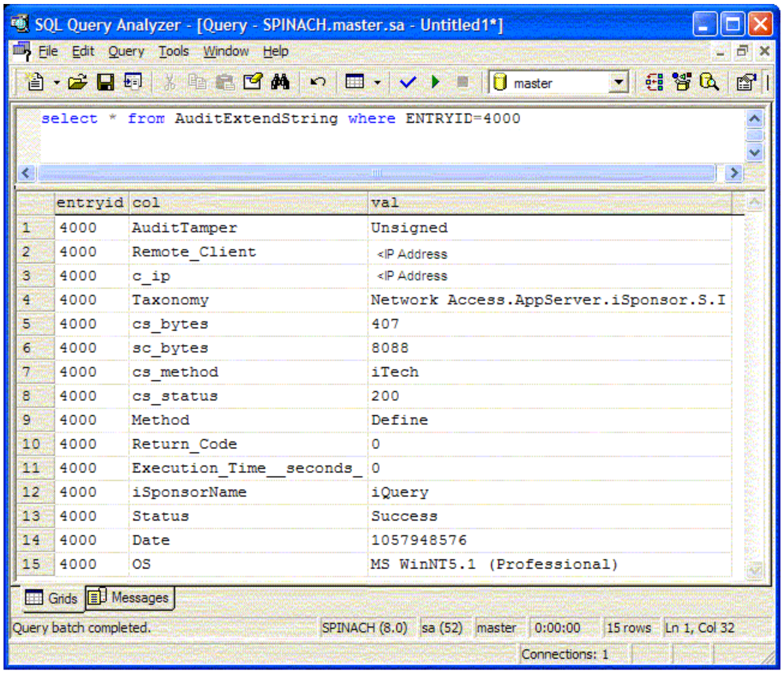
```
select MSGTEXT from SEOSDATA where ENTRYID=4000
```

```
  MSGTEXT
1 AuditTamper  : Unsigned  Remote Client  : <IP Adress>          c_ip  : <IP Adress>
```

Query batch completed. SPINACH (8.0) sa (52) master 0:00:00 1 rows Ln 1, Col 29

Connections: 1

The following is a sample of a MSGTEXT column expanded into AuditExtendString:



SQL Query Analyzer - [Query - SPINACH.master.sa - Untitled1*]
File Edit Query Tools Window Help
master
Grids
Messages

```
select * from AuditExtendString where ENTRYID=4000
```

	entryid	col	val
1	4000	AuditTamper	Unsigned
2	4000	Remote_Client	<IP Address
3	4000	c_ip	<IP Address
4	4000	Taxonomy	Network Access.AppServer.iSponsor.S.I
5	4000	cs_bytes	407
6	4000	sc_bytes	8088
7	4000	cs_method	iTech
8	4000	cs_status	200
9	4000	Method	Define
10	4000	Return_Code	0
11	4000	Execution_Time__seconds_	0
12	4000	iSponsorName	iQuery
13	4000	Status	Success
14	4000	Date	1057948576
15	4000	OS	MS WinNT5.1 (Professional)

Query batch completed. SPINACH (8.0) sa (52) master 0:00:00 15 rows Ln 1, Col 32

Connections: 1

After the load jobs complete, you can create view polices to convert these entries into selectable columns of the event.

View Policies

After you run a load job to expand the SEOSDATA.MSGTEXT field into multiple entries in the AuditExtendString table, you can create view policies to construct logical views on the data.

You can create views as follows:

- Automatically, based on the event's LOGNAME, containing all possible column names
- Automatically, based on the event's taxonomy, containing all possible column names
- Custom, based on the filters and columns provided in the policy

After you submit the view policy job, Post Collection Utility constructs a view job as follows:

- Gives it the name and description specified in Name and Description so that you can view its status.
- Starts at the time you specified for Initial Start Time, or if you did not specify an initial start time, when you start the job manually.
- After the job completes, it will run again when the value in DB Poll Interval seconds is reached.

The view job runs as follows:

- Connects to the ODBC data source (using Data Source Name, Type, User Name, and Password)
- Processes each SEOSDATA event that is greater than the last processed ID **and** matches the criteria specified in the filters as follows:
  - If you do not specify filters/columns, Post Collection Utility automatically construct logical views on all SEOSDATA and AuditExtendString data. The views are named:

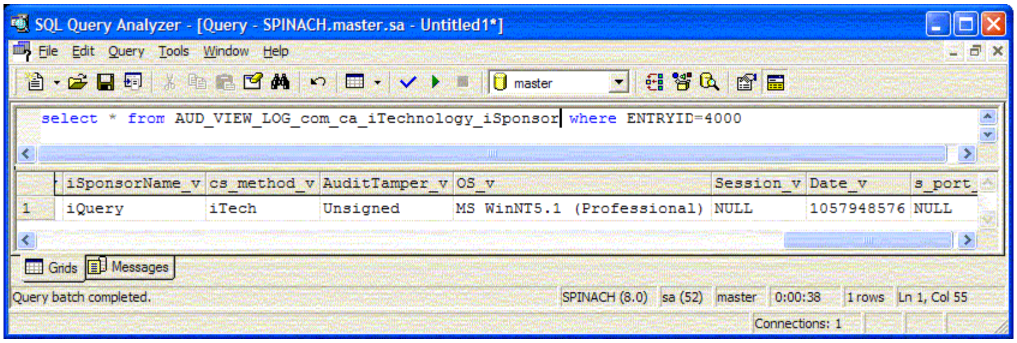
    AUD_VIEW_LOG_{logname}

    AUD_VIEW_TAX_{taxonomy}
  - If you provide filters/columns, Post Collection Utility constructs a single custom logical view, selected by the filters provided, and including/excluding the columns provided. The view is named AUD_VIEW_CUST_{viewname}.

The view job joins the two tables using AuditExtendString.entryid and SEOSDATA.ENTRYID. The new table contains the following:

- All of the columns of the original SEOSDATA
- Each expanded AuditExtendString.{column} as {column_v}

The event shown through the automatically-created view called "AUD_VIEW_LOG_com_ca_iTechnology_iSponsor":



SQL Query Analyzer - [Query - SPINACH.master.sa - Untitled1*]
File Edit Query Tools Window Help
master
Grids
Messages

```
select * from AUD_VIEW_LOG_com_ca_iTechnology_iSponsor where ENTRYID=4000
```

	iSponsorName_v	cs_method_v	AuditTamper_v	OS_v	Session_v	Date_v	s_port_
1	iQuery	iTech	Unsigned	MS WinNT5.1 (Professional)	NULL	1057948576	NULL

Query batch completed. SPINACH (8.0) sa (52) master 0:00:38 1 rows Ln 1, Col 55
Connections: 1

You use the view to select all column fields for the event. For example, you can use it to create reports using third-party report writers.

Prune Policies

Use prune policies to automatically discard events older than a specified date. Prune policies help you manage the size of the collector database through retention policies.

Using the submitted prune policy, Post Collection Utility constructs a prune job as follows:

- Gives it the name and description specified in Name and Description so that you can view its status.
- Starts at the time you specified for Initial Start Time, or if you did not specify an initial start time, when you start the job manually.
- After the job completes, it will run again when the value in DB Poll Interval seconds is reached.

The prune job runs as follows:

- Connects to the ODBC data source (using Data Source Name, Type, User Name, and Password)
- Finds all SEOSDATA entries that meet the following criteria:
  - Were created more than “Prune days” ago, and
  - Meet the criteria specified in the Filters
- If you select Test Prune Policy, the job stops
- Processes each SEOSDATA entry found

The prune job does the following:

- Deletes the SEOSDATA event
- Deletes the AuditExendString entries where AuditExtendString.entryid match the SEOSDATA.entryid
- Deletes the AuditSign entry where AuditSign.entryid matches the SEOSDATA.entryid

Tamper Policies (Sign and Verify)

You use tamper policies to detect post-collection event tampering. Post Collection Utility accomplishes this by digitally signing events, and then verifying the signatures periodically.

There are two types of tamper polices:

Sign Policies

Use sign policies to initially sign incoming events.

Verify Policies

Use verify policies to periodically verify previously signed events.

Sign Policies

Use sign policies to initially apply a digital signature to incoming events.

Using the submitted sign policy, Post Collection Utility constructs a sign job as follows:

- Gives it the name and description specified in Name and Description so that you can view its status.
- Starts at the time you specified for Initial Start Time, or if you did not specify an initial start time, when you start the job manually.
- After the job completes, it will run again when the value in DB Poll Interval seconds is reached.

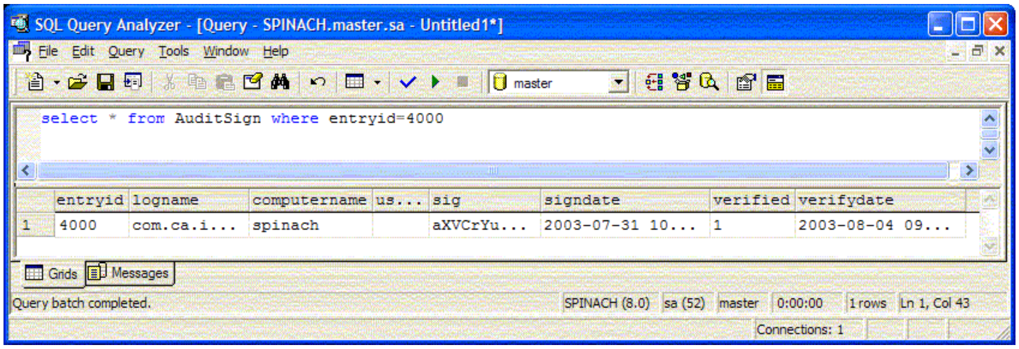
The sign job runs as follows:

- Connects to the ODBC data source (using Data Source Name, Type, User Name, and Password)
- Processes each SEOSDATA event that meets the following criteria:
  - Greater than the last processed ID
  - Has not been previously signed
  - Meets the criteria specified in the filters
- Keeps track of last SEOSDATA.ENTRYID that was processed

The sign job does the following:

- Uses the data in the original event to generate an MD5 hash.
- Uses the Post Collection Utility private key (generated at first install of Post Collection Utility) to digitally sign the MD5 hash.
- Creates an entry in the AuditSign table, consisting of the following information:
  - The event's ENTRYID specified in SEOSDATA.entryid
  - The digital signature
  - The date the event was signed
  - The original event's values in the SEOSDATA columns COMPUTERNAME, USERNAME, and LOGNAME
  - Verified set to 1 in order to mark the event as NON-TAMPERED
  - Verifydate set to the date the event was signed

The following is a sample of an AuditSign entry:


SQL Query Analyzer - [Query - SPINACH.master.sa - Untitled1*]
File Edit Query Tools Window Help
master
Grids
Messages

```
select * from AuditSign where entryid=4000
```

	entryid	logname	computername	us...	sig	signdate	verified	verifydate
1	4000	com.ca.i...	spinach		aXVCrYu...	2003-07-31 10...	1	2003-08-04 09...

Query batch completed. SPINACH (8.0) sa (52) master 0:00:00 1 rows Ln 1, Col 43

Connections: 1

After the event is signed, you can verify its signature using a verify policy.

Verify Policies

Use verify policies to periodically verify previously signed events.

Using the submitted verify policy, Post Collection Utility constructs a verify job as follows:

- Gives it the name and description specified in Name and Description so that you can view its status.
- Starts at the time you specified for Initial Start Time, or if you did not specify an initial start time, when you start the job manually.
- After the job completes, it will run again when the value in DB Poll Interval seconds is reached.

The verify job runs as follows:

- Connects to the ODBC data source (using Data Source Name, Type, User Name, and Password)
- Processes each AuditSign entry that meets the following criteria:
  - Was not processed in the last Verify min hours
  - Was initially signed less than Verify max days ago
  - Is currently considered NON-TAMPERED
  - Meets the criteria specified in the filters

The verify job does the following:

- Uses the data in the original event to generate an MD5 hash.
- Uses the Post Collection Utility private key (generated at first install of Post Collection Utility) to digitally sign the MD5 hash.
- Compares the digital signature to the sig column in the AuditSign table as follows:
  - If the signature matches, mark the event as NON-TAMPERED (AuditSign verified column set to 1)
  - If the signature does not match, mark the event as TAMPERED (AuditSign verified column set to 0)

XML format of Policies

Load Policy File Tags

Load Policy consists of a filter to select events from the Audit Database (SEOSDATA) and load the selected events into AuditExtendString table.

The following sample Load Policy will load records, from SQL Server Audit database, that match LOGNAME=”NT-System” criteria, to AuditExtendString table. The job will start at 3:45 am. When it reaches the end of SEOSDATA table, it will wait for 60 seconds (pollinterval) before checking for new rows in SEOSDATA.

```
<LoaderPolicy>
            <action>load</action>
<source name="eTAudit_DSN" type="SQL Server" user="sa" password="" pollinterval="60" starttime="03:45" />
            <deletefirst>no</deletefirst>
            <filter col="LOGNAME" oper="EQUAL" val="NT-System"/>
</LoaderPolicy>
```

The following tags are in this policy file:

<action>

Type of Policy which can be Load, View, Prune, Sign, Verify

<Source>

name

ODBC Database Source Name (DSN)

Type

Database type: Access, SQL Server, Oracle, Ingres

User

Database logon user

Password

Database logon user password

Pollinterval

Frequency (in seconds) with which the Post Collection Utility Engine will access the SEOSDATA table and load the AuditExtendString table with selected new events.

