StorageTek Automated Cartridge System Library Software

Security Guide

Release 8.5
E96386-03
October 2020

ORACLE®

StorageTek Automated Cartridge System Library Software Security Guide, Release 8.5

E96386-03

Contents

Preface

1 Overview

2 Secure Installation

3 Security Features

Preface

This publication describes the security features of Oracle's StorageTek Automated Cartridge System Library Software (ACSLS). It is intended for anyone involved with using security features and secure installation and configuration of ACSLS.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

1
Overview

This section gives an overview of ACSLS and explains the general principles of application security.

Topics include:

- Product Overview
- General Security Principles

> **Note:**
>
> Throughout this document, the Automated Cartridge System Library Software product is referred to as ACSLS.

Product Overview

ACSLS is Oracle's tape library server software that controls one or more StorageTek tape libraries for open systems clients. An Automated Cartridge System (ACS) is a tape library or a group of tape libraries connected through pass-thru-ports (PTPs). ACSLS manages one or more ACSs through "control path" commands sent across a network. The software includes a system administration component, interfaces to client system applications, and library management facilities.

General Security Principles

The following principles are fundamental to using any product securely.

Keeping Software Up To Date

One of the principles of good security practice is to keep all software versions and patches up to date. This document assumes that you are running ACSLS 8.5 or a later release, with all relevant maintenance applied. Running the latest ACSLS release assures that you have the latest enhancements and fixes. Contact Oracle for the latest patches for ACSLS.

Apply all significant security patches to the OS and to services installed with the OS. Please apply these patches selectively, because applying all available updates may install new features and even new OS releases that ACSLS has not been tested with.

Restricting Network Access to Critical Services

Keep both the ACSLS and the libraries that it manages behind a firewall.

Using a private network for TCP/IP communications between ACSLS and tape libraries is recommended.

Following the Principle of Least Privilege

The principle of least privilege states that users should be given the least amount of privilege to perform their jobs. User privileges should be reviewed periodically to determine relevance to current job responsibilities.

On ACSLS, this means that operators who only issue routine commands using `cmd_proc` should login as the `acssa` user. System administrators who login as the `acsss` user also have access to a wider range of utilities and configuration commands. Use of the `acsdb` user ID is not needed for normal operations.

Monitoring System Activity

System security stands on three legs: good security protocols, proper system configuration, and system monitoring. Auditing and reviewing audit records address this third requirement. Each component within a system has some degree of monitoring capability. Follow audit advice in this document and regularly monitor audit records.

Keeping Up To Date on Latest Security Information

Oracle continually improves its software and documentation. Check this document every release for revisions.

2

Secure Installation

This section outlines the planning and implementation process for a secure installation and configuration and describes recommended deployment topologies for ACSLS.

Topics include:

Understand Your Environment

To better understand security needs, consider the following questions:

Which resources need to be protected?

The key resources that ACSLS manages are tape libraries, drives, and cartridges. They need to be protected from inadvertent as well as malicious access. For example, prevent people from mistakenly logging into a different ACSLS server by using different passwords for the ACSLS user IDs on different servers.

From whom are the resources being protected?

You want to protect the tape storage resources from both unauthorized internal and external access.

What will happen if the protections on strategic resources fail?

ACSLS can mount cartridges on tape drives. If a user can connect to the tape drive through the data path, they can read data on the tape if it is not encrypted.

Users who have access to both ACSLS and a tape library can enter and eject cartridges from a tape library.

Recommended Procedure for Securing ACSLS

When securing ACSLS and required infrastructure components, follow this procedure to ensure that ACSLS will continue to function after the changes are made:

1. Install ACSLS.
2. Verify that ACSLS is functioning correctly. Include configuring and auditing libraries, mounting and dismounting tapes, entering and ejecting tapes, and backing-up and restoring the database.
3. Implement the change to increase security.
4. Verify that ACSLS still functions correctly.

Securing ACSLS Internet Communication

This section describes recommendations for deploying ACSLS to secure Internet access.

Topics include:

- Secure ACSLS and Tape Libraries Behind the Corporate Firewall
- ACSLS Firewall Secure Option
- Ethernet Ports Used for ACSLS Communication
- Configuring Firewalls Running on the ACSLS Server

Secure ACSLS and Tape Libraries Behind the Corporate Firewall

ACSLS and the tape libraries it supports should be deployed behind the corporate firewall. If people working remotely need to login to the ACSLS server, they can access it through a VPN.

> **Note:**
>
> If you have an IPv4-based edge firewall, it should be configured to drop all outbound IPv4 protocol 41 packets and UDP port 3544 packets to prevent Internet hosts from using any IPv6-over-IPv4 tunnelled traffic to reach internal hosts.

