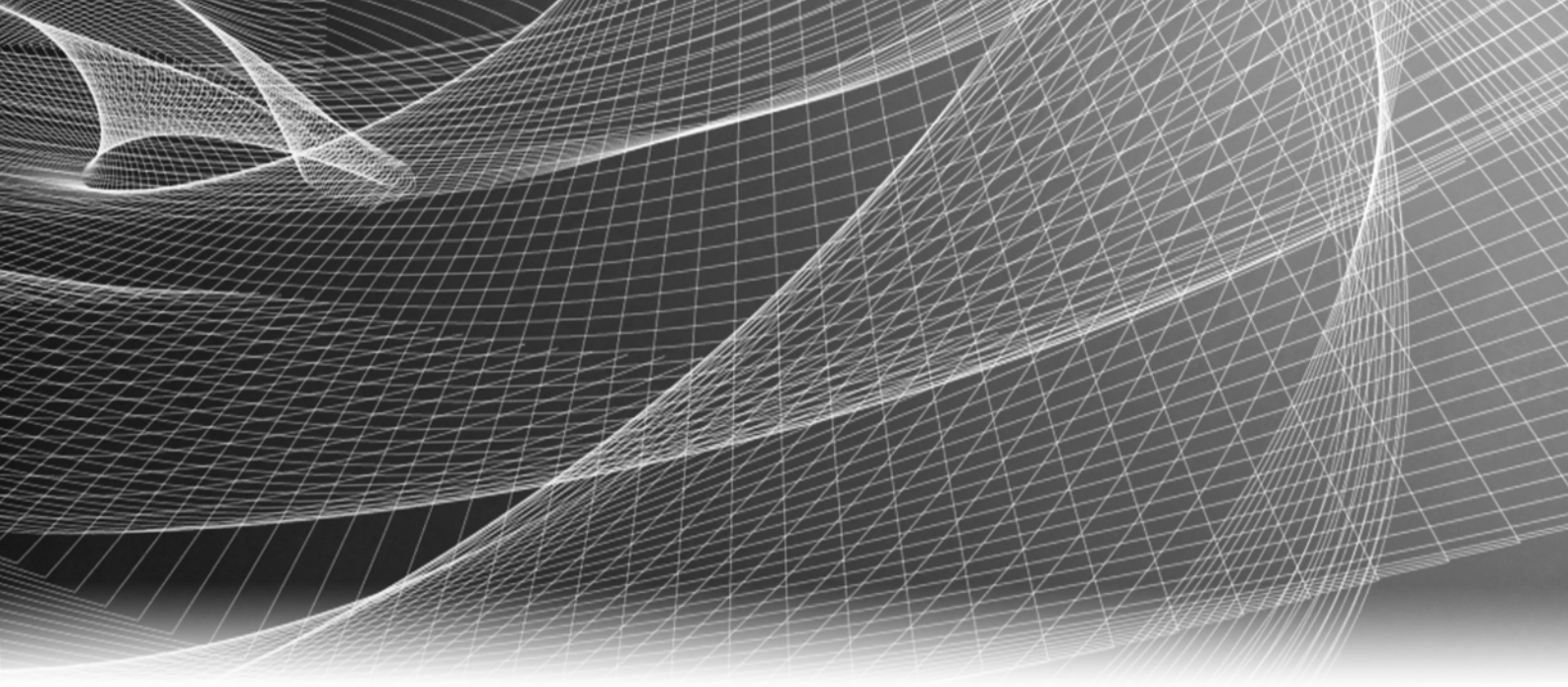


EMC® Avamar® 6.1

Product Security Guide

P/N 300-013-347
REV 08

EMC²

Published June, 2013

CONTENTS

Chapter 4 Data Security and Integrity

Chapter 5 System Monitoring, Auditing, and Logging

Chapter 6 Server Hardening

Appendix A Port Usage and Firewall Requirements

Index

TABLES

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 the EMC online support website. Check the EMC online support website to ensure that you are using the latest version of this document.

Purpose

This publication discusses various aspects of EMC Avamar product security.

Audience

This publication is primarily intended for EMC Field Engineers, contracted representatives, and business partners who are responsible for configuring, troubleshooting, and upgrading Avamar systems at customer sites, as well as system administrators or application integrators who are responsible for installing software, maintaining servers and clients on a network, and ensuring network security.

Revision history

The following table presents the revision history of this document.

Table 1 Revision history

Revision	Date	Description
08	June 17, 2013	Fixed minor typos in the Port Usage appendix.
07	December 13, 2012	Added a new step 5 to the procedure "Generating and installing a server certificate" on page 44.
06	December 7, 2012	Corrected information in "Avamar data port listing" on page 108.
05	October 25, 2012	Updates for release 6.1 Service Pack 1. • In "Avamar data port listing" on page 108, removed port 53 (TCP), added port 69 (TCP), and changed ports 8005 and 8009 to 8505 and 8509. • Revised the section "Password complexity, length, and reuse" on page 103 to reflect changes in Service Pack 1.
04	July 31, 2012	Update "Where to get help" on page 11 in the Preface.

