Enterprise Audit Shell Administrator’s Guide

H&S Enterprise Readiness

Enterprise Audit Shell Administrator's Guide
H&S Enterprise Readiness

Legal Notice

Table of Contents

List of Tables

List of Figures

Preface

1. What is Enterprise Audit Shell

Enterprise Audit Shell enables organizations to centrally control and audit UNIX shell access. Audit logs are recorded and archived detailing shell input and output, which can be played back and reviewed. Enterprise Audit Shell can be used with all UNIX accounts including administrative, user and application accounts. Enterprise Audit Shell is designed to be scalable and modular so it can be incorporated into a multitude of environments transparently and seamlessly.

2. Terminology and Notation

The terms "Enterprise Audit Shell" and "EAS" will be used interchangeable to refer to the software that accompanies this documentation.

An *administrator* is generally a person who is in charge of installing and running the server. A *user* could be anyone using, or wants to use, any part of EAS. These terms should not be interpreted too narrowly; this documentation set does not have fixed presumptions about system administration procedures.

We use `/usr/local` as the root directory for the installation and `/etc/eash` as the configuration directory. These directories may vary on your site; details can be derived in the *Administrator's Guide*.

In a command synopsis, brackets (`[` and `]`) indicate an optional phrase or keyword. Anything in braces (`{` and `}`) and containing vertical bars (`|`) indicates that you must choose one alternative.

Examples will show commands executed from various accounts and programs. Commands executed from a UNIX shell may be preceded with a dollar sign ("`$`"). Commands executed from particular user accounts such as root are specially flagged and explained.

3. Bug Reporting Guidelines

When you find a bug in EAS we want to hear about it. Your bug reports play an important part in making EAS more reliable because even the utmost care cannot guarantee that every part of EAS will work on every platform under every circumstance.

The following suggestions are intended to assist you in forming bug reports that can be handled in an effective fashion.

If the bug is obvious, critical, or affects a lot of users, the bug will be corrected immediately. It could also happen that we will tell you to update to a newer version to see if the bug happens there.

3.1 Identifying Bugs

Before you report a bug, please read and re-read the documentation to verify that you can really reproduce the problem. If it is not clear in the documentation whether you can do something or not, please report that too; it is a bug in the documentation. If it turns out that the program does something different from what the documentation says, that is a bug. That might include, but is not limited to, the following circumstances:

- A program terminates with a fatal signal or an operating system error message that would point to a problem in the program. (A counterexample might be a "disk full" message, since you have to fix that yourself.)
- A program produces the wrong output for any given input.
- A program refuses to accept valid input (as defined in the documentation).
- A program accepts invalid input without a notice or error message. But keep in mind that your idea invalid input might be our idea of an extension or compatibility with traditional practice.
- EAS fails to execute or install according to the instructions on supported platforms.

Here "program" refers to any executable, not only the backend server.

Being slow or resource hogging is not necessarily a bug. Read the documentation or call support for help tuning your applications.

3.2 What to Report

The most important thing to remember about bug reporting is to state all the facts. Do not speculate what you think went wrong, what "it seemed to do", or which part of the program has a fault. If you are not familiar with the implementation you would probably guess wrong and not help us a bit. And even if you are, educated explanations are a great supplement to but no substitute for facts. Reporting the bare facts is relatively straightforward (you can probably copy and paste them from the screen) but all to often important details are left out because someone thought it does not matter or the report would be understood anyway.

The following items should be contained in every bug report:

- The exact sequence of steps *from program start-up* necessary to reproduce the problem.
- The output you got. Please do not say that it "didn't work" or "crashed". If there is an error message, show it, even if you do not understand it. If the program terminates with an operating system error, say which. If nothing at all happens, say so. Even if the rest of your test case is a program crash or otherwise obvious it might not happen on our platform. The easiest thing is to copy the output from the terminal, if possible.

  > **Note:** In case of fatal errors, the error message reported by the client might not contain all the information available. Please also look at the log output of the server. If you do not keep your server's log output, this would be a good time to start doing so.

- The output you expected is very important to state. IF you just write "This command gives me that output." Or "This is not what I expected.", we might run it ourselves, scan the output, and think it looks OK and is exactly what we expected. We should not have to spend the time to decode the exact semantics behind your commands.
- Anything you did at all differently from the installation instructions.
- Any command line options and other start-up options, including concerned environment variable or configuration files that you changed from the default.
- The EAS version. You can run the command `eash_version` to find out the version.
- Platform information. This includes the kernel name and version, C library, processor, memory information. In most cases it is sufficient to report the vendor and version.

Do not be afraid if your bug report becomes rather lengthy. That is a fact of life. It is better to report everything the first time than have us squeeze the facts out of you.

Chapter 1. Installation Instructions

1.1 Short Version

1.1.1 AIX

EAS supports the native AIX package manager. You may use *smit installp* to install the package or from the command-line:

```
# cp EAS-version.bff /tmp
# geninstall –d /tmp EAS-version
```

1.1.2 Solaris

EAS supports the native Solaris package manager. Use the *pkgadd* command to install the package from the command-line:

```
# pkgadd –d EAS-version.pkg
```

1.1.3 Other platforms

We're adding new support for other platforms everyday. We prefer using the native operating system's package manager where possible. If you're operating system isn't listed above, don't worry – we're working to integrate it into a package.

For all other operating systems we simply supply a tar file containing the EAS binaries and minimum configuration files. This file needs to be extracted from the root / directory and it will install into */usr/local* and */etc/eas*.

```
# cd /
# gunzip –c EAS-version.tar.gz | tar xvf -
```

1.2 Supported Platforms

EAS has been certified to work on the platforms listed below.

OS	Processor	Version	Reported
AIX	RS6000	4.3.3	2005-10-06
AIX	RS6000	5.1	2005-10-06
AIX	RS6000	5.2	2005-10-06
AIX	RS6000	5.3	2005-10-06
FreeBSD	x86	4.11	2005-10-06
HP-UX	PA-RISC	11	2005-10-06
Linux	Alpha	2.2.20	2005-10-06
Linux	AMD64	2.6.9	2005-10-06
Linux	x86	2.4.21	2005-10-06
Mac OS X	PPC	5.5	2005-10-06
Solaris	SPARC	2.6	2005-10-06
Solaris	SPARC	2.7	2005-10-06
Solaris	SPARC	2.8	2005-10-06
Solaris	SPARC	2.9	2005-10-06
Solaris	SPARC	10	2005-10-06

Table 1 - Supported Platforms

Chapter 2. EAS Server Configuration

The EAS Server is configured through the /etc/eas/easd_config configuration file. The configuration file should be owned by root with permissions of 0400. The strict permissions ensure that the configuration files are not tampered with.

```
-r--------    1 root     root        13085 Oct 10 21:42 /etc/eas/easd_config
```

- Comments begin with the pound sign (#) and continue to the end of the current line.
- Options consist of key-value pairs separated by white space.

2.1 Port

Use this option to specify which port EAS will listen on for incoming connections. The default is port 5556 and it's recommended that this value not be changed.

Format:

```
##############################################################################
# Section: TCP/IP
##############################################################################
# Usage: Port { value }
# Value: integer
# Default: 5556
# Description:  Which port to listen for new requests. 1 - 65536.
##############################################################################
Port 5556
```

Figure 1 - EAS Server Configuration: Port

2.2 KeepAlive

Use this option to send TCP keepalive packets to clients.

Format:

```
##############################################################################
# Syntax: KeepAlive { value }
# Value: yes | no
# Default: yes
# Description:  Specifies whether the daemon should send TCP keepalive
#               packets to the client.
##############################################################################
KeepAlive yes
```

Figure 2 - EAS Server Configuration: KeepAlive

2.3 NotificationHook

This option allows the system administrator to install a notification hook in the EAS Server. Upon a successful authentication the NotificationHook will be called and the return code evaluated.

The NotificationHook can be a script or an executable. The NotificationHook will be forked into the background and a clean environment will be set with the following environment variables set:

Environment Variable Name	Description
EASH_EFFECTIVE_GID	Effective GID of the client.
EASH_EFFECTIVE_GR_NAME	Effective group name of the client.
EASH_EFFECTIVE_PW_NAME	Effective user name of the client.
EASH_EFFECTIVE_UID	Effective UID of the client.
EASH_ID	EAS Audit ID (eas_replay and eas_report)
EASH_IP	Client's IP address.
EASH_ORIGINAL_GID	Client's original GID.
EASH_ORIGINAL_GR_NAME	Client's original group name.
EASH_ORIGINAL_PW_NAME	Client's original user name.
EASH_ORIGINAL_UID	Client's original UID.
EASH_REAL_GID	Real GID of the client.
EASH_REAL_GR_NAME	Real group name of the client.
EASH_REAL_PW_NAME	Real user name of the client.
EASH_REAL_UID	Real UID of the client.
EASH_TERMINAL	Client's terminal.

Table 2 - NotificationHook Environment Variables

```
##############################################################################
# Section: Event Notification
##############################################################################
# Usage: NotificationHook { value }
# Value: string
# Default: disabled
# Description:  Specify an executable to be called when a user has connected
#               and authenticated to the server.  This executable will be
#               forked into the background and a clean environment will be
#               set with the following environment variables set:
#
#               EASH_EFFECTIVE_GID      - effective gid
#               EASH_EFFECTIVE_GR_NAME  - effective group name
#               EASH_EFFECTIVE_PW_NAME  - effective username
#               EASH_EFFECTIVE_UID      - effective uid
#               EASH_ID                 - EAS Audit ID (eas_replay)
#               EASH_IP                 - remote IP address
#               EASH_ORIGINAL_GID       - original gid
#               EASH_ORIGINAL_GR_NAME   - original group name
#               EASH_ORIGINAL_PW_NAME   - original username
#               EASH_ORIGINAL_UID       - original uid
#               EASH_REAL_GID           - real gid
#               EASH_REAL_GR_NAME       - real group name
#               EASH_REAL_PW_NAME       - real username
#               EASH_REAL_UID           - real uid
#               EASH_TERMINAL           - original terminal
#
# Note:         This is generally used to send email upon a connection.
# Example       #!/bin/sh
# script:       cat <<EOF | mailx -s "$EASH_ORIGINAL_PW_NAME opened a session"
#               $EASH_ORIGINAL_PW_NAME opened a session as
#               $EASH_EFFECTIVE_PW_NAME from $EASH_IP
#
#               To review this session type `eas_replay $EASH_ID'
#               EOF
#               exit 0
#
##############################################################################
#NotificationHook /usr/libexec/custom_notification_script
```

Figure 3 - EAS Server Configuration: NotificationHook

The notification hook can be used to provide additional authentication. For example the script called by NotificationHook could query an external database or authentication source using the provided environment variables. If the NotificationHook script returns a non-zero return code the requesting client will be denied access. If the return code is zero the client is granted access.

