eTrust® Access Control for UNIX and Linux

Utilities Guide

r8 SP1

CA Product References

This document references the following CA products:

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

Contact Technical Support

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

Contents

Chapter 1: Utilities by Category 9

eTrust AC Utilities 9
Categories 9
User Utilities 10
Administrator Utilities 11
Installation Utilities 13
Support Utilities 14
Password Utilities 14
Daemons 14

Chapter 2: Utilities in Detail 17

ChangeEncryptionMethod 17
dbmgr 17
- dbmgr -create Function—Create a Database 18
- dbmgr -dump Function—Display Database Information 20
- dbmgr -export Function—Create Script 23
- dbmgr -migrate Function—Copy Data to a Flat File 25
- dbmgr -util Function—Manage Existing Database 27
- dbmgr -backup Function—Backup a Database 29
- dbmgr -restore Function—Restore a Database 29

defclass 30
DictImport 31
dmsmgr 31
- -create Function—Create a DMS or a DMA 32
- -remove Function—Remove a DMS or a DMA 33
- -cleanup Function—Remove Obsolete Nodes 33

eacpg_gen 34
exporttngdb 37
issec 38
migopts 39
policydeploy 40
policyreport 42
seagent 46
seaudit 47
seauxd 59
sebuildla 60
sechkey 65

seclassadm 68
secompas 72
secons 74
secrepsw 85
sedbpchk 86
seerrlog 88
segrace 89
segracex 91
seini 93
selang 96
seldapcred 103
seload 104
selock 106
selockcom 112
selogmix 114
selogrcd 116
selogrd 118
semsgtool 133
senable 135
senone 137
SEOS_load 138
SEOS_syscall 139
seosd 140
seostngd 147
seoswd 150
sepass 153
sepmd 163
Administering Subscribers 164
Truncating the Update File 166
Dual Control 167
Managing the Policy Model Log File 169
Other PMDB Administration 170
sepmdadm 172
sepmdd 177
sepropadm 183
sepurgdb 185
sereport 186
seretrust 191
serevu 193
sessfgate 199
sesu 200
sesudo 202

seuidpgm 207
seversion 210
sewhoami 211
uninstall_eTrustAC 212
UxImport 213

Appendix A: Trace Messages 217

Conventions 217
Messages 217

Appendix B: The lang.ini File 239

lang.ini File Tokens 240
general 240
history 241
newres 242
newusr 243
properties 244
User-Defined Properties 244
The Definition Files 244
The Tokens File 245
The Attributes File 246
unix 247

Appendix C: String Matching 249

Wildcard Expressions 249
Wildcard Matching 249
Character Lists 250
Examples: Wildcard Matching 251

Index 253

Chapter 1: Utilities by Category

This section contains the following topics:

eTrust AC Utilities (see page 9)
Categories (see page 9)
User Utilities (see page 10)
Administrator Utilities (see page 11)
Installation Utilities (see page 13)
Support Utilities (see page 14)
Password Utilities (see page 14)
Daemons (see page 14)

eTrust AC Utilities

eTrust AC has many utilities. As a convenient overview, this chapter classifies them by category. Some utilities are listed more than once. For a description of the utilities arranged alphabetically, see the chapter “Utilities in Detail.”

Categories

This chapter groups the eTrust AC utilities into the following categories:

- **User utilities** for typical users of the system
- **Administrator utilities** for administrators to manage and configure eTrust AC
- **Installation utilities** for product installation, system startup, or the removal of eTrust AC from the system
- **Support utilities** for technical support
- **Password utilities** for replacing passwords
- **Daemons** for performing eTrust AC functions

User Utilities

defclass

Defines basic Unicenter TNG asset types in each database and every new PMDB that is defined.

exporttngdb

Migrates the current Unicenter Security data into a local eTrust AC database or PMDB.

segrace

Displays various login and password settings for a user.

segracex

Allows user to replace an expired password.

selock

Locks the user's screen and displays a screen saver.

selockcom

Controls the selock utility.

senone

Executes a shell as if it were invoked by a non eTrust AC user.

sepass

Serves in place of the UNIX passwd and yppasswd commands.

sesu

Serves in place of the UNIX su command.

sesudo

Executes commands for one user with the permissions of another user.

sewhoami

Serves in place of the UNIX whoami command and reports the eTrust AC username, which is harder to change than the UNIX username.

Administrator Utilities

dbmgr

Creates, manages, and maintains the eTrust AC database.

dmsmgr

Creates or removes a DMS or a DMA from an eTrust AC computer, or maintains the DMS database to remove obsolete objects.

eacpg_gen

Automatically generates eTrust AC control policies.

ChangeEncryptionMethod

Changes the encryption method of existing policy models.

issec

Displays the eTrust AC security daemons' status.

policydeploy

Deploys or removes a policy from a Policy Model hierarchy or on an eTrust AC end-point.

policyreport

Generates offline (static) HTML reports based on information in a DMS.

seaudit

Displays selected data from the eTrust AC audit log.

sebuildla

Creates a lookaside database.

sechkey

Changes the encryption key for various eTrust AC programs.

seclassadm

Adds new classes to the eTrust AC database.

secons

Controls the eTrust AC daemons.

secrepsw

Creates password file without shadowing.

sedbpchk

Checks the integrity of the eTrust AC database. Backs up the database if the database passes the check.

seerrlog

Lists records in the eTrust AC error log.

selang

Invokes the selang command shell.

seldapcred

Encrypts and stores a provided credential for use by LDAP-enabled eTrust AC utilities (such as sebuildla) for retrieving data from an LDAP Directory Information Tree (DIT). Together with the value of the ldap_userdn token in the [seos] section of the seos.ini file, it lets the utility authenticate to the LDAP service.

selogmix

Splits and merges audit files.

semsgtool

Maintains, decodes, and creates eTrust AC message files.

senable

Re-enables a previously disabled user account.

sepmd

Administers PMDBs.

sepmdadm

Creates PMDBs.

sepurgdb

Purges the eTrust AC database.

sereport

Provides reports-accessible from a web browser-of database and Policy Model information.

seretrust

Retrusts untrusted programs.

seversion

Displays the version information of an eTrust AC module.

uninstall_eTrustAC

Removes eTrust AC from the station.

Installation Utilities

DictImport

Imports an external dictionary to the eTrust AC database for password checks.

exporttngdb

Migrates the current Unicenter Security data into a local eTrust AC database or PMDB.

migopts

The eTrust AC program run at installation that translates the current Unicenter Security environment into the global settings of either a local eTrust AC database or PMDB.

seload

The utility that loads the eTrust AC extension to the UNIX kernel and executes the eTrust AC daemons.

SEOS_load

The eTrust AC interception module loader for all stations except Sun Solaris.

SEOS_syscall

The eTrust AC interception module.

seostngd

eTrust AC synchronization daemon (for Unicenter TNG).

sepropadm

The administrator of eTrust AC database properties.

seuidpgm

The extractor of the setuid programs in a UNIX file system.

UxImport

The extractor of the user, group, and host information in a UNIX system and, if installed, in NIS.

uninstall_eTrustAC

The utility for removing eTrust AC from the station.

Support Utilities

sedbpchk

Checks the integrity of the eTrust AC database, and if it passes, backs up the database.

seini

Displays information about the eTrust AC database and initialization files and sets the values of tokens in the initialization files.

Password Utilities

secompas

Compares UNIX and eTrust AC passwords for all eTrust AC users.

sepass

Serves in place of the UNIX passwd and yppasswd commands.

Daemons

mfsd

Daemon for mainframe synchronization.

seagent

eTrust AC agent daemon (the Agent).

seauxd

eTrust AC auxiliary daemon.

selogrcd

Collector daemon for the eTrust AC log routing system.

selogrd

Transmitter daemon for the eTrust AC log routing system.

seosd

eTrust AC authorization daemon (the Engine).

seoswd

eTrust AC watchdog daemon (the Watchdog).

sepmdd

Policy model daemon.

serevu

Daemon for dealing with users who have committed too many login failures.

sersvd

Daemon enabling the Remote Status View (RSV).

sessfgate

Daemon to route reformatted Unicenter Security APIs through the message queue to eTrust AC.

Chapter 2: Utilities in Detail

ChangeEncryptionMethod

Changes the encryption methods of policy models. Three encryption methods are available. When you run this utility, you are asked to choose one of the following encryption methods:

- AES (128bit, 192bit, or 256bit)
- DES
- TRIPLEDES
- SCRAMBLE

After you choose the method, the utility searches for existing Policy Models in the system, decrypts them by running "sepmd -de pmd_name", and then changes the encryption method by linking libcrypt to the new shared library: libdes, libtripledes, or libscramble.

Note: To run ChangeEncryptionMethod eTrust AC needs to be running. To change the encryption method, the utility asks you whether it can temporarily shut down eTrust AC.

dbmgr

Creates, manages, and maintains the eTrust AC database files.

Note: This utility replaces the following utilities from previous versions: dbdump, rdbdump, dbutil, secredb, sedb2scr, and semigrate.

The dbmgr utility handles several tasks, each described separately in this section:

- Creating a new database
- Generating reports on database records
- Creating a script that defines a database
- Copying data from a database to a flat file
- Managing and maintaining a database

dbmgr -create Function—Create a Database

The dbmgr -create function generates a new empty database. Use this function only at installation time, or when you want to create a database or PMDB. eTrust AC creates the database in the current directory. When you run dbmgr with the -create function, it automatically adds a user called root, with the ADMIN, AUDITOR, and IGN_HOL attributes.

Notes:

- Use this function only for creating a new database.
- If you want to add user-defined classes to the new database, first run the seclassadm utility after creating the new database. This utility saves data about the new classes in the registry file. However, before adding the classes to the database, be sure the CreateNewClasses token (in the [seosdb] section of the seos.ini file) is set to the default value: yes.

Syntax

```
dbmgr {-create | -c} switch [option]
```

Switches

-c

Creates a new database.

-cq

Creates a new database without a user prompt.

-h

Lists the help screen.

Options

-d

Prints database layout documentation. The output contains a full description of the structure and property formats used in the database. You cannot use this option with the -v option.

-f *filename*

Directs output to the specified file, instead of the standard output device. You must include this option when working from the UNIX GUI.

-n

Specifies the location (full path) of the eTrust AC database to back up.

When you are creating a new database, a basic class scheme is generated. When you are adding new classes to the database using the seclassadm utility, the class information is stored in a file in the database directory. In order to back up a specific database with its class scheme (such as a policy model database), specify its location with the -n option. The user-defined class information is taken from that location. If you do not specify the -n option, the class information file is searched for in the local directory were the database is to be created. If it is not found there, the file is taken from the active eTrust AC security database directory.

-o

Adds Unicenter TNG classes to an existing database.

-t *terminalName*

Creates the specified terminals in the database and authorizes them to the users specified with -u.

-u *userNames*

Creates the specified users in the database and defines them as administrators.

-v

Disables the progress messages. You cannot use this option with the -d option.

-w

Creates a new database that includes Unicenter TNG classes.

Files

The -create function uses the following files:

seos.ini

- lang.ini
- seos_new.cls

dbmgr -dump Function—Display Database Information

The dbmgr -dump function reports on the records in the database. If you include the -r switch, the function operates on the database currently being used by the authorization daemon; otherwise, it operates on the database located in the current directory. This function performs the following operations:

- Displays information for records of a specified class
- Displays information for a single record of a specified class
- Displays information for all records of a class, except a specified one
- Generates lists of classes and property definitions
- Generates a list of groups that a user belongs to
- Generates a list of records of a particular class

Syntax

```
dbmgr {-dump | -d} switches [options]
```

Switches

-r

Displays information about the database currently being used by the authorization daemon.

If you omit this switch, dbmgr displays information about the database in the current directory.

c

Lists the names of all classes defined in the database.

d *class property* / **dn *class property***

Displays the values of selected properties for all records of a class. The *class* parameter specifies the class. The *property* parameter specifies the list of properties whose values are to be displayed. To specify more than one property, separate the property names with a space. To read the property list from a file, replace *property* with an “at” sign (@) followed by the name of the file. Leave a space before the “at” sign. Each property must appear on a separate line. If you omit the *property* parameter, the values of all the properties are listed.

If you specify the dn switch, properties with unknown values are not displayed.

e *class record property* / en *class record property*

Displays the values of selected properties for all records of a class *except* for a single, specified record. The *class* parameter specifies the class. The *record* parameter specifies the name of the record to omit from the list. The *property* parameter specifies the list of properties whose values are to be displayed. To specify more than one property, separate the property names with a space. To read the property list from a file, replace *property* with an "at" sign (@) followed directly (with no intervening space) by the name of the file. Each property must appear on a separate line. If you omit the *property* parameter, the values of all the properties are listed.

If you specify the en switch, properties with unknown values are not displayed.

fc

Lists all class information for all classes in the database.

fp *class*

Lists all property information on properties of the specified class.

g *user*

Lists the groups that the specified user is a member of.

l *class*

Lists all the records in the specified class.

o *class record property* / on *class record property*

Displays the values of selected properties for a single record of a class. The *class* parameter specifies the class. The *record* parameter specifies the record. The *property* parameter specifies the list of properties whose values are to be displayed. To specify more than one property, separate the property names with a space. To read the property list from a file, replace *property* with an "at" sign (@) followed directly (with no intervening space) by the name of the file. Each property must appear on a separate line. If you omit the *property* parameter, the values of all the properties are listed.

If you specify the on switch, properties with unknown values are not displayed.

p *class*

Lists the names of the properties of the specified class.

Options

-f *filename*

Directs output to the specified file, instead of the standard output device. You must include this option when working from the UNIX GUI.

Files

The -dump function uses the database files, if these files are located in the current directory. The -dump function also uses the seos.ini file.

Notes:

- Only technical support personnel should use this function.
- Specify only one switch with the -dump or -dump -r function.
- This function assumes that the seosd daemon is not running; you must invoke it from the directory where the database resides.
- If you use the -r switch, the seosd daemon must be running, and you must have the ADMIN, AUDITOR, or SERVER attribute.
- To execute this function, you must have READ and WRITE permission on the database files *eTrustACdir*/seosdb/seos_* (where *eTrustACdir* is the directory in which you installed eTrust AC).
- By using the full_year token, you can display the year in two or four digits. The default is “yes,” which means four digits.

dbmgr -export Function—Create Script

The dbmgr -export function generates a script that consists of the selang commands required to define an existing database. These commands are written to standard output. Use this function to replicate a database on other stations.

To write the generated commands to a file, use redirection. You can then create a new database from the file, by instructing selang to read the commands from the file.

Notes:

- Rather than piping the output from this function to selang, you should examine the script before it executes.
- Using Policy Manager, you cannot export a database remotely.

Syntax

```
dbmgr -export | -e switch [option]
```

Switches

-l

Dumps the database in the current directory.

-r

Dumps the database currently being used by seosd.

Options

-c *className(s)*

Dumps the database for the specified class or classes. To use this option, you must precede it, in the same command line, with either the -l or -r switch.

-f *filename*

Directs output to the specified file, instead of the standard output device. You must include this option when working from the UNIX GUI.

Files

The -export function uses the database files; it does not use the seos.ini file.

Note: When you invoked this function with the -l switch, it assumes the eTrust AC daemons are not running. If the daemons are running, then it assumes you are operating on a different database from the one being used by the daemons.

- To use the -r switch, you must have the ADMIN or SERVER attribute, and the eTrust AC daemons must be running.

- You cannot copy database files from one architecture to another when using UNIX commands such as cp or tar, if the files do not use the same byte order. For example, you cannot copy a database from a Sparc based machine to an Intel based machine, because each uses a different byte order.

See Also

The seclassadm, selang, and seerrlog utilities in this chapter.

dbmgr -migrate Function—Copy Data to a Flat File

The dbmgr -migrate function copies data from user and program records in an existing database to a flat file. It can also copy the data from the flat file into a new database. The database from which the data is imported must be version 1.21 or later.

When you copy a flat file into a new database, it is important to use the same version of this function that you used to create the flat file. If you have more than one version, it is strongly recommended that you use the most recent version.

Syntax

```
dbmgr migrate | -m switch [option]
```

Switches

-r *filename*

Read the database in the current directory and copy certain data into the flat file specified in the command line.

-w *filename*

Read the flat file specified in the command line and copy the data into the database in the current directory.

Options

-f *filename*

Directs output to the specified file, instead of the standard output device. You must include this option when working from the GUI.

-s

Read the information from the database using the eTrust AC server rather than reading the database directly. This option is only valid with the -r switch.

To run the command with the -s option, you must have administrator privileges and R (read) and W (write) access to the terminal.

Imported Data

The imported USER data includes the following:

OLD_PASSWD

The old passwords of the user; that is, the user's password history.

PASSWRD_L_C

The date and time the user password was last changed.

PASSWD_L_A_C

The date and time the user's password was last changed by an administrator.

LAST_ACC_TERM

The terminal from which the user last logged in.

LAST_ACC_TIME

The date and time the user record last logged in.

The imported PROGRAM data includes the following:

UNTRUSTREASON

The reason that the program was untrusted.

ACCSWHO

The user who last tried to access the program.

ACCSTIME

The time the program was last accessed.

Notes:

- The -migrate function always reads from or writes to the database in the current directory unless you include the -s option.
- Always create a backup of the database before using this function.
- For better security, delete the old database, the script used to build the new database, and the flat file created by this function after copying the data from the old database into the new database.
- The flat file is written in binary format.
- You cannot use the semigrate command in version 8.0.

Example

The following steps illustrate how to copy data from an existing database into a new database. The old database is assumed to be in the directory /tmp/old_db. The new database is assumed to be in the directory *eTrustACdir*/seosdb (where *eTrustACdir* is the directory in which you installed eTrust AC).

1. Become the superuser by logging in as root or su to root.
2. If the eTrust AC daemons are running, shut them down with the following command:

   ```
   # secons -s
   ```

3. Create a backup of the old database by copying it to a different location or to a backup medium.
4. Copy the database into /tmp/old_db, then create a script that duplicates the old database by running the dbmgr utility on the old database:

   ```
   # cd /tmp/old_db
   # /opt/CA/eTrustAccessControl/bin/dbmgr -export -l > lang_script
   ```

5. Create a new database:

```
# cd /opt/CA/eTrustAccessControl/seosdb
# /opt/CA/eTrustAccessControl/bin/dbmgr -c -cq
```

6. Execute the script generated in the previous step and create the new database:

```
# cd /opt/CA/eTrustAccessControl/seosdb
# /opt/CA/eTrustAccessControl/bin/selang -l /tmp/old_db/lang_script
```

7. Execute the dbmgr utility to create a flat file containing data from the old database:

```
# cd /tmp/old_db
# /opt/CA/eTrustAccessControl/bin/dbmgr -migrate -r flat_file
```

8. Load the data from the flat file into the new database:

```
# cd /opt/CA/eTrustAccessControl/seosdb
# /opt/CA/eTrustAccessControl/bin/dbmgr -migrate -w /tmp/old_db/flat_file
```

The seos.ini File

The -migrate function uses the database files in the current directory; it does not use the seos.ini file.

See Also

- The dbmgr -export function in this section.
- The seerrlog and secons utilities in this chapter.

dbmgr -util Function—Manage Existing Database

The dbmgr -util function performs management and maintenance operation on a database. It assumes eTrust AC is not currently running. Invoke it from the directory where the database resides.

Note: Using Policy Manager, you cannot maintain a database remotely.

Syntax

```
dbmgr -util | -u switches [option]
```

Switches

-all *filename*

Performs all index checks; same as specifying the index and free switches.

-build *filename*

Builds indexes of a DBIO based on data records.

-check

Performs a fast sanity and consistency check on all index entries for all database files. When using this switch, do not use the *filename* parameter.

-close

Closes the database if it is open.

-dump

Dumps the data file as ASCII on the standard output device.

-dup *SourceFile DestinationFile*

Duplicates the DBIO file based on the file header.

-fast

Performs a fast sanity check on all index entries for all the database files. When using this switch, do not use the *filename* parameter.

-free *filename*

Checks for a free index.

-index *filename*

Checks the consistency of the index.

-key *filename*

Sequentially scans an index file.

-load *DatabaseFile ASCIIfile*

Loads an ASCII file and converts it into a DBIO file.

-scan *filename*

Scans the database sequentially.

-scana *filename*

Scans the database sequentially, including deleted records.

-stat *filename*

Lists the header information of the database file.

-verify

Verifies that certain predefined objects exist in the database; for example, SEOS, ADMIN, and UACC for all classes.

Options

-f *filename*

Directs output to the specified file, instead of the standard output device. You must include this option when working from the UNIX GUI.

filename

The name of the database file. The database file has the extension *dat*.

Files

The -util function uses database files, but it does not use the seos.ini file or any other special files.

dbmgr -backup Function—Backup a Database

The dbmgr -backup function creates an online backup of the eTrust AC database in the specified directory.

This function is available whether the eTrust AC daemons are running or not.

Syntax

```
dbmgr -backup | -b backup_directory
```

Files

The -backup function uses database files in the specified directory.

dbmgr -restore Function—Restore a Database

The dbmgr -restore performs an online restore of the eTrust AC database in the specified directory.

This function is available whether the eTrust AC daemons are running or not.

Syntax

```
dbmgr -restore | -r restore_from_directory
```

Files

The -restore function uses database files in the specified directory.

defclass

eTrust AC defines basic Unicenter TNG asset types in each eTrust AC database and every new PMDB that is defined. This script defines user-defined security asset types as eTrust AC classes in the eTrust AC database.

Two scripts, uni_migrate_master.sh and uni_migrate_node.sh automatically execute this shell script file. It can, and should, be called manually whenever a new PMDB is created.

Syntax

```
defclass
```

Files

No special files are used.

DictImport

Prepares dictionary files to be imported into the eTrust AC database with the -f flag or command.

After installing eTrust AC, you must import the dictionary file into the eTrust AC database and then activate it, so you can set password protection.

The DictImport program sets the use_dbdict password rule to **db** and activates the DICTIONARY class and PASSWORD class.

Note: The centralized dictionary is disabled if the PASSWORD class is not active.

Syntax

```
DictImport.sh [-h] [-o selangFilename] [-f dictionaryFilename]
```

Switches

-f *dictionaryFilename*

Generates selang commands that import all the dictionary words from the specified file.

Note: When the -f flag is not set, the Dictionary file defined in the [passwd] section of the seos.ini file is imported.

-h

Displays the help screen.

-o *selangFilename*

Writes selang commands to the specified file.

Note: When the -o flag is not set, the commands are written to STDOUT.

dmsmgr

The dmsmgr utility lets you:

- Create a DMS or a DMA on a computer where eTrust AC is installed.

  Note: You can also do this during installation.
- Remove a DMS or a DMA from an eTrust AC computer.
- Remove obsolete nodes from the DMS database.

  These are HNODE objects that represent eTrust AC nodes that have been unavailable for a specified amount of time.

-create Function—Create a DMS or a DMA

Use the dmsmgr -create function to create a Deployment Map Server (DMS) or a Deployment Map Server (DMA) on a computer where eTrust AC is installed.

Note: You can also create a DMS or a DMA during installation.

```
dmsmgr -create -dms <name> [-admin <users>] [-desktop <hosts>]
dmsmgr -create -dma [<hosts>] [-admin <usernames>] [-desktop <hosts>] \
[-subscriber <dms_names>]
```

-admin <*users*>

(Optional). Defines a comma-separated list of administrators for the created DMS or DMA.

Note: Whether specified or not, the user running the utility always gets administration rights for the created DMS or DMA.

-desktop <*hosts*>

(Optional). Defines a comma-separated list of computers that have TERMINAL access rights to the computer with the created DMS or DMA.

Note: Whether specified or not, the terminal running the utility always gets administration rights for the created DMS or DMA.

-dma [<*hosts*>]

Creates a DMA on the specified comma-separated list of <*hosts*>. If no host is specified, creates the DMA on the local computer.

Note: You can create a DMA from a remote computer if you have the appropriate sub-administration rights and the computer you are running the utility from has TERMINAL rights to the computer where you want to install the DMA.

-dms <*name*>

Creates a DMS with the <*name*> specified on the local host.

-subscriber <*dms_names*>

(Optional). Defines a comma-separated list of DMS nodes that the DMA created will send notifications to. Specify each DMS in the following format: <*DMS_name*>@<*hostname*>.

-remove Function—Remove a DMS or a DMA

Use the dmsmgr -remove function to remove a DMS or a DMA on a computer where eTrust AC is installed.

```
dmsmgr -remove {-dms <name> | -dma [<hosts>]}
```

-dms <*name*>

Removes the specified <*name*> DMS from the local host.

-dma [<*hosts*>]

Removes DMAs from the specified comma-separated list of <*hosts*>. If no host is specified, removes the DMA from the local computer.

Note: You can remove a DMA from a remote computer if you have the appropriate sub-administration rights and the computer you are running the utility from has TERMINAL rights to the computer where you want to remove the DMA.

-cleanup Function—Remove Obsolete Nodes

Use the dmsmgr -cleanup function to remove obsolete nodes from the DMS database. These are HNODE objects that represent eTrust AC nodes that have been unavailable for a specified amount of time.

Note: As a routine maintenance procedure, you should clean the DMS from these obsolete nodes.

```
dmsmgr -cleanup <number> -dms <name>
```

-cleanup <*number*>

Defines that the utility removes HNODE objects that represent eTrust AC nodes that have been unavailable for more than <*number*> of days.

-dms <*name*>

Defines the <*name*> of the DMS you want to remove the obsolete nodes from.

eacpg_gen

eacpg is also known as Policy Generator. This menu-driven utility provides an easy method to define a policy for eTrust AC applications. Policy Generator could effectively be used on a test system that has no eTrust AC rules in it. It aims to protect enterprise applications and/or operating systems and their confidential data by applying security best practices around those critical electronic assets.

Syntax

```
eacpg_gen -u <username> -g <groupname> -p <programname, full path> -o
<.........> (etc.)
```

Options

-u *user*

User for the process to run as

-g *group*

Group name that will own the process

-p *path*

Full path to the executable

-o *owner*

Owner of the policy

-w *wheel*

Sets as 'secadmins' group (recommended)

-m *machine*

Machine name

-a *apply policy*

Sets whether to apply the generated rules

-s *save policy to file*

Full path and the file name to save the policy rules

-# *step 1-2*

Should be set to 2

-x *toggle warn/fail mode*

Toggle between warn and fail mode

Note: Make sure that the secadmin and group secadmins exist in the database before you run.

Files

- eacpg_gen.exe
- eacpg_selang.exe
- eacpg_seaudit.exe
- eacpg_filter.exe
- eacpg_os.exe

The eacpg files are located under <eAC root dir>/bin/

Description

Application cells are created with a “default-deny” paradigm. These policies are similar to the concept of a UNIX chroot() jail. When such a policy is generated for an Internet facing application, the risk of host compromise via that application is greatly reduced.

An application cell is an ACL rule that blocks an application. For each application eacpg generates a number of application cells. The application cell enforces access to specific resources only. Any process protected with a cell policy cannot access resources it has not specifically been given access to in the policy. This keeps would-be attackers from writing to unauthorized areas of disk or executing unauthorized binaries.

Policy generation has several key steps:

- Initialization
- Application inspection
- Application testing
- Policy generation
- Applying the policy
- Testing the policy

User Perspective

Initialization:

1. Execute the policy generator:

   /eacpg_gen

2. Place the system into Warning Mode (type “y” at the prompt). For details, see the Implementation and Administrator guides.

3. Supply the policy generator with the full path to the executable, for example:

   ```
   /work/WebServers/apache_1.3.26/bin/htppd
   ```

4. Enter a user for the process to run as, or click enter to accept the default username (the default is recommended).

5. Enter a group name that will own the process, or click enter to accept the default (the default is recommended).

6. Verify that the information is correct (type “y” at the prompt).

Application Inspection:

7. Application inspection has begun. This is where the policy generator begins to collect data on the process you are creating a policy for.

   Verify the information on the screen and press enter.

Application Testing:

8. Start the application. For example:

   ```
   ./apachectl start
   ```

9. Stop the application. For example:

   ```
   ./apachectl stop
   ```

Note: At this point after you have started and stopped the application. It is best to start it again and allow for normal usage data to be collected. You can allow this inspection to take place for as long as you would like, the longer it runs the more data the policy generator can collect and the more accurate the resulting policy will be. When you feel you have collected enough data, continue to the next step.

Policy Generation:

10. Save the policy to a file (enter *filename*.txt and press return).

Applying the policy:

11. Apply the policy (type “y” at the prompt).

12. Put the system into “Fail” mode to begin policy enforcement (type “y” at the prompt).

Testing the Policy:

13. Test the policy. Below is a sample screen showing a policy test on a file named evil.html.

```
Linux:/srv/www/htdocs: #telnet localhost 80
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /evil.html
<!DOCTYPE HTML PUBLIC “-//IETF//DTD HTML 2.0//EN”>
<HTML><HEAD>
<TITLE>403 Forbidden</TITLE>
<HEAD><BODY>
<H1>Forbidden</H1>
You don't have permission to access /evil.html
on the server. <P>
<HR>
<ADDRESS>Apache/1.3.26 Server at linux.local Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
Linux:/srv/www/htdocs# []
```

Now that the policy is applied, the file evil.html is no longer available. This is because it was outside the scope of our normal usage profile.

exporttngdb

The exporttngdb program migrates the current Unicenter Security data into a local eTrust AC database or PMDB.

Two scripts, uni_migrate_master.sh and uni_migrate_node.sh automatically execute this program. Even though both scripts are run on the master machine, the uni_migrate_master.sh script calls it first to migrate the global Unicenter Security data into the Global PMDB. The uni_migrate_node.sh script calls it to migrate the local Unicenter Security data to the local SeOS DB.

Syntax

```
exporttngdb
```

Files

No special files are used.

issec

Displays the eTrust AC security daemons' status.

The issec utility displays the status of eTrust AC security daemons. If you do not specify any switches, the following information appears:

- The eTrust AC version and installation directory
- The status of the eTrust AC kernel extension
- The status of three major eTrust AC daemons: seosd, agent, and watchdog
- The status of eTrust AC daemons: serevu, selogrd, and selogrcd
- The status of the PMDB daemon and its name
- The status of the daemons that have been specified in the [daemons] section of seos.ini

Syntax

```
issec switch
```

Switches

-b

Displays the status and pid of major daemons (seosd, agent, and watchdog).

-k

Checks if eTrust AC kernel extension is loaded.

-h

Displays the help screen.

Files

No special files are used.

migopts

Translates current Unicenter Security environment settings into the global settings of either a local eTrust AC database or PMDB.

The migopts program is responsible for translating the current Unicenter Security environment into the global settings of either a local eTrust AC database or PMDB.

Two scripts, uni_migrate_master.sh and uni_migrate_node.sh automatically execute this program. Even though both scripts are run on the master machine, the uni_migrate_master.sh script calls it first to migrate the Unicenter Security environment into the Global PMDB. The uni_migrate_node.sh script calls it to migrate the local Unicenter Security environment to the local eTrust AC database.

The migopts program can, and should, be executed manually whenever a new PMDB is created.

Syntax

```
migopts
```

Files

No special files are used.

policydeploy

Use the policydeploy utility to store a policy on DMS nodes, or to deploy or undeploy a stored policy on a Policy Model hierarchy or an eTrust AC end-point.

```
policydeploy -store name -ds file1 -uds file2 [-dms list]
policydeploy -deploy name[#xx] -root dbs [-dms list]
policydeploy -undeploy name[#xx] -root dbs [-uds file2] [-dms list]
```

-deploy *name*[*#xx*]

Prompts you for whether you want to deploy the specified stored policy version on a defined eTrust AC Policy Model hierarchy.

To deploy the latest stored version of the policy, you can omit the policy version number.

-dms *list*

(Optional) A comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local eTrust AC database.

-ds *file1*

Defines the path name of the file containing the deployment rules. These are the commands necessary to construct the policy you want to deploy on each computer in a hierarchy.

Important! Policy deployment does not support commands that set user passwords. Do not include such commands in your deployment script file. UNIX (native) selang commands are supported but will not show in deviation reports.

-root *dbs*

Defines a comma-separated list of databases where the policy should be deployed or undeployed.

Note: If the root database is a Policy Model parent, the policy will be deployed or undeployed throughout its subscribing databases. If the root database is an eTrust AC end-point, the policy will be deployed or undeployed on the specified database only.

-store *name*

Prompts you for whether you want to store the policy *name* on the DMS nodes specified by the command or in the local eTrust AC database.

If no previous version of the policy *name* is stored on the DMS, version 1 of the policy is created (policy *name*#01). If a previous version of this policy exists, a new version of the policy is created (*name#last_version+1*).

Note: Policy *names* cannot include the # (hash) character which is reserved for denoting policy version numbers and is added automatically.

-uds *file2*

Defines the path name of the file containing the rules required to undeploy the policy. These are the commands necessary to undeploy the policy from a computer in the hierarchy.

When you undeploy a policy:

- If you do not specify a policy undeployment script, rules are taken from the undeployment rules the stored policy contains.
- If you specify a policy undeployment script, the DMS will still record the original rules that were provided when you stored the policy and not the new script supplied.

-undeploy *name*[*#xx*]

Prompts you for whether you want to undeploy the specified policy version *name#xx* from a defined eTrust AC Policy Model hierarchy.

To undeploy the latest stored version of the policy, you can omit the policy version number.

policyreport

Use the policyreport utility to create a host- or policy- centric report for a Policy Model hierarchy.

```
policyreport [-f] -name <name> -targetpath <path> -mode h -dms <name> \
-root <dbs> [-norec] [-dev] [-tree] [-hide p,d] -hn <hosts> -hstat <status> \
-sd <DD-MM-YYYY> -ed <DD--MM-YYYY> -st <HH:MM> -et <HH:MM>
```

```
policyreport [-f] -name <name> -targetpath <path> -mode p -dms <name> \
-root <dbs> [-norec] [-dev] [-hide p,d] -pn <policies> -pstat {<status>|None} \
-sd <DD-MM-YYYY> -ed <DD--MM-YYYY> -st <HH:MM> -et <HH:MM>
```

-dev

(Optional). Specifies to include deviation calculation results in the report.

Important! The deviation calculation does not check whether native rules are applied. It also ignores rules that remove objects (user or object attributes, user or resource authorization, or actual users or resources) from the database. For example, the calculation cannot verify whether the following rule is applied:
rr FILE /etc/passwd

-dms *<name>*

(Optional) A comma-separated list of DMS nodess from which information is collected. If you do not specify a DMS, the report information is collected from *DMS__@localhost*

Note: DMS nodess should be specified in the following format: *<DMS_name>@<hostname>*

-ed *date*

Defines the end date to use for filtering. Listings whose status changed after the specified date are not included. The format of *date* is *dd-mm-yyyy*.

-et *time*

Defines the end time to use for filtering. Listings whose status changed after the specified time are not included. The format of *time* is *hh*:*mm*, in 24-hour format. To delineate a time frame within a particular day, use this option in conjunction with -sd *date* or -ed *date* or both.

-f

(Optional). Specifies that the utility will run in "forced" mode, ignoring all warnings.

Use this option to add additional content to an existing report (or *refresh* the report with current information). Using the same name for the report you can then run the utility to update the report for areas that were updated since you last created the report or with options or filters you did not include when creating the original report.

-hide {p|d|p,d}

(Optional). Specifies report columns to hide:

p - Hides the Policies column.

d - Hides the Deviations column.

-hn [<*hosts*>]

(Optional). Defines a host name mask for filtering hosts included in the report. For example, **-hn prod*** specifies that only computers whose host name begins with **prod**, are included in the host report.

Note: Leaving the mask blank is the same as specifying the ***** wildcard. On UNIX, you must specify the mask in double quotation marks.

-hstat [<*stats*>]

(Optional). Defines a host status mask for filtering hosts included in the report. Possible host statuses are: Available, Unavailable, Sync (synchronizing), or Unknown.

Note: Leaving the mask blank is the same as specifying the ***** wildcard. On UNIX, you must specify the mask in double quotation marks.

-mode {h|p}

Defines whether the report generated is host- (h) or policy- (p) centric.

-name <*name*>

Defines the name of the report. Report files (XML and HTML) are stored in a structure under a directory carrying this name.

-norec

(Optional). Specifies to create a detailed host report only for the databases specified by the -root flag (does not to include their respective subscribers). Use this option when specifying the * wildcard for the -root flag to create or refresh subsets of detailed reports.

-pn [<*policies*>]

(Optional). Defines a policy name mask for filtering policies included in the report. For example, **-pn prod*** specifies that only policies whose name begins with **prod**, are included in the policy report. To include all versions of a policy add the #* suffix to the policy name.

Note: Leaving the mask blank is the same as specifying the ***** wildcard. On UNIX, you must specify the mask in double quotation marks.

-pstat [<*stats*>|None]

(Optional). Defines a policy status mask or a comma-separated list for filtering policies included in the report. If you specify *None*, the report includes only hosts with no policy status.

Possible policy statuses are: Deployed, Undeployed, Transferred, Failed (deployed with failures), Queued, TransferFailed, SigFailed (signature failed), UndeployFailed (undeployed with failures), or Unknown.

Note: Leaving the mask blank is the same as specifying the ***** wildcard. On UNIX, you must specify the mask in double quotation marks.

-root <*dbs*>

Defines a comma-separated list of databases for which you want information in the report.

Note: Report information is then gathered recursively for all subscriber databases of the root databases you specify (unless you specify the -norec flag or if the root database is an eTrust AC end-point).

