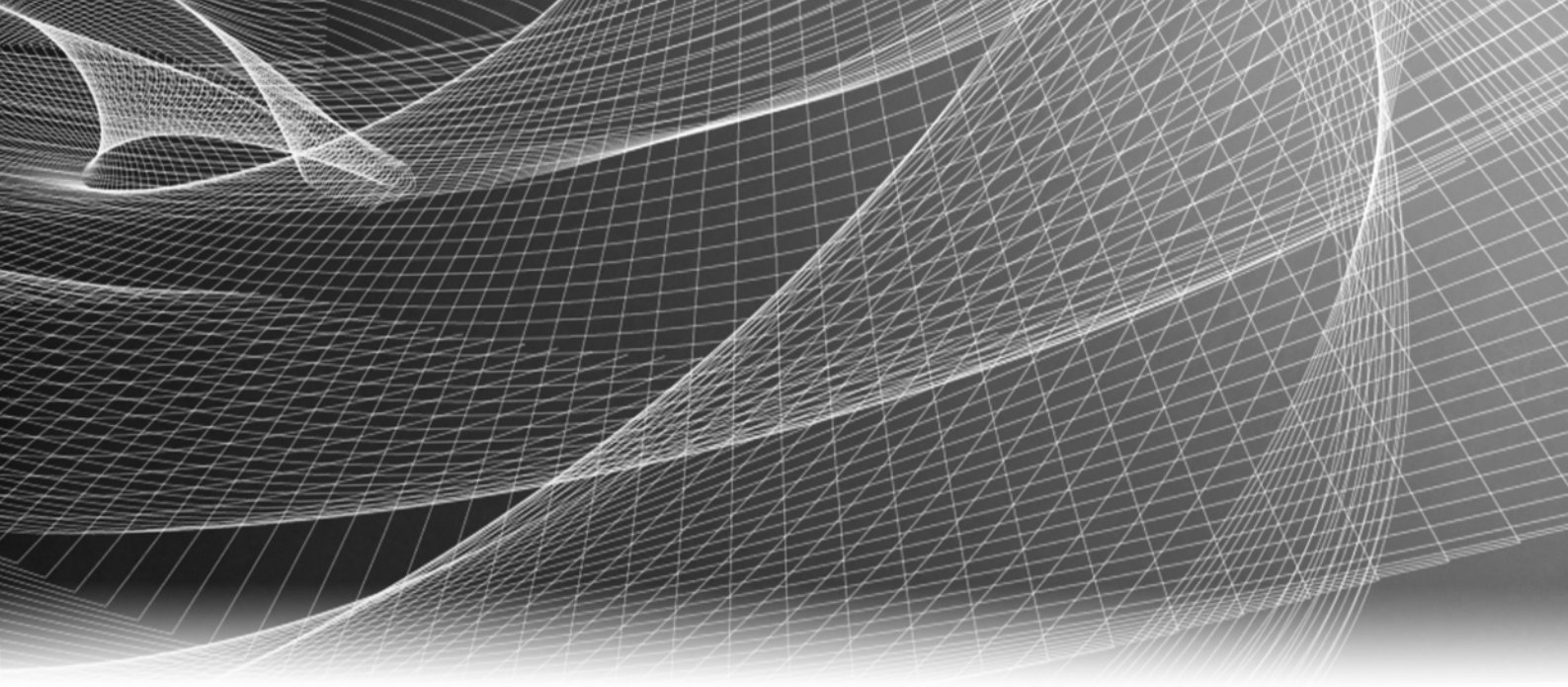

EMC® Symmetrix®

Security Configuration Guide

REV 01

EMC²

Copyright © 2002 - 2013 EMC Corporation. All rights reserved. Published in the USA.

Published April, 2013

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support web site.

CONTENTS

Preface

Chapter 1 Overview

What's new in Solutions Enabler 7.6 9
Other security documentation 9

Chapter 2 Security Deployment Checklists

Solutions Enabler security configuration checklist 12
Unisphere for VMAX security configuration checklist 14

Chapter 3 Solutions Enabler security configuration

SYMAPI home directory 15
Symmetrix access control 16
Host-based access control 16
User-based access control 21
Solutions Enabler log files and settings 22
Log file retention 23
Controlling daemon logging 24
Assigning a custom audit log activity ID 24
Communication (port) security settings 25
Options to modify port settings 27
Solutions Enabler client/server security settings 27
Securing remote connections using SSL 28
Configuring the Solutions Enabler server host 29
Configuring the Solutions Enabler client host 31
Secure session configuration summary 32
Certificate files 33
Managing certificate files 38
Managing the Solutions Enabler server 43
Data security settings 47
Securing Solutions Enabler configuration files 47
Running commands as a non-privileged user 48
Daemon security settings 48
Daemon processes on UNIX 48
Securing host directories on the server host 49
Authorizing daemon connections 50
Securing SRM operations 51
Authorizing SRM daemon connections 51
Setting operating-system-level permissions for SRM 53
Connecting to the SRM database 53
Database start-up options 53
Maintenance: back up Solutions Enabler state 56

Chapter 4 Unisphere for VMAX security configuration

Unisphere access control 58
User-based access control 58

About user authorization 61
Assigning user authorization 62
Unisphere port usage 63
Replacing the self-signed certificate file 63
Other security settings 65
Link-and-launch security 65
Unisphere data security 66
Security alert system 66
Session timeout 66
Root access requirements 66

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your EMC representative if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. New versions of this document might be released on EMC Online Support at www.support.EMC.com. Check EMC Online Support to ensure that you are using the latest version of this document.

Purpose

This guide helps you to securely deploy, use, and maintain Solutions Enabler version 7.6 (Solutions Enabler) and Unisphere® for VMAX® version 1.6 (Unisphere).

Audience

This manual provides both guide and reference information for command-line users and script programmers that focus on configuring and managing storage on Symmetrix arrays.

Related documentation

The following EMC publications provide additional information:

- *EMC Unisphere for VMAX Product Guide*
- *EMC Unisphere for VMAX Installation Guide*
- *EMC Unisphere for VMAX* online help
- *EMC Solutions Enabler Complete Documentation Set*
- *EMC Solutions Enabler Release Notes*
- *EMC Solutions Enabler Installation Guide*
- *EMC Solutions Enabler SYMCLI Command Reference Guide*
- *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide*
- *EMC Solutions Enabler Symmetrix Migration CLI Product Guide*
- *EMC Solutions Enabler TimeFinder Family CLI Product Guide*
- *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide*
- *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*
- *EMC Solutions Enabler Symmetrix SRDF/Star CLI Product Guide*
- *EMC Solutions Enabler Symmetrix SRM CLI Product Guide*
- EMC host connectivity guides for your operating system

Conventions used in this document

EMC uses the following conventions for special notices:

Note: A note presents information that is important, but not hazard-related.

IMPORTANT

An important notice contains information essential to software or hardware operation.

NOTICE

NOTICE is used to address practices not related to personal injury.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, file names, directory names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, file names, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables
‹ ›	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support, product, and licensing information can be obtained on EMC Online Support as described next.

Note: To open a service request through EMC Online Support, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Product information

For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to EMC Online Support (registration required) at:

https://support.EMC.com

Technical support

EMC offers a variety of support options.

Support by Product—EMC offers consolidated, product-specific information on the Web at:

https://support.EMC.com/products

The Support by Product web pages offer quick links to Documentation, White Papers, Advisories (such as frequently used Knowledgebase articles), and Downloads, as well as more dynamic content, such as presentations, discussion, relevant Customer Support Forum entries, and a link to EMC Live Chat.

EMC Live Chat — Open a Chat or instant message session with an EMC Support Engineer.

eLicensing support

To activate your entitlements and obtain your Symmetrix license files, visit the Service Center on http://support.EMC.com, as directed on your License Authorization Code (LAC) letter emailed to you.

For help with missing or incorrect entitlements after activation (that is, expected functionality remains unavailable because it is not licensed), contact your EMC Account Representative or Authorized Reseller.

For help with any errors applying license files through Solutions Enabler, contact the EMC Customer Support Center.

If you are missing a LAC letter, or require further instructions on activating your licenses through EMC Online Support, contact EMC's worldwide Licensing team at licensing@emc.com or call:

- North America, Latin America, APJK, Australia, New Zealand: SVC4EMC (800-782-4362) and follow the voice prompts.
- EMEA: +353 (0) 21 4879862 and follow the voice prompts.

CHAPTER 1
Overview

This guide helps you to securely deploy, operate, and maintain your Symmetrix® deployment using Solutions Enabler version 7.6 (Solutions Enabler) and Unisphere® for VMAX® version 1.6 (Unisphere).

The guide is organized as high-level checklists of security-related activities:

- Solutions Enabler security configuration checklist 12
- Unisphere for VMAX security configuration checklist 14

and detailed procedures to help you complete the activities:

- Solutions Enabler security configuration 15
- Unisphere for VMAX security configuration 57

What's new in Solutions Enabler 7.6

Solutions Enabler version 7.6 includes the following security updates:

- Multiple individual domain names and IP addresses can be specified in a single security certificate, eliminating the need for wildcards in the Common Name, and protecting against man-in-the-middle attacks. See:

  "About wildcards in DNS names" on page 34

  "Rules for wildcards in DNS names" on page 36

  "Managing backward compatibility of certificates" on page 37

- User-defined Activity ID's make it easier to filter specific audit log entries.

  See "Assigning a custom audit log activity ID" on page 24.

- New security SYMAPI_SECURITY_LEVEL control option in the options file replaces the SYMAPI_SERVER_SECURITY_LEVEL option in the options file and the SECURITY_LEVEL option in the daemon options file.

  See "SYMAPI_SECURITY_LEVEL ‹SECURE |ANY |NONSECURE ›" on page 29.

- New SYMAPI_FIPS option enables/disables FIPS 140-2 compliant encryption of Solutions Enabler server/client sessions.

  See "SYMAPI_FIPS ‹ ENABLE|DISABLE›" on page 29.

Other security documentation

This document only describes Solutions Enabler and Unisphere management operations and does not cover:

- Starting/stopping the Solutions Enabler server

Starting/stopping the Solutions Enabler server is described in *EMC Solutions Enabler Installation Guide*.

- Starting/stopping the Unisphere for VMAX server

  Starting/stopping the Unisphere server is described in *EMC Unisphere for VMAX Installation Guide*.

- Managing user authorization for Unisphere

  Unisphere user roles, group roles, user IDs, and managing user account settings are described in *EMC Unisphere for VMAX Product Guide* and the *EMC Unisphere for VMAX* online help.

- Managing authentication for Unisphere

  Login authentication, configuring authentication authorities, and viewing authentication authorities are described in *EMC Unisphere for VMAX Product Guide* and the *EMC Unisphere for VMAX* online help.

- Data access using device masking

  Device masking is described in *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide*.

- Auto-provisioning Groups

  Auto-provisioning groups are described in *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide.*

- IPSec

  EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide describes IPSec capabilities and how to deploy them.

- SMI-S Provider

  EMC SMI-S Provider Release Notes describes how to deploy SMI-S Provider securely.

CHAPTER 2
Security Deployment Checklists

This chapter summarizes the tasks you should perform to improve the security of your deployment.

- Solutions Enabler security configuration checklist 12
- Unisphere for VMAX security configuration checklist 14

Solutions Enabler security configuration checklist

Table 1 Solutions Enabler security configuration checklist (page 1 of 3)

Purpose of security activity	Security activity	Where to learn more
Host-based access control		
Restrict which hosts may access specific functionality.	Enable alternate access IDs on the server host. Enable alternate access IDs on the client host.	"Host-based access control" on page 16 and *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide.*
Use alternate host access IDs to increase security.	Obtain alternate host access IDs. Generate the alternate access ID and store it in the lockbox.	"Host access IDs" on page 16 ""Alternate access IDs" on page 17, "Enabling alternate access IDs" on page 18, and "Enabling alternate access IDs using a passphrase" on page 18
Restrict which hosts and users may perform management operations.	Use Symmetrix access control or user authorization to restrict hosts.	"Host-based access control" on page 16 and"User-based access control" on page 21
User-based access control		
Restrict user access to specific functionality.	Use the `symauth` command to limit which users can access specific functionality.	"User-based access control" on page 21
Log files and settings		
Limit the amount of disk space used by Solutions Enabler log files.	Arrange for log files to be cleaned up automatically. Configure a log rotation/removal schedule.	"Solutions Enabler log files and settings" on page 22, "Log file retention" on page 23, and *EMC Solutions Enabler Installation Guide*
Clean up old logs to conserve disk space.	Configure a log rotation/removal schedule.	"Log file retention" on page 23
Client/server security settings		
Use the Solutions Enabler Server on a remote host that is: • Directly connected to the Symmetrix® array, or • On the Symmetrix Service Processor.	Use client/server mode to connect to the `storsrvd` daemon. **Note:** While performance may be impacted, security can be significantly improved.	• *EMC Solutions Enabler SYMCLI Command Reference Guide* • *EMC Solutions Enabler Installation Guide*
Configure only specific ports to allow access through a firewall or NAT.	If a firewall or NAT router exists between client and server hosts, you may need to allow firewall or NAT router access.	"Communication (port) security settings" on page 25
Certificate files		
Enforce client host security certificate verification.	Set security_clt_secure_lvl=MUSTVERIFY in the `daemon_options` file.	"Solutions Enabler client/server security settings" on page 27 and "security_clt_secure_lvl" on page 30

Table 1 Solutions Enabler security configuration checklist (page 2 of 3)