2.4 HookFailureCritical

This option can be used to over-ride the default behavior of NotificationHook. The default behavior is to deny the client access if the return code from NotificationHook is non-zero. Setting the option HookFailureCritical to "no" will always grant the client access regardless of the NotificationHook return code.

Format:

```
################################################################################
# Usage: HookFailureCritical { value }
# Value: yes | no
# Default: yes
# Description:  If the executable specified by NotificationHook has return
#               code of non-zero OR if the executable specified by
#               NotificationHook fails - EAS will terminate the session.
################################################################################
#HookFailureCritical yes
```

Figure 4 - EAS Server Configuration: HookFailureCritical

2.5 HookTimeout

This option is used to set the timeout of NotificationHook. Upon timeout the client is denied access. The default value is 5. The value specified is in seconds.

```
################################################################################
# Usage: HookTimeout { value }
# Value: integer
# Default: 5
# Description:  Use this option to set a timeout on the NotificationHook.
#               Value is in seconds.  Legal values are 1 - 65536.
################################################################################
#HookTimeout 5
```

Figure 5 - EAS Server Configuration: HookTimeout

2.6 Digital Signatures

These options have been placed together under the umbrella "Digital Signatures." Digital Signatures are applied to the EAS audit files that are stored in */var/log/easd*. Using combinations of options a wide variety customization is available. The following options are available under "Digital Signatures:"

Option	Description	Default Value
SignMode	Add file's permissions to the signature.	Yes
SignOwner	Add file's owner to the signature.	Yes
SignInode	Add file's inode to the signature.	No
SignCtime	Add file's ctime to the signature.	No
SignMtime	Add file's mtime to the signature.	No

Table 3 - Digital Signatures

The usage of SignCtime and SignMtime need to be used carefully. These two options are turned off by default because of the sheer strictness it places on the signatures.

SignCtime	SignMtime
The file's ctime is changed by writing or by setting inode information. Setting inode information occurs when you modify the file's: owner group link count mode etc	The file's mtime is changed by file modifications: mknod(2) truncate(2) pipe(2) utime(2) write(2) (or more then zero bytes)

Table 4 - SignCtime vs. SignMtime

Special notes about SignCtime and SignMtime:

SignCtime and SignMtime work great when you need absolute audit log integrity, but these options are too strict when it comes to disaster recovery. If you need to copy the audit logs and database to an alternate server, both the file's mtime and ctime will be changed upon the file transfer, thus invalidating the digital signature.

Format:

```
################################################################################
# Section: Digital Signatures
################################################################################
# Usage: SignMode { value }
# Usage: SignOwner { value }
# Usage: SignInode { value }
# Usage: SignCtime { value }
# Usage: SignMtime { value }
################################################################################
# Value: yes | no
################################################################################
# Default: SignMode yes
# Default: SignOwner yes
# Default: SignInode no
# Default: SignCtime no
# Default: SignMtime no
################################################################################
# Description:  This option will add the file's inode to the SHA1 signature.
#
# Special:      Once these options are set, previous audit logs are subject
#               to the terms of the strictness.  For example if you disable
#               this option all previous audit logs using this option will
#               not be verifiable through EAS Replay.
#
#               You must have a standard with these options and not change it
#               mid-stream.
#
# Note:         It's highly recommended that the default values be not be
#               changed.  The default values represent high security and
#               integrity with the trade-off of being able to copy the audit
#               logs to a different log server.
#
# Option        SignMode        adds the file's permissions to the signature
# details:      SignOwner       adds the file's uid and gid to the signature
#               SignInode       adds the file's inode to the signature
#               SignCtime       adds the file's ctime to the signature
#                               (the file's ctime is changed by writing or by
#                                setting inode information)
#                                       * owner
#                                       * group
#                                       * link count
#                                       * mode
#                                       * etc
#               SignMtime       adds the file's mtime to the signature
#                               (the file's mtime is changed by file
#                                modifications)
#                                       * mknod(2)
#                                       * truncate(2)
#                                       * pipe(2)
#                                       * utime(2)
#                                       * write(2) (of more than zero bytes)
#                               The mtime is not changed for changes in
#                               owner, group, link count or mode.
################################################################################
#SignMode yes
#SignOwner yes
#SignInode no
#SignCtime no
#SignMtime no
```

Figure 6 - EAS Server Configuration: Digital Signatures

2.7 PidFile

Use this option to specify the file that will contain the UNIX PID of the EAS Daemon (easd). The default is */var/log/run/easd.pid*. This file will be used by the EAS stop and start scripts to determine which pid to identify the EAS Daemon (easd) process.

```
################################################################################
# Section: EAS Server Configuration
################################################################################
# Usage: PidFile { value }
# Value: string
# Default: /var/run/easd.pid
# Description:  This file will contain the process ID of the easd daemon.
################################################################################
PidFile /var/run/easd.pid
```

Figure 7 - EAS Server Configuration: PidFile

2.8 SessionDirectory

Specify the directory you wish to store the EAS audit logs and database. The default is */var/log/easd*

```
################################################################################
# Usage: SessionDirectory { value }
# Value: string
# Default: /var/log/easd
# Description:  This directory will store session output and timing
#               information.
################################################################################
SessionDirectory /var/log/easd
```

Figure 8 - EAS Server Configuration: SessionDirectory

The SessionDirectory houses all of the EAS Server Audit Logs. Audit Logs are written to the SessionDirectory in a specific manner:

```
$SessionDirectory/$IP/$ORIGINAL_PW_NAME/$REAL_PW_NAME/$ROWID
```

Figure 9 – Example 1 Session Directory Layout

Variable	Description
$SessionDirectory	The path specified by the option SessionDirectory
$IP	The IP address of the client.
$ORIGINAL_PW_NAME	The original username of the client.
$REAL_PW_NAME	The real username of the client.
$ROWID	The unique identifier of the client used by the database.

Table 5 - SessionDirectory Layout

```
/var/log/easd/127.0.0.1/dhanks/root-1
```

Figure 10 - Example 2 Session Directory Layout

2.9 User

As with any other server daemon that is connected to the world at large, it is advisable to run EAS under a separate user account. This user account should only own the data itself that is being managed by the server, and should not be shared with other daemons. (Thus using the user “nobody” is a bad idea.)

The default is to run with root privileges.

To add a user account to your system, look for a command **useradd** or **adduser**. The user name eas is often used but by no means required.

Use this option to specify the username or UNIX UID the EAS Daemon (easd) should run as. Please note that the UNIX GID will be the default GID of the UID provided as described by /etc/passwd.

Format:

```
################################################################################
# Usage: User { value }
# Value: string | integer
# Default: 0
# Description:  Specify the name or UID of the user easd should run as.
#               Please note that the GID will be the default GID of the UID
#               provided.
#
# Special:      This value needs to be set before EAS Daemon is started for
#               the first time.  It can be changed at a later date under the
#               following conditions:
#
#                       1) StrictSignatures is off
#                       2) You recursively change the owner of the
#                          SessionDirectory and all its files.
#
# Note:         It's recommended you never change this value once EAS has
#               been started for the first time due to the StrictSignatures.
#               Disabling StrictSignatures increases the risk for
#               manipulating audit logs.
################################################################################
User 0
```

Figure 11 - EAS Server Configuration: User

2.10 IdleTimeout

Use this option to specify the shell timeout in seconds. The shell idle time is increased when both no input or output is received. When the shell idle time reaches the defiled IdleTimeout, the client will be disconnected and the idle timeout will be logged.

```
##############################################################################
# Syntax: IdleTimeout { value }
# Value: integer
# Default: 7200
# Description:  Specify idle timeout in seconds.  If the client does not
#               send output or input within the given timeout the server will
#               terminate the connection.  A value of -1 will disable the
#               idle timeout.  Default value of 7200 seconds (2 hours)
##############################################################################
IdleTimeout 7200
```

Figure 12 - EAS Server Configuration: IdleTimeout

2.11 Sync

Use this option to adjust the way the EAS Daemon (easd) writes to disk. The default is Unbuffered / asynchronous due to performance considerations. Please note that Sync needs to be set to "_IOFBF" if you wish to be able to "snoop" upon running audit logs. When Sync is set to "_IOFBF" serious performance problems can occur because each byte needs to be flushed to disk upon each write. The recommended and default setting is unbuffered / asynchronous / _IONBF.

Value	Description
_IONBF	Unbuffered / asynchronous
_IOLBF	Line buffered (writes buffer to disk when a new line is encountered).
_IOFBF	Fully buffered / synchronous. This option isn't recommended and will cause performance problems. The catch is that if you want to "snoop" on running audit logs, this option needs to be enabled.

Table 6 - Synchronization Options

```
##############################################################################
# Usage: Sync { value }
# Value:  _IONBF | _IOLBF | _IOFBF
# Default: _IONBF
# Description:  _IONBF unbuffered
#               _IOLBF line buffered
#               _IOFBF fully buffered
#
#
# Special:      If you want to snoop on active sessions, you need to specify
#               _IOFBF to fully buffer the audit logs.  Using _IONBF or
#               _IOLBF will lead to unexpected results.
#
# Note:         It's recommended that you leave buffering turned off for
#               performance reasons.  _IONBF is the default setting.
##############################################################################
#Sync _IONBF
```

Figure 13 - EAS Server Configuration: Sync

2.12 SyslogFacility

Specify the default syslog facility that EAS Daemon (easd) should write logs to. The default is LOG_AUTH.

SyslogFacility	Description
LOG_AUTH	Security/authorization messages (DEFAULT).
LOG_CRON	Cron and at.
LOG_DAEMON	System daemons without separate facility value.
LOG_FTP	Ftp daemon.
LOG_KERN	Kernel messages.
LOG_LOCAL0 through LOG_LOCAL7	Reserved for local use.
LOG_LPR	Line printer.
LOG_MAIL	Mail.
LOG_NEWS	USENET.
LOG_SYSLOG	Generally reserved for syslogd.
LOG_USER	Generic user-level messages.
LOG_UUCP	UUCP.