-sd *date*

Defines the start date to use for filtering. Listings whose status changed prior to the specified date are not included. The format of *date* is *dd-mm-yyyy*.

-st *time*

Defines the start time to use for filtering. Listings whose status changed prior to the specified time are not included. The format of *time* is *hh*:*mm*, in 24-hour format. To delineate a time frame within a particular day, use this option in conjunction with -sd *date* or -ed *date* or both.

-targetpath <*path*>

Defines the full path for the location where the report is created.

Note: If you do not specify this flag, the report is generated to a default location:

<*eTrustACDir*>/data/reports/

-tree

(Optional). Specifies that the host report will display a graphical representation of the hierarchy.

Note: Filtered out parents still display in this type of report if any of their subscribers are included in the report.

seagent

The seagent daemon accepts requests from remote stations, and applies them to the local eTrust AC and UNIX databases, or to the PMDBs. It also checks that the Watchdog daemon seoswd is running, and if it is not, restarts it.

Seagent waits for connections on the seoslang and seoslang2 TCP services (whose default values are 8890 and 8891 respectively). When a connection request arrives, seagent forks a child process to handle the communication on the connection, and continues waiting for new connections.

The child processes of seagent get the requests from the client, and apply them to the local database.

The Agent is also responsible for the following:

- Updating the UNIX user file /etc/passwd, the system's shadow password file, and the UNIX group file /etc/group
- Alerting Policy Model daemons when an update is sent
- Alerting the parent Policy Models of both the local host and any Policy Model on the machine when a subscriber station (that has been down) is available for updating

eTrust AC uses only ports 8890 and 8891. We recommended that you do not change these ports.

The seagent agent uses the RPC mechanism and therefore the portmapper must be running on the local machine. For additional information on the portmapper, check your system documentation.

Syntax

```
seagent
```

Files

The following special files are used:

- seos.ini
- /etc/passwd
- /etc/group
- The system's password shadow file

See Also

The seoswd and sepmdd utilities in this chapter.

seaudit

Audit file viewer. The seaudit utility displays the records in the eTrust AC audit log file. The eTrust AC authorization daemon log submits records when an access to a resource requires auditing, as specified in the audit mode property of the resource or accessing user.

Using the full_year token, you can display the year in four digits or two digits. The default (“yes”) means four digits.

Switches are mandatory arguments that control the operation. You must specify at least one switch. Options are optional arguments that prevent unwanted data from being displayed.

You can use string matching in the switches and options. To see how eTrust AC performs string matching, see the appendix “String Matching.” Some UNIX shells automatically expand mask arguments; therefore, when invoking seaudit from such a shell, you should prevent the masks from being handled by the shell by typing a backslash (\) before an asterisk or question mark.

Note: The seaudit utility does not display a password even if you entered one as part of a chusr, editusr, or newusr command. The seaudit utility displays a series of asterisks (***) instead of the clear text password.

Syntax

```
seaudit switches [options]
```

Switches

-a

Lists all records except TCP records and those sent to the audit log by the tracing facility.

To list TCP records with actual port number (network id) to which connection was made, add the -c flag: seaudit -a -c.

-c

Shows “connected” INET records. These are records generated for session ID tracking, which list the port number of successful TCP connections.

For example, a user (user1) opens a telnet session from comp1 to comp2, both with eTrust AC installed. eTrust AC on comp2 can be configured (logconnected token)send acknowledgement to comp1 with the credentials of the user who logged in through the telnet session (this may be a user other than user1). When comp1 receives this acknowledgement, it creates a TCP-CONNECTED record (a session establishment record) that can then be displayed using the -c option.

Note: For more information about the logconnected token, see the seos.ini file in the *Reference Guide*.

-h

Displays a screen containing help and examples.

-i *host service*

Lists the INET audit records of the TCP requests received from the specified hosts for the specified services. Both *host* and *service* are masks that identify the set of hosts and services that are searched for.

To list TCP records with network id (port number) to which connection was made, add the -c flag: seaudit -i -c hostservice.

-l *user1, user2, ... terminal*

Shows the LOGIN records logged for the specified users on the specified terminal.

Both *user* and *terminal* are masks. This switch also controls the display of records created by serevu when it enables and disables users, and records created by the authorization daemon when an invalid password is entered.

-r *class resource user1, user2, ...*

Shows the general resources audit of the specified class on the specified resource for the specified users.

- *class* is a mask that identifies the class to which the accessed resource belongs.
- *resource* is a mask that identifies the names of the resources that were accessed.
- *user* is a mask of the name of the user who accessed the resource.

-s

Lists the startup and shutdown messages from the eTrust AC daemons.

-st | -stat

Displays descriptions of watchdog messages with more detail.

-t

Displays the table of log codes.

-tr

Displays trace records of all the users whose activities are being traced.

-trr *resource*

Displays the trace records of the specified resource.

-tru *uid|user1, uid|user2, ...*

Displays the trace records of the users with the specified numeric uids or user names.

-u *command class record user*

Displays database update audit records:

– *command* is a mask identifying the set of selang commands to search for.

– *class* is a mask identifying the classes to search for.

– *record* is a mask identifying the records to search for.

– *user* is a mask identifying the users who executed the commands.

-w

Lists the watchdog audit records.

Options

-detail

Displays detailed information about each record.

-delim *delimiter*

Uses *delimiter* as a delimiter before the first field and between the remaining fields. For example, seaudit -a -delim \"\,\" makes fields appear in quotation marks separated by a comma.

-delim2 *delimiter*

Same as the -delim option, except that the delimiter does not appear before the first field.

-ed *date*

Specifies the end date. Records logged after the specified date are not listed. The format of *date* is *dd-mm-yyyy*. You can use the string today to set the end date as today. You can also use the string “today” followed by - and a number. This specifies the end date as the specified number of days before today. For example, *today-3* means that the end date is three days ago.

-et *time*

Specifies the end time. Records logged after the specified time are not listed. The format of *time* is *hh*:*mm*, in 24-hour format. You can use the string *now* to set the end time as now. You can also use the string “now” followed by - and a number. This specifies the end time as the specified number of minutes before now. For example, *now-60* means that the end time is sixty minutes-one hour-ago. To delineate a time frame within a particular day, use this option in conjunction with -sd *date* or -ed *date* or both.

-f

Specfies that failures should **not** be displayed.

-fn *fileName*

Specifies the name of the audit log file to be searched.

-g

Specifies that successful (granted) accesses should not be displayed.

-gn

Specifies that successful (granted) accesses should not be displayed, except for notify records.

-logout

Specifies that logout records should not be displayed.

-millennium

Specifies that years should be displayed with four digits instead of two. For example, display 1997 instead of 97.

-n

Specifies that internet addresses should be displayed instead of host names in TCP/IP records.

-notify

Specifies that NOTIFY audit records should not be displayed.

-o *host*

Specifies that only records originating from the specified host should be displayed. This option is only applicable when browsing records from a consolidated audit file created by the selogrcd log-routing collection daemon.

-pwa

Specifies that password attempt records should **not** be displayed.

-sd *date*

Specifies the start date. Records logged prior to the specified date are not listed. The format of *date* is *dd-mm-yyyy*. You can use the string *today* to set the start date as today. You can also use the string “today” followed by - and a number. This specifies the start date as the specified number of days before today. For example, *today-3* means that the start date is three days ago.

-st *time*

Specifies the start time. Records logged prior to the specified time are not listed. The format of *time* is *hh*:*mm*, in 24-hour format. You can use the string *now* to set the start time as now. You can also use the string “now” followed by - and a number. This specifies the start time as the specified number of minutes before now. For example, *now-60* means that the start time is 60 minutes (one hour) ago. To delineate a time frame within a particular day, use this option in conjunction with -sd *date* or -ed *date* or both.

-v

Specifies that port numbers should be displayed instead of service names.

-warn

Specifies that warning records should **not** be displayed.

Output

Each record that seaudit displays contains data arranged in columns. The data in the first three columns has the same meaning for all types of records. From the fourth column to the end, the contents and meaning of the data that appears depends on the type of record.

Note: The following table describes the output for the most common type of record.

Column	Contents	Description
1	Date	The date the (attempted) access occurred.
2	Time	The time the (attempted) access occurred.

Column	Contents	Description
3	Alphabetic return code	The eTrust AC return code indicating what happened. The valid values and their meanings are: **A**-An attempt to log in failed because an invalid password was entered multiple times. **D**-eTrust AC denied access to a resource, did not permit a login, or did not permit an update to the database because the accessor did not have sufficient authorization. **E**-Serevu enabled a disabled user account. **F**-An attempt to update the database failed. **I**-Serevu disabled a user account. **M**-eTrust AC was started or shut down. **O**-A user logged out. **P**-eTrust AC permitted access to a resource or permitted a login. **S**-The database was successfully updated. **U**-A trusted program (setuid or setgid) was changed; therefore it is now untrusted. **W**-An accessor's authority was insufficient to access the specified resource; however, eTrust AC allowed the access because is set in the resource.
4	Type of Event / Class	The type of event being audited or the class on which the action was performed.
5	Accessor / Class	If the previous column contains a class name, this column contains the name of the accessor who executed the command. If the previous column contains UPDATE, then this column contains the class in which the action was performed.
6	Access type / Accessor	If the previous column contains the accessor name, this column contains the access type, if relevant. If the previous column contains the class name, this column contains the name of the accessor who executed the command.
7	Stage Code	A number which indicates at which stage eTrust AC decided what action to take and why. For a complete list of Stage Codes, run seaudit -t.

Column	Contents	Description
8	Audit Record Code	A number that represents the reason why eTrust AC wrote an audit record. For a complete list of Audit Record Codes, run seaudit -t.
9	Terminal / Resource	If column four contains LOGIN or LOGOUT, then this column contains the name of the terminal from which the login or logout was performed. Otherwise, this column contains the name of the resource being accessed or updated.
10	Terminal / Program	If column four contains UPDATE, then this column contains the name of the terminal from which the update was made. Otherwise, this column contains the name of the program that accessed the resource.
11	Command	If column four contains UPDATE, then this column contains a complete copy of the command entered by the accessor. If the command is a password update, the password itself is not displayed. If column four does not contain UPDATE and an action is being performed on the CLASS object via a remote terminal, then this column displays the IP address of remote terminal.

Trace Records

The following table describes the output for trace records, starting from the fourth column.

Column	Contents	Description
4	Type of Event	**TRACE**-Indicates that the record was created because all the user's or resource's activities are being traced.
5	UNIX UID	The UNIX UID of the process.
6	Effective UID	The effective UID of the process.
7	eTrust AC UID	The UID that eTrust AC associates with the process.
8	Stage Code	A number which indicates at which stage eTrust AC decided what action to take and why.
9	Resource / Action	The name of the resource being accessed or updated or the action being performed.

Column	Contents	Description
10	Trace file details	Additional details about the resource being accessed or the action being traced. The format of these fields is the same as the trace messages described in the appendix “Trace Messages”.

Sample Output

The output generated by seaudit looks similar to the following:

```
07 Dec 03 16:58 P PROGRAM      John       Exec     59  2 /usr/bin/enq
07 Dec 03 16:58 D TERMINAL     Smith      Read     55  3 xt3
07 Dec 03 20:21 P LOGIN        Bill               55  2 athena
07 Dec 03 21:04 P PROGRAM      Dennis     Exec     59  2 /usr/bin/su
07 Dec 03 21:04 P SURROGATE    Dennis             58  2 GROUP.system
07 Dec 03 21:04 P SURROGATE    Dennis             58  2 USER.root
07 Dec 03 21:41 U PROGRAM      seoswd              1  0 /tmp/testsuid
07 Dec 03 22:09 D HOST         telnet                   athena
07 Dec 03 22:10 O LOGOUT       Bill               49  2 athena
07 Dec 03 22:20 A LOGIN        Bill                8  2 athena
07 Dec 03 22:20 I LOGIN        Bill                0  5
07 Dec 03 22:25 E LOGIN        Bill                0  5
07 Dec 03 22:30 M START                                 seosd
07 Dec 03 22:32 M SHUTDOWN     Lynelle           452 seosd
```

Trace audit records look similar to the following:

```
17 Jan 04 22:27 P TRACE   244  244  244  0  FORK    : P=17010
U=244   G=201  Child=17013     pgm:/usr/bin/rlogin
17 Jan 04 22:28 P TRACE   244  244  244  0  SUID   >  P=17020
U=244  (R=244  E=0   S=0 ) to (R=244  E=244  S=244 ) () BYPASS
```

The remainder of this section discusses the output from the program.

```
07 Dec 03 16:58 P PROGRAM      John       Exec     59  2 /usr/bin/enq
```

On 7 December 2003 at 16:58, eTrust AC permitted (P) the user John to execute the *setuid* program */usr/bin/enq*. eTrust AC checked the Exec permissions for the record */usr/bin/enq* in the PROGRAM class. The eTrust AC authorization algorithm granted the operation based on code 59, Resource UACC check; that is, there exists in the database a record called */usr/bin/enq* in the PROGRAM class with the default access property set to allow execution by all users. The event was logged in the audit log file because of code 2, User audit mode; that is, the *audit* property of the user's record is set to include successful accesses.

```
07 Dec 03 16:58 D TERMINAL     Smith      Read     55  3 xt3
```

On 7 December 2003 at 16:58, eTrust AC denied (D) a login request from the user Smith for the terminal xt3. The requested access was READ, the normal access authority for records of the TERMINAL class. Access to the terminal was denied because of code 55, Resource ACL check for the user; that is, the user Smith is defined in the ACL of the TERMINAL class record xt3 with access NONE. The event was logged in the audit log file because of code 3, Resource audit mode; that is, the audit property of the xt3 record is set to include failed accesses.

```
07 Dec 03 20:21 P LOGIN        Bill               55  2 athena
```

The user Bill was permitted (P) to log in from the terminal athena. The login was allowed because of code 55, Resource ACL check for user; that is, the user Bill has READ authority in the ACL of the TERMINAL class record athena. The event was logged because of the user's audit property.

```
07 Dec 03 21:04 P PROGRAM      Dennis     Exec     59  2 /usr/bin/su
07 Dec 03 21:04 P SURROGATE    Dennis              58  2 GROUP.system
07 Dec 03 21:04 P SURROGATE    Dennis              58  2 USER.root
```

The user Dennis was permitted (P) to execute the */usr/bin/su* program in order to substitute to the user root. The program later requested *setgid* to the group 0 (in this case, the group *"system"*) and *setuid* to the user root. Both requests were granted based on code 58, that is, the user Dennis is a member of a group that appears in the ACLs of these resources. All three events were logged because of the resources' audit properties.

```
07 Dec 03 21:41 U PROGRAM      seoswd              1  0 /tmp/testsuid
```

The Watchdog marked the program */tmp/testsuid* as untrusted (U). The program was marked untrusted because of code 1, File Stat information changed. Code 1 is a global catch for all modifications to the file status information, including modifications to the time, size, owner-user and group-and mode entries. The program testsuid was only created for test purposes and the information was changed using the touch utility. The digit 0 is placed in a column that is only used if the reason is 1. In this case, this column displays the return value of the *errno* variable. To find out the meaning for the error, see */usr/include/errno.h* or */usr/include/sys/errno.h* file on the local station.

```
07 Dec 03 22:09 D HOST         telnet                  athena
```

The host athena was denied access to the telnet service.

```
07 Dec 03 22:10 O LOGOUT       Bill               49  2 athena
```

The user Bill logged out from the system. eTrust AC knows about most process terminations in the system. When all processes associated with Bill's credentials have terminated, Bill is considered to be logged out. The LOGOUT class entry and the O in the result column identify logout records. The code 49 indicates a LOGOUT audit record. The code 2 indicates the event was logged due to the user's audit mode. eTrust AC reports logouts only if logins are also reported for the user.

```
07 Dec 03 22:20 A LOGIN        Bill                8  2 athena
```

The user Bill tried to access the system. The code 8 indicates that the login procedure failed because Bill failed to provide the correct password. The serevu daemon detected and logged the event.

```
07 Dec 03 22:20 I LOGIN        Bill                0  5
```

This audit record is submitted by the serevu daemon when it disables a user's login because of too many password attempts. The digit 5 identifies the entry as being submitted by serevu.

```
07 Dec 03 22:25 E LOGIN        Bill                0  5
```

This audit record was submitted by the serevu daemon when it re-enabled the login of user Bill that was previously revoked by the daemon.

```
07 Dec 03 22:30 M START                                seosd
07 Dec 03 22:32 M SHUTDOWN     John                 452 seosd
```

These audit records indicate the startup and shutdown of the seosd daemon. Seosd started at 22:30 and John brought seosd down at 22:32. John was allowed to take seosd down because he has the ADMIN attribute-reason code 452. Reason code M indicates startup or shutdown of the seosd daemon.

The following examples are trace on user records.

```
17 Jan 04 22:27 P TRACE   244  244  244  0  FORK    : P=17010
U=244   G=201  Child=17013     pgm:/usr/bin/rlogin
```

eTrust AC intercepted a fork request made by process 17010 associated with uid 244 and group id 201. The child process id is 17013. The program running in the parent process (and initially also in the child process) is */usr/bin/rlogin*. The eTrust AC stage code is 0.

```
17 Jan 04 22:28 P TRACE   244  244  244  0  SUID   >  P=17020
U=244  (R=244  E=0   S=0 ) to (R=244  E=244  S=244 ) () BYPASS
```

eTrust AC granted the setuid request without checking any SURROGATE access rule. In the message text, 17020 is the issuing process id; 244 is the userid associated with this process; *r*, *e*, and *s* are the real, effective, and saved uids of process 17020; 244 is the target effective, real, and saved uids with which the setuid request was issued. The checks were bypassed because the current real uid is the same as the target uid and therefore the setgid request does not change the security scope of the user.

The seos.ini File

The seaudit utility uses the following tokens in the seos.ini file:

Section	Token
logmgr	audit_back audit_group audit_log audit_size error_back error_group error_log error_size BackUp_Date
message	filename

For more information about these tokens, see the *Administrator Guide*.

Other Files

The seaudit utility uses the following additional special files:

- The eTrust AC audit log file specified in the seos.ini file, usually *eTrustACDir*/seos.audit (where *eTrustACDir* is the directory where you installed eTrust AC), unless an audit log file is explicitly specified on the command line. The audit log file cannot be defined in the database and only eTrust AC can write to the file. Users can only have READ access to the file.
- The eTrust AC messages file, *eTrustACDir*/data/seos.msg
- /etc/passwd
- /etc/group
- /etc/hosts
- /etc/services

Examples

- The following command lists all audit records since 3 January 2004:

```
seaudit -a -sd 04-Jan-2004
```

- The following command lists the failed logins of the user root from any terminal on 3 January 2004:

```
seaudit -sd 04-Jan-2004 -ed 04-Jan-2004 -l root \* -g
```

- The following command lists all accesses of user John to every resource of class DBFIELD*:*

```
seaudit -r DBFIELD \* John
```

- The following command lists all audit records that were logged between 17:00 (yesterday) and 08:00 (today):

  ```
  seaudit -a -st 17:00 -et 08:00
  ```

- The following command lists all audit records that were logged between 08:00 and 17:00:

  ```
  seaudit -a -st 08:00 -et 17:00
  ```

- The following command lists all warning records for logins and resource accesses for a single user:

  ```
  seaudit -login \* \* -resource \* \* \* -grant -failure -logout -pwa
  ```

- The following command lists all login records for two users:

  ```
  seaudit -login “user1, user2”
  ```

- The following command lists all audit records from yesterday:

  ```
  seaudit -a -sd today-1 -ed today-1
  ```

- The following command lists all audit records that trace the activity of a single user with UID 244 attempting to access files:

  ```
  seaudit -tru 244 -trr FILE
  ```

- The following command lists all audit records that trace the activity of two users:

  ```
  seaudit -tru “user1, 244”
  ```

See Also

- The chapter “The Audit Browser: seauditx” in the *User Guide.*
- The selang, selogrd, selogrcd, seosd, seoswd, and serevu utilities in this chapter.

seauxd

The eTrust AC auxiliary daemon.

The seauxd daemon is an internal feature that eliminates the deadlock that occurs when eTrust AC runs on hosts with NIS or DNS servers or other name-caching daemons. This daemon eliminates the need for the lookaside database, which was previously used to solve deadlock problems.

To activate seauxd, set one or both of the name_resolving and TNG_calendars tokens to yes.

The seauxd daemon is started by the seosd daemon according to initialization settings. The seauxd daemon performs the following functions:

- Analysis requests from seosd
- Name resolution-To activate this function set the token name_resolving in the [seauxd] section of the seos.ini file to yes. When this function is activated, seauxd resolves names of users, groups, host, and services. It eliminates the deadlocks that occur when eTrust AC runs on hosts with NIS, DNS, or other name caching daemons. seauxd helps you avoid using a lookaside database.
- Unicenter TNG calendar retrieval-To activate this function set TNG_calendars token in the [seauxd] section of the seos.ini file to yes. When this function is activated, seosd sends the list of Unicenter TNG calendars to seauxd. The seauxd daemon calls Unicenter TNG, updates the status of each calendar, and returns an updated list of calendars to seosd.

The [seauxd] section of seos.ini contains of number tokens to allow you to fine tune the seauxd daemon.

Files

The following special files are used:

- seos.ini
- seauxd.log

The seos.ini File

The seauxd utility uses the following tokens in the *seauxd* section, in the seos.ini file:

- allow_any_platf
- name_resolving
- TNG_calendars

See Also

The seosd utility in this chapter.

sebuildla

The sebuildla utility creates a lookaside database for use by the eTrust AC daemon, seosd. The seosd daemon uses the database to translate UNIX UIDs to user names, GIDs to group names, host IP addresses to host names, and service ports to port names. The database contains only the number to name translation. sebuildla also lets you add information from the LDAP Directory Information Tree (DIT) to the user lookaside database.

Before using sebuildla to build the lookaside databases, specify the full path of the lookaside database, in the lookaside_path token in the seos.ini file.

The first time you build the lookaside database, use the following command:

```
sebuildla -a
```

This creates *all* of its components. Single files of the database can be updated later by using the relevant switches.

If you installed eTrust AC on a NIS, NIS+, or DNS server, you should place calls to the sebuildla utility in the related makefiles.

Note: By default, the lookaside database files (groupdb.la, hostdb.la, servdb.la, and userdb.la) are protected against all user access other than access with the sebuildla program.

The sebuildla utility scans the resolution mechanisms in the system, such as /etc files and NIS, to build the lookaside databases.

- sebuildla reads /etc/resolv.conf to get the domain name used.
- sebuildla uses the system resolution option to create the lookasaide database. (This is usually the net caching daemon.)
- eTrust AC uses the /etc/nsswitch.conf file (for the net caching daemon or any other system resolution option) to decide where to retrieve data from.

For example, if the /etc/nsswitch.conf file contains the following line for hosts, information is retrieved from the local machine's files first (/etc/hosts); it then retrieves information from the DNS and then the NIS:

```
hosts:      files dns nis
```

If the file contains the following line instead, information is retrieved only from your local machine's files. The look aside database will contain only the hosts that are in /etc/hosts:

```
hosts:      files
```

Note: If a host has a fully qualified name, sebuildla uses it.

Variations in machine configuration may cause instances where sebuildla does not list all the names of a local environment. In this case, you can use sebuildla to load all the required entries from a list file. To do this, create a list file with each object name on a separate line. The utility reads this list file and ensures that all the objects in the list file are added to the relevant lookaside database if necessary. sebuildla ignores duplicate objects.

The following table lists the files sebuildla uses to build each lookaside database.

Objects in	Are added to the
eTrustACDir/ladb/userlist	users lookaside database
eTrustACDir/ladb/grouplist	groups lookaside database
eTrustACDir/ladb/hostlist	hosts lookaside database
eTrustACDir/ladb/servlist	services lookaside database

Syntax

```
sebuildla switch option
```

Switches

-a

Creates *all* the lookaside database files.

-e

Creates a hosts lookaside database file excluding the DNS.

-g

Creates a groups lookaside database file.

-h

Creates a hosts lookaside database file with the DNS.

-n

Collects information from an LDAP Directory Information Tree (DIT) and appends it to the users lookaside database it creates from the primary user data source (-u switch). You can only use this switch in conjunction with the -u switch or the -a so it is most useful when the LDAP DIT provides additional user data and is not used as the system's naming service.

Before you use this switch, follow these steps:

a. Set the following seos.ini file tokens for eTrust AC to find the LDAP service: ldap_base, ldap_hostname, and ldap_userdn.

b. Run the seldapcred utility to store the encrypted LDAP password.

c. (Optional) Set the ldap_port and ldap_timeout tokens for your environment.

   The time it takes to retrieve information from the LDAP service depends on how fast the LDAP service is, and how much user data is stored in the DIT. You can adjust the ldap_timeout token in the [seos] section of the seos.ini file to account for these aspects.

d. (Optional) If you are using a non-standard schema, set the ldap_uid_attr, ldap_uidNumber_attr, and ldap_user_class tokens.

Note: For more information about seos.ini tokens, see the *Reference Guide*.

-s

Creates a services lookaside database file.

-u

Creates a users lookaside database file.

Note: You can specify the -n switch in conjunction with the -u switch to add user data that is collected from an LDAP service.

-G

Lists the contents of the groups lookaside database files.

-H

Lists the contents of the hosts lookaside database files.

-S

Lists the contents of the services lookaside database files.

-U

Lists the contents of the users lookaside database files.

Options

-l

Loads the lookaside database using only the list file. -l excludes the resolution mechanism of the system.

-f

Fast loads the lookaside database hosts (only) using the -h switch. (See the following note.)

File Format

In the format of the files in the *eTrustACDir*/ladb directory,

- sebuildla ignores empty lines or lines that begin with an exclamation point (!), number sign (#), or a semicolon (;).
- Other lines represent entries that sebuildla must add to the appropriate lookaside database, if the entry can be resolved.
- The user, group, host, or service name must start in the first position of the line.

Note: You can use dbmgr -dump -r to create the list files. For example, to create a list of the hosts defined in class HOST in the local database, enter:

```
# dbmgr -dump -r l HOST > /opt/CA/eTrustAccessControl/ladb/hostlist
```

The *-l* switch makes a single request from DNS for a list of all hosts in the default domain, instead of querying the DNS server for the FQDN of each host entry as it is obtained. The fast load option is effectual only if DNS is installed. Only host names in the default domain are made fully qualified. Fully qualified names are left as such. Host names scanned from the system mechanism that are not fully qualified, and are not found in the default domain, are left unqualified. Host names loaded from the hostlist file that are not fully qualified are discarded.

The seos.ini File

Depending on the switches you specify, the sebuildla utility uses the following tokens in the seos.ini file:

Section	Token
seosd	lookaside_path
seosd	lookaside_allowdupuid
seos	ldap_base
seos	ldap_hostname
seos	ldap_method
seos	ldap_port

Section	Token
seos	ldap_timeout
seos	ldap_uid_attr
seos	ldap_uidNumber_attr
seos	ldap_user_class
seos	ldap_userdn

Other Files

The sebuildla utility uses the following additional files:

- /etc/group
- /etc/hosts
- /etc/passwd
- /etc/resolv.conf
- /etc/nsswitch.conf
- /etc/services

The sebuildla utility also uses the following files located in the directory specified in the token lookaside_path:

- groupdb.la
- grouplist
- hostdb.la
- hostlist
- servdb.la
- servlist
- userdb.la
- userlist

See Also

The seosd utility in this chapter.

sechkey

Changes the encryption key for various eTrust AC programs and also allows you to set an encryption key for the audit log routing daemon, selogrd.

The sechkey utility handles several tasks:

- Changing the eTrust AC encryption key
- Changing the selogrd daemon encryption key
- Clearing the selogrd daemon encryption key

With the sechkey utility, you can set an encryption key to protect your management communications with the eTrust AC daemons (seosd, seagent, sepmdd, and selogrd) and the Policy Model daemon.

Before changing the libcrypt key, the sechkey utility decrypts any record that was encrypted in the updated Policy Model file, and encrypts all records again after changing the key. This operation is performed for any Policy Model installed on the machine. We recommend performing auto-truncate on all encrypted update files before changing the key.

In addition, whenever you use an eTrust AC API to create a program that communicates with an eTrust AC daemon, your new program's communications are encrypted with the same key.

Important! When changing the encryption key, you must set all machines at that site to the same key, or communication problems result.

Syntax

```
sechkey {oldkey | -d} {newkey | -d | -n} [-nopmd | -r hostname]

sechkey -k newkey

sechkey -c
```

Parameters

***oldkey* | -d**

Specifies the (current) encryption key that you want to change.

To specify the original key, use -d instead of *oldkey*.

***newkey* | -d | -n**

Specifies the new encryption key, to which you want to change. Encryption keys are 55 characters long. A longer *newkey* is automatically truncated, a shorter one is automatically padded.

To change to the original key, use -d instead of *newkey*.

To list the programs that are using the current key, rather than change to a different key, use -n.

-nopmd

Changes the old key to *newkey*, without updating the Policy Model update file with the new key.

-r *hostname*

The name of the remote machine whose encryption key you want to change.

To use this command, eTrust AC must be running on both the local and remote machines. This parameter does not actually change the key; rather, it saves information so that the next time you start eTrust AC on the remote machine (using seload -c), the key is changed. For more information, see the seload command in this chapter.

-k

Specifies the selogrd encryption key that you want to change to. The encryption key is saved in a new file or updated in the old one.

-c

Clears the selogrd encryption key. The default key is saved in the key file.

Notes:

- To change or set the eTrust AC encryption key on the local machine, you must first shut down eTrust AC. To change or set the key on a remote machine, eTrust AC must be running on both the local and remote machines.
- We recommend that you change the key from the default after you finish installing or upgrading eTrust AC on all machines at your site to prevent unauthorized users from accessing the system.
- To prevent communication failure between hosts, be sure to set the same encryption key on all machines.
- If you use the Policy Manager (stand-alone version), the defaults for the UNIX server and the Windows server are the same, so communication is enabled and encrypted.
- Some shells do not support special characters. In such cases, add a slash (\) before every special character or refer to your UNIX documentation.

Files

The seos.ini file specifies the encryption method and the encryption files used for auditing. The UseEncryption token sets the encryption method. The RefuseUnencrypted token specifies whether selogrcd will accept unencrypted audit. The CipherName token points to the encryption library. The KeyFile token points to the file where the key is saved.

Note: The saved key itself is encrypted with the default encryption method.

Example: Check whether the system is still using the default encryption key

Enter the following command:

```
# /usr/seos/bin/sechkey -d -n
```

The following result is displayed:

```
eTrustAC sechkey v8.00 (132) - internal key changer
Copyright 2004 Computer Associates International, Inc.
Searching '/usr/seos/lib/libcrypt' for key...found
Searching '/usr/seos/bin/seload' for key...found
```

seclassadm

Administers eTrust AC classes.

Note: This utility is for third-party developers and programmers.

The seclassadm utility adds new classes to the database. Invoke it from the directory in which the database resides, while the eTrust AC daemons are **not** running.

Specify one command only. The switches are optional, and you can specify more than one switch.

Note: Running seclassadm creates a file in the seosdb directory with the new class information. When you create a new database with dbmgr -c, user-defined classes are created in the new database if the CreateNewClasses token in the seos.ini file is set to yes (the default).

Important! If you use advanced policy-based management and reporting, when adding or deleting classes, or adding or deleting class properties in the eTrust AC database, you must also do the same in the eTrust AC initial database (init_ac_db). This database is used for policy deviation calculations; it is located at the path specified by the init_ac_db token of the seos.ini file (by default, *<eAC_InstallDir>*/data/devcalc/init_ac_db).

Syntax

```
seclassadm command [switches]
```

Commands

-add *class-name*

Adds a new resource class to an existing database. *Class-name* is the name of the new class. eTrust AC reserves class names that are in uppercase characters. When adding a class, use at least one lowercase character. Class names can be up to 79 characters long.

After creating a new class, you must enable the class by using the setoptions command under selang. For more information, see the descriptions of selang and setoptions in the *Reference Guide.*

-del *class-name*

Deletes the specified resource class from the database.

-upd *class-name*

Updates the specified resource class in the database. The syntax for this command is:

seclassadm -upd <ClassName> {-|+}c

where the 'c' switch indicates a change in the class case functionality, and the {-|+} indicates whether the class does not/does support case-sensitive objects, respectively. If other switches are entered, they are ignored.

Switches

-a *types*

Sets the types of access that are allowed for the class. *Types* is a string that represents the allowed accesses. A character represents each access type. Specify the characters of the types you want the class to support. The types can be specified in any order. The string must not contain any blank or other non-alphabetical characters. The following access types are supported. (The character that represents the access type precedes the access type.)

- C-control
- D-delete
- E-create
- F-filescan
- M-chmod
- O-chown
- R-read
- S-security
- T-utime
- U-update
- V-rename
- W-write
- X-exec

-d *access*

Sets the default access of the class-the access that is assigned to a user when you execute the authorize command without specifying an access authority. This is the implicit access used by the authorize command; do not confuse it with the default access assigned to a resource. The possible accesses are listed in the -a *types* switch.

Access represents the implicit access. Specify the character or characters that represent the access. If you specify more than one character, the order of the characters does not matter. The string of characters cannot contain any blanks or other non-alphabetic characters.

-f

Forces eTrust AC to accept a new class name, even if the name contains all upper case letters.

-g

Makes the new class a resource that groups members of an existing class. The relationship between the existing class and the new group class is like the relationship between the TERMINAL and GTERMINAL classes in the database. A resource that groups members of an existing class must begin with the upper case letter G. That is, it has the same name as the existing class, but begins with the prefix 'G'.

-o

Creates _default record for the new class and set its default access.

-p

Sets the position (full path) of the local eTrust database.

-r

Specifies a resource description object (for eTrust Web AC classes).

-t

Specifies a Unicenter TNG class.

Files

The seclassadm utility uses the database files if these files are located in the current directory.

The seos.ini file is not used.

Examples

- To add a resource class by the name “dbfield,” enter:

  ```
  seclassadm -add dbfield
  ```

- To add a resource class by the name *report* with only READ access, enter:

  ```
  seclassadm -add report -d R -a R
  ```

- To add a resource class by the name *batch_jobs* with READ, WRITE, and MODIFY permissions and READ access as the default when not specified, enter:

  ```
  seclassadm -add batch_jobs -d R -a RWM
  ```

- To add a new class whose objects are groups of resources in the class DEPTA, with access execute and implicit access execute, enter:

  ```
  seclassadm -add DEPTA -d X -a X -g -f
  ```

See Also

- The dbmgr and sepropadm utilities in this chapter.
- The setoptions command in the *Reference Guide*.

secompas

Compares passwords in the eTrust AC database with the passwords in the UNIX password file.

The secompas utility compares the user passwords in the eTrust AC database with the passwords in the UNIX password file. For each user, one line appears in the standard output that contains the user name and a message indicating whether or not the passwords match or that the user is not defined in eTrust AC or UNIX. After comparing all the users, secompas displays the total number of users it compared and the number of users whose passwords do not match.

The utility only adds to the counter of unmatched passwords when the password exists in both environments and is not the same. If a user is not defined in an environment, or the password is missing from an environment, secompas does not add to the counter of unmatched passwords.

Syntax

```
secompas options
```

Options

-db

Prevents the display of the 'Not in database' message.

-h

Displays help.

-ok

Prevents the display of the 'OK' message.

-ux

Prevents the display of the 'Undefined in UNIX' message.

Authorization

Only users with the ADMIN attribute can use this utility.

Output

The utility produces several types of messages that are sent to the standard output. The messages and their meanings are as follows:

OK

The eTrust AC password matches the UNIX password.

***** PASSWORDS DO NOT MATCH *****

The eTrust AC password does not match the UNIX password of the user.

No password in Access Control database

Either the user is not defined in the database or the user is defined in the database but does not have a password in it.

Undefined in UNIX

The user is defined in the eTrust AC database but not in UNIX.

No password in UNIX password file

The user is defined in UNIX but does not have a password.

***** NO MATCH - UNIX }DISABLED *****

The user account was disabled in the UNIX environment; secompas identifies a disabled user account by the asterisk (*) in front of the password in the /etc/passwd file.

Example

For example, the command *secompas* may produce the following output:

```
Checking root           : No password in Access Control database.
Checking tst_001        : Undefined in UNIX.
Checking tst_002        : No password in UNIX password file
Checking tst_003        : *** PASSWORDS DO NOT MATCH. ***
Checking tst_004        : *** NO MATCH - UNIX DISABLED ***
Checking tst_005        : OK

Total of 6 users found in database.
2 unmatched password(s) found. (1 UNIX DISABLED).
```

Files

The */etc/passwd*, the shadow password files, and NIS/NIS+ password maps are used. The seos.ini file is not used.

secons

The eTrust AC security console.

The secons command line utility provides a control console to the eTrust AC daemons. secons performs various operations, such as:

- Control tracing of the eTrust AC authorization daemon (seosd)
- Enable and disable login
- Get login status
- Display runtime statistics
- Enable and disable cache of file authorization.
- Display files cache table.
- Shut down the eTrust AC server daemons (according to a list of *hosts* / *ghosts* given in the command line)
- Shut down the eTrust AC audit daemons

Syntax

```
secons options
```

Trace Control Options

-t+

Enables tracing. Causes the eTrust AC daemon seosd to dump messages that specify its operations and actions to the trace file.

-t-

Disables tracing. Stops the eTrust AC daemon seosd from dumping messages to the trace file.