Purpose of security activity	Security activity	Where to learn more
Replace SYMAPI-generated security certificates with more secure customer-supplied (trusted) certificates.	Replace SYMAPI-generated certificates.	"Replacing SYMAPI-generated certificates" on page 38
On client hosts		
By default, the client event daemon listens on an unused port chosen by the operating system. Modify this port setting if the daemon uses asynchronous events.	Modify the port on which the client event daemon listens.	"Communication (port) security settings" on page 25
Specify the host (HostName) and port (NNNN) on which the server daemon is listening.	For SYMCLI users, modify the `netcnfg` file with the hostnames or IP addresses of your servers.	"Options to modify port settings" on page 27
On server hosts		
By default, the `storsrvd` daemon listens on port 2707.	Modify the port on which the `storsrvd` daemon listens	"Options to modify port settings" on page 27
Configure the `storsrvd` daemon to be automatically started by the operating system.	Start the `storsrvd` daemon automatically.	"Starting the Solutions Enabler server" on page 43

Limit the set of client hosts that the server will accept connections from.	Configure the following:
	• `<SYMAPI_HOME>/config/nethost` file,
	• The following entries in the `<SYMAPI_HOME>/config/daemon_options` file:
	`max_sessions`
	`max_sessions_per_host`
	`max_sessions_per_user`
		"Restricting access to the Solutions Enabler server " on page 43

Restrict the functionality that the `storsrvd` daemon is allowed to perform on behalf of remote client hosts.	Edit the following options in the options file: • SYMAPI_ACC_ADMIN_VIA_SERVER • SYMAPI_ACC_DISPLAY_VIA_SERVER • SYMAPI_ALLOW_SCRIPTS_VIA_SERVER • SYMAPI_CTRL_VIA_SERVER	"Restricting functionality in the Solutions Enabler server" on page 46 and"IBM z/OS-specific behavior for control operations" on page 47
Securing directories and daemons		
Protect the SYMAPI directory and its contents so that only appropriate administrators have write access.	Protect the `<SYMAPI_HOME>/config` directory.	"Running commands as a non-privileged user" on page 48
Protect the `<SYMAPI_HOME>/db` directory to grant non-root users access.	Add users to the `daemon_users` file to allow non-root users (non-administrator on Windows) to access SYMCLI commands. Run SYMCLI commands as a user other than root user/administrator.	
Protect the db directory.	Limit write access privileges to the `<SYMAPI_HOME>/db` directory to authorized users only. All other users should have {no access | read-only access} to this directory.	"Securing Solutions Enabler configuration files" on page 47

Table 1 Solutions Enabler security configuration checklist (page 3 of 3)

Purpose of security activity	Security activity	Where to learn more
On UNIX systems, restrict privileges of the `storsrvd` daemon, (which is network facing).	Run the `storsrvd` daemon as a non-root user.	"Daemon security settings" on page 48
Limit which users have write privileges to the config directory.	Limit access to the `<SYMAPI_HOME>/config` directory to authorized users only. All other users should have limited access (read-only or no access, if possible) to this directory.	"Securing host directories on the server host" on page 49
Minimize injection attacks and other issues.	Limit use of directories in `storsrvd`. Configure the `storsrvd` daemon to allow operations from clients on only certain directories.	
Securing SRM operations		
Limit access to SRM functionality.	Limit permission to the SRM daemon. Edit the common daemon authorization file, `daemon_users`.	"Authorizing SRM daemon connections" on page 51
Limit security exposure by using a database account in SRM with minimal privileges.	Configure a minimally privileged account for SRM database access	
Protect directories and files.	Restrict access privileges for directories and files.	"Setting operating-system-level permissions for SRM" on page 53
Start up and shut down the database server manager instance.	Configure database startup options.	"Database start-up options" on page 53

Unisphere for VMAX security configuration checklist

Table 2 Unisphere for VMAX security configuration checklist

Purpose of security activity	Security activity	Where to learn more
Access control		
Authenticate users with a user account stored on a LDAP-SSL server.	Set up LDAP-SSL authentication.	*Configuring authentication* in the *Unisphere for VMAX product Guide* and online help.
User-based access control		
Restrict the management operations users can perform on a Symmetrix system.	Assign users the minimum access they require.	"User roles" on page 59, "About user authorization" on page 61, and "Assigning user authorization" on page 62
Certificate files		
Replace generated certificates with customer-supplied (trusted) certificates for secure communications.	Replace pre-generated SSL certificates.	"Replacing the self-signed certificate file" on page 63

CHAPTER 3
Solutions Enabler security configuration

This chapter describes security settings for Solutions Enabler 7.6, including:

- Symmetrix access control 16
- Host-based access control 16
- User-based access control 21
- Solutions Enabler log files and settings 22
- Communication (port) security settings 25
- Solutions Enabler client/server security settings 27
- Certificate files 33
- Managing the Solutions Enabler server 43
- Data security settings 47
- Daemon security settings 48
- Securing SRM operations 51

SYMAPI home directory

This guide assumes that the Solutions Enabler ‹*SYMAPI_HOME*› directory is located at:

- Windows: `c:\Program Files\EMC\SYMAPI...`
- UNIX: `/var/symapi/...`
- z/OS: `/var/symapi/...`

Note: By default, the location of ‹*SYMAPI_HOME*› is the same for both z/OS and UNIX.

The location for ‹*SYMAPI_HOME*› on z/OS systems can be changed during installation. The *EMC Solutions Enabler Installation Guide* provides more information about ‹*SYMAPI_HOME*›.

Open VMS file locations are discussed in the *EMC Solutions Enabler Installation Guide*.

Pathnames presented in this document use a UNIX-specific format: forward slashes (/) instead of the backslashes (\) typically used on Windows platforms.

Symmetrix access control

Symmetrix provides two types mechanisms to control access: the `symacl` command and the `symauth` commands.

- The `symacl` command provides host-based access control that can restrict host access to selected sets of devices across multiple Symmetrix arrays.

  `symacl` functionality is also referred to as Symmetrix Access Control.

- The `symauth` command provides user-based authorization that assigns a user or group to a role. Roles limit the management operations users can perform on a Symmetrix array.

IMPORTANT

When configuring Symmetrix Access Control, it is important to:
- Give access rights only to authorized users and groups,
- Assign only the privileges users require to perform their tasks,
- Grant ADMIN rights to a very limited number of users (for example, assign ADMIN rights to only known users and a select administrative group).

The *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* has detailed descriptions of the `symacl` and `symauth` commands.

The *EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide* describes how to set up and perform host-based access control.

Host-based access control

Symmetrix host-based access control limits the management operations that a host can perform. Host-based access control provides highly granular control over management operations.

Host access IDs

Symmetrix Access Control identifies individual management hosts using an access ID. There are two different methods to generate these access IDs:

- **Alternate access ID**: A host's access ID can be generated at random or from a user-defined passphrase, and stored to a secure location on the local disk.

  Alternate access IDs are supported for all platforms.

  Alternate access IDs are available in Solutions Enabler 7.3.2 and higher.

  EMC strongly recommends alternate access IDs on platforms where the hardware-based access ID is derived from a network interface MAC address.

  "Alternate access IDs" on page 17 provides more information on this topic.

- **Hardware-based access ID** (default): The host's access ID is derived from hardware characteristics of that host:

  - On x86 (32-bit Intel), x86_64 (64-bit Intel/AMD), and IA 64 platforms, a network interface MAC address is used.

- On other platforms, characteristics of the host, such as a processor identifier, are used.

"Hardware-based access IDs" on page 17 provides more information on this topic.

IMPORTANT

When MAC addresses generate access IDs, the IDs may be unreliable or ineffective under some circumstances, including clustering environments, virtual environments, or following a hardware change.
For added security on x86 (32-bit Intel), x86_64 (64-bit), IA64, and BS2000 hardware platforms, EMC recommends that you use "Alternate access IDs" instead of hardware-based access IDs.

Hardware-based access IDs

Note: On IBM platforms running in client/server environments with `storsrvd`, you must use job #14MSACL in RIMLIB to generate the unique ID on the host.

Enabling hardware-based access IDs

1. Confirm that the `SYMAPI_ALTERNATE_ACCESS_ID` option is disabled.

   In the ‹*SYMAPI_HOME*›/config/options file, verify that the option is set to DISABLE, as shown below:

   ```
   SYMAPI_ALTERNATE_ACCESS_ID = DISABLE
   ```

2. If the option is not set to `DISABLE`, do one of the following:
   - Edit the entry to set it to `DISABLE`.
   - Remove or comment out the statement.
3. Run the `symacl -unique` command to generate and display an encrypted access ID.
4. Add this access ID to the appropriate access groups.

Alternate access IDs

Alternate access IDs are available for all platforms. When alternate access IDs are enabled, Solutions Enabler can:

- Randomly generate an alternate access ID, as described in "Enabling alternate access IDs" on page 18, or
- Generate an alternate access ID based on a passphrase or file, as described in "Enabling alternate access IDs using a passphrase" on page 18.

Solutions Enabler securely stores this alternate access ID on the local disk. The `symacl` man page provides more information on the `symacl -unique` command.

Enable the `SYMAPI_ALTERNATE_ACCESS_ID` option in the following file:

```
<SYMAPI_HOME>/config/options
```

When the `SYMAPI_ALTERNATE_ACCESS_ID` option is enabled and an alternate access ID is generated, copies of the ID, are securely stored on disk in the following lockbox files:

When enabled and in use, two copies of the alternate access ID (a primary and backup), are securely stored on disk in the following files:

```
<SYMAPI_HOME>/config/lockboxp
<SYMAPI_HOME>/config/lockboxb
```

These files are encrypted. If the primary copy is found to be corrupt, the backup is used. These files hold other security-related information (and keys), in addition to these alternate access IDs.

⚠CAUTION

Do not delete the `lockboxp` and `lockboxb` files, as they hold other security-related information (and keys) in addition to the alternate access IDs.

EMC recommends that you:

- Maintain backup copies of the `lockboxp` and `lockboxb` files, and
- Secure the backups appropriately.

If these files are lost (for example, during a disk replacement or file system re-image), any alternate access IDs contained in those files are lost along with the other security information that Solutions Enabler stores there.

Enabling alternate access IDs

1. Add the following option to the ‹*SYMAPI_HOME*›/config/options file:

   ```
   SYMAPI_ALTERNATE_ACCESS_ID = ENABLE
   ```

2. Run the `symacl -unique` command.

   Solutions Enabler:

   - Recognizes that the above option is set,
   - If one does not already exist for the host, generates an access ID,
   - Securely stores the access ID on the local disk, and
   - Displays the access ID.

   Note: If you run the `symacl -unique` command after enabling the `options` file setting, the new alternate access ID is different than the hardware-based access ID generated prior to enabling this option.
   Any hardware-based access ID previously used to identify this host in an access group must be updated with the new alternate access ID using Solutions Enabler.

3. Add this new, alternate access ID to the appropriate access groups.

   When an access ID is required on this host, the alternate access ID that was stored to disk is used.

Enabling alternate access IDs using a passphrase

1. Add the following option in the ‹*SYMAPI_HOME*›/config/options file:

   ```
   SYMAPI_ALTERNATE_ACCESS_ID = ENABLE
   ```

2. Run the `symacl -unique` command using the `-passphrase` option. The syntax when using the `-passphrase` option is as follows:

   ```
   symacl -unique [-passphrase [Passphrase|-file PassFile]]
   ```

   Note: Passphrases can be from 4 - 1000 characters in length.

   For example, to activate an alternate access ID using a passphrase:

   ```
   symacl -unique -passphrase Passphrase
   ```

   To activate an alternate access ID using a passphrase stored in a file on the local disk:

   ```
   symacl -unique -passphrase -file pathname
   ```

   Note: In client/server mode, the file location is on the client host.

   If no access ID already exists for the host, Solutions Enabler generates an access ID using the passphrase, securely stores it on the local disk, and displays it.

3. Add this new, alternate access ID to the appropriate access groups.

   When an access ID is required on this host, the alternate access ID that was stored to disk is used.

Disabling an alternate access ID

1. Do one of the following to disable the `SYMAPI_ALTERNATE_ACCESS_ID` option in the options file:

   - Change the following setting in the `options` file to:

     ```
     SYMAPI_ALTERNATE_ACCESS_ID = DISABLE
     ```

   - Remove the line from the `options` file.

2. Run the `symacl -unique` command.

   This command recognizes that the option was reset, and disables the alternate access ID stored on the local disk. A copy of the access ID remains securely stored on disk but is not used. If you choose to enable the option in the future, the same value is used.

Changing a host's alternate access ID

Note: Solution Enabler access control changes must be made from an administrative host with ADMIN rights to the array and rights to make `symacl` changes.

If you only have one such administrative host, and you change its alternate access ID, once that change is made, the host can no longer make access control changes because the new access ID is not yet in an access group.

EMC recommends that you enable a second administrative host prior to attempting to change a host’s alternate access ID.

For example, to change the access ID for Host-1:

1. Log in to another administrative host, such as Host-2.

2. Remove any existing Host-1 definitions from the access group for all Symmetrix arrays to which Host-1 has access.
3. From Host-1, follow the steps outlined in "Enabling alternate access IDs" on page 18 or "Disabling an alternate access ID" on page 19 to enable or disable the alternate access ID mechanism and obtain a new access ID.
4. From Host-2, add Host-1 back into its access group using its new access ID to any Symmetrix arrays to which it requires access.

Access control in client/server mode

Anyone with access to Symmetrix management software can execute functions on any Symmetrix device. Many applications can issue management commands to any device in a Symmetrix deployment.

Such shared systems may be vulnerable to a host accidentally or intentionally tampering with another's devices. To prevent this, use the `symacl` command to set up and restrict host access to defined sets of devices across the Symmetrix arrays.

The *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* describes how to set up and perform host-based access control.

By default, client/server mode operations are executed on the server host using the access ID of the server. Access control checks are performed against the rules established for the server host, regardless of which client host initiated the operations.

You can use the access ID of the client host instead of the server host to perform this check. When this is enabled, access control rules must be established for, and checked against, the client hosts from which the operations are issued.

To use the access ID of the client host, you must make changes in the `options` file on the client and the server host, as explained in the following sections.