ACSLS Firewall Secure Option

If client applications, which use ACSLS to mount tapes and manage tape libraries, are separated from ACSLS by a firewall, we recommend enabling the Firewall Secure Option. Even if the client applications are not separated from ACSLS by a firewall, implementing the Firewall Secure Option provides additional ACSLS security by restricting the ports used for communication between ACSLS and its client applications, as shown below. For these reasons, the `CSI_FIREWALL_SECURE` static variable defaults to TRUE in ACSLS 8.5.

For details, refer to the “Firewall Secure Option” appendix in the *ACSLS Administrator’s Guide*.

Ethernet Ports Used for ACSLS Communication

The following ports are used on the ACSLS server. Ensure that any firewalls are configured to allow traffic to these ports. This included firewalls implemented by ipfilter on Solaris or iptables on Linux.

- 22 both directions – used for ssh access.
- 111 portmapper, unless portmapper has been disabled.
- 115 used for SFTP (Secure File Transfer Protocol).
- 5432 default port for internal communication from ACSLS to the PostgreSQL database (the `PGPORT` environment variable for the acsss user ID).
  If port 5432 is taken, the next available higher port number is used.

  > **Note:**
  >
  > Port 5432 only needs to be accessible from localhost (127.0.0.1).

- 7001 and 7002 - used by WebLogic and the ACSLS GUI.
- 30031 or the ACSLS CSI's listening port, set by *CSI_INET_PORT*.
- 50003 port used for internal communication from the ACSLS GUI and Java components to legacy ACSLS processing. This is not configurable.

For client applications to communicate with ACSLS through the ACSAPI, the following ports must be open:

- The client application must be able to communicate with the ACSLS CSI's listening port. This defaults to 30031, and is set by the *CSI_INET_PORT* static variable.
  You can discover which ports are in use by ACSLS to listen for requests from ACSAPI clients with the following command from your Unix shell:

  ```
  rpcinfo -p | egrep "300031 | 536871166"
  ```

  The port IDs will be listed in the last field of the display.
- The ACSAPI client (for example, a NetBackup or SAM-QFS server) sets its fixed incoming port using the *SSI_INET_PORT* environmental variable. Specify a port in the range of 1024-65535, excluding ports 50001 and 50004. The ACSLS server must be able to communicate with this port.

  > **Note:**
  >
  > On an ACSAPI client server, ports 50001 and 50004 are used for AF_INET domain IPC communication to the mini-Event Logger and from client applications to the SSI.

See the "Firewall Secure Option" appendix in the *ACSLS Administrator's Guide* for more details about communication between client applications and ACSLS.

If the XAPI component is installed, the XAPI server uses a fixed listening port to receive incoming TCP requests from ELS clients. The XAPI listening port is defined by the *XAPI_PORT* static variable. *XAPI_PORT* defaults to 50020. It must be between

1024 and 65535, and cannot conflict with any other port used by ACSLS or other applications.

Refer to the XAPI Client Interface appendix in the *ACSLS Adiminstrator’s Guide* for more details about the *XAPI_PORT*. This appendix also provides details about how to display and set the *XAPI_PORT* static variable.

Ports that must be open on an SL8500 or SL3000 library:

50001 – Used for all normal communication between ACSLS and the library.

ACSLS communicates with these ports on an SL8500 or SL3000 library’s 2A and 2B Ethernet connections. If communication from ACSLS to these ports is blocked, ACSLS cannot manage the library.

Configuring Firewalls Running on the ACSLS Server

Besides external firewalls, firewall protection can be implemented on your ACSLS server through `ipfilter` on Solaris or `iptables` on Linux. This describes how to manage these firewalls running on your ACSLS server.

- Managing `ipfilter` on Solaris:
  Consult the man pages for `ipf` and `ipfilter` for detailed information.
  - The `ipfilter` firewall is enabled (disabled) by 'root' using the command:

    ```
    svcadm enable ipfilter  (svcadm disable ipfilter)
    ```

  - To learn the current status of `ipfilter`:

    ```
    svcs ipfilter
    ```

  - Firewall policies are defined in the file: `/etc/ipf/ipf.conf`
    To allow free communication between components on the local host (e.g. between ACSLS and WebLogic or between the GUI and the ACSLS database), include a statement such as:

    ```
    pass in quick from 127.0.0.1 to 127.0.0.1
    ```

    or:

    ```
    pass in quick from 127.0.0.1 to all
    ```

    You must define policies that allow access to all of the ports that are needed for ACSLS. For example, to include a policy allowing remote Web-based browsers to access the ACSLS GUI, you need to open up ports 7001 and 7002.