-tt

Toggles the tracing status between enabled and disabled.

-ts

Displays the current tracing status.

-tc

Clears the trace file, that is, removes all records from the trace file. This option can be used whether or not seosd is running.

-tv -file *fileName*

Browses the specified file instead of *eTrustACDir*/log/seosd.trace. This option can be used whether or not seosd is running.

Using the full_year token, you can display the year in four digits or two digits. The default is *yes* meaning four digits.

-tv *sizeInKB*

Displays an online trace view. Starts a browse session on the trace file and operates in a manner similar to the tail -f system utility.
Optionally, you can give a size so that only the last *sizeInKB* is shown. The default value is 2 KB. Specifying 0 shows the entire trace file. This option can be used whether or not seosd is running. To browse a trace file other than the default one, use the -file *fileName* option, naming the file to be viewed.
To stop this operation, press Ctrl+c.

Using the full_year token, you can display the year in four digits or two digits. The default is *yes* meaning four digits.

Login Control Options

-d+

Enables concurrent logins for the user executing the command.

-d-

Disables concurrent logins for the user executing the command. Using this command disables any concurrent logins of the user name to the local computer. It is possible to have this command in the .login or .cshrc file of a user to disable concurrent logins.

-ds

Displays the concurrent logins setting for the user executing the command.

-l+

Enables concurrent logins system wide. By default, eTrust AC enables login, but in cases where the system is shut down for maintenance, you can disable login for a specific period. This option enables login.

-l-

Disables concurrent logins system wide.

-ls

Displays system-wide login status.

-u+ *userName*

Enables concurrent logins for the specified user.

-u- *userName*

Disables concurrent logins for the specified user.

-us *userName*

Displays the concurrent logins setting for the specified user.

File Cache Options

-C+

Enables caching of file authorization. Causes the eTrust AC daemon (seosd) to cache the last file authorization results in a table.

-C-

Disables caching. Stops the eTrust AC daemon (seosd) from caching authorization results in a table.

-CA value

Sets the maximum number of authorization records in a table. The default value is 80. Allowed values range from 1-800.

-CC interval

Sets the cache clean interval in minutes. The default is 60 minutes. Allowed values include any nonzero number.

-CD

Displays the cache table to the standard output. See Cache Settings Standard Output.

-CF value

Sets the maximum number of file records in a table. The default value is 20. Allowed values range from 1-200.

-CI init_value

Sets the initial priority value for a new record in the cache table. The default is 10.

-CP interval

Sets the cache priority computing interval. The default is one record. Allowed values range from 1-10.

-CU value

Sets the maximum number of user records in a table. The default value is 50. Allowed values range from 1-500.

Miscellaneous Options

-i

Displays runtime statistics as formatted text. For a description of the displayed information, see Examples in this section.

-m

Sends a message to the console, adds text to the trace file that was produced by the eTrust AC authorization daemon.

-rl

Updates tokens from the seos.ini file, inside seosd, without shutting down the daemon.

-s [*stationNames*]

Shuts the eTrust AC daemons down on the local or remote hosts. You can specify a group of hosts by entering the name of a gterminal record. If no terminal is specified, the daemon is shut down on the local terminal only. If you use this option from a remote terminal, the utility requests password verification. You also need admin privileges on both remote terminal and local terminal, as well as write permission to the local terminal on the remote host database.

-S [*daemon_name*]

If the *daemon_name* is not defined, the command terminates the eTrust AC daemons and attempts to terminate active daemons selogrd, selogrcd and serevu. If the selogrd token, selogrcd token, or serevu token in the [daemons] section of seos.ini file is set to "yes", it sends the termination request to the running eTrust AC main daemon or sends the termination signal to the specified daemon if eTrust AC is down.

If *daemon_name* is specified as selogrd, selogrcd, or serevu, then the command does not terminate the eTrust AC daemons. If the appropriate token in the [daemons] section of seos.ini file is set to “yes”, it sends the termination request to the running eTrust AC main daemon or it sends the termination signal to that daemon if eTrust AC is down.

Note: When the main eTrust AC daemon is down, you can use the command *secon -s* for user root.

-sc

Displays processes that are still executing eTrust AC code.

You cannot unload eTrust AC if an application, which is loaded on top of eTrust AC, has an open system call (syscall) that is hooked by eTrust AC. Once you know what processes are still executing eTrust AC code, you can shut down these processes and unload the eTrust AC kernel module. You can then use UNIX exits to automatically shut down these processes before unloading the kernel and then then restart them after the kernel unloaded.

Note: For more information about eTrust AC kernel loader UNIX exits, see the *Administrator Guide*.

The output displays as a two-column table with the system call number in the first column, and the process identifier in the second column.

By default, eTrust AC monitors system calls. You must set the syscall_monitor token in the seos.ini file to 0 (disabled) if you do *not* want eTrust AC to monitor system calls.

-sk

Shuts down all eTrust AC daemons and prepares the eTrust AC kernel extension to be unloaded.

The seos.ini File

The secons utility uses the following tokens in the seos.ini file:

Section	Token
seosd	trace_file trace_file_type trace_to FileCache_files FileCache_users FileCache_auths FileCache_CleanInt FileCache_PriorInt FileCache_InitPrio

For more information, see the *Administrator Guide*.

Other Files

No other special files are used.

Notes:

- The secons command line utility is available to both system administrators and regular users. The options displayed and allowed for users who do not have the ADMIN attribute are a subset of the total options available: -d+, -d-, -ds, and -m.
- Only users marked as ADMIN or OPERATOR can shut down the eTrust AC daemons.
- When the eTrust AC daemon is not running, you can still use the options -tv and -tc if you use them in super-user (root) mode.

Examples

- To shut down the eTrust AC daemon, enter:

  ```
  secons -s
  ```

- To shut down the eTrust AC daemon on remote hosts HOST1 and HOST2, enter:

  ```
  secons -s HOST1 HOST2
  ```

- To place the string “Start Event” in the eTrust AC trace file, enter:

  ```
  secons -m 'Start Event'
  ```

- To display the runtime statistics, enter:

  ```
  secons -i
  ```

  The screen output generated from secons -I resembles the following:

  ```
  /opt/CA/eTrustAccessControl/bin/secons -I
  secons  Access Control Console Utility
  Copyright 2004 Computer Associates International, Inc.

  Runtime Statistics:
  -------------------
  INet Statistics:
              Requests Denied   : 0
              Requests Granted  : 17
              Errors found      : 0

  Queues Size:
              Audit Log: 0
              Error Log: 0
  Cached Tables Info:
              ACEE Handles      :      11
              Protected clients :       0
              Trusted Programs  :      77
              Untrusted Programs:       0
  Access Control Database info: (record Count & First Free Id)
              classes     :     18 ( CID     0x0012 )
              Properties  :    223 ( PID     0x00df )
              Objects     :    152 ( OID 0x000000a8 )
              PropVals    :    972 ( N/A )
  ```

  Following is a detailed discussion of the sample output.

  ```
      INet Statistics:
              Requests Denied   : 0
              Requests Granted  : 17
              Errors found      : 0
  ```

  Statistics on the network access authorization performed by eTrust AC. These lines summarize the number of denials, grants, and errors during the authorization of network requests.

  ```
      Queues Size:
              Audit Log: 0
              Error Log: 0
  ```

  Since eTrust AC creates logging with file locking, it is possible that certain events are held in memory and written to log files after a while. If these values exceed 10, then an error could be interfering with the eTrust AC logging facility.

```
Cached Tables Info:
        ACEE Handles      :      11
        Protected clients :       0
        Trusted Programs  :      77
        Untrusted Programs:       0
```

- *ACEE* (Accessor Element Entry) is a table containing logged-in processes.
- *Protected clients* lists the number of cached clients. Usually, this value is 0.
- *Trusted Programs* lists the number of entries in class PROGRAM that are cached in memory. Normally, all programs should be cached as trusted.
- *Untrusted Programs* displays the number of programs that were found to be untrusted.

```
Access Control Database info: (record Count & First Free Id)
        classes     :     18 ( CID     0x0012 )
        Properties  :    223 ( PID     0x00df )
        Objects     :    152 ( OID 0x000000a8 )
        PropVals    :    972 ( N/A )
```

General information regarding the size of the database and the number of records in each part of the database.

- To change settings of the cache, enter one or more options:

```
secons -CF 60 -CU 60 -CA 60
secons -CF 60 -CC 20 -CP 2 -CI 2
secons -CI 2
```

- To display the cache table, enter:

  ```
  secons -CD
  ```

  The output generated on the screen resembles the following:

```
#secons -CD
eTrust secons v8.0 (8.0) - Console Utility

================================================================================
            FILE CACHE (configuration, statistics, and dispatcher data)
--------------------------------------------------------------------------------
sizes(bytes)       tables:                | max records:      | intervals
cache   head      files    users    auth  | files users auths |clean prio
--------------------------------------------------------------------------------
40244    44        5600     4200    30400 |  20    50    80   |   60   1
================================================================================
table  |statistics        | priority    |min | rec  | average      |pri |init
name   |  hits misses (ok)| maxim  minim|ind | used | usage   life |fact|prio
--------------------------------------------------------------------------------
files  |    5     1   83% |   0      0  |  0 |   1  |              |    |
users  |    5     1   83% |  10      2  |  0 |   1  |  0       0   |  1 | 10
auths  |    4     2   66% |   2         |  0 |   2  |              |    |
================================================================================
FILE TABLE
--------------------------------------------------------------------------------
No  type    pid priority user                                 file name
--------------------------------------------------------------------------------
0   EXPL    372      0    0                                     /etc/shadow
================================================================================
USER TABLE
--------------------------------------------------------------------------------
No   user name     prio   life   used   UID  EUID  RUID auth prev(file)next
--------------------------------------------------------------------------------
0    root           2       2      7     0    0     0    0    50(   0)  50
================================================================================
AUTHORIZATION RESULT TABLE  (R - Result: 'P'-permit, 'D'-deny ...)
--------------------------------------------------------------------------------
No  R ACEE acc  Log stage prv(usr)nxt time       terminal  program
--------------------------------------------------------------------------------
0   P   6  read  0  00036 80(  0)  1   07:48:25            /usr/bin/login
================================================================================
```

Cache Settings Standard Output

The output consists of five parts:

1. The cache configuration. It contains the following fields:
   - Size of the cache (in bytes)
   - Size of the cache header (in bytes)
   - Size of the file table (in bytes)
   - Size of the user table (in bytes)
   - Size of the results table (in bytes)
   - The maximum number of file records
   - The maximum number of user records
   - The maximum number of result records
   - Statistic: hits in the table
2. The table of file records. It contains the following fields:
   - Sequential number of the record
   - Type of the file (EXPLICIT, IMPLICIT)
   - Process ID number
   - Priority of the record, is sum of its users priorities
   - Appropriate user record number in the table of users
   - Name of the file
3. The table of users. It contains the following fields:
   - Sequential number of the record
   - User name
   - Priority of the record
   - Record lifetime counter
   - Record usage counter
   - User ID; user effective ID; really used by security ID
   - Appropriate authorization record number in the table of authorization
   - Previous user record number in the chain of users
   - Appropriate file record number
   - Next user record in the chain of users

4. The table of authorization results. It contains the following fields:
   - Terminal
   - Stage
   - Granted stage
   - Result - authorization result (P or D)
   - ACEE number
   - Access type
   - Logging options flag value
   - The stage number the decision was made
   - Previous authorization record number in the chain of records
5. Appropriate user record number
   - Next authorization record number in the chain of records
   - Statistic: the number of missed records in the table
   - Authorization class
   - Program name (with the via parameter)
   - Notification string
   - Update time (GMT)
6. Dispatcher Data. It contains the following fields:
   - Statistic: number of missed records in the table
   - Statistic: number of hits in the table
   - Maximum priority in a table
   - Minimum priority in a table
   - Number of entries with minimum priority
   - Number of used records
   - Average usage (only for users table)
   - Average life (only for users table)
   - Priority calculation factor (only for users table)
   - Initial value of the record priority (only for users table)

secrepsw

Creates Policy Model password and shadow files.

The *secrepsw* utility with the -c switch generates a password record for every user in the /etc/passwd file. This is necessary for administering users defined by PMDBs operating over a UNIX environment. It can also create and remove shadow files. Only root can use this utility.

Note: This utility is located in the lbin directory.

Syntax

```
secrepsw [switch]
```

Switches

-c

Creates a new Policy Model password file from the /etc/passwd and /etc/shadow files on the local machine.

-h

Displays help.

-r *PolicyModel*

Transfers user names and passwords from the Policy Model's shadow file back to the original Policy Model password file (passwd).

-s *PolicyModel*

Transfers user names and passwords from the Policy Model password file (passwd) to the Policy Model's shadow file

Files

The seos.ini file is not used. You must change the shadow token in the pmd.ini file to “yes” before using the secrepsw utility.

sedbpchk

Checks the integrity of the database, and if the database passes the checks, creates a backup copy of the database.

The sedbpchk script copies the runtime database to a temporary location, performs various database integrity checks on the temporary database, and-if the database passes the checks-copies the temporary database into a backup location.

If the database does not pass the integrity tests, sedbpchk tries to determine whether any updates were applied to the database while the copy was being made. If there were updates, the conclusion that the database is corrupted may not be accurate.

If there were no updates while the database was being copied, the conclusion that the database is corrupted is probably true. In that case, a mail message is sent to the system administrator, who can then use the backup directory to override the corrupted runtime database.

Note: This script is **not** foolproof. It may conclude that a database is corrupted when it is not. However, the conclusion that a database is okay is always accurate.

You must have root and ADMIN privileges to run this script. Before using sedbpchk, we recommend that you review the script, located in *eTrustACDir*/lbin as sedbpchk.sh, to confirm that the values of the following fields match the needs of your site.

MAIL_TO

The name of the user who is sent the notification that the database is corrupt.

RETRIES

The number of times the utility checks the database when it suspects that the database is corrupted before sending the notification.

eTrustACDir

The location of the eTrust AC installation directory.

SE_BINDIR

The location of the eTrust AC binary files directory.

SE_DB_DIR

The location of the eTrust AC runtime database directory.

SE_BCKDIR

The location of the backup database directory.

SE_TMPDIR

The location of the temporary database directory.

Syntax

sedbpchk

Files

The seos.ini file is not used. No other special files are used.

See Also

The dbmgr utility in this chapter.

seerrlog

Displays the records in the eTrust AC error log.

The seerrlog utility lists the records contained in the eTrust AC error log file. The user who invokes the utility must either have permission to read the error log file, or be a member of the group that can read the error log files-the group in the token error_group.

Syntax

```
seerrlog switch
```

Switches

-s *date*

The start date for the list. Lists records written on and after the specified date. The format of *date* is *dd-mm-yyyy*.

-e *date*

The end date for the list. Lists records written up to and including the specified date.
The format of *date* is *dd-mm-yyyy*.

-d

Does not print the detailed information of failures.

-h

Displays the help screen.

-f *filename*

Specifies the error log file from which the list is to be generated.

Examples

- To list all error records written since 3 January 2004, specify:

  ```
  seerrlog -s 03-Jan-2004
  ```

- To list all error records written between 3 January 2003 and 1 January 2004, specify:

  ```
  seerrlog -s 03-Jan-2003 -e 01-Jan-2004
  ```

The seos.ini File

The seerrlog utility uses the following tokens in the seos.ini file:

Section	Token
logmgr	error_log error_group

For more information, see the *Administrator Guide*.

Other Files

The seerrlog utility uses the eTrust AC error log file, located in *eTrustACDir*/log/seos.error. You cannot define this file in the database, and only eTrust AC can write to the file.

See Also

- The seaudit utility in this chapter.
- The chapter “The Audit Browser: seauditx” in the *User Guide.*

segrace

Displays various login settings for a user.

The segrace command line utility displays the number of grace logins left for a user; the number of days remaining until the user's existing password expires; or the date and time the user last logged on, and from which terminal. For more information on the grace login property of a user, see the *Administrator Guide*.

Notes:

- Before segrace can work, the system administrator must activate eTrust AC password checking by entering the command:

  ```
  eTrustAC> setoptions class+(PASSWORD)
  ```

  Subsequently, every time a user's password is changed, the new password is checked against the password quality rules set in the database.
- If you invoke segrace without any parameters, and no grace logins are found for a user, it does not display anything.
- We recommend that you run the segrace command every time a user logs in. To do so, add the command line to /etc/profile and /etc/csh.login (or /etc/.login for Solaris).
- To permit segrace to count grace logins, you must use the sepass utility to change passwords. For more information, see the sepass utility in this chapter.
- If users have no grace logins left, segrace invokes the sepass utility, which requests that the users replace their passwords. Your site may decide which command to execute instead of the sepass utility by specifying another utility in the sepass_command token in the segrace section of the seos.ini file.

Syntax

```
segrace options [userName]
```

Options

-d *days*

Displays the number of days that remain until the user's current password expires. The number appears only if the number of days you specify in the *days* parameter is greater than, or equal to, the interval value in the eTrust AC option. If you omit the *days* parameter, segrace uses a default of seven days. This option works only if the user's password was changed using sepass.

-h

Displays the Help screen.

-l

Displays the date and time the user last logged in, and from which terminal.

-p

Prompts for a new password when a user's password has expired.

Parameters

userName

If you specify a user name, and the requestor has the ADMIN attribute, segrace displays the required login information for the specified user.

If you do not specify a user name, segrace displays the login details for the current user.

segracex

Sets a new password in the X-Windows environment.

The segracex utility checks whether the user's password has expired. If it has, segracex displays a window in which the user can replace the password.

The segracex utility is designed to be linked to the user initialization scripts that are invoked after the user logs in to the desktop environment.

The utility checks eTrust AC grace login attribute of the user. If the number of remaining grace logins for the user is:

- zero, segracex forces the user to change the password.
- positive but less than the value specified in the grace parameter of the user or the global grace setting (if there is one), segracex advises the user to change the password.
- equal to or greater than the value specified in the grace parameter of the user or the global grace setting (if there is one), segracex does nothing.

When changing the password, segracex prompts the user for the old password. It then prompts the user for the new password.

- If eTrust AC password checking is enabled, segracex checks whether the new password complies with the password rules that are set in the database. If the new password passes the quality check, the user is again prompted for the new password.
- If password checking is disabled, the user is immediately re-prompted for the new password.

  When the new password is entered for the second time, the two copies of the new password are compared. If the copies are not identical, the user is prompted again for the new password.

If the two new passwords are identical, the password is updated in the following ways:

- The local host password files-/etc/passwd and any security files-and the local database are updated.
- If a value is defined in the passwd_pmd or parent_pmd token in the [seos] section of the seos.ini file, the appropriate PMDB is updated, which then propagates the update to its subscribers both in the UNIX environment and the database. If the token nis_env in the [passwd] section of the seos.ini file has a value (either nis or nisplus), the NIS or NIS+ server is updated. When a password is set on a master NIS server, the NIS password map is automatically reconstructed.

Syntax

```
segracex [-user userName]
```

Parameters

userName

If you specify a user name, and the requestor has the ADMIN attribute, segracex operates on the specified user.

If you do not specify a user name, *segracex* operates on the current user.

Files

The customizable resources, such as colors and fonts, are in the segracex file. During standard installation of eTrust AC, this file is placed in the following directories:

- For all platforms except Sun Solaris: /usr/lib/X11/app-defaults
- For the Sun Solaris platforms: /usr/lib/openwin/app-defaults

  The icon with the eTrust AC trademark is in the BigTradeMark_BW.xpm file, which you must put into the *eTrustACDir*/data/segracex directory after installation.

See Also

The segrace and sepass utilities in this chapter.

seini

Displays information about the eTrust AC database and initialization files for any host. This utility also sets the values of tokens in the initialization files for any host.

For any host, the seini utility can do the following:

- Display the path of the eTrust AC database
- Display the path of an initialization (.ini) file
- Display the contents of a token from an initialization file
- Set the value of a specific token in a specific section of an initialization file
- Delete a specific token from a specific section of an initialization file

If you do not specify any switch, seini displays the paths of the database and the seos.ini file.

Note: The seini utility can only update the seos.ini file when seosd is **not** running, or when a rule in the database specifically permits it.

seini can perform an intelligent token and section search, by including certain tokens in the seos.ini file. This feature checks for spelling errors by comparing each token or section with the one you specified until it finds an exact or partial match (within a 25% error margin). If it finds the relevant token or section, seini performs the specified operation; otherwise it displays an error message. For more information, see the *Reference Guide*.

Note: The intelligent search feature works only on the host where you invoke the seini utility.

Syntax

```
seini switches parameters
```

Switches

-d [*host*]

Displays the path of the database on the remote host. If you do not specify a host, seini displays the path of the local host.

-f [*host*.]*section*.*token* [*iniFile*]

Displays the value of the token in the section of the specified initialization file on a specified host. If seini cannot find the specified section or token, an empty line appears. You must separate the host, section, and token names with a period (.). If you do not specify the *iniFile*, eTrust AC searches the seos.ini file for the section and token. To display information about the local machine, simply specify the section and token names.

-g *section*

Displays a list of tokens for a specific section.

-h

Displays the help screen.

-H [*host*]

Specifies the remote host to be used with the -f, -r, -s, and -sn flags.

-i [*host*]

Displays the path of the initialization file seos.ini. If you do not specify a host, seini displays the path of the local host.

-r [*host.*]*section.token* [*iniFile*]

Deletes the token from the section of the initialization file in the specified host. If you do not specify the *iniFile*, eTrust AC deletes the token from the seos.ini file. To delete information on the local machine, specify the section and token names only.

-s [*host.*]*section.token newValue* [*iniFile*]

Sets the value of the token in the section of the initialization file in the specified host. If you do not specify the *iniFile* parameter, eTrust AC sets the value in the seos.ini file. If the section or token does not exist, and you specified a remote host, eTrust AC creates that section or token.

To create a section or token on the local machine, use the -sn switch.

-sn [*host.*]*section.token newValue* [*iniFile*]

Sets the value of the token in the section of the initialization file in the specified host. If you do not specify the *iniFile* parameter, eTrust AC sets the value in the seos.ini file. If the section or token does not exist, and you specified the local host, eTrust AC creates that section or token.

To create a section or token on a remote machine, use the -s switch.

Output

The output depends on the switch you specify in the command line. The following table provides sample outputs for different switches.

-i

/opt/CA/eTrustAccessControl/

-f seosd.trace_file

/opt/CA/eTrustAccessControl/log/seosd.trace

-f dummy.keyword

(no output)

-s seosd.trace_to *file*

The token seosd.trace_to now set to *file* (was file,stop)

The seos.ini File

The seini utility can display all of the tokens in the seos.ini file, as well as the path of the file.

Other Files

The seini utility also displays all tokens in any of the other .ini files. The name of the initialization file must always end in the suffix .ini. You can work on an .ini file from any remote host as long as you have WRITE and ADMIN privileges.

selang

The selang command shell.

The selang utility invokes a command shell that provides access to the eTrust AC database and the UNIX environment. The database is updated dynamically by issuing selang commands from within the command shell. selang commands are described in *Reference Guide*.

The result of the command's execution is sent to the standard output unless you include the -o option.

Syntax

```
selang options
```

Options

-c *command*

Executes *command* and exits. If *command* contains any spaces, enclose the entire string in single quotation marks. For example:

```
selang -c 'showusr rosa'
```

-d *dbdirectory*

Updates the database in the specified directory.

-f *fileName*

Reads the commands from the specified file rather than from the terminal's standard input. As selang executes the commands in the input file, the number of the line currently being executed appears on the screen. The selang prompt does not appear on the screen. After selang executes the commands in *fileName*, it exits.

-h

Displays the help screen.

-l

Updates the local database. This option replaces sedlang (the shell script selang invokes this command). It is only valid when seosd is not running, and only root can execute it.

-o *fileName*

Writes the output in the specified file. Each time you invoke selang, it creates a new, empty file. If you specify the name of an existing file, selang writes over the information currently in the file.

-p *policyModelName*

Updates the database of the specified PMDB, which must be in the local station (this is the database in the PMDB subdirectory). Changes to the database are not propagated to subscribers.

Note: This option is not valid if either sepmdd or seosd is running on the specified PMDB and is not the same as using the *hosts* command. For more information about the hosts command, see the *Reference Guide*.

Important! Do not make changes that require propagation in this mode. If you use native mode when making updates, eTrust AC updates only the native host files (as defined in the seos.ini file).

-r *fileName*

Reads the commands from the specified file. The file should consist of commands in normal selang syntax, separated by semicolons or line breaks.

After executing the commands in *fileName*, selang prompts the user for input.

If you do not specify *fileName*, selang uses the .selangrc file in the user's home directory.

-s

Does not display the copyright message.

-u *username password*

Runs selang without invoking a password.

To use this option, you must set the check_password token in the seos.ini file to yes; this prompts “Enter your password” when you run selang -u. You have three attempts to login.

The token no_check_password_users in the [lang] section of the seos.ini file contains a list of users that are bypass the password checking during a login to selang.

Note: If the check_password token is set to no, selang does not require any passwords.

Usage

Screen Prompt

When you enter the selang environment, a special selang prompt appears. The exact form of the prompt depends on your working environment. It looks similar to this:

```
eTrustAC>
```

If you are working in a UNIX environment, following an env(unix) command, you see:

```
eTrustAC(unix)>
```

Standard Smart Features

selang supports many of the command line entry features available in tcsh and other smart shells.

Special Characters

selang also supports the following special characters:

- # or *

  At the beginning of a line, indicates that the line is a comment; the line is not executed. Comment lines are useful when inputting the selang commands from a file.

- !

  At the beginning of the line, indicates that the rest of the line is a shell command. selang sends the command to the operating system shell program for execution; eTrust AC does not execute the line.

- *Up-arrow, Down-arrow, or* ^

  Retrieves a command from the history list (see History in this section).

- \

  As the last character of a line, indicates the command continues on the following line.

- ;

  Terminates a command and introduces a new command on the same line.

- | *pipe*

  Pipes the command output to the specified *pipe*.

- Tab

  Serves for word completion (see Shortcuts in this section).

- Ctrl+D

  With the cursor positioned at the end of the line, displays a list of words that match the word completion string in the command line.

  With the cursor positioned anywhere else on the line, deletes the character to the right of the cursor.

- Esc, Esc

  Displays the help text for the command in the command line. All the text in the command line is preserved, so that you can continue typing the command from where you left off.

Longer Lines

Type one selang command per line. To continue a command on the following line, type a backslash (\) at the end of the line.

History

selang stores executed commands in a *history list*. Use the up and down arrow keys to display commands in the command line from the history list. To see only the commands that start with a specific character or string, type the beginning of the command before using the up and down arrows. When you press Enter, the text currently displayed in the command line is executed.

The selang command shell supports the following shortcuts that use the commands stored in the history list:

Shortcut	Runs
^^ [*string*]	The previous command. If you specify *string*, selang appends it to the original command.
^*n* [*string*]	The *n*th command in the history list, where *n* is a positive integer. If you specify *string*, selang appends it to the original command.
^-*n* [*string*]	The *n*th command from the end of the list, where *n* is a positive integer. If you specify *string*, selang appends it to the original command.
^*mask* [*string*]	The most recently issued command that begins with *mask*, where *mask* is a text string. If you specify *string*, selang appends it to the original command.

Command Line Editing

You can edit the command line. Use the left and right arrow keys to move around within the line. You can insert characters by typing them directly in place, and delete characters with the standard Backspace and Delete keys, or by pressing Ctrl+D.

UNIX Exits

A *UNIX exit* is a program that you specify-a shell script or an executable-that runs automatically before or after a user or group is added or updated. For more information on UNIX exits, see the *Administrator Guide*.

Shortcuts

You can use several additional techniques to save keystrokes in the selang command shell:

Command Recognition

selang recognizes which command you want to execute as soon as you have typed in enough characters to distinguish it from all the other available commands. For example, the only command beginning with the letters ho is the hosts command. As soon as you type “ho,” selang can recognize the intended command. On the other hand, there are several commands that begin with the string “new.” You must add enough characters to distinguish between newusr, newgrp, newfile, and newres.

Abbreviations

Each command is also associated with a one- to four- letter abbreviation. For example, because there are several commands beginning with the string new, you can also use the abbreviation nu for the command newusr. These abbreviations are documented as part of the command syntax for each command in the *Reference Guide*. You can enter commands in either uppercase or lowercase. Record and class names, however, are case-sensitive.

Word Completion

Press Tab in the middle of a word to complete the word. Word completion is context-sensitive. If more than one word matches the specified string, selang uses the shortest word or word fragment that matches the string. For example, if you type the letter n, selang supplies ew, to form the word new.

If this is not the required word, type another one or two characters and press Tab again to complete the word. Press Ctrl+D to see all the possible options. This is useful if you are not sure which command to use. Using the example in the previous paragraph, if you add the letter u to the word new and press Tab, selang supplies sr, giving you the command newusr.

Words that are not part of the selang commands are stored in memory for use by the word completion feature later on in the same session. For example, if you type newusr Mercedes, and then later type showusr Me followed by Tab, the abbreviation Me is expanded to Mercedes, as follows:

```
showusr Mercedes
```

This assumes that you have not entered any other user name that begins with “Me.”

The seos.ini File

selang uses the following tokens in the seos.ini file:

Section	Token
passwd	AllowedGidRange AllowedUidRange DefaultHome DefaultShell YpGrpCmd YpMakeDir YpPassCmd YpServerGroup YpServerPasswd YpServerSecure
lang	check_password exit_timeout help_path post_group_exit post_user_exit pre_group_exit pre_user_exit query_size timeout use_unix_file_owner

Other Files

selang uses the following files:

- lang.ini

  This file contains configuration information that selang uses. The utility uses the lang.ini files in one or both of the following locations:

  – The directory where the seos.ini file is located.

  – The user's home directory.

  If you specify a token in only one of these lang.ini files, selang uses the value from that file. If you specify a token differently in the two lang.ini files, the value in the user's home directory overrides the other one.

  The values for the tokens DefaultShell and DefaultHome in the server's seos.ini file override the values set in the tokens DefaultShell and HomeDirPrefix in the lang.ini file.

  By default, the sample lang.ini files are located in the directory *eTrustACDir*/samples/lang.init. For a description of the tokens in the lang.ini file, see the appendix “The lang.ini File.”

- .selangrc

  This file, located in your home directory, is the default file for the -r option. It is a file of selang commands that execute automatically each time you invoke selang. It is your responsibility to write the file if you want it.

- A pair of help files and a pair of help index files. Do not edit these files.
  - lang.hlp
  - lang.idx
  - langunix.hlp
  - langunix.idx

seldapcred

Use the seldapcred utility to encrypt and store a credential you provide. This credential is used by LDAP-enabled eTrust AC utilities (such as sebuildla) for retrieving data from an LDAP Directory Information Tree (DIT). Together with the value of the ldap_userdn token in the [seos] section of the seos.ini file, it lets the utility authenticate to the LDAP service. For a simple authentication, the credential is a password corresponding to the ldap_userdn value. For SASL authentication, the credential has different semantics.

The seldapcred utility writes the encrypted credential to *<eTrustACDir>*/etc/ldapcred.dat

```
seldapcred [-h] [-w [credential]]
```

-h

Show utility usage.

-w [*credential*]

Defines the credential you want seldapcred to encrypt and store. If you do not provide input to the seldapcred utility, it prompts you to enter this value. By using the interactive mode in this way, you prevent exposing the credential to other users.

More information:

sebuildla (see page 60)

seload

Loads the eTrust AC extension to the UNIX kernel and starts the eTrust AC daemons.

The seload utility loads eTrust AC daemons locally and remotely. It also determines whether the eTrust AC extension to the UNIX kernel is loaded on the specified host. If seosd is not running, seload starts the daemon on the specified host. If you omit the -r switch and parameter, the seosd daemon runs on the local host.

You can instruct seload to load one of the following daemons on the remote host: seosd, selogrd, selogrcd, or serevu. This process depends on the tokens.

Use seload if eTrust AC is placed in the boot sequence of the server station.

Notes:

- When eTrust AC is installed, sample initialization files for every operating system supported by eTrust AC are placed in the *eTrustACDir*/samples/system.init directory. Use these files if eTrust AC is to be started as part of the system initialization.
- The seload utility requires that the executable se_loadtest be located in *eTrustACDir*/lbin (where *eTrustACDir* is the installation directory). This program determines whether the eTrust AC extension to the UNIX kernel is loaded.
- When working remotely, the seload utility requires the following:
  - The executable rseloadd is located in eTrust AC dir/lbin. This program runs on the remote host and activates seload.
  - The file /etc/services contains seosload service. You should add this file during eTrust AC installation.
  - The file /etc/inetd.conf contains the rseloadd program. You can add this program during eTrust AC installation.

Syntax

```
seload options
```

Options

-c

Changes the encryption key that was set using the sechkey -r command.

-nopmd

If you specify the -c switch with the -nopmd switch, seload does not update the Policy Model update file with the new key.

-r *host* [*daemon*]

Loads the seosd daemons, and any other daemon specified in the [daemons] section of the seos.ini file.

If you specify a *daemon*, seload starts only that daemon; it ignores the seos.ini token. You must supply with the daemon's full path.

The seos.ini File

The seload utility uses the following tokens in the seos.ini file:

Section	Token
daemons	*token=text*
seos	SEOSPATH

The token in the [daemons] section is used only if you specify a value-the token has no default value. If you do specify a value, seload substitutes the value in the token for the standard values of the specified utility or program. For example, if you specify the value selogrd=yes, seload automatically starts the selogrd daemon after it starts the seosd daemon.

For more information, see the *Administrator Guide*.

Other Files

The seload utilite*TrustACDir*/lbin/se_loadtest file

- *eTrustACDir*/lbin/rseloadd
- /etc/services
- /etc/inetd.conf
- y uses the following additional special files:

selock

Screen saver and locker.

The selock utility protects your X terminal or station whenever you are away from your work area for any length of time. selock supports three modes of operation:

- Monitor mode
- Saver mode
- Lock mode

The default settings of selock combine the saver and lock modes.

Syntax

```
selock options
```

Options

-delay *period*

Specifies the amount of time the system icon appears at one location on the screen before fading away and moving elsewhere on the screen. This is the standard screen saver activity and prevents screen burn-in. The time period is entered in microseconds. The default value is 5000000 (five million).

-display *hostname:display#.screen#*

Specifies which display monitor to lock. You can find the display and screen numbers in an X-session listing of your system. You must have authorization from the user currently running the alternate display monitor specified here. The default assumption is that you want to lock your own display.

-folevels *levels*

Specifies the number of fade-out steps for the system icon. Increasing the number of fade-out levels causes smoother fading, but the icon takes longer to fade out. The default value is 20.

-fodelay *factor*

Modifies the length of time each fadeout level remains visible on the screen. This allows the user to extend the amount of time spent in each step without increasing the number of levels. The default value is 10.

-help

Displays a help screen that explains the various selock options.

-idelay *seconds*

Specifies the amount of time, in seconds, that passes after you log in before monitoring starts. If selock is part of your .login shell, this delay is needed while your system gets organized after you first log in. The default value is 30 seconds.

-lock-timeout *minutes*

If transparent=off, specifies the time, in minutes, selock spends in saver mode before changing to lock mode.

If transparent=on, specifies the time, in minutes, selock spends in monitor mode before changing to lock mode.

The default value, 0 invokes the lock mode immediately, effectively bypassing the saver mode.

-pixmapFile *fileName*

Specifies the XPM file that selock displays in the background when the screen is locked and the transparent=on.

-pw-timeout *seconds*

Specifies the length of time the password dialog box remains on the screen. The default value is 30 seconds. Note that too large a number can cause problems with the X-server.

-segrace { on | off }

Specifies for selock to invoke segracex after identifying the user and password. However, selock does not invoke segrace if the user ID and password belong to the user whose name appears in the unlocking_user token (located in the [selock] section of the seos.ini). The default value is off.

Note: The segracex utility checks whether the user's password expired; if it has, a dialog appears in which the user can select a new password. For more information, see the segracex utility in this chapter.

-timeout *minutes*

Specifies the period of user inactivity after which selock switches from the monitor mode to the save mode. The default value is 10 minutes.

-transparent { on | off }

Specifies whether selock leaves the contents of the screen visible when in lock mode. If you specify *on*, the display and update of on-going processes continues. To indicate that the screen is locked, selock changes the background by displaying the contents of the file specified with the -pixmapFile option.
The default value is *off*.

-user *user-name*

Specifies the user whose password is prompted for in the password dialog box, when user activity is detected in lock mode. The default value is the current user name. The password of root is accepted, regardless of which user is specified by the user option.

-workhours (*hh*:*mm*-*hh*:*mm*)

Specifies the period in which the user can unlock the screen. Before or after the specified period, the password dialog box does not appear if you touch the keyboard or mouse.

The default value is 00:00-24:00; that is, the user can always unlock the screen.

-xmin *pixels*

Specifies the minimum horizontal distance, in pixels, that the system icon jumps at each move. The default value is 100.

-xmax *pixels*

Specifies the maximum horizontal distance, in pixels, that the system icon jumps at each move. The default value is 300.

-ymin pixels

Specifies the minimum vertical distance, in pixels, that the system icon jumps at each move. The default value is 80.

-ymax pixels

Specifies the maximum vertical distance, in pixels, that the system icon jumps at each move. The default value is 250.

Monitor Mode

The monitor mode is the initial mode of selock. In this mode, selock monitors keyboard and mouse activity. If selock detects no keyboard or mouse activity during the time-out period-and the transparent parameter is off-selock automatically switches to the saver mode.