Server host options file settings

On the server host, the `SYMAPI_USE_ACCESS_ID` option controls the source of the access ID used for the client/server sessions:

```
SYMAPI_USE_ACCESS_ID = CLIENT | SERVER | ANY
```

The behavior of this option is as follows:

- `CLIENT` - An access ID supplied by the client host is used. If the client did not provide an access ID, operations fail. This can occur if the client is running a version of Solutions Enabler lower than 7.4 or if this functionality was not configured on the client.
- `SERVER` (default) - The server always uses its own access ID and ignores an access ID, if any, provided by the clients.
- `ANY` - The server uses an access ID provided by a client. If one is not provided, the server uses its own access ID.

Client host options file settings

On the client host, the `SYMAPI_ALTERNATE_ACCESS_ID` option must be enabled to use this alternate access IDs:

```
SYMAPI_ALTERNATE_ACCESS_ID = ENABLE
```

Additionally, you must set the following option to control whether the client can send its own access ID to the server:

```
SYMAPI_CLIENT_SIDE_ACCESS_ID = ENABLE | DISABLE
```

The behavior of this option is as follows:

- `ENABLE` - The client sends its access ID to the server in client/server mode.
- `DISABLE` (default) - The client does not send its access ID to the server in client/server mode.

IMPORTANT

After you enable the `SYMAPI_ALTERNATE_ACCESS_ID` and `SYMAPI_CLIENT_SIDE_ACCESS_ID` options, you must run the `symacl -unique` command on the client host to generate the access ID and store it in the lockbox on the client host.

User-based access control

Symmetrix user authorization assigns individual users to roles. These user roles limit the management operations that users can perform. User-based controls are less granular than host-based controls.

Use the `symauth` command, SMC (Symmetrix Management Console), or Unisphere® for VMAX® 1.0 or higher, to assign users to management roles to restrict the types of operations they can perform.

Solutions Enabler does not support an explicit authentication mechanism for users. Solutions Enabler uses the credentials that users supply when logging onto the local system, as provided by the operating system.

SMC or Unisphere for VMAX 1.0 or higher pass the user's authenticated identity to Solutions Enabler.

EMC Solutions Enabler Symmetrix Array Management CLI Product Guide provides more information about the `symauth` command.

EMC Unisphere for VMAX Product Guide provides more information about managing Unisphere user accounts.

User identification

Internally, Solutions Enabler represents a user identity as a string assembled from the user's name and authentication source. The possible encodings are:

`H:HostName\UserName`	A user authenticated by the local operating system.
`D:DomainName\UserName`	A user authenticated by a specific domain on Windows.
`L:ServerName\UserName`	A user authenticated by an LDAP server. (SMC or Unisphere for VMAX 1.0 and higher.)
`C:HostName\UserName`	A user authenticated by the private SMC or Unisphere for VMAX 1.0 or higher authentication service on some host. (SMC or Unisphere for VMAX 1.0 and higher.)

Solutions Enabler uses these identities in a number of ways. A user name is included in records written to the Symmetrix array's secure Audit Log. This identifies the user that initiated the activity being logged. A user identity is the basis for optional user authorization rules that restrict management access to Symmetrix arrays.

Support for all user groups

Solutions Enabler checks all groups to which a user belongs. During rights checking, each user group is examined for a role mapping and the "or" of all rights granted to each group.

The `symauth show -username` command displays all groups to which a user belongs.

Authorization rules can be configured for a Symmetrix array that map either a user or group name to a management role.

For both user and group authorization, the contained name can be fully qualified (such as `D:Corp\Jones` and `D:Corp\Sales`) or unqualified (such as Jones and Sales).

The rights that are called out for any of these entries matching the user's identity are granted to that user.

Authorization entries with an unqualified group name are only considered if there are no group entries with a fully qualified name matching the user.

The `symauth` man page provides more information on this topic.

Solutions Enabler log files and settings

- Solutions Enabler writes a new log file every day. Solutions Enabler log files are maintained at:

  `<SYMAPI_HOME>/log/symapi_yyyymmdd.log`

  Where *yyyymmdd* is the year, month, and day.

- Each Solutions Enabler background daemon writes to a pair of log files. Solutions Enabler daemon log files are maintained at:

  `<SYMAPI_HOME>/log/storXXXX.log0`
  `<SYMAPI_HOME>/log/storXXXX.log1`

  Where *storXXXX* is the name of the daemon (for example, `storapid`).

- The secure audit log records operations on Symmetrix arrays. The audit log is maintained on the storage array itself.

  Information from the audit log is retrieved using the `symaudit` SYMCLI command.

  The *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* includes more information about audit logs.

Table 3 summarizes information about Solutions Enabler log files.

Table 3 Solutions Enabler, daemon, and audit log file types and locations

Log type and location	Description
SYMAPI Log files `<SYMAPI_HOME>/log/symapi_yyyymmdd.log`	Where *yyyymmdd* is the numerical value for the year, month, and day. For example, `symapi_20120920.log` is the SYMAPI log for September 20, 2012. Solutions Enabler writes errors and other significant conditions to this log. By default, Solutions Enabler keeps these files forever. Use the `SYMAPI_LOGFILE_RETENTION` option, as described in "Log file retention" on page 23, to configure when to automatically remove these files.

Daemon log files
`<SYMAPI_HOME>/log/storXXXX.log0`
`<SYMAPI_HOME>/log/storXXXX.log1`
	Where *storXXXX* is the name of the daemon. For example: `storapid.log0`, `storapid.log1`, `storgnsd.log0`, `storgnsd.log1`.
	Each Solutions Enabler daemon maintains a pair of log files. The daemons alternate between these two files, switching from one to the other, when the default maximum size of approximately 1 MB is reached.

Symmetrix audit Log (maintained on the Symmetrix array)	A secure audit log containing a record of configuration changes, security alarms, service operations, and security-relevant actions maintained on each Symmetrix array. Records are written to this by Solutions Enabler, software running on the Service Processor, and the Enginuity™ Operating Environment. Information from this log can be retrieved using the `symaudit` SYMCLI command. The *EMC Solutions Enabler Symmetrix Array Management CLI Product Guide* provides more information about the audit log. You can configure the Solutions Enabler event daemon, `storevntd`, to automatically stream audit entries as they appear from this log to an external log service (EMC RSA® enVision®, syslog, Simple Network Management Protocol (SNMP), or the Windows Event Service). The *EMC Solutions Enabler Installation Guide* provides more information on configuring the Solutions Enabler event daemon.

Log file retention

The following option setting controls how long the Solutions Enabler log files are retained.

Table 4 Solutions Enabler log retention

Option name and location	Description
SYMAPI_LOGFILE_RETENTION = *NN* `<SYMAPI_HOME>/config/options`	Solutions Enabler log files can be automatically removed `NN` days after they were created. **Note:** The log files might not be removed after the *NN* days are reached. This value indicates to the system when a given file can be removed by the logging logic during its normal operation. Valid values for *NN* are between 5 and 1825 (or between 5 days and 5 years). If running on the Symmetrix Service Processor, you can only set this to the default value of 0 (keep them forever) or 30. The default options file (`README.options`) installed with Solutions Enabler has more information about this option.

Controlling daemon logging

All Solutions Enabler daemons use a consistent infrastructure for logging activities, which you can customize using the general logging options in the `daemon_options` file (Table 5). In addition, the `daemon_options` file also includes daemon-specific options that allow you to further customize logging for a particular daemon (for example, `storevntd` and `storsrvd`).

By default, each daemon records its log data in a pair of files (*daemon_name*`.log0` and *daemon_name*`.log1`) in the Solutions Enabler logging directory. The daemons alternate logging from one file to the other as the files become full.

Optionally, you can configure each daemon to record its logs to a dated log file in the form *daemon_name-yyyymmdd*`.log`. Each daemon begins recording to a newly dated log file on the first write after 12 a.m.

Table 5 shows the general logging configuration options you can use to customize the Solutions Enabler daemon log files. For details on the syntax and values, refer to the `<SYMAPI_HOME>/config/daemon_options` file.

Table 5 Logging configuration options in the daemon_options file

Option	Description
logfile_type	Controls file switching strategy. Possible values are WRAP or DATED.3
logfile_size	Used for wrapping log files, this option specifies the maximum number of KBs to write before a switch to the other file of the pair.
logfile_retention	Used for dated log files, this option indicates how many days to retain old log files.
logfile_perms	Specifies the permissions on any newly created log files.

Assigning a custom audit log activity ID

By default, Solutions Enabler generates a random activity ID to identify each session. This ID appears in the audit log entries for that session. Since the default ID is random, filtering audit log entries based on the ID is difficult.

Starting in Solutions Enabler version 7.6, you can specify a custom activity ID, making it easier to filter audit log entries.

An optional argument `-actid <Activity ID>`, allows you to set a custom activity ID on all operations performed as part of that CLI command.

When you use the `-actid <Activity ID>` argument, entries in the audit log for that command are tagged with the specified activity ID prefixed with "U_".

User-defined activity IDs must meet the following requirements:

- Maximum of 14 characters (not including the automatic prefix)
- Include only alphanumeric characters

  Underscore (_), and hyphen (-) characters are allowed.

If you try to define an Activity ID with more than 14 characters, the operation fails, and an error is displayed.

All active SYMCLI commands, with the exception of `symaudit` command, accept the `-actid` argument. For example:

```
symqos -sid 237 -cp -name cptest -devs 410:411,1170 addall -actid
   CPTest1
```

Output of `symaudit` queries in verbose mode report the user-defined activity ID. For example:

```
symaudit list -sid 237 -v -activity_id U_CPTest1

              A U D I T   L O G   D A T A

Symmetrix ID            : 000190300237

  Record Number         :   711815
    Records in Seq      :        2
.
.
.
    Username            : H:dldv0181\root
    Activity ID         : U_CPTest1

  Record Number         :   711816
    Records in Seq      :        2
.
.
.
```

Communication (port) security settings

This section describes:

- Ports used by the Solutions Enabler server, the event daemon, CLARiiON, and network connections.
- Options to modify the port settings

Server port usage

In client/server mode, the Solutions Enabler server (the `storsrvd` daemon) listens by default at TCP/IP port 2707 for client connections.

You can configure a port by adding an entry to `<SYMAPI_HOME>/config daemon_options` file as follows:

```
storsrvd:port = nnnn
```

At client hosts, the configuration file `<SYMAPI_HOME>/config/netcnfg` can be changed to reflect the use of this non-default port.

The port used by the `storsrvd` daemon cannot be changed while the server is running. If you want to use a different port, you must specify it in the `daemon_options` file, and restart the `storsrvd` daemon.

Event daemon port usage

When using the asynchronous events mechanism in client/server mode, the event daemon at the client host listens at a TCP/IP port for events being forwarded from the event daemon at the server. By default, the client event daemon asks the operating system to pick an unused port for it to use.

You can configure a specific port for use by adding an entry to the `<SYMAPI_HOME>/config /daemon_options` file on the client host as follows:

```
storevntd:event_listen_port = nnnn
```

CLARiiON port usage

While performing CLARiiON® management operations, Solutions Enabler running on a host sometimes needs to open a TCP/IP connection to either port 443 or 2163 on the CLARiiON array. A configuration setting on the array determines which of these ports is the correct one. Solutions Enabler tries both to make the proper determination.

Network port usage

If a firewall or network address translator is present, these ports must be open. Typically, this is:

- A firewall between the Solutions Enabler client and the server hosts.
- A firewall between the management server and the CLARiiON array.

Table 6 lists the network ports used by Solutions Enabler.

Table 6 Network ports used by Solutions Enabler

Component	Protocol	Port	Description
Solutions Enabler	TCP/IP	2707	In client/server mode, the Solutions Enabler server daemon, `storsrvd`, listens on this port for connections from client hosts. Refer to "storsrvd:port = NNNN" on page 27 You can change the default port as described in "Server port usage" on page 25.
Event daemon	TCP/IP	Dynamically assigned	In client/server mode, the event daemon, `storevntd`, on a client host listens on this port for asynchronous events sent to it from a server host. By default, this is picked at random by the client host event daemon. Refer to "SvcName - TCPIP HostName - NNNN SECURE" on page 27 For information on setting a specific port value, refer to "Event daemon port usage" on page 26.
CLARiiON	TCP/IP	443 or 2163	A configuration file on CLARiiON storage arrays controls whether it listens for connections from management hosts over port 443 or 2163. When Solutions Enabler needs to communicate with an array, it attempts both values. Refer to "CLARiiON port usage" on page 26.

Options to modify port settings

Table 7 lists the options to modify the settings of port s used by Solutions Enabler.

Table 7 Options to modify port settings used by Solutions Enabler

Option location and name	Description
storsrvd:port = *NNNN* `<SYMAPI_HOME>/config/daemon_options`	On the server hosts, this directs the Solutions Enabler server, the `storsrvd` daemon, to listen for connections at this port instead of the default 2707. If the default value is changed for the server, you must edit this entry for this server in the `<SYMAPI_HOME>/config/netcnfg` file as explained in the next row.
SvcName - TCPIP *HostName* - *NNNN* SECURE `<SYMAPI_HOME>/config/netcnfg`	On client hosts, the `netcnfg` file maps service names (*SvcName*), used with the SYMCLI, to a host (*HostName*) and port (*NNNN*) on which the appropriate server is listening. If a non-default server port is configured, it also makes corresponding changes to the servers in this file. For more information, refer to "Configuring the Solutions Enabler client host" on page 31.
storevntd:event_listen_port = *NN NN* `<SYMAPI_HOME>/config/daemon_options`	In client/server mode, the event daemon, `storevntd`, on a client host listens on this port for asynchronous events sent to it from a server host. By default, this is picked at random by the client-host event daemon. On client hosts, this setting directs the event daemon to listen at this specific port for events sent from the event daemon on the server host, instead of using a random port assigned by the local operating system. This setting is automatically transmitted to the event daemon on the server host as needed.