Table 7 - Syslog Facilities

Format:

```
##############################################################################
# Section: Syslog Configuration
##############################################################################
# Syntax: SyslogFacility { value }
# Value: string
# Default: LOG_AUTH
# Description:  Specify the syslog facility that easd should log to.
# LOG_AUTH      security/authorization messages (DEFAULT)
# LOG_CRON      cron and at
# LOG_DAEMON    system daemons without seperate facility value
# LOG_FTP       ftp daemon
# LOG_KERN      kernel messages
# LOG_LOCAL0 through LOG_LOCAL7
#               reserved for local use.
# LOG_LPR       line printer
# LOG_MAIL      mail
# LOG_NEWS      USENET
# LOG_SYSLOG    generally reserved for syslogd
# LOG_USER      default genertic user-level messages
# LOG_UUCP      UUCP
##############################################################################
SyslogFacility LOG_AUTH
```

Figure 14 - EAS Server Configuration: SyslogFacility

2.13 SyslogPriority

Specify the default syslog priority that EAS Daemon (easd) should write logs to. The default is LOG_INFO.

Priority	Description
LOG_EMERG	System is unstable.
LOG_ALERT	Action must be taken immediately.
LOG_CRIT	Critical conditions.
LOG_ERR	Errors conditions.
LOG_WARNING	Warning conditions.
LOG_NOTICE	Normal, but significant conditions.
LOG_INFO	Information messages (DEFAULT).
LOG_DEBUG	Debug-level messages.

Table 8 - Syslog Priorities

Please note that EAS Daemon (easd) will always use the following priorities under the following conditions:

Priority	Condition
LOG_CRIT	When a critical error is encountered.
LOG_ERR	When an error has occurred.
LOG_DEBUG	When the LogLevel is set to any of the DEBUG levels.
User-defined SyslogPriority	All other messages.

Table 9 - Syslog Priorities and Conditions

Format:

```
##############################################################################
# Syntax: SyslogPriority { value }
# Value: string
# Default: LOG_INFO
# Description:  Specify the default syslog priority that easd should log
#               with.
# LOG_EMERG     system is unstable
# LOG_ALERT     action must be taken immediately
# LOG_CRIT      critical conditions
# LOG_ERR       error conditions
# LOG_WARNING   warning conditions
# LOG_NOTICE    normal, but significant conditions
# LOG_INFO      information messages (DEFAULT)
# LOG_DEBUG     debug-level messages
#
# Special:      Please note that EAS will always use
#               LOG_CRIT on critical error conditions.
#               LOG_ERR on error conditions.
#               LOG_DEBUG when the LogLevel is set to DEBUG[123]
#               Otherwise the default SyslogPriority will be used.
##############################################################################
SyslogPriority LOG_INFO
```

Figure 15 - EAS Server Configuration: SyslogPriority

2.14 LogLevel

Specify the level of output you wish to receive from the EAS Daemon (easd).

LogLevel	Description
INFO	This is the default – logs information messages to syslog.
DEBUG1	Debug level 1 – logs system calls
DEBUG2	Debug level 2 – logs function calls
DEBUG3	Debug level 3 – logs everything (warning: a lot of output will be generated)

Table 10 - EAS Daemon LogLevels

Format:

```
################################################################################
# Syntax: LogLevel { value }
# Value: string
# Default: INFO
# Description:  Specify the log level for easd.
# INFO          this is the default (SyslogPriority) - logs informational
#               messages to syslog
# DEBUG1        debug level 1 (LOG_DEBUG) - logs system calls
# DEBUG2        debug level 2 (LOG_DEBUG) - logs function calls
# DEBUG3        debug level 3 (LOG_DEBUG) - (warning) logs all function calls
#               and data
################################################################################
LogLevel INFO
```

Figure 16 - EAS Server Configuration: LogLevel

2.15 Cipher

Define permitted SSL ciphers in a colon delimited list. For a complete list see “openssl ciphers” The EAS default is “HIGH:MEDIUM” We suggest that this value not be changed unless you know what you’re doing.

Cipher String	Description
DEFAULT	the default cipher list. This is determined at compile time and is normally ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH. This must be the first cipher string specified.
ALL	all ciphers suites except the eNULL ciphers which musti be explicitly enabled.
HIGH	"high" encryption cipher suites. This currently means those with key lengths larger than 128 bits.
MEDIUM	"medium" encryption cipher suites currently those using 128 bit encryption.
LOW	"low" encryption cipher suites currently those using 64 or 56 bit encryption algorithms but excluding export cipher suites.
EXPORT	export encryption algorithms. Including 40 and 56 bits algorithms.
EXPORT40	specifies 40 bit export encryption algorithms.
EXPORT56	56 bit export encryption algorithms.
NULL	the "NULL" ciphers that is those offering no encryption. Because these offer no encryption at all and are a security risk they are disabled unless explicitly included.
TLSv1 SSLv3 SSLv2	TLS v1.0 SSL v3.0 or SSL v2.0 cipher suites respectively.
DH	cipher suites using DH including anonymous DH.
ADH	anonymous DH cipher suites.
3DES	cipher suites using triple DES.
DES	cipher suites using DES (not triple DES).
RC4	cipher suites using RC4.
RC2	cipher suites using RC2.
IDEA	cipher suites using IDEA.
MD5	cipher suites using MD5.
SHA1	cipher suites using SHA1.

Table 11 - Server SSL Cipher Strings

Format:

```
##############################################################################
# Syntax: Cipher { value1:value2:... }
# Value: string
# Default: HIGH:MEDIUM
# Description:  Define permitted SSL ciphers in a colon delimited list.
#               For a complete list see "openssl ciphers"
##############################################################################
Cipher HIGH:MEDIUM
```

Figure 17 - EAS Server Configuration: Cipher

2.16 Method

Define SSL method to use. The default value is “SSLv3”. It’s recommended that this value not be changed.

Method	Description
TLSv1	TLS version 1.
SSLv2	SSL version 2.
SSLv3	SSL version 3 (DEFAULT).
SSLv23	SSL version 2 and 3 compatibility mode.

Table 12 - Server SSL Methods

Format:

```
##############################################################################
# Section: SSL Configuration
##############################################################################
# Syntax: Method { value1 | value2 | value3 | value4 }
# Value: string
# Default: SSLv3
# Description:  OpenSSL method.
# TLSv1         TLS version 1
# SSLv2         SSL version 2
# SSLv3         SSL version 3
# SSLv23        SSL version 2 and 3 compatibility mode
##############################################################################
Method SSLv3
```

Figure 18 - EAS Server Configuration: Method

2.17 PrivateKey

Specify private key and certificate file. The file should begin with a PEM encoded private key followed by a PEM encoded certificate. The PEM file can contain several certificates that you trust. Use the “eas_mkcerts” utility to generate the public and private keys you will need for the server and client.

```
##############################################################################
# Syntax: PrivateKey { value }
# Value: string
# Default: /etc/eas/certs/server.pem
# Description:  Specify private key and certificate file.  The file should
#               begin with a PEM encoded private key followed by a PEM
#               encoded certificate.  The PEM file can contain serveral
#               certificates that you trust.
##############################################################################
PrivateKey /etc/eas/certs/server.pem
```

Figure 19 - EAS Server Configuration: PrivateKey

2.18 CertificateAuthority

Specify certificate authority file. If you want to trust additional certificates, append them to the file. By default the certificates in the PrivateKey are trusted.

```
##############################################################################
# Syntax: CertificateAuthority { value }
# Value: string
# Default: /etc/eas/certs/root.pem
# Description:  Specify certificate authority file.  If you want to trust
#               additional certificates, append them to the file.  By
#               default the certificates in in the PrivateKey are trusted.
##############################################################################
CertificateAuthority /etc/eas/certs/root.pem
```

Figure 20 - EAS Server Configuration: CertificateAuthority

2.19 RandomFile

If your operating system requires that you specify more random data to feed SSL, use the RandomFile option. The file specified by RandomFile will be read for entropy – the most obvious choice is /dev/urandom. By default this option isn't required.

```
##############################################################################
# Syntax: RandomFile { value }
# Value: string
# Default: disabled
# Description:  Specify the default file to read(2) random data so that
#               OpenSSL can be correctly seeded.  Default is /dev/urandom
##############################################################################
#RandomFile /dev/urandom
```

Figure 21 - EAS Server Configuration: RandomFile

2.20 EGDFile

If your operating system requires that you specify more random data to feed SSL and you do not have */dev/urandom* to use with the RandomFile option, use the EGDfile option. The file specified by EGDFile should point to the UNIX socket created by EGD. By default this option isn't required.

```
##############################################################################
# Syntax: EDGFile { value }
# Value: string
# Default: disabled
# Description:  Specify path to Entropy Gathering Daemon socket.  Use this
#               option if you don't have /dev/urandom or /dev/random
##############################################################################
#EGDFile /var/run/egd-pool
```

Figure 22 - EAS Server Configuration: EGDFile

Chapter 3. EAS Client Configuration

The EAS Client is configured through the /etc/eas/eash_config configuration file. The configuration file should be owned by root with permissions of 0400. The strict permissions ensure that the configuration files are not tampered with.

```
-r--------    1 root     root        13085 Oct 10 21:42 /etc/eas/eash_config
```

- Comments begin with the pound sign (#) and continue to the end of the current line.
- Options consist of key-value pairs separated by white space.

3.1 Port

Use this option to specify which port EAS to use when connecting to a log server. The default is 5554.

Format:

```
################################################################################
# Usage: Port { value }
# Default: 5554
# Value: integer
# Description:  Which port to use when connecting to log server. 1 - 65536.
################################################################################
Port 5554
```

Figure 23 - EAS Client Configuration: Port

3.2 TCPTimeout

When connecting to a remote EAS Server you can specify the number of seconds to wait before timing out. The default is 2 seconds. This value is specified in number of seconds.