Saver Mode

In the saver mode selock blanks the entire screen and displays a system icon that shifts position. The default system icon is the eTrust AC logo and is located in the file *eTrustACDir*/data/admin/Selogo.xpm (by default, *eTrustACDir* is /opt/CA/eTrustAccessControl.) You can select an icon of your own choice by replacing this file. The icon file must be in XPM version 3.3 format.

The blank screen and shifting icon provide two operational advantages:

- Reduced risk of screen viewing by unauthorized people
- Reduced screen burn-in

You can manipulate the appearance and repositioning of the eTrust AC logo. For more information, see the selock options in this section.

When selock detects any keyboard or mouse activity, it immediately returns from the saver mode to the monitor mode, restoring the screen display to what it was before it switched to saver mode. No password entry is required for the transition from the monitor mode to the saver mode.

If selock remains in the saver mode for the period specified by the lock-timeout parameter, it automatically switches to the lock mode. selock does not give any visual indication of the transition from the saver mode to the lock mode.

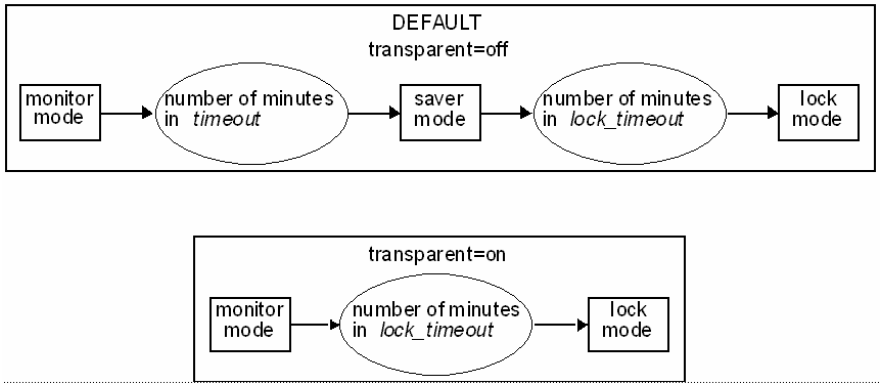
Lock Mode

In lock mode with the default settings, selock continues to display a moving eTrust AC logo on a black background. When selock detects any keyboard or mouse activity, a dialog containing a prompt for the user's password appears.

When the user enters the correct password, selock switches back to the monitor mode. If the user enters an incorrect password, the password-entry dialog closes and selock remains in the lock mode.

If you set the -transparent option to *on*, selock locks the screen but displays the contents and updates the on-going processes. The background of the screen changes to indicate that the screen is locked. When you use the lock mode, save mode is never invoked.

The following is a graphic representation of the two methods of working:



Usage

- Less security, more convenience

  Use the time-out option to set the time-out to a large value, such as 10 minutes, and the lock-timeout option to set the lock time-out to an even larger value, such as 60 minutes. This prevents selock from excessively interrupting your work by switching to the *saver* mode. Further, this setting locks your screen only in cases when your terminal is left inactive for extended periods.

- More security, less convenience

  Use the time-out option to set the time-out to a small value, such as 1 minute, and lock-timeout option to set the lock time-out to a small value, between 0 and 2 minutes. This always hides your work soon after you stop accessing your terminal, and requires a password for restoring access. To ensure that selock always requires password-entry to reactivate your terminal after the saver mode starts, use the lock-timeout option to set the lock timeout to zero.

- The selock command can be part of the X startup shell, so that it starts automatically every time the user logs in to the system. The script must be run under the user ID, not under the root ID. The way you integrate the selock command into the startup script depends on the specific environment of the site. For more information on startup scripts, see the documentation for your UNIX system.

Notes:

- Two users can always unlock a locked screen. By default, these users are the current user and root. However, you can replace root with any other user if you specify the other's user name in the unlocking_user token, located in the [selock] section of the seos.ini file. You can replace the current user with any other user by using the -user option when executing selock.
- The selock utility can find the password of users who can unlock a screen even if those users change their passwords while selock is active.
- The pw-timeout option specifies the maximum time allowed for the password to be entered. If selock detects that the password was not entered correctly within the specified period, the password-entry dialog closes and selock remains in lock mode.

The seos.ini File

The selock utility uses the following token in the seos.ini file:

Section	Token
selock	unlocking_user

Copyright Notices

selockcom

Control program for the selock utility.

The selockcom utility controls the currently active selock process. This includes restarting and stopping selock, as well as switching between the lock, saver, and monitor modes.

Note: When selock is loaded, it disables the terminal's built-in screen saver to prevent race or overlap conditions between selock and the built-in screen saver. If you stop selock with the selockcom exit switch, note that no screen saver is active on your terminal. You can restart either selock or the terminal's built-in screen saver using the standard X command xset s on. For more information on the xset command, see the xset(1) documentation.

Syntax

```
selockcom switch options
```

Switches

-activate

Switches selock from the monitor mode to the saver mode without waiting for the predefined time-out period to pass. The keyboard is locked and the eTrust AC logo appears on the screen.

-deactivate

Switches selock back to the monitor mode. This switch simulates user input to the selock process. If selock is currently in the lock mode, the password dialog appears; enter your password to return to the monitor mode. If selock is in the saver mode, you are returned to the monitor mode.

-exit

Terminates the selock process. You can also terminate selock by sending it a sigterm signal. As a last resort, you can also use the sigkill signal (kill -9). If you use the last method, selock does not exit gracefully; therefore you should not normally use it. If you are running a virtual-root window manager, using kill -9 forces you to restart the window manager to restore the virtual window.

-restart

Terminates the selock process and then immediately restarts it with the same command line options as the previous invocation. This is a good way to get selock to re-read the resource database if the database was changed since you last invoked selock.

-lock

Switches selock to the lock mode, regardless of the current lock-timeout value.

Options

-display *hostname*:*display#*.*screen#*

Instructs selockcom to control the selock process operating on the specified display. This option allows you to control selock from a remote terminal.

You can find the display and screen numbers in an X-session listing from your system. To do this, you must have authorization from the user currently running the specified display monitor. The default assumption is that you want to lock your own display.

Copyright Notices

selogmix

Splits and merges eTrust AC audit log files.

Use the selogmix utility to split and merge eTrust AC audit log files. Switches are mandatory arguments that control the operation. You must specify at least one switch. Use the options to specify file names.

Syntax

```
selogmix switches [options]
```

Switches

-h

Displays the help screen and examples for selogmix.

-s

Splits a specified audit log file.

-m

Merges two audit log files.

Options

-fn *file_name*

Specifies the name of the audit log file to be split or the resulting file of a merge. If you omit this option, selogmix uses the file name specified by the audit_log token in the [logmgr] section of the seos.ini file.

-l *file_name1 file_name2*

Specifies the files used in the merge or split operation.

You must specify both file names for this option. For merging, specify the two file names you want to merge; for splitting, specify the two destination files. If you omit this option, selogmix uses the file name specified by the audit_log token in the seos.ini file and adds sequential numbers to the end of the file name.

-c *weight1*:*weight2*

Specifies the correlation of file sizes for splitting files where *weight1* indicates the relative weight of the first file and *weight2* indicates the relative weight of the second file. If you skip this option, selogmix uses a one-to-one correlation.

-t *number_of_days*

Specifies a number of days. You can only use this option for splitting files. Specify how many days from the end of logging to put into a separate file. If you omit *number_of_days*, selogmix separates one last logging day.

-d

Specifies to run selogmix in debug mode. In this mode, selogmix displays all settings.

-I

Specifies to run selogmix in interactive mode. In this mode, selogmix prompts you for confirmation before overwriting existing files; otherwise, it overwrites without confirmation.

The seos.ini File

The tokens relevant to the selogmix utility are in the [message] and [logmgr] sections of the seos.ini file.

Section	Token
message	filename
logmgr	audit_log

Examples

1. To split a log file into two files of equal size, use the following command:

   ```
   selogmix -s
   ```

   The original audit file is named *eTrustACDir*/log/seos.audit

   The new split files are named *eTrustACDir*/log/seos.audit1 and *eTrustACDir*/log/seos.audit2.

2. To separate records for the last two days from the log file, use the following command:

   ```
   selogmix -s -t 2
   ```

3. To split a log file into two files with a defined correlation in size, use the following command:

   ```
   selogmix -s -c 1:2
   ```

4. To merge two specified files into one named file, use the following command:

   ```
   selogmix -m -l seos.audit1 seos.audit2 -fn seos.audit.merge
   ```

See Also

The seaudit utility in this chapter.

selogrcd

Collector daemon for the eTrust AC log routing system.

The eTrust AC log routing daemons, selogrd and selogrcd, provide system administrators with convenient, selective access to the audit log records.

The selogrcd utility is the collection daemon. This daemon collects the selected audit log records sent by various satellite systems and stores them in the audit collection file. The default file is *eTrustACDir*/log/seos.collect.audit.

You can force selogrcd to start a new audit file by sending it a USR1 signal. Once you have the selogrcd process ID, send it a USR1 signal using a kill command such as:

```
# kill -USR1 processID
```

When it receives a USR1 signal, selogrcd renames the existing audit file to *eTrustACDir*/log/seos.collect.bak and creates a new audit file. You can also use a cron job to perform this task periodically. A sample script that performs this task is provided in the directory *eTrustACDir*/samples/selogrcd.

Syntax

```
selogrcd options
```

Options

-d

Specifies the debug mode. In this mode, selogrcd does not become a daemon. It sends debug information to the terminal.

-h

Shows the usage screen.

-l *lock-file-name*

Specifies the name of the lock file to be used (*lock-file-name*). By default, selogrcd uses the file /tmp/selogrcd (or in Sun Solaris 4.x /var/spool/locks/selogrcd). Use this option only if your /tmp system does not support file locking.

Features

Two tokens enhance audit collection file management. Both tokens are in the [selogrd] section of the seos.ini file

- Use the Caudit_size token to specify the maximum size of the audit collection file. When the file reaches this size, eTrust AC creates a backup file and opens a new file.
- Use the CbackUp_Date token to specify a automatic backup interval and timestamp for the audit collection file.

The seos.ini File

The tokens relevant to the collector daemon are found in the selogrd and logmgr sections of the seos.ini file.

Section	Token
selogrd	Caudit_size CbackUp_Date CollectFile CollectFileBackup ConsolePort ServicePort
logmgr	audit_group

Other Files

The collector daemon uses the following additional special files:

- *eTrustACDir*/etc/selogrcd.ext (used by the LogRoute API, described in the *SDK Developer Guide*.)
- *eTrustACDir*/log/seos.collect.audit
- *eTrustACDir*/log/seos.collect.bak
- *eTrustACDir*/log/seos.audit

See Also

- The selogrd and seaudit utilities in this chapter.
- The chapter “The Audit Browser: seauditx” in the *User Guide.*
- You can expand the functionality of the selogrcd daemon by writing programs at your site that use the APIs provided with eTrust AC. For more information, see the *SDK Developer Guide*.

selogrd

Emitter daemon for the eTrust AC log routing system.

The eTrust AC log routing, daemons selogrd and selogrcd, provide system administrators with convenient, selective access to the audit log records.

The selogrd utility is the emitter daemon. This daemon distributes selected local audit log records to the various destination hosts; reformats audit log records into email messages, ASCII files, or user windows; and sends out notification messages based on audited events.

Note: The eTrust AC daemon must be up and running before the log routing daemons can collect any meaningful information on eTrust AC events. If the eTrust AC daemon is not running, selogrd routes only old audit records.

The log routing daemons use a configuration file to determine where each audit log record is sent, the format in which the log record is written, and which records are routed. By default, selogrd uses the *eTrustACDir*/log/selogrd.cfg file. For a discussion of the format of the configuration file, see The Log Route Configuration File in this section. The names of the configuration file and other global environment variables that selogrd and selogrcd use are specified in the eTrust AC initialization file, seos.ini.

The selogrd daemon periodically restarts and reads the configuration file. In addition, you can force the selogrd daemon to restart at a specified time. To do so, you must send the following HUP signal:

```
kill -HUP processID
```

where *processID* is the selogrd process ID. (Use the UNIX ps command to find it; see your UNIX documentation for more information.)

The selogrd utility provides API access for programmers working under eTrust AC. The Logroute API allows programmers to incorporate their own options into the eTrust AC audit log system to support in-house alerts not provided by the current log-routing facility. The Logroute API also allows programmers to use the log routing daemons to provide functions to their own programs. For more information on all the eTrust AC APIs, see the *SDK Developer Guide*.

Syntax

`selogrd` *options*

Options

-audit *audit-file-name*

Specifies that the utility use the file name provided instead of the file listed in seos.ini for the input audit file.

-config *config-file-name*

Specifies that the utility use the file name provided instead of the file listed in seos.ini for the configuration file.

-d

Specifies the debug mode.

-data *data-file-name*

Specifies that the utility use the file name provided instead of the file listed in seos.ini to store routing progress information.

-h

Shows the usage screen.

-pmdb *policy-model-name*

Instructs selogrd where to route audit data from a PMDB. The command tells selogrd to send audit data from the PMDB that you specified in the command, to the audit file that you specified in the audit_log token in the pmd.ini file of the PMDB.

By default, selogrd uses the data file and lock file that consist of the Policy Model name. If you specify the data file or lock file or both on the command line, those files override the default values. The lock file and data file names should be different from those of the selogrd that route the audit data of the station. selogrd can only support Policy Model names of 12 characters.

The audit data that is sent from a PMDB appears in the collected audit file as if it comes from a station with the name policy-model-name@station-name

Encryption

You can encrypt audit log records. When you use encryption, selogrd encrypts audit log record before sending it to the collector (selogrcd or audit log router). The collector in turn decrypts the received records.

eTrust AC provides two encryption styles for selogrd-eTrust AC standard encryption, and eTrust Audit encryption. For encryption, selogrd uses functions from shared library objects, as specified in the [selogrd] section of the seos.ini file.

Standard encryption uses the shared library libcrypt; Audit encryption uses functions from a file specified by the CipherName token. By default, the file name is adcipher, which is a symbolic link to the desired shared library. The eTrust AC installation process places four shared libraries in the eTrust AC/lib directory: lib1des, lib3des, libIDEA, and libblowfish.

eTrust AC maintains the standard encryption key in the shared library, while the Audit encryption uses a separate file as specified by the KeyFile token (default value: adcipher.bin).

Use the UseEncryption token to determine the type of encryption:

- To use eTrust AC standard encryption, specify UseEncryption=native
- To use eTrust encryption, specify UseEncryption=eTrust, and enter the appropriate values for the CipherName and KeyFile tokens.
- To disable selogrd encryption, specify UseEncryption=no.

Use the RefuseUnencrypted token to accept or deny unencrypted audit. It is used in conjunction with the UseEncryption token and is redundant if the UseEncryption is set to no:

- To refuse unencrypted audit, specify RefuseUnencrypted=yes
- To accept both encrypted and unencrypted audit, specify RefuseUnencrypted=no

Note: The selogrcd daemon uses the same tokens in the seos.ini file.

To change the encryption key, use the sechkey utility, described in this chapter.

Important! If you send records to the audit collector, be sure that both selogrd and the collector use the same shared encryption file and encryption key.

SMTP Mail

selogrd can send records to email targets directly. You can direct email messages through a mailer utility (the old method), or directly to the mail exchange server using SMTP. To do this, set the UseSmtpMail token in the [selogrd] section of the seos.ini file.

The new method does not use UNIX mail utility; rather, it establishes a direct connection with mail server, and uses SMTP protocol to send mail.

You can also specify the following:

- A time-out in case the mail server does not answer, using the SmtpTimeLimit token
- The “From:” mail header field, using the SmtpMailFrom token
- The mail server host address, using the SmtpMailServer token

The Log Route Configuration File

The following is the format of the configuration file, followed by a detailed explanation.

```
section-name-1

routing-method destination
[{include|exclude} match-field(match-pattern) ...]
...
.
section-name-2
routing-method destination
[{include|exclude} match-field(match-pattern) ...]
...
.
...
```

Specifying Audit Records

The configuration file is a list of which audit records to route-and which not to route-to various destinations. To specify audit records, you describe the contents of one or more particular fields. You can use the standard UNIX pattern matching (the wildcards * and ?).

For example, to specify records that deal with users whose user names begin with the letters dbms, you would enter the following:

```
User(dbms*)
```

This example matches users with names like dbms1, dbms_mgr, and so on.

To specify the same users, but only the records that deal with their login attempts, you would enter:

```
User(dbms*) Class(LOGIN)
```

Note: When a line specifies records in terms of more than one field, it specifies only the records that match *all* those fields.

At the beginning of the same line that specifies the records, you specify whether you want the records included or excluded. For example, to include those records in the routing enter the following:

```
include User(dbms*) Class(LOGIN).
```

This type of line appears in the overall format as:

```
[{include|exclude} match-field(match-pattern) ... .]
```

Here, the "..." means that the first match-field(match-pattern) pair can be followed by further pairs.

You can use any of the following for match-field(match-pattern):

Access(*access-type*)

For the type of access required; *access-type* is any one of the following:

ACL, Chdir, Chgrp, Chmod, Chown, Connect, Control, Create, Erase, Exec, Kill, Modify, Owngrp, Password, Read, Rename, Replace, Update, Utimes, and Write.

Class(LOGIN)

For login records.

Class(LOGOUT)

For logout records.

Class(PWCHANGE)

For password administration.

Class(HOST)

For TCP/IP records.

Class(UPDATE *eTrust AC-class*)

For database administration. eTrust AC-class is any of the accessor or resource classes (such as USER, GROUP, FILE, HOSTNP...) or a pattern for the class name to match. Thus for all database administration, you can specify UPDATE * .

Class(*eTrust AC-class*)

For access to protected resources. For example, Class(FILE) refers to records reporting file access attempts.

Note that you can use an asterisk to combine Class(*eTrust AC-class*) and Class(UPDATE *eTrust AC-class*) as *Class(*eTrust AC-class*)*. For example, specifying Class(*FILE) is like specifying both Class(FILE) and Class(UPDATE FILE). It refers both to attempts to access files and to attempts to update records in the FILE class.

Code(*return-code*)

For the eTrust AC return code indicating what happened; return-code can take the following values. (See also Example 1 in this section.)

A-An attempt to log in failed because an invalid password was entered repeatedly.

D-eTrust AC denied access to a resource, did not permit a login, or did not permit an update to the database because the accessor did not have sufficient authorization.

E-Serevu enabled a disabled user account.

F-An attempt to update the database failed.

I-Serevu disabled a user account.

M-The executed command started or shut down a daemon.

O-A user logged out.

P-eTrust AC permitted access to a resource or permitted a login.

S-The database was successfully updated.

T-An audit record was written because all the actions of the user are being traced.

U-A Trusted program (setuid or setgid) was changed; therefore it is no longer Trusted.

W-An accessor's authority was insufficient to access the specified resource; however, eTrust AC allowed the access because warning mode is set in the resource.

Host(*host-name*)

For the host involved in a TCP/IP connection.

Object(*resource-name*)

For the resource that the user is attempting to access.

Reason(*reason-number*)

For the reason that the audit record is triggered.

Service(*service-name*)

For the name of the service requested from the remote host, such as telnet or ftp.

Source Host(*hostname*)

For the name of the host that contributed the record to the consolidated audit.

Stage(*stage-number*)

For the stage at which access was granted or denied. (See the lists of stage codes in the *Reference Guide*.)

Terminal(*terminal-name*)

For the terminal that is attempting access or administration.

Uid(*uid-number*)

For the uid of the user who is attempting access or administration.

User(*username*)

For users attempting access or administration; username is a name or pattern.

Note: Although some variables are more likely to be specified as patterns, you can use a pattern for any variable-even for something like a stage number.

Refining with Further Lines

To refine your specifications, you can filter by differing criteria at the same time. Simply add one include/exclude line after another. For example:

```
include User(dbms*) Class(*LOGIN*).
exclude Terminal(console_*).
```

The example specifies all login attempts by users whose names begin with dbms and who are at terminals that do not have names beginning with console_.

Specifying the Destination

Use a line *above* your sequence of include and exclude lines to specify the destination for the audit records you are including. For example:

```
mail weekwatch
include User(dbms*) Class(*LOGIN*).
exclude Terminal(console_*).
```

The example specifies that the email address weekwatch receives a report on all login attempts by users whose names begin with dbms and who are at terminals that do not have names beginning with console_.

This type of line appears in the format of the log route configuration file as:

routing-method destination

You can use any of the following methods:

mail *address*

To email the audit record; *address* is the destination address. If it is not in the form user@host, it is checked against local user lists and the NIS mail alias map.

Note: If address is a user name and surrogate requests to that user's account are audited, the audit records accumulate endlessly.

screen *username*

To display the audit record on the screen of the specified user, if that user is logged in at the current host when selogrd forwards the audit record. If the user is not logged in, the display is canceled, not postponed.

cons *hostname*

To send the audit record to the Security Administrator GUI of the secmon utility on the specified host. If that host is not available, the display is terminated, not postponed.

file *textfilename*

To write the audit record in the specified ASCII file; *textfilename* must be an absolute path name and selogrd must have access to the file.

host *hostname*

To send the audit record to the audit log collector on the specified host. If that host is not available, selogrd tries again later.

notify mail *or* notify default

To email the audit record to the address that the audit record itself specifies.

notify screen

To display the audit record on the screen of the user that the audit record itself specifies. If the user is not logged on, the display is canceled, not postponed.

uni hostname

To send the audit record to the Unicenter TNG event manager on the specified host. You must also set selogrd to load the uni.so shared library, which is found in the *eTrustACDir*/lib directory. Note that the installation performs this task for you if it finds Unicenter TNG installed on the specified host and you choose to do it.

Proper Sequence for Lines

It is important to arrange your include and exclude lines in proper sequence, properly delimited.

- You must precede each sequence of lines (or single line) that you want to treat as a single complex filter with a title line, and end it with a terminating line that consists of a single dot; for example:

  dbms login from non-console

  ```
  mail weekwatch
  ```

  include User(dbms*) Class(*LOGIN*).
  ```
  exclude Terminal(console_*).
  .
  ```

The full sequence, including the title line and terminating line, is called a *section* of the file.

- If both include and exclude lines match the same audit record in the same section, the last match overrides all others.
- If no lines match a particular audit record, then the first line of the section is the deciding line for that record. (If the first line is an include line, then the failure to match excludes the record. If the first line is an exclude line, then the failure to match includes the record for routing.)
- If the section includes no include and exclude lines, then it includes all audit records for routing.

How Sections Coexist

Whereas the lines of a section work together to produce a single decision as to whether or not a record is to be sent, different sections in the configuration file work entirely independently. Whether or not an audit record is sent by one section, has no influence on whether the same audit record is sent by another section.

You can send the same selection of audit records to more than one destination, and the same destination can receive more than one selection of audit records.

In your configuration file, the total of all the include and exclude lines-from all the sections together-must not exceed 64 lines.

Including Comments

To add a comment line to the configuration file, begin the line with a semicolon.

Example 1

The following is a sample configuration file, followed by its explanation.

```
; Product : eTrust Access Control
; Module  : selogrd
; Purpose : route table for audit log routing daemon
;
;-----------------------------------------------------
 Rule#1
 mail jones@admhost
 include       Class(*LOGIN*) Code(D).
 .
 Rule#2
 mail smith
 include       Class(*SURROGATE*) Object(USER.root*).
 .
 Rule#3
 host venus
 exclude       Class(UPDATE SU*).
 .
 Rule#4
 host venus
 include       Class(*PROGRAM*) Object(/usr/bin/ps).
 .
```

The first five lines are comment lines.

The next four lines make up the first section, named Rule#1. They tell selogrd to mail a log record to the address jones@admhost whenever a login request is denied (code D reports denial):

```
Rule#1
mail jones@admhost
include       Class(*LOGIN*) Code(D).
.
```

The next section is named Rule#2. It tells selogrd to mail a log record to the address smith whenever someone attempts to use the su command to enter the root account (the objects in the SURROGATE class are targets for the su command):

```
Rule#2
mail smith
include       Class(*SURROGATE*) Object(USER.root*).
.
```

The next section is named Rule#3. It tells selogrd to send a log record to the collector on host venus whenever someone attempts database administration, unless the class name begins with the letters SU (the matching classes are SURROGATE and SUDO):

```
Rule#3
host venus
exclude       Class(UPDATE SU*).
```

The last section is named Rule#4. It tells selogrd to send a log record to the collector on host venus whenever someone attempts to use the ps command:

```
(Code 1 8pt) Rule#4
host venus
include       Class(*PROGRAM*) Object(/usr/bin/ps).
.
```

Example 2

The following configuration file sends *all* audit records to the collector on the station named loghost:

```
 ; Product : eTrust Access Control
 ; Module  : selogrd
 ; Purpose : route table for audit log routing daemon
 ;
 ;-----------------------------------------------------
  Rule#1
  host loghost
  .
```

Return Codes

You can associate each type of record in the configuration file with one or more eTrust AC return code. (For a complete list of the return codes see the description of *code(return-code*) in Specifying Audit Records in this section.) The following table describes the record types and their associated return codes.

Record Type	Class or Event	Associated Return Codes
Login	LOGIN	D, P, W
	LOGINDISABLE	I
	LOGINENABLE	E
Logout	LOGOUT	O
TCP/IP	HOST	D, P
Resource classes	Class name	D, P, W
Watchdog	PROGRAM	U
	SECFILE	U
Password administration	PWCHANGE	D, F, S
Down	SHUTDOWN	D, S
Start	START	S
eTrust AC database administration	UPDATE	D, F, S
Trace on user	TRACE	F, D, P

SNMP Traps

For systems that use the Internet network management protocol SNMP (Simple Network Management Protocol), you can configure selogrd to create SNMP traps using eTrust AC audit records.

To implement the SNMP traps, first locate the SNMP shared objects provided in the eTrust AC libraries, and then configure selogrd correctly using these shared objects.

The shared objects-usually found in the directory *eTrustACDir*/lib- are called snmp.*xx* and libsnmp.*xx*, where the *xx* extension varies according to the platform. The possible extensions are:

- .o AIX platform
- .sl HP platform
- .so All other platforms

To configure selogrd to use the shared objects:

14. Create a file called *eTrustACDir*/etc/selogrd.ext.
15. Define where the SNMP shared objects are by adding a single line to the file *eTrustACDir*/etc/selogrd.ext with the appropriate path for the snmp.so. (It is enough to specify this shared object for the other to automatically be linked.) For example:
    ```
    snmp /opt/CA/eTrustAccessControl/lib/snmp.so
    ```
16. Finally, you must configure the selogrd.cfg file to specify what type of action should trigger SNMP traps, and which location should be notified when SNMP traps are triggered. Configuration is very similar to that for other auditing notification, with the delivery system specified as snmp.

    For example, suppose you want to have SNMP traps activated when eTrust AC starts and shuts down, and have notification of these SNMP traps sent to AuditPC. You can do this by adding the following section to the selogrd.cfg configuration file:

    ```
    snmpRule
    snmp AuditPC
    include Class(START).
    include Class(SHUTDOWN).
    .
    ```

    Similarly, you can activate the SNMP traps by other actions or types of access, or have them sent to other locations. For more information, see Specifying Audit Records in this section.

Note: If you want to use the SNMP extension of selogrd, and eTrust AC (or its upgrades) is not installed in the default location, you must set the following environment variables before running selogrd:

- In AIX, set LIBPATH to *eTrustACdir*/lib
- In Solaris, set LD_LIBRARY_PATH to *eTrustACdir*/lib
- In Linux, set LD_LIBRARY_PATH to *eTrustACdir*/lib
- In HP, set SHLIB_PATH to *eTrustACdir*/lib

  where *eTrustACdir* is the directory where you installed eTrust AC.

Configuration

In order to start selogrd or selogrcd automatically when seosd starts, set the seos.ini tokens selogrd or selogrcd in the [daemons] sections to yes. Then when you run seload, seload starts the daemons for you.

For example, the appropriate tokens in the [daemons] section of the soes.ini should look as follows:

```
selogrd = yes
selogrcd = yes
```

Since the log-routing facility uses RPC to route audit records, placing a log audit collector behind a firewall does not allow simple blocking of UDP ports because there is no way to know which port the portmapper assigns to the server daemon. To solve this problem, you can use the token ServicePort to assign a predefined port to the server daemon.

If the firewall allows port 111 from outside the network (portmapper port), you should only change the seos.ini file in the server. If the firewall does not allow communication to portmapper in the protected network, both clients and server must agree on a specific port.

You can ensure this by setting the same value in the ServicePort token in the seos.ini files of both clients and the server. You can specify a number-which means that the daemons bind to the specified port-or a service name. If you specify a service name, both clients and the server must have the same service resolution. For example, if you specify the service name seoslogr, then the following to the /etc/services file of the clients and the server:

```
seoslogr 2022/udp # Audit log-routing
```

If the clients or the server are using NIS to resolve services, you must update the NIS services map.

The seos.ini File

The tokens relevant to selogrd are in the [selogrd] and [logmgr] sections of the seos.ini file.

Section	Token
selogrd	Caudit_size CBackUp_Date ChangeLogFactor CipherName CollectFile CollectFileBackup ConsolePort DataFile Interval KeyFile Mailer MaxErrorSending MaxSeqNoSleep RefuseUnencrypted RouteFile SavePeriod sendmail_header_format ServicePort SmtpMailFrom SmtpMailServer SmtpTimeLimit tec_conf_file UseEncryption UseSmtpMail
logmgr	audit_group audit_log

For more information about these tokens, see the *Administrator Guide*.

Other Files

The selogrd utility uses the following additional special files:

- /etc/passwd
- /etc/services

- For all stations but SunOS
  - /tmp/selogrd
  - /tmp/selogrcd
  - For SunOS stations
  - /var/spool/locks/selogrd
  - /var/spool/locks/selogrcd
- *eTrustACDir*/etc/selogrd.ext (used by the LogRoute API, described in the *SDK Developer Guide*)
- *eTrustACDir*/log/seos.audit
- *eTrustACDir*/log/logroute.cfg
- *eTrustACDir*/log/logroute.dat
- For SNMP users
  - snmp.*xx* and libsnmp.*xx*-usually to be found in the directory *eTrustACDir*/lib
  - *eTrustACDir*/etc/selogrd.ext

See Also

- The selogrcd and seaudit utilities in this chapter.
- The chapter “The Audit Browser: seauditx” in the *User Guide.*
- You can expand the functionality of the selogrd daemon by writing programs at your site that use the APIs provided with eTrust AC. For more information, see the *SDK Developer Guide*.

semsgtool

Maintains the eTrust AC message file.

The semsgtool utility can perform the following functions:

- Show a single message from the eTrust AC message file.
- List an entire section of messages.
- Dump the entire file into ASCII files, one ASCII file for each section.
- Build a new message file.
- Change message to a new one.
- List messages, including substring.

You can only specify one command each time you execute semsgtool.

Syntax

```
semsgtool option fileName [parameter]
[-number | -n] [message-file] <sub-str>
```

Lists messages, including sub-str

```
[-change | -c] [message-file] [0x<error-code> | <section# msg#>] <new-
message>
```

Changes message to a new one. Creates new message file as [message-file].new

Options

-build | -b *asciiSourceFile OutputMessageFile*

Creates a new eTrust AC message file from an ASCII source file.

-dump | -d *messageFile*

Dumps the message file into several files, one file for each section of the message file. This creates ASCII source files that later can be used to create new eTrust AC message files.

-list | -l [*messageFile*] *sectionNumber*

Lists all the messages in a given section in the file *messageFile*. If you do not specify *messageFile*, semsgtool uses the message file as specified in the filename token of the seos.ini file. For sectionNumber, you can specify a hex number or a decimal number; you must precede a hex number with 0x.

-show | -s [*messageFile*] *messageCode*

Shows the message associated with a specific message code. If you do not specify *messageFile*, semsgtool uses the message file as specified in the filename token of the seos.ini file. For *messageCode*, you can specify a hex number or two parameters that represent a section code and a message code (in decimal or hex numbers). You must precede a hex number with 0x.

This option provides the same functionality as the seerr utility, which existed in earlier versions.

The seos.ini File

The semsgtool utility uses the following token in the seos.ini file:

Section	Token
message	filename

Other Files

The semsgtool utility uses the file specified in the filename token of the seos.ini file. The default value is *eTrustACDir*/data/seos.msg.

Notes:

- The eTrust AC message file is composed of sections and message numbers. Each section holds messages for different eTrust AC modules or sub-modules.
- This utility replaces the utility seerr, which existed in earlier versions of eTrust AC.

Examples

- To list the message associated with the error code 0x0205, enter:

```
# semsgtool -s 0x205
```

- To create a modified eTrust AC message file, do the following:

17. cd <message file directory>
18. semsgtool -change seos.msg 0x2501 "This is the new message"
19. cp seos.msg.new seos.msg

senable

Enables a previously disabled user account.

The senable utility enables the login of a user that was disabled for any reason, at any location at which the user was disabled, including PMDBs. For example, a user may have been disabled by the serevu daemon, or because the user's suspend date or expire date arrived.

After enabling the user account, senable calls the sepass utility, which prompts for a new user password. If you use the -n option, senable does not call sepass, and restores the most recently used password.

The senable utility enables an undefined user account by deleting that account from the local /etc/passwd file.

You must have ADMIN or PWMANAGER attributes on two hosts to use the -host option:

- The host with the account to be changed from disabled to enabled
- The host where you enter the senable command

To execute senable remotely, you must explicitly mention your local terminal needs in a rule that grants it WRITE permission for accessing the remote station; otherwise, you cannot perform eTrust AC administration there. For more information about remote administration restrictions, see the *Administrator Guide*.

Note: To use senable, seosd must be running, and you must have the ADMIN or PWMANAGER attribute.

Syntax

```
senable options
```

Options

-host hostname

Selects the host with the account to change from disabled to enabled.

-n

Runs the command non-interactively.

***userName* [*userName* ...]**

Specifies one or more user names for accounts being changed from disabled to enabled. Separate each name with a space.

-h

Displays the help screen for this utility.

See Also

The serevu utility in this chapter.

senone

Executes a command as a non-eTrust AC user process.

The senone utility executes a command issued by a highly authorized user as a non-authorized user process.

Note: Only highly authorized users who are testing untrusted programs should use this utility.

When you invoke the senone utility, it deletes the process credentials from the authorization daemon. senone then executes a shell with the credentials of a user who is not defined to eTrust AC. From this point on, any program invoked from within this shell is executed with the credentials of the non-eTrust AC user.

We recommend that users who are logged in as root not run untrusted programs. Even when running untrusted programs with senone, unexpected problems can occur.

If you invoke senone without specifying a command, it executes the user's shell as defined in /etc/passwd.

Notes:

- Because senone does not change the invoker's user ID, the user's UNIX privileges remain unchanged.
- To use senone, the eTrust AC authorization daemon seosd must be running.

Syntax

```
senone [{-h | command}]
```

Options

-h

Displays usage information.

command

The command to be executed by senone.

Files

The /etc/passwd file is used. The seos.ini file is not used.

See Also

The sewhoami and sesu utilities in this chapter.

SEOS_load

The eTrust AC interception module loader for all stations except Sun Solaris 2.

The SEOS_load utility controls the dynamic eTrust AC kernel module (SEOS_syscall). The interception module must be loaded before running any eTrust AC utility.

Note: You can use UNIX exits to automatically run programs before and after loading and unloading the kernel. For more information about eTrust AC kernel loader UNIX exits, see the Administrator Guide.

On streams supported platforms, this utility loads the eTrust AC module to streams depending on the SEOS_use_streams token in the [SEOS_syscall] section of the seos.ini file. If the token is set to yes, the module is pushed into streams.

Syntax

```
SEOS_load [-i|-k|-s|-u]
```

Options

(none)

Load the kernel extension into the kernel and push the kernel module into streams.

-i

(For HP-UX and Sun Solaris platforms only.) Display information about the eTrust AC kernal extension.

-k

(For HP-UX and Sun Solaris platforms only.) Load the eTrust AC module into the kernel without pushing into streams.

-s

(For HP-UX and Sun Solaris platforms only.) Insert the eTrust AC kernel module into streams. This option ignores the SEOS_use_streams token in the SEOS_syscall section of the seos.ini file.

-u

(For AIX, HP-UX, Linux, and Sun Solaris platforms.) Unload the eTrust AC kernel extension from the kernel and the remove the module from streams.

You cannot unload eTrust AC if an application, which is loaded on top of eTrust AC, has an open system call (syscall) that is hooked by eTrust AC. Use **secons -sc** to find these processes. You can then shut down these processes and unload the eTrust AC kernel module, or use UNIX exits to automatically shut down these processes before unloading the kernel and then then restart them after the kernel unloaded.

Files

The SEOS_load utility uses the following tokens in the seos.ini file.

Section	Token
SEOS_syscall	SEOS_use_streams

See Also

The SEOS_syscall utility in this chapter.

SEOS_syscall

The eTrust AC interception module.

SEOS_syscall is the image of the kernel interception mode, which is loaded into the kernel by SEOS_load.

On AIX, HP-UX, Linux, and Solaris platforms the image is loaded into the kernel in a dynamic manner and can be unloaded from the kernel without rebooting using the following command:

```
SEOS_load -u
```

Syntax

```
SEOS_syscall
```

Files

The seos.ini file is not used. No special files are used.