Solutions Enabler client/server security settings

In Solutions Enabler client/server mode, client host operations are automatically forwarded to the `storsrvd` daemon on a server host for execution.

By default, traffic transmitted between client and server hosts is encrypted using SSL.

This section describes the mechanisms to operate these connections in a secure manner.

Network encryption

On platforms where Solutions Enabler supports SSL, servers default to securing all connections using SSL.

Solutions Enabler uses OpenSSL version 1.0.1c with FIPS-140-2 validated, certificate #1747 (FIPS) enabled. FIPS is enabled on the following platforms:

- Linux x86 32 and 64 bit platforms
- Windows x86 32 and 64 bit platforms

By default, clients and servers negotiate a session using TLS 1.2 regardless whether FIPS is enabled.

If a client is running an earlier version of Solutions Enabler, SSLv3 with AES-128 + SHA1 is used.

Securing remote connections using SSL

The default/initial configuration for SYMAPI client and server communications, (on platforms that support it) is to negotiate only SECURE sessions. You can modify the security level at which the client and server are operating.

Before you begin

Before modifying the security level, you should:

- Understand that the security level specifies the capability of the local side and the local side's expectation of the remote side.
- Know whether the host is SSL-capable or SSL-incapable.

The possible security levels are:

- **Level 3 (SECURE)** — (Default) Indicates that only secure sessions will be negotiated between the client and server. This is the highest level of security, and it should only be used when there is no chance of an SSL-incapable client attempting to connect with the server, or an SSL-capable client connecting to an SSL-incapable server.
- **Level 2 (ANY)** — Indicates that either secure or non-secure sessions will be negotiated between the client and server on SSL-capable platforms.
- **Level 1 (NONSECURE)** — Indicates that only non-secure sessions will be negotiated between the client and server. This level is intended as a last resort in situations where SSL cannot be used for some reason or is undesirable. In addition, this level can also be useful in matters of performance and availability.

Note: The default security level is SECURE on platforms that support secure communications and NONSECURE on platforms that do not support secure communications.

The following messages may be issued by the server if SSL-related problems occur:

- ANR0141E through ANR0145E,
- ANR0147E,
- ANR0148E.
- ANR0150E through ANR0153E.
- ANR0155E

The *EMC Solutions Enabler Installation Guide* provides details about SYMAPI server daemon messages.

Session negotiation behavior

Table 8 details the type of session negotiated if a client and server are at the same or different security levels (implied or configured).

Table 8 **session negotiation behavior**

Client security level	Server security level	Negotiated session type
SECURE	SECURE	SECURE
	ANY	SECURE
	NONSECURE	Rejected
NONSECURE	NONSECURE	NONSECURE
	ANY	NONSECURE
	SECURE	Rejected
ANY	ANY	SECURE
	SECURE	SECURE
	NONSECURE	NONSECURE

Configuring the Solutions Enabler server host

Two options specify whether the Solutions Enabler server accepts only secure sessions from clients and whether FIPS 140-2 cryptographic algorithms are used for client/server communication:

- **SYMAPI_SECURITY_LEVEL ‹SECURE |ANY |NONSECURE ›**

  This option specifies whether the Solutions Enabler server accepts only secure sessions from clients.

  The default value for SYMAPI_SECURITY_LEVEL is SECURE. That is, the Solutions Enabler server accepts only secure sessions from clients.

- **SYMAPI_FIPS ‹ ENABLE|DISABLE›**

  When SYMAPI_SECURITY_LEVEL is set to SECURE, this option enables/disables FIPS 140-2 compliant encryption of Solutions Enabler server/client sessions (on Linux and Windows platforms).

  The default value for the SYMAPI_SECURITY_LEVEL option is DISABLE.

On Linux and Windows platforms; when the SYMAPI_SECURITY_LEVEL option is set to SECURE, enabling the SYMAPI_FIPS option enables FIPS 140-2 compliant encryption for client/server communications.

On other platforms, when SYMAPI_SECURITY_LEVEL option is set to SECURE, the existing Solutions Enabler 7.5 encryption algorithms are used for client/server communication.

Backward compatibility to Solutions Enabler 7.5 and earlier

Solutions Enabler 7.6 provides backward compatibility to 7.5 and earlier versions using the following logic to select the security level:

- Look for SYMAPI_SECURITY_LEVEL in the options file.

- If SYMAPI_SECURITY_LEVEL is specified in the options file, use it.
- If the SYMAPI_SECURITY_LEVEL security level is not specified in the options file, the server looks for storsrvd:security_level in the daemon_options file.
  - If the storsrvd:security_level is *not* specified on the server, look for SYMAPI_SERVER_SECURITY_LEVEL.
    - If the SYMAPI_SERVER_SECURITY_LEVEL is *not* specified, use the default for the platform: SECURE everywhere except OVMS, BS2K, or AS400 which use NONSECURE.
    - If the SYMAPI_SERVER_SECURITY_LEVEL *is* specified, use the specified value and post a message saying it was used instead of the storsrvd:security_level.
  - If the storsrvd:security_level is specified, use it.

Allow non-secure client sessions

NOTICE

EMC does not recommend this activity.

You can allow non-secure sessions from clients that cannot or are not configured to negotiate secure sessions by modifying the SYMAPI_SECURITY_LEVEL in the ‹*SYMAPI_HOME*›/config/daemon_options file.

To permit non-secure sessions from clients, do one of the following:

- To accept both secure and non-secure sessions, change the setting to ANY:

  ```
  SYMAPI_SECURITY_LEVEL = ANY
  ```

- To accept only non-secure sessions, change the setting to NONSECURE:

  ```
  SYMAPI_SECURITY_LEVEL = NONSECURE
  ```

Note: Regardless of the value of the SYMAPI_SECURITY_LEVEL option, the cryptographic algorithms used for non client/server communications on Linux and Windows are FIPS 140-2 compliant.

security_clt_secure_lvl

By default, a server verifies a client's subject certificate, if it is able to provide one. You can modify this configuration as follows:

- To configure the server to accept only connections for which the client can be verified, add the following line to the `<SYMAPI_HOME>/config/daemon_options` file at that host:

  ```
  storsrvd:security_clt_secure_lvl = MUSTVERIFY
  ```

- To configure a server to bypass the validation of a client certificate, set the `security_clt_secure_lvl` statement to `NOVERIFY` in the `<SYMAPI_HOME>/config/daemon_options` file:

  ```
  storsrvd:security_clt_secure_lvl = NOVERIFY
  ```

- To enable a server to validate client certificates, set the `security_clt_secure_lvl` statement to `VERIFY` (the default value) in the `<SYMAPI_HOME>/config/daemon_options` file:

  ```
  storsrvd:security_clt_secure_lvl = VERIFY
  ```

Table 10 on page 33 provides a summary of the values of the security_clt_secure_lvl statement.

Configuring the Solutions Enabler client host

NOTICE

The following section includes a description of the steps to disable secure communications. EMC does not recommend this activity.

By default, the Solutions Enabler client attempts to negotiate a secure session with the server when both are capable of doing so.

To allow non-secure sessions with a server that cannot negotiate a secure session, do one of the following:

- To allow a mix of secure and non-secure sessions to servers, depending on the capability of the server, change the following setting to `ANY` in the `<SYMAPI_HOME>/config/options` file:

  ```
  SYMAPI_SECURITY_LEVEL = ANY
  ```

- To require a client to negotiate non-secure sessions to all servers, change the following setting to `NONSECURE` in the `<SYMAPI_HOME>/config/options` file:

  ```
  SYMAPI_SECURITY_LEVEL = NONSECURE
  ```

To configure the session security for specific server hosts, specify `NONSECURE` or `ANY` in the `<SYMAPI_HOME>/config/netcnfg` file for the server in question. This file maps service names to server hostnames (or IP addresses) and port numbers for Solutions Enabler SYMCLI commands.

The format of records in the `netcnfg` file is as follows:

`ServiceName` `- TCPIP` *`HostName IP-Address Port SecurityLevel`*

Where:

- *`ServiceName`* is the name by which the server is known. This is the same value used by the `SYMCLI_CONNECT` environment variable for SYMCLI commands.
- *`HostName`* is the name of the host on which the server resides. Specify either *`HostName`* or *`IP-Address`*.
- *`IP-Address`* is the IP address of the server. Specify either *`HostName`* or *`IP-Address`*.
- *`Port`* is the port number (default 2707) on which the server is listening.
- *`SecurityLevel`* is one of three security levels:
  - `SECURE` negotiates secure sessions:
  - `NONSECURE` negotiates non-secure sessions

- `ANY` negotiates both secure and non-secure sessions, depending on the server configuration.

If you do not specify a security level, `SECURE` is used for secure-capable platforms, and `NONSECURE` is used for secure-incapable platforms, depending on the configuration of the server.

IMPORTANT

The security level specified in the `<SYMAPI_HOME>/config/netcnfg` file takes precedence over the one in the `options` file.

The *EMC Solutions Enabler Symmetrix CLI Command Reference* provides more information on this topic.

Secure session configuration summary

Table 9 lists the host operating systems that support SSL.

Table 9 Host operating systems that support SSL

Operating systems that support SSL
AIX (32- and 64-bit)
HP-UX (32- and 64-bit) HP-UX Itanium (64-bit)
Linux (32-bit) Linux Itanium (64-bit) Linux AMD (64-bit) Linux/390 (32-bit)
Solaris (32- and 64-bit)
Windows (32-bit) Window Itanium (64-bit) Windows AMD (64-bit)
z/OS

Note: Solutions Enabler does not support SSL on iSeries, BS2000, or OpenVMS.

Table 10 provides a summary of the secure session settings.

Table 10 Secure sessions summary

Option name, possible values, and location	Description
storsrvd:security_clt_secure_lvl =MUSTVERIFY |VERIFY |NOVERIFY `<SYMAPI_HOME>/config/daemon_options`	On server hosts, controls how the server validates client certificates. **Note:** This option is not supported on z/OS hosts, where it defaults to NOVERIFY. **MUSTVERIFY**: The server requires clients to send a valid certificate. **VERIFY** (default): The server verifies a client's certificate, if one is sent. **NOVERIFY**: The server does not verify client certificates. "security_clt_secure_lvl" on page 30 provides more information.
storsrvd:security_level =SECURE | NONSECURE |ANY `<SYMAPI_HOME>/config/deamon_options`	On server hosts, controls whether servers establish a secure session. **SECURE** (default): Secure sessions are always used. All other connection types are refused. **NONSECURE**: Non-secure sessions are used; secure sessions are not used. **ANY**: A secure session is established when supported by the client; otherwise a non-secure session is used.
SYMAPI_SECURITY_LEVEL = SECURE | ANY | NONSECURE `<SYMAPI_HOME>/config/options`	Specifies whether the Solutions Enabler server accepts only secure sessions from clients. Applies to both server and client. **SECURE** (default): Secure sessions are always used. All other connection types are refused. **NONSECURE**: Non-secure sessions are used; secure sessions are not used. **ANY**: A secure session is established when supported by the client; otherwise a non-secure session is used. "Configuring the Solutions Enabler server host" on page 29 provides more information.

Certificate files

Solutions Enables uses OpenSSL to generate certificates for secure client-server communication. The client and server verify each other's identity based on the information contained in the certificates.

During installation, the user has the option to install the certificate component. If the user chooses to install the certificate component, a default set of certificates is generated. These certificates are signed by a self-signed root certificate.

Solutions Enabler installs a root certificate and key used to generate subject certificates that identify client and server hosts. The installation process automatically generates a subject certificate for the host on which the install is executed.

The generated certificates can be replaced with certificates that you generate or that are issued to you by a commercial certificate authority (CA). See "Replacing SYMAPI-generated certificates" on page 38.

Subject certificates are generated for both client and server hosts. The subject certificates represent the identity of the host without respect to whether the host acts as a client or a server. Thus, a single set of certificates can be used in both the client and server.

If desired, the client and server can be configured separately to use other sets of certificates. By default, both the client and the server validate the certificate of the peer during secure session negotiation. The client always validates the server's certificate, and you cannot disable this validation when a secure session is negotiated.

The following certificate files enable a client to verify a server's identity and a server to verify a client's identity:

- `symapisrv_cert.pem` is the subject certificate file. It is created specifically for its particular host during installation. It is signed by the EMC SPEA Root certificate. This file must be in the `cert` directory on the SYMAPI client and server for client/server security to work.
- `symapisrv_trust.pem` is the EMC SPEA Root certificate used to sign the SYMAPI certificate file. This file must be in the `cert` directory on every client and server.
- `symapisrv_key.pem` is the key file associated with the subject certificate. It is created specifically for its particular host during installation. It is generated during the certificate creation process. This file must be in the `cert` directory on the SYMAPI client and server for client/server security to work.

The `cert` directory is located at:

- Windows: `<SYMAPI_HOME>\config\cert`
- UNIX and z/OS: `<SYMAPI_HOME>/config/cert`

Note: By default, the location of cert directory is the same for z/OS as UNIX. The location for z/OS systems can be changed during installation.

About wildcards in DNS names

There are several use-cases where the host may have multiple network identities:

- The host has multiple NICs (and thus multiple IP addresses).
- The host is known in multiple domains.
- The host is part of a cluster where a failover causes a different hostname to be in effect at the time the host is issuing the handshake request.

In Solutions Enabler 7.5 and earlier, the Subject identifier of a certificate contains the host information. Specifically, the Common Name (CN) part of the certificate contains the host identity. The CN had to match the Fully Qualified Domain Name (FQDN) of the host being authenticated. Solutions Enabler 7.5 and earlier allowed wildcards (*) in the CN to facilitate this matching.

In Solutions Enabler 7.6, the **-san** argument of the `manage_server_cert create` command allows multiple individual domain names and IP addresses to be specified in a single certificate, eliminating the need for wildcards in the CN.