    ```
    pass in quick from any to any port = 7001
    pass in quick from any to any port = 7002
    ```

    After you discover which ports are used by ACSLS to listen for requests from ACSAPI clients, add '`pass in quick`' statements for each of these ports.

It may be necessary to include a 'pass in quick' statement for the RPC portmapper port, 111.

The last statement in your proposed rule set, "block in from any", states that no traffic should reach the host unless specifically allowed in previous statements.

- Managing `iptables` on Linux:
  - The `iptables` firewall is enabled (disabled) by 'root' using the command:

    ```
    service iptables start (service iptables stop)
    ```

  - To check the status of `iptables`:

    ```
    service iptables status
    ```

  - The policy file for iptables is `/etc/sysconfig/iptables`:
    You must define policies that allow access to all of the ports that are needed for ACSLS. For example, to include a policy that allows remote http/https access to the ACSLS GUI, you should update that file to include exceptions for ports 7001 and 7002 using statements like:

    ```
    -A input -p tcp --dport 7001 -j ACCEPT
    -A input -p tcp --dport 7002 -j ACCEPT
    ```

    After you discover which ports are used by ACSLS to listen for requests from ACSAPI clients, you will need to add exceptions for each of these to the iptables policy file. It may be necessary to include an exception statement for the RPC portmapper port, 111.

Installing and Configuring Solaris

This section describes how to install and configure Solaris securely.

Suggestions include:

- Apply all significant security patches to the OS and to services installed with the OS. Please apply these patches selectively, because applying all available updates may install new features and even new OS releases that ACSLS has not been tested with.
- Disable telnet and rlogin. Use ssh instead. Also disable ftp and use sftp instead. Disable the telnet, rlogin, and ftp services by issuing the following commands as `root`.

  To see all the services, use the `svc` command.

  To disable telnet, rlogin, and ftp, use the following commands:

  ```
  svcadm disable telnet
  svcadm disable rlogin
  svcadm disable ftp
  ```

- Do not disable ssh. You want users to remotely login to the ACSLS using ssh, not telnet or rlogin. Also do not disable sftp.

- ACSLS requires rpc-bind. Do not disable it.
  If Solaris is installed with the Secure by Default option, you must alter a network configuration property for rpc-bind to permit ACSAPI clients to send requests to ACSLS.

  Refer to the *ACSLS Installation Guide*, "Installing ACSLS on Solaris" chapter, "Installing Solaris" section for details.
- Some Ethernet ports on the ACSLS server need to be open for communication with ACSLS. Client applications use specific Ethernet ports for communication with ACSLS, and ACSLS communicates with specific ports on tape libraries. See Ethernet Ports Used for ACSLS Communication for the ports that need to be available for ACSLS communication. On the ACSLS server ensure that ipfilter is configured to allow traffic to the ports used by ACSLS.

Determine your Solaris auditing policy. The “Auditing in Oracle Solaris” section in "Oracle System Administration: Security Services" can help you plan for what events to audit, where your audit logs should be saved, and how you want to review them.

Installing and Configuring Linux

Suggestions to install and configure Linux securely:

- Apply all significant security patches to the OS and to services installed with the OS. Please apply these patches selectively, because applying all available updates may install new features and even new OS releases that ACSLS has not been tested with.
- Make sure that telnet and rlogin are not installed or disabled. Use ssh instead. Also make sure that ftp is not installed or disabled, and use sftp instead.

  To see all services, login as root and issue the following command:

  ```
  service –-status-all
  ```
- To delete services permanently, issue the following command:

  ```
  svccfg delete -f service-name
  ```
- Do not disable ssh. You want users to remotely login to the ACSLS using ssh, not telnet or rlogin. Also do not disable sftp.
- Network services, specifically rpcbind, must be enabled to allow ACSLS client communication.
  When launching rpc on Linux, launch it with the `-i` flag.
- Some Ethernet ports on the ACSLS server need to be open for communication with ACSLS. Client applications use specific Ethernet ports for communication with ACSLS, and ACSLS communicates with specific ports on tape libraries. See Ethernet Ports Used for ACSLS Communication for the ports that need to be available for ACSLS communication. On the ACSLS server ensure that iptables is configured to allow traffic to the ports used by ACSLS.

Auditing Linux Security

Determine your Linux auditing policies. The “Configuring and Using Auditing” section in Oracle Linux: Security Guide for Release 6 can help you plan for what events to audit, where your audit logs should be saved, and how you want to review them.

Some useful logs and commands for auditing Linux security include:

- View `var/log/secure` as root to see the history of login attempts and other access messages.
- The command, `last | more` provides a history of users logged in.
- The `/var/log/audit/audit.log.[0-9]` keeps a log of access attempts that were denied by SELinux. You must be user root to view these.