See Also

The SEOS_load utility in this chapter.

seosd

The eTrust AC authorization daemon. The executable file seosd is the main eTrust AC daemon. A daemon is a process that has disconnected from both its controlling TTY and its parent process. The eTrust AC daemon makes the runtime decisions required to grant or deny access to a resource.

Only root can invoke seosd, and only a user with the ADMIN or OPERATOR attribute can shut it down.

The eTrust AC daemon opens, reads, and updates the database. No other process can access this database while the eTrust AC daemon is running. The eTrust AC daemon also blocks any write, delete, or rename access to critical files, such as the eTrust AC audit and trace files and, optionally, the eTrust AC binary files.

Syntax

```
seosd [-d | argument]
```

The seos.ini File

You control the behavior of many seosd functions by setting values for the tokens in the [seosd] section of the seos.ini file.

This section describes the values of these tokens.

Section	Token

Section	Token
seosd	bypass_filenames
	bypass_suid_for_login
	bypass_suid_program
	bypass_system_files
	bypass_TCPIP
	cron_program
	dbdir
	device_file
	dns_server
	domain_names
	enf_register
	FileCache_files
	FileCache_users
	FileCache_auths
	FileCache_CleanInt
	FileCache_PriorInt
	FileCache_InitPrio
	grace_admin
	GroupidResolution
	HostResolution
	IsolatedDaemon
	kill_ignore
	lookaside_path
	lookaside_allowdupuid
	network_cache_timeout
	nfs_devices
	protect_bin
	resolve_rebind
	resolve_timeout
	rt_priority
	ServiceResolution
	trace_file
	trace_file_type
	trace_filter
	trace_space_saver
	trace_to
	under_NIS_server
	use_lookaside
	use_nfs_devices
	use_standard_functions
	use_trusted_script
	UseFileCache
	UseNetworkCache
	UseridResolution
	watchdog_refresh

trcfilter.init

The eTrust AC daemon also uses the trcfilter.ini initialization file.

This optional file contains entries that specify filter masks for filtering out eTrust AC trace messages and trace messages that are sent to the audit file (for use with the "trace" audit mode). Each line of the file contains a regular expression. When a message is sent to the trace file, seosd checks whether the message matches one of the entries in the trcfilter.init file. It writes the trace message to the file only if it does not match any of the expressions specified in the trcfilter.init file.

For example, the following trcfilter.init file causes all messages that begin with “INFO” or “WATCHDOG” to be discarded. They are not written to the trace file.

```
WATCHDOG*
INFO*
```

Audit Access Filter: audit.cfg

This optional file offers an additional way to filter, or block, audit messages from seosd. You can supply a filter file that seosd reads during startup, defining audit records that should not be generated. This filter helps to limit the size of the seos.audit file by keeping only the records needed. You can set filtering rules for class name, object name, user name, group name, program name, access rights, and authorization result. Audit filter rules are written in the *eTrustACDir*/etc/audit.cfg file.

Example:Audit Access Filter

In the following example, if root successfully reads a file, seosd does not send a message to the audit file. If root cannot read the file, seosd sends a message to the audit file.

```
FILE;*;root;*;R;P
```

When a message is sent to the audit file, seosd checks whether the message matches one of the following entries in the audit.cfg file:

Field	Rule
Class	The class name should be written in upper case
Object	The name of the resource can be written using a pattern (*)
User	The user name can be written using a pattern (*)
Program	The program being used can be written using a pattern (*)
Access	The access rights must adhere to the rules
Result	The authorization result must be P (permit) or D (denied)

Input	Meaning
Chdir	chdir - change directory
Chgrp	chgrp - change group
Chmod	chmod- change mode
Chown	chown- change owner
Cre	create
Del	delete
Join	join user to group
Kill	kill
Modify	modify
R	read
Rename	mv-change file name
Sec	sec, acl - change acls'
Utime	change time
W	write
X	execute

Note: Audit.cfg cannot filter root login and logout records such as the following:

```
07 Jan 2004 12:43 P LOGIN        root       54  2 _CRONJOB_         ETC_CRON
07 Jan 2004 12:43 O LOGOUT       root       49  2 _CRONJOB_
```

Performance Considerations

Whenever eTrust AC must perform UID to username, GID to groupname, ipaddr to host name, and port to service translations, it may impact eTrust AC performance. How eTrust AC performs these translations depends on the value of certain tokens in the seos.ini file-in particular, the under_NIS_server, use_lookaside, GroupidResolution, HostResolution, ServiceResolution, UseridResolution, and resolve_timeout tokens.

When native operating system mechanisms perform the resolution, the impact on system performance is relatively small. When translating ipaddr to host name, an external mechanism such as DNS must perform the translation. This may result in significant degradation of system performance. This degradation occurs because, while seosd is waiting to receive the host name, all other processes that eTrust AC has intercepted must also wait until seosd completes its processing.

Notes:

- The seosd executable becomes a daemon only if one or both of the following conditions is true:
  - The trace messages are not sent to the screen; that is, you set the trace_to token in the seos.ini file to file, file,stop, or none.
  - You specify any argument *except* -d on the command line when invoking the utility.

  If none of these conditions are true, seosd remains a regular process, connected to the terminal from which you invoked it.
- If you invoke seosd without an argument, seosd runs as a daemon. If you invoke seosd with the -d argument, seosd runs as a daemon, but forces tracing to the trace_file.
- If you invoke the eTrust AC daemon while another copy of seosd is running, the invocation process terminates.
- The following processes are invoked during seosd startup:
  - seagent, the eTrust AC agent daemon.
  - seoswd, the eTrust AC watchdog daemon.

  The eTrust AC daemon is completely initialized only after these daemons are also running. After initialization, these three daemons maintain a type of handshaking protocol to ensure they are all alive and responding. If one of these daemons is found to be absent, one of the other two daemons automatically restarts it.
- If you set the value of the under_NIS_server token to no, seosd allows UNIX to translate UID, GID, IP addresses, and port numbers by taking data from the following sources:

Type of Station	Source
Stand-alone	Seosd uses the following files for translations; ■ /etc/passwd for UID to user name ■ /etc/group for GID to group name ■ /etc/hosts for IP address to host name ■ /etc/services for service ports to service names

Type of Station	Source
	The source of the information varies, depending on the operating system and its version number. The information is usually taken from /etc files and the NIS server. However, in some systems, the /etc files are not the source and the order in which translation is made is changed during system configuration. For instance, in the Solaris 2.*x* system the file /etc/nsswitch.conf determines the translation order.
DNS client	Translation for users, groups, and services is performed using /etc files. Host names are translated by calls to the DNS server and, on some systems, the /etc/hosts file is also read.
NIS and DNS clients	The ipaddr to host name translation is performed by DNS. For user, group, and service translations, the translations are performed in the same way as NIS client translations.

- If you set the value of the under_NIS_server token to yes, seosd performs its own translations. If seosd caches data for its translations, the sources of its data are as follows:

Type of Station	Source
NIS server	The server machine usually behaves as both server and client, and consults the NIS server daemon for any type of translation. The files which contain the sources of the NIS resolution maps are usually located in /var/yp, but the location may vary, depending on the site configuration, and the type and version of the operating system.
DNS server	The source of the information used for translation depends on the configuration of the site. DNS does not have an option to scan its resolution database; therefore, eTrust AC cannot use caching, and must use a lookaside database. You must configure the lookaside database so that the utility sebuildla uses a host list file. For more information, see sebuildla in this chapter.
all others	Same as DNS server.

In versions 2 and higher of eTrust AC, seosd can also use the tokens GroupidResolution, HostResolution, ServiceResolution, UseridResolution, and resolve_timeout to control the translation process. For more information on these tokens, see the *Administrator Guide*.

See Also

You can expand the functionality of the seosd daemon by writing programs at your site that use the APIs provided with eTrust AC. For more information, see the *SDK Developer Guide*.

seostngd

The eTrust AC synchronization daemon for Unicenter TNG.

Unicenter Security and eTrust AC together manage the administration of your enterprise IT environment before total migration occurs. To reduce the complexity of using different product tools to perform administrative tasks, we are providing a synchronization daemon.

This daemon is called seostngd. eTrust AC sends Policy Model database (PMDB) updates through CA Common Communication Interface (CAICCI) to seostngd. The daemon listens for updates on CAICCI and then translates the messages into equivalent cautil commands to update the Unicenter Security database with this global data.

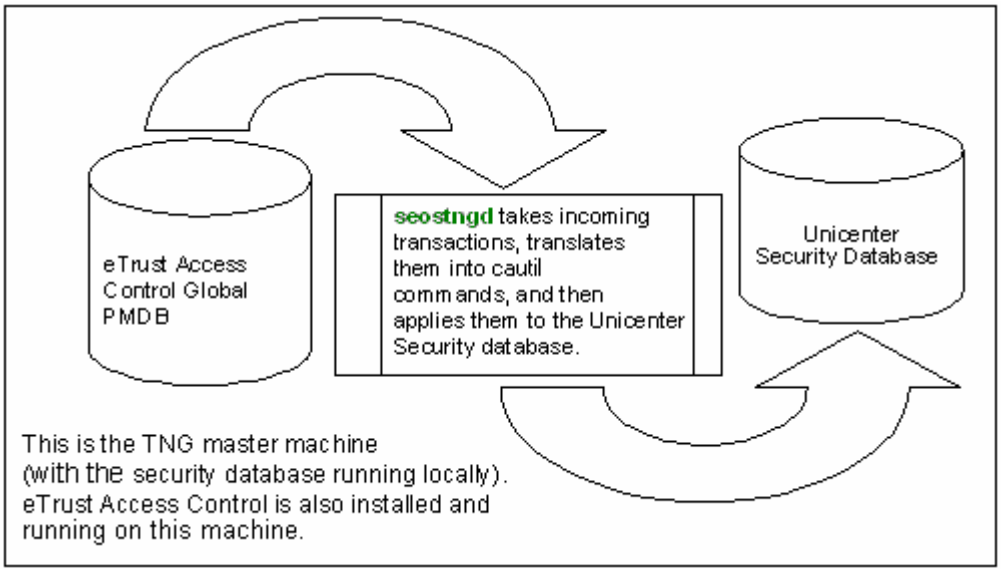
Current Unicenter TNG processing can still update other Unicenter TNG client installations. You must run seostngd on the same machine as the Unicenter Security database (normally referred to as the Unicenter master machine.) eTrust AC should also be running on the same machine.

Syntax

```
seostngd
seostngd -stop
```

How seostngd Works

The following figure demonstrates the function of seostngd:



The major task of seostngd is to take changes you make on eTrust AC (using either the Windows-based or UNIX-based GUIs or selang commands) and apply them to the Unicenter Security database. If the changes can be applied to the Unicenter Security database successfully, you will see the same behaviors as you did on eTrust AC.

For example, when you create a USER object with eTrust AC, you should see the same USER object being created in Unicenter TNG (as long as the required fields are present).

SEOSTNGD Limitations

Different maximum field lengths between eTrust AC and Unicenter Security can cause truncation of data values. The following table lists the significant differences in supported field lengths.

Field	Unicenter TNG	eTrust AC
User ID	20 characters	256 characters
Password	8 characters	14 characters
User group ID	8 characters	254 characters
Asset group ID	8 characters	255 characters

- Unicenter TNG does not support renaming a user or asset object, so the seostngd daemon ignores eTrust AC “rename” commands for users and assets.
- Unicenter TNG user groups and asset groups must have at least one member. If no members are specified in the eTrust AC command for creating user groups or asset groups, the corresponding Unicenter TNG user group or asset group is not created until at least one member is added.
- If the last member is removed from an eTrust AC user group or asset group, that user group or asset group is removed from Unicenter TNG.
- Unicenter TNG assets require at least one defined accessor, so a new Unicenter TNG asset is not created until at least one eTrust AC “authorize” command is executed for the asset.
- eTrust AC removes any associated rules for an object when it is deleted; however, Unicenter Security does not.
- eTrust AC users have an admin attribute that has a similar meaning as when a Unicenter TNG user is a member of the SSF_AUTH user list in the Security Options. However, Unicenter TNG does not provide any automatic way for manipulating remote Security Options, so manual modifications to SSF_AUTH user list are required.

- In order to force the creation of a new Unicenter TNG user, the new eTrust AC user must be created in the Native environment with a value supplied for the eTrust AC password field. Unicenter TNG has default password restrictions that require a minimum length of six characters-two must be alphabetic and one must be numeric.
- The eTrust AC “authorize” command supports “*” as an accessor ID, but this is not supported in Unicenter TNG. The seostngd daemon ignores eTrust AC “authorize” commands like this.
- The eTrust AC “authorize” command supports conditional access rules using the “via” parameter, but this is not supported in Unicenter TNG. The seostngd daemon ignores eTrust AC “authorize” commands like this.
- If the “access” parameter is not specified on an eTrust AC “authorize” command, the seostngd daemon grants READ permission for any UNIX-FILE or Unicenter TNG asset group, and grants all permissions for any other Unicenter TNG predefined asset type.

seoswd

The eTrust AC watchdog daemon.

The watchdog (seoswd) monitors the file information and digital signatures of programs that are defined in the database as trusted programs. Monitoring is performed in the background with a minimal load on the system. The eTrust AC agent daemon seagent automatically starts seoswd.

The seoswd daemon performs the following functions:

- It monitors the programs that you defined in the PROGRAM class of the database. If the watchdog detects that a program was modified, it notifies the eTrust AC daemon, seosd, which marks the program as untrusted. The seosd daemon does not allow an untrusted program to run. The seosd daemon also marks the program's status change to untrusted in the database and creates an audit record.
- It monitors files that are defined as secured files. These files are defined in the SECFILE class in the database.
- It monitors seosd to ensure it is running. If the watchdog detects a problem with seosd, it automatically restarts it.
- The seoswd daemon uses the system log syslogd to notify the security administrators when it detects that seosd has stopped responding. All system log messages are submitted as AUTH facility. For more information on the system log facility, see your system man pages under the syslogd and syslog.conf sections.
- It reports several events to eTrust AC, and creates audit records for programs and secured files that were found to be altered.
- It allows you to specify interval and fixed scanning schedules for trusted programs and secure files.
- The watchdog ignores any signal except SIGHUP; you cannot kill the seoswd daemon unless you first shut down seosd. However, if you execute the command kill -SIGHUP *pid*, the watchdog scans all trusted programs and secure files in the database.

There are two ways in which you can set up the Watchdog scanning mechanism:

1. Determine a start time and then repeat scans at a given interval.

   For example, when checking trusted programs, the Watchdog will start the first scan at *PgmTestStartTime* and will check all the trusted programs. Rescanning will take place *PgmTestInterval* seconds after the beginning of the previous scan.

2. Scan at given times.

Note: In both cases, the Watchdog will sleep periodically for a predetermined rest period *(PgmRest* seconds) during each scan. The Watchdog rests in order to prevent system overload.

You can choose to use one mechanism or both simultaneously. For example, starting at 12:00, scan every 4 hours as well as at 13:00 and 17:30.

In addition to the above mentioned mechanisms for routine scanning of the trusted programs and secured files, there is a way to perform a one-time scan on demand by sending a HUP signal (see token SignalMinInterval).

To learn more about defining trusted programs to eTrust AC, see the *Reference Guide*.

If you invoke seoswd without an argument, it runs as a daemon. If you invoke seoswd with the -d argument, it runs as a daemon, but displays all debug information on the terminal from which you invoked it.

Syntax

```
seoswd [-d]
```

The seos.ini File

The seoswd utility uses the following tokens in the seos.ini file:

Section	Token
seoswd	IgnoreScanInterval
seoswd	PgmRest
seoswd	PgmTestInterval
seoswd	PgmTestStartTime
seoswd	PgmTestTime
seoswd	RefreshParams
seoswd	SecFileRest
seoswd	SecFileTestInterval
seoswd	SecFileTestStartTime
seoswd	SecFileTestTime
seoswd	SeosAYT
seoswd	SignalMinInterval
seoswd	UnTrustMissing
seos	SEOSPATH

Other Files

No other special files are used.

Note: The seosd daemon automatically starts the seagent daemon, which in turn automatically starts the seoswd daemon.

See Also

- The seaudit, seerrlog, selang, seretrust, seosd, and seagent utilities in this chapter.
- The chapter “The Audit Browser: seauditx” in the *User Guide.*

sepass

Sets a new password or replaces an existing password in the local host, in a Policy Model, or in the NIS or NIS+ server, as applicable.

The sepass utility changes the user password. Additionally, privileged users can use sepass to change the passwords of other users. When changing your own password, sepass prompts you for your old password.

Note: If seosd is not running, sepass runs a default password program. The DefaultPasswdCmd token in the passwd section of the seos.ini file specifies the default password program.

When changing the password of another user, sepass prompts you for your own password or the password of the user whose password you are changing. In both cases, sepass then prompts for the new password.

- If you enabled eTrust AC password checking, sepass checks whether the new password complies with the password rules set in the database. If the new password passes this quality check, the user is again prompted for the new password.
- If you disabled password checking, the user is immediately re-prompted for the new password.

  After the user enters the new password for the second time, sepass compares the two copies of the new password. If the copies are not identical, the user is prompted again for the new password.

  If the two new passwords are identical, sepass updates the password as follows:
- The local host password files-/etc/passwd and any security files such as the password shadow file-and the local database are updated, if the user is defined in them.

  If the token nis_env in the [passwd] section of the seos.ini file has a value (either *nis* or *nisplus*), the NIS or NIS+ server is updated. When a password is set on a master NIS server, the NIS password map is automatically reconstructed.

Syntax

```
sepass [-d][-h][-l][-p][-s policy_model@hostname] [-g number] [-x] [userName]
```

Switches

-d

Instructs sepass to display all the information it has regarding the password update, such as on which stations the update succeeded and if you did not activate setoptions class+(PASSWORD), that the password's quality was not checked. This switch is useful when debugging.

-g *number*

Defines the number of grace logins for *userName*.

-h

Displays the help screen.

-l

Instructs sepass to replace the password only on the local station; that is, in the local password file (usually /etc/passwd), security files, and the local database.

In the NIS/NIS+ environments, users are not usually defined in the /etc/passwd file of the client; therefore, the password on the client station is not updated.

In NIS/NIS+ server statins, the password is updated locally and propagated by NIS/NIS+.

This switch and the -p and -s switches are mutually exclusive.

-p

Instructs sepass to change the password only on the remote station and on the PMDB at the host specified in the switch. This switch and the -l and -s switches are mutually exclusive.

-s *policy_model@hostname*

Instructs sepass to replace the password on the local station and on the PMDB at the host specified in the switch. This switch and the -l and -p switches are mutually exclusive.

-x

Instructs sepass to replace the password as if changed by the user *username*. This switch updates the time and date of the last change in the database. Grace logins are terminated.

Note: To let you change the root password as if changed by root, you have to set the RootPwAsOwn appropriately. For more information about seos.ini tokens, see the *Reference Guide*.

[*username*]

(Optional) Defines the name of the user whose password sepass changes. If you omit this option, your own password is set.

Connecting a User to a Policy Model

When sepass modifies a password, the change propagates to the appropriate Policy Model and its subscribers, in both the UNIX environment and the database. The -s switch specifies the PMDB. For example:

```
-s policy_model@hostname.
```

If you do not include the -s switch, the user can nevertheless be identified with a Policy Model in one of the following ways:

- The Policy Model is specified in the user record.

  If the parameter pmdb is set in the user record of the user whose password is being changed, then the PMDB specified in the pmdb parameter is updated, which then propagates the update to its subscribers both in the UNIX environment and the database.

- The user is associated with a profile group and that group has a Policy Model.
- The passwd_pmd token is set in the [seos] section of the seos.ini file. The user is then identified with the Policy Model indicated.
- The parent_pmd is set in the [seos] section of the seos.ini file. The value of that token identifies the user to a specific Policy Model.
- The secondary_pmd token is set in the seos.ini file. This secondary PMDB is used when users are not defined in the passwd PMDB (or parent PMDB, if you did not set the passwd_pmd token). Administrators cannot set user passwords in the secondary PMDB, though users can set their own.

See the *Administrator Guide*, for more information about the seos.ini tokens.

Notes:

- Switches -l, -s and -p are, in effect, mutually exclusive. If you include them simultaneously, the password is only replaced locally.
- To change the root password in the PMDB, root must be defined to the native PMDB. For example:

```
eu root native
Root Password Propagation
```

An ADMIN user (or root) can propagate the root password to a PMDB using the sepass -p or -s options. The PMDB then propagates the password to its subscribers.

Note: To propagate root password changes, you must set the AllowRootProp token in the [passwd] section of the seos.ini file to yes. You must also set the AllowedUidRange to a lower limit of 1 (by default the lower limit is 100 and the upper limit is 30000).

Privileged User

Privileged users can change passwords of other users. The definition in sepass of privileged users and the passwords they can change follows.

- A user with the ADMIN or PWMANAGER attribute can change the password of any user; however, if the nochngpass property is set to yes, a PWMANAGER cannot change the password.
- A user with the GROUP-ADMIN or GROUP-PWMANAGER attribute can change the password of any user within the scope of the group-the user records that are owned by the group.

- A user with MODIFY or PASSWORD authority in the access control list (ACL) of the USER record in the ADMIN class can change the password of any user.
- The owner of a user record can change the password of the user who is defined by the record.

Password Synchronization

Normally, all the passwords of a user on the various subscribers and PMDBs in a particular hierarchy are identical. However, it is possible that they are not identical-that the password for a particular user on a subscriber station is different from the user's password in the PMDB. When this happens, the ordinary user and privileged users have particular problems when they want to update the password.

- If users-whether or not they are privileged-attempt to change **their own** passwords, and the passwords on the local station and the specified PMDB are **not** identical, sepass first changes the passwords on the local station. It then sends a message informing the users that the passwords were successfully changed locally, but that the change on the PMDB failed.

  However, if you set the pass_auth token in the seos.ini file to no, sepass does not compare passwords entered by users with the passwords stored in remote stations.
- If privileged users attempt to change the passwords of **other** users, and the passwords on the local station and the specified PMDB are **not** identical, sepass changes the passwords on the local station and the specified PMDB. The PMDB then propagates the new password to **all** of its subscribers.

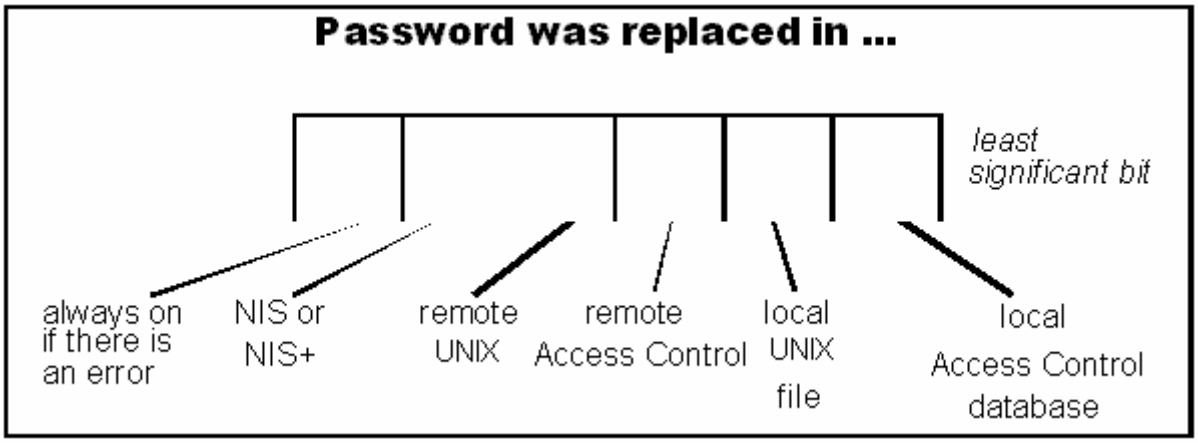
NIS/NIS+

In NIS and NIS+ environments:

- A user with the ADMIN attribute can only change the password of another user on the local station, because the administrator may not have the ADMIN attribute on the remote host.
- Users are not usually defined in the /etc/passwd file of the client; therefore, the -l switch has no effect.
- Users are usually defined in the /etc/passwd file of the server. The sepass utility updates the /etc/passwd file, the NIS maps, and NIS/NIS+ propagates the new password to its clients.

Return Value

On success, the utility returns 0. If password replacement does not entirely succeed, the utility returns a nonzero code that indicates where the password was replaced. The code is as shown in the following diagram.



To ensure that the value is nonzero for errors, the eighth bit is always set. If sepass fails to change the password anywhere, the return value is 0x80.

If the old password is wrong, a message is printed to the TTY monitor and the eTrust AC audit trail.

Setting Password Quality Rules

By default, eTrust AC does not check a new password against the password quality rules. To activate password quality checking, invoke selang and enter the following command:

```
setoptions class+(PASSWORD)
```

Now, every time a user's password is changed, the new password is checked against the password quality rules set in the database.

To view the current set of password rules, invoke selang and enter the following command:

```
setoptions list
```

The set of password quality rules contains the following parameters:

interval

Maximum life of password in days.

min_life

The minimum number of days between password changes.

History

Number of old passwords to store for each user.

Length

Minimum password length (number of characters).

Alpha

Minimum number of alphabetic characters, such as a,b,A,B.

Alphanum

Minimum number of alphanumeric characters, such as a,B,1.

bidirectional

Enables complex password checking. Bidirectional encryption means new password cannot contain or be contained by previous passwords. Passwords are compared regardless of case sensitivity.

prohibited

Limits the characters that users can enter in passwords by specifying prohibited characters.

Numeric

Minimum number of numeric characters, such as 1,2,3.

Lowercase

Minimum number of lowercase characters, such as a,b,c.

Uppercase

Minimum number of uppercase characters, such as A,B,C.

Use_dbdict

Defines the location of the dictionary for password checking. Use use_dbdict to set to check the password against the eTrust AC database. Use use_dbdict- to check the password against an external file.

Special

Minimum number of special characters, such as $,%,^.

Max_rep

Maximum number of repetitive characters, such as aaa, bbbb.

Grace

Number of logins allowed after a password has expired.

oldpwchk

The new password is neither contained in, nor contains, the password being changed.

namechk

The new password is neither contained in, nor contains, user's name.

To set the system-wide password history value to 12, invoke selang and enter:

```
eTrustAC> setoptions password(history(12))
```

To set the system-wide password interval to 30 days, invoke selang and enter:

```
eTrustAC> setoptions password(interval(30))
```

To define the location of the password dictionary to the **database**, enter:

```
eTrustAC> setoptions password(rules(use_dbdict))
```

To define the location of the password dictionary to a **file**, enter:

```
eTrustAC> setoptions password(rules(use_dbdict-))
```

To set any other password rule, invoke selang and enter:

```
eTrustAC> setoptions password(rules(length(..) numeric(..) ..))
```

Note: You can set more than one rule in a setoptions command.

To deactivate password checking, invoke selang and enter:

```
eTrustAC> setoptions password(rules-)
```

Defining a Password Dictionary

The password file is a target for *cracking utilities*. Cracking utilities use dictionaries and encrypt the words in those dictionaries to find matches in the password file. To avoid such an attack, we recommend that you provide a custom dictionary that contains a list of words that are not used as passwords.

You can define dictionary checks in two ways:

- Against the eTrust AC database

  To set dictionary checking to the database, follow these steps:

  a. Activate the DICTIONARY class with the following selang command:

  ```
  setoptions class+(DICTIONARY)
  ```

  b. Configure eTrust AC to use a dictionary in the database with the following selang command:

  ```
  setoptions password(rules(use_dbdict))
  ```

Note: You can perform these steps automatically by using the DictImport utility. The DictImport utility imports an external dictionary to the database for password checks. For more information, see the *Utilities Guide*.

- Against an external file

  To check passwords against an external file, follow these steps:

  a. Set the UseDict token in the [passwd] section of the seos.ini file to yes.

  b. Set the Dictionary token in the [passwd] section of the seos.ini file to the name of the dictionary file. Usually, the dictionary file is called /usr/dict/words. To find the exact location of this file in your system, check the man page for the UNIX spell utility.

  c. Enter the following selang command:

  ```
  setoptions password(rule(use_dbdict-)
  ```

  Subsequently, a new password cannot be a word that appears in the dictionary file.

The seos.ini File

The sepass utility uses the following tokens in the seos.ini file:

Section	Token
passwd	AllowRootProp Check_Adm_Rules Dictionary DefaultPasswdCmd GeneratePasswd nis_env NisPlus_server only_pmdb quite_mode UseDict
pmd	pass_auth
seos	parent_pmd passwd_pmd secondary_pmd

Other Files

The sepass utility uses the following additional special files:

- /etc/passwd
- On IBM AIX platforms:
  - /etc/security/limits
  - /etc/security/passwd
  - /etc/security/user
- /etc/shadow on Sun Solaris platforms
- /.secure/etc/passwd on HP-UX 9.x platforms
- TCB files on HP-UX 10.x configured to use TCB.

Notes:

- eTrust AC must be installed and running on your system before you can use sepass.
- If you do not supply the *user-name* parameter, your own user name is assumed.
- Passwords are stored and transferred over the network in an encrypted format.
- When a user changes another user's password, sepass checks whether the user who entered the sepass command is authorized to change the specified user's password. If the user has such authority, sepass sets the new password without first checking the quality of the new password. This is called an admin change of the password. In addition, sepass sets the number of grace logins for the user whose password was changed to one. Thus, the user whose password was changed can log in only once, and must set a new password during that session.

  When you set the token Check_Adm_Rules in the [passwd] section of the seos.ini file to yes, password changes performed by admin users **must** use password rules.

  The new password specified by the user is subject to password quality checking and, when sepass sets the new password, it also sets the user's grace logins setting in accordance with the rules defined in the database. For more information on the number of grace logins remaining for a user, see the segrace utility in this chapter. The sepass utility does not use the UNIX admin change option (that is, the UNIX automatic password expiration feature).
- To be able to propagate passwords, you must have the authority to change a password on the PMDB as described previously.

Examples

The sepass utility must work in a variety of environments. Here are some guidelines on how to use sepass in various situations:

- To change your own password on the local host, enter the command:

  ```
  sepass -l
  ```

  Note: If no PMDB is defined at the site, you can omit the -l switch. If a PMDB is in use at the site, omitting the -l switch changes your password on all subscriber databases of the PMDB. In an NIS/NIS+ client, this switch does not change the password; in an NIS/NIS+ server, the password is changed and then propagated.

- To change the password of any user other than you own, on the local host only, enter the command:

  ```
  sepass -l username
  ```

  username must exist in the /etc/passwd file, the appropriate UNIX security files, and the database.

  In an NIS/NIS+ client, sepass does not change the password. In an NIS/NIS+ server, the password is changed and then propagated

- To change the password of a user on several stations at a site where NIS is not in use, proceed as follows:

  a. Create a PMDB. For more information, see the *Administrator Guide*.

  b. Add all the users whose details must be distributed to the subscriber computers, to both the UNIX and the eTrust AC environments of the PMDB.

  c. Subscribe all the stations to receive the updated passwords to the PMDB.

  d. On every subscriber, set the tokens in the [seos] section of the seos.ini file to the names of your PMDB. For example:

  ```
  passwd_pmd = PMD1@morocco
  parent_pmd = PMD1@casablanca
  ```

  e. Enter the command:

  ```
  sepass username
  ```

  When sepass completes execution, the user's password is changed on all the subscriber databases.

See Also

You can expand the functionality of the sepass utility by writing programs at your site that use the APIs provided with eTrust AC. For more information, see the *SDK Developer Guide*.

sepmd

Administers a PMDB. It handles several tasks, each described separately in this section:

- Administer subscribers
- Truncate the update file
- Administer Dual Control
- Manage the Policy Model log file
- Other administration

Note: You must run the sepmd utility on the host where the Policy Model resides.

Administering Subscribers

Syntax

sepmd *options parameters*

Options

-n

Creates a new subscriber and then updates it retroactively to the Policy Model. For general rules that apply for updating a subscriber, see the description for the -s option. This option sends the contents of the entire PMDB-including the LOGINAPPL and SPECIALPGM objects-to the new subscriber. You may want to filter out these objects if the subscriber's LOGINAPPL and SPECIALPGM objects differ from those of the parent.

Note: For more information, see the *Administrator Guide*.

If the send_unix_env token in the seos.ini file is set to yes, the -n option also sends the contents of Policy Model password and group files. We recommended that you view the database, by using dbmgr -export -l, to ascertain the commands being forwarded.

A subscriber added with -n is marked as 'sync', indicating that it is now in synchronization mode, and receives all of the PMDB rules. When the subscriber has received all the rules, it is released from synchronization mode, and becomes a regular subscriber. The -n option may take some time to process. If there are multiple or contradictory updates, the last one is used.

Important! When you subscribe an eTrust AC end-point or a PMDB to another PMDB using *sepmd -n*, the new parent PMDB should not contain any policies (POLICY object names) that already exist in the new subscriber. You must undeploy each existing policy from the subscriber and then delete the POLICY object and linked RULESET object from the subscriber before you subscribe it to the new parent PMDB.

-r

Removes the subscriber from the list of unavailable subscribers maintained by the sepmdd utility, thus making the subscriber available for immediate updates. Normally, if a subscriber is down and cannot receive updates from the Policy Model, sepmdd tries to send updates to that subscriber only after a certain period of time, determined by the _retry_timeout_ token in the seos.ini file. However, if you use the -r parameter, sepmdd skips the waiting period and tries to send updates to the subscriber immediately.

-s

Assigns a subscriber to the Policy Model. When you subscribe a host to a Policy Model, the host must be up, and eTrust AC must be running on that host. Additionally, the PMDB must be the parent PMDB of the subscribed host. You establish this relationship with the parent_pmd token in the subscriber's seos.ini file, which must contain the name of the PMDB to which the host is being subscribed.

This option allows you to add a subscriber in the middle of the update file, so that the newly added subscriber receives updates retroactively from the subscription offset forward.

Use the -C option to see the valid update offsets (see Other PMDB Administration). If the specified offset is in the middle of an update, the offset is moved forward to the beginning of the next update. If the offset is invalid (because it is smaller than the first offset or larger than the last), an error message appears.

When you subscribe a Policy Model to another Policy Model,

- the token parent_pmd in the pmd.ini file of the subscribed Policy Model must contain the name of the Policy Model to which it is subscribing (its parent Policy Model).
- eTrust AC must be running on the host in which the subscribed policy resides.

A PMDB should have only one parent. If you decide to establish a Policy Model with more than one parent give the parent_pmd token the name of a file containing a list of the parent Policy Models. However, establishing more than one parent is not recommended because you risk inundating your database with unreliable instructions from multiple sources.

-sm

Assigns a mainframe subscriber to the Policy Model.

-u

Removes a subscriber from the Policy Model subscription list.

Parameters

policyModel

The name of the Policy Model.

subscriber

The subscriber station or the host of the subscriber PMDB.

Truncating the Update File

Syntax

```
sepmd option parameters
```

Options

-t

Deletes entries from the update file. If the force_auto_truncate token in the pmd.ini file is set to no, *sepmd -t* does not truncate the update file. If the token is set to yes, *sepmd -t* truncates the update file even if there are no subscribers to the Policy Model.

If you are using offset (manual cutting), you can find the offset by running sepmd with the -L parameter.

Note: You must use the true offset provided in the -L parameter to truncate the file, and not an offset derived by subtracting from the start offset.

If you are using auto, sepmd calculates the offset of the first unpropagated entry and deletes all the entries before it. Using auto saves the step of running the utility with the -L parameter.

If a subscriber received fewer than all updates before the specified offset, sepmd displays an error message and does not truncate the file. If you want to truncate the file anyway, do the following:

- Unsubscribe the host that was not updated
- Truncate the file
- Resubscribe the host to the Policy Model

If you do this, the subscriber fails to receive one or more updates from the Policy Model. The subscriber's offset changes to the last offset of the updates file.

Parameters

policyModel

Specifies the name of the Policy Model.

auto

Instructs sepmd to calculate the offset of the first unpropagated entry and to delete all the entries before it.

offset

Specifies the distance from the beginning of the update file to the position of a particular subscriber. Offsets appear in hex format.

Dual Control

Dual Control is a method of operating on PMDBs that divides the process of executing transactions (that is, executing one or more commands) into two stages: creating the transaction, and checking the transaction before executing it.

In the first stage, a *Maker* creates a transaction that is placed in a file. The transaction stays in the file until it is authorized or rejected. As long as the transaction is in the file, the Maker can retrieve it, change the commands, or delete it. the sepmd utility gives each transaction a unique ID number when it is created.

In the second stage, a *Checker*-any administrator **except** the user who created the commands-processed the transactions (authorizes or rejects). If the transaction is authorized, the commands are executed and the PMDB is changed accordingly. If the Checker rejects the transaction, the commands are deleted and the PMDB is **not** changed. The Checker cannot authorize some of the commands in a transaction and reject others; the transaction must be processed as a whole.

When you use Dual Control, the name of the PMDB must be “maker”. The is_maker_checker token must have the value yes in the pmd.ini file, and the [pmd] section of the seos.ini file. For more information about the seos.ini and pmd.ini files, see the *Administrator Guide*.

Syntax

```
sepmd options
```

Options

-m d *transactionID*

Deletes the transaction. A transaction is one or more commands that must be approved before they are implemented on the PMDB. Only the user who created the transaction can delete it.

transacationID is the unique identifying number that sepmdd gives to the transaction when it is created.

-m l

Lists the unprocessed transactions (awaiting the Checker) of the user who invoked the command. Each transaction is listed with its ID number, the name of its Maker (the user who created the transaction-in this case the same user who invoked the command), and its description, if any.