Format:

```
################################################################################
# Usage: TCPTimeout { value }
# Default: 2
# Value: integer
# Description:  Specify the number of seconds to wait for a TCP connection
#               to the LogServer.  Default is 2.
################################################################################
TCPTimeout 2
```

Figure 24 - EAS Client Configuration: TCPTimeout

3.3 LogServer

Specify the IP address or hostname of the remote EAS server. Multiple definitions can be used to create a list of EAS servers to be tried in the event a EAS server is unreachable. The default value is "*localhost*" but this is incorrect. You always want to specify a remote EAS server so that the audit logs are stored physically different location. It's also recommended that if EAS is to be used on the EAS server, that the EAS server send its audit logs to a different server.

Format:

```
################################################################################
# Usage: LogServer { value }
# Value: string
# Default: localhost
# Description:  Specify the IP address or hostname of the remote log server.
#               Multiple definitions can be used to create a list of log
#               servers to be tried in the event a log server is unavailable.
#
# Note:         Although the default value is localhost, this isn't correct.
#               You always want to specify a REMOTE LogServer, so that the
#               audit logs are not stored locally and subject to
#               manipulation.
################################################################################
LogServer localhost
#LogServer remotehost1
#LogServer remotehost2
#LogServer disasterrecovery1
#LogServer disasterrecovery2
```

Figure 25 - EAS Client Configuration: LogServer

3.4 DefaultShell

Specify the default shell that is to be used when "*eash*" is to be used as a login shell in */etc/passwd*. This option can be overridden with the symlink facility. To use an alternate shell create a symlink to the absolute path of *eash*.

The format is: *eash_path_to_shell*

For example if you want to use the C-shell (*/bin/csh*), assuming that the absolute path to *eash* is */usr/local/bin/eash* create a symlink with the following command:

```
# ln –s /usr/local/bin/eash /usr/local/bin/eash_bin_csh
```

Figure 26 - Symlink eash to execute another shell

Any shell can be appended – just replaced the character "_" with "/"

```
###############################################################################
# Usage: DefaultShell { value }
# Value: string
# Default: /bin/sh
# Description:  Specify the default shell eash should use when being called
#               as a login shell.
# Special:      This option can be over-rided with the symlink option.  To
#               use an alternate shell create a symlink to the absolute path
#               of eash.
#
#               The format is: eash_path_to_shell
#
#               For example if you want to use the C-shell (/bin/csh):
#               (assuming eash's absolute path is /usr/local/bin/eash)
#               (as root)
#
#               # ln -s /usr/local/bin/eash /usr/local/bin/eash_bin_csh
#
#               Any shell can be appended - just replace "/" with "_"
#
#       Note:   All symlinks must be owned by root.
###############################################################################
DefaultShell /bin/sh
```

Figure 27 - EAS Client Configuration: DefaultShell

3.5 BannerFile

If you wish to display a message of the day or the company security policy upon each shell session use the BannerFile option to specify the file to display. This file must exist on each EAS client server.

```
################################################################################
# Syntax: BannerFile { value }
# Value: string
# Default: disabled
# Description:  Specify the corporate policy or banner file to display
#               before each session.
# Note:         This file must exist on each EAS client.
################################################################################
#BannerFile /etc/corporate-policy
```

Figure 28 - EAS Client Configuration: Bannerfile

3.6 BannerPause

If a banner is to be displayed with BannerFile and you wish to impose a delay before the session is started, use the BannerPause option. The value should be the number of seconds to pause. If the value is -1 the pause is disabled and the user can access the shell immediately.

```
################################################################################
# Syntax: BannerPause { value }
# Default: -1
# Value: integer
# Description:  Specify the number of seconds to pause before the user is
#               allowed to use the session.  Use -1 to disable.
################################################################################
BannerPause -1
```

Figure 29 - EAS Client Configuration: BannerPause

3.7 Cipher

Define permitted SSL ciphers in a colon delimited list. For a complete list see “openssl ciphers” The EAS default is “HIGH:MEDIUM” We suggest that this value not be changed unless you know what you’re doing.

Cipher String	Description
DEFAULT	the default cipher list. This is determined at compile time and is normally ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH. This must be the first cipher string specified.
ALL	all ciphers suites except the eNULL ciphers which musti be explicitly enabled.
HIGH	"high" encryption cipher suites. This currently means those with key lengths larger than 128 bits.
MEDIUM	"medium" encryption cipher suites currently those using 128 bit encryption.
LOW	"low" encryption cipher suites currently those using 64 or 56 bit encryption algorithms but excluding export cipher suites.
EXPORT	export encryption algorithms. Including 40 and 56 bits algorithms.
EXPORT40	specifies 40 bit export encryption algorithms.
EXPORT56	56 bit export encryption algorithms.
NULL	the "NULL" ciphers that is those offering no encryption. Because these offer no encryption at all and are a security risk they are disabled unless explicitly included.
TLSv1 SSLv3 SSLv2	TLS v1.0 SSL v3.0 or SSL v2.0 cipher suites respectively.
DH	cipher suites using DH including anonymous DH.
ADH	anonymous DH cipher suites.
3DES	cipher suites using triple DES.
DES	cipher suites using DES (not triple DES).
RC4	cipher suites using RC4.
RC2	cipher suites using RC2.
IDEA	cipher suites using IDEA.
MD5	cipher suites using MD5.
SHA1	cipher suites using SHA1.

Table 13 - Client SSL Cipher Strings

```
##############################################################################
# Syntax: Cipher { value1:value2:... }
# Value: string
# Default: HIGH:MEDIUM
# Description:  Define permitted SSL ciphers in a colon delimited list.
#               For a complete list see "openssl ciphers"
##############################################################################
Cipher HIGH:MEDIUM
```

Figure 30 - EAS Client Configuration: Cipher

3.8 Method

Define SSL method to use. The default value is "SSLv3". It's recommended that this value not be changed.

Method	Description
TLSv1	TLS version 1.
SSLv2	SSL version 2.
SSLv3	SSL version 3 (DEFAULT).
SSLv23	SSL version 2 and 3 compatibility mode.

Table 14 - Client SSL Methods

Format:

```
##############################################################################
# Section: SSL Configuration
##############################################################################
# Syntax: Method { value1 | value2 | value3 | value4 }
# Value: string
# Default: SSLv3
# Description:  OpenSSL method.
# TLSv1         TLS version 1
# SSLv2         SSL version 2
# SSLv3         SSL version 3
# SSLv23        SSL version 2 and 3 compatibility mode
##############################################################################
Method SSLv3
```

Figure 31 - EAS Client Configuration: Method

3.9 PrivateKey

Specify private key and certificate file. The file should begin with a PEM encoded private key followed by a PEM encoded certificate. The PEM file can contain several certificates that you trust. Use the "eas_mkcerts" utility to generate the public and private keys you will need for the server and client.

```
##############################################################################
# Syntax: PrivateKey { value }
# Value: string
# Default: /etc/eas/certs/server.pem
# Description:  Specify private key and certificate file.  The file should
#               begin with a PEM encoded private key followed by a PEM
#               encoded certificate.  The PEM file can contain serveral
#               certificates that you trust.
##############################################################################
PrivateKey /etc/eas/certs/server.pem
```

Figure 32 - EAS Client Configuration: PrivateKey

3.10 CertificateAuthority

Specify certificate authority file. If you want to trust additional certificates, append them to the file. By default the certificates in the PrivateKey are trusted.

```
##############################################################################
# Syntax: CertificateAuthority { value }
# Value: string
# Default: /etc/eas/certs/root.pem
# Description:  Specify certificate authority file.  If you want to trust
#               additional certificates, append them to the file.  By
#               default the certificates in in the PrivateKey are trusted.
##############################################################################
CertificateAuthority /etc/eas/certs/root.pem
```

Figure 33 - EAS Client Configuration: CertificateAuthority

3.11 RandomFile

If your operating system requires that you specify more random data to feed SSL, use the RandomFile option. The file specified by RandomFile will be read for entropy – the most obvious choice is /dev/urandom. By default this option isn't required.

```
##############################################################################
# Syntax: RandomFile { value }
# Value: string
# Default: disabled
# Description:  Specify the default file to read(2) random data so that
#               OpenSSL can be correctly seeded.  Default is /dev/urandom
##############################################################################
#RandomFile /dev/urandom
```

Figure 34 - EAS Client Configuration: RandomFile

3.12 EGDFile

If your operating system requires that you specify more random data to feed SSL and you do not have */dev/urandom* to use with the RandomFile option, use the EGDfile option. The file specified by EGDFile should point to the UNIX socket created by EGD. By default this option isn't required.

```
##############################################################################
# Syntax: EDGFile { value }
# Value: string
# Default: disabled
# Description:  Specify path to Entropy Gathering Daemon socket.  Use this
#               option if you don't have /dev/urandom or /dev/random
##############################################################################
#EGDFile /var/run/egd-pool
```

Figure 35 - EAS Client Configuration: EGDFile

Chapter 4. SSL

EAS uses SSL for both encryption and authentication. More specifically EAS uses the Public Key Infrastructure (PKI).

4.1 Certificates

A certificate associates a public key with the real identity of an individual, server, or other entity, known as the subject. Information about the subject includes identifying information (the distinguished name), and the public key. It has the identification and signature of the Certificate Authority which issued the certificate, and the period of time during which the certificate is valid. It may have additional information (or extensions) as well as administrative information for the Certificate Authority's use, such as a serial number.

4.2 Certificate Authorities

A Certificate Authority certificate provides assurance that the identity of the holder of the private key of a key-pair is really who the certificate says it is. The Certificate Authority does this by verifying the information in a certificate request before granting the certificate.

4.3 Generating New Certificates

There are many options when generating new certificates. Obviously you could do this yourself or use a third party commercial vendor to supply and sign the certificates. The recommended method is to use OpenSSL itself. EAS comes with utilities to generate certificates and perform all the hard work for you.