Starting with version Solutions Enabler 7.6, wildcards are used is as follows:

- Wildcards are allowed in the CN and in arguments for the **–san** option.

- To support backward compatibility, a new `storsrvd` option in the daemon_options file; `security_cert_allow_wildcards` allows you to control whether or not wildcards are supported when authenticating certificates from pre-7.6 clients.
- New keywords extend the `manage_server_cert create` command to accommodate the changes.

manager_server_cert create

IMPORTANT

The format for wildcards varies depending on the platform.
On Unix platforms, **-san** arguments MUST be in quotes.
On Windows, **–san** are NOT in quotes.

On Unix:

```
manage_server_cert.sh create "host*" -san "* *.lss.emc.com"
```

On Windows:

```
manage_server_cert.bat create host* -san * *.lss.emc.com
```

The syntax for the `manage_server_cert create` command is:

```
manage_server_cert create
     common_name
     -san <san_list>
     -o <organization_name>
     -ou <organizational_unit>
```

- **common name** (mandatory) - Can be at most 55 characters long and can contain wildcards. If the common name (CN) is not specified, the script guesses the host name on which the script is being run and uses that in the CN field of the Subject property of the certificate.
- **-san** (optional) - The Subject Alternative Name (**-san**) option allows you to specify multiple hosts, including wildcards.

  If you do not use the **-san** option (the default), you can enter only one hostname as the CN.

  There is no limit on the number of hosts you can specify using the **-san** option.

  Enter a space-separated list of a of DNS names and IP addresses. The IP addresses can be IPv4 and IPv6 addresses.
- **-o** (optional) - Specifies the organizationName attribute in the certificate. The default value is “EMC Corporation”.
- **-ou** (optional) - Specifies the organizationalUnit attribute in the certificate. The default value is "Storage Platform Enablers and Applications”.

Table 11 lists examples of the manage_server_cert create script in Solutions Enabler Version 7.5 and earlier and how to use the **-san** option in Solutions Enabler Version 7.6.

Table 11 manage_server_cert create: **Solutions Enabler 7.5 and 7.6**

Solutions Enabler 7.5 and earlier	Solutions Enabler 7.6
Specify a single hostname: manage_server_cert.sh create aaa.bbb.com	No change.
Specify a hostname with a wildcard: manage_server_cert.sh create aaa.bbb.com ccc.bbb.com *.emc.com	Use the **-san** option to allow wildcards: `manage_server_cert.sh create host_name-san aaa.bbb.com ccc.bbb.com *.emc.com`
Specify an IP address with a wildcard: manage_server_cert.sh create 10.244.245.153 10.245.*	Wildcards are not allowed for IP addresses. Use the **-san** option and specify the individual IP addresses: `manage_server_cert.sh create 10.244.245.153 -san 10.245.150.110 10.245.150.111 10.245.150.112`
Use a wildcard for the entire hostname: manage_server_cert.sh create *	Use the **-san** option to allow the wildcard: `manage_server_cert.sh create -san "*"`
Specify a hostname with the wildcard in the anything other than the leftmost label: manage_server_cert.sh create aaa.bbb.com aaa.*.com	Wildcards are allowed only in the leftmost label of the FQDN. Use the **-san** option and specify the individual labels and hosts: `manage_server_cert.sh create aaa.bbb.com -san aaa.ccc.com aaa.ddd.com aaa.eee.com`
Specify a DNS name using a wildcard: manage_server_cert.sh create *.com	Wildcards are not allowed in without the **-san** option, and must comply with "Rules for wildcards in DNS names". Use the **-san** option and specify the individual DNS names: `manage_server_cert.sh create -san aaa.abc.com bbb.abc.com ccc.abc.com`

Rules for wildcards in DNS names

The following are rules for wildcards in DNS names with **-san** extensions.

Note: The following wildcard rules do not apply to wildcards in the CN.

- Wildcards are allowed only in the leftmost label in a DNS name. DNS names in which the wildcard is other than the leftmost label are ignored. For example:

  dldv0180.*.emc.com - Ignored. Wildcard is not the leftmost label.

  *.lss.emc.com - Is matched.

- If the wildcard is the only character in the leftmost label of the DNS name, only the leftmost label of the host FQDN will be matched. For example:

  host aaa.bbb.emc.com matches host *.bbb.emc.com

  host aaa.bbb.emc.com does not match *.emc.com

- If a DNS name includes a label with a wildcard plus other characters, the wildcard is matched based on regular rules. For example:

  dldv0180.lss.emc.com matches dldv*.lss.emc.com

  dldv0180.lss.emc.com also matches *01*.lss.emc.com.

Managing backward compatibility of certificates

Certificates generated by Solutions Enabler 7.6 are backward compatible with certificates generated by Solutions enabler 7.5. For example:

- A Solutions Enabler 7.6 server can verify a certificate generated by an older (Solutions Enabler 7.5 or earlier) client.
- An older client (Solutions Enabler 7.5 or earlier), can verify a certificate generated by a Solutions Enabler 7.6 server if the certificate's CN contains:
  - A Fully Qualified Domain Name (FQDN) - if the server host name can be resolved to a FQDN, or
  - An IP address corresponding to the server - if the server host name cannot be resolved to a FQDN.

  In cluster configurations, if the Solutions Enabler 7.6 server's certificate does not contain wildcards in the CN, the Solutions Enabler 7.5 client will not verify the server if the server fails over and presents a different host ID than that present in the CN.

  If a Solutions Enabler 7.6 server is running in a clustered environment, Solutions Enabler 7.5 and older clients must have certificates for each host node of the server cluster.

⚠CAUTION

When generating certificates on Solutions Enabler 7.6 servers, be careful not to add non-DNS host names in the CN if Solutions Enabler 7.5 and older clients will connect to the server.

Enabling/disabling support for wildcards in the CN

`storsrvd` allows you to control whether wildcards are accepted in the Common Name (CN) field in client certificates. The syntax is:

```
storsrvd:security_cert_allow_wildcards [enable|disable]
```

- The `enable` setting allows a Solutions Enabler 7.6 server that receives an older certificate (Solutions Enabler 7.5 or earlier) from a client to authenticate the client based on wildcards in the CN.
- The `disable` setting prevents a Solutions Enabler 7.6 server that receives an older certificate (Solutions Enabler 7.5 or earlier) from a client from authenticating the client based on wildcards in the CN. This is the default setting.

Use the `stordaemon setvar` command to enable or disable the `security_cert_allow_wildcards` option. For example:

```
stordaemon setvar storsrvd -name security_cert_allow_wildcards=enable
  storsrvd                    Variable was set
```

Use the `stordaemon getvar` command to display the setting:

```
stordaemon getvar storsrvd -name security_cert_allow_wildcards
  storsrvd                    security_cert_allow_wildcards =enable
```

Managing certificate files

The following sections describe how to use the `manage_server_cert.sh` and `manage_server_cert.bat` scripts to manage certificate files, including:

- "Replacing SYMAPI-generated certificates" on page 38
- "Remove unneeded files (optional)" on page 39
- "Changing a host's name" on page 39
- "Working with a host in multiple domains" on page 40
- "Working with a Windows cluster" on page 40
- "Adding virtual server names to a certificate" on page 41
- "Working with a multi-homed host" on page 42
- "Listing the hostnames in the certificate" on page 42

Note: The *EMC Solutions Enabler Installation Guide* provides more information about installing certificates.

Replacing SYMAPI-generated certificates

You can replace a SYMAPI-generated certificate with one generated by your certificate authority or an external certificate authority (for example, Verisign).

When creating a replacement certificate, you must adhere to the following rules in order for the certificate to work with the client/server:

- Restrictions about wildcards as described in Table 11 on page 36.
- The certificate must be created in Privacy Enhanced Mail (.pem) format.

Once you have created the certificate, do the following on the host:

1. Move the certificate file and the associated key file to the `cert` directory.
2. Move the trusted certificate files that signed the created certificate to the same `cert` directory.
3. Run the `manage_server_cert update` command in the `cert` directory. The format for this command differs for each operating system:
   - In UNIX and Linux environments, run the following:

     ```
     /usr/storapi/bin/manage_server_cert.sh update
     ```

   - In Windows environments, run the following:

     ```
     C:\Program Files\EMC\SYMAPI\config\cert> "\
        Program Files\EMC\SYMCLI\bin\
        manage_server_cert.bat" update
     ```

4. Edit the following options in the `config` directory:
   - If this is a client certificate, edit the following in the `options` file.
     - Change the SYMAPI_SECURITY_ALT_CERT_FILE option to specify the name of the new certificate file.

  - Change the SYMAPI_SECURITY_ALT_KEY_FILE option to specify the name of the new key file.
- If this is a server certificate, edit the following in the `daemon_options` file.
  - Change the `storsrvd:security_alt_cert_file` option to specify the name of the new certificate file.
  - Change the `storsrvd:security_alt_key_file` option to specify the name of the new key file.

Remove unneeded files (optional)

To allow you to recreate the SYMAPI-generated certificate on a host, certain files are left on the host. If there is no need to recreate the SYMAPI-generated certificate, EMC recommends that you remove these files.

Note: Once you remove these files, you will need to perform a full Solutions Enabler install should you need to recreate the SYMAPI-generated certificate files.

To remove these files, run the `manage_server_cert secure` command in the `cert` directory. The format for this command differs for each operating system:

- In UNIX and Linux environments, run the following:

```
/usr/storapi/bin/manage_server_cert.sh secure
```

- In Windows environments, run the following:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\
   Program Files\EMC\SYMCLI\bin\
   manage_server_cert.bat" secure
```

NOTICE

EMC recommends that you only run this command on systems where you will no longer generate certificates.

Changing a host's name

If you change the name of the host on which the client or server is running, you must also change its name in its certificate. To do this, you must run the `manage_server_cert.sh create` command in the host's `cert` directory. This command reads the hostname from the environment and recreates the certificate. The format for this command differs for each operating system:

- In UNIX and Linux environments, run the following:

```
/usr/storapi/bin/manage_server_cert.sh create
```

- In Windows environments, run the following:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
Files\EMC\SYMCLI\bin\manage_server_cert.bat" create
```

The following example illustrates the `manage_server_cert.sh create` command:

```
/usr/storapi/bin/manage_server_cert.sh create
/usr/storapi/bin/manage_server_cert.sh using /usr/storapi/bin/storssl64 to create
   keys
```

```
Generating a 1024 bit RSA private key
.............++++++
.....................................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd
   host001/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=support@emc.com/O=EMC
   Corporation/OU=Storage Platform Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
   /var/symapi/config/cert.
```

Working with a host in multiple domains

When client/server is running on a host that is in multiple domains (for example, `host001.emc.com` and `host001.example.com`), a client or server may not be able to verify the certificate being sent by the other host. This occurs because only one fully qualified name is written to the certificate during installation.

For small numbers of short names (such as those above), you can explicitly spell them out, without using the wildcard.

If the names are long, or if there are many of them, use the **-san** option to add multiple names to a certificate.

To create a certificate, run the following command from the `cert` directory:

- In UNIX and Linux environments, run the following:

  ```
  /usr/storapi/bin/manage_server_cert.sh create host_name -san host
  host host_wildcard ...
  ```

- In Windows environments, run the following:

  ```
  C:\Program Files\EMC\SYMAPI\config\cert> "\
     Program Files\EMC\SYMCLI\bin\
     manage_server_cert.bat" create -san host host host_wildcard ...
  ```

For example, running the following command in a UNIX environment:

```
/usr/storapi/bin/manage_server_cert.bat create lab1 -san
10.244.245.153 *.lss.emc.com lqam*.lss.emc.com
```

produces output similar to the following:

```
Signature ok
subject=/CN=storsrvd
   lab1/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=support@emc.com/O=EMC
   Corporation/OU=Storage Platform Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
   /var/symapi/config/cert.
```

Working with a Windows cluster

This section describes procedures to perform when working with Windows clusters.

Creating a certificate that includes both the hostname and cluster name

During client/server operations, if either the client or server is in a Windows cluster, the hostname sent to the other system may be a cluster name and not a hostname. When the host tries to verify the certificate, it will not find the cluster name since it is not included during installation on the server.

To correct this problem, you can create a certificate with the hostname and the cluster name in it by running the following command from the `cert` directory:

```
full_path_to_bin_directory\manage_server_cert.bat create -san
fully_qualified_host_name fully_qualified_cluster_name
```

Where *`full_path_to_bin_directory`* is the location of the `bin` directory.

For example, running the following command:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\
Program Files\EMC\SYMCLI\bin\
manage_server_cert.bat" create host001.emc.com -san
host001cluster.emc.com
```

produces output similar to the following:

```
Generating a 1024 bit RSA private key
.........++++++
.....................................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd
   host001.emc.com/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=support@emc.com/
   O=EMC Corporation/OU=Storage Platform Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
   /var/symapi/config/cert.
```

Adding virtual server names to a certificate

If virtual server names are defined in the Windows cluster, you should also add these names to the certificate by running the following command from the `cert` directory:

```
C:\Program Files\EMC\SYMCLI\bin\\manage_server_cert.bat create
clustername -san servername1 servername2 servername3 servername4
servername5 servername6 servername7 ...
```

Where *`full_path_to_bin_directory`* is the location of the `bin` directory.

Note: You can specify any number of arguments in the manage_server_cert.bat script.

For example, running the following command from the `cert` directory:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
Files\EMC\SYMCLI\bin\
manage_server_cert.bat" create cluster -san svr1 svr2 svr3 svr4 svr5
svr6 svr7
```

produces the following output:

```
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....................................................++++++
.....................++++++
writing new private key to 'symapisrv_key.pem'
-----
Loading 'screen' into random state - done
Signature ok
subject=/CN=storsrvd cluster/ST=Massachusetts/
C=US/L=Hopkinton/emailAddress=support@emc.com/O=EMC Corporation/OU=Storage Platf
orm Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
```

```
C:\Program Files\EMC\SYMAPI\config\cert.
```

You can also use the following command to perform the same operation:

```
C:\Program Files\EMC\SYMAPI\config\cert> "\Program
Files\EMC\SYMCLI\bin\manage_server_cert.bat" create cluster -san
svr*
```

Working with a multi-homed host

When the client/server is running on a multi-homed host (that is, a host with multiple hostnames), a client or server may not be able to verify the certificate being sent by the other host. This occurs because only one fully qualified name, which is obtained from the hostname command, is written to the certificate during installation.