Starttime

Time at which the policy job will run after it has been loaded by the Post Collection Utility Engine.

Runonce

If set to true, the job will exit after completing its task

Burstload

if set to "true", the job will run in burst mode (see burst interval below)

burstinterval

if burstinterval is not zero and burstload is set to "true", the job will run in the database for burstinterval seconds, then stop for pollinterval.

<deletefirst>

If set to "yes", the AuditExtendString table will be dropped and recreated at the start of the job

<filter>

col

One of the SEOSDATA required column

Oper

Comparison operator: EQUAL, NEQ, LEQ, GREATEQ, LESS, GREATER, LIKE, NOTLIKE, WITHINSET, NOTINSET

Val

Value of the column

[logic]

Optional AND or OR. The first filter does not need a logic attribute

[lparens]

Optional integer specifying how many left parentheses.

[rparens]

Optional integer specifying how many right parentheses.

View Policy File Tags

View Policy defines a virtual table or view on SEOSDATA table and AuditExtendString table. The view will have as columns, all columns from SEOSDATA (except SEARCHSTRINGS and MSGTEXT), and selected col values from AuditExtendString (see How View Policies Function).   View Policy can include a default subset of AuditExtendString.col columns, user-selected AuditExtendString.col columns, or all AuditExtendString.col columns except a few ones.

The following sample View Policy creates a View for events from the audited system MyComputer with SEOSDATA columns and two AuditExtendString.col columns: OS and Info.

```
 <LoaderPolicy>
        <action>view</action>
        <source name="eTAudit_DSN" type="SQL Server" user="sa" password="" runonce="true"
starttime="14:45"/>
<filter col="COMPUTERNAME" oper="EQUAL" val="MyComputer"/>
        <columns type="include">
                <col>OS</col>
                <col>Info</col>
        </columns>
</LoaderPolicy>
```

The following tags are in this policy file:

<action>

Type of Policy which can be Load, View, Prune, Sign, Verify

<Source>

name

ODBC Database Source Name (DSN)

Type

Database type: Access, SQL Server, Oracle, Ingres

User

Database logon user

Password

Database logon user password

Pollinterval

Frequency (in seconds) with which the Post Collection Utility Engine will sample the AuditExtendString table for new col values. You should either specify pollinterval or runonce.

Starttime

Time at which the policy job will run after it has been loaded by the Post Collection Utility Engine.

Runonce

The policy will run once and create the View with field names available. This option is appropriate for Include Rule policy.

Burstload

if set to “true”, the job will run in burst mode (see burst interval below)

burstinterval

if burstinterval is not zero and burstload is set to “true”, the job will run in the database for burstinterval seconds, then stop for pollinterval.

<filter>

col

One of the SEOSDATA required column

Oper

Comparison operator: EQUAL, NEQ, LEQ, GREATEQ, LESS, GREATER, LIKE, NOTLIKE, WITHINSET, NOTINSET

Val

Value of the column

[logic]

Optional AND or OR. The first filter does not need a logic attribute

[lparens]

Optional integer specifying how many left parentheses.

[rparens]

Optional integer specifying how many right parentheses.

<columns>

type

Possible values: include or exclude to specify Include or Exclude Rules. If the <columns> tag is missing, Post Collection Utility will sample the AuditExtendString table to create a View with all field names found in the table.

<col>

AuditExtendString.col values (event-specific fields) to be included or excluded from the View.

Prune Policy File Tags

Prune Policy consists of prune criteria to select events from the Audit database table for removal from Audit database. You can also add filters to further narrow the selected records for pruning.

The following sample prune policy will prune records from an Access database, that are older than 30 days and that match LOGNAME = “iRecorder” criteria.

```
<LoaderPolicy>
        <action>prune</action>
        <source name="eAudit_DSN" type="access" accessfilename="c:\Program Files\CA\eTrust
Audit\database\SEOSDATA.mdb" user="Administrator" password=""/>
        <prune prunedays="30" no-op="false"/>
        <filter col="LOGNAME" oper="EQUAL" val="iRecorder"/>
</LoaderPolicy>
```

The following tags are in this policy file:

<action>

Type of Policy which can be Load, View, Prune, Sign, Verify

<Source>

name

ODBC Database Source Name (DSN)

type

Database type: Access, SQL Server, Oracle, Ingres

User

Database logon user

Password

Database logon user password

Pollinterval

Frequency (in seconds) with which the Post Collection Utility Engine will access the SEOSDATA table, AuditExtendString table and AuditSign table to prune records

Starttime

Time at which the policy job will run after it has been loaded by the Post Collection Utility Engine.

Runonce

If set to true, the job will exit after completing its task

Burstload

Not used (ignored) in the Prune Policy

Burstinterval

Not used (ignored) in the Prune Policy

<deletefirst>

Not used (ignored) in the Prune Policy

<prune>

prunedays

Events older than 'prunedays' in the Audit databases will be pruned. If a filter is defined, only records matching both 'prunedays' and the filter will be deleted.

no-op

If set to 'True', no actual pruning will take place, Post Collection Utility will provide a report containing all events that would be deleted if the 'no-op' was set to 'False'. If set to 'False', the events matching the policy will be actually deleted from the table.

<filter>

col

One of the SEOSDATA required column

oper

Comparison operator: EQUAL, NEQ, LEQ, GREATEQ, LESS, GREATER, LIKE, NOTLIKE, WITHINSET, NOTINSET

Val

Value of the column

[logic]

Optional AND or OR. The first filter does not need a logic attribute

[lparens]

Optional integer specifying how many left parentheses.

[rparens]

Optional integer specifying how many right parentheses.

Note: Multiple policies can be scheduled which will be able to enforce complex retention policies for different sets of events. For example, using advanced filters, important or critical events in the database can be retained for longer periods than the informational events.

Sign Policy File Tags

Sign Policy consists of a filter to select events from the Audit database (SEOSDATA), digitally signs them and records their signature in AuditSign table. (See How Tamper Policies Function for description of AuditSign's row).

The following sample Sign Policy will digitally sign records, from SQL Server Audit database, that match LOGNAME="NT-System" criteria. Signature of the records will be stored in AuditSign table. The job will start at 3:45 am. When it reaches the end of SEOSDATA table, it will wait for 60 seconds (pollinterval) before checking for new rows in SEOSDATA.

```
<LoaderPolicy>
        <action>sign</action>
<source name="eTAudit_DSN" type="SQL Server" user="sa" password="" pollinterval="60" starttime="03:45"/>
        <deletefirst>no</deletefirst>
        <filter col="LOGNAME" oper="EQUAL" val="NT-System"/>
</LoaderPolicy>
```

The following tags are in this policy file:

<action>

Type of Policy which can be Load, View, Prune, Sign, Verify

<Source>

name

ODBC Database Source Name (DSN)

Type

Database type: Access, SQL Server, Oracle, Ingres

User

Database logon user

Password

Database logon user password

Pollinterval

Frequency (in seconds) with which the Post Collection Utility Engine will access the SEOSDATA table for new events to sign.

Starttime

Time at which the policy job will run after it has been loaded by the Post Collection Utility Engine.

Runonce

If set to true, the job will exit after completing its task

Burstload

if set to “true”, the job will run in burst mode (see burst interval below)

burstinterval

if burstinterval is not zero and burstload is set to “true”, the job will run in the database for burstinterval seconds, then stop for pollinterval.

<deletefirst>

If set to “yes”, the AuditSign table will be dropped and recreated at the start of the job

<filter>

col

One of the SEOSDATA required column

Oper

Comparison operator: EQUAL, NEQ, LEQ, GREATEQ, LESS, GREATER, LIKE, NOTLIKE, WITHINSET, NOTINSET

val

Value of the column

[logic]

Optional AND or OR. The first filter does not need a logic attribute

[lparens]

Optional integer specifying how many left parentheses.

[rparens]

Optional integer specifying how many right parentheses.

Verify Policy File Tags

Verify Policy consists of verify criteria to select events from the Audit database for tampering verification. You can also add filters to further narrow the set of records for verification.

The following sample verify policy will verify records from an Access database, that have not been verified in the last 12 hours.

```
<LoaderPolicy>
        <action>verify</action>
        <source name="eAudit_DSN" type="access" accessfilename="c:\Program Files\CA\eTrust
Audit\database\SEOSDATA.mdb" user="Administrator" password=""/>
        <verify verifyminhours="12" verifymaxdays="0"/>
</LoaderPolicy>
```

The following tags are in this policy file:

<action>

Type of Policy which can be Load, View, Prune, Sign, Verify

<Source>

name

ODBC Database Source Name (DSN)

Type

Database type: Access, SQL Server, Oracle, Ingres

User

Database logon user

Password

Database logon user password

Pollinterval

Frequency (in seconds) with which the Post Collection Utility Engine will access the SEOSDATA table and AuditSign table to verify records

Starttime

Time at which the policy job will run after it has been loaded by the Post Collection Utility Engine.

Runonce

If set to true, the job will exit after completing its task

Burstload

if set to “true”, the job will run in burst mode (see burst interval below)

burstinterval

if burstinterval is not zero and burstload is set to “true”, the job will run in the database for burstinterval seconds, then stop for pollinterval.

<deletefirst>

If set to “yes”, the AuditSign table will be dropped and recreated at the start of the job

<verify>

verifyminhours

Verify all records in the last “verifyminhours”. If verifyminhours is 0, verify all records, subject to the verifymaxdays criteria (see below).

Verifymaxdays

Verify all records that have been signed “verifymaxdays” ago. If verifymaxdays is 0, verify all records regardless of when they are signed.

<filter>

col

One of the SEOSDATA required column

Oper

Comparison operator: EQUAL, NEQ, LEQ, GREATEQ, LESS, GREATER, LIKE, NOTLIKE, WITHINSET, NOTINSET

val

Value of the column

[logic]

Optional AND or OR. The first filter does not need a logic attribute

[lparens]

Optional integer specifying how many left parentheses.

[rparens]

Optional integer specifying how many right parentheses.

Chapter 14: iRouter: The Bridge between iRecorder and the Audit Router

This section contains the following topics:

Introduction

The iRouter is an integral component of the eTrust Audit Client for Windows, Solaris, and Linux platforms. It is installed when the Client is installed on these platforms.

About the iRouter

eTrust Audit r8 includes several recorders. These recorders use the SAPI (Submit API) to send events to the eTrust Audit Router (eTrust Audit Client component). eTrust Audit iRecorders use iTechnology communication protocol (HTTPS, Port 5250) to deliver events in XML format.

eTrust Audit iRouter is a middleware component that sits between the iRecorders and the eTrust Audit Router. It receives events in XML format from iRecorders over HTTPS port 5250 and forwards the events to eTrust Audit Router using SAPI.

You can install iRecorders on a remote host or on the same host where the iRouter is installed. iRecorders are totally independent and do not require any eTrust Audit components to be installed on the iRecorder host.

However, you must install the eTrust Audit iRouter on the same host where the eTrust Audit Router is installed (eTrust Audit Action Manager component is always installed with the Router component). In other words, iRouter and eTrust Audit Router cannot communicate if installed on two separate hosts.