-m la

Lists all the unprocessed transactions of all the Makers. Each transaction is listed with its ID number, the name of its Maker, and its description, if any.

-m lo

Lists the unprocessed transactions (awaiting the checker) of all the Makers *except* the transactions of the user who invoked the command.

-m p *transactionID code*

Processes a transaction. When the Checker (any admin user *except* the Maker who created the transaction) enters an ID number, all the commands in the specified transaction appear in a list.

transacationID is the unique identifying number that sepmdd gives to the transaction when it is created.

By entering the appropriate *code,* the Checker can do the following:

– 0-Reject the transaction, in which case all the commands in the transaction are deleted and no changes are implemented in the PMDB

– 1-Authorize the transaction, in which case the commands are immediately implemented in the PMDB

– 2-Unlock the transaction so that it can be processed later, or by a different Checker.

This parameter does not work in the following circumstances:

- If one or more of the commands in the transaction pertain to the user who invoked the command
- If the transaction is locked by a different Checker
- If the transaction was created by the user who invoked the command-Makers cannot act as Checkers for their own transactions
- If the specified transaction ID does not exist
- If the user who invokes the command does not have the authority to be a Checker

-m r *transactionId*

Retrieves or locks a transaction.

- If you are the user who created the transaction (the Maker) this parameter retrieves a specific, unprocessed transaction. After you retrieve the transaction, you can direct it to an appropriate file and use the ASCII editor of your choice (vi, emacs, and so on) to update the transaction.
- If you are a user who is **not** the Maker (Checker) this parameter locks the transaction prior to processing. You cannot change a locked transaction.

transacationID is the unique identifying number that sepmdd gives to the transaction when it is created.

Managing the Policy Model Log File

Syntax

sepmd *options parameters*

Options

-cl

Clears the contents of the Policy Model log file.

-dl

Displays the Policy Model log file.

-kl

Makes the Policy Model log file unavailable.

-sl

Makes the Policy Model log file available. This option is only useful if the pmd_log_level is set to a value of 1 or 2.

Parameters

policyModel

The name of the Policy Model.

The Policy Model log file provides a detailed audit trail of Policy Model data base activities. For example:

```
    Wed Nov  4 10:08:02 2003 pmdb1:Processing list request for
missouri.yourco.com
    Wed Nov  4 10:08:02 2003 pmdb1:Processing list request for
oregon.yourco.com
    Wed Nov  4 10:09:14 2003 pmdb1:Empty request
    Wed Nov  4 10:09:15 2003 pmdb1:Processing shutdown request
    Wed Nov  4 10:09:15 2003 pmdb1:Delete filters
    Wed Nov  4 10:10:04 2003 pmdb1:Opened error logs
    Wed Nov  4 10:10:04 2003 pmdb1:Try to load filters
    Wed Nov  4 10:10:04 2003 pmdb1:Filters file : nis_filter.dat
```

Running the sepmdd daemon for the first time automatically creates the Policy Model log file. The pmd.ini file contains the token pmd_log_level that you can set to one of the following values:

- **0** - Do not log any entries.
- **1** - List only error messages.
- **2** - List error and informational messages (default value).

A warning message in the log file tells you if you have exceeded file size limitations. The max_log_size token in the pmd.ini file permits you to add more memory.

Other PMDB Administration

Syntax

```
sepmd options policyModel
```

Options

-c

Clears the Policy Model error log.

-C

Displays all commands in the update file, and their offsets. The offset indicates the location of the update inside the file. Commands in the UNIX environment appear with the prefix “UNIX”.

-de

Decrypts the information in the encrypted updates.dat file. Data encryption for this file occurs when you set the UseEncryption token, in the [pmd] section of the pmd.ini file, to yes.

-e

Displays the Policy Model error log.

-k

Shuts down the Policy Model daemon safely. Do not use the kill command to shut down the Policy Model daemon.

-l

Lists the subscribers of the Policy Model.

-L

Lists the subscribers of the Policy Model and their status, including number of errors, availability, offset, synchronization mode, and the next command to be propagated. The update file contains all updates that must be, or have been, propagated by the Policy Model. The offset indicates the location of the next update that must be sent to a subscriber. Both initial and latest offsets also appear.

-ri

Reloads the Policy Model's pmd.ini file and the seos.ini file while the daemon is running. This enables the daemon to register configuration changes in certain tokens. You can only use this option at intervals of one minute or more. The -ri option checks configuration changes in the following tokens: parent_pmd, _retry_timeout_, _min_retries_, and _shutoff_time_.

-S

Starts the daemon. Use this option to start the daemon when you do not have any other commands to execute.

Parameters

policyModel

The name of the Policy Model.

Authorization

You must have admin authority in the Policy Model to use sepmd for starting or querying the Policy Model.

Files

sepmd also uses the following files:

- updates.dat - updates file
- ERROR_LOG - error log

See Also

- Setting Up a PMDB and Dual Control in the chapter “Managing the Policy Model Database” in the *Administrator Guide.*
- The sepmdadm, sepmdd, and sesudo utilities in this chapter.

sepmdadm

Creates the definitions needed to run a PMDB.

The sepmdadm utility is a script consisting of the eTrust AC and UNIX commands required to define a PMBD, to define the relationship of the PMDB to PMDBs above and below it, and to define its subscriber stations. By default, the user root is defined as the administrator and auditor of the PMDB. You must run the sepmdadm utility locally, although you can also run it through a remote shell. When you use sepmdadm to create a new PMDB, you probably want to follow up by pointing subscribers to the PMDB (see Pointing Subscriber Stations to the PMDB) and by synchronizing the UIDs and GIDs (see UID/GID Synchronization in the *Administrator Guide*).

You can run this utility in either a interactive or non-interactive mode:

- In non-interactive mode, you enter arguments in the command line. The utility builds the PMDB and its hierarchy according to the values it receives.
- In interactive mode, you do not enter arguments in the command line. The sepmdadm utility asks the user if the desired mode is interactive. If the user answers “y,” then the utility proceeds to ask the user for option values.

When creating a new PMDB with sepmdadm, you identify the stations that are the subscribers of the Policy Model. However, you must also update the parent_pmd token in each subscriber's seos.ini file with the name of the PMDB to which you have subscribed the station. If you do not do this, the subscribers do not accept updates from the PMDB.

By subscribing several stations to the same PMDB, and by subscribing one PMDB station to another, you can create a hierarchy of PMDBs.

Syntax

```
sepmdadm options
```

Options

-admin *name*

Specifies the administrator of the PMDB.

-auditor *name*

Specifies the auditor of the PMDB.

-c | -clean

Removes the Policy Model. This option shuts down the daemon, removes the file protections from the database, and deletes the Policy Model directory with all its contents. You cannot use this option with the -nonconfirm option.

-desktop *hostname*

Specifies a station from which the administrators can administer PMDBs located on the local host. If you do not specify any stations, the administrators can only administer the PMDBs from the local host.

-h | -help

Displays the help screen.

-i | -interactive

Runs sepmdadm in interactive mode.

-nis | -NIS

Performs NIS setup on the Policy Model. You must use this option if the PMDB is installed on a NIS server.

-noconfirm

Specifies that the user is not asked to confirm answers. This option is useful when invoking sepmdadm from within a shell script in non-interactive mode.

-parentpmd *pmdbName*

Specifies the name of the parent PMDB to which this PMDB is subscribed. If you use this parameter with the -subsconfig parameter, sepmdadm updates the parent_pmd token in the seos.ini file. If you use this parameter without the --subsconfig parameter, sepmdadm updates the parent_pmd token in the pmd.ini file.

-passwdpmd *pmdbName*

Specifies the PMDB to which sepass sends password updates. This option updates the passwd_pmd token in the [seos] section of the seos.ini file.

Note: You can use this parameter only when you also use the -subsconfig parameter.

When creating a multi-level Policy Model, set this parameter to the PMDB at the top of the pyramid, so that password changes can be propagated to all levels in the PMDB system.

-pmdname *name*

Specifies the name of the PMDB to be created.

-pwmanager *name*

Specifies the password manager of the PMDB.

-seosdir *directory*

Specifies the directory in which eTrust AC is installed. Use this option only if eTrust AC is not installed in the default directory.

-subsconfig

Specifies that the local station is a subscriber. When using this parameter, you must specify the parameters -parentpmd *pmdbName* and -passwdpmd *pmdbName* to update the relevant tokens in the seos.ini file.

Note: The parameters should follow the -subsconfig option when configuring a subscriber.

-subscriber *name*

Specifies subscribers of this PMDB. They can be PMDBs or stations.

Examples

Following are examples of interactive and non-interactive methods of creating a PMDB and linking subscriber stations to it.

Creating the Policy Model Database

Suppose you have a station called bigcentral, where you want to maintain a PMDB for other stations to subscribe to. To create the PMDB at bigcentral, run sepmdadm there. This utility is located in the directory *eTrustACDir*/bin.

Using Only the Command Line

To create a PMDB at bigcentral named pmdb1 with workstat1 and workstat2 as subscribers and adm1 and adm2 as administrators, use the following command at bigcentral:

```
bigcentral # /opt/CA/eTrustAccessControl/bin/sepmdadm --pmdname pmdb1\
             --subscriber workstat1\
             --subscriber workstat2\
             --admin adm1\
             --admin adm2
```

Using a Dialog

Instead of the command options shown in the previous example, you can provide answers to sepmdadm prompts. After each answer, press Enter.

1. At bigcentral, issue the sepmdadm command with the -i parameter:
2. bigcentral # /opt/CA/eTrustAccessControl/bin/sepmdadm -i
3. The sepmdadm utility asks for the name of the Policy Model. Enter pmdb1
4. The sepmdadm utility asks you to define a subscriber. Enter workstat1
5. The sepmdadm utility asks for the next subscriber. Enter workstat2
6. The sepmdadm utility asks for the next subscriber. Just press Enter to indicate that there are no more subscribers at this time.

You must still perform a complementary procedure at the subscriber stations. (See Pointing Subscriber Stations to the PMDB for a description of this procedure.)

7. The sepmdadm utility asks for the parent PMDB. Just press Enter to indicate that the new PMDB is not the subscriber of another PMDB.
8. The sepmdadm utility asks for the password PMDB. Press Enter to indicate no password PMDB for your new PMDB.

   Note: By specifying a password PMDB, you can subscribe your station to a hierarchy in which PMDBs are automatically kept identical. A password PMDB is the top node of such a hierarchy. A password change at any station in the hierarchy is replicated upward to the password PMDB and then downward to all the other subscribers.
9. The sepmdadm utility asks whether your station is running NIS or NIS+. Enter “y” if it is running either one; otherwise, enter “n” or just press Enter.
10. The sepmdadm utility asks whether you want to define users with special attributes for the PMDB.
    a. The administrators of a PMDB are users authorized to change the properties of the PMDB, and to change the values of tokens in the pmd.ini file.
    b. The sepmdadm utility asks you to enter the name of the administrator. Enter adm1.

       The sepmdadm utility asks for the next administrator. Enter adm2.

       The sepmdadm utility asks for the next administrator. Just press Enter to indicate that you have no more administrators to specify at this time.
    c. The auditors of a PMDB are users authorized to view the audit log files of the PMDB.
       The sepmdadm utility asks you to enter the name of the auditor. Just press Enter to indicate that you have no auditor to specify at this time.
    d. The password managers of a PMDB are users authorized to change passwords in the PMDB.
       The sepmdadm utility asks you to enter the name of the password manager. Just press Enter to indicate that you have no password manager to specify at this time.
    e. The desktop managers of a PMDB are stations from which administrators can manage the PMDB.
       The sepmdadm utility asks you to enter the name of the desktop manager. Just press Enter to indicate that you have no auditor to specify at this time.

11. The sepmdadm utility now displays the selections you have made and asks you to confirm them. If you answer yes, eTrust AC builds the PMDB using the answers you supplied. If you have made a mistake in one or more choices, answer no and begin again.

Pointing Subscriber Stations to the PMDB

To establish a station as a subscriber to a PMDB, it is not sufficient to specify the subscriber's name at the PMDB's station; you must also perform a procedure at the subscriber station.

To subscribe a station to a PMDB interactively, issue the following command at the subscriber station and follow the instructions that appear on your screen:

```
sepmdadm --subsconfig -i
```

To subscribe the local station to a PMDB using the command line, you must use the parameters --parentpmd and --passwdpmd, in addition to the parameter --subsconfig.

For example, to subscribe the local station to the PMDB called pmdb2 located on HOST2 and to the password PMDB called master1 located on HOST1, enter the following command:

```
sepmdadm --subsconfig --parentpmd pmdb2@HOST2 --passwdpmd master1@HOST1
```

UID/GID Synchronization

Because you may receive messages that refer to users by UID rather than by username, it is important to know each user's UID. However, if you are using a PMDB do not pay attention to how your new users' UIDs are assigned, the users may receive different UIDs on each subscriber machine. It is therefore best to ensure that you can depend on each user to have the same UID everywhere; and the same is true of GIDs. For more information, see UID/GID Synchronization in the chapter “Managing the Policy Model Database” in the *Administrator Guide*.

Deleting Files

To delete files in the Policy Model directory, you must delete the rules from the database or bring eTrust AC down. You can execute changes to the directory even though the Policy Model already exists, by running sepmdadm.

Files

The sepmdadm utility uses the seos.ini file, but no other special files are used.

See Also

The sepmd and sepmdd utilities in this chapter.

sepmdd

The Policy Model daemon.

The sepmdd daemon is the PMDB daemon. The sepmdd daemon performs the following functions:

- It administers the eTrust AC and UNIX databases of the Policy Model.
- It administers the subscribers' database.
- It propagates changes from the PMDB to the subscriber databases.

Syntax

```
sepmdd policyModel
```

Parameters

policyModel

The name of the Policy Model.

How sepmdd Works

The eTrust AC agent, seagent, starts sepmdd; You do not need to run sepmdd explicitly. The sepmdd daemon runs under the logical user id "_seagent" for eTrust AC, and with the user id "root" in UNIX. You cannot designate another logical user under which sepmdd runs.

The PMDBs are stored in a common directory. You specify the name of the common directory with the _pmd_directory_ token in the [pmd] section of the seos.ini file, on the station where the Policy Models reside. Each Policy Model resides in a subdirectory of the common directory. The name of the Policy Model is the name of the subdirectory in which it resides.

When sepmdd starts, it checks whether any subscriber databases need updating, and updates them if necessary. After this startup process, sepmdd waits for user requests, which are sent by the Policy Model management program, sepmd, and by the selang utility, using seagent.

When sepmdd receives a request, it applies the request to the PMDB and sends the result back to the user. If the request should be propagated, sepmdd propagates the update to its subscriber databases.

The sepmdd daemon attempts to update a subscriber database for the period specified in the _QD_timeout_ token. If the maximum time elapses and the daemon does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list. After it completes its first scan of the subscriber list, sepmdd then performs a second scan, in which it tries to update the subscribers that it did not succeed in updating during its first scan. During the second scan, it tries to update a subscriber until the connect system call times out (approximately 90 seconds).

Note: The _QD_timeout_ token may exist in both the seos.ini and pmd.ini files. If it does, sepmdd uses the value in the pmd.ini file.

If a subscriber is unavailable during the second scan, sepmdd attempts to send it updates every 30 minutes. To modify this interval, set the _retry_timeout_ token. Since the updates must be sent in the order in which they are received, sepmdd does not send subsequent updates to the subscriber database until it becomes available.

If you set the pull_option token in the [pmd] section of the subscriber database's seos.ini file to yes, the subscriber database is updated as soon as possible. seagent informs the parent Policy Models that the host is up for every Policy Model on the machine, and that its subscriber PMDBs are up, and sepmdd sends the update immediately. For more information on the pull_option token, see the *Administrator Guide*.

Whenever sepmdd fails to update a subscriber database, it writes a warning message in the Policy Model error log. For more information about the Policy Model error log see Setting Up a PMDB in the chapter “Managing the Policy Model Database” in the *Administrator Guide.*

eTrust AC attempts to fully qualify subscribers as they are added or deleted from the Policy Model.

To remove a subscriber from the list of unavailable subscribers, enter the following command:

sepmd -r policyModel subscriber

If a subscriber database rejects an update, as can occur if the subscriber database differs from the PMDB, sepmdd writes an error message in the Policy Model error log and continues.

To view the error log, enter the following command on the host where the PMDB resides, enter:

sepmd -e policyModel

You can have sepmdd automatically shut itself down after a period of inactivity. By default, however, sepmdd does not shut itself down. If you want sepmdd to shut itself down, set the _shutoff_time_ token to a value greater than 0. This value indicates the minutes of inactivity allowed before sepmdd shuts itself down. To shut sepmdd down manually, enter:

sepmd -k *policyModel*

Important! Do **not** use the UNIX command *kill -9* to shut down sepmdd manually; this may destroy the PMDB.

UID/GID Synchronization

Because you may receive messages that refer to users by UID rather than by username, it is important to know each user's UID. But if you are using a PMDB and you pay no attention to how your new users' UIDs are assigned, the users may receive different UIDs on each subscriber machine. It is best to ensure, instead, that you can depend on each user to have the same UID everywhere; and the same is true of GIDs. See UID/GID Synchronization in the chapter "Managing the Policy Model Database" in the *Administrator Guide*.

Filter Mechanism

You may want your PMDB to selectively update the subscriber stations below it. To define which records are sent to the subscriber stations, point the filter token in the pmd.ini file to a filter file. Updates to the subscriber stations are then limited to the records that pass the filter file.

A filter file consists of lines with six fields per line. The fields contain the following information:

- The form of access permitted or prohibited.
  The possible values are AUTHORIZE_DELETE, AUTHORIZE_MODIFY, CREATE, DELETE, DEPLOY, EDIT, FILESCAN, GET, SEOS_ACCS_READ, JOIN_DELETE, JOIN_MODIFY, MODIFY, READ, START, or UNDEPLOY.
- The environment affected.
  The possible values are ETRUST, UNIX, NT, or NATIVE
- The class of the record.
  The possible values include all classes in eTrust AC, including user-defined classes.
- The objects within the class that the rule covers.
  For example, User1, AuditGroup, or TTY1
- The properties that the record grants or cancels.
  For example, OWNER and FULL_NAME in the filter line for user records means that any command having those user properties are filtered. You must enter each property exactly as it appears in the *Reference Guide*.
- Whether such records should be forwarded to the subscriber station or not.
  The possible values are PASS or NOPASS

You can use an asterisk in any field to mean "all possible values." If more than one line covers the same records, the first applicable line is used.

In each line of the filter file, spaces separate the fields. In fields with more than one value, semicolons separate the values. Any line beginning with "#" is considered a comment line. Empty lines are not allowed. Here is an example of a line from a filter file:

CREATE	**eTrust**	**USER**	*****	**FULL-NAME;OBJ_TYPE**	**NOPASS**
form of access	*environment*	*class*	*record name (* =all)*	*properties*	*treatment*

For example, suppose the file with this line is named TTY1_FILTER, and the pmd.ini file of the Policy Model TTY1 contains the line filter=/opt/CA/eTrustAccessControl/TTY1_FILTER. The Policy Model TTY1 does not send records that create new eTrust AC users with the FULL_NAME and OBJ_TYPE (Admin, auditor, and so on). The asterisk means "regardless of name."

The following are the selang commands that are relevant for each access value:

Access	**selang Command**
AUTHORIZE_DELETE	authorize-
AUTHORIZE_MODIFY	authorize
CREATE	newres, newusr, newgrp, newfile
DELETE	rmres, rmusr, rmgrp, rmfile, join- (UNIX)
DEPLOY	deploy
EDIT	editres, editusr, editgrp, editfile
FILESCAN	search
GET	get devcalc
JOIN_DELETE	join-
JOIN_MODIFY	join
MODIFY	chres, chusr, chgrp, chfile, join (UNIX)
READ	list
START	start devcalc
UNDEPLOY	deploy- (undeploy)

eTrust AC does not validate rules; therefore, if you enter an invalid value in a rule, the rule never matches an update transaction.

The pmd.ini File

Each PMDB has its own pmd.ini file. This file contains the tokens that define and determine the activity of the PMDB. The file resides in the Policy Model directory.

The sepmdd utility creates a pmd.ini file-if it does not already exist-with the minimum number of tokens needed. The sepmdd daemon uses all the tokens in the pmd.ini file on the station on which the Policy Model resides. For a description of the pmd.ini file, see the *Reference Guide*.

The seos.ini File

The sepmdd utility uses the following tokens in the seos.ini file on the station on which the Policy Model resides:

Section	Token
pmd	_pmd_directory_ _min_retries_ is_maker_checker pull_option _QD_timeout_ _retry_timeout_ _shutoff_time_

Other Files

No other special files are used.

Notes:

When you use selang and choose a Policy Model as your target (using hosts pmd@hostname), queries to sepmdd apply to the PMDB but not to the various subscriber databases.

- Make sure that a PMDB does not become a subscriber of itself. If a PMDB is subscribed to itself, the Policy Model may block or the network may become overloaded, filling the disk in the process.
- When updating a Policy Model in the UNIX environment of selang, you can neither specify more than one user in the newusr command, nor specify more than one group in the newgrp command.
- When updating UNIX file attributes from selang, the Policy Model generates a message stating that the command was passed to its subscribers.
- When working on a Policy Model, you cannot query the status of UNIX file attributes.
- If you set the value of _shutoff_timeout_ to zero, the sepmdd daemon remains up and running indefinitely until you shut it off manually. Use the command sepmd -k to shut down the Policy Model daemon.

See Also

The seagent, sepmd, and sepmdadm utilities in this chapter.

sepropadm

Administers eTrust AC database properties.

Note: This utility is used by eTrust AC technical support personnel only.

The sepropadm utility adds, updates, and deletes properties in the database. You must invoke this utility from the directory in which the database resides, and while the eTrust AC daemons are **not** running. The sepropadm utility can add only one property at a time.

Important! Do **not** use sepropadm with a description file that was **not** certified by eTrust AC technical support personnel.

To enable use of these properties in selang commands, update the paths of the user-defined property files in the [property] section of lang.ini file. For more information, see the appendix “The lang.ini File.”

Important! If you use advanced policy-based management and reporting, when adding or deleting classes, or adding or deleting class properties in the eTrust AC database, you must also do the same in the eTrust AC initial database (init_ac_db). This database is used for policy deviation calculations; it is located at the path specified by the init_ac_db token of the seos.ini file (by default, *<eAC_InstallDir>*/data/devcalc/init_ac_db).

Syntax

```
sepropadm file
```

Parameters

file

A description file supplied by eTrust AC support personnel. The description file uses the following format:

Lines that begin with a semicolon (;) are comments and are not processed.

There must be one line that begins with the hash symbol (#). This line must precede the description lines.

The description line to add a new property, which must conform to the following format:

```
CLASS=%s PROPERTY=%s TYPE=%d SIZE=%d FLAGS=%x
```

The description line to update a new property, which must conform to the following format:

```
CLASS=%s OBJECT=%s PROPERTY=%s VALUE=%s
```

The description line to delete a new property which must conform to the following format:

```
CLASS=%s PROPERTY=%s
```

Files

The eTrust AC database files are used.

Example

```
The following is a sample description file.
 ; Sample Patch File for the eTrust Access Control database
 ; Copyright 2004 Computer Associates International, Inc.
 ; ------------------------------------------
 ; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
 # seclassadm database add property patch utility
 ; Format is :
 CLASS=PROGRAM PROPERTY=MD5 TYPE=31 SIZE=16 FLAGS=0
```

See Also

The dbmgr, seclassadm, and lang.ini utilities in this chapter.

sepurgdb

Purges the eTrust AC database of references to undefined records.

The sepurgdb utility searches the entire database for references to undefined records, and then deletes those references from the database, thereby reducing the size of the database. For purposes of safety, first back up the database, and then invoke the utility while the eTrust AC daemons are *not* running.

Notes:

When a record is deleted, references to it in lists such as ACLs or lists of group membership are usually left as is, to reduce processing time. This does not cause any problems, since eTrust AC assigns a previously unused, unique ID to each new record. The only reason you would use this utility is to free up some disk space.

- To run sepurgdb, you must be root.
- You must invoke sepurgdb from the directory containing the database files.
- The database management system uses pre-allocated disk space. The size of the database file normally does not change significantly after purging. When the size of the database is increased later, the file size may not change significantly due to the pre-allocation.

Syntax

```
sepurgdb FilePath [Username]
```

Parameters

FilePath

The file name to be used for the log files. The sepurgdb utility creates two log files:

- *FilePath*.err contains a log of errors encountered.
- *FilePath*.log contains a log of actions taken.

You can merge the two logs and direct them to the standard output by specifying a minus sign (-) for *FilePath*.

Username

(Optional) Defines the name of the user that is used to replace deleted owners (users that no longer exist) of the group connection for the USER record .

Note: The specified user must exist in the database, otherwise the utility ignores this option.

sereport

Provides reports-accessible from a web browser-of database and Policy Model information.

The sereport utility provides a variety of reports in a user-friendly format that is accessible with a web browser. sereport operates on the current database used by the authorization daemon.

Notes:

- To use sereport, you need READ privileges in all queried databases.
- The configuration file *eTrustACDir*/etc/sereport.cfg is created by default.
- You need a web browser to benefit from sereport.

Syntax

```
sereport [-f|-file pathname] -r|-report number [-h help] [-host hostnames]
```

Switches

-r | report *number*

The report number to display.

Options

-f | -file *pathname*

The full path of the configuration file. If you do not use the -f option, sereport uses the file /*eTrustACDir*/etc/sereport.cfg as a default.

-h

Show help.

-host *hostnames*

The names of the host you want to report on. This token is optional, and if you do not use it, sereport takes the host from the config file.

Reports

You can generate the following reports by setting appropriate tokens in the sereport configuration file.

Report Number	Title and Description	Configuration File Section	Tokens
Report 1	Administrative Privileges Display specified administrative privileges of users.	admin_report	Object_pattern User_Mode Hostname Report_place

Report Number	Title and Description	Configuration File Section	Tokens
Report 2	Login Limitation Display login limitations of users.	disablelogins_report	Object_pattern User_Mode Hostname Report_place Properties
Report 3	Dormant Accounts Display inactive accounts by date (days). If an account does not have any login information, the create time is used to calculate dormant days.	dormant_report	Object_pattern Hostname Report_place Dormant_account
Report 4	Last Login Display last login date of users.	login_report	Object_pattern User_Mode Hostname Report_place
Report 5	Password Change Display list of users whose passwords must be changed within the specified number of days.	passwd_report	Object_pattern User_Mode Hostname Days_to_change Report_place
Report 6	Warning Mode Display resources with objects in warning mode.	warning_report	Class_Name Object_pattern Hostname Report_place
Report 7	Untrusted Programs Display programs in untrusted mode.	untrust_report	Object_pattern Hostname Report_place
Report 8	Users' Privilege Access Rights Show access privileges of users to specified resources.	accessor_report	Class_Name Object_pattern Hostname Accessor Report_place
Report 9	Compare users/groups in databases Display users and groups that are defined in some but not all, databases.	grp_usr_compare	Object_pattern Hostname Report_place
Report 10	Compare Protected Resources Display whether resources are defined in the specified databases.	res_compare	Class_Name Object_pattern Hostname Report_place

Report Number	Title and Description	Configuration File Section	Tokens
Report 11	Compare Access Rights Display the differences in resource restrictions between a Policy Model and a subscriber database.	acc_compare	Class_Name Object_pattern Hostname Report_place
Report 12	Compare Users' Information Display differences in user definitions between a Policy Model and a subscriber database.	usr_compare	Object_pattern Hostname Report_place Properties
Report 13	Compare PMDB and Subscriber Display the rules (as defined by the Class_Name and Object_pattern tokens) that exist on the PMDB, but do not exist on the subscriber database. **Note:** If all of the rules on the PMDB exist on the subscriber database, then the databases are reported as IDENTICAL.	pmdb_compare	Class_Name Object_pattern Hostname Report_place

sereport Configuration File Token Descriptions

Additional information on the tokens, listed in the following table, appears in the configuration file.

Accessor

The pattern (mask) for accessor selection. Use * to select all accessors.

Class_Name

A list of classes.

Days_to_Change

The number of days left until the user is requested to change passwords.

Dormant_account

The period the account is to be considered dormant.

Hostname

A list of hosts from which the data is retrieved.

Objects_pattern

The pattern (mask) for object selection. Use * to select all objects.

Properties

Attributes associated with the objects.

Report_place

The full path location where the report is printed.

User_Mode

A list of user modes, separated by commas.

Title

Specifies the color of the report's title.

class_title

Specifies color of the report's class_title.

background

Specifies the color of the title report's background. The background and logo must be written in full path.

logo

Creates the logo. The background and logo must be written in full path.

The *.jpg files are by default in */eTrustACDir*/data/reports

How to Use sereport

1. Set the relevant tokens in the relevant section of the configuration file.
2. Run the utility by entering the command.

   /eTrustACDir/bin/sereport -f *full_path* -r *report_number*

   You have the option of including [-host *list of hosts*].
3. Start your web browser.
4. The report is now available by opening the file you designated in the report_place token of the sereport cfg file.

seretrust

Generates the selang commands for retrusting programs and securing files.

Programs with setuid and setgid bits are stored in the database with their full descriptions, including their inode values. If you restore the system from backups, the programs occupy different inodes. eTrust AC detects the mismatch between the inodes and marks all the trusted programs as untrusted. The seretrust utility locates the trusted programs that are defined in the database and updates their inode values, so that when you invoke eTrust AC, the trusted programs remain trusted.

If you do not specify any switches, only untrusted programs and untrusted secured files are processed.

The seretrust utility also checks whether programs have been changed but have not yet been caught by the Watchdog or seosd. (This means that in the eTrust AC database, these programs are still marked as trusted.) These programs are added to seretrust output with a note that the program content or timestamp has been changed, and the program needs to be retrusted.

Notes:

The program generates a script that contains the commands required to retrust every trusted program and secured file in the database.

- The output is directed to the standard output device. To direct the output to a file, use the redirection commands.
- If you omit the -l parameter, seretrust obtains the list of programs and files to be retrusted from the eTrust AC daemon.

Syntax

```
seretrust switches [-a] path
```

Switches

-a

Processes all trusted and untrusted objects. (You can specify this switch alone or with any of the other switches.)

-h

Displays help for this utility.

-l

Extracts information about the programs and files from the database in the current directory.

-p

Processes records in the PROGRAM class only.

-s

Processes records in the SECFILE class only.

Parameters

path

Specifies the path of the programs to be retrusted. The specified directory and all subdirectories are processed.

Example

To create a script file that can retrust both program and security files, follow these steps:

1. Log in as an eTrust AC database administrator.
2. Enter the following seretrust command.

   ```
   seretrust > Retrust_script_name
   ```

   Both trusted programs and secured files are processed because you did not specify any switches; the root path is used because you did not specify any base path.

   seretrust displays the following information on the screen :

   ```
   Retrusting PROGRAMs & SPECFILEs, Base path = /
   Total of 0 entries retrusted. (Class=SECFILE)
   Total of 16 entities retrusted. (class=PROGRAM)
   ```

   The following is the contents of the script created after issuing the previous seretrust command:

   ```
   cr PROGRAM /usr/bin/chgrpmem trust
   cr PROGRAM /usr/bin/chie trust
   cr PROGRAM /usr/bin/crontab trust
   cr PROGRAM /usr/bin/cu trust
   cr PROGRAM /usr/bin/ecs trust
   cr PROGRAM /usr/bin/newgrp trust
   cr PROGRAM /usr/bin/rmquedev trust
   cr PROGRAM /usr/bin/rsh trust
   cr PROGRAM /usr/bin/sysck trust
   cr PROGRAM /usr/bin/uuname trust
   cr PROGRAM /usr/lib/methods/showled trust
   cr PROGRAM /usr/lib/mh/post trust
   cr PROGRAM /usr/lib/mh/slocal trust
   cr PROGRAM /usr/lpp/X11/bin/xlock trust
   cr PROGRAM /usr/lpp/X11/bin/xterm trust
   cr PROGRAM /usr/sbin/chvirprt trust
   ```

3. To retrust the programs and files, issue the following command:

   ```
   selang -f Retrust_script_name
   ```

serevu

Deals with users who have tried unsuccessfully to log in.

The serevu utility deals with users who have had a specified number of failed logins during a specified period. Depending on your specifications, it can disable, report, or ignore the user. By default, it disables the user in the UNIX environment of the local station. If no such user exists locally, serevu checks the NIS information to find the user.

If you set a value in the passwd_pmd token, in the [seos] section of the seos.ini file, eTrust AC updates the appropriate PMDB, which then propagates the update to its subscribers. If you did not set a value in the passwd_pmd token, eTrust AC uses the value in the parent_pmd token, which then propagates the update to its subscribers.

Notes:

- For the serevu utility to work properly, the user root must have write access to the file /etc/passwd.
- If you define a remote machine in the serevu.cfg file, you must also give login authorization to the remote mechine.
- On Linux platforms, the serevu utility works only with PAM.

Syntax

serevu *switch options*

Switches

daemon

Runs the utility as a daemon. This is the default value.

nodaemon

Runs the utility as a regular process.

Options

-d [*dd* | FOREVER]

Specifies the amount of time for which the user's login is disabled. Use the suffix m for minutes, h for hours, d for days, or w for weeks. For seconds, do not use a suffix. To disable the user indefinitely, specify FOREVER.

-f *nn*

Specifies the number of failed logins. The serevu utility disables the accounts of users who reach this number of failed logins over the specified period.

Note: We recommend that the number of failed logins, defined by the **-f** option **or** by the value of the **def_fail_count** token in the seos.ini file, always be the same as the value of allowed unsuccessful login attempts set on your system. (On Solaris, for instance, the system values for this are set in /etc/default/login by the RETRIES token.)

Default values are five for Solaris and Linux, three for HP-UX and AIX, and three **or** five for NCR. See your operating system documentation for more details.

-h

Displays the help screen.

-s *ss*

Specifies the period, in seconds, that serevu scans for failed logins. If the scan period is 300 seconds (the default value in the seos.ini file), serevu searches for failed logins that occurred during the prior 300 seconds.

-t *tt*

Specifies the period, in seconds, that elapses between successive serevu checks. The default value is 120 seconds.

serevu Messages

Messages from serevu are sent to the following locations:

system log

Start messages and their parameters.
Each time user accounts are enabled or disabled.
Warnings of undefined users.
(These types of events are called LOG_NOTICE.)

Warnings of multiple root login attempts. This type of event is called LOG_WARNING.

eTrust AC trace file

Each time user accounts are enabled or disabled.
Warnings of undefined users.

eTrust AC audit file

Every failed login.
Each time user accounts are enabled or disabled.

Disabling Accounts

Before disabling accounts, serevu checks whether the particular accounts are defined in the local host or in NIS. If the users are local, serevu adds an asterisk before the users' password in the /etc/passwd file, to disable them. To enable disabled user accounts, serevu deletes the asterisk. If the users are in NIS, serevu disables them in the NIS password file.

If the users are not defined in NIS or cannot be found in the /etc/passwd file, serevu disables them by adding the users to the local /etc/passwd file with the invalid password *NO_PASSWORD*. This feature is not available on IBM AIX platforms.

The senable utility enables these users by deleting them from the local /etc/passwd file. After serevu enables users, they must select a new password.

eTrust AC database administrators can use senable to enable previously disabled accounts.

Note: The name “root” here refers to any user whose uid is zero (0).

Even if root is disabled according to the utility's criteria, serevu does not disable the user root but does send warning messages to the system log and trace.

By editing the serevu configuration file, a user with the ADMIN attribute can customize the action serevu takes for a specific user or for groups of users.

Starting serevu

serevu is normally run by root. There are various ways for other users to run serevu.

- If root has ADMIN status, root can authorize another user to invoke serevu by defining a sesudo job. The seos.ini file need not be changed.
- If root does not have the ADMIN property, you can still set up another user to run serevu as follows:

  a. Give the user ADMIN status.

  b. Give the user WRITE permission to the terminals.

  c. Give the user READ, WRITE, and CREATE permission to the *eTrustACDir*/etc/serevu.cfg files.

  The user can then start serevu from a command line in either of two ways:

  – Define a sesudo job that invokes serevu or use the UNIX environment.

  – Use su to become root, and then invoke serevu.

- As root, you can also start serevu at boot, without requiring anyone to enter the serevu command. In this case, do not define a sesudo job.

Killing serevu

A user can execute a kill command to serevu. When this happens, the utility first intercepts the kill command and saves a list of all the disabled user accounts in the file specified in the token save_disable_path, in the [serevu] section of the seos.ini file. Only then does serevu terminate.

The serevu Configuration File

You can use the serevu configuration file to customize the activity of serevu. You can select:

- Where messages regarding disabled user accounts should be sent.
- The environments in which user accounts are disabled.
- Which users or groups of users to single out for specific processing.

By default, this file is in the *eTrustACDir*/etc directory and is named serevu.cfg. In the file, the format of each line is:

```
userName,action,time
```

The components of the line are:

- **userName**-A name or mask that identifies the names of the specified users. You can specify a user's complete name or use the standard selang wildcards. To see how eTrust AC performs string matching, see the appendix “String Matching.”
- **action**-Any one of the values in the following table. To specify more than one action per *userName*, specify *userName* in more than one line.

AUDIT

Send a message to the audit log when the user account is disabled (by one of the other actions).

DPMDS

Disable the user in the eTrust environment of the PMDB.