To correct this problem, you can create a certificate for the multi-homed host containing all of its hostnames by running the following command from the `cert` directory:

- In UNIX and Linux environments, run the following:

  ```
  /usr/storapi/bin/manage_server_cert.sh create -san
  multi-homed-name1 multi-homed-name2 [...]
  ```

- In Windows environments, run the following:

  ```
  C:\Program Files\EMC\SYMCLI\config\cert> "\Program
  Files\EMC\SYMCLI\bin\manage_server_cert.bat
  create -san multi-homed-name1 multi-homed-name2 [...]
  ```

For example, running the following command on host 'host000' in a UNIX environment:

```
/usr/storapi/bin/manage_server_cert.sh create
-san host001.emc.com host002.emc.com host003.emc.com
```

produces output similar to the following:

```
Generating a 1024 bit RSA private key
.....++++++
.........................++++++
writing new private key to 'symapisrv_key.pem'
-----
Signature ok
subject=/CN=storsrvd
   host000.lss.emc.com/ST=Massachusetts/C=US/L=Hopkinton/emailAddress=support@emc.
   com/O=EMC Corporation/OU=Storage Platform Enablers and Applications
Getting CA Private Key
The files symapisrv_cert.pem and symapisrv_key.pem were created in the directory
   /var/symapi/config/cert.
```

Listing the hostnames in the certificate

The command to list the hostnames contained in the certificate varies depending on the operating system:

- In UNIX and Linux environments, run the following:

  ```
  /usr/storapi/bin/manage_server_cert.sh list
  ```

- In Windows environments, run the following:

  ```
  C:\Program Files\EMC\SYMAPI\config\cert> "\
     Program Files\EMC\SYMCLI\bin\
     manage_server_cert.bat" list
  ```

For example, running the following command in a Windows environment:

```
/usr/storapi/bin/manage_server_cert.bat list
```

produces output similar to the following:

```
The host names in this machine's certificate:
lab1
10.244.245.153
*.lss.emc.com
lqam*.lss.emc.com
```

Managing the Solutions Enabler server

This section describes activities to manage the Solutions Enabler server, including:

- "Starting the Solutions Enabler server"
- "Restricting access to the Solutions Enabler server " on page 43
- "Restricting the number of concurrent connections" on page 45
- "Restricting functionality in the Solutions Enabler server" on page 46
- "IBM z/OS-specific behavior for control operations" on page 47

Starting the Solutions Enabler server

The `storsrvd` daemon does not run by default. You must explicitly start it before it can accept connections from remote clients. Run the following command to start the `storsrvd` daemon:

```
stordaemon start storsrvd
```

You can configure the `storsrvd` daemon to start automatically whenever a server host starts by running the following command:

```
stordaemon install storsrvd -autostart
```

Daemons are started differently on z/OS and Open VMS platforms. The *EMC Solutions Enabler Installation Guide* provides detailed instructions on starting the Solutions Enabler server.

Restricting access to the Solutions Enabler server

Use the `<SYMAPI_HOME>/config/nethost` file on a server host to restrict the hosts and users from which the `storsrvd` daemon accepts connections. If the `nethost` file is not present, connections are accepted from all client hosts.

Each line of the `nethost` file identifies authorized hosts with an optional comma-separated list of user names. If a host's user list is omitted or is specified as an asterisk (*), all users connecting from that host are accepted.

When the `nethost` file exists, only the nodes/users listed in this file are allowed to connect to the server to execute remote SYMAPI functions. The trusted host file uses the following format:

```
node                    user-1 [, …, user-n]
address                 user-1 [, …, user-n]
*                       user-1 [, …, user-n]
node                    *
address                 *
*                       *
```

Note: `*` denotes a wildcard for any host or any user, and that no spaces are permitted between user names (such as `joe,sally`). For example:

```
# From Client host host001, all users may connect.
host001        *
# From Client host host002, only users joe and sally may connect.
host002       joe,sally
# An IP address can be used instead of a hostname.
180.100.90.75 *
```

Connections from hosts or users not in the `nethost` file are refused. When a connection is refused, an error message containing the requesting client's user and hostname is written to the `storsrvd` log file on the server.

The `nethost` file is reprocessed every time a new session starts. Any updates made to the file while the server is running will be applied to all future sessions.

Note: The server considers the contents of this file before deciding whether it will negotiate a secure session with the client. If the client host and user are not defined in the `nethost` file, a secure session will not be negotiated. "Securing remote connections using SSL" on page 28 provides more information on the SSL security settings.

Considerations for specifying node and address

The identity of a client host may vary from the perspective of the server, since the server can accept connections from IPv4 and IPv6 clients.

Thus, the network address of the client could be an IPv4 or IPv6 address.

If you specify the network address in the `nethost` file instead of the node name, then the exact syntax of the address is important. If you incorrectly specify an address, connections from some clients may be denied.

EMC recommends that you specify the node name (or the FQDN) since proper DNS configuration usually ensures that the name of the client host is consistent, regardless of the network address.

If you must specify the address, keep these factors in mind:

- The rules for specifying an IPv4 address are unchanged and are simple: Specify the complete address in its dotted-decimal form, without leading zeros in each octet. For example:

  ```
  172.23.191.20          user1
  10.243.142.82          user1
  ```

- If you want to specify an IPv6 address, follow these shorthand rules (part of the IPv6 standard):
  - Leading zeros in each quartet can be omitted.
  - Contiguous sets of zeros can be replaced by two adjacent colons, but only once in an address. If there are multiple non-adjacent sets of contiguous sets of zeros, only one set of double colons can be used. The other set of zeros must be specified. For example:

    ```
    3FFE:80C0:22C:18:250:88FF:FEAD:F92F
    ```

If you are uncertain about the address syntax, ask your network administrator to determine the exact syntax. For most UNIX and Linux hosts, the `ifconfig -a` command can be used to display the IPv6 address of a machine. In a Microsoft Windows environment, use the `ipconfig /all` command to display the IPv6 address.

- If you have IPv4 client hosts that connect to IPv6-capable servers on AIX or Linux, the client network address appears as IPv4-mapped addresses. The server host file validation logic takes this into account and treats IPv4-mapped addresses as though they are native IPv4 addresses. Thus you can specify the regular IPv4 address as described in the first point above.
- You may have to experiment to find the right address.

Restricting the number of concurrent connections

The maximum number of concurrent connections from client hosts is controlled by the `max_sessions` daemon option. The default and maximum value is 100. When a new session arrives that exceeds the threshold, it is refused.

Concurrent sessions may be limited based on the source hostname or username of the client:

- **Limiting by source host** is based on the IP address of the host where the client session originates. User name is not considered when counting concurrent connections from hosts.
- **Limiting by source user** is based on the user identity format described in the "User identification" on page 21. Only two types of user identity formats are counted:
  - The H: format identifies that the client user has been authenticated by the local operating system. This format is used when the client comes from any UNIX or Linux type of host, or from a Windows host where the user has logged into the local system (not a Windows domain). In the host authentication case, the user is considered the same only when logging in from the same host with the same user name.
  - The D: format is used when the client user has logged into a Windows domain. In this case, a user can log into the same domain from different host computers. Such a user identity is considered the same, without respect to the source host that initiates the session.

Two configuration statements for `storsrvd` control session refusal from specific sources:

- `storsrvd:max_sessions_per_host=value` – This option specifies the maximum number of concurrent sessions from any specific host. If a new session from the source host exceeds the threshold for that host, the session is refused.
- `storsrvd:max_sessions_per_user=value` – This option specifies the maximum number of concurrent sessions from any specific user. If a new session from the same user exceeds the threshold for that user, the session is refused.

Table 12 on page 46 lists the `storsrvd` session control options and values.

Note: These options and values are only used by the `storsrvd` daemon and apply to SYMAPI remote sessions. There is no impact on the use of the `stordaemon` control CLI or any other Solutions Enabler daemon.

Best practices for setting the `storsrvd` session control options:

- Set `max_sessions_per_host` and `max_sessions_per_user` to a value less than `max_sessions`. Specifically:
  - Set `max_sessions` to the highest number of concurrent sessions you will tolerate without respect to the source host or user of the session.
  - Set `max_sessions_per_host` and `max_sessions_per_user` to lower values, reflecting the maximum number of concurrent sessions from specific sources you will tolerate.
- Both `max_sessions_per_host` and `max_sessions_per_user` can be used concurrently to count sessions.
- It is possible to set either `max_sessions_per_host` and `max_sessions_per_user` to 0, but doing so refuses all new connections. EMC recommends that if you want to refuse all sessions temporarily, set `max_sessions` to 0. When you wish to resume accepting new sessions, change `max_sessions` to a non-zero value.

Table 12 **storsrvd daemon session control options and values**

Option name	Values	Default	Notes
max_sessions	0 – All new sessions are refused. 1 – 100 – Maximum (host and user) sessions allowed.	100	Default of 100 is compatible with previous releases.
max_sessions_per_host	0 – All new sessions are refused. 1 – 100 – Maximum number of sessions allowed from a specific host. NOLIMIT -Disables counting of sessions from a specific host.	NOLIMIT	The NOLIMIT value provides backward compatibility. NOLIMIT is case-insensitive: NOLIMIT = nolimit
max_sessions_per_user	0 – All new sessions are refused. 1 – 100 – Maximum number of sessions allowed from a specific user. NOLIMIT – Disables counting of sessions from a specific user.	NOLIMIT	

Restricting functionality in the Solutions Enabler server

Settings in the `<SYMAPI_HOME>/config/options` file on a server host can be used to restrict the functionality that the `storsrvd` daemon is allowed to perform on behalf of remote client hosts.

Check to make sure all reference to the options file have a path name of `<SYMAPI_HOME>/config/options`. These options can be changed by editing the `options` file while the server is running. The running server uses the new settings for all future sessions.

Since these are not specified in the `<SYMAPI_HOME>/daemon_options` file, they cannot be changed using the `stordaemon setvar` command.

Table 13 lists the options to restrict the `storsrvd` daemon:

Table 13 Options that restrict storsrvd functionality

Option name (in <*SYMAPI_HOME*>/config/options)	Description
SYMAPI_ACC_ADMIN_VIA_SERVER	Enable/disable Symmetrix Access Control changes. Default is ENABLE.
SYMAPI_ACC_DISPLAY_VIA_SERVER	Enable/disable Symmetrix Access Control information displays. Default is ENABLE.*
SYMAPI_ALLOW_SCRIPTS_VIA_SERVER	Enable/disable Symmetrix TimeFinder® pre-action and post-action scripts. Default is DISABLE.
SYMAPI_CTRL_VIA_SERVER	Enable/disable Symmetrix control operations in general. Default is DISABLE.*

*When set to DISABLE, this class of functionality is not available through the server.

IBM z/OS-specific behavior for control operations

By default, a Solutions Enabler server running on any z/OS host allows configuration changes when requested by a remote client. The *EMC Solutions Enabler Installation Guide* provides additional information.

NOTICE

If control operations are left enabled by default, remote Open Systems users (client/server mode) can make changes to the Symmetrix configuration on your mainframe system.

Data security settings

Solutions Enabler maintains important configuration data in a number of files. It is important that you back up and protect these files at all times. If lost, functionality that depends on the data in these files may be impacted.

Table 14 Data configuration files and locations

File location	Description
<*SYMAPI_HOME*>/config/emcpwddb.dat	This file stores connectivity information (including user names and passwords) used to interact with CLARiiON storage arrays and VMware/Hyper-V Virtual Infrastructure Services. It is managed using the `symcfg authorization` SYMCLI command. The file is encrypted to protect its contents and prevent tampering.
<*SYMAPI_HOME*>/db/symapi_db.bin	The Solutions Enabler database file contains Symmetrix array topology information (arrays, devices, directors, and other information) and device group definitions when Global Name Services is not in use.

Securing Solutions Enabler configuration files

Solutions Enabler stores its configuration files in the following directory:

<*SYMAPI_HOME*>/config

Protect the files in the `config` directory by making sure only authorized Solutions Enabler administrators have write access to this directory.

Running commands as a non-privileged user

Following an initial installation of Solutions Enabler, most SYMCLI commands can only be run as a root user on UNIX systems and by an administrator on Windows systems. To allow other users to execute these commands (for example `symcfg discover`), you must grant them write access to the following directories and their contents:

```
<SYMAPI_HOME>/config
<SYMAPI_HOME>/db
```

Similarly, non-root users on UNIX and non-administrators on Windows must be authorized to explicitly (using `stordaemon`) or implicitly (using ordinary commands) make use of the Solutions Enabler daemons. This is done by adding an entry for a specific user in the file `<SYMAPI_HOME>/config/daemon_users`. For example:

```
# Allow user 'jones' to make use of the storapid daemon:
jones       storapid

# A '*' character at the end of a name can be used
# as a simple wildcard. The following allows user 'jones'
# to make use of any of the Solutions Enabler daemons:
jones       stor*
```

The `<SYMAPI_HOME>/config/README.daemon_users` file installed with Solutions Enabler provides additional information on this topic.

Daemon security settings

Solutions Enabler uses a number of helper daemon processes: `storapid`, `storsrmd`, `storsrvd`, `storgnsd`, `storrdfd`, `storevntd`, and `storwatchd`.