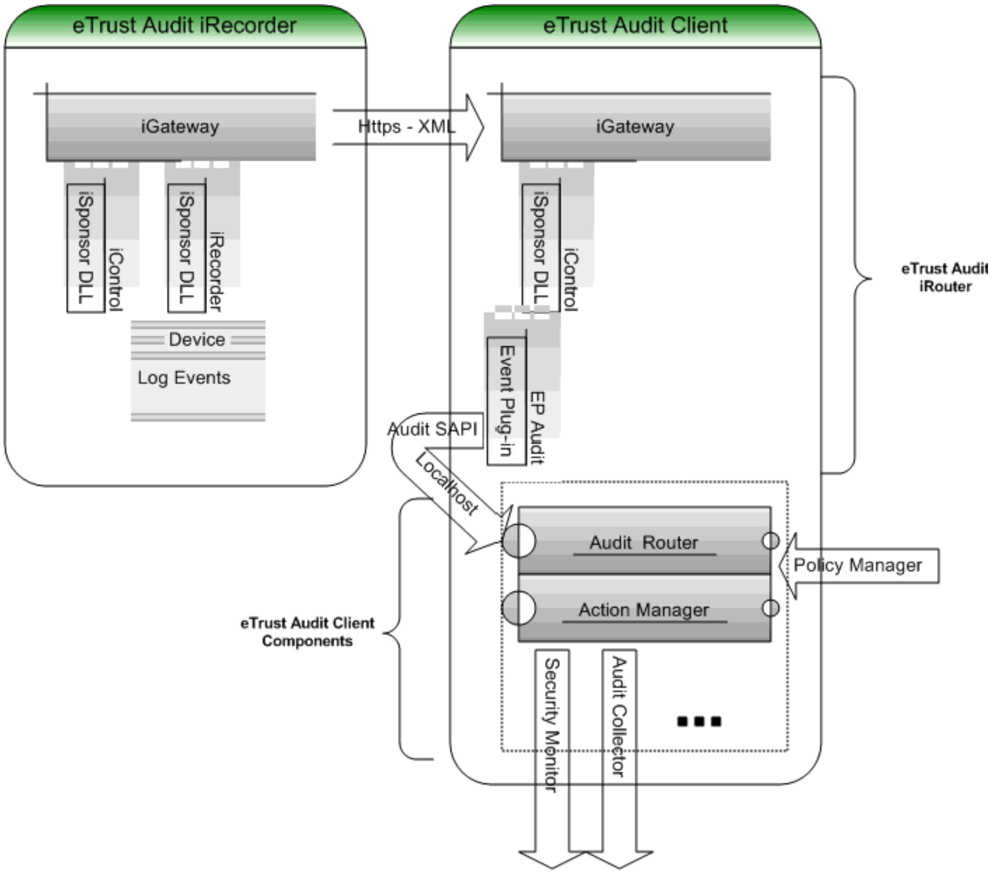
iRecorders and iRouter can be installed on the same host or on two separate hosts.

iRouter Architecture

eTrust Audit r8 recorders can be deployed in two different ways:

1. Traditional eTrust Audit recorders use the eTrust Audit Submit API (SAPI) to send log events to an eTrust Audit Router and eTrust Action Manager for further processing as defined in the eTrust Audit Policy Manager. This architecture leads to some restrictions in the eTrust Audit Recorder development and deployment:
   - Because SAPI uses RPC, the recorders cannot be easily deployed across firewalls.
   - Deployments of new recorders that are not predefined require manual modifications of existing eTrust Audit Routers and Action Managers.
2. iRecorders, based on the iTechnololgy SDK, use well known HTTPS (secure HTTP) protocol to send events in XML format. These iRecorders can send events to an existing eTrust Audit system. This is possible thanks to the iRouter. Its relationship with iRecorders is depicted in the diagram below.

iRecorders, just like traditional recorders, send log events to an eTrust Audit Router and eTrust Action Manager for event processing. They require an intermediate component, called the eTrust Audit iRouter, which is installed on an existing eTrust Audit Client and provides a bridge between the iRecorder and native eTrust Audit Client. Tokens are converted from XML format to Audit SAPI format and submitted to the native Audit Router as shown in the following illustration:



The iRecorder architecture allows easy deployment across firewalls and new iRecorder development does not require changes in the existing eTrust Audit deployment.

How To Install the iRouter

The following topics describe how to install the iRouter.

Install the iRouter

The iRouter is installed with Audit Client.

Uninstall the iRouter

The iRouter is removed when you uninstall Audit Client. You can also use the Add/Remove Programs applet in the Control Panel to uninstall the iRouter by double-clicking
"CA eTrust Audit iRouter". For Solaris and Linux, when you uninstall the Audit Client, the iRouter is also uninstalled, or you can enter the following command to uninstall iRouter independently of Audit:

```
'cat/opt/CA/SharedComponents/iTechnology.location'/recorder_uninstall.sh -u iRouter"
```

Download iRouter from ca.com

You can download and install the iRouter from the CA Support site. To install the downloaded package, you will need two components:

- iRouter installation package from http://supportconnect.ca.com (http://supportconnect.ca.com)
- iGateway package for Windows from ftp://ftp.ca.com/pub/iTech/downloads (ftp://ftp.ca.com/pub/itech/downloads)

Download these packages into the same directory and run the iRouter install package. The iRouter install package automatically installs the iGateway package, if needed.

Detailed installation instructions for the iRouter are provided in the next topics.

Locate iRouter on the Media

To run the installation from the *Post Collection Utility, iRecorders, and iRouter* media, follow these steps:

1. Insert the media into the media drive.

   The DemoShield should start automatically and display the installation menu. If the DemoShield does not automatically start, click Start, Run and enter the following command:

   media-Drive:\setup.exe

   where *media-Drive* is your media drive letter designation.

   - Linux: \eTrust\Audit\Client\Linux
   - Solaris: \eTrust\Audit\Client\Solaris

2. Select the iRouter from the list and follow the detailed installation instructions, which are covered in the following topics.

Install the iRouter on Solaris

To install the iRouter on Solaris, follow these steps:

1. Log in as root.
2. Insert the media in the system's media drive.

   Solaris mounts your media on a system directory such as /cdrom.
3. Type df -k and locate an entry that contains /cdrom under the heading Mounted on. If the entry exists, the media is mounted.

   If there is no such entry in the df output, mount the media according to instructions in the *Solaris System Administration Manual*. For example:

   ```
   mount –F hsfs –o ro /dev/dsk/c0t1d0/s1s7cc_v10n1 /cdrom
   ```
4. Change directory to the iRouter directory:

   ```
   cd /cdrom/eTrust/Audit/iRouter/Solaris
   ```
5. Copy the iGateway package to a temp directory on the local hard disk, such as /tmp:

   ```
   cp iGateway_version_sunos.sh /tmp
   ```
6. Copy the iRouter package to the same directory as the iGateway package in the last step:

   ```
   cp iRouter-version-sunos.sh /tmp
   ```

   Note: The iRouter package name uses the following convention: iRouterPackage-version-os.sh where *iRouterPackage* is the name of the install package file, *version* is the version number, and *os* is the operating system. For example, if the install package is iRouter-1_53_3_040119-sunos.sh, the version number is 1_53_3_040119, and the operating system is sunos.
7. Change directory to the temp directory and unmount the media:

   ```
   cd /tmp
   umount /cdrom
   ```
8. Run the iRouter installation package by entering the following command:

   ```
   sh iRouter-version-sunos.sh /tmp
   ```

   The installation script begins.

The installation script guides you through the installation and configuration of the iRouter.

Perform a Silent Installation for Solaris

To perform a silent installation for Solaris, follow these steps:

1. Copy the appropriate version of the iRouter package and the iGateway package as instructed in Install the iRouter on Solaris (see page 295), Steps 1-6.
2. Follow the instructions in Create a Response File for UNIX/Linux Packages (see page 299) to create a response file to install the iRouter.
3. Enter the following command:

   ```
   sh PackageName-version-os.sh -s ResponseFilePath.resp
   ```

   where *PackageName* is the filename of the package, *version* is the version number, *os* is the operating system, and *ResponseFilePath* is the full path of the response file.

   For example, if the full path name of the response file is /tmp/irouter.resp, enter the following:

   ```
   sh iRouter-version-sunos.sh -s /tmp/irouter.resp
   ```

The installation proceeds silently without any prompts.

Perofrm a Silent Uninstallation

To perform a silent uninstallation for either Solaris or Linux, follow these steps:

1. Log in as root.
2. Enter the following command:

   ```
   /opt/CA/igateway/recorder_uninstall.sh -u iRouter
   ```

The process removes the iRouter from the system entirely.

Install the iRouter on Linux

To install the iRouter on Linux, follow these steps:

1. Log in as root.
2. Insert the media in the system's media drive.

   Linux mounts your media on a system directory such as /mnt/cdrom.
3. Type **df** and locate an entry which starts with /dev/cdrom. If the entry exists, the media is mounted.

   If there is no such entry in the df output, mount the media according to instructions in the *Linux System Administration Manual*. For example:

   ```
   mount -t iso9660 /dev/cdrom /mnt
   ```
4. Change directory to the iRouter directory:

   ```
   cd /mnt/eTrust/Audit/iRouter/Linux
   ```
5. Copy the iGateway package to a temp directory on the local hard disk, for example /tmp:

   ```
   cp iGateway_version_linux.sh /tmp
   ```
6. Copy the iRouter package to the same directory as the iGateway package in the last step:

   ```
   cp iRouter-version-linux.sh /tmp
   ```

   Note: The iRouter package name uses the following convention: iRouterPackage-version-os.sh where *iRouterPackage* is the name of the install package file, *version* is the version number, and *os* is the operating system. For example, if the install package is iRouter-1_53_3_040119-linux.sh, the version number is 1_53_3_040119, and the operating system is Linux.
7. Change directory to the temp directory and unmount the media:

   ```
   cd /tmp
   umount /mnt
   ```
8. Run the iRouter installation package by entering the following command:

   ```
   sh iRouter-version-linux.sh
   ```

   The installation script begins.

The installation script guides you through the installation and configuration of the iRouter.

Perform a Silent Installation for Linux

To perform a silent installation for Linux, follow these steps:

1. Copy the appropriate version of the iRouter package and the iGateway package as instructed in Install the iRouter on Linux (see page 297), Steps 1-6.
2. Follow the instructions in Create a Response File for UNIX/Linux Packages (see page 299) to create a response file to install the iRouter.
3. Enter the following command:

   ```
   sh PackageName-version-os.sh -s ResponseFilePath.resp
   ```

   where *PackageName* is the filename of the package, *version* is the version number, *os* is the operating system, and *ResponseFilePath* is the full path of the response file.

   For example, if the full path name of the response file is /tmp/irouter.resp, enter the following:

   ```
   sh iRouter-version-linux.sh -s /tmp/irouter.resp
   ```

The installation proceeds silently without any prompts.

Perofrm a Silent Uninstallation

To perform a silent uninstallation for either Solaris or Linux, follow these steps:

1. Log in as root.
2. Enter the following command:

   ```
   /opt/CA/igateway/recorder_uninstall.sh -u iRouter
   ```

The process removes the iRouter from the system entirely.

Customize Response Files for Silent Installation

The response files provided with the package contain an example of a silent install session. You should customize the silent installation to fit your environment.

Create a Response File for Unix or Linux Packages

To create a custom silent installation response file that you can use for silent installations on similar UNIX and Linux target systems, follow these steps::

1. Choose a system that is similar or identical to the target system.
2. Log in as root.
3. Change directory to the location that contains the iRouter package by entering the following command according to your destination platform:

   Note: If the iRouter package is on the *Post Collection Utility, iRouters, and iRouter* media, follow your *System Administration* documentation about how to mount a media and mount it on a directory such as: /mnt (or /cdrom).

   - **On Linux,** enter the following command:

     ```
     cd /cdrom/eTrust/Audit/iRouters/Linux
     ```

   - **On Solaris,** enter the following command:

     ```
     cd /cdrom/eTrust/Audit/iRouters/Solaris
     ```

4. Enter the following:

   ```
   sh iRouter-version-os.sh -g ResponseFileName
   ```

   where *version* is the version number, *os* is the operating system (sunos or linux), and *ResponseFileName* is the full path name of the response file.

   For example:

   ```
   sh iRouter-version-sunos.sh -g /tmp/iRouter_setup.resp
   ```

   The installation procedure performs a sample installation and records all the customized answers in the response file.

After the mock installation is complete, you can use the response file to silently install the iRouter package on similar target systems.

How to Configure and Use the iRouter

The following topics describe how to configure and use the iRouter.

Starting and Stopping the iRouter

The iRouter is run as a sub-component of the iTechnology iGateway service.

To start the iRouter on Unix or Linux, start the iGateway service by following these steps:

1. Log in as root.
2. Enter the following command:

```
/opt/CA/igateway/S99igateway start
```

Stop the iRouter

The iRouter is run as a sub-component of the iTechnology iGateway service.

To stop the iRouter on UNIX or Linux, start the iGateway service by following these steps:

1. Log in as root.
2. Enter the following command:

```
/opt/CA/igateway/S99igateway stop
```

Configure the iRouter

iRouter configuration parameters are kept in a configuration file for iControl called iControl.conf, located in the iGateway installation directory (for example C:\Program Files\CA\igateway).

The iRouter configuration parameters are automatically set during the iRouter installation and do not require any changes for the normal operation of the iRouter.

If you do want to modified any parameters, follow these steps:

1. Before making any changes, stop the iTechnology iGateway service
2. Make your needed parameter changes to the iControl.conf file and save it.
3. Restart the service for your changes to take effect.

Sample Configuration File

iRouter configuration parameters are kept in a shared configuration file called iControl.conf. Each iGateway installation always includes iControl.conf with a standard set of parameters.

After installing iRouter, the configuration file is automatically updated to reflect the presence of iRouter. In other words, iControl.conf looks different if the iRouter is installed. The following is a sample iControl.conf configuration file provided for information only.

```
<ISponsor>
        <Name>iControl</Name>
        <ImageName>iControl</ImageName>
        <Version>1.53.3.040119</Version>
        <DispatchEP>iDispatch</DispatchEP>
        <ISType>DSP</ISType>
        <PreLoad>true</PreLoad>
        <RouteEvent>false</RouteEvent>
        <RouteEventThreads>5</RouteEventThreads>
        <RouteEventHost>localhost</RouteEventHost>
        <SafDeliverThreads>1</SafDeliverThreads>
        <EventsToCache>100</EventsToCache>
        <EventCPlugin>epAudit</EventCPlugin>
</ISponsor>
```

If the iRouter is installed, exclusive parameters are set as shown below:

```
<RouteEvent>false</RouteEvent>
<RouteEventHost>localhost</RouteEventHost>
<EventCPlugin>epAudit</EventCPlugin>
```

If they are not set correctly or if there are duplicates of these lines with different settings, edit the file and make sure that these lines are entered correctly.