SELinux Security

ACSLS 8.5 is designed to run in optional Security Enhanced Linux environments. SELinux provides access control to files, directories, and other system resources that go beyond the traditional protection found standard in Unix environments. In addition to owner-group-public permission access, SELinux includes access control based on user role, domain, and context. The agent that enforces access control over all system resources is the Linux kernel.

The root user on a Linux system can set enforcement on or off with the `setenforce` command:

```
setenforce [Enforcing | Permissive | 1 | 0 ]
```

Use Enforcing or 1 to put SELinux in enforcing mode. Use Permissive or 0 to put SELinux in permissive mode.

To view the current system enforcement status, use the command `getenforce`:

Three SELinux policy modules are loaded into the kernel when you install ACSLS: `allowPostgr`, `acsdb`, and `acsdb1`. These modules provide the definitions and enforcement exceptions that are necessary for ACSLS to access its own database and other system resources while SELinux enforcement is active. With these modules installed, you should be able to run normal ACSLS operations, including database operations such as `bdb.acsss`, `rdb.acsss`, `db_export.sh` and `db_import.sh` without the need to disable SELinux enforcement.

If problems occur, you may need to disable SELinux or run in permissive mode. For more information, refer to the "Troubleshooting" appendix in the *StorageTek ACSLS Administrator's Guide*.

Installing and Configuring ACSLS

This section explains how to securely install ACSLS.

Perform a Standard ACSLS Installation

Performing a standard ACSLS installation ensures that you will have all necessary components.

If you are migrating to a latter ACSLS release from a previous ACSLS release, review your settings for dynamic and static variables to see if you want to use more secure options, especially regarding the Firewall Secure Option.

Use Strong Passwords for the ACSLS User IDs

ACSLS requires the ACSLS user IDs: `acsss`, `acssa`, `postgres` and `acsdb`. Choose strong passwords for these IDs, and change the passwords on a regular basis.

Restrict Access to ACSLS Files

ACSLS generally restricts access to the ACSLS files to only `acsls` group, which includes the `acsss`, `acssa`, `acsdb`, and root user IDs. Some database and diagnostic files are only accessible by a single acsls user ID. During DB install user ID `postgres` and group ID `postgres` are used. ACSLS runs with a umask setting of 027.

ACSLS files should not be made world readable or writable. However, restricting access beyond the installation defaults may cause ACSLS functions to fail.

Set ‘root’ as the Effective User ID for Three ACSLS Files

The installation script advises customers that the effective user id of 'root' must be set (`setuid`) in three executable files in the `/export/home/`ACSSS file system:

- `acsss` (This binary must be run with 'root' privileges because it is used to start and stop system services required by the ACSLS application.)
- `db_command` (This binary starts and stops the PostgreSQL database engine that controls and maintains the ACSLS database.)
- `get_diags` (This binary is invoked by a customer to collect comprehensive system diagnostic information that may be needed in the context of a service support call.)

During the installation of ACSLS with `pkgadd`, customers are prompted:

```
Do you want to install these as setuid/setgid files?
```

By answering y to the prompt, you allow these three commands to be run by users in the acsls group, even though the utilities perform certain system operations that require root privileges.

Review Settings for ACSLS Static and Dynamic Variables

The ACSLS static and dynamic variables control the behavior of many ACSLS functions. Set these variables using the `acsss_config` utility. Secure settings for many of these variables are discussed in this document. When the options for a variable are presented by `acsss_config`, replying with a question mark (?) will cause a detailed explanation of the variable to be displayed. This information is also available in the “Setting Variables that Control ACSLS Behavior” chapter of the *ACSLS Administrator’s Guide*.

Configuring WebLogic

ACSLS 8.5 uses WebLogic for its web server. WebLogic is installed with ACSLS.

Refer to *Oracle Fusion Middleware; Understanding Security for Oracle WebLogic Server 11g Release 1 (10.3.6)* for the options for securing a WebLogic server, and the audit trail possibilities with WebLogic.

Use the ACSLS userAdmin.sh Utility to Create and Maintain ACSLS GUI users

The `userAdmin.sh` menu-driven utility is used to administer ACSLS GUI user passwords. You can add users, remove users, list users, and change user passwords. WebLogic must be running to use this utility. If it is not up, this utility starts WebLogic and confirms that it is online before displaying the menu.

The `userAdmin.sh` utility must be run by root, and requires `acsls_admin` authentication. The `acsls_admin` user account is configured during ACSLS installation.

Using the ACSLS GUI

This section describes security issues related to the ACSLS GUI. Topics include:

- Install the Latest JRE Version on GUI Client Systems
- Accessing the ACSLS GUI
- ACSLS GUI Certificates
- SCI Certificate
- ACS Wallet

> **Note:**
>
> Make sure the latest version of the Java Runtime Environment (JRE) is installed on the systems that will use the ACSLS GUI to access ACSLS.