4.3.1 Extract EAS Certificate Tools

Locate the package *eas-mkcerts.tar* and extract it in a secure location.

```
<dhanks@localhost>:~$ tar xvf eas-mkcerts.tar
certs/
certs/mkcerts
certs/banners/
certs/banners/1
certs/banners/2
certs/banners/3
certs/banners/4
certs/banners/5
certs/banners/6
certs/banners/7
certs/banners/8
certs/banners/9
certs/banners/10
certs/conf/
certs/conf/client.cnf
certs/conf/root.cnf
certs/conf/server.cnf
```

Figure 36 - Extract EAS Certificate Tools

4.3.2 mkcerts

Change your directory to *certs/* and execute the EAS certificate tool *mkcerts*

```
<dhanks@localhost>:~$ cd certs/
<dhanks@localhost>:~/certs$ ./mkcerts
```

Figure 37 - Execute EAS Certificate Tool

4.3.3 Create Root Certificate Authority

EAS Certificate Tool will now create the Root Certificate Authority. This is a 1024-bit RSA encrypted private key. When prompted for a “PEM pass phrase” enter a strong password that will protect the private key.

```
================================================================================
  ___                _         ___                _     ___   _
 / __|_ _ ___ __ _ _| |_ ___  | _ \___  ___| |_   / __| /_\
| (__| '_/ -_) _` |  _/ -_) |   / _ \/ _ \  _| | (__ / _ \
 \___|_| \___\__,_|\__\___| |_|_\___/\___/\__|  \___/_/ \_\
================================================================================
Generating a 1024 bit RSA private key
.........................................++++++
.++++++
writing new private key to 'CA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figure 38 - Create Root Certificate Authority

4.3.4 Sign Root Certificate Authority

EAS Certificate Tool will now create a self-signed Root Certificate Authority.

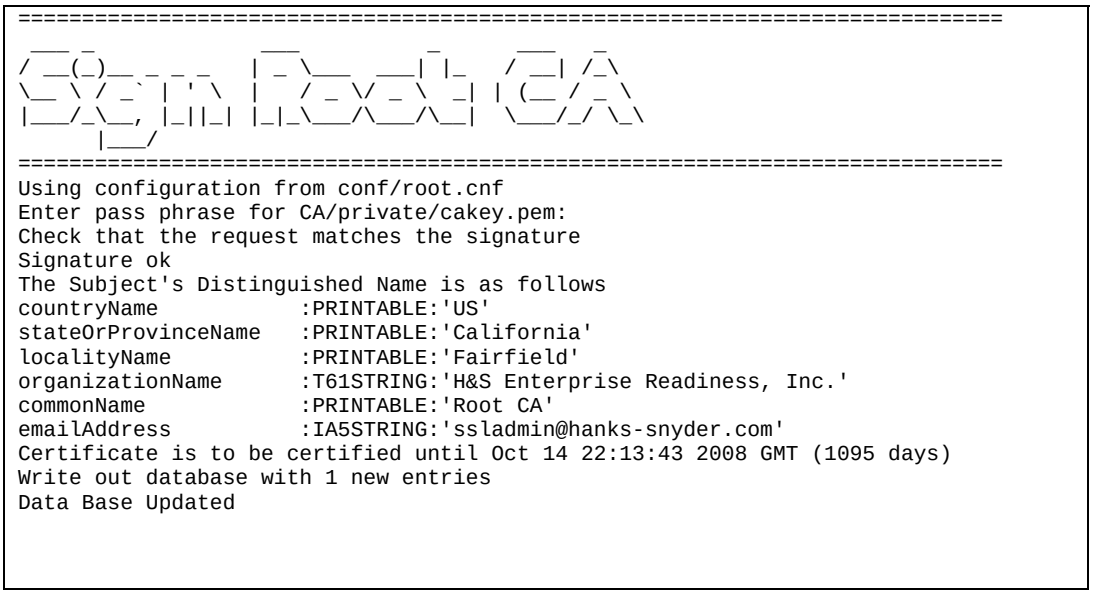

Figure 39 - Sign Root Certificate Authority

4.3.5 Create Client Certificate Signing Request

EAS Certificate Tool will now create a client certificate signing request (CSR).

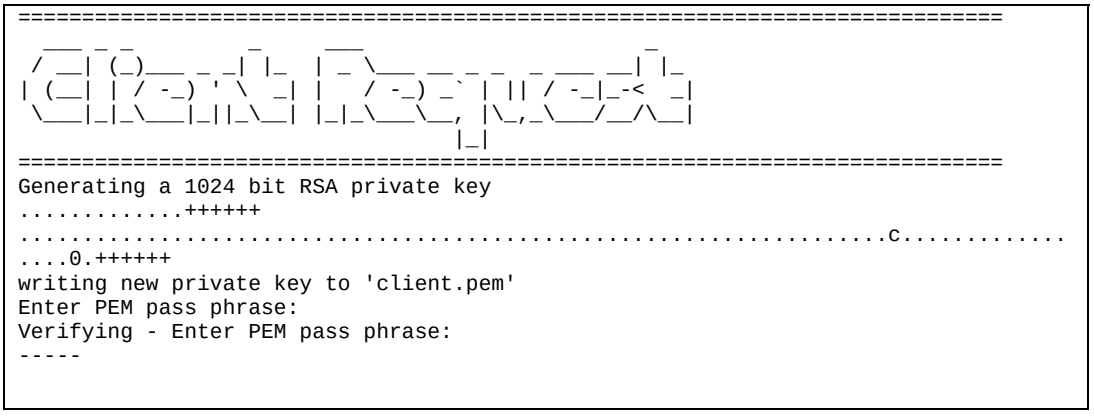

Figure 40 - Create Client CSR

4.3.6 Sign Client Certificate Signing Request

EAS Certificate Tool will now sign the client certificate signing request (CSR).

```
================================================================================
 ___ _                ___ _ _              _     ___                          _
/ __(_)__ _ _ _      / __| (_)___ _ _ _| |_  | _ \___ __ _ _  _ ___ __| |_
\__ \ / _` | ' \  | (__| | / -_) ' \  _| |   / -_) _` | || / -_|_-<  _|
|___/_\__, |_||_|  \___|_|_\___|_||_\__| |_|_\___\__, |\_,_\___/__/\__|
      |___/                                       |_|
================================================================================
Using configuration from conf/root.cnf
Enter pass phrase for CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'California'
localityName          :PRINTABLE:'Fairfield'
organizationName      :T61STRING:'H&S Enterprise Readiness, Inc.'
commonName            :PRINTABLE:'Root CA'
emailAddress          :IA5STRING:'ssladmin@hanks-snyder.com'
Certificate is to be certified until Oct 15 22:13:48 2006 GMT (365 days)
Write out database with 1 new entries
Data Base Updated
```

Figure 41 - Sign Client CSR

4.3.7 Remove Client PEM

EAS Certificate Tool will now remove the client PEM so that a password isn't needed every time *eash* is executed.

```
================================================================================
 ___                          ___ ___ __  __
| _ \___ _ __  _____ _____  | _ \ __|  \/  |
|   / -_) '  \/ _ \ V / -_) |  _/ _|| |\/| |
|_|_\___|_|_|_\___/\_/\___| |_| |___|_|  |_|
================================================================================
Enter pass phrase:
writing RSA key
```

Figure 42 - Remote Client PEM

4.3.8 Create Server Certificate Signing Request

EAS Certificate Tool will now create a server certificate signing request (CSR).

```
================================================================================
 ___                          ___                                  _
/ __| ___ _ ___ _____ _ _   | _ \___ __ _ _  _ ___ __| |_
\__ \/ -_) '_\ V / -_) '_| |   / -_) _` | || / -_|_-<  _|
|___/\___|_|  \_/\___|_|   |_|_\___\__, |\_,_\___/__/\__|
                                   |_|
================================================================================
Generating a 1024 bit RSA private key
.....................................................................++++++
.................................++++++
writing new private key to 'server.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
```

Figure 43 - Create Server CSR

4.3.9 Sign Server Certificate Signing Request

EAS Certificate Tool will now sign the server certificate signing request (CSR).

```
================================================================================
 ___ _                  ___                  ___                          _
/ __(_)__ _ _ _  _    / __| ___ _ ___ _____ _ _  | _ \___ __ _ _  _ ___ __| |_
\__ \ / _` | ' \  \__ \/ -_) '_\ V / -_) '_| |   / -_) _` | || / -_|_-<  _|
|___/_\__, |_||_| |___/\___|_|  \_/\___|_|   |_|_\___\__, |\_,_\___/__/\__|
      |___/                                           |_|
================================================================================
Using configuration from conf/root.cnf
Enter pass phrase for CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'California'
localityName          :PRINTABLE:'Fairfield'
organizationName      :T61STRING:'H&S Enterprise Readiness, Inc.'
commonName            :PRINTABLE:'Root CA'
emailAddress          :IA5STRING:'ssladmin@hanks-snyder.com'
Certificate is to be certified until Oct 15 22:13:53 2006 GMT (365 days)
Write out database with 1 new entries
Data Base Updated
```

Figure 44 - Sign Server CSR

4.3.10 Remove Server PEM

EAS Certificate Tool will now remove the server PEM so that a password isn't needed every time *easd* is executed.

```
================================================================================
 ___                                ___ ___ __  __
| _ \___ _ __  _____ _____  | _ \ __|  \/  |
|   / -_) '  \/ _ \ V / -_) |  _/ _|| |\/| |
|_|_\___|_|_|_\___/\_/\___| |_| |___|_|  |_|
================================================================================
Enter pass phrase:
writing RSA key
```

Figure 45 - Remove Server PEM

4.4 Securing the New Certificates

Now that you have generated new certificates you need to configure and install them. After the generation you should be left with three files:

```
<dhanks@localhost>:~/certs$ ls -l *.pem
-rw-r--r--    1 dhanks   dhanks       3861 Oct 15 15:13 client.pem
-rw-r--r--    1 dhanks   dhanks       2974 Oct 15 15:13 root.pem
-rw-r--r--    1 dhanks   dhanks       3861 Oct 15 15:13 server.pem
```

Figure 46 - New Certificates

The *client.pem* is to be installed on any client using *eash* connecting to an EAS server.

The *server.pem* is to be installed on the EAS Daemon *easd* server.

The *root.pem* is to be installed on both the EAS Daemon *easd* and any clients using *eash* connecting to an EAS server.

These files need to be owned by *root* with the permissions *400*.

4.4.1 chown and chmod