Daemon processes on UNIX

On UNIX, daemons run as a root user by default as a result of their executables being marked `setuid-to-root`.

The `storsrvd`, `storgnsd`, `storevntd`, and `storwatchd` daemons can optionally be configured to run as an identity other than a root user. This can be set:

- During Solutions Enabler installation using the `-daemonuid=Name` option, which when used with the `-silent` option, changes ownership of daemons to non-root user, or
- Post-installation using the `stordaemon` command. For information on which daemons are affected by this option, refer to the `stordaemon` man page.

To configure the GNS daemon to run under the *bin* user account:

```
stordaemon setuser storgnsd -user bin
```

To configure all daemons to run under the bin user account:

```
stordaemon setuser all -user bin
```

Authorized users are allowed to control daemons using the `stordaemon` command line utility. For example, to start the SRM daemon:

```
stordaemon start storsrmd
```

Note: On 64-bit systems, the command is: **`stordaemon start storsrmd64`**.

Non-root and non-administrative users must be defined in the `daemon_users` file to obtain authorization for using daemons and other daemon services.

For additional information, refer to:

- The `stordaemon` man page.
- `<SYMAPI_HOME>/config/README.daemon_users` file installed with Solutions Enabler.

Securing host directories on the server host

The Solutions Enabler `storsrvd` daemon can run with `setuid-to-root` privileges for UNIX systems and system account file privileges for Windows systems.

These privileges are typically greater than the privileges granted to users making use of the `storsrvd` daemon process. This can present security vulnerabilities in situations where a user through a CLI or some other application provides a pathname on which one of the `storsrvd` daemon can operate, such as a backup file to be written to or read from.

To prevent these security vulnerabilities for the `storsrvd` daemon running as a root user, you can specify a list of secure directories in which the `storsrvd` daemon can read, write, and execute files. Existing mechanisms protect the Solutions Enabler database and log file locations. Specify a list of secure directories for the `storsrvd` daemon to protect other operations, such as backups and restores.

Specifying a secure directory path

Review the following before specifying a `secure_directory_path` for the `storsrvd` daemon running as a root user:

- The supplied pathname directories must exist when the daemon is started or the `daemon_options` file is reloaded.

  Nonexistent paths are ignored.
  All subdirectories below the specified directories are also treated as being secure.
- A total of 32 secure directory locations can be maintained.
- Once the `storsrvd` daemon has read the `security_directory_path` statement, directories specified cannot be removed without changing the value in the `daemon_options` file and restarting the daemon.
- New directories can be added while the `storsrvd` daemon is running by editing the `daemon_options` file and reloading it using the command `stordaemon action storsrvd -cmd reload`.
- If the `secure_directory_path` option is not present, the behavior is as it was before Solutions Enabler V7.4. That is, no security checks are performed).
- The `secure_directory_path` option does not apply to the following pathnames:

- Pathnames provided in the `options` or `daemon_options` files.

  These files are assumed to be protected by an administrator.

- Pathnames accessed (read or written) by the SYMCLI itself.

  In client/server mode, these occur under the identity of the user and are subject to standard access control checks against the user identity.

- Pathnames accessed by an API on the client host in client/server mode because these occur under the identity of the user and are not a security risk.

Windows platforms

On Windows platforms, the secure directory path is a list of directories separated by a semicolon (;). Use the backward slash (\) when specifying the directory name.

To apply the secure_directory_path to the `storsrvd` daemon:

```
storsrvd:secure_directory_path = c:\Temp\dir1;c:\Users\SE
```

UNIX platforms

On UNIX platforms, the secure directory path is a list of directories separated by a semicolon (;) or a colon (:). Use the forward slash (/) when specifying the directory name.

To apply the secure_directory_path to the `storsrvd` daemon:

```
storsrvd:secure_directory_path = /tmp/dir1;/opt/dir2;/users/se
```

Listing secure directories

To display a list of secure directories in effect for the `storsrvd` daemon:

```
stordaemon getvar storsrvd -name secure_directory_path
```

Authorizing daemon connections

By default, daemons only accept connection requests from users running with root or administrator privileges.

For non-root users to use this feature, create a `daemon_users` file (initially installed as `README.daemon_users`) with a list of allowed usernames. The location of this directory varies according to the operating system.

Using a text editor, a system administrator can add entries to this file using the following formats:

smith storapid	Local user smith is authorized to use the `storapid` daemon.

ENG/smith storapid	Windows local user smith in the ENG domain is authorized to use the `storapid` daemon.
smith storora*	Local user smith is authorized to use any daemon whose name begins with `storora`. For example, the SRM Oracle DB daemons. (The `*` is a wildcard.)
smith stororad freeze,...	Local user smith is authorized to perform freeze and thaw operations using the `stororad` daemon. The third column consists of a comma separated list of operations that the user is authorized to perform. Valid values are: • freeze: The user is authorized to perform DB freeze and thaw operations. • startup_instance: The user is authorized to start a DB instance. • shutdown_instance: The user is authorized to shutdown a DB instance.

Note: Privileged users are automatically authorized, and do not need to be added to this file.

Note: For more information, refer to the `daemon_users` file.

Securing SRM operations

This section describes activities pertaining to configuring security for EMC's Storage Resource Management (SRM).

Authorizing SRM daemon connections

Access to SRM functionality is controlled by limiting permission to the SRM daemon. This access is controlled using the common daemon authorization file, `daemon_users`. This file is located in the following directories:

UNIX	`/var/symapi/config/daemon_users`
Windows	`c:\Program Files\EMC\SYMAPI\config\daemon_users`

Note: It is important to protect this file so that only privileged administrators can modify it.

Users meeting any of the following criteria are permitted to control and use the SRM daemon:

- Authorized users: UNIX users with root access, and Windows users that are a members of the Administrators group
- Users listed in the `daemon_users` file located on each host from which they require access

For example, the following lines in the `daemon_users` file would permit users named `smith` and `jones` to use the SRM daemon:

```
smith     storsrmd
jones     storsrmd
```

Various SRM control operations for authorized local users can be defined by adding the appropriate control operation keyword to the `daemon_users` file command line in the third column, as shown in the following example:

```
smith     storsrmd     file_allocate, file_extend
jones     storsrmd     file_allocate, file_extend
```

Note: Multiple control operations can be added to the line, separated by a comma.

Table 15 lists the available SRM control operations by keyword that can be added for a user. The keyword must be added to the `daemon_users` file to authorize the user to perform the specified SYMCLI control operations.

Table 15 SRM control operations

Keyword	Control operation	SYMCLI command
`file_allocate`	Allocate a file	Not available
`file_extend`	Extend a file	Not available
`file_trim`	Trim a file	Not available
`fs_flush`	Flush a file system	Not available
`fs_freeze`	Freeze a file system	Not available
`fs_thaw`	Thaw a file system	Not available
`lv_create`	Create a logical volume	`symlv create`
`lv_delete`	Delete a logical volume	`symlv delete`
`lv_resize`	Resize a logical volume	`symlv extend` `symlv reduce`
`lv_mirror_add`	Add a logical volume mirror	`symlv add`
`lv_mirror_remove`	Remove a logical volume mirror	`symlv remove`
`vg_create`	Create a volume group	`symvg create`
`vg_delete`	Delete a volume group	`symvg destroy`
`vg_device_add`	Add a device to a volume group	`symvg adddev`
`vg_device_remove`	Remove a device from a volume group	`symvg rmdev`
`vg_recover`	Recover a volume group	`symvg recover`

Note: For any directories and files being accessed for SRM control and mapping operations, operating-system-level permission is required. "Setting operating-system-level permissions for SRM" provides more information on this topic.

Setting operating-system-level permissions for SRM

The operating-system-level permissions for any directories and files being accessed for the control and mapping operation must be set for the user. The following examples show various administrator settings for directories and files to allow read (r), write (w), and execute (x) privileges:

Permits root user access:

```
-rwx--------  root  other  /usr/vxfs/root.bin
```

Permits user smith and root access:

```
-rwx-------  smith  symapi  /usr/vxfs/smithroot.bin
```

Permits all users access:

```
-rwxrwxrwx  root  other  /usr/vxfs/allusers.bin
```

Connecting to the SRM database

A connection must be established between the SRM database and the database commands to allow access to the database. To connect to the database, set `SYMCLI_RDB_CONNECT` to your username and password. When working with one type of database, use `SYMCLI_RDB_TYPE` to save keystrokes with a set of commands.

Note: For a non-root user on Solaris, the Sybase database daemon must be manually started from the root user before calling any Sybase database functions.

If the connection is remote, a network service name must be supplied with the password.

For SYMCLI mapping access to an Oracle database, the user must have one of the following:

- Select any table privilege
- DBA role
- SYSDBA role

To invoke control access of an Oracle database, the user must have administrator privileges.

Database start-up options

The database control commands start up and shut down of the specified database server manager instance.

Database manager instance start up or shut down can be done on the client side. This removes the limitation of most databases that only allows start-up and shutdown of their database manager instance on the database server side.

Use the `symrdb startup -type DbType` CLI command to start up the specified database, including: Sybase, Oracle, SQL Server, and DB2/UDB databases.

Note: Database start-up and shutdown options are not available for Informix databases.

Sybase

The following options are available for Sybase databases:

```
symrdb startup -type Sybase -f runserver_file [-m] [-t delay_time]
```

where:

runserver_file — Specifies the absolute pathname of a runserver file used as a reference each time a Sybase server is restarted.

-m — Starts the database in single user mode.

delay_time — Specifies the estimated time, in seconds, to start up the Sybase server.

SQLServer

The following start-up options are available for SQLServer databases:

```
symrdb startup -type SqlServer -s instance [-c] [-f] [-m] [-n] [-x]
                    [-p master_file_path] [-e error_log_path]
                    [-l master_log_path] [-g virtual_addr_space]
                    [-t trace_number]
```

where:

instance — Instance name to be started.

master_file_path — Master database file.

error_log_path — Error log file.

master_log_path — Master database log file.

virtual_addr_space — Amount of virtual address space, in megabytes.

trace_number — Trace number.

DB2/UDB

The following start-up options are available for DB2/UDB databases:

```
symrdb startup -type IBMUDB [-c computer]
[-n node [ADDNODE -u hostname -p port
[-nt netname] [NODE | CATALOG -tsn tablespace_node]] |
[RESTART [-u hostname] [-p port] [-nt netname]] |
[STANDALONE]]
```

where:

ADDNODE — Issues the ADD NODE command.

RESTART — Issues the RESTART DATABASE command.

STANDALONE — Start the node in STANDALONE mode.

NODE — Specifies that the containers for the temporary tablespaces should be the same as those for the specified node.

CATALOG — Specifies that the containers for the temporary tablespaces should be the same as those for the catalog node of each database.

profile — Specifies the name of the profile.

node — Specifies the node number.

hostname — Specifies the system name.

port — Specifies the port number.

netname — Specifies the net name.

tablespace_node — Specifies the node number from which the temporary tablespace definitions should be obtained.

computer — Specifies the computer name.

Oracle

The following start-up options are available for Oracle databases:

```
symrdb startup -type Oracle [[FORCE][RESTRICT][PFILE=FileName][QUIET]
                     [MOUNT [-db DbName] |
                     [ [OPEN | OPEN_READ_ONLY | OPEN_READ_WRITE |
                     OPEN_READ_WRITE_RECOVER | RECOVER | OPEN_RECOVER]
                     [-db DbName] ] | NOMOUNT]] |
                     [ [PFILE=FileName] MIGRATE [QUIET]]
```

where:

FileName — Specifies the file to be used while starting up the instance.

DbName — Specifies the database name to mount or open. Refer to the Oracle document for the definition.

FORCE — Shuts down the current Oracle instance (if it is running) with the shutdown option ABORT before restarting it.

RESTRICT — Allows only Oracle users with the RESTRICTED SESSION system privilege to connect to the database.

QUIET — Suppresses the display of the System Global Area information for the starting instance.

MOUNT — Mounts the database but does not open it.

OPEN — Mounts and opens the database.

NOMOUNT — Causes the database not to be mounted upon instance start-up.

MIGRATE — Starts the database in OPEN MIGRATE mode and sets system initialization parameters to specific values required to enable the database upgrade or downgrade scripts to run.

OPEN_RECOVER — Opens the database and performs media recovery, if necessary, before starting the instance.

OPEN_READ_ONLY — Specifies READ ONLY to restrict users to read-only transaction, preventing them from generating redo logs.

OPEN_READ_WRITE — Specifies READ WRITE to open the database in read/write mode, allowing users to generate redo logs. This is the default.

OPEN_READ_WRITE_RECOVER — Specifies READ WRITE to open the database in read/write mode and specifies that media recovery should be performed, if necessary, before starting the instance.

Maintenance: back up Solutions Enabler state

Back up the following directories and their contents to preserve the Solutions Enabler configuration on a host:

```
<SYMAPI_HOME>/config
<SYMAPI_HOME>/db
```

The other directories under <*SYMAPI_HOME*> contain less critical data that is recreated by Solutions Enabler as needed.

CHAPTER 4
Unisphere for VMAX security configuration

This chapter describes security settings for Unisphere for VMAX 1.6, including:

- Unisphere access control 58
- User-based access control..... 58
- Unisphere port usage..... 63
- Replacing the self-signed certificate file 63
- Other security settings 65

Unisphere access control

Unisphere supports the following types of user authentication:

- **Windows** (local and domain-based): Users have a Windows account on the SMAS server.

  Users login with Windows domain\username and password.

- **LDAP**: User accounts are stored on an LDAP-SSL server.

  Users login with their LDAP-SSL username and password (they must have a user account on a LDAP-SSL server).

  To use this method, a Unisphere Administrator or SecurityAdmin must set up LDAP-SSL authentication in Unisphere. *Configuring authentication* in the *Unisphere for VMAX Product Guide* and online help includes instructions for completing this task.