ACSLS GUI Certificates

This section describes creating a GUI certificate for ACSLS which is used by Weblogic. This is different and not to be confused with SCI certificates which are described in a different section. The `AcslsDomain` in WebLogic is accessed using the secure protocol, https. This protocol uses encrypted communication between browser and server using private keys and digital certificates. The following sections describe the options to obtain and create a GUI certificate:

GUI ACSLS Demo Certificate

ACSLS/Weblogic ships with a so-called 'demo' certificate. This provides a minimal level of encryption security, but is insufficient for most needs today. This certificate is overwritten during installation of ACSLS by an automatically generated certificate. Refer to the section below for more information on the GUI generated certificate.

GUI Auto-Generated Certificate

During the `./install.sh` phase of the ACSLS installation, a GUI certificate is automatically generated and installed into Weblogic that is specific to your ACSLS server. This certificate has a 2048 bit key and is self-signed. This generated certificate provides a better level of encryption and security than the default demo certificate, as described in the section above. The GUI generated certificate is also valid for 1824 days.

Most browsers will accept the certificate, however they may present warnings which will require users to accept an exception because the certificates are self-signed. See "GUI Certificates Signed by a Third Party Signing Authority" below for a higher level of security certificates which are not self -signed.

If a customer wishes to re-generate a GUI self-signed certificate and have it automatically installed, simply re-run `./install.sh` on the ACSLS server and respond y when asked to re-generate the certificates.You can also use this procedure to re-generate a GUI certificate if it expires.

> **Note:**
>
> `./install.sh` will re-install weblogic when re-generating and re-installing certificates for the GUI.

Manually Configure a Self-Signed GUI Certificate

The *ACSLS Installation Guide* describes an optional, manual method of creating and installing a customized GUI self-signed certificate. The guide provides a step-by-step method for ACSLS administrators to configure a self-signed digital certificate that is 2048 bits in length. In the section entitled 'Configuring an SSL Encryption Key', this method provides a certificate that is supported on all browsers. Users who access an https site with a self-signed certificate are advised not to proceed with the site unless they have personal knowledge that the web resource is a trusted site. In the context of ACSLS users and the library control server, this level of trust is usually well understood, and in most cases, there is no need for the site to prove its integrity using third-party signature verification. You must use the `acs_cert_wallet` utility to store the GUI certificate password in this case.

GUI Certificates Signed by a Third Party Signing Authority

Each customer must determine whether they need to provide certificate authentication by a third-party signing authority such as Verisign or Entrust.net. The procedure for generating such a signed digital certificate is described in the Oracle online document, Configuring Identity and Trust at:

http://docs.oracle.com/cd/E13222_01/wls/docs92/secmanage/identity_trust.html

Install the Latest JRE Version on GUI Client Systems

Ensure that the latest version of the Java Runtime Environment (JRE) is installed on the systems that will use the ACSLS GUI to access ACSLS.

Accessing the ACSLS GUI

Open a browser and enter a URL with the server hostname or IP address in the following format:

```
https://myAcslsHostName.myDomainName:7002/SlimGUI/faces/Slim.jsp
```

or:

```
 or https://127.99.99.99:7002/SlimGUI/faces/Slim.jsp
```

It is best to use the fully-qualified host name or the IP address of the host machine. Some pages, including the ACSLS help pages, may not display properly if the URL cannot be fully resolved by WebLogic.

If you use http with port 7001, WebLogic will automatically re-route you to https on port 7002.

Since WebLogic is using the secure https protocol, your browser may warn you that the site security certificate has not been registered, and therefore is untrusted. If you are confident that the URL is your local ACSLS machine, you are safe to proceed. At this point, you should see the login screen.

SCI Certificate

This section describes SCI Certificate information for ACSLS. The SCI certificate is used in communication between SL4000 libraries and ACSLS using the SCI protocol which is layered on https protocol. This is not to be confused with GUI certificates which are used for communication between the ACSLS GUI and browsers. See ACSLS GUI Certificates for more information about GUI certificates. The following sub-sections describe various SCI certificate options a user has for ACSLS:

Auto-Generated SCI Certificate

When ACSLS is installed, an SCI certificate customized to the targeted ACSLS server is generated during `./install.sh` phase. This SCI certificate has a 1024 bit key and is self-signed. This SCI generated certificate has a level of encryption and security sufficient for communications with SL4000. The SCI generated certificate is also valid for 1824 days. An SCI certificate is generated at install time regardless of whether SL4000 libraries will be connected to ACSLS. See "SCI Certificates Signed by a Third Party Signing Authority", below, for a higher level certificate security which are not self -signed.