```
<dhanks@localhost>:~/certs$ su
Password:
<root@localhost>:/home/dhanks/certs$ chown 0:0 *.pem
<root@localhost>:/home/dhanks/certs$ chmod 400 *.pem
<root@localhost>:/home/dhanks/certs$ ls -l *.pem
-r--------    1 root     root         3861 Oct 15 15:13 client.pem
-r--------    1 root     root         2974 Oct 15 15:13 root.pem
-r--------    1 root     root         3861 Oct 15 15:13 server.pem
<root@localhost>:/home/dhanks/certs$
```

Figure 47 - Securing the New Certificates

4.5 Installing the New Certificates

4.5.1 client.pem

The certificate *client.pem* should be installed on all EAS clients that will be connecting to the EAS server. Specifically any EAS clients that will be using *eash* should have the *client.pem* certificate installed.

What	Value
File System Location	*/etc/eas/certs/client.pem*
EAS Client Configuration File */etc/eas/eash_config*	*PublicKey /etc/eas/certs/client.pem*

Table 15 - Installing client.pem

4.5.2 server.pem

The certificate *server.pem* should be installed on all EAS Daemon server. Specifically any EAS Servers that will be using *easd* should have the *server.pem* certificate installed.

What	Value
File System Location	*/etc/eas/certs/server.pem*
EAS Server Configuration File */etc/eas/easd_config*	*PublicKey /etc/eas/certs/server.pem*

Table 16 - Installing server.pem

4.5.3 root.pem

Both EAS Clients and Servers must have the *root.pem* installed. Specifically any server that has EAS installed should have the *root.pem* certificate installed.

What	Value
File System Location	*/etc/eas/certs/root.pem*
EAS Client Configuration File */etc/eas/eash_config*	*CertificateAuthority /etc/eas/certs/root.pem*
EAS Server Configuration File */etc/eas/easd_config*	*CertificateAuthority /etc/eas/certs/root.pem*

Table 17 - Installing root.pem

Chapter 5. The EAS Server

The EAS Server is responsible for accepting new client requests and creating audit logs of each client session.

5.1 EAS Server Command-line options

The EAS Server *easd* only supports minimal command-line options. All of the functionality is controlled and configured through the configuration file */etc/eas/easd_config*.

Command-line Option	Description
-h	Show help synopsis.
-v	Display version information.

Table 18 - EAS Server (easd) Command-line Options

5.2 EAS Server Signal Handler

The EAS Server handles signals in different ways providing different functionality. Signals can be sent with the kill(1) command and must be called from *root* or the owner of the *easd* process.

Signal	Number	Functionality
SIGHUP	1	Restart the EAS Server daemon *easd*.
SIGUSR1	10	Change the current LogLevel of *easd*.
SIGINT SIGQUIT SIGTERM SIGABRT SIGPIPE	2 3 15 6 13	Stops EAS Server *easd*. **Special note**: this also terminates the client connection associated with that instance of *easd*.

Table 19 - EAS Server Signal Handler

5.2.1 SIGHUP

If you wish to restart the EAS Server, for example after making changes to the configuration file */etc/eas/easd_config*, execute the command:

```
# kill 1 `cat /var/run/easd.pid`
```

Figure 48 - Example: restarting EAS Server

5.2.2 SIGUSR1

If you wish to change the current LogLevel of the running EAS Server *easd* for debugging purposes execute the command:

```
# kill                                  10 `cat /var/run/easd.pid`
```

Figure 49 - Example: Changing LogLevel of EAS Server

Each time you send the signal SIGUSR1(10) to the EAS Server *easd* the LogLevel will change in a round-robin fasion.

Old LogLevel	New LogLevel
INFO	DEBUG1
DEBUG1	DEBUG2
DEBUG2	DEBUG3
DEBUG3	INFO

Table 20 - LogLevel Round-robin Layout

5.3 EAS Server Logs

All EAS Server logs are written to syslog(2). Refer to chapter 2 sections 2.12 and 2.13 to configure the way EAS Server writes to syslog(2).

5.4 Starting and Stopping the EAS Server

5.4.1 Starting the EAS Server

```
# /usr/local/sbin/easd
```

Figure 50 - Starting the EAS Server

5.4.2 Stopping the EAS Server

```
# kill `cat /var/run/easd.pid`
```

Figure 51 - Stopping the EAS Server

5.5 EAS Server Error Messages

Error Message	Description
HookTimeout %i out of range.	The HookTimeout value is out of range. The value should be between 1 and 65536.
Port %i out of range.	The Port value is out of range. The value should be between 1 and 65536.
Invalid log level.	The LogLevel specified is incorrect. The valid levels are “INFO”, “DEBUG1”, “DEBUG2” and “DEBUG3”
Invalid syslog facility.	The SyslogFacility is incorrect. Please see Chapter 2 section 2.12 for valid syslog facilities.
Invalid syslog priority.	The SyslogPriority is incorrect. Please see Chapter 2 section 2.13 for valid syslog priorities.
Invalid SSL method.	The Method is incorrect. Please see Chapter 2 section 2.16 for valid methods.
Mkdir: <ERROR MESSAGE> <ERRNO>	The specified directory cannot be created. Reference the error message to correct the problem. This generally happens because of permissions.
Chmod: <ERROR MESSAGE> <ERRNO>	The specified file’s mode cannot be changed. Reference the error message to correct the problem. This generally happens because of permissions.
Realloc: <ERROR MESSAGE> <ERRNO>	This generally happens when the system is out of memory.
Strdup: <ERROR MESSAGE> <ERRNO>	This generally happens when the system is out of memory.
Unknown username.	The username specified from the User option from */etc/eas/easd_config* cannot be found in */etc/passwd*
Unknown UID.	The UID specified from the User option from */etc/eas/easd_config* cannot be found in */etc/passwd*
Invalid argument. Yes or no.	The value specified needs to be either “yes” or “no”
Set a timeout over 60 seconds.	You need to specify an IdleTimeout of at least 60 seconds or more.
Bad configuration option	The option specified is incorrect.
Missing argument.	The option specified requires an argument.
Invalid syntax.	The syntax of the configuration file is incorrect.
Couldn’t create lock file: <ERROR MESSAGE> <ERRNO>	The lock file couldn’t be created. This is generally because of permissions. Make sure that the user that executes the EAS Server *easd* has write access to the PidFile.
Lock file is empty.	This means that something other than the EAS Server *easd* created the file and it needs to be removed before EAS Server *easd* can start.
Easd[%ld] is already running.	EAS Server *easd* is already running.
Fork: <ERROR MESSAGE> <ERRNO>	This generally happens when the system is low on resources or out of memory.
Sqlite3_open: <ERROR MESSAGE>	This generally happens when *easd* cannot read and write to the database specified by SessionDirectory.

Table 21 - EAS Server Error Messages

Chapter 6. The EAS Client

The EAS Client is responsible for determining what shell to user; providing a shell to the user while transparently logging all shell activity and sending the audit log to the EAS Server.

6.1 EAS Client Command-line options

The EAS Client *eash* only supports minimal command-line options. All of the functionality is controlled and configured through the configuration file */etc/eas/eash_config*.

Command-line Option	Description
-c	Execute specified command.
-h	Show help synopsis.
-v	Display version information.

Table 22 - EAS Client Command-line Options

6.2 EAS Client Signal Handler

The EAS Client handles a minimum amount of signals, all of which terminate the session.

Signal	Number	Functionality
SIGINT SIGQUIT3 SIGTERM SIGABRT SIGPIPE 13	2 15 6 	Stops EAS Client *eash*.

Table 23 - EAS Client Signal Handler

6.3 Using EAS Client

The EAS Client eash is designed to be used directly from the command-line; as a login shell; and supports remote command execution such as file transfers with *scp* or *rsync*.

6.3.1 EAS Client Environment

Once the EAS Client *eash* has been invoked, the following environment variables are inserted into the shell environment for your convenience.

Environment Variable	Description
EASH_EFFECTIVE_GID	Your effective GID.
EASH_EFFECTIVE_GR_NAME	Your effective group name.
EASH_EFFECTIVE_PW_NAME	Your effective username.
EASH_EFFECTIVE_UID	Your effective UID.
EASH_ORIGINAL_GID	Your original GID.
EASH_ORIGINAL_GR_NAME	Your original group name.
EASH_ORIGINAL_PW_NAME	Your original username.
EASH_ORIGINAL_UID	Your original UID.
EASH_REAL_GID	Your real GID.
EASH_REAL_GR_NAME	Your real group name.
EASH_REAL_PW_NAME	Your real username.
EASH_REAL_UID	Your real UID.

Table 24 - EAS Client Environment Variables

6.3.1 SHELL Environment Variable

If you wish to use a specific shell when using the EAS Client *eash* you can specify that shell through the *SHELL* environment variable. For example if you wish to use the shell */bin/bash* execute the command:

```
$ SHELL=/bin/bash eash
```

Figure 52 - Example 1: Using SHELL environment variable with eash

or

```
$ export SHELL=/bin/bash
$ eash
```

Figure 53 - Example 2: Using SHELL environment variable with eash

or

```
$ setenv SHELL /bin/bash
$ eash
```

Figure 54 - Example 3: Using SHELL environment variable with eash

Special note: the shell must exist in */etc/shells* to be considered valid.

6.3.2 Using EAS Client (eash) as a Login Shell

The EAS Client *eash* is very flexible when it comes to being used as a login shell. To use the EAS Client *eash* as a login shell, simply set the user's login shell from */etc/passwd* to the absolute path of the EAS Client *eash*. For example:

```
dhanks:x:500:500::/home/dhanks:/usr/local/bin/eash
```

Figure 55 - Example /etc/passwd entry using EAS Client as a login shell

When invoked as a login shell in this fashion the default shell is defined in the EAS Client *eash* configuration file */etc/eas/eash_config* with the option DefaultShell.

6.3.3 The Symlink Trick

We understand that every user and application simply cannot use the same DefaultShell because each user and application has specific needs to perform their job. For example the application *SAP* is notorious for using the shell */bin/csh* and *oracle* likes to use either */bin/sh* or */usr/bin/ksh*.

To apply this customization you need to create a symlink to the EAS Client *eash* with the pathname to the shell you wish you use appended to the name replacing the character "/" with "_".

For example to force the user *oracle* to use */usr/bin/ksh*

```
# ln –s /usr/local/bin/eash /usr/local/bin/eash_usr_bin_ksh
```

Figure 56 - Using eash as a login shell over-riding DefaultShell option

Now set oracle's shell to */usr/local/bin/eash_usr_bin_ksh* in */etc/passwd*