DPMDX

Disable the user in the UNIX environment of the PMDB.

DSECU

Disable the user in the eTrust AC database of the local station.

DUNIX

Disable the user in the UNIX environment of the local station. This is the default action of serevu.

NONE

Do not disable this user and do not record failed logins.

SYSLOG

Send a message to the system log.

TRACE

Send a message to the system trace.

Note: By choosing either the DPMDS or DPMDX action, you can use serevu to disable users at all computers that are managed by the PMDB.

- **time**-a numeric value that specifies the amount of time that the user account is disabled. You can only use this variable with the DPMDS, DPMDX, DSECU, and DUNIX actions.

  The time value can be one of the following:

  – A number with no suffix, for seconds

  – A number plus the suffix m, for minutes

  – A number plus the suffix h, for hours

  – A number plus the suffix d, for days

  – A number plus the suffix w, for weeks

  – FOREVER if the user should not be automatically enabled after any specific period of time

For example:

- To disable all users whose names begin with “sys” in the eTrust environment of the PMDB, and to prevent serevu from automatically enabling them, enter:

  ```
  sys*,DPMDS,FOREVER
  ```

- To disable users whose names begin with “acct” in the UNIX environment of the local station for 30 minutes, and to send messages about them to the eTrust AC trace file, enter:

  ```
  acct*,DUNIX,30m
  acct*,TRACE
  ```

- To prevent user accounts whose names begin with “mgmt” from being disabled, enter:

  ```
  mgmt*,NONE
  ```

The seos.ini File

serevu uses the following tokens in the seos.ini file. For descriptions of these tokens, see the appendix “The seos.ini Initialization File” in the *Administrator Guide*.

Section	Token

Section	Token
serevu	admin_user config_file def_fail_count def_disable_time def_sleep_time def_diff_time save_disable_path

Note: The amount of time a user account is disabled cannot be less than the amount of time between each serevu scan. The amount of time a user account is disabled should be a multiple of the time between each serevu scan.

On Solaris systems, serevu uses the shadow file to disable and enable users. The name of the shadow file is located in the YpServerSecure token, in the [passwd] section of the seos.ini file.

Other Files

If serevu falls, it saves the names of the user accounts that it disabled in a special file. The default value in most environments is *eTrustACDir*/log/serevu_disable.users.

Notes:

- If serevu is reinstating a user and finds that the user has already been reinstated by some other means, the utility leaves the user's entry alone.
- serevu does not disable undefined users. Regarding undefined users, serevu sends a warning to the system log and trace.

See Also

The seaudit and senable utilities in this chapter.

sessfgate

Routes and reformats Unicenter Security APIs from the message queue to eTrust AC.

The Unicenter Security APIs on UNIX all channel into a message queue. The sessfgate utility processes the API requests sent through the message queue and routes these reformatted and rerouted requests to eTrust AC. The utility then translates the return codes of eTrust AC to Unicenter TNG equivalents.

In order to activate the gateway, you must run the Unicenter Integration setup procedure. The Unicenter Integration setup installs the sessfgate program in the *eTrustACDir*/tng/bin directory (where *eTrustACDir* is the directory where you installed eTrust AC, by default /opt/CA/eTrustAccessControl). After Unicenter Security is shutdown and eTrust AC is started, sessfgate accepts API requests instead of SSF.

Syntax

```
sessfgate [-i|-s|-l] -t
```

Switches

-I

Specifies to start the gateway.

-s

Specifies to stop the gateway.

-l

Specifies the status.

-t

Toggles the tracing file (log file = opt/CA/eTrustAccessControl/log/sessftrace.log).

Note: If you run seload before running Unicenter TNG, you must start sessfagte manually with the following command:

```
eTrustACDir/tng/bin/sessfagte -I
```

where *eTrustACDir* is the directory in which you installed eTrust AC.

sesu

The eTrust AC version of the UNIX su command.

The sesu utility provides a transparent su command that does not require the user to furnish the password of the substituted user. The authorization process is based on the eTrust AC access rules as defined in class SURROGATE and, optionally, on the password of the user executing the command.

Note: Do not use this command during the implementation period, until you have defined all users to the eTrust AC database. This prevents you from opening up the entire system to users who are not defined to eTrust AC. To protect against inadvertent use of this program, it is marked in the file system so that no one can run it. The security administrator must mark the program as executable and setuid to root before you can use it.

Syntax

```
sesu options name
```

Options

-

Sets the environment to that of the target user. You can use this option with the -c option.

-c

Executes only the specified command. You can use this option with the - option.

-h

Displays the help screen.

Parameters

name

Changes the ID associated with the session to the ID specified by the parameter *name*.

The seos.ini File

The sesu utility uses the following tokens in the seos.ini file:

Section	Token
sesu	AlwaysTargetShell FilterEnv old_sesu Path SystemSu UseInvokerPassword

Other Files

The sesu utility uses the following additional special files:

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

Notes:

- If the eTrust AC authorization server is not found, the program executes the system's standard su command.
- If you use both the - and -c option, you must follow the - option with the target user's name.
- The default target user is root.
- If /etc/shells exist, sesu does not allow su to root unless the shell is specified in that file.

Examples

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

  ```
  sesu
  ```

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

  ```
  sesu -
  ```

- The following command surrogates to the user John.

  ```
  sesu John
  ```

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

  ```
  sesu - Carol -c “ls -la /home/carol”
  ```

sesudo

Executes commands for one user with the permissions of another user.

The sesudo utility borrows the permissions of another user (the *target* user) to perform one or more commands. This allows regular users to perform, for example, actions-such as the mount command-that require superuser authority.

The rules governing user authority to perform commands in this way are defined as access rules in the SUDO class. A record in the SUDO class contains a command script, and can specify both users who are permitted to run the script with sesudo and users who are forbidden to.

Syntax

sesudo *options*

Options

-h

Displays the help screen.

-list

Lists sesudo commands you can execute. These are the SUDO records defined in the eTrust AC database that you are authorized to execute.

profile_name

The name the security administrator gives to the target user's command.

Preparing to Use sesudo

You must complete several steps before you can use the sesudo command. You must perform the first step only once. Perform the other steps every time a new user is given the authority to execute the sesudo command, or every time a new record is defined in the SUDO class.

1. Define the sesudo program as a trusted setuid program owned by root. Do this once per eTrust AC installation. The format of the command is:

   ```
   newres PROGRAM /opt/CA/eTrustAccessControl/bin/sesudo defaccess(NONE)
   ```

   Note: To specify a target user(other than root), use the parameter targuid(user_id). If you want sesudo to ask for the user's password before executing the command, use the password parameter.

2. Give a user the authority to execute the sesudo program. Do this once for every user who is entitled to this authority. The format of the command is:

   ```
   authorize PROGRAM /opt/CA/eTrustAccessControl/bin/sesudo uid(user_name)
   ```

3. Permit the user to surrogate to the target user id. Do this for every user; and for each user, do it for every target user id that you want available to that user. For example, to make root available as a target ID, the format of the command is:

   ```
   authorize SURROGATE USER.root uid(user_name) \
   via(pgm(/opt/CA/eTrustAccessControl/bin/sesudo))
   ```

4. Define new records in the SUDO class for every command script to be executed by users. For each command script, you can define permitted and forbidden parameters, permitted users, and password protection. For details, see chres/editres/newres in the chapter “The selang Command Language” in the *Reference Guide*.

5. Permit or forbid the user to access the command script's record in the SUDO class. Do this for every command script that a user should be able to access. The format of the command granting permission is:

   ```
   authorize SUDO profile_name uid(user-name)
   ```

   If defaccess is none, specify each user who is granted permission with the authorize command. If defaccess is not none, use the authorize command to specify each user who is forbidden access.

   The sesudo utility can display the command before executing. Display depends on the value in the echo_command token in the [sesudo] section of the seos.ini file. The default value calls for no display, but you can change the value.

6. The output of the sesudo utility depends on the command being performed. Error messages are sent to the standard error device (stderr), usually defined as the terminal's screen.

Specifying the Comment Parameters for the SUDO Class

When you define the profile for the SUDO command, the COMMENT property contains the target user's command, which a normal user can execute. In addition to specifying a command, you can specify permitted and prohibited parameters.

For example, the following definition contains the COMMENT property with both permitted and prohibited parameters:

```
authorize SUDO profile_name uid(user-name) \
comment('command; pro1|pro2|...proN;per1|per2|...perN')
```

command

The target user's command, which a normal user can execute.

***permitted parameters* (for example, per1|per2|...perN)**

The parameters that you specifically allow the regular user to execute. These parameters can contain patterns or variables.

***prohibited parameters* (for example, pro1|pro2|...proN)**

The parameters that you prohibit the regular user from executing. These parameters can contain patterns or variables.

Prohibited and permitted parameters can also contain the following variables:

Variable	Description
$A	Indicates an alpha value
$G	Indicates an existing eTrust AC group name
$H	Indicates the parameter starts with the user's home directory
$N	Indicates a numeric value
$O	Indicates the user's name
$U	Indicates an existing eTrust AC user name
$e	Indicates a SUDO command with no parameters for the rule
$f	Indicates an existing file name
$g	Indicates an existing UNIX group name
$h	Indicates an existing host name
$r	Indicates UNIX read access to the file name
$u	Indicates an existing UNIX user name
$w	Indicates UNIX write access to the file name
$x	Indicates UNIX execute access to the file name

The sesudo utility checks each parameter entered by the user in the following manner:

1. Test if parameter N matches permitted parameter N. (If permitted parameter N does not exist, the last permitted parameter is used.)
2. Test if parameter N matches prohibited parameter N. (If prohibited parameter N does not exist, the last prohibited parameter is used.)

   If all the parameters match permitted parameters, and none match prohibited parameters, sesudo executes the command.

Examples

1. If you do not want to allow any parameters, define the profile as follows:

   ```
   newres SUDO profile_name comment('command;*')
   ```

2. If you want to allow a user to invoke the name parameter, define the profile as follows:

   ```
   newres SUDO profile_name comment('command;;NAME')
   ```

   In the above example, the only parameter the user can enter is NAME.

3. If you want to prevent a user from using -9 and -HUP but allow a user to use all other parameters, define the profile as follows:

   ```
   newres SUDO profile_name comment('command;-9 -HUP;*')
   ```

4. To prohibit two parameters-the first is a UNIX user name and the second is a UNIX group name-and permit two parameters-the first is numeric and the second is alphabetic-define the profile as follows:

   ```
   newres SUDO profile_name comment('command;$u | $g ;$N | $A')
   ```

   The user cannot enter a UNIX user name, but can enter a numeric parameter for the first operand; and the user cannot enter the UNIX group name but can enter an alphabetic parameter for the second operand.

5. If there are several prohibited parameters for several operands in the command, define the profile as follows:

   ```
   newres SUDO profile_name comment('cmd;pro1 pro2 | pro3 pro4 | pro5 pro6')
   ```

   pro1 and pro2 are the prohibited parameters of the first operand of the command; pro3 and pro4 are the prohibited parameters of the second operand of the command; and pro5 and pro6 are the prohibited parameters of the third operand of the command.

6. To permit or deny a user to execute the su_cmd with no parameters but permit a user to execute testuser with no parameters, define the profile as follows:

   ```
   newres SUDO su_cmd comment('su -;$e;testuser)
   ```

Return Values

Each time sesudo runs, it returns one of the following values.

-2

Target user not found, or command interrupted

-1

Password error

0

Execution successful

10

Problem with usage of parameters

20

Target user error

30

Authorization error

Files

The sesudo uses the following files:

- seos.ini
- The sesudo utility uses the token echo_command in the sesudo section of the seos.ini file.
- _eACPRODPATH_/etc/sesudo.ext

  This is a configuration file that contains the shared library paths.

seuidpgm

Extracts trusted programs.

The seuidpgm utility extracts all the programs whose Set-User-ID bit or Set-Group-ID bits are on. seuidpgm traverses a file system and creates the selang commands for adding these programs to the PROGRAM class.

seuidpgm creates the commands in the selang command language and writes them to the standard output. You can use a pipeline to the selang utility, or redirect the output to a file. We recommended that you redirect the output to a file, because then you can edit the output to remove unwanted programs or add additional programs. Use this procedure to search for undesirable setuid programs in the your system.

seuidpgm descends through the paths specified at the command line to all subdirectories of the starting path. Multiple start paths are allowed.

You can specify any number of options. When specifying more than one option, separate the options with spaces.

If a program is a setuid program and has write access, seuidpgm treats the program like all other setuid programs, but also sends a warning to standard error.

For more information on how to control PROGRAM class records, see the *Administrator Guide*.

Syntax

```
seuidpgm options startDir
```

Options

-d

Automatically creates entries for setuid and setgid programs in the PROGRAM class, with defaccess set to execute, instead of analyzing the file permissions in UNIX to determine the permitted file access. In some cases, one setuid or setgid program executes another one. If you do not include this option, the program trying to execute the setuid or setgid program is *not* able to execute it.
We recommend that you use this option.

-f

Creates rules for both the FILE and PROGRAM classes.

-g

Creates GROUP records for setgid programs. Do not use this option unless you have not already run the UxImport utility.

-l

Create a single permit for programs which have *hard* or *symbolic* links.

-n

Does not traverse NFS at all.
We recommend that you use this option.

-o

Writes the file names to the standard output but does not create selang commands.

-p

Enables setuid programs from NFS directories, but only when the mount table allows setuid from that mounted file system.

-q

Runs the utility in Quiet-Mode; error messages are not sent to standard error.

-s

Creates entries for setuid/setgid programs in class SECFILE, instead of creating entries for the PROGRAM class.

-u

Creates USER records for setuid programs. Do not use this option unless you have not already run the UxImport utility.

-x *path*

Excludes a directory from the tree. The specified directory is not searched for setuid and setgid programs. This option must be the last option specified in the command line. *Path* is the full path of the directory to be excluded. To exclude more than one directory, repeat the -x option for each directory.

Files

The seuidpgm utility does not use the seos.ini file. It uses the following files:

/etc/passwd

- /etc/group
- The system mount-table file.

Notes:

- If you want to scan your file system from some directories only (not from the root directory) and to include the -l option, use multiple starting paths on the command line; otherwise the -l option may be inefficient.
- We recommended that you run the UxImport utility to define users and groups before running the seuidpgm utility. However, if you have not run UxImport, you can use seuidpgm with the -g and -u options to define users and groups.

Examples

- The following command prints selang commands to add all programs with set-user-id or set-group-id turned on, defaccess execute, checking for duplicate names or the same inode, in quiet mode, and without passing through NFS. The program scans from the /usr directory and its subdirectories, the /var directory and its subdirectories, and the /etc directory and its subdirectories. Output is directed to the file seprogs.seos in your home directory.

  ```
  seuidpgm -dlqn /usr /var /etc > ~/seprogs.seos
  ```

  The output should look similar to the following:

  ```
  ## ************************************************
  ## seuidpgm List Sun Feb 9 14:24:16 1997
  # Start Path= /usr
  # ***********************************************
  nr PROGRAM /usr/lpp/bos/inst_root/lpp/inu_LOCK defaccess(EXEC)
  nr PROGRAM /usr/lpp/X11/bin/xlock defaccess(EXEC)
  nr PROGRAM /usr/bin/setsenv defaccess(EXEC)
  nr PROGRAM /usr/bin/shell defaccess(EXEC)
  nr PROGRAM /usr/bin/su defaccess(EXEC)
  nr PROGRAM /usr/bin/sysck defaccess(EXEC)
  nr PROGRAM /usr/bin/tcbck defaccess(EXEC)
  nr PROGRAM /usr/bin/usrck defaccess(EXEC)
  nr PROGRAM /usr/bin/vmstat defaccess(EXEC)
  ```

- The following command scans the root directory and all its subdirectories, except the /home directory:

  ```
  seuidpgm -qln / -x /home
  ```

See Also

The UxImport, selang, seoswd, seosd, and selang utilities in this chapter.

seversion

Displays version information for an eTrust AC program module.

The seversion utility displays information regarding the version of an eTrust AC module. You can display the following data:

- The global and minor version numbers.
- The date and time that the module was compiled.
- The station that the module was compiled on.

This utility has the following format:

```
seversion switch module
```

Switches

-a

Displays the requested information in table format.

-e

Displays the requested information in extended table format. It shows all the information of the original table format, but also includes type information for the module. Also, for executables, information regarding linked libraries appears.

-g

Displays only the global version number, omitting titles.

-h

Displays the help screen.

-m

Displays only the minor version number, omitting titles.

-t

Displays only the module type, omitting titles.

-5

Displays the MD5 signature, omitting titles.

Parameters

module

The file name of the module whose version number you want to display.

Example

To display version information for the sesudo utility, enter the following command:

```
seversion /opt/CA/eTrustAccessControl/bin/sesudo
A message similar to the following appears on the screen:
Access Control SeVersion v8.0 - Display Module's Version
Copyright 2004 Computer Associates International, Inc.
Running Under:  HP HP-UX
Module [/opt/CA/eTrustAccessControl/bin/sesudo] version is: 1.20 (1.00)
Compiled On: May 11 2004 15:12:55    HP-UX 904
MD5 Signature: 9F6A61283AAED1732F8BA129E1780AE9
```

sewhoami

Displays the user's eTrust AC user ID and other security credentials.

The sewhoami utility displays the user name as it is known to the eTrust AC authorization daemon. sewhoami is similar to the whoami utility provided by UNIX systems, but it produces different and often more useful information.

- If the user executes an su command and then executes the UNIX whoami utility, it displays the user name according to the user ID acquired after executing the su command.
- If the user executes an su command and then executes the eTrust AC sewhoami utility, it displays the original login ID of the user; it also displays authorization information.

This utility has the following format:

```
sewhoami options
```

Options

-a

Displays the user's credentials; that is, the contents of the user's ACEE. For more information on the ACEE, see the *Administrator Guide*.

-d (-debug)

Displays the ACEE handle associated with the user and the handle's name in the database.

See Also

- The whoami and su utilities supplied with your UNIX operating system.
- The sesu utility in this chapter.

uninstall_eTrustAC

Removes eTrust AC from the current station.

The uninstall_eTrustAC utility deletes part or all of the eTrust AC product from the station on which you execute the command. The default, -all, removes the entire product from the station. You can remove only the eTrust AC administration tools.

This utility has the following format:

`uninstall_eTrustAC` *`options eTrustACDir`*

Options

-admin

Removes only administration tools such as Security Administrator and seauditx from the station.

-all

Removes the entire product from the station. This is the default value.

-f

Removes eTrust AC in silent mode.

-force

Forces uninstall to proceed even if the kernel extension unload process fails. (The eTrust AC kernel extension should be unloaded prior to uninstall.)

-help

Displays the help screen.

-ignore_dep

Specifies that the uninstallation procedure will not check for dependency with other products.

Parameters

eTrustACDirectory

The directory where eTrust AC is installed. The default directory is /opt/CA/eTrustAccessControl.

Example

To remove eTrust AC completely from this station, if it was installed in the default directory /opt/CA/eTrustAccessControl, enter the command:

```
uninstall_eTrustAC
```

UxImport

Extracts UNIX users, groups, hosts, and TCP services from the UNIX operating system.

The uximport utility extracts information from the UNIX operating system about the defined users, groups, terminals, hosts, and TCP services. It extracts information from NIS, if it is installed, and the system is configured accordingly. It also provides DNS support. You should use uximport as part of the installation procedure.

uximport automatically processed the extracted information to generate selang commands that you can use to add users and groups to the eTrust AC database. The generated commands are printed to the standard output. Use redirection to a file, or pipeline to the selang utility.

Syntax

```
UxImport switches [options]
```

Switches

-a

Generates the selang commands required to import users, groups, and hosts, and to join users to their default groups.

-c

Generates the selang commands required to explicitly join users to their default groups.

Note: If you also import groups with the -g switch, eTrust AC generates the commands that join users to the groups to which they are explicitly linked.

-g

Generates the selang commands required to import groups from UNIX and NIS to the eTrust AC database.

-h

Generates the selang commands required to import hosts from UNIX, NIS, and DNS to the eTrust AC database. uximport extracts host information from the file /etc/hosts and from NIS, and builds HOST resources. For each host entry in the file /etc/hosts or extracted from NIS, the appropriate newres command is built, and permission to receive any TCP service is assigned to that host.

In addition, DNS is supported with the -d option. In some machines, information from the file /etc/hosts and NIS is ignored if the specified DNS daemon is running. In Solaris, the information gathered depends on the configuration of the system in the file /etc/nsswitch.conf.

-t

Generates the selang commands required to import terminal rules from UNIX and NIS to the eTrust AC database.

uximport extracts host information from the file /etc/hosts and from NIS, and builds TERMINAL resources. For each entry in the file */etc/hosts* or extracted from NIS, the appropriate newres TERMINAL command is built and permission to log in from the terminal is granted.

In addition, DNS is supported with the -d option. In some machines information from the file /etc/hosts and NIS is ignored if the specified DNS daemon is running. In Solaris, the information gathered depends on the configuration of the system in the file /etc/nsswitch.conf.

-T

Generates the selang commands required to import TCP services from UNIX and NIS to the eTrust AC database. The names are set according to GECOS in UNIX. The names are truncated to 40 characters if they are longer.

-u

Generates the selang commands required to import users from UNIX and NIS to the eTrust AC database. The actual user names are set according to GECOS in UNIX. The names are truncated to 40 characters if they are longer.

Options

-d

Specifies the use of DNS for generating the list of hosts and terminals to import. Must be accompanied by the -h or -t switch.

-f

Skips search for multiple occurrences of the same name. By not using the standard uximport processes, this option handles the importing of many users and groups speedily, and saves memory. The -f option does not apply to hosts; you should combine them with one or more of the following switches: -u, -g, or -a. Also, use one of these switches when including the -c switch in conjunction with the -f option.

Join and surrogate rules are printed along with create records.

-G

Creates SURROGATE class rules for groups. uximport adds a record to the SURROGATE class for each group it defines, therefore making SURROGATE requests protected resources. It also adds rules so that root can surrogate to each of the groups.

-gr *n*

Specifies the number of grace logins for all users, forcing users to change their passwords after *n* logins. This ensures that the PASSWD_L_C property in the USER record is updated.

-o *owner*

Sets ownership rules for each record. We recommended that you use this option to prevent root from automatically becoming the owner of all the records. *Owner* is the name of the user or group to be assigned ownership of all records defined by uximport.

Note: You must specify this option as a separate argument followed by *owner*.

-pr *groupname*

Assigns a profile group to users. If you specify this option, eTrust AC uses that group when building a user's profile; otherwise, it uses the primary UNIX group.

-r

Specifies to continue scanning after a failure.

-s

Creates SURROGATE class rules for users and groups. The uximport function adds a SURROGATE record for every group it defines, thereby making SURROGATE requests to the group into protected resources.

-U

Creates SURROGATE class rules for users. uximport adds a record to the SURROGATE class for each user it defines, therefore making SURROGATE requests into protected resources. It also adds rules so that root can surrogate to each of the users.

-v

Displays the status of the program (verbose mode). We recommended that you use this option if your site has many users, groups, or hosts, so that you can verify the program's progress.

Example

The following command extracts all information of users, groups, and hosts from the UNIX and NIS databases. It then creates the selang commands that add those records to the database. uximport then creates SURROGATE class records and provides progress indication. Output is directed to the file uxinfo.seos in your home directory.

```
UxImport -a -s -v > ~/uxinfo.seos
```

See Also

- The seerrlog, selang, and seuidpgm utilities in this chapter.
- The chapter “Interacting with LDAP” in the Administrator Guide. The ldap2seos program extracts user information from an LDAP database for importing into eTrust AC.

Appendix A: Trace Messages

This section contains the following topics:

Conventions (see page 217)
Messages (see page 217)

Conventions

All messages begin with a date and time prefix, followed by an event-type word in uppercase and a symbol such as **:**, **!**, or **>**. The following table explains the meaning of the symbols.

:

eTrust AC was signaled for an event or performed an action.

>

eTrust AC made an authorization decision resulting in *D* (Deny), *P*, (Permit), or *BYPASS* (The event did not require the interpretation of an access rule-for example, a setuid request to the same UID as the current UID.)

!

eTrust AC detected an error-for example, a request from an unknown process.

Messages

The symbols described in the previous section precede the event arguments, described in this section.

ACTION : eTrust killed P=ppp

eTrust AC denied a setuid or login request and killed the requesting process (ppp) as a precautionary measure.

ALARM ! Uid uuu breached the system!!!

An unknown process made a request such as fork, exec, or setuid. The process is unknown to eTrust AC and, in addition, the UID assigned to the process is not assigned to any other process in the system. This implies that the user logged in without eTrust AC being notified. This situation can occur as a result of a software bug or if the user logged in immediately after eTrust AC scanned the current process status but before completing initialization.

APIAUTH ! P=ppp U=uuu ChangePasswd(user) Error 0xerr

Process *ppp*, associated with user *uuu*, wants to change the password of *user*. The result of this request was an error with its code specified in hex. Use the semsgtool utility to determine the nature of the error.

APIAUTH ! P=ppp U=uuu CheckPasswd(user) Error 0xerr

Process *ppp*, associated with user *uuu*, wants to check the validity of a new password for *user*. The result of this request was an error with its code specified in hex. Use the semsgtool utility to determine the nature of the error.

APIAUTH ! P=ppp U=uuu Error, Unknown API Service nnn

Process *ppp* used the Application Interface and passed a service code that the eTrust AC Programming Interface does not support, probably because of user error. Check the cause of the error, correct the source, and recompile it.

APIAUTH ! P=ppp U=uuu GeneralResourceProc Error nnn >description

Process *ppp*, working under UID *uuu*, issued a request to access a general resource; however, the specified resource cannot be resolved. Either the specified class is not defined or the specified access is not known, probably because of user error. Check your code, correct it, and recompile.

APIAUTH ! P=ppp U=uuu in VerifyCreate only for ROOT

Process *ppp*, working under UID *uuu*, issued a VerifyCreate request to build an ACEE. This operation is permitted only to multiuser processes that are associated with UID 0 (root).

If the specified process is to run as a multiuser process, rerun the process under root authorities. If not, determine why the process issued the request.

APIAUTH : P=ppp U=uuu in VerifyDelete only for ROOT

Process *ppp*, working under UID *uuu*, issued a VerifyDelete request to remove an ACEE. This operation is allowed only to multiuser processes that are associated with UID 0 (root).

If the specified process is supposed to run as a multiuser process, rerun it under root authorities. If not, determine why the request was issued.

APIAUTH ! P=ppp U=uuu LoginProc Error nnn >description

Process *ppp*, working under UID *uuu*, requested to verify a user's login. The eTrust AC login verification procedure failed. Contact your vendor's technical support.

APIAUTH ! P=ppp U=uuu NULL ACEE Error VerifyCreate (ACEEH=hhh)

A user process marked as “server” made a request a to create an ACEE (probably as the server process was handling login for an accessor). The result is a NULL ACEE for one of the following reasons:

- The specified user is not defined in the eTrust AC database.
- The issuer of the VerifyCreate request did not provide all the information correctly.
- The specified user is not allowed to log in.

APIAUTH ! P=ppp U=uuu NULL ACEE Error VerifyDelete (ACEEH=hhh)

Process *ppp*, associated with user *uuu*, and which is probably marked as a 'server' process, has requested to delete the ACEE handle *hhh* (probably as part of handling the user's signoff). However, no ACEE is associated with this handle, so eTrust AC cannot delete it.

APIAUTH : P=ppp U=uuu Request with ACEEH=1 > New ACEEH=hhh

Process *ppp*, working under UID *uuu*, requested access to a general resource and supplied an ACEE handle of -1. eTrust AC used the ACEE handle associated with the requesting process. This message is typical of single user processes that request access to a resource. No action is required.

APIAUTH ! P=ppp U=uuu VerifyCreate(ACEEH=hhh) Error nnn

Process *ppp*, working under UID *uuu*, issued a request to VerifyCreate (to build an ACEE). The VerifyCreate procedure failed. Contact your vendor's technical support.

APIAUTH > P=ppp U=uuu VerifyCreate DENY (Result=[P/D/C]) string

The VerifyCreate request was denied for one of the following reasons:

- The specified user cannot login due to time or day rules
- The user cannot work from the specified terminal
- The specified password (if supplied) is incorrect
- One of the reasons described in the messages that follow.

APIAUTH > P=ppp U=uuu VerifyCreate OK (ACEEH=hhh)!

The VerifyCreate request was granted. An Accessor Environment Element (ACEE) was built in storage. eTrust AC returned an ACEE handle (ACEEH) to the calling program. If the specified user is not defined to eTrust AC, the function returned an ACEEH of -1.

APIAUTH ! P=ppp U=uuu VerifyDelete(ACEEH=hhh) [OK | Error 0xerr]

Process *ppp*, associated with user *uuu*, which is probably marked as a 'server' process, has requested the deletion of the ACEE handle *hhh* (probably as part of handling the user's signoff). The result of the VerifyDelete request is either OK or error; if the latter, the error code appears in hex as err. Use the utility semsgtool to determine the nature of the error.

APIAUTH > P=ppp U=uuu VerifyRequest(ACEEH=hhh, C=ccc, R=rrr, A=nnn) DENY (Result='D')Why ? detaileddenialreason

The request to access resource *rrr* of class *ccc* with access *xxx* was denied. If the ACEEH is -1, the denial was based on universal-access rules. If the ACEEH is not -1, the denial was based on the user associated with the specified handle. The second line provides a detailed reason for the denial.

APIAUTH > P=ppp U=uuu VerifyRequest(ACEEH=hhh, C=ccc R=rrr, A=xxx) PASS

The request to access a resource *rrr* of class *ccc* with access *xxx* was granted. If the ACEEH is -1 (the user is not defined to eTrust AC), the permission to access the resource was based on universal-access rules. If the ACEEH is not -1, the permission was based on access rules relating to the user associated with the specified handle.

CONNECT : P=ppp U=uuu ACEEH=hhh from ipip:port1 to socket 6000 host=iiii

A request to open a window on host *iiii* (X-Terminal or station) was made by process *ppp* associated with UID *uuu*.

Note: The port number is always 6000; all other TCP/IP connect requests are ignored by eTrust AC.

CONNECT > P=ppp U=uuu from ipip:port1 to socket 6000 host=iiii BYPASS

eTrust AC bypassed the CONNECT request without interpreting access rules, because the program executing in process *ppp* is the registered XDM program.

CONNECT > Result: [P/D/C] P=ppp ACEEH=hhh TERM=tttWhy ? detaileddecisiontext

The CONNECT result is **D** (Deny) or **P** (Permit). The second line provides a reason for the decision.

ERROR ! Cannot fork. Errno nnn.

During initialization, eTrust AC forks a few times to become a daemon. The fork request failed with the specified error number.

If you cannot determine the cause of the problem, contact your vendor's technical support.

ERROR ! Exec of eTrust agent failed ddd

The Engine cannot start up the Agent daemon. Check that the seagent executable is located in the right place, usually *eTrustACDir*/bin/seagent. If this file exists in the correct location, report the problem to your vendor's technical staff. In the message text, *ddd* is the error number that eTrust AC received from the operating system when trying to execute seagent.

ERROR ! Failed to get memory for LOGIN programsERROR ! Failed to get memory for NFS devicesERROR ! Failed to get memory for PRIV programsERROR ! Failed to get memory for XDM programs

These messages imply a severe shortage of memory. Either your computer does not meet the minimum memory requirements to run eTrust AC, or a software bug was found. Contact your vendor's technical support.

ERROR ! Failed to get memory for PROC table

When seosd starts up, it must scan all the running processes to resolve all required information on each running process. seosd failed to allocate memory for this purpose; therefore, it terminates execution. This is caused by a severe memory shortage.

ERROR ! Failed to register login pgm: programname

During startup, eTrust AC registers all executable files that are to be treated as login programs. The list of login programs is defined in the eTrust AC code for each operating system environment.

You can override this list using the loginpgms.init file discussed in *Administrator Guide*. The specified *program-name* cannot be located on the file system during startup. eTrust AC ignores the program and startup continues.

ERROR ! Failed to register privileged pgm: programname

During startup, eTrust AC registers all executable files that are to be treated as privileged programs. The list of privileged programs is defined in the eTrust AC code for each operating system environment.

You can override the list of privileged programs by using the privpgms.init file as discussed in the *Administrator Guide*. The specified *program-name* cannot be located on the file system during startup. eTrust AC ignores the program and startup continues.

ERROR ! Failed to register XDM pgm: programname

During startup, eTrust AC registers all executable files that are to be treated as XDM programs. The list of XDM programs is defined in the eTrust AC code for each operating system environment.

You can override the list of XDM programs using the xdmpgms.init file as discussed in the *Administrator Guide*. The specified *program-name* cannot be located on the file system during startup. eTrust AC ignores the program and startup continues.

ERROR : No Memory for FileDb List

During startup, seosd cannot allocate memory to hold the list of protected files. This is probably due to a severe shortage of memory. The seosd daemon is terminated.

ERROR ! No Memory for GroupDb ListERROR ! No Memory for HostDb ListERROR ! No Memory for ServDb ListERROR ! No Memory for UserDb List

These messages imply a severe shortage of memory. Either your computer does not have the minimum memory required to run eTrust AC, or a software bug was found. Contact your vendor's technical support.

ERROR ! PreMatureExec Assuming FORK Child=ppp Parent=PPP

This message indicates that process ID (*ppp*) issued an EXEC system call, which is not known to seosd. Normally, such messages indicate that seosd was not yet informed of the FORK system call that preceded the EXEC request. It may indicate a problem in the serialization locks that the eTrust AC extension to the UNIX kernel, SEOS_syscall, must maintain.

If the *ppp* in the message text is the pid of seagent, you can ignore the message. If you get the message more than once, report the problem to your vendor's technical support.

ERROR ! P=ppp Exec Failed

eTrust AC received an EXEC event, but the inode number of the executable was zero. This message occurs when invoking a script file that does not contain the #! shell-program declaration line at the beginning. No action is necessary.

ERROR ! eTrust file table set failed

seosd attempted to set the file table (a table of all eTrust AC protected files); however, SEOS_syscall refused this request. The most likely causes are insufficient memory in the kernel, or different versions of seosd and SEOS_syscall. eTrust AC file protection cannot continue to function properly.

If you can, resolve the version mismatch. If everything looks fine, report the problem to your vendor's technical support.

ERROR ! seosini_ShutDown rv=errorno

eTrust AC encountered an error during shutdown. Report the error to your vendor's technical support.

ERROR ! String too general 'path'

An attempt was made to define a generic rule for file protection, probably through a newfile or newres FILE command. However, the specified path cannot be a generic file access rule. The file rule is not defined.

ERROR ! Unknown request: Type:ttt Pid=ppp, Buff=bbb

eTrust AC received a request from its system call, but the request type *ttt* is not recognizable. This can be due to a software version mismatch between the eTrust AC system call and seosd, or because of a software error. The request came from process *ppp*, and *bbb* is a printout of the request buffer. Report the problem to your vendor's technical support.

EXEC : P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName [Attached to: ipaddress]

eTrust AC received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 indicates that eTrust AC has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a regular program (that is, not setuid or setgid); therefore, eTrust AC grants its execution without invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, eTrust AC reports this in the message text.

EXEC sg: P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

eTrust AC received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means eTrust AC has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setgid program; eTrust AC determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, eTrust AC reports this in the message text.

EXECsu : P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

eTrust AC received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means eTrust AC has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setuid program; eTrust AC determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, eTrust AC reports this in the message text.

EXECsusg: P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

eTrust AC received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means eTrust AC has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setuid and setgid program; eTrust AC determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, eTrust AC reports this in the message text.

EXEC > P=ppp U=uuu (R=rrr E=eee S=sss) to (E=EEE) BYPASS

Although the program is setuid, setgid, or both, and its execution should have invoked the access rule decision mechanism, eTrust AC bypassed this check because the owner of the file *EEE* is the same as the current effective UID (*eee*). The program execution cannot change the scope of the privileges of the process. If the program is defined in the database as a trusted program and was modified or otherwise tampered with, program execution is not granted.

EXEC > Result: 'R' [stage=sss gstag=ggg ACEEH=hhh rv=rc]Why? DetailedDecisiontext

eTrust AC checked the authority of the user to execute the program and the result *R*, where *R* is either D (deny) or P (permit). The stage *sss* and the granting-stage *ggg* indicate which phase of the decision flow determined the result. The ACEE handle *hhh* was used as the accessor to the program. If the result is 'C' (check) it means eTrust AC did not make a decision, probably because of a software error-contact your vendor's technical support and provide them with the return value *rc*. *Detailed-Decision-text* is a textual description of the stage and granting-stage. If the result was *P*, the program is executed successfully. If the result is *D*, the program will not be executed and the user receives a permission denied message.

EXECARGS: 'execution arguments'

Because of an EXEC syscall, eTrust AC displays the executed command line with all the arguments passed to it.

EXIT : Going down...

eTrust AC started the shutdown process and disabled the interception of system calls.

FATAL ! in seosrt_InitDatabase (nnn) Layer = nnn Stage = nnn Return Code = 0xnnn

eTrust AC cannot initialize the database I/O routines. The possible reasons are:

- No eTrust AC database in the directory is identified by the dbdir token in the seos.ini file.
- The user invoking eTrust AC is not root.
- The database is corrupt.

If you cannot correct the problem, contact your vendor's technical support.