If a customer wishes to re-generate a SCI self-signed certificate and have it automatically installed, simply re-run `./install.sh` on the ACSLS server and respond y when asked to re-generate the certificates. You can also use this procedure to re-generate a SCI certificate if it expires.

Manually Generate an SCI Self-Signed Certificate

If you would like to manually generate a custom SCI Self-signed certificate use the following outline command example. This example will create a self-signed 1024 bit key certificate in a .pem format which can be used by ACSLS (gSOAP) in SL4000 library communications. Use a copy of the file `openssl.cnf` so as not to disrupt the auto-generated SCI certificate procedure.

```
# Generate the gSOAP root CA certificate.......
echo "Generating the gSOAP root CA certificate......."
openssl req -newkey rsa:1024 -sha1 -keyout rootkey.pem -out rootreq.pem
-passout pass:$keyPassword -subj "$subjectName" -days $validDays >>
$generateLogFile 2>&1
openssl x509 -req -in rootreq.pem -sha1 -extfile $keyPath/openssl.cnf
```

```
-extensions v3_ca -signkey rootkey.pem -out cacert.pem -days $validDays
-passin pass:$keyPassword >> $generateLogFile 2>&1
cat cacert.pem rootkey.pem > root.pem

# Create a certificate and signing request
echo "Generating the gSOAP server certificate......."
openssl req -newkey rsa:1024 -sha1 -keyout serverkey.pem -out
serverreq.pem  -passout pass:$keyPassword -subj "$subjectName" -days
$validDays >> $generateLogFile 2>&1

# Sign the certificate with the root CA
echo "Signing the gSOAP server certificate......."
openssl x509 -req -in serverreq.pem -sha1 -extfile $keyPath/openssl.cnf
-extensions usr_cert -CA root.pem -CAkey root.pem -CAcreateserial -out
servercert.pem  \
        -days $validDays -passin pass:$keyPassword >> $generateLogFile
2>&1

# Bundle the CA certificate cacert with the certificate file
echo "Bundling the gSOAP server certificate......."
cat servercert.pem cacert.pem > servertmp.pem
mv -f servertmp.pem servercert.pem

# Bundle certificates with the private key file
cat serverkey.pem servercert.pem > $(ACS_HOME)/data/external/server.pem
```

> **Note:**
>
> The final product is a combined private key and server Certificate in a .pem file format located at: `$(ACS_HOME)/data/external/server.pem`.

SCI Certificates Signed by a Third Party Signing Authority

Each customer site must determine whether they need to provide certificate authentication by a third-party signing authority such as Verisign or Entrust.net. The procedure for generating such a signed digital certificate is described in the Oracle online document, *Configuring Identity and Trust* at:

http://docs.oracle.com/cd/E13222_01/wls/docs92/secmanage/identity_trust.html

ACS Wallet

This section describes information related to username and password information that is specific to SL4000 configuration for usernames, passwords, and certificate passwords.

ACS Usernames and Passwords

With the introduction of authenticated connections between SL4000 libraries and ACSLS, a wallet structure is now included with ACSLS installations. For each ACS and hence each SL4000 library connection to ACSLS, there is an authentication username and password associated to that specific SL4000 library. This authentication is done through username and password, based on that defined on the library side. This username and password must be established on the library side first so that ACSLS can confirm proper authentication during configuration to the SL4000 library. Moreover, the authentication can server as confirmation of the connect library connection when multiple SL4000 libraries are involved. It is also highly recommended that different usernames or passwords are used for each library in a multiple SL4000 library case scenario.

During configuration at `acsss_config` and `config acs new time`, the ACS authentication information is store in a secure wallet in ACSLS. Each ACS can have a unique and different username and password. The wallet itself is encrypted by a AES 256 bit key.

SCI Certificate Password

The SCI Certificate password for decryption is stored in the same wallet structure as the ACS authentication information. If the SCI Certificate is regenerated, the password is also updated. Moreover, if the SCI Certificate is ever manually generated, then the `acs_cert_wallet` utility must be used to update the password information.

3 Security Features

This section describes the specific security mechanisms offered by ACSLS.

Topics include:

- The Security Model
- Configuring and Using Authentication
- Audit Considerations
- Configuring and Using ACSLS Audit Logs
- Configuring and Using Solaris Audit Logs
- Configuring and Using Linux Audit Logs
- Configuring and Using WebLogic Audit Logs

The Security Model

ACSLS security requirements arise from the need to protect data: first, from accidental loss and corruption; and second from deliberate unauthorized attempts to access or alter that data. Secondary concerns include protecting against undue delays in accessing or using data, or even against interference to the point of denial of service.