```
oracle:x:500:500::/home/oracle:/usr/local/bin/eash_usr_bin_ksh
```

Figure 57 - Examle /etc/passwd entry for oracle using eash as login shell with /usr/bin/ksh as shell

Special note: the shell must exist in */etc/shells* to be considered valid.

6.4 EAS Client Session Movies

The EAS Client *eash* has the ability to record your own movies of your shell session. Just specify the filename you wish to save your session to as the first argument to *eash* and after your session has ended you can play it back with *eas_play*.

For example to create a training video in */tmp/training.eas* type:

```
$ eash /tmp/training.eas
```

Figure 58 - How to make your own session movie

Chapter 7. EAS Database

The database schema used by EAS is fairly straight-forward and easy to use.

7.1 EAS Database Schema

Column Name	Description
id	The unique identifier for the row.
real_uid	The real UID of the client.
real_gid	The real GID of the client.
effective_uid	The effective UID of the client.
effective_gid	The effective GID of the client.
original_uid	The original UID of the client.
original_gid	The original GID of the client.
port	The incoming TCP/IP port from the client.
duration	The duration, in seconds, of the session.
real_pw_name	The real username of the client.
real_gr_name	The real group name of the client.
effective_pw_name	The effective username of the client.
effective_gr_name	The effective group name of the client.
original_pw_name	The original username of the client.
original_gr_name	The original group name of the client.
terminal	The original terminal of the client.
ip	The IP address of the client.
status	The status of the session. R - RUNNING COMPLETE - COMPLETED EJECTED - Client kicked for idling too long.
stype	The sub-type of the session. COMMAND - command was executed, e.g. scp SESSION - eash used from command-line. LOGIN - eash used as a login shell.
method	SSL method used.
cipher	SSL cipher used.
sysname	Client's sysname from uname(2)
nodename	Client's nodename from uname(2)
release	Client's release from uname(2)
version	Client's version from uname(2)
machine	Client's machine from uname(2)
remote_command	Command executed by eash.
pid	PID of easd child.
created	Time and date when the session was created.
modified	Last time and date of last modification.

Table 25 - EAS Database Schema

7.1 EAS Database SQL

This is the SQL command that was used to create the EAS database.

```
CREATE TABLE USER
(
 id     INTEGER PRIMARY KEY AUTOINCREMENT,
 real_uid     INTEGER NOT NULL,
 real_gid     INTEGER NOT NULL,
 effective_uid   INTEGER NOT NULL,
 effective_gid   INTEGER NOT NULL,
 original_uid   INTEGER NOT NULL,
 original_gid   INTEGER NOT NULL,
 port   INTEGER NOT NULL,
 duration     INTEGER NOT NULL,
 real_pw_name   VARCHAR(63) NOT NULL,
 real_gr_name   VARCHAR(63) NOT NULL,
 effective_pw_name      VARCHAR(63) NOT NULL,
 effective_gr_name      VARCHAR(63) NOT NULL,
 original_pw_name       VARCHAR(63) NOT NULL,
 original_gr_name       VARCHAR(63) NOT NULL,
 terminal     VARCHAR(63) NOT NULL,
 ip     VARCHAR(16) NOT NULL,
 status     VARCHAR(63) NOT NULL,
 stype   VARCHAR(63) NOT NULL,
 method     VARCHAR(63) NOT NULL,
 cipher     VARCHAR(63) NOT NULL,
 sysname     VARCHAR(63) NOT NULL,
 nodename     VARCHAR(63) NOT NULL,
 release     VARCHAR(63) NOT NULL,
 version     VARCHAR(63) NOT NULL,
 machine     VARCHAR(63) NOT NULL,
 file_session   VARCHAR(63),
 hash_session   VARCHAR(63),
 dns    VARCHAR(127),
 remote_command     VARCHAR(255),
 pid     INTEGER NOT NULL,
 created      DATETIME,
 modified     DATETIME
);
CREATE TRIGGER INSERT_USER_CREATED AFTER INSERT ON USER
BEGIN
 UPDATE USER SET created = DATETIME('now', 'localtime') WHERE id = new.id;
 UPDATE USER SET modified = DATETIME('now', 'localtime') WHERE id = new.id;
END;
CREATE TRIGGER INSERT_USER_MODIFIED AFTER UPDATE ON USER
BEGIN
 UPDATE USER SET modified = DATETIME('now', 'localtime') WHERE id = new.id;
END;
```

Figure 59 - Table USER SQL Command

Chapter 8. EAS Database Tool

The EAS Database Tool is provided for debugging purposes and creating backups of the database. If you wish to experiment with the database, it's recommended that you experiment with a test and development instance of the database.

8.1 EAS Database Tool Command-line Options

Command-line Option	Description
-init filename	Read/process named file.
-echo	Print commands before execution.
-[no]header	Turn headers on or off.
-column	Set output mode to "column"
-html	Set output mode to "HTML"
-line	Set output mode to "line"
-list	Set output mode to "list"
-separator 'x'	Set output field separator.
-nullvalue 'text'	Set text string for NULL values.
-version	Show version.
-help	Show help synopsis.

Table 26 - EAS Database Tool Command-line Options

8.2 EAS Database Tool Interface

The EAS Database Tool has a very powerful interface that allows the user access to a complete set of SQL-92 compliant commands to interact with the database.

Internal Command	Description
.databases	List names and files of attached databases.
.dump TABLE	Dump the database in an SQL text format.
.echo ON|OFF	Turn command echo on or off.
.exit	Exit EAS Database Tool.
.explain ON|OFF	Turn output mode suitable for EXPLAIN on or off.
.header ON|OFF	Turn display of headers on or off
.help	Show help synopsis.
.import FILE TABLE	Import data from FILE into TABLE.
.indices TABLE	Show names of all indices on TABLE.
.mode MODE TABLE	Set output mode where MODE is one of: csv Comma-separated values. column Left-aligned columns. (See .width) html HTML <table> code. insert SQL insert statements for TABLE. line One value per line. list Values delimited by .separator string. tab Tab-separated values.
.nullvalue STRING	Print STRING in place of NULL values.
.output FILENAME	Send output to FILENAME.
.output stdout	Send output to the screen
.prompt MAIN CONTINUE	Replace the standard prompts.
.quit	Exit EAS Database Tool.
.read FILENAME	Execute SQL in FILENAME.
.schema TABLE	Show the CREATE statements.
.separator STRING	Change separator used by output mode and .import.
.show	Show the current values for various settings.
.tables PATTERN	List names of tables matching a LIKE pattern.
.timeout MS	Try opening locked tables for MS milliseconds.
.width NUM NUM …	Set column widths for "column" mode.

Table 27 - EAS Database Tool Interface Commands

Chapter 9. Backup and Recovery

The EAS Database Tool is used to create and restore the EAS database. It's recommended that the EAS Database be backed up at least once a day during non-peak usage.

9.1 Creating a Backup of the EAS Database

```
# eas_dbtool /var/log/easd/db .dump > /var/log/easd/db.backup
```

Figure 60 - Creating a backup of the EAS Database

As you can see creating a backup is fairly straight forward and doesn't require down-time.

9.2 Creating a Backup of the EAS Audit Logs

The EAS audit logs are just regular UNIX files that can be copied to a different location. We recommend using *find* and *cpio*.

```
# mkdir /var/log/easd/backup/
# cd /var/log/easd && find . ! –name db | cpio –pdum /var/log/easd/backup
40323 blocks
#
```

Figure 61 - Creating abackup of the EAS Audit Logs

9.3 Restoring EAS Database from a Backup

Obviously this goes without saying, but when you perform a database restoration all data that is contained in the previous database is lost.

Make sure that the EAS Server is stopped before you perform a database restoration. It's possible that the database is currently open and being modified.

```
# kill `cat /var/run/easd.pid`
```

Figure 62 - Stopping the EAS Server

Use the EAS Database Tool to import the backup file into a new database.

```
# eas_dbtool /var/log/easd/db
sqlite> .read /path/to/database/backup
sqlite> .quit
#
```

Figure 63 - Restoring EAS Database from a previous backup

9.4 Restoring EAS Audit Logs from a Backup

Once again it should go without saying that any previous data will be over-written when you restore data.

The EAS Audit Logs are regular UNIX files and can be copied into place. Using the example from section 9.2, we just used the commands *find* and *cpio* to perform the backup. Assuming we have a complete EAS Audit Log backup in */var/log/easd/backup* we would execute the following command:

```
# cd /var/log/easd/backup/
# find . ! -name db | cpio -pdum /var/log/easd
40323 blocks
```

Figure 64 - Restoring EAS Audit Logs from previous backup

Chapter 10. EAS Replay

The true audit power of Enterprise Audit Shell is shown with the EAS Replay tool. This tool verifies the audit log signature to certify its integrity and replays the session just as it was originally recorded. EAS Replay offers a wide variety of replay options. Sessions can be played back in their original format; the speed can be interactively increased or decreased; or the sessions can be dumped to *STDOUT* and redirected if you wish to export the session as a file.

10.1 EAS Replay Usage

```
Usage: eas_replay [-a] [-d speed] [-f from] [-gh] [-i IP] [-l limit] [-ns] [-t to] [-r]
[-w maxwait] [-v] [ID]
```

Figure 65 - EAS Replay Usage

10.2 EAS Replay Command-line Options

Command-line Option	Description
-a	Show all sessions.
-d speed	Speed to playback – default is 1.0.
-f from	Limit records by the “From” field.
-g	Group by username.
-h	Display help synopsis.
-i IP	Limit records by the “IP” field.
-l limit	Limit the number of records.
-n	No wait – dump session to *STDOUT*.
-s	Snoop on the session.
-t to	Limit records by the “To” field.
-r	Reverse sort.
-w maxwait	Set the maximum amount of time you wish to wait.
-v	Display version information.

Table 28 - EAS Replay Command-line Options

10.3 Querying Audit Logs

10.3.1 Show All Audit Logs

```
[root@localhost root]# eas_replay -a
================================================================================
Date (s1/\)         From (s2/\)     To              IP              Type   ID
=================== =============== =============== =============== ==== ====
2005-10-16 12:32:54 dhanks          dhanks          127.0.0.1       S       1
2005-10-16 12:32:59 dhanks          dhanks          127.0.0.1       C       2
2005-10-16 12:33:05 dhanks          root            127.0.0.1       S       3
2005-10-16 12:33:09 root            root            127.0.0.1       C       4
2005-10-16 12:33:22 root            root            127.0.0.1       S       5
2005-10-16 12:33:40 dhanks          dhanks          127.0.0.1       C       6
================================================================================
Sessions: 3
Commands: 3
Total: 6
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 6
================================================================================
[root@localhost root]#
```

Figure 66 - Show all audit logs

10.3.2 Show All Audit Logs Grouped by Username