Configure Event Pulling

For information about event pulling mode compared to event pushing, see About Event Pulling (see page 303).

To set your configuration file to enable the Event Pulling mode of operation, follow these steps:

1. On the iRouter host machine, open the iControl.conf configuration file for editing.

   See Sample Configuration File (see page 301) for further information on this file.

2. Add the following tag to the file, before the </iSponsor> line:

   <RetrieveEventHost>*host*</RetrieveEventHost>

   where *host* is the full host name or IP address of the iRecorder.

3. Add the above tag on another line in the file for each of the iRecorder host machine (*host*) your iRouter needs to access for event pulling.

   Your iRouter configuration file is set, and you can now configure the iRecorder file.

4. On the iRecorder host machine, open the iControl.conf configuration file for editing.

5. Add the following tag to the file, before the </iSponsor> line:

   <StoreEventHost>*host*</StoreEventHost>

   where *host* is the full host name or IP address of the iRouter.

6. Add the above tag on another line in the file for each of the iRouter host machines (*host*) your iRecorder needs to service.

The event pulling mode of operation is enabled, allowing better event communication for firewall-protected environments.

About Event Pulling

The configuration illustrated in the sample configuration file is established during installation and is called *event pushing*. Event pushing means the iRecorder is actively sending events to the iRouter.

The tag <RouteEventHost> in the iControl.conf configuration file is set up to contain the iRouter host machine's fully qualified domain name or IP address. Also, the tag <RouteEvent> in the iControl.conf file is set to *true* to activate the sending mechanism on the host. This mode, in which the iRouter sends, or pushes, events is the normal event pushing mode of operation for iRecorders and the iRouter.

However, this mode of operation is not always suitable in situations where firewalls are configured to block incoming traffic. One such scenario is when your iRecorders are deployed outside the intranet (in the DMZ or on the Internet) and your iRouter is inside the intranet. This type of scenario lends itself to *event pulling.* Event pulling is the mode of operation in which your iRouter accesses the iRecorder host to get, or pull, the events. This mode provides a remedy when firewalls prevent external iRecorders from sending or pushing events to internal iRouters.

For more information, see Introduction to iTechnology. You can configure event pulling in your configuration file by referring to the topic, Configure Event Pulling (see page 302).

Test the iRouter

To test proper functionality of the iRouter, follow these steps:

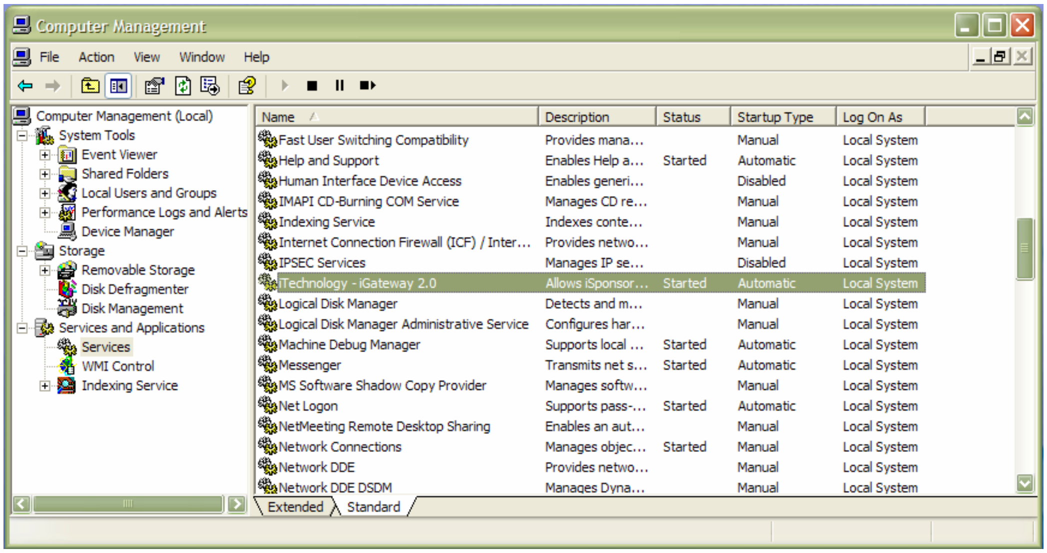
1. Ensure that the iTechnology iGateway service is running.

   The iGateway service is required for the proper operation of the eTrust Audit iRouter. It receives XML data such as events from iRecorders and passes it to the iRouter to send it to the eTrust Audit Log Router.

2. Right-click the My Computer icon on the system's desktop.

3. Click Manage.

   The Computer Management window appears:



Name	Description	Status	Startup Type	Log On As
Fast User Switching Compatibility	Provides mana…		Manual	Local System
Help and Support	Enables Help a…	Started	Automatic	Local System
Human Interface Device Access	Enables generi…		Disabled	Local System
IMAPI CD-Burning COM Service	Manages CD re…		Manual	Local System
Indexing Service	Indexes conte…		Manual	Local System
Internet Connection Firewall (ICF) / Inter…	Provides netwo…		Manual	Local System
IPSEC Services	Manages IP se…		Disabled	Local System
iTechnology - iGateway 2.0	Allows iSponsor…	Started	Automatic	Local System
Logical Disk Manager	Detects and m…		Manual	Local System
Logical Disk Manager Administrative Service	Configures har…		Manual	Local System
Machine Debug Manager	Supports local …	Started	Automatic	Local System
Messenger	Transmits net s…	Started	Automatic	Local System
MS Software Shadow Copy Provider	Manages softw…		Manual	Local System
Net Logon	Supports pass-…	Started	Automatic	Local System
NetMeeting Remote Desktop Sharing	Enables an aut…		Manual	Local System
Network Connections	Manages objec…	Started	Manual	Local System
Network DDE	Provides netwo…		Manual	Local System
Network DDE DSDM	Manages Dyna…		Manual	Local System

4. Check that iGateway service is Started.

5. If it is not running, start the service.

With the service started, your iRouter is working.

Appendix A: The Submit API (SAPI)

This section contains the following topics:

Introduction

eTrust Audit provides an API, the Submit API (SAPI), to submit audit events to the Router. The Submit API provides a simple means of adding new sources of audit information. Any third-party application intended to submit events to Audit should use the SAPI calls.

Because the objective of Audit is to enable event analysis, both online and offline, it is important that events from different sources conform to a single concept. On the other hand, it is vital that native auditing information be preserved. The SAPI allows for both:

- If a submitted application's events are to be analyzed by Audit, it must map events to the common format. The unified format simplifies management, reporting, and analysis. For example, Intrusion Detection rules for generic events such as logon/logoff can be easily administered cross-platform.
- Translators are functions that translate external data representation (such as UNIX time_t) to SAPI internal string format. Each translator is identified by name. Currently three translators are supported: string, timet, and long.
- The client is free to add fields for native information. Auditors can report on events from a certain source by using the terms specific to the source.

Mapping

Messages are created by mapping to fields defined in the header file AC_SAPITokens.h. The SAPI format is completely free. However, some fields are mandatory and others are strongly recommended.

Message Routing

After mapping, the resulting message is submitted to a router. By default, events are submitted to the router resident on the local machine. You can configure the SAPI to submit to the router of your choice.

Following a successful submit operation, eTrust Audit provides guaranteed delivery according to the filters and actions specified in the router's filter rules file (router.cfg).

Submitting a Message to the Router

To submit events to the SAPI, follow these steps:

1. Create a SAPI context by using SAPI_Init. The context is helpful in the case of multiple threads.

   Note: You must use SAPI_Init before any other SAPI function.

2. Create a message handle by using SAPI_NewMessage.
3. By using the message handle, you add items (fields) to the message with SAPI_AddItem.
4. With the same handle, submit the message to the router with SAPI_SubmitMsg.
5. After a message has been successfully submitted, use SAPI_RemoveMessage to clear it from memory.

Handling Submit Failures

If the attempt to submit a message fails, you can remove it, or try to submit it again. If the message is not removed, it stays in memory.

Note: After the first submit attempt, the message is locked and cannot be changed.

Compiling and Linking

To use the Submit API, you must include a header file with prototypes and structure definitions in your source code.

The header file is etsapi.h

For mapping, use AC_SAPITokens.h.

Libraries

On UNIX: SAPI includes two shared libraries:

- etsapi.so
- etbase.so

In Windows: the corresponding files are:

- etsapi.dll
- etbase.dll.

Sample SAPI Routine

The following is a simple example of SAPI usage. The following application sends a single message containing five fields (category of event, native event ID, logname, source, and info). The field, timestamp, is added by default.

Note: SAPI_Init and SAPI_Destroy should be used only once per application—not once per message as in this demonstration.

```
#include "etsapi.h"
#include "AC_SAPITokens.h"

/*
 * Usage : test [host]
 */
int main(int argc, char *argv[])
{
    SAPI_CTX        ctx;        /* SAPI context               */
    SAPI_HANDLE_I    h;          /* handle for new message      */
    SMStatus        rv;         /* return value to check        */
    SMStatus        remote_rv;   /* return value from the receiver */


    Char            msg_buffer[1024];
    long            eventId    = 123456;
    char            category[] = "General";
    char            logname[]  = "test_log";
    char            source[]   = "test_recorder";
    char            info[]     = "test_recorder information";

    rv = SAPI_Init(&ctx, NULL);    /* Create a new SAPI context */
    if (rv != SAPI_SUCCESS)
    {
        printf("SAPI_Init: failed code : 0x%X\n", rv);
        return 1;
    }

    /* set destination host, default - localhost */
    if (argc > 1)
    {
    rv = SAPI_SetRou


ter(ctx, argv[1]);
    if (rv != SAPI_SUCCESS)
    {
      printf("SAPI_SetRouter: host = '%s', failed code : 0x%X\n",
            argv[1], rv);
      return 1;
    }
    else
      printf("Set destination host %s\n", argv[1]);
    }

    rv = SAPI_NewMessage(ctx, &h);  /* Create a new SAPI message */
    if (rv != SAPI_SUCCESS)
    {
      printf("SAPI_NewMessage: failed code : 0x%X\n", rv);
      return 1;
    }


    /* Add a new items to a message */

    rv = SAPI_AddItem(ctx, h,
                SAPI_TRANS_DATATYPE_STRING,
                SAPI_CATEGORY_FLD,
                category);
    if (rv != SAPI_SUCCESS)
    {
      printf("SAPI_AddItem: failed code : 0x%X\n", rv);
      return 1;
    }

    rv = SAPI_AddItem(ctx, h,
                SAPI_TRANS_DATATYPE_LONG,
                SAPI_NATIVEID_FLD,
                &eventId);
    if (rv != SAPI_SUCCESS)
    {
      printf("SAPI_AddItem: failed code : 0x%X\n", rv);
      return 1;
    }

    rv = SAPI_AddItem(ctx, h,
                SAPI_TRANS_DATATYPE_STRING,
                SAPI_LOGNAME_FLD,
                logname);
    if (rv != SAPI_SUCCESS)
    {
      printf("SAPI_AddItem: failed code : 0x%X\n", rv);
      return 1;
    }

    rv = SAPI_AddItem(ctx, h,
                SAPI_TRANS_DATATYPE_STRING,
                SAPI_SOURCE_FLD,
                source);
```

```
if (rv != SAPI_SUCCESS)
{
   printf("SAPI_AddItem: failed code : 0x%X\n", rv);
   return 1;
}

rv = SAPI_AddItem(ctx, h,
                SAPI_TRANS_DATATYPE_STRING,
                SAPI_INFO_FLD,
                info);
if (rv != SAPI_SUCCESS)
{
   printf("SAPI_AddItem: failed code : 0x%X\n", rv);
   return 1;
}

/* Print the content of a message to a buffer */
rv = SAPI_DumpMessage(ctx, h, msg_buffer, sizeof(msg_buffer));
if (rv != SAPI_SUCCESS)
{
   printf("SAPI_DumpMessage: failed code : 0x%X\n", rv);
   return 1;
}
else
{
   printf("SAPI message:\n %s\n", msg_buffer);
}

/*Submits the message to a SAPI router.*/

rv = SAPI_SubmitMsg(ctx, h, &remote_rv);
if (rv == SAPI_SUCCESS)
   printf("SAPI_SubmitMsg OK, remote return code : 0x%X\n", remote_rv);
else
   printf("SAPI_SubmitMsg: failed code :0x%X\n", rv);

/*Remove a message from the given context.*/
rv = SAPI_RemoveMessage(ctx, h);
if (rv != SAPI_SUCCESS)
{
   printf("SAPI_RemoveMessage: failed code : 0x%X\n", rv);
  return 1;
}

/* destroy SAPI context and free all its allocations */
rv = SAPI_DestroyCTX(ctx);
if (rv != SAPI_SUCCESS)
{
   printf("SAPI_DestroyCTX: failed code :0x%X\n", rv);
```

```
        return 1;
    }

    return 0;
}
```

SAPI Functions Reference

SAPI functions use the following type definitions.