The critical security features that provide these protections are:

- Authentication – ensures that only authorized individuals get access to the system and data.
- Authorization – provides access control to system privileges and data. This builds on authentication to ensure that individuals only get appropriate access.
- Audit – allows administrators to detect attempted breaches of the authentication mechanism and attempted or successful breaches of access control.

Configuring and Using Authentication

By default on Linux or Solaris, ACSLS users are authenticated by PAM (Pluggable Authentication Modules).

Please refer to the Solaris man pages or the *Linux-PAM System Administrators Guide*.

Users of the ACSLS GUI are authenticated by the embedded LDAP server in WebLogic.

Refer to the document, *Managing the Embedded LDAP Server*:http://docs.oracle.com/cd/E13222_01/wls/docs81/secmanage/ldap.html

Audit Considerations

It is important to keep audited information manageable.

Although auditing is relatively inexpensive, limit the number of audited events as much as possible. Doing so minimizes the performance impact on the execution of audited statements and the size of the audit trail, making it easier to analyze, understand, and manage.

Use the following general guidelines when devising an auditing strategy:

- Evaluate the purpose for auditing:
  After you have a clear understanding of the reasons for auditing, you can devise an appropriate auditing strategy and avoid unnecessary auditing.
- Audit knowledgeably:
  Audit the minimum number of statements, users, or objects required to get the targeted information.

Configuring and Using ACSLS Audit Logs

ACSLS has several logs of information that let you record and inspect ACSLS activity. This section describes the ACSLS Audit logs.

Topics include:

- ACSLS Log Directory
- ACSLS Log/sslm Directory
- Viewing ACSLS Audit Trails from the GUI's Log Viewer
- View System Events from the GUI

> **Note:**
>
> - You can view most ACSLS Audit logs using vi and other editors. System Events can only be viewed by using the ACSLS GUI.
> - Most ACSLS Audit logs can be automatically archived when they reach a customer defined size, and a customer specified number of logs will be retained. To avoid filling the ACSLS filesystem there is a configurable limit to the number of logs that will be retained. If you want to retain more of these log files or retain them on another system, you need to develop your own procedure to archive them in a location that has sufficient space.
> - The size, number of archived logs to retain, and other characteristics of these files are defined by ACSLS dynamic and static variables.

ACSLS Log Directory

The ACSLS log directory is controlled by the LOG_PATH static variable. The default is the $ACS_HOME/log directory. This directory includes these logs:

acsss_event.log

This records messages for significant ACSLS system events, library events, and errors.

When the acsss_event.log reaches a threshold size defined by the LOG_SIZE dynamic variable, it is copied to the event0.log and cleared. During the copy process, the retained event logs are copied into higher numbered retained logs and the highest numbered retained log is overlaid. For example: the event8.log is copied over the event9.log, the event7.log is copied over the event8.log, ..., the event0.log is copied over the event1.log, the acsss_event.log is copied over the event0.log, and the acsss_event.log is cleared. This is controlled by the following variables:

- EVENT_FILE_NUMBER specifies the number of event logs to retain.
- LOG_SIZE specifies the threshold size at which the event log is copied to a retained event log and truncated.

Use the greplog utility to filter the acsss_event log to include or to exclude messages containing specific keywords. See greplog in the “Utilities” Chapter in the *ACSLS Administrator’s Guide* for more details.

Configuration Logs

There are two logs that record details when ACSLS updates the library configuration stored in the ACSLS database. Configuration changes from both acsss_config and Dynamic Config (the config utility) are recorded here.

- acsss_config.log
  Records the details of all configurations or re-configurations of the library(s) that ACSLS supports. The last configuration change is appended to the record of previous configurations.
- acsss_config_event.log
  Records events during the configuration or re-configuration process.

rpTrail.log

Records the response to all requests to ACSLS from ACSAPI clients or cmd_proc, and all requests to the GUI or the SCSI Client interface to logical libraries except for database queries. The information logged includes the requestor, the request, and the request’s time stamp.

rpTrail.log is managed by the following variables:

- *LM_RP_TRAIL* enables this audit trail of ACSLS events. The default is TRUE.
- *RP_TRAIL_LOG_SIZE* specifies the threshold size at which the rpTrail.log is compressed and archived.
- *RP_TRAIL_FILE_NUM* specifies the number of archived rpTrail logs to retain.
- *RP_TRAIL_DIAG* specifies whether the rpTrail messages should include additional diagnostic information. The default is FALSE.

Library Volume Statistics

Records all events affecting volumes (cartridges) in a tape library, including whenever a volume is mounted, dismounted, moved, entered, ejected, or found by audit or Cartridge Recovery. If Library Volume Statistics is enabled, this information is recorded in the `acsss_stats.log`.