FILE : P=ppp U=uuu (D=dev I=inode) acc : pathname

Process *ppp* associated with userid *uuu* attempted to access an eTrust AC protected file. In the message text, *dev* and *inode* are the device and inode of the file being accessed, respectively; *acc* is the access mode (that is, READ, WRITE, and so on); and *pathname* is the real path name of the file being accessed.

FILE > Result 'D' eTrust File Only 'filename'

The result of the file access request is D (denial) because only eTrust AC can access this file. Even if the access rules permit access, eTrust AC is hard-coded to deny access to this file.

FILE > Result: 'R' [stage=sss gstag=gs ACEEH=hhh rv=rv (recordname)Why? detailedreasontext

The result R of the file access request is either D (deny) or P (permit). The stage *sss* and granting stage *gs* are mapped to a text-string reason, on the second line (following “Why?”). In the message text *hhh* is the accessor handle associated with the request's accessor and *record-name* is the name of the access rule record that triggered the decision to deny or permit access.

FORK : P=ppp U=uuu G=ggg Child=cppp Pgm:ProgramName

eTrust AC intercepted a fork request made by process *ppp* associated with UID *uuu* and GID *ggg.* The child process id is *cppp*. *Program-Name* is the program running in the parent process (and, initially, also in the child process). eTrust AC never denies a fork request; it is always granted. Variations of the fork system call, such as vfork and kfork, are also reported as fork requests.

GETCRED : P=ppp, Get Credentials by Ticket

This is an information-only message, which indicates that *ppp* (usually the process ID of the Policy Model daemon, sepmdd) requested the credentials of a specific ticket holder (a client process that requests the services of sepmdd). For more information, see the description of GTICKET in this appendix, and the description of sepmdd in the chapter “Utilities in Detail.”

GPEERNAM: P=ppp, ADDR=addr, N=desc

eTrust AC intercepted the getpeername() system call to verify which IP address is associated with the current process. This system call is always granted. In the message text, *ppp* is the process id issuing the getpeername() call and *addr* is the IP address associated with the socket descriptor *desc.*

GTICKET: P=ppp, Get Authentication Ticket

This is an information-only message, which indicates that *ppp* requested seosd to issue an authentication ticket for it. Whenever the Policy Model client, sepmd, communicates with sepmdd, the server verifies the identity of the client through the passed ticket. The client sends the acquired ticket to the server using socket communication. The server then passes this ticket to seosd to get the credentials of the ticket holder with the GETCRED request. In this way, sepmdd ensures the identity of the client requesting its services.

INET : P=ppp, from ipaddress:localport to port portnumber

eTrust AC intercepted an incoming Internet accept request that was issued by the remote *ip-address* requesting the TCP/IP service *port-number*.

INET > Result: 'R' ipaddr>locport, stg=stage gtsg=gstageWHY ? DetailedReasonText

The result *R* of the Internet request is P (permit) or D (deny). In the message text, *ip_addr* is the IP address of the request. *Detailed-Reason-Text* is the textual description that indicates which stage and granting stage phase of the decision flow made the final decision to deny or allow the TCP/IP service for the requesting host.

INFO : AutoDisabling Tracedue to tight fsspace (space)

The trace facility automatically disables itself when the amount of free space left in the file system where the trace file resides, goes below a threshold specified by the trace_space_saver token in the seos.ini file. In the message text, *space* is the amount of free space left on the file system.

INFO : Can't fetch fs freespace (errno=err)

The Auto Disable feature of the trace facility cannot determine the amount of free space in the file system. In the message text, *err* is the error integer received from the UNIX statfs() call. Report the problem to your vendor's technical support.

INFO : DB Query

The seosd daemon received a request to extract information from the eTrust AC database.

INFO : DB Request

The seosd daemon received a request to modify or query data in the eTrust AC database.

INFO : Filter Mask: 'mask' is registered

The seosd daemon registers each filter mask that is read from the trcfilter.init file, so that messages matching the mask are not sent to the trace file.

INFO : GroupList Registered with nnn entries

When seosd runs under the NIS server, it caches all group entries (from /etc/group and NIS maps) at startup, so that seosd can solve GID to group name translations without invoking ypserv processes and TCP/IP requests. This message also indicates that the under_NIS_server token in seos.ini is set to YES. If the station where eTrust AC is running is not the NIS server, set the under_NIS_server token to NO. In the message text, *nnn* is the number of group entries that were cached.

INFO : HostList Registered with nnn entries

The seosd daemon caches all entries from /etc/hosts at startup. In the message text, *nnn* is the number of host entries cached.

INFO : Login program: programname is registered

The seosd daemon must recognize all the programs through which users log in to the system. eTrust AC treats a setuid system call invoked by a login program as a login request, and not as a setuid request. In the message text, *program-name* is the full path of the login program that was registered. The seosd daemon takes the names of the login programs internally, from the eTrust AC startup code or from the loginpgms.init file (if it exists).

Note: loginpgms.init either adds login programs to the internal list or overrides the internal list, depending on its contents.

INFO : NFS Device Majors Registered, nnn entries

The checks that the Watchdog performs for trusted programs include checking the device number on which the file resides. This check can lead to errors if the file resides on an NFS mounted file system-especially an auto-mounted file system-for which device numbers can have a different value after boot. For this reason, eTrust AC registers the major device numbers of NFS file systems so that they can ignore the non-stable minor device number. eTrust AC has a list of major device numbers for NFS mounted file systems in each environment. If your installation uses a network mounted file system that eTrust AC does not recognize, contact your vendor's technical support for instructions about adding major device numbers to the list. In the message text, *nnn* is the number of major device numbers registered as NFS mounted file systems.

INFO : P=ppp ended

Process *ppp* ended. seosd disassociates this process number from its ACEE (accessor environment element). If process *ppp* was the last process associated with its ACEE, (that is, no other parent processes or subprocesses use the same environment), then the ACEE is removed from storage. This message is not issued immediately after the process has terminated; it is issued only when eTrust AC performs some “garbage collection” to reuse process entries in its internal tables.

INFO : P=ppp Exec Failed

This message indicates that process *ppp* failed to execute the last EXEC syscall, because UNIX refused this request (after eTrust AC granted the execution). Therefore, eTrust AC restores the value of the former executable that was associated with this process, as the program running under this process ID. In most cases, the process terminates. This is not necessarily an error, and you need not take any special action. However, you should use UNIX tools to isolate the reason that execution failed. In most cases, the reason is that a shell script does not have the “#!/bin/sh” header on the first line.

INFO : P=ppp Unknown TTY type typename

The seosd daemon cannot determine if the process *ppp* is using a real TTY or a pseudo TTY. Contact your vendor's technical support.

INFO : Privileged program: programname is registered

The seosd daemon registers a few privileged programs. Such programs are allowed to setuid to any user without checking the SURROGATE class. Currently, you can only make /bin/sendmail a privileged program, due to its flow requirements. You must keep this list as small as possible; we recommended that seoswd monitor all privileged programs to make sure they remain trusted. In the message text, *program-name* is the full path of the registered program. You can override the internal list of privileged programs through the privpgms.init file.

INFO : Restricted File Table set with nnn entries

During startup, seosd found *nnn* entries for eTrust AC protected files, and successfully passed this list to the eTrust AC extension of the UNIX kernel. This is an information-only message.

INFO : SEOS_syscall UnRegister rc=nnn

During shutdown, seosd unregisters itself to the kernel so that it can start up again. In the message text, *nnn* is the return code, which should be zero. If the return code is not zero, report the problem to your vendor's technical support.

INFO : ServList Registered with nnn entries

The seosd daemon caches all entries from /etc/services at startup. In the message text, *nnn* is the number of host entries that were cached.

INFO : ServList registered with nnn portmapper entries

While starting up, seosd registered nnn TCP/IP services that are resolved by the portmapper. This is an information-only message.

INFO : Set site

The seagent daemon, the eTrust AC daemon responsible for communication with other eTrust AC stations, sent seosd a connection request from a remote station.

INFO : Setting PV C=ccc O=ooo P=ppp

The seoswd daemon set the value of property *ppp* in object *ooo* of class *ccc*.

INFO : UserList Registered with nnn entries

When seosd runs under the NIS server, it caches all user entries (from /etc/passwd and NIS maps) at startup, so that seosd can solve UID to user name translations without invoking ypserv processes and TCP/IP requests. This message also indicates the under_NIS_server token in seos.ini is set to YES. If the computer where eTrust AC is running is not an NIS server, set under_NIS_server token to NO in seos.ini. In the message text, *nnn* is the number of user entries that were cached.

INFO : XDM program: programname is registered

XDM programs are those programs that display the userid and password box on X-terminals. XDM programs run under *superuser*, who usually cannot open windows on X-terminals. However, the XDM program must open a window on an X-terminal to present a box with the userid and password for the user to specify. seosd therefore bypasses terminal checking if the program issuing the CONNECT request is a registered XDM program. You can override the internal list of XDM programs through the xdmpgms.init file.

KILL : P=ppp U=uuu kill [Process | All Except] (nn): (proclist)

Process *ppp* associated with user *uuu* attempted to kill all the processes listed in *proclist* (or all the processes except the processes in the list). In the message text, *nn* is the number of target processes.

KILL > Result 'R' [stage=sss gstag=gs rv=rr] ACEEH=hhhWhy? detailedreasontext

The result R of the kill event is either D (deny) or P (permit). In the message text, *sss*, *gs*, and *rr* are the stage, granting stage, and return value of the eTrust AC decision routines, and *hhh* is the accessor handle associated with the kill event. The *detailed-reason-text* appears in the second line and is a derivation of the stage and granting stage codes.

LOGIN : P=ppp User=uuu Terminal=ttt

The seosd daemon intercepted a login request from user *uuu* working on terminal *ttt* under process number *ppp*. A Login Result message should follow this message.

LOGIN > Result: 'R' [stage=stage gstag=gstage rv=nnn] ACEEH=hhh[Why ?detaileddenialreason]

The result of the login request R is either D (deny) or P (permit). In the message text, *stage* and *gstage* are numbers indicating the phase in the eTrust AC flow that made the decision to grant or deny the login request. If the login was permitted, *hhh* is the ACEE handle that is now associated with the issuing process. If the login was denied, *hhh* is set to -1 and a *detailed-denial-reason* appears in the second line. If the *detailed-denial-reason* relates to resource access (such as “no rule granting access to resource”), the resource in question is the terminal from which the user issued the login request.

LOGIN > Result: 'D' Login Disabled for ALL

The login request was denied because login is currently disabled for all users.

LOGIN > Result: 'D' Login Disabled for U=uuu

The login request was denied because login is currently disabled for the specific user. The reason can possibly be that this user is already logged in.

MESSAGE : string

A marker message is placed in the trace file by console request.

NEWPASS : Set new password

The sepass utility requested to set a new password for a userid.

PW_ATTCK: P=ppp make nnn attempts in sss seconds from terminal

The seosd daemon detected that process *ppp*, which is running one of the registered login programs, made *nnn* attempts to specify a user/password combination with no success. eTrust AC concluded that a password guess attack originated at the terminal specified in the message text, and wrote an audit record to the eTrust AC audit file. PWATTACK audit records can trigger actions by the log routing daemons (selogrcd and selogrd).

RESTART : DBSERV restarted by Watchdog (P=ppp)

The seoswd daemon has restarted seosd. In the message text, *ppp* is the process ID of seosd.

SCONSOLE: Login Disabled For UID: uuu

The eTrust AC console utility, secons, issued a request to disable a login request for the userid *uuu*. From this point, login requests for the specified userid are denied.

SCONSOLE: Login is already Disabled for U=uuu

The secons utility issued a request to disable login request for the userid *uuu*. However, login is already disabled for this userid.

SCONSOLE: Login is not Disabled for U=uuu

The secons utility issued a request to re-enable login for the userid *uuu*. However, login is already enabled for this userid.

SCONSOLE: Login Is Now Disabled

The secons utility issued a request to disable login for all users. From this point on, login requests by any user are denied.

SCONSOLE: Login Is Now Enabled

The secons utility issued a request to re-enable login for all users. From this point on, login requests are allowed.

SCONSOLE: Login ReEnabled for U=uuu

The secons utility issued a request to re-enable login for a specified user. From this point on, login requests for this specific user are allowed.

SCONSOLE: No more space in Disabled Logins Table

The secons utility issued a request to disable login for a particular user. However, the login disable table is full. Contact your vendor's technical support.

SCONSOLE: U=uuu is not allowed for operation

A user without the OPERATIONS attribute tried to use one of the secons switches that are not allowed for non-OPERATIONS users.

SCONSOLE: U=uuu is not allowed to disable login for U=uuu2

The user *uuu* tried to disable login for user *uuu2* through secons. However, only root and user *uuu2* are allowed to disable login for *uuu2*.

SCONSOLE: U=uuu is not allowed to Reenable login for U=uuu2

The user *uuu* tried to re-enable login for user *uuu2* through secons. However, only root and *uuu2* are allowed to re-enable login for *uuu2*.

SETGRPS : P=ppp to grouplist

The process *ppp* issued the setgroups system call for the groups specified in *grouplist*.

SGID : P=ppp U=uuu G=ggg to GGG (GROUP.groupname) ACEEH=hhh D=devnum I=inode

Process *ppp*, running with the authorities of UID *uuu* and GID *ggg*, issued a setgid system call for the GID *GGG*. eTrust AC checks the authority of that process using the SURROGATE class and object GROUP.*groupname*, and uses *hhh* as the accessor handle for the request. In the message text, *devnum* and *inode* are the device and inode of the issuing program, respectively. A “SGID Result” message should follow this one.

SGID > P=ppp U=uuu (RG=rg EG=eg SG=sg) to (RG=trg EG=teg SG=tsg) () BYPASS

eTrust AC granted the setgid request without checking any SURROGATE access rules. In the message text, *ppp* is the issuing process id; *uuu* is the userid associated with this process; *rg*, *eg*, and *sg* are the real, effective, and saved GID of that process; and *trg*, *teg*, and *tsg* are the target effective, real, and saved GID with which the setgid request was issued. The reason for the bypass is usually because the current real or saved GID is the same as the target GID, and therefore the setgid request does not change the security scope of the user.

SGID > Result: 'R' [stage=stage gstag=gstage ACEEH=hhh]Why? detailedreasontext

eTrust AC checked the setgid request against a SURROGATE access rule and the result R is P (permit) or D (deny). The decision was made on behalf of the accessor handle *hhh*. In the message text, *detailed-reason-text* is the reason for the denial or grant.

SHUTDOWN! Request Denied. U=uuu not allowed to SHUTDOWN the Server

The userid *uuu* tried to shut down seosd using secons; however, this user's profile does not have the OPERATIONS attribute. The request was therefore denied.

SHUTDOWN: Server going down upon operator's request

The seosd daemon started shutting down following a request from an authorized operator.

SHUTDOWN: Terminating eTrust daemon daemonname P=ppp RV=nnn

eTrust AC terminated its daemon *ppp* as part of its shutdown process; eTrust AC also shuts down seoswd and seagent.

STARTUP: eTrust daemon PID=ppp

The seosd daemon was started; its process ID is *ppp*.

STREAM c: P=ppp Closes Stream Id=iii

Process *ppp* closed a stream with stream ID *iii*. eTrust AC keeps track of all stream-open and stream-close operations to determine later-when a TCP/IP request is processed on behalf of a specific stream-id-which process ID owns the stream.

STREAM o: P=ppp Opens Stream Id=iii

Process *ppp* opened a stream with stream ID *iii*. eTrust AC keeps track of all stream-open and stream-close operations to determine later-when a TCP/IP request is processed on behalf of a specific stream-id-which process ID owns the stream.

SUID > P=ppp U=uuu (R=r E=e S=s) to (R=tr E=te S=ts) (reason) BYPASS

eTrust AC granted the setuid request without checking any SURROGATE access rules. In the message text, *ppp* is the issuing process id; *uuu* is the userid associated with this process; *r*, *e*, and *s* are the real, effective and saved UIDs of process *ppp*; and *tr*, *te*, and *ts* are the target effective, real, and saved UIDs with which the setuid request was issued. The reason for the bypass is usually because the current real or saved UID is the same as the target UID, and therefore the setuid request does not change the security scope of the user. Other possible reasons are that the program issuing the setuid system call is a privileged program (in which case *reason* is For Priv), or that the issuing program is a login program that switches UIDs several times before and after the actual login (in which case *reason* is specified as For Login).

SUID : P=ppp U=uuu (R=r E=e S=s) to USER.username (R=tr E=te S=ts)D=devnum I=inode

Process *ppp*, running with the authority of userid *uuu*, issued a setuid system call to change the current real, effective, or saved UID to UID *uuu*. eTrust AC checks the authority of that process using the SURROGATE class and object USER.*usernam* for that request. In the message text, *devnum* and *inode* are the device and inode of the issuing program, respectively. A “SUID Result” message should follow this one.

SUID > Result: 'R' [stage=stage gstag=gstage ACEEH=hhh rv=rv]Why? detailedreasontext

eTrust AC checked the setuid request against a SURROGATE access rule and the result R is P (permit) or D (deny). The decision was made on behalf of the accessor handle *hhh*. In the message text, *detailed-reason-text* is the reason for the denial or grant.

VERPASS : Verify password

eTrust AC received a request to verify password validity for a user.

WAKE_UP : Server going up

The seosd daemon started to initialize.

WARNING : Associate P=ppp ACEEH=hhh

eTrust AC performs an association between a process and an accessor handle (ACEEH) for any fork request. This message indicates that the association cannot be performed, either because the handle *hhh* is -1 or because *hhh* is not a valid accessor handle. In the latter case, contact your vendor's technical support.

WARNING : Can't verify P=ppp

This message follows an Unknown P= message that indicates a fork request from an unknown process. eTrust AC tries to determine who the user is that UNIX associates with that user. This verification task cannot be completed. A possible reason is that the process has already terminated. If this is not the case, contact your vendor's technical support.

WARNING : DeAssociate P=ppp ACEEH=hhh

eTrust AC performs a dissociation between a process and an accessor handle (ACEEH) for any process that is terminated. This message indicates that the dissociation cannot be performed, either because the handle *hhh* is -1 or because *hhh* does not exist as a valid accessor handle. In the latter case, report the problem to your vendor's technical support.

WARNING : ExecArg for entry with P=ppp not NULL

This warning appears when eTrust AC finds a new process that was not known to the system, and for which the executing program is not known. In most cases, you can ignore the message. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Failed to get ACEEH of P=ppp

eTrust AC was requested to check the authority of process *ppp* but there was no valid accessor handle for that process. In most cases, the reason is that the user associated with the process is not an eTrust AC defined user, or that the process is unknown to the eTrust AC system. In both cases, eTrust AC gives this process only universal access rights. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Login for P=0 ???

When this message appears during startup in systems other than AIX, you can ignore it. If it appears during normal work (after seosd is started and functions), or during startup under AIX, then it identifies a software error, in which case you should contact your vendor's technical support.

WARNING : eTrust failed to kill P=ppp reason=nnn

As a measure of caution, eTrust AC kills processes trying to get sensitive privileges that may create loopholes. Such events can be attempts to surrogate the UID (setuid system-call) with no permission. eTrust AC attempted to kill the violating process, but failed to do so. The reason for the failure is detailed in the reason code returned by the kill system call.

WARNING : Terminal for entry with P=ppp not NULL

This warning appears when eTrust AC finds a new process that was not known to the system and for which the executing program is not known. In most cases, you can ignore the message. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Unknown P=ppp

This message indicates a fork request that was issued by a process not known to eTrust AC. If this message appears for seoswd or seagent during startup, you can ignore it. At other times, it can imply a software error because eTrust AC cannot verify the actual authority of that process. For the latter case, contact your vendor's technical support.

WATCHDOG: Ask if I'm Here (AYT)

The seoswd daemon tried to verify whether seosd is alive and give the expected response. In the message text, AYT is the seoswd “are you there” challenge. You can and should ignore this message; filter it out using the trcfilter.init file. The message implies normal behavior of seoswd.

WATCHDOG: Init initializationtext

The seoswd initialization message, which you can ignore.

WATCHDOG: Log logtext

The seoswd daemon issued a log request. The log request is detailed in *log-text.*

WATCHDOG: SecFile operation result

The seoswd daemon requested the daemon to extract information regarding secured files. In the message text, *operation* can be GETFIRST or GETNEXT; the result can be OK if such information was extracted, or NOFOUND if there are no more secured files in the eTrust AC database. This message signifies normal behavior of seoswd to scan secured files.

WATCHDOG: Timer

The seoswd daemon issues a timer request every few seconds (as set by the seos.ini file). You can and should filter out this message using the trcfilter.init file.

WATCHDOG: Trust Pgm: programname [OK | NOTOK]

The seoswd daemon marked the specified program as a trusted program. This implies that the specified program passed the digital signature tests. In the message text, OK means the trust operation completed successfully, and NOTOK means that seoswd failed to mark the program as trusted. The reason for NOTOK is probably a corrupted database, in which case you should contact your vendor's technical support.

WATCHDOG: Untrust Pgm: programname [OK | NOTOK]

The seoswd daemon marked the specified program as untrusted. This implies that the specified program did not pass the digital signature checks of seoswd. In the message text, OK means that the untrust operation has completed successfully, and NOTOK means that seoswd failed to mark the program as untrusted. A possible reason for NOTOK can be a corrupted database, in which case you should contact your vendor's technical support.

Appendix B: The lang.ini File

This section contains the following topics:

lang.ini File Tokens (see page 240)
general (see page 240)
history (see page 241)
newres (see page 242)
newusr (see page 243)
properties (see page 244)
unix (see page 247)

lang.ini File Tokens

This appendix describes the tokens in the lang.ini file, used by the selang utility.

The lang.ini file contains the following sections:

general

Contains default parameters that apply to more than one type of resource; that is, both new resources and new users.

history

Contains default parameters for the selang history mechanism.

newres

Contains the default values that are assigned to the properties of new resource records. The default value is assigned unless you explicitly set a different value.

newusr

Contains the default values that are assigned to the properties of new user records. The default value is assigned unless a different value is explicitly set.

properties

Contains tokens that specify values for user-defined properties, such as file locations for user-defined properties. The tokens have no default values; you must set them explicitly.

unix

Contains the default values that are assigned when a new user is defined to UNIX from within the selang command shell. The default value is assigned unless you explicitly set a different value.

The remainder of this appendix describes each of these sections.

general

The [general] section contains default parameters that apply to more than one type of resource.

defaultOwner

The name of the owner assigned to a new record.

If you do not specify a value, the creator of the new record is assigned as owner.

history

The [history] section contains default parameters for the selang history mechanism.

HistFile

The name of the file where the commands in the history list are stored. The command list is loaded at the beginning of each session.

No default value; that is, the history list is not saved at the end of a session.

HistSize

The number of commands (a positive integer between 10 and 100) stored by the history mechanism.

Default: 30

newres

The [newres] section contains default values that are assigned by the newres command. The newres command creates new resource records in the database. Each token in this section represents a newres parameter. Parameters not represented in the lang.ini file are assigned default values that are hard-coded in eTrust AC.

DefaultAudit

The default audit mode for the new resource. Valid values are: none, all, success, failure.

Default: failure

DefaultDay

The default day restrictions that apply to the resource. Valid values are: anyday, weekdays, mon, tue, wed, thu, fri, sat, sun.

Default: anyday

DefaultNotify

The default email address to which alert messages regarding the resource record are sent.

No default value; that is., no notification message is sent.

DefaultTime

The default time restrictions that apply to the resource. Valid values are: anytime, startTime:endTime.

Default: anytime

DefaultWarning

Whether warning mode is enabled by default. Valid values are: yes, no.

Default: no

If you do not specify a value for a token, the default value specified in the table is applied.

newusr

The [newusr] section contains the default values assigned by the newusr command, which creates new user records in the database. Each token in this section represents a newusr parameter. Parameters not represented in the lang.ini file are assigned default values that are hard-coded in eTrust AC.

DefaultAudit

The default audit mode for the new user. Valid values are: none, all, success, failure, loginsuccess loginfailure.

Default: failure loginfailure

DefaultDay

The default day restrictions that apply to the user when logging in to the system. Valid values are: anyday, weekdays, mon, tue, wed, thu, fri, sat, sun.

Default: anyday

DefaultExpire

The default expiry date for the user record. Valid values are: expire[*dd*/*mm*/*yy*], expire-.

Default: expire-

DefaultLocation

The default location in which the user works.

No default value

DefaultNotify

The default email address to which alert messages are sent when the user logs in.

No default value; that is., no notification message is sent.

DefaultOrg

The organization for which the user works.

No default value

DefaultOrgUnit

The organizational unit in which the user works.

No default value

DefaultTime

The default time restrictions that apply to the user when logging in to the system. Valid values are: anytime, startTime:endTime.

Default: anytime

If you do not specify a value for a token, the default value specified in the table is applied.

properties

The [properties] section contains parameters that apply to user-defined properties.

UserDefinedTokensFile

The path for a definition file that contains context information for user-defined properties.

Default: none

UserDefinedAttributesFile

The path for a definition file that contains attribute information for user-defined properties.

Default: none

User-Defined Properties

This section is complimentary to the sepropadm utility (see sepropadm in the chapter “Utilities in Detail”). It defines the selang context by which database properties created with sepropadm are recognized. Two definition files that use a format similar to the one used by sepropadm accomplish this. The location of these files is specified in the two tokens of this section.

Note: The properties must be defined in the database (using the sepropadm utility), before the definition files are loaded by selang. The definition files are loaded automatically when selang is run, during the initialization phase.

When these properties are defined in both the appropriate definition files and the database, you can use them in selang commands like any other eTrust AC defined property.

Important! Do **not** use the sepropadm utility with a description file that was **not** certified by your vendor's support personnel.

The Definition Files

To get selang to recognize the new user-defined properties, selang loads two *.def files during its initialization: the Tokens file and the Attributes file.

The Tokens File

User Defined Tokens File

A definition file supplied by your vendor's support personnel. The definition file has the following format:

Lines that begin with a semicolon (;) are comments and are not processed.

One line must begin with the hash symbol (#). This line must precede the description lines.

The description line must conform to the following format:

```
TOKEN=%s DOMAIN=%d CLASS=%d COMMAND=%d
```

The following is a sample definition tokens file:

```
; Sample Token Definition File for user defined properties
; Copyright 2004 Computer Associates International, Inc.
; -----------------------------------------
; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
# token definition file
; Format is :
TOKEN=EMAIL DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=NOEMAIL DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=EMAIL DOMAIN=1 CLASS=USER COMMAND=218
TOKEN=AGE DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=AGE DOMAIN=1 CLASS=USER COMMAND=218
TOKEN=NOAGE DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=TERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=217
TOKEN=NOTERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=205
TOKEN=TERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=205
```

The Attributes File

User Defined Attributes File

A definition file supplied by your vendor's support personnel. The definition file has the following format:

Lines that begin with a semicolon (;) are comments and are not processed.

One line must begin with the hash symbol (#). This line must precede the description lines.

The description line must conform to the following format:

```
PROPERTY=%s TYPE=%d FLAGS=%x
```

The following is a sample definition attributes file:

```
; Sample Attributes Definition File for user defined properties
; Copyright 2004 Computer Associates International, Inc.
; -----------------------------------------
; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
# attributes definition file
; Format is :
PROPERTY=EMAIL TYPE=306 FLAGS=8000
PROPERTY=EMAIL TYPE=5 FLAGS=8000
PROPERTY=AGE TYPE=306 FLAGS=8000
PROPERTY=AGE TYPE=5 FLAGS=8000
PROPERTY=TERMLOCATION TYPE=306 FLAGS=8000
PROPERTY=TERMLOCATION TYPE=5 FLAGS=8000
```

Important! Do **not** use selang with a definition file that was **not** certified by your vendor's support personnel.

unix

The [unix] section contains the default values that are assigned by the newusr command when a user is added to UNIX. Each token in this section represents an argument of the *unix* parameter. UNIX arguments not represented in the lang.ini file are assigned default values that are hard-coded in eTrust AC.

DefaultPGroup

The default group assigned to new users. If you specify a default shell in the server's seos.ini file, it overrides the value specified here.

Default: other

DefaultShell

The default shell of new users. If you specify a default shell in the server's seos.ini file, it overrides the value specified here.

Default: /bin/sh

DefaultHome

The default home directory of the system. If you specify a default shell in the server's seos.ini file, it overrides the value specified here. The user's home directory is a subdirectory of the specified system home directory. For example, if the system home directory is /home, the new user's home directory is /home/userName. If you specify a home directory prefix in the server's seos.ini file, it overrides the value specified here.

For those familiar with earlier versions, the token DefaultHome replaces HomeDirPrefix.

Default: /home

Appendix C: String Matching

This section contains the following topics:

Wildcard Expressions (see page 249)
Examples: Wildcard Matching (see page 251)

Wildcard Expressions

This section describes the syntax that can be used to build wildcard expressions.

eTrust AC performs string matching (globbing) using the wildcard matching and character lists.

Wildcard Matching

eTrust AC supports the following wildcard characters:

Character	Match
* (asterisk)	Any sequence of zero or more characters.
? (question mark)	Any single character.

Character Lists

A character list enclosed by square brackets ([ ]) can contain one or more characters. eTrust AC uses these characters as positive or negative matching criteria.

A character list can be composed of one or more characters. For this type of list, eTrust AC matches any single character in the list. If the list within the brackets is preceded by a caret (^), eTrust AC matches any single character, which is **not** in the list.

A range is a type of character list that specifies a range of characters. eTrust AC matches all the characters in the list, inclusively. If a caret (^) precedes the list, eTrust AC excludes all the characters in the specified list. You can specify both ends of the range, or only its first or last character.

The following table describes the character lists that can be used. Remember, in this syntax, you include the square brackets. Each of the expressions *ch1*, *ch2*, and *chN*, stands for a single character.

List	Description
[*ch1ch2...chN***]**	eTrust AC matches any single character in the list enclosed by the square brackets.
[^*ch1ch2...chN***]**	eTrust AC matches any single character that is **not** in the list enclosed by the square brackets.
[*ch1-ch2***]**	eTrust AC matches any single character in the range, inclusive.
[^*ch1-ch2***]**	eTrust AC matches any single character that is **not** in the inclusive range.
[-*ch2***]**	eTrust AC matches any single character with an ASCII value lower than or equal to the specified character (*ch2*).
[^-*ch2***]**	eTrust AC matches any single character with an ASCII value equal to or higher than the specified character (*ch2*).
[*ch1-***]**	eTrust AC matches any single character with an ASCII value equal to or higher than the specified character (*ch1*).
[^*ch1-***]**	eTrust AC matches any single character with an ASCII value equal to or lower than the specified character (*ch1*).

Examples: Wildcard Matching

To make a single character a “don't care” character that matches any other single character, use a question mark (?):

Specify	To match
mmc?	mmc3, mmcx, mmc5
mmc?.t	mmc1.t, mmc2.t
mmc04.?	mmc04.a, mmc04.1

To match any string of zero or more characters, use an asterisk (*):

Specify	To match
i.c	main.c, list.c, and so on
st*.h	stdio.h, stdlib.h, string.h, and so on
*	All records of the specified class

To match any character in a list, follow one of these examples:

Specify	To match
[abcgk]	a, b, c, g, or k
[^abcgk]	Any character other than a, b, c, g, or k, such as A, B, d, e, f, or @.
[a-z]	Any character between a and z, inclusive.
[^a-z]	Any character with an ASCII value less than “a” or greater than “z.”
[Z-]	Any character with an ASCII value greater than Z's, such as a, b, \, or ~.
[^-A]	Any character with an ASCII value **not** lower than A's, such as B, a, c, or ~.

Index

A

abbreviating commands • 96
access types • 47, 68
ACEE • 227, 231, 233
administering databases • 184
API • 65
  LogRoute • 116, 118
  Unicenter TNG • 200
asset types, Unicenter TNG • 30
audit access filter • 140
audit log
  encrypting records • 118
  output • 47
  split and merge • 114
  trace records • 47
audit.cfg • 140
authorization daemon • 140

C

cache settings • 118, 133, 135, 137
caching
  programs • 74
changing passwords • 91
character lists • 251
characters, special • 96
chmod access • 68
chown access • 68
class
  adding new • 68
  deleting • 68
command • 137
  history • 96
  recognition • 96
  shells • 96
  shortcuts • 96
command line, editing commands • 96
configuration files, log routing • 118
contacting technical support • 3
control access • 68
create access • 68
customer support, contacting • 3

D

daemons
  authorization • 140
  PMDB • 178
  synchronization • 148
  watchdog • 151
database
  administration of properties • 184
  creation • 18
  information • 93
  maintenance • 17
  migration • 25
  purging records • 186
  reports • 187
  utilities • 27, 29, 30
dbdump • 17
dbmgr • 17
dbutil • 17
defclass • 30
delete access • 68
dictimport utility • 31
disabled user accounts • 135
DNS • 60
dual control • 168

E

encrypting audit log records • 118
encryption key • 65
exec access • 68
expired password • 106

F

failed logins • 194
file cache options • 74
filescan access • 68
filter files, PMDB • 178
filtering configuration file • 140
firewall • 118

G

GID, synchronizing • 173, 178
globbing • 251
grace login • 214
groups, extracting • 214

I

initialization files • 93

interception • 138, 139

K

key, encryption • 65

L

lang.ini
  default parameters • 242
  file • 242
  history • 243
  newres tokens • 244
  newusr tokens • 245, 249
  properties tokens • 246
  unix tokens • 249
  usage in selang • 96
log file, PMDB • 170
log route configuration file • 118
login settings • 89
loginpgms.init file • 223
LogRoute API • 116, 118
lookaside database
  files • 60

M

merging audit files • 114
message file • 133

N

named daemon (DNS) • 214
NIS • 60, 118, 214
non-authorized user process • 137
non-eTrust user process • 137

P

password
  changing • 91, 154
  expired • 106
  quality rules • 154
  setting new • 91
  settings • 154
  synchronization • 154
permissions • 203
PMDB
  administration • 164
  creating • 173
  daemon • 178
  filtering • 178
  managing log files • 170
  reports • 187
policy model error log • 178
portmapper • 46, 118, 232
ports, in use • 46

R

rdbdump • 17
read access • 68
rename access • 68
report generation • 20
reporting • 187
retrust programs • 192
root • 201, 227, 235
root password propagation • 154
RPC • 46, 118

S

scripts, database • 23
seaudit • 47
  seos.ini tokens • 47
sebuildla
  seos.ini tokens • 60
sechkey • 65
secons
  seos.ini tokens • 74
secredb • 17
security access • 68
sedb2scr • 17
sedlang • 96
seerr utility • 133
seerrlog
  seos.ini tokens • 88
segrace • 89
segracex • 91
seini • 93
selang
  command history • 96
  command recognition • 96
  editing the command line • 96
  lang.ini tokens • 96
  seos.ini tokens • 96
  word completion • 96
seload
  seos.ini tokens • 104
selock • 106
  lock mode • 106
  monitor mode • 106
  saver mode • 106
  seos.ini tokens • 106

selogmix • 114
  seos.ini tokens • 114
selogrcd
  seos.ini tokens • 116
selogrd
  daemon • 65
  seos.ini tokens • 118
selogrd.cfg • 118
semigrate • 17
semsgtool • 133
  seos.ini tokens • 133
senable • 135
senone • 137
seos.ini file • 227, 230, 232
seos.ini tokens
  seaudit • 47
  sebuildla • 60
  secons • 74
  seerrlog • 88
  selang • 96
  seload • 104
  selock • 106
  selogmix • 114
  selogrcd • 116
  selogrd • 118
  semsgtool • 133
  seosd • 140
  seoswd • 151
  sepass • 154
  sepmdd • 178
  serevu • 194
  ServicePort • 118
  sesu • 201
SEOS_load • 138
SEOS_syscall • 139
seosd • 140, 148
  seos.ini tokens • 140
seoswd • 151
  seos.ini tokens • 151
sepass • 154
  seos.ini tokens • 154
sepmd • 164
sepmdadm • 173
sepmdd • 178
  seos.ini tokens • 178
sepropadm • 184
sepurgdb • 186
sereport • 187
seretrust • 192
serevu • 194
  seos.ini tokens • 194
ServicePort • 118
sessfgate • 200
sesu • 201
  seos.ini tokens • 201
sesudo • 203
setoptions command • 154
setting a new password • 154
seuidpgm • 208
seversion • 211
sewhoami • 212
shortcuts • 96
Simple Network Management Protocol • 118
smart shells • 96
SMTP mail • 118
SNMP traps • 118
special characters in selang • 96
splitting audit files • 114
string matching • 251
su • 201
subscribers, PMDB • 165
support, contacting • 3
synchronization
  daemon • 148
  GID • 173, 178
  UID • 173, 178

T

TCP services, extracting • 214
tcsh • 96
technical support, contacting • 3
tokens • 93
trace
  messages • 219
trcfilter.init file • 140, 229, 239
truncating the update file • 167
trusted programs • 74, 192, 208

U

UID, synchronizing • 173, 178
undefined records • 186
under_NIS_server • 230
Unicenter TNG
  APIs • 200
  asset types • 30
  synchronization • 148
UNIX
  exits • 96

users • 214
update access • 68
update file, truncating • 167
user
permissions • 203
settings • 89
user accounts
enabling • 135
unlocking • 135
user ID • 212
users, extracting • 214
utilities
dictimport • 31
utime access • 68
UxImport • 214

V

version information • 211

W

watchdog daemon, seoswd • 151
wildcard matching • 251
write access • 68

X

XDM • 222, 223, 224, 232
xdmpgms.init file • 224
X-Terminals • 232