SAPI_CTX

SAPI context contains state information for all SAPI calls

SAPI_HANDLE_l

SAPI message handles used for referring to specific messages

SAPI_HANDLE_lp

SAPI message handles used for referring to specific messages

The SAPI uses the functions in the following topics to pass messages to the eTrust Audit router.

SAPI_Init

This function must be called before any other SAPI functions can be used.

Syntax

```
SMStatus SAPI_Init( SAPI_CTX            *ctx,
                    char                *config );
```

Parameters

ctx

The address of pointer to SAPI context.

config

The configuration (reserved for future use).

SAPI_NewMessage

The SAPI_NewMessage function creates a handle to new message in the given context. The message is also filled with automatic arguments for mandatory fields with their default values.

Syntax

```
SMStatus SAPI_NewMessage( SAPI_CTX              * ctx,
                    SAPI_HANDLE_lp          Handle );
```

Parameters

ctx

The SAPI context. This parameter’s value originates with SAPI_Init.

handle

The address of the handle to return on success.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context.

SAPI_AddItem

The SAPI_AddItem function adds a new Item to a message. If an Item by the given name already exists, it is replaced by the given Item.

Syntax

```
SMStatus SAPI_AddItem( SAPI_CTX                ctx,
              SAPI_HANDLE_I            handle,
              char                *item_type,
              char                *name,
              void                *value );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

handle

The handle to a message. This parameter's value originates with SAPI_NewMessage.

item_type

The external raw data type. The available item types are as follows:

long

The value should point to address of long.

string

The value should point to a null terminated char string.

timet

The value should point to the address of a time_t.

name

The item name

value

The binary raw data.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context.

SAPI_SubmitMsg

The SAPI_SubmitMsg functin submits the message to a SAPI router.

Note: After the message has been submitted, you must free it with SAPI_RemoveMessage.

Syntax

```
SMStatus SAPI_SubmitMsg( SAPI_CTX              ctx,
             SAPI_HANDLE_I              handle,
             SMStatus               *sapi_remote_rv );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

handle

The handle to a message. This parameter's value originates with SAPI_NewMessage.

sapi_remote_rv

The return value of the remote function.

Return Values

The function returns SAPI_SUCCESS on success.

SAPI_RemoveMessage

SAPI_RemoveMessage removes a message in the given context. Use the function to clear sent messages from memory.

Syntax

```
SMStatus SAPI_RemoveMessage( SAPI_CTX              ctx,
               SAPI_HANDLE_I             Handle );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

handle

The handle to a message. This parameter's value originates with SAPI_NewMessage.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context.

SAPI_DumpMessage

The SAPI_DumpMessage function prints the content of a message in the given context to a buffer. Function prints the string values of the message fields.

Syntax

```
SMStatus SAPI_DumpMessage( SAPI_CTX              ctx,
                 SAPI_HANDLE_I         handle,
                 char                 * buffer,
                 int                  Size );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

handle

The handle to a message. This parameter's value originates with SAPI_NewMessage.

Buffer

The buffer to output.

Size

The size of the buffer.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context. It also returns SAPI_BADPARAM_RC for too small buffer size.

SAPI_DestroyCTX

The SAPI_DestroyCTX function frees current SAPI context and all unsent messages and gracefully shuts the client side of SAPI.

Syntax

```
SMStatus SAPI_DestroyCTX( SAPI_CTX              ctx );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

Return Values

The function returns SAPI_SUCCESS on success.

SAPI_SetRouter

The SAPI_SetRouter function registers the name of a new router host.

Syntax

```
SMStatus SAPI_SetRouter( SAPI_CTX          Ctx,
            char *        hostname );
```

Parameters

ctx

The SAPI context. This parameter’s value originates with SAPI_Init.

hostname

The name of the host where the router resides.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADPARAM_RC for an invalid context.

SAPI_SetRouterPort

The SAPI_SetRouterPort function changes the default SAPI router port number.

Syntax

```
SMStatus SAPI_SetRouterPort( SAPI_CTX              Ctx,
             unsigned short              Portnum );
```

Parameters

ctx

The SAPI context. This parameter’s value originates with SAPI_Init.

portnum

The user-defined port number to be registered in portmap. If you specify 0, the port number will be set by portmap.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context.

SAPI_SetRouterTimeout

The SAPI_SetRouterTimeout function changes the default SAPI router timeout period.

Syntax

```
SMStatus SAPI_SetRouterTimeout( SAPI_CTX            Ctx,
                    unsigned long           Timeout );
```

Parameters

ctx

The SAPI context. This parameter's value originates with SAPI_Init.

timeout

The user-defined timeout period, in seconds.

Return Values

The function returns SAPI_SUCCESS on success and SAPI_BADCTX_RC for an invalid SAPI context.

SAPI Return and Error Codes

The following macros process return codes for all SAPI calls.

Each return code is composed from (most to least):

- 1 bit - success or failure code
- 16 bits - software component ID number. In the case of the SAPI, the ID number is 11 (SAPI_RC_BASE).
- 12 bits - meaningful portion of return code

Macro	Purpose
_SM_IS_FAIL(rc) (rc>>31)	The macro checks whether the call failed. In case of failure, the macro returns TRUE or 1.
_SM_RC_PKG(rc) ((rc>>12)&0xffff)	The macro extracts and returns the software component ID number.
_SM_RC_CODE(rc) (rc&0xfff)	The macro extracts and returns the meaningful portion of the return code.

The following table describes the return and error codes defined in etsapi.h:

Name	Construction	Meaning
SAPI_SUCCESS	0	Function returned successfully.
SAPI_MALLOC_RC	_SM_RC_FAIL(SAPI_RC_BASE,1)	SAPI could not allocate memory.
SAPI_NOHANDLE_RC	_SM_RC_FAIL(SAPI_RC_BASE,2)	Requested SAPI message handle could not be found.
SAPI_BADPARAM_RC	_SM_RC_FAIL(SAPI_RC_BASE,3)	Function received a bad parameter (most commonly a NULL pointer).
SAPI_NOITEM_RC	_SM_RC_FAIL(SAPI_RC_BASE,4)	Low-level internal code, should not appear in normal operation.
SAPI_ALRDYEXIST_RC	_SM_RC_FAIL(SAPI_RC_BASE,5)	A field by the same name already exists in the message.
SAPI_UNSUPPORTED_RC	_SM_RC_FAIL(SAPI_RC_BASE,6)	Unsupported SAPI type.
SAPI_NOAUTOARG_RC	_SM_RC_SUCCESS(SAPI_RC_BASE,7)	Low-level internal code, should not appear in normal operation.
SAPI_BADCTX_RC	_SM_RC_FAIL(SAPI_RC_BASE,8)	Function got an invalid SAPI context for input.
SAPI_MSGLOCKED_RC	_SM_RC_FAIL(SAPI_RC_BASE,9)	Low-level internal code, should not appear in normal operation.
SAPI_NOTHINGTOSEND_RC	_SM_RC_SUCCESS(SAPI_RC_BASE,10)	Low-level internal code, should not appear in normal operation.
SAPI_NOTREROUTING_RC	_SM_RC_FAIL(SAPI_RC_BASE,11)	Low-level internal code, should not appear in normal operation.
SAPI_REROUTINGMODE_RC	_SM_RC_FAIL(SAPI_RC_BASE,12)	Low-level internal code, should not appear in normal operation.

Fields for SAPI

The SAPI format is completely free, except for certain mandatory fields, generally, those affecting intrusion detection and security auditing. If the submitting application does not provide values for such fields, the SAPI will provide a default value.

Additional fields can be added as you choose. However, for security-related events it is strongly recommended to map to the predefined SAPI fields. Unless events map to the SAPI fields, they will be treated generically by the eTrust Audit viewers.

Predefined fields are defined in the file AC_SAPITokens.h. User-defined field names should be unique.

It is recommended to identify the log or source in all user-defined field names. For example, the first of these two macro definitions is specific to the SAPI and the second, to Oracle.