Library Volume Statistics is managed by the following variables:

- *LIB_VOL_STATS* enables this Library Volume Statistics. The default is `OFF`.
- *VOL_STATS_FILE_NUM* specifies the number of archived acsss_stats.log files to retain.
- *VOL_STATS_FILE_SIZE* specifies the threshold size at which the acsss_stats.log is archived.

ACSLS Log/sslm Directory

Within the ACSLS log directory, information about the ACSLS GUI and the SCSI Client interface to logical libraries is logged in the `sslm` directory. This directory includes links to WebLogic audit logs. The `sslm` directory includes these logs:

slim_event.g#.log[.pp#]

This records both events from the ACSLS GUI and the SCSI client interface. It includes messages of logical library configuration changes, and SCSI client events.

- The `.g#` is the generation number of this log.
- The `.pp#` is the parallel process number of this log. If there are multiple processes logging at the same time, the logs from the additional processes will be assigned a parallel process number.

smce_trace.log

This traces activity from SCSI clients to ACSLS logical libraries using SCSI Media Changer Interface emulation.

guiAccess.log

This is a link to WebLogic’s access.log. See Configuring and Using WebLogic Audit Logs.

AcslsDomain.log

This is a link to WebLogic’s AcslsDomain.log. See Configuring and Using WebLogic Audit Logs.

AdminServer.log

This is a link to WebLogic’s AdminServer.log. See Configuring and Using WebLogic Audit Logs.

Viewing ACSLS Audit Trails from the GUI's Log Viewer

Access the Log Viewer from the Configuration and Administration section of the GUI Navigation Tree. The Log Viewer displays information combined from the `acsss_event.log` and the `smce_trace.log`.

View System Events from the GUI

You can also view System Events from the Configuration and Administration section of the GUI navigation tree. Every discrete library operation is recorded in the System Events log. Each record in this log contains an event time stamp, an event type, and a description of the event.

Configuring and Using Solaris Audit Logs

Determine your Solaris auditing policy. The "Oracle Solaris Auditing" section in the publication *Oracle System Administration: Security Services* can help you plan for what events to audit, where your audit logs should be saved, and how you want to review them.

If you have not enabled custom Solaris audit trails, these audit trails of logins and Unix commands issued by the `acsss`, `acsdb`, and `acssa` users are available:

- Users who are currently signed on to Unix are recorded in the Unix `utmpx` and past user access is recorded in the `wtmpx` database.
- Use the `last` command to see all access to a user ID (for example, `last acsss`). For more information see the man pages for: `wtmpx`, `last`, and `getutxent`.
- The `.*_history` (that is [dot]*_history) files in a user's home directory record the commands issued by that user.
  For the `acsss` user these may include:
  - `.bash_history`
  - `.psql_history`
  - `.sh_history`

  On Solaris `/var/adm/sulog` records successful and unsuccessful attempts to execute `su` and become superuser or another user.

Configuring and Using Linux Audit Logs

Refer to the "Configuring and Using Auditing" and "Configuring and Using System Logging" sections in the *Oracle Linux: Security Guide* for Release 6 for details about collecting and analyzing audit and system logs.

Configuring and Using WebLogic Audit Logs

Refer to *Oracle Fusion Middleware; Understanding Security for Oracle WebLogic Server 11g Release 1 (10.3.6)* for the options for securing a WebLogic server, and the audit trail possibilities with WebLogic.

WebLogic records access to the ACSLS GUI in the following directory:

```
/export/home/SSLM/AcslsDomain/servers/AdminServer/logs
```

This directory includes the following files:

- access.log
  - There are archived versions named `access.log#####` (for example, `access.log00001`)
  - This provides a detailed audit trail of a GUI user activity.
  - For logins, look for “AcslsLoginForm”.

  > **Note:**
  >
  > There is a link to the access log in: `$ACS_HOME/logs/sslm/guiAccess.log`.

- AcslsDomain.log
  - This reports WebLogic and ACSLS GUI operations.

  > **Note:**
  >
  > There is a link to the access log in: `$ACS_HOME/logs/sslm/AcslsDomain.log`.

- AdminServer.log
  - This reports WebLogic and ACSLS GUI operations.

  > **Note:**
  >
  > There is a link to the access log in: `$ACS_HOME/logs/sslm/AdminServer.log`.

4

Security Considerations for Developers

This section provides information useful to developers developing or supporting applications that use ACSLS to manage Oracle’s StorageTek Tape Libraries.

Enable the Firewall Security on the Client Application’s Server

Restrict the ports used for communication and disable portmapper on the client’s application server by enabling firewall security. Refer to the *CSC Developer’s Toolkit User’s Guide*, "Appendix B: Firewall-Secure Operation."