  ⚠WARNING

  Unisphere is subject to LDAP-SSL man-in-the-middle attack. Unisphere does NOT validate the LDAP server's SSL certificate. EMC strongly recommends using VPN solutions if the customer's LDAP servers are outside the DMZ, and in the cases where the customer's Unisphere deployment is vulnerable to man-in-the-middle attack (due to malicious insiders).

- **Local Unisphere users**: Users have local Unisphere accounts.

  Users login with their Unisphere username and password.

  Local user accounts are stored locally on the SMAS server host and work in much the same way as the other methods to validate user credentials.

  To use this method, a Unisphere Initial Setup User, Administrator, or SecurityAdmin must create a local Unisphere user account.

  Unisphere for VMAX product Guide and online help includes instructions for completing this task.

Trust between Unisphere and the Symmetrix system is established using the sequence described in "Unisphere port usage" on page 63.

User-based access control

Unisphere uses "roles" and groups to restrict which management operations a user can perform on a Symmetrix system.

The steps to create and manage Unisphere user accounts, including user authorization are described in *EMC Unisphere for VMAX Product Guide* and *EMC Unisphere for VMAX* online help.

Default user account

An Initial Setup User is created during installation.

The Initial Setup User is a temporary role to install and set up Unisphere. Once an Administrator or SecurityAdmin is assigned to a Symmetrix system, the Initial Setup User can no longer access or view the system from the Unisphere console.

The steps to create the Initial Setup User are described in *EMC Unisphere for VMAX Product Guide* and *EMC Unisphere for VMAX* online help.

User roles

Unisphere includes the following user roles:

- **None** - Provides no permissions.
- **Monitor** - Performs read-only (passive) operations on a Symmetrix system excluding the ability to read the audit log or access control definitions.
- **StorageAdmin** - Performs all management (active or control) operations on a Symmetrix system in addition to all Monitor operations. This role does not allow users to perform security operations.
- **Administrator** - Performs all operations on a Symmetrix system, including security operations in addition to all StorageAdmin and Monitor operations.
- **SecurityAdmin** - Performs security operations on a Symmetrix system in addition to all Monitor operations.
- **Auditor** - Grants the ability to view, but not modify, security settings for a Symmetrix system (including reading the audit log, symacl list, and symauth) in addition to all Monitor operations. This is the minimum role required to view the Symmetrix audit log.
- **Perf Monitor** - Performs the same operations as a monitor, with the addition of being able to set performance alerts and thresholds.
- **Initial Setup User** - Defined during installation, this temporary role provides administrator-like permissions for the purpose of adding local users and roles to Unisphere. For more information on the Initial Setup User, see the online help.

Individual and group roles

Users access a Symmetrix system or component directly through a role assignment and/or indirectly through membership in a user group that has a role assignment.

User groups enable administrators to assign roles to multiple users simultaneously. User groups are created on the SMAS server according to its operating system and assigned roles with Unisphere.

If a user has two different role assignments (one as an individual and one as a member of a group), the permissions assigned to the user will be combined.

For example, if a user is assigned a Monitor role and a StorageAdmin role through a group, the user will be granted Monitor and StorageAdmin rights.

User IDs

The following section describes the SYMAPI format to create users and roles.

Note: This format displays in the footer bar of the Unisphere GUI, but not in the User/Role list view or creation wizard.

Users and user groups are mapped to their respective roles by IDs. These IDs consist of a three-part string in the form:

`Type:Domain\Name`

In the event that a user is matched by more than one mapping, the user authorization mechanism uses the more specific mapping:

- If an exact match (e.g., D:sales\putman) is found, that is used.
- If a partial match (e.g., D:*\putman) is found, that is used.
- If an unqualified match (e.g., putman) is found, that is used.
- Otherwise, the user is assigned a role of None.

Valid values for `Type`, `Domain`, and `Name` are as follows:

`Type` Type of security authority used to authenticate the user or group.
Possible types are:

`L` A user or group authenticated by LDAP. In this case, *`Domain`* specifies the fully qualified name of the domain controller on the LDAP server.
For example:
`L:danube.com\Finance`
indicates that user group Finance will log in through domain controller danube.com.
Once configured, individual LDAP users and groups can log in to Unisphere using a simple username, or simple group name, respectively.
For example, Finance.

`C` A user or group authenticated by the SMAS server.
For example:
`C:Boston\Legal`
indicates that user group Legal will log in through Unisphere sever Boston.

`H` A user or group authenticated by logging into a local account on a Windows host. In this case, `Domain` specifies the hostname.
For example:
`H:jupiter\mason`
indicates that user mason will log in on host jupiter.

`D` A user or group authenticated by a Windows domain. In this case, `Domain` specifies either the simple domain name (for individual users) or the fully qualified domain name (for groups).
For example:
`D:sales\putman`
indicates user putman will log in through Windows domain sales.
Once configured, individual Windows domain users can log in to Unisphere using a simple username.
For example, putman.
Group Windows domain users can log in to Unisphere using either a simple domain name\group name or a fully qualified domain name\group name.

`V` A user or group authenticated by a virtualization domain. In this case, `Domain` specifies the virtualization domain name.

`Domain`

Within role definitions, IDs can be either fully qualified (as above), partially qualified, or unqualified.

When the Domain portion of the ID string is an asterisk (*), the asterisk is treated as a wildcard, meaning any host or domain.

Note: When configuring group access, the `Domain` portion of the ID must be fully qualified.

For example:

`D:ENG\jones`	Fully qualified path with a domain and username (for individual domain users).
`D:ENG.xyz.com\ExampleGroup`	Fully qualified domain name and group name (for domain groups).
`D:*\jones`	Partially qualified that matches username jones with any domain.
`H:HOST\jones`	Fully qualified path with a hostname and username.
`H:*\jones`	Partially qualified that matches username jones within any host.
`jones`	Unqualified username that matches any jones in any domain on any host.

`Name`

Specifies the username relative to that authority. It cannot be longer than 32 characters and spaces are allowed if delimited with quotes.

Usernames can be for individual users or user groups.

About user authorization

User authorization restricts the management operations users can perform on a Symmetrix system.

By default, authorization rules are enabled on a Symmetrix system for Unisphere users, regardless of whether authorization has been enabled on the Symmetrix system.

An Administrator or SecurityAdmin user can map individual users or groups of users to specific user roles which determine the operations the users can perform.

The authorizations on a Symmetrix system determine the privileges the Initial Setup User has on the system. The following section describes the relationship between the Initial Setup User and Symmetrix authorizations:

- If Symmetrix authorization is enabled, authorization rules are always enforced. The Initial Setup User could be locked out if no authorization rule exists for the user.
- If Symmetrix authorization is disabled and there are no authorization rules on the Symmetrix system, the Initial Setup User is granted Admin privileges.
- If Symmetrix authorization is disabled and there are no Admin or Security authorization rules on the Symmetrix system, the Initial Setup User is granted Admin privileges. All other rules are enforced as defined.

- When Symmetrix authorization is disabled and Admin or Security Admin authorization rules are defined on the Symmetrix system, if the Initial Setup User does NOT have an auth rule explicitly defined, the initial setup user will have NO permissions. All other rules are enforced as defined.

Assigning user authorization

Before you can use the Unisphere REST API, you must assign user authorization for each Symmetrix VMAX array the user is permitted to access. Use the following procedure to assign user authorization for arrays:

1. Log into Unisphere.
2. At the bottom of the Common Tasks panel click Administration.
3. Click Users and Roles to open the Users and Roles list view.
4. Create a user login profile for each Symmetrix array to be accessed by the user, and assign them to any of the following roles:
   - Monitor
   - StorageAdmin
   - Administrator
   - SecurityAdmin

Note: These user roles are valid for the currently available REST resource methods (GET, POST, and DELETE). When they become available, only Admin and Storage Admin roles will be able to initiate PUT methods.

Unisphere port usage

Unisphere components use the following ports:

Table 16 Unisphere port usage

Component	Port
Mysql	3324
https	8443
vApp Manager	5480
CLI	9999
Messaging	5445
Messaging-throughput	5455
Jacorb	3528
Remoting	4447

Replacing the self-signed certificate file

During installation, if the installer chooses to install the certificate component, a default set of certificates is generated. These certificates are signed by a self-signed root certificate.

This section describes the steps to replace the self-signed certificate created at installation with a CA-signed certificate.

Before you begin

You will need the keystore password to complete the following procedures. The keystore password is generated during installation. It is stored in the following file:

‹SMAS installation›/jboss/server/default-em/deploy/jbossweb.sar/server.xml

where ‹SMAS installation› is the directory where SMAS is installed.

Open the file and search for "keystorePass".

Note: There may be multiple elements that contain this property. All but element should be commented out. The correct element has the following entry:

keystoreFile="${jboss.server.home.dir}/conf/keystore"

Step 1: Generate a certificate request.

a. Type the following commands where `<SMAS installation>` is the directory where SMAS is installed:

   On **Windows** systems:

```
cd <SMAS installation>\jboss\server\default-em\conf

<SMAS installation>\jre\bin\keytool -certreq -alias tomcat -file
tomcatcert.csr -keystore keystore
```

On **Linux** systems:

```
cd <SMAS installation>/jboss/server/default-em/conf
```

```
<SMAS installation>/jre/bin/keytool -certreq -alias tomcat -file
tomcatcert.csr -keystore keystore
```

A prompt to enter the keystore password appears. This is the password described in "Before you begin".

b. Type the password and press ENTER.

A file named tomcatcert.csr is generated. This file is a certificate request.

The next step varies depending on whether the following error message appears:

```
"Failed to establish chain from reply"
```

– If the error message appears, proceed to Step c.

– If the error message does not appear, skip to Step d.

c. The error appeared because your CA is not in the keystore's default trust chain (the keystore comes with a default set of trusted CAs). You must import a trusted certificate into your keystore to serve as the root for the trust chain.

To import a trusted certificate into your keystore, type the following commands where `<SMAS installation>` is the directory where SMAS is installed and `<Trusted CA cert file>` is the signed certificate you received from your CA:

⚠CAUTION

Be very careful about importing a new trusted certificate.

On **Windows** systems:

```
cd <SMAS installation>\jboss\server\default-em\conf
```

```
<SMAS installation>\jre\bin\keytool -import -alias root -file
<Trusted CA cert file> -keystore keystore -trustcacerts
```

On **Linux** systems:

```
cd <SMAS installation>/jboss/server/default-em/conf
```

```
<SMAS installation>/jre/bin/keytool -import -alias root -file
<Trusted CA cert file> -keystore keystore -trustcacerts
```

Skip to "Step 3: Start the SMAS server".

d. Send the certificate request to your Certificate Authority (CA).

The CA will validate the request and provide you with a signed certificate.

Step 2: Import the CA-signed certificate into the keystore.

a. Stop the SMAS server.

b. Type the following commands where `<SMAS installation>` is the directory where SMAS is installed and `<signed certificate file>` is the signed certificate you received from your CA:

On **Windows** systems:

```
cd <SMAS installation>\jboss\server\default-em\conf

<SMAS installation>\jre\bin\keytool -import -alias tomcat -file
<signed certificate file> -keystore keystore -trustcacerts
```

On **Linux** systems:

```
cd <SMAS installation>/jboss/server/default-em/conf

<SMAS installation>/jre/bin/keytool -import -alias tomcat -file
<<signed certificate file>> -keystore keystore -trustcacerts
```

A prompt to enter the keystore password appears. This is the password described in "Before you begin".

c. Type the password and press ENTER.

Step 3: Start the SMAS server

The steps to start and stop the server are described in *EMC Unisphere for VMAX Installation Guide*.

Other security settings

This section describes other security-related topics applicable to Unisphere for VMAX.

Link-and-launch security

Link-and-launch clients connect to Unisphere only using HTTPS. The client and Unisphere are required to establish mutual trust.

The Link-and-launch client (acting as SSL client) must establish trust either by:

- Explicitly importing Unisphere self-signed certificate into the client's trust store, or
- Implicitly, if the Unisphere self-signed certificate (generated during installation) is replaced with a certificate issued by a mutually-trusted CA (a CA trusted by both Unisphere and the Link-and-Launch client).

Unisphere's trust with the client's launching application is established by explicit registration (by Admin/SecurityAdmin) of the Link-and-Launch client's ID.

The client must provide a valid username of the Unisphere user in whose context the Link-and-Launch is performed. Once trust is established, a one time password (token) is issued to trusted Link and Launch clients. The tokens are then exchanged as a means to single sign on into Unisphere.

Username format is documented in "User IDs" on page 59.

Note: Unisphere supports Link-and-Launch ONLY in the context of the users with Admin, StorageAdmin, or Monitor roles.

When the transport is fully secured (by mutual trust establishment), and the user is validated (during initial registration connection), Unisphere issues the client a one-time password (OTP).

In the following request, the client exchanges the OTP for a launch token. The exchange must take place within OTP's time-to-live of 10 minutes, otherwise the process (of OTP acquisition and OTP-to-token exchange) must be started anew.

The token is valid only for a single launch-and-link until Unisphere server reboots.

Unisphere data security

You can export and import some Unisphere configuration settings. This feature allows you to configure multiple Unisphere installations more easily.

Settings are protected by the sha512 hash with a random salt, concatenated with a customer-defined one-time password.

The password is communicated out-of-band by means the customer deems necessary.

Security alert system

Administrator or StorageAdmin users can configure Unisphere to deliver alert notifications for SNMP, e-mail, and Syslog.

The steps to configure alerts, manage alert thresholds, and view alert-related information are described in *EMC Unisphere for VMAX Product Guide* and *EMC Unisphere for VMAX* online help.

Session timeout

Unisphere sessions timeout after 8 hours of user inactivity.

The timeout interval is not configurable.

When timeouts occur, the user is logged out, but the user account is not locked.

Root access requirements

Unisphere requires Root/Administrator access privileges for installation, deployment, and operations.