```
#define SAPI_DATE_FLD              ”Date”
#define ORA_AUDIT_OPTION              “ORA_Audit_Option”
```

Field Properties

Each SAPI field has three properties: name, type, and value. Field types are assigned when submitting messages. Available types are date, string and long.

The SAPI fields discussed below are organized by priority.

- Mandatory fields must be present in every record.
- Common predefined fields are important for event identification and description.
- Optional, category-specific fields provide further characterization of events. Other fields are specific to event sources.

Mapping Examples

The following are examples of mapping of SAPI fields.

Event	User	Category	Subcategory	ObjClass	ObjName	Oper
User account was created	“Administrator”	Account Management	Administration	USER	newuser	Create

Event	User	Category	Subcategory	ObjClass	ObjName	Oper
Registry key was deleted	“richard”	Object Access	Administration	REGKEY	“HKEY_USERS\ . . . “	Delete
Process was stopped (NT)	“joan”	Object Access	Activation	PROCESS	“FINDFAST.EXE”	Stop
Windows NT was shut down	“SYSTEM”	Security Systems		OS		Stop
A file was opened for read	“joan”	Object Access	Usage	FILE	“c:\winnt\system.ini”	Read

Mandatory Fields for Event Identification

The SAPI requires that certain fields be present in each message you submit. These fields contain data on the time, place, and status of events. For some fields, values are strictly predefined.

SAPI_LOCATION_FLD "Location"

The name of the host where the event was originated. Name format is UNIX qualified name or UNC (if DNS is not available).

Examples

host.mydomain.com (UNIX qualified name
\\mydomain\host (UNC).

Default Value

The name of machine where submitter is resident.

SAPI_LOGNAME_FLD "Log"

The logical log name that uniquely identifies the native auditing type. That is, the logical name of the source of audit information.

Examples

NT-System, NT-Application
UNIX for syslog and sulog files
Oracle for Oracle logs

Default Value

The submitter must supply the contents for this field.

SAPI_SOURCE_FLD "Src"

The name of the software component that issued the event.

Note: The audit mechanism may serve more than one process or application. When a native auditing environment has more than one instance on the same machine, this field will contain the instance identification.

Examples

Windows NT—Security, Disk, NETLOGON
UNIX—telnetd, ftpd

Default Value

The submitter must supply the contents for this field.

SAPI_DATE_FLD "Date"

When the event was originated. Date contains both date and time in standard ISO format (text format that includes date, time and time zone).

Examples

20010201T080001-0500 means Feb. 1, 2001at 8:00:01 EST
20010202T080001+0000 means Feb. 2, 2001 at 8:00:01 GMT

Default Value

The date and time at machine where the event is submitted.

SAPI_STATUS_FLD "Status"

The status, which the event describes. Values for Status are strictly predefined:

"S" SAPI_STATUS_SUCCESS

Event for a successful operation.

"F" SAPI_STATUS_FAILURE

Event for a failure operation.

"D" SAPI_STATUS_DENIED

Event for a failure operation where the reason is insufficient privileges.

We recommend that you use "F" SAPI_STATUS_FAILURE even for failure operations that is caused by insufficient privileges.

Note: All source specific statues should be converted into one of SAPI statuses. To keep the original value put it into specific field:

<SRC>_Status, where <SRC> is an unique identifies the source of audit information.

Default Value

"S"

Common Predefined Fields for Event Identification

The following fields are used by most events. They are not mandatory, but they are strongly recommended for each SAPI message.

SAPI_USER_FLD "User"

The name of the user (or principal as some systems define) who performed the audited operation.

Example

Windows NT—Administrator, my_domain\john
UNIX—"root," "john"

Default Value

None.

SAPI_USERID_FLD "UID"

The native user ID.

Example

Windows NT—S-1-5-21-1793529420-1590284213-401-284377-1208
UNIX—0 (root user)

Optional Predefined Fields for Event Identification

Certain fields providing event identification are optional.

SAPI_LOCATIONIP_FLD "LocationIP"

The IP address where the event was originated.

Example

112.111.248.116

SAPI_LOGFILENAME_FLD "LogF"

The physical file name (full path name), if available, in cases where the audit does not reside in a fixed file.

Example

UNIX—/usr/logs/trace1.log

SAPI_RECORDERVER_FLD "RecVer"

The version of the submitter for the native auditing environment.

Common Predefined Fields for Event Description

The following fields provide general information about events. They are not mandatory, but it is recommended to set their values (if available) for each SAPI message.

Reserved fields specific to predefined security event categories are listed later in this chapter.

SAPI_CATEGORY_FLD "Category"

The security-related events fall into predefined categories. If the event belongs to one of the categories, it is highly recommended to set the field's value. The field can be left empty, or it can have a user-defined category if the predefined values are not matched.

Example

"System Access" SAPI_CATEGORY_SYSACC for any logon or logoff operation

"Account Management" SAPI_CATEGORY_ACCOUNT for user account definition

SAPI_SUBCAT_FLD "Subcat"

Enables subdivision of events within a category. You can fill this field by using either a pre-defined value or any other string value.

SAPI_SEVERITY_FLD "Severity"

The logical severity of the event set by eTrust Audit policies (not by application severity).

Values for Severity are strictly predefined.

"0" SAPI_SEVERITY_INFO

"1" SAPI_SEVERITY_WARNING

"2" SAPI_SEVERITY_CRITICAL

"3" SAPI_SEVERITY_FATAL

SAPI_OPERATION_FLD "Oper"

The operation performed on an object. Values are chosen from a list of predefined values. In cases where the predefined values are not suitable, native auditing values may be used.

Example

"Write" SAPI_OPER_WRITE—edited a file or registry key
"Start" SAPI_OPER_START—started a service

SAPI_OBJCLASS_FLD "ObjClass"

The class of the object of the operation. Values are chosen from a list of predefined values. In cases where the predefined values are not suitable, native auditing values may be used.

Example

"FILE," "REGKEY"

SAPI_OBJNAME_FLD "ObjName"

The name of the object on which the operation is performed.

Example

"C:\WINNT\system.ini"—a file name
"notepad.exe"—a process name

SAPI_OBJCLASS2_FLD "SecObjClass"

The class of the second object that participated in the event (if it exists).

Example

"Group"—in case of joining a user to a group

SAPI_OBJNAME2_FLD "SecObjName"

The name of the second object that participated in the event (if it exists).

Example

"Administrators"—as the name of the group a user was added to

SAPI_NATIVEOID_FLD "OID"

The native object ID (handle) from auditing or operating system.

Example

Windows NT—"24"

SAPI_PID_FLD "PID"

The Process ID of the process that performed the operation, if available.

Example

WINDOWS NT—"2309196368"

SAPI_NATIVEID_FLD "NID"

The native ID of the event, in native auditing environments that enumerate events.

Example

Windows NT—"562" for closed handle event, "592" for process creation.

SAPI_INFO_FLD "Info"

The free-text event information.

Example

Windows NT—A process has exited.

Process ID: 215487040

User Name: user_john

Domain: My_Domain

Logon ID: (0x0,0x3ED6)

UNIX—printer/tcp: "Print services stopped"

Mapping Events to Predefined Categories

For each security event category, records can be built from a certain set of SAPI fields, in addition to the mandatory identifying fields.

Predefined security-related categories are:

- System Access
- Account Management
- Object Access
- Policy Management
- Security Systems
- Network
- Detailed Tracking
- Physical Security

Other events (generally, start and stop notifications for applications) fall into the one of the following categories:

- System \ Application
- Administration
- General

System Access Events

System access events include logon, logoff, and change of user identity (impersonation).

SAPI_CATEGORY_FLD "Category"

"System Access" SAPI_CATEGORY_SYSACC

SAPI_SOURCE_FLD "Src"

The software component that generated the message.

Example

Windows NT—"Security"
UNIX—"login," "telnetd," in.telnetd," rshd," "in.rshd," "Xsession" (XDMCP), "ftpd," "in.ftpd," "rlogind," "in.rlogind," "fingerd," ffingerd"

SAPI_OPERATION_FLD "Oper"

"Logon" SAPI_OPER_LOGON
"Logoff" SAPI_OPER_LOGOFF

SAPI_USER_FLD "User"

The name of the logged-on user.

SAPI_SURROGATEUSER NAME_FLD "SurrogateUser"

The name of the new user when logging on from another user. For example, the UNIX command su root generates a SurrogateUser value of "root."

SAPI_INFO_FLD "Info"

May contain reason for failed logon.

SAPI_LOGONTYPE_FLD "LogonType"

For logon operations, the type of logon. Values for LogonType are strictly predefined.

Example

"Interactive" SAPI_LOGONTYPE_INTERACTIVE—local user logon
"Server" SAPI_LOGONTYPE_SERVER—logon to server, domain or shared drive

SAPI_TERMINAL_FLD "Term"

The terminal name or ID from which the operation is initiated.

Example

"pts/7"

SAPI_REMOTEHOST_FLD "RemHost"

The name or address of the remote host for operations that are performed remotely (name should follow Location field format).

Account Management Events

Account management events include the creation, changing, and deletion of users, groups, profiles and roles, as well as the granting of permissions. For security purposes, special care should be taken to audit the addition of users to the administrators group, and the addition of significant authorizations.

The management of permissions on the system level is mapped to “Account Management,” and the management of auditing is mapped to “Policy Management.” For individual objects, both permissions and auditing setups are mapped to “Object Access.”

SAPI_CATEGORY_FLD “Category”

“Account Management” SAPI_CATEGORY_ACCOUNT

SAPI_SUBCAT_FLD “Subcat”

“Permission” SAPI_SUBCAT_PERMISSION
“Audit” SAPI_SUBCAT_AUDIT
“Password” SAPI_SUBCAT_PASSWORD

SAPI_OPERATION_FLD “Oper”

Some possible values are predefined.

Examples

“Create” SAPI_OPER_CREATE
“Delete” SAPI_OPER_DELETE
“ChangeProperty” SAPI_OPER_CHANGEPROPERTY
“Lock” SAPI_OPER_LOCK
“Unlock SAPI_OPER_UNLOCK

SAPI_OBJCLASS_FLD “ObjClass”

“USER” SAPI_OBJCLASS_USER
“GROUP” SAPI_OBJCLASS_GROUP

SAPI_OBJNAME_FLD “ObjName”

The nName of user or group.

SAPI_OBJCLASS2_FLD “SecObjClass”

The class of the secondary object.

Example

When adding a user to a group, “USER” is the primary object and “GROUP” is the secondary object.

When changing permissions, the secondary object is “PRIVILEGE” SAPI_OBJCLASS_PRIVILEGE.

SAPI_OBJNAME2_FLD “SecObjName”

The name of the secondary object.

SAPI_INFO_FLD “Info”

The free-text description of the operation.

Object Access Events

Object access events include any access to resources such as files and the registry. Usually these accesses are audited only for critical objects.

For individual objects, both permissions and auditing setups are mapped to “Object Access.” The management of permissions on the system level is mapped to “Account Management.”

SAPI_CATEGORY_FLD “Category”

“Object Access” SAPI_CATEGORY_OBJACC

SAPI_SUBCAT_FLD “Subcat”

“Password” SAPI_SUBCAT_PASSWORD
“Usage” SAPI_SUBCAT_USAGE
“Audit” SAPI_SUBCAT_AUDIT
“Activation” SAPI_SUBCAT_ACTIVATION
“Permission” SAPI_SUBCAT_PERMISSION

SAPI_OBJCLASS_FLD “ObjClass”

The name of the object on which the operation is performed. In cases where the predefined values are not suitable, native auditing values may be used.

Example

“REGKEY” — for registry key
“FILE” – for file or folder

SAPI_OBJNAME_FLD “ObjName”

The name of the accessed object.

SAPI_OPERATION_FLD “Oper”

The operation to perform on the object.

Examples

“Execute” SAPI_OPER_EXECUTE“Start” SAPI_OPER_START_RL
“Stop” SAPI_OPER_STOP
“Kill” SAPI_OPER_KILL
“Create” SAPI_OPER_CREATE
“Delete” SAPI_OPER_DELETE
“ChangeProperty” SAPI_OPER_CHANGEPROPERTY
“Rename” SAPI_OPER_RENAME
“TakeOwnership” SAPI_OPER_TAKEOWNERSHIP
“ChangePermission” SAPI_OPER_CHANGEPERMISSION
“Lock” SAPI_OPER_LOCK
“Unlock” SAPI_OPER_UNLOCK
“Open” SAPI_OPER_OPEN
“Read” SAPI_OPER_READ_RL
“Write” SAPI_OPER_WRITE
“Edit” SAPI_OPER_EDIT

SAPI_NATIVEOID_FLD (optional)

The object ID used by the native environment.

SAPI_PID_FLD (optional)

The ID of the process that accesses the object.

SAPI_COMMAND_FLD “Command” (optional)

The original command that caused the event (in case of a command line interface usage).

Example

eTrust Access Control Definition of new resource “new user(john)”

SAPI_INFO_FLD “Info”

The free-text event information.

Policy Management Events

Policy management events include changes in audit policy, changes in password policy, and other events on the system level. This category usually includes very few events.

For individual objects, permissions and auditing setups are mapped to “Object Access.”

SAPI_CATEGORY_FLD “Category”

“Policy Management” SAPI_CATEGORY_POLICY

SAPI_SUBCAT_FLD “Subcat”

“Audit” SAPI_SUBCAT_AUDIT
“Activation” SAPI_SUBCAT_ACTIVATION
“Permission” SAPI_SUBCAT_PERMISSION

SAPI_OPERATION_FLD “Oper”

“Create” SAPI_OPER_CREATE
“Delete” SAPI_OPER_DELETE

SAPI_OBJCLASS_FLD “ObjClass”

“POLICY” SAPI_OBJCLASS_POLICY

Oracle—map “Audit_Option” to this field

SAPI_OBJNAME_FLD “ObjName”

The object name.

SAPI_INFO_FLD “Info”

The free-text event information.

Security System Status Events

Security system status events include events related to the change in the status of security systems. For example, the stopping and starting of operating systems and the clearing of audit logs.

SAPI_CATEGORY_FLD "Category"

"Security Systems"
SAPI_CATEGORY_SECURITYSYS

SAPI_OPERATION_FLD "Oper"

"Restart" SAPI_OPER_RESTART
"Startup" SAPI_OPER_STARTUP
"Shutdown" SAPI_OPER_SHUTDOWN
"Clear" SAPI_OPER_CLEAR

SAPI_OBJCLASS_FLD "ObjClass"

"Service" (or daemon) SAPI_OBJCLASS_SERVICE
"Log" SAPI_OBJCLASS_LOG
"Process" SAPI_OBJCLASS_PROCESS
"OS" SAPI_OBJCLASS_OS

SAPI_OBJNAME_FLD "ObjName"

The name of started or stopped program.

SAPI_INFO_FLD "Info"

The free-text event information.

Physical Security System Events

Physical security system events include events related to the change in the status of physical security systems, for example, the switching of cameras, opening, closing, and locking doors, and so on.

SAPI_CATEGORY_FLD "Category"

"Physical Security"
SAPI_CATEGORY_SECURITYPH

SAPI_OPERATION_FLD "Oper"

"Restart" SAPI_OPER_RESTART
"Open" SAPI_OPER_OPEN
"Lock" SAPI_OPER_LOCK
"Unlock" SAPI_OPER_UNLOCK

SAPI_OBJCLASS_FLD "ObjClass"

The class of the audited objects.

SAPI_OBJNAME_FLD "ObjName"

The name of the audited objects.

SAPI_INFO_FLD "Info"

The free-text event information.

Network Events

Network events include:

- Incoming and outgoing communication events from eTrust Access Control
- eTrust Intrusion Detection (former SessionWall)
- Events from other network products to be integrated with eTrust Audit

Network events should map to identification fields.

SAPI_CATEGORY_FLD "Category"

"Network" SAPI_CATEGORY_NETWORK

SAPI_OPERATION_FLD "Oper"

"Connect" SAPI_OPER_CONNECT
"Disconnect" SAPI_OPER_DISCONNECT

SAPI_OBJCLASS_FLD "ObjClass"

"PORT" SAPI_OBJCLASS_PORT PORT
"HOST" SAPI_OBJCLASS_HOST
"TERMINAL" SAPI_OBJCLASS_TERMINAL
"DOMAIN" SAPI_OBJCLASS_DOMAIN
"PROCESS" SAPI_OBJCLASS_PROCESS
"PRINTER" API_OBJCLASS_PRINTER_RL

SAPI_OBJNAME_FLD "ObjName"

The object name, name of host, terminal, domain and so on.

SAPI_INFO_FLD "Info"

The free-text event information.

The following additional fields contain network objects.

SAPI_REMOTEIP_FLD "RemIP"

The remote IP address.

SAPI_AFTYPE_FLD "AddressFamily"

The address family.

SAPI_NETSERVICENAME_FL "NetServiceName"

The service or daemon

Example

"FTP"

SAPI_PORT_FLD "Port"

The local port number

Example

"7890"

SAPI_REMOTEPORT_FLD "RemotePort"

The remote port number.

Example

"8765"

SAPI_PROTOCOL_FLD "Protocol"

The protocol.

Example

"TCP," "UDP"

SAPI_URL_FLD "URL"

URL

Example

"www.ca.com"

SAPI_DIRECTION_FLD "Direction"

The event direction: inbound or outbound.

Example

"IN"
"OUT"

SAPI_EVENT_COUNT_FLD "EventCount"

The count of events, if the event is aggregated.

SAPI_SENDER_HOSTNAME_FLD "SenderHostName"

Host sending the message.

SAPI_SENDER_IP_FLD "SenderIP"

IP of host sending the message.

SAPI_SENDER_PORT_FLD "SenderPort"

Port number of the message sender.

Example

"9876"

SAPI_RECEIVER_HOSTNAME_FLD "ReceiverHostName"

Host receiving the message.

SAPI_RECEIVER_IP_FLD "ReceiverIP"

IP of host receiving the message.

SAPI_RECEIVER_PORT_FLD "ReceiverPort"

Port number of the message receiver.

Example

“8765”

Detailed Tracking Events

Both Windows NT and eTrust Access Control offer detailed tracking—in Windows NT, for processes (by PID). In eTrust Access Control, tracking can be activated for other fields as well.

SAPI_CATEGORY_FLD "Category"

"Detailed Tracking" SAPI_CATEGORY_TRACKING

SAPI_OPERATION_FLD "Oper"

"Start" SAPI_OPER_START
"Stop" SAPI_OPER_STOP

SAPI_OBJCLASS_FLD "ObjClass"

"PROCESS" SAPI_OBJCLASS_PROCESS

SAPI_PID_FLD "PID"

The process ID.

SAPI_OBJNAME_FLD "ObjName"

The object name, name of started or stopped program

SAPI_INFO_FLD "Info"

The event description.

SAPI_USER_FLD "User"

The user name.

SAPI_USERID_FLD "UID"

The user ID.

SAPI_SURROGATEUSER NAME_FLD "SurrogateUser"

The name of new identity of a user who changed his identity via set user etc. (available on systems that retain the original identity).

Example

UNIX—for set user operation, UserName may be "john" and SurrogateUser may be "root"

SAPI_SURROGATEUSERID _FLD "SurrogateUId"

The ID of the SurrogateUser, as explained above.

SAPI_EUSERNAME_FLD "EffectiveUser"

The effective user name. The effective user is the user whose rights are in effect for the described event.

SAPI_EUSERID_FLD "EffectiveUserId"

The ID of the effective user, as explained above.

System/Application, Administration and General Events

These events include start and stop notifications for applications not directly involved in security auditing (that is, not mapped to another category). Fields will be application-specific. Identification fields are mandatory.

SAPI_CATEGORY_FLD "Category"

"System and Application" SAPI_CATEGORY_STATUS
"Administration" SAPI_CATEGORY_ADMIN
"General" SAPI_CATEGORY_GENERAL

SAPI_INFO_FLD "Info"

The free-text event information.

Event Fields Internal to eTrust Audit

Internal fields may be filled for each event by eTrust Audit. These fields may be present in each record, but need not be filled by third-party submitters.

SAPI_ROUTINGINFO_FLD "RoutInfo"

For debug purposes only—a concatenation of the names of all the routers that have handled the event.

SAPI_RULENAME_FLD "Rule"

For debug purposes only—name of the eTrust Audit policy that originated the event.

Reserved Keywords

The following words may not be used as field names, since they have specific meanings in the filter language.

ADD
AM
AT
CASE
CI
CS
DATE_YACC
DAY
DECR
DECREMENT
DEFINE
DELETE
DELETE_YACC
DIFFERENT
DY
EQUAL
EXISTS
FATAL_ERROR
GREATER
INCR
INCREMENT
INSENSITIVE
INTEGER
LESS
MATCHES
MONTH
NAME
NEWEVENT
NOT
NUMBER
OF
OR
PART
PM
REL_OP
SCAN_ERROR
SENSITIVE
SET
STRING
STRING_CONST
SUB
SUBTRACT
THAN
TIME

TIMESTAMP
TO
VARIABLE
YR

The names of months (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, and DEC) are also reserved.

Appendix B: Encup Utility

This section contains the following topics:

Introduction

The Encup utility is used to encrypt a string using the outgoing encryption (see page 87) method and a default Audit key. Encup passes a buffer that contains a user name or a password associated with that user name. The source of the information is a file or standard input.

In Audit, the user name and password are never stored in plain text. They must be encrypted in a way so that other Audit components can decrypt and use the information to authenticate. The Encup utility is used, in this instance, to get the encrypted form of new user names and/or passwords. The encrypted form is either displayed on the screen or captured in a file. It then replaces the old encrypted user name or password in the appropriate registry entries (Windows) or ini entries (Unix).

For example, the Policy Manager, Collector, Viewer and Reporter need the corresponding Audit database user name and password to login to the Audit database. Database recorders are also another example, since they need to authenticate to user databases with different users in order to collect Audit events.

Note: It is highly recommended that you save backup copies of the registry or .ini files before making changes.

Help and Usage

Enter: *encup -help* to get a list of additional options.

Following are some useful options:

- encup -i   encup does not prompt, displays the user’s input, doesl not ask for confirmation, and displays the encrypted string.
- *encup -o filename* encup stores the encrypted string in file filename.

Executing Encup

The Encup utility is located in the *install_dir*/bin directory, where *install_dir* is the directory where you installed eTrust Audit.

For more information about the Encup utility, follow these steps:

1. Open a command prompt session.
2. Enter the following command from the *install_dir*/bin directory:

   ```
   encup –help
   ```

Note: Also see Encryption section this guide.

Basic Encup Usage - Windows

To use the encup utility on Windows systems

1. Access a command prompt.
2. Change the directory to Audit_Install_Dir\bin.
3. Enter the command, encup.

   (You can press Control-C at any time to exit encup.)

   Encup prompts for UserName or Password.
4. Enter the UserName or Password that you want to encrypt.
5. Enter the user name.

   To keep the information hidden, Encup will NOT display the string just entered.
6. Confirm the UserName or Password when Encup prompts you to re-enter the same string for confirmation.

   If the second string is different from the first one, Encup prints an error message and prompts again to enter the user name.

   If the second string is the same, Encup prints the encrypted form. For example:

   ```
   6bab15697fe25c9aad47a85614c5b35877733c925067e2775b9e81c3d53e8131df8dd6ae
   ```

   Encup automatically exits.
7. Copy and paste the encrypted string into the appropriate registry key.
8. Repeat the procedure for the password, if necessary.

Basic Encup Usage - UNIX

To use the encup utility on UNIX systems

1. Log in into a terminal session.
2. Change directory to Audit_Install_Dir\bin at the shell prompt.
3. Enter the command encup.

   (You can press the Interrupt key sequence (usually Control-C) at any time to exit encup.)

   Encup prompts for UserName/Password:
4. Enter the UserName or Password that you want to encrypt.

   To keep the information hidden, encup will NOT display the string just entered.
5. Confirm the UserName or Password when Encup prompts to reenter the same string for confirmation:

   If the second string is not the same as the first one, encup prints an error message and prompts you to enter the user name again.

   If the second string is the same, encup prints the encrypted form. For example:

   ```
   6bab15697fe25c9aad47a85614c5b35877733c925067e2775b9e81c3d53e8131df8dd6ae
   ```

   Encup automatically exits.
6. Copy and paste the encrypted string into the appropriate registry key.
7. Repeat the procedure for the password, if necessary.

Appendix C: Digital Certificates

This section contains the following topics:

Digital Certificates Table and Notes

The following table, and the notes that come afterward, describe the digital certificates in use with eTrust Audit and other eTrust products:

Certificate File Name	Default Location	Purpose	First Issuance	Expiration	Regenerate after Expiration?	Method to Regenerate
iGateway.p12	$IGW_HOME	iGateway server certificate	At installation	3650 days (10 years)	Optional	iGateway itself regenerates the iGateway.p12 if the file is missing from $IGW_HOME. To regenerate manually: 1. Stop the iGateway service. 2. Remove igateway.p12. 3. Restart the iGateway service.
AuditAdminCert.p12	$IGW_HOME	Audit application certificate for eIAM server.	When registering with eIAM (IAMT) server.	3650 days (10 years)	Yes	From $Audit_HOME/IAMT, execute the commands: LD_LIBRARY_PATH=.;$LD_LIBRARY_PATH; export LD_LIBRARY_PATH **Windows**: Safex –f AuditIssueCert_win32.xml –h <IAMThost> -u EiamAdmin –p <EiamAdminPassword> **Solaris**: Safex –f AuditIssueCert.xml –h <IAMThost> -u EiamAdmin –p <EiamAdminPassword>

Certificate File Name	Default Location	Purpose	First Issuance	Expiration	Regenerate after Expiration?	Method to Regenerate
igcert.p12	$IGW_HOME	Not needed	N/A			
rootcert.p12	$IGW_HOME	iAuthority's Certificate p12 file	At installation	3650 days (10 years)	Optional	iAuthority itself regenerates the rootcert.p12 if the file is missing from $IGW_HOME. To regenerate manually: 1. Stop the iGateway service. 2. Remove rootcert.p12. 3. Start the iGateway service.
rootcert.pem	$IGW_HOME	iAuthority's Certificate pem file	Same as rootcert.p12.	3650 days (10 years)	Optional	iAuthority itself regenerates the rootcert.p12 if the file is missing from $IGW_HOME. To regenerate manually: 1. Stop the iGateway service. 2. Remove rootcert.p12. 3. Start the iGateway service.
iPozDsa.pem	$IGW_HOME	Copied as itechpoz-<Hostname>.pem	At installation	3650 days (10 years)	Optional	1. Stop the iGateway service. 2. Set GenerateDsaCerts to true in iPoz.conf. 3. Restart the iGateway service.
iPozRouterDsa.pem	$IGW_HOME	Copied as itechpoz-<Hostname>router.pem	At installation	3650 days (10 years)	Optional	1. Stop the iGateway service. 2. Set GenerateDsaCerts to true in iPoz.conf. 3. Restart the iGateway service.
trusted.pem	$DXHOME/config/ssld	eTrust Directory Trusted Root Certificate(s)	May 29 01:36:03 2003 GMT	May 27 01:36:03 2008 GMT	Yes	Run the dxcertgen utility, or obtain a product upgrade, if available.

Certificate File Name	Default Location	Purpose	First Issuance	Expiration	Regenerate after Expiration?	Method to Regenerate
iTechPoz-trusted.pem	$DXHOME/config/ssld	eTrust IAM Trusted Root Certificate (same as rootcert.pem)	Same as rootcert.p12.	3650 days (10 years)	Optional	Copy rootcert.pem and change the filename.
iTechPoz-*<hostname>*.pem	$DXHOME/config/ssld/personalities	eTrust IAM iTechPoz DSA personality (same as iPosDsa.pem)	At installation	3650 days (10 years)	Optional	Copy iPozDsa.pem and change the filename.
itechPoz-*<hostname>*Router.pem	$DXHOME/config/ssld/personalities	eTrust IAM iTechPoz host router DSA personality (same as iPosRouterDsa.pem)	At installation	3650 days (10 years)	Optional	Copy iPozRouterDsa.pem and change the filename.
dxadmin.pem	$DXHOME/config/ssld/personalities	eTrust Administration Daemon personality	Aug 31 05:24:19 2004 GMT	Aug 30 05:24:19 2009 GMT	Yes	New dxadmin.pem files are provided by CA Support or through a product upgrade.

p12 File Notes:

- The p12 certificate format is PKCS#12, and all are password protected and self-signed.
- The p12 certificates are signed using SHA-1 (Digest size 160 bits); RSA pub-priv key size 1024 bits.

pem File Notes:

- The pem certificate format is Base-64 Encoded, and all are self-signed except the dxadmin.pem file, which is signed by CA.
- The files trusted.pem and dxadmin.pem have Signature=md5WithRSAEncryption, Hash Algorithm=MD5, and Strength=1024. The remaining files have Base-64 Encoding of DER format ASN1 data obtained from the related p12 cert.

Regeneration Notes:

If you obtain a new root certificate, then you need to regenerate DSA certificates (iPozDsa.pem, iPozRouterDsa.pem, iTechPoz-<*hostname*>.pem, and iTechPoz-<*hostname*>Router.pem) using the Dxcertgen utility from that root certificate. The Dxcertgen utility does not generate root certificates – it creates DSA certificates from given a root certificate.

Appendix D: Disaster Recovery

This section contains the following topics:

Disaster Recovery Introduction

Disaster recovery scenarios typically involve two computer systems. A primary server provides normal services. A secondary server, which has a mirror configuration of the primary server, plays a standby role. The secondary server remains ready to be activated if the primary server fails.

The r8 SP2 Data Tools, Policy Manager, and Reporter-Viewer server components support a two system disaster recovery scenario if the following conditions are fulfilled:

- Replicate the Policy Manager database between the primary and secondary servers periodically.
- Replicate the Reporter-Viewer's scheduled Reporter jobs between primary and secondary servers periodically. These jobs are located by default in the directory, <Audit_dir>/dat/Reports.
- Replicate the Reporter-Viewer's user-defined Viewer filters between primary and secondary servers periodically. These jobs are found in the directory, <Audit_dir>/dat/Filters.
- Run the utility, ac_pmdb2hd (see page 352), to generate Policy Manager queues and compiled objects for distribution and auto-correction of tampered Clients at the time you switch from primary to secondary. The distribution and polling queues are located in the directory, <Audit_dir>/dat/queue/distrib. Compiled policy files are located in the directory, <Audit_dir>/dat/policy. Compiled MP files are located in the directory, <Audit_dir>/dat/mp.

ac_pmdb2hd Utility

The ac_pmdb2hd utility is used to create required recovery objects from the Policy Manager database. These objects are needed for disaster recovery between a primary and a secondary server. Use this utility to generate the following objects from the Policy Manager database:

- Folders of compiled Policies and message parsing (MP) files for all distributed groups. These policies and MP files are needed for handling distribution to the Audit Nodes (ANs).
- Polling queues for all registered hosts. These are needed for polling Clients to request end-point status, such as Client status information about the policies/MP files received and whether they have been tampered.

This utility requires Administrator or root privilege to run.

Initially, the utility stops the Distribution Server and iGateway services before generating the needed data. When the utility completes processing, it automatically restarts the Distribution Server and iGateway services.

ac_pmdb2hd Utility Syntax

The ac_pmdb2hd utility uses the following syntax:

```
ac_pmdb2hd [-help] [-pollq del|keep] [-distq del|keep] [-pdir del|keep] [-mpdir del|keep] [-poll on|off]
```

-pollq [del|keep]

This parameter specifies whether the utility should delete the polling queue before restore. The default value is del.

-distq [del|keep]

This parameter specifies whether the utility should delete the distribution queue before restore. The default value is del.

-pdir [del|keep]

This parameter specifies whether the utility should delete the Policy folder before restore. The default value is del.

-mpdir [del|keep]

This parameter specifies whether the utility should delete the MP folder before restore. The default value is del.

-poll [on|off]

This parameter turns polling on or off. The default value is on.

Disaster Recovery Examples

To activate the secondary server's Policy Manager

1. Access a command prompt on the secondary server.
2. Ensure that the Policy Manager database is online.
3. Run the ac_pmdb2hd utility from the secondary server using the command:

   ```
   ac_pmdb2hd –poll on
   ```

   This command recreates the distribution and polling queues, generates the compiled policies and MP files, and turns on polling.

   Since polling is on, the Distribution Server polls each audit node (Client) in the queue to get respective end-point status. It then compares that status with information in the Policy Manager database. If there is any discrepancy, the Distribution Server auto-corrects the audit nodes by sending the right policies and/or MP files.

To reactivate the primary server's Policy Manager

1. Access a command prompt on the secondary server.
2. Stop the Distribution Server and iGateway services.
3. Access a command prompt on the primary server.
4. Run the ac-pmdb2hd utility using the command:

   ```
   ac_pmdb2hd –pollq keep –distq keep –poll on (or off)
   ```

   This command keeps the existing distribution and polling queues. It can optionally turn polling on or off depending on what has been previously activated in the primary.

Appendix E: Using the eTSAPISend Program

This section contains the following topics:

Introduction

eTSAPISend.exe is a program that lets you send messages and events to an eTrust Audit router. This executable does not depend on eTrust Common Services, so you can run it even if the Common Services is not installed.

The following describes the characteristics and usage of eTSAPISend:

Options

For predefined fields, you can specify the field values by using the command line options as described in the following list:

-cat

The event category. Enclose the category in double quotes.

Possible values: 'System Access', 'Account Management', 'Object Access', 'General', 'Policy Management', 'Security Systems', 'Physical Security', 'Network', 'Detailed Tracking', 'System and Application', 'Administration'.

-dat

The date and time whose value is either MM/DD/YYYY or MM/DD/YYYY HH:MM:SS. Enclose the date value in double quotes. If you do not specify this optional parameter, the local system time is automatically applied to the message or event.

-evt

The event type, which should be empty unless it is an Alert type. Any number larger than 0 will set the event as Alert.

-inf

Detailed information in the message. Enclose the details in double quotes.

-loc

The location (\\Domain\Computer) where the event originated. Enclose the location name in double quotes.

-nam

The logical log name that uniquely identifies the native auditing type.

-nid

The native ID (Event ID). This is a number.

-nod

The target eTrust Audit router node. If you do not specify this option, the message or event is sent to the eTrust Audit router on the local host.

-opr

The operation that was performed.

Possible values: 'Login', 'Logoff', 'Create', 'Delete', 'Restart', 'Startup', 'Shutdown', 'Clear', 'Open', 'Lock', 'Unlock', 'Connect', 'Disconnect', 'Start', 'Stop'.

-src

The submitter, such as the OS, process name, or application issuing the event.

-sta

The event status.

Possible values: 'S', 'F', 'D'.

-usr

The user name associated with the event or message.

-port

eTrust Audit Router Port

-sev

Event Severity.

Possible values: '0','1','2','3'.

For more details , see The Submit API (SAPI) (see page 305).

User-defined Options

For each user-defined field, add the field name followed by the field value as command line arguments. You can include any number of predefined and user-defined fields in a message or event.

Syntax

You specify the message fields as command line options. The fields can be predefined or user-defined.

Enter the command as follows:

```
eTSAPISend options
```

Note: On Solaris platforms, add /usr/ucblib to the LD_LIBRARY_PATH variable.

Example

Consider the following sample command:

```
eTSAPISend -nod systema -cat "System Access" -opr Logon -sta F
-nam NT-Security -loc "\\MYDOMAIN\SYSTEMA" -usr SYSTEM -evt 70 -src Security
-nid 529 -inf "Logon Failure" -dat "08/06/2002 16:00:30" User-defined SomeValue
```

This command does the following:

- Sends the message to the eTrust Audit router on systema.
- The category of the message is System Access.
- The operation performed is a Logon.
- The status of the message is F, for failed.
- The logical name of the log file from which the message was sent is NT-Security.
- The location of the source where the message originated is SYSTEMA machine on the MYDOMAIN domain.
- The user is SYSTEM.
- The event type is 70.
- The submitter of the event is Security.
- The event id is 529.
- The text of the message is Logon Failure.
- The date on which the event occurred is 08/06/2002 at 16:00:30.
- There is a user-defined value of SomeValue.

Sample Batch File

The following is an example of how to issue eTSAISend in batch:

```
REM Failed Logon

eTSAPISend -nod systemb -cat "System Access" -opr Logon -sta F -nam NT-Security -loc "\\mydomain\systema" -usr SYSTEM -evt 70 -src Security -nid 529 User-defined1 SomeValue1 -inf "Logon Failure" User-defined2 SomeValue2

eTSAPISend -nod systemb -cat "System Access" -opr Logon -sta F -nam NT-Security -loc "\\mydomain\systema" -usr SYSTEM -evt 70 -src Security -nid 529 -inf "Logon Failure" -dat "08/06/2002 16:00:30" User-defined SomeValue

REM User Account Changed

eTSAPISend -nod systemb -cat "Account Management" -sta S -nam NT-Security -nid 642 -inf "User Account Changed" -loc "\\mydomain\systema" -usr SYSTEM -src Security

REM Critical File Access Failure

eTSAPISend -nod systemb -nam NT-Security -cat "Object Access" -nid 560 -inf "Object Type: File" -sta F -loc "\\mydomain\systema" -usr SYSTEM -src Security
```

Appendix F: Status and Maintenance Utilities

This section contains the following topics:

Introduction

The following utilities help you to gather system status information and to manage database passwords.

Audit Status (Acstat) Utility

The acstat utility is a command-line utility used to display the status of the eTrust Audit installation on the current host.

The status includes information like the following:

- Host system information such as hardware information, memory, OS version, and disk space
- NLS data
- Environment information
- IP configuration and network status (NETSTAT) information
- HOSTS file entries
- List of installed Java runtime executables (JRE)
- List of installed products
- eTrust Audit information such as Audit version, last installation time, installation directory, cipher used for outgoing messages, list of ciphers used to decrypt incoming messages, registry key values for Windows installations, and so forth
- List of services and their statuses
- Component name and version, including running modules and start time, and more

The utility is located in the *bin* directory under the eTrust Audit installation directory. For Windows installations, the default location is \Program Files\CA\eTrust Audit\bin. For UNIX or Linux installations, the default location is /opt/CA/eTrustAudit/bin.

To run the acstat utility

1. Access a command prompt using one of the following methods:
2. Windows: Click Start then Run. Enter: command or cmd. A console window opens.
3. UNIX or Linux: Enter the command telnet or login, or start a Terminal console if running X-Windows.
4. Navigate to the eTrust Audit installation directory, then to the bin directory.
5. Enter the following command:

   ```
   acstat
   ```

   The eTrust Audit installation status is displayed.

Database Password Change (Acchgpwd) Utility

The *acchgpwd* utility is used to change the password of the Audit Database user account.

The new encrypted password replaces the old one in the Windows registry or in the UNIX *eaudit.ini* configuration file.

Running acchgpwd

The acchgpwd utility is a command-line utility.

It is located in the bin directory under the Audit installation directory. Usually for Windows, it is located at \Program Files\CA\eTrust Audit\bin. For Unix or Linux, it is located typically at /opt/CA/eTrustAudit/bin.

To run acchgpwd, execute the following steps:

1. Access a command prompt.
2. For Windows, click Start->Run. Enter: command or cmd. A console window opens.

   For UNIX/Linux, either run telnet, log in, or start a Terminal console if running X-Windows.
3. Navigate to the Audit installation directory. Then navigate to the bin directory.
4. Enter the command, acchgpwd.

   After displaying the Audit copyright notice, acchgpwd prompts for the old password.
5. Enter the old password.

   Note: The acchpwd utility will not validate input until the last prompt. Also, all inputs are masked by asterisks '*'.

   Acchgpwd then prompts for the new password.
6. Enter the new password.

   Acchgpwd prompts again for the new password for confirmation.
7. Re-enter the same password that you entered in step 5 above.

   Acchgpwd then validates all input. If the input is valid, that is, the old password is the one in use, or the new password and the re-entered password are the same, acchpwd then encrypts the new password, replaces the old with the new one in the Windows registry or Audit ini file and then displays the message: "Password has been changed". The new password takes affect at the next restart of the Collector service.

If the old password does not correspond to the one in use, acchgpwd responds with: "Wrong password" then exits with no change to the current configuration. If the confirmation for the new password fails, acchgpwd displays: "Invalid password" then exits with no change to the current configuration.

The following is an example of a successful session with acchgpwd:

```
$ acchgpwd
eTrust Audit r8.0 (67.26)
Copyright 2004 Computer Associates International, Inc.
Enter old password: ****
Enter new password: ***
Confirm new password: ***
Password has been changed
```

Collector Status (Acreminfo) Utility

Acreminfo is used to display status information about the Collector (version, system usage, configuration) as well as bulk insert statistics (receive rate, insert rate, and so forth.) Descriptions of the values listed by the utility are available in Acreminfo Output Descriptions (see page 367).

The output of acreminfo is as follows:

```
C:\Program Files\CA\eTrust Audit\bin>acreminfo
------------------------------------------------------------
     acreminfo

Usage : acreminfo -collector [-host <host>] [-port <port>]
    [-info ver|proc] | [-stat reset|interval <value>]
Type          :  collector
Host          : 'localhost'
Port          :  0
------------------------------------------------------------

eTrust Audit Collector on host 'aulab06' MS WinNT5.0

     Configuration
     _____________

Collector Port                 0
Collector Sapi Port            0
Portmap Name                   portmap.exe
DB Plugin Name                 OCCD
Maximal Buffer Size for Insert    16777216 K
Maximal record number for Insert  200
Maximal Time Wait for Insert      5 sec
Maximal Number of DB Sessions     21

     System Info
     ___________

Total Physical Memory          2097151 K
Free Physical Memory           1835952 K

Current Working Set Size       16204 K
Current Pagefile Usage         12412 K

     ODBC
     ____

eAudit_DSN
Microsoft SQL Server ODBC Driver   2000.81.9031.38
```

```
        Statistics
        __________

Current Time                 Thu Nov 11 21:19:35 2004
Start Time                   Thu Nov 11 18:52:14 2004
Statistics Start Time        Thu Nov 11 20:49:17 2004
Events read from queue files     387 events from 21 file(s)
Active Threads               1
Total Events Received        1156987
Total Events Inserted        1156800
Total Events Rejected        0
Average Receive Rate         636.4 event/sec
Average Insert Rate          636.3 event/sec
Thread Slot Status           .*...................
Disconnect Time              N/A
Reconnect Time               N/A
Last Error Time              N/A
Last Error
Active Connections           21
Open Connections             20
Busy Connections             1
Bad Connections              0

Sampling Interval            5 sec
Events Received              3902
Events Inserted              3800
Events Rejected              0
Receive Rate                 780.4 event/sec
Insert Rate                  760.0 event/sec
```

Acreminfo Output Descriptions

The following describes the fields displayed as output from the acreminfo utility:

Current Time

The time at which the acreminfo utility was most recently run, as reported by the host operating system.

Start Time

The time at which the Collector service most recently started/restarted.

Total Events Inserted

The total number of events successfully inserted into the database since the Start Time.

Total Events Rejected

The total number of events that failed insertion into the database since the Start Time.

Average Receive Rate

The total number of events received by the Collector (whether inserted or rejected), divided by the difference between the Start Time and the Current Time. (units: events/sec)

Average Insert Rate

The total number of events inserted by the Collector into the database, divided by the difference between the Start Time and the Current Time. (units: event/sec)

Sampling Interval

The interval, usually 5 seconds, at which samples are taken for the Events Received, Events Inserted, Events Rejected, Receive Rate, and Insert Rate statistics. (units: seconds).

Events Received

The total number of events received by the Collector (whether inserted or rejected) during the Sampling Interval period.

Events Inserted

The total number of events successfully inserted by the Collector into the database during the Sampling Interval period.

Events Rejected

The total number of events that failed insertion into the database during the Sampling Interval period.

Receive Rate

The Events Received value divided by the Sampling Interval value. (units: events/sec)

Insert Rate

The Events Inserted value divided by the Sampling Interval value. (units: events/sec)

Index