```
[root@localhost root]# eas_replay -ag
================================================================================
Date (s2/\)         From (s1/\)     To              IP              Type   ID
=================== =============== =============== =============== ==== ====
2005-10-16 12:32:54 dhanks          dhanks          127.0.0.1       S       1
2005-10-16 12:32:59 dhanks          dhanks          127.0.0.1       C       2
2005-10-16 12:33:05 dhanks          root            127.0.0.1       S       3
2005-10-16 12:33:40 dhanks          dhanks          127.0.0.1       C       6
2005-10-16 12:33:09 root            root            127.0.0.1       C       4
2005-10-16 12:33:22 root            root            127.0.0.1       S       5
================================================================================
Sessions: 3
Commands: 3
Total: 6
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 5
================================================================================
[root@localhost root]#
```

Figure 67 - Show all audit logs grouped by username

10.3.3 Show Audit Logs by Specific Username

```
[root@localhost root]# eas_replay -f root
================================================================================
Date (s1/\)         From (s2/\)     To              IP              Type   ID
=================== ============== ============== ============== ==== ====
2005-10-16 12:33:09 root            root            127.0.0.1       C       4
2005-10-16 12:33:22 root            root            127.0.0.1       S       5
================================================================================
Sessions: 1
Commands: 1
Total: 2
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 5
================================================================================
[root@localhost root]#
```

Figure 68 - Show audit logs by specific username

10.3.4 Show Audit Logs by Specific IP Address

```
[root@localhost root]# eas_replay -i 127.0.0.1
================================================================================
Date (s1/\)         From (s2/\)     To              IP              Type   ID
=================== ============== ============== ============== ==== ====
2005-10-16 12:32:54 dhanks          dhanks          127.0.0.1       S       1
2005-10-16 12:32:59 dhanks          dhanks          127.0.0.1       C       2
2005-10-16 12:33:05 dhanks          root            127.0.0.1       S       3
2005-10-16 12:33:09 root            root            127.0.0.1       C       4
2005-10-16 12:33:22 root            root            127.0.0.1       S       5
2005-10-16 12:33:40 dhanks          dhanks          127.0.0.1       C       6
================================================================================
Sessions: 3
Commands: 3
Total: 6
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 6
================================================================================
[root@localhost root]#
```

Figure 69 - Show audit logs by specific IP address

10.3.5 Limit Audit Logs by the First 5 Records

```
[root@localhost root]# eas_replay -l 5
================================================================================
Date (s1/\)         From (s2/\)     To              IP              Type   ID
=================== ============== ============== ============== ==== ====
2005-10-16 12:32:54 dhanks          dhanks          127.0.0.1       S       1
2005-10-16 12:32:59 dhanks          dhanks          127.0.0.1       C       2
2005-10-16 12:33:05 dhanks          root            127.0.0.1       S       3
2005-10-16 12:33:09 root            root            127.0.0.1       C       4
2005-10-16 12:33:22 root            root            127.0.0.1       S       5
================================================================================
Sessions: 3
Commands: 2
Total: 5
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 5
================================================================================
[root@localhost root]#
```

Figure 70 - Limit audit logs by the first 5 records

10.3.6 Example of Complicated Query

- From “dhanks”
- To “dhanks”
- From the IP “127.0.0.1”
- Group the results by username
- Limit result set to 2 records.
- Reverse the results.

```
[root@localhost root]# eas_replay -f dhanks -t dhanks -i 127.0.0.1 -g -l2 -r
================================================================================
Date (s2\/)         From (s1\/)     To              IP              Type   ID
=================== ============== ============== ============== ==== ====
2005-10-16 12:33:40 dhanks          dhanks          127.0.0.1       C       6
2005-10-16 12:32:59 dhanks          dhanks          127.0.0.1       C       2
================================================================================
Commands: 2
Total: 2
================================================================================
Playback usage: eas_replay ID [MULTIPLIER] [MAXWAIT]
Note: if you replay an active (R) session, snoop-mode will be enabled.
Example: eas_replay 2
================================================================================
[root@localhost root]#
```

Figure 71 - Example of complicated query

10.4 Viewing an Audit Log

To view an Audit Log simply give *eas_replay* the *ID* you wish to view. The *ID* is obtained from the *eas_replay* result set as described in the previous section 10.3.

```
[root@localhost root]# eas_replay 6
```

Figure 72 - Viewing an audit log

10.5 Dumping an Audit Log to *STDOUT*

To dump an Audit Log to *STDOUT* simply give *eas_replay* the *ID* you wish to view and include the "-n" option for "no wait". The *ID* is obtained from the *eas_replay* result set as described in the previous section 10.3.

```
[root@localhost root]# eas_replay –n 6
```

Figure 73 - Dumping an audit log to *STDOUT*

Chapter 11. EAS Report

Another powerful tool of Enterprise Audit Shell is the reporting functionality. EAS Report creates reports in HTML that is -//W3C//DTD HTML 4.01//EN compliant. EAS Report also takes advantage of Cascading Style Sheets (CSS) so that the power lies in your hands how the reports look and feel.

EAS Report takes the same arguments as EAS Replay, the only difference being that EAS Report outputs HTML only.

It's recommended that you fine-tune what type of audit log criteria you want to report on with EAS Replay before pushing it through EAS Report. The same command-line arguments you use with EAS Replay are the same command-line arguments that you would use with EAS Report.

11.1 EAS Report Command-line Options

Command-line Option	Description
-a	Show all sessions.
-c css _file	Point to another CSS file. (Default is */etc/eas/css/report.css* for inventory reports and */etc/eas/css/detailed.css* for detailed session reports)
-f from	Limit records by the "From" field.
-g	Group results by username.
-h	Display help synopsis.
-I IP	Limit records by the "IP" field.
-l limit	Limit the number of records.
-t to	Limit records by the "To" field.
-r	Reverse sort.
-v	Display version information.

Table 29 - EAS Report Command-line Options

To obtain a detailed report supply EAS Report *eas_report* the *ID* as the first argument.

```
# eas_report 7
```

Figure 74 - Obtaining a detailed report

11.2 Example Reports

11.2.1 Example Inventory Report

Enterprise Audit Shell Audit Report

2005-10-16 18:11:32

Date	From	To	IP	Type	Signature	ID
2005-10-16 12:32:54	dhanks	dhanks	127.0.0.1	Session	Invalid	000000001
2005-10-16 12:32:59	dhanks	dhanks	127.0.0.1	Command	Invalid	000000002
2005-10-16 12:33:05	dhanks	root	127.0.0.1	Session	Verified	000000003
2005-10-16 12:33:09	root	root	127.0.0.1	Command	Verified	000000004
2005-10-16 12:33:22	root	root	127.0.0.1	Session	Verified	000000005
2005-10-16 12:33:40	dhanks	dhanks	127.0.0.1	Command	Verified	000000006
2005-10-16 16:37:25	dhanks	dhanks	127.0.0.1	Session	Verified	000000007

Figure 75 - Example EAS Inventory Report

11.2.2 Example Detailed Report

Enterprise Audit Shell Detailed Report

2005-10-16 19:03:26

ID	000004
Type	COMMAND
Status	COMPLETE
Duration	8 seconds.
Created	2005-10-16 12:33:09
Last modified	2005-10-16 12:33:18
IP	127.0.0.1:34060
SSL Method	SSLv3
SSL Cipher	AES256-SHA
System	Linux 2.4.20-6 (#1 Thu Feb 27 10:01:19 EST 2003)
PID	24365
Terminal	/dev/pts/4
Command	top
Original user	uid=0(root) gid=0(root)
Real user	uid=0(root) gid=0(root)
Effective user	uid=0(root) gid=0(root)
Session	/var/log/easd/127.0.0.1/root/root-4
Signature	8e315e47a8aa3d18a69e28cc9908b2427bde49c0 (signature verified)

Figure 76 - Example EAS detailed report

11.3 Cascading Style Sheets (CSS) Layout

11.3.1 CSS Layout for the Inventory Report

The inventory report makes use of CSS so that the look and feel of the report can be changed on-demand and isn't subject to a rigid layout.

The following CSS classes are defined in the EAS Inventory Report:

CSS Class	Description
hdate	Date Header.
hfrom	From Header.
hto	To Header.
hip	IP Header.
htype	Type Header.
hsignature	Signature Header.
hrowid	Row ID Header.
odd	Denotes an odd numbered row.
even	Denotes an even numbered row.
date	Date data.
from	From data.
to	To data.
ip	IP data.
type	Type data.
rowid	Row ID data.
empty	Last row on the left.
total	Last row on the right.
invalid	Denotes an invalid signature.
verified	Denotes a verified signature.

Figure 77 - CSS layout for the EAS inventory report

11.3.2 CSS Layout for the Detailed Report

The detailed report makes use of CSS so that the look and feel of the report can be changed on-demand and isn't subject to a rigid layout.

The following CSS classes are defined in the EAS Detailed Report:

CSS Class	**Description**
type	Type data.
status	Status data.
duration	Duration data.
created	Created data.
modified	Modified data.
ip	IP data.
method	SSL Method data.
cipher	SSL Cipher data.
system	System UNAME data.
pid	UNIX PID data.
terminal	Terminal data.
command	Command data.
original_pw_name	Original username data.
real_pw_name	Real username data.
effective_pw_name	Effective username data.
session	Session data.
signature	Signature data.
invalid	Denotes an invalid signature.
verified	Denotes a verified signature.

Figure 78 - CSS layout for the EAS detailed report

Chapter 12. EAS Play

For the unprivileged user there is the EAS Play utility. This allows anyone, who has access to an EAS movie, to play it as it was originally recorded. For example if someone records their session with "*eash /tmp/movie.eas*" they could e-mail you the */tmp/movie.eas* file and you could replay their session.

12.1 EAS Play Command-line options.

Command-line Option	Description
-d speed	Speed to playback. Default is 1.0.
-h	Display help synopsis.
-n	No wait – dump output to *stdout*.
-s	Snoop on a running session.
-w maxwait	Maximum time you want to wait on the session.
-v	Display version information.

Table 30 - EAS Play Command-line Options

Index