Table 1 Revision history

Revision	Date	Description
A03	June 30, 2012	Revised "Additional firewall configuration to support replication" on page 104
A02	June 15, 2012	Added the following topics: • "Additional firewall configuration to support replication" on page 104 • "Uninstalling level-2 hardening packages" on page 105
A01	April 25, 2012	First release of Avamar 6.1

Related documentation

The following EMC publications provide additional information:

- *EMC Avamar Release Notes*
- *EMC Avamar Administration Guide*
- *EMC Avamar Operational Best Practices*

Conventions used in this document

EMC uses the following conventions for special notices:

⚠DANGER

DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.

⚠WARNING

WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.

⚠CAUTION

CAUTION, used with the safety alert symbol, indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

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

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

IMPORTANT

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

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, SQL 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

The Avamar support page provides access to licensing information, product documentation, advisories, and downloads, as well as how-to and troubleshooting information. This information may enable you to resolve a product issue before you contact EMC Customer Service.

To access the Avamar support page:

1. Go to https://support.EMC.com/products.
2. Type a product name in the **Find a Product** box.
3. Select the product from the list that appears.
4. Click the arrow next to the **Find a Product** box.
5. (Optional) Add the product to the **My Products** list by clicking **Add to my products** in the top right corner of the **Support by Product** page.

Documentation

The Avamar product documentation provides a comprehensive set of feature overview, operational task, and technical reference information. Review the following documents in addition to product administration and user guides:

- Release notes provide an overview of new features and known limitations for a release.
- Technical notes provide technical details about specific product features, including step-by-step tasks, where necessary.
- White papers provide an in-depth technical perspective of a product or products as applied to critical business issues or requirements.

Knowledgebase

The EMC Knowledgebase contains applicable solutions that you can search for either by solution number (for example, esgxxxxxx) or by keyword.

To search the EMC Knowledgebase:

1. Click the **Search** link at the top of the page.
2. Type either the solution number or keywords in the search box.
3. (Optional) Limit the search to specific products by typing a product name in the **Scope by product** box and then selecting the product from the list that appears.
4. Select **Knowledgebase** from the **Scope by resource** list.
5. (Optional) Specify advanced options by clicking **Advanced options** and specifying values in the available fields.
6. Click the search button.

Live chat

To engage EMC Customer Service by using live interactive chat, click Join Live Chat on the Service Center panel of the Avamar support page.

Service Requests

For in-depth help from EMC Customer Service, submit a service request by clicking Create Service Requests on the Service Center panel of the Avamar support page.

Note: To open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

To review an open service request, click the Service Center link on the Service Center panel, and then click View and manage service requests.

Facilitating support

EMC recommends that you enable ConnectEMC and Email Home on all Avamar systems:

- ConnectEMC automatically generates service requests for high priority events.
- Email Home emails configuration, capacity, and general system information to EMC Customer Service.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

BSGDocumentation@emc.com

Please include the following information:

- Product name and version
- Document name, part number, and revision (for example, A01)
- Page numbers
- Other details that will help us address the documentation issue

CHAPTER 1
Introduction

The following topics introduce and describe Avamar security:

Overview

EMC® Avamar® is backup and recovery software with integrated data deduplication technology. This Product Security Guide provides an overview of the settings and security provisions that are available in Avamar to ensure secure operation of the product. Security settings are split into the following categories:

- "User Authentication and Authorization" on page 19 provides an overview of Avamar user accounts and the authentication and authorization mechanisms available for those accounts.
- "Client/Server Access and Authentication" on page 41 describes settings available to limit access by client components.
- "Data Security and Integrity" on page 77 describes settings available to ensure protection of the data that Avamar manages.
- "System Monitoring, Auditing, and Logging" on page 87 provides an overview of the features available to monitor events in the Avamar environment and to audit the operations performed. It also provides a list of log files that are available for each feature on each component in the system.
- "Port Usage and Firewall Requirements" on page 107 lists the ports and protocols that Avamar uses for client-server communication for all applicable firewalls.

Related documents

Specific product configuration instructions reside in the Avamar documentation that is available on the EMC Online Support website at https://support.EMC.com. Where appropriate in this guide, specific documents are referenced, including the *EMC Avamar Administration Guide*.

Security patches

Each Avamar release is available with a set of up-to-date security patches. If you install any other security patches or security applications that are incompatible with Avamar, then you must remove them and restore the Avamar system to its previous working configuration. Then file a support case with EMC Customer Service and include the specific security updates that you applied.

NOTICE

It is the responsibility of the customer to ensure that the Avamar system is configured to protect against unauthorized access. Back up all important files before you apply new security patches, applications, or updates.

Email home notification using ConnectEMC

When configured and enabled, the “email home” feature automatically emails configuration, capacity, and general system information to EMC Technical Support using ConnectEMC. Summary emails are sent once daily; critical alerts are sent in near-real time on an as needed basis.

The *EMC Avamar Administration Guide* provides details on how to enable the email home feature.

Remote access

If EMC Technical Support must connect to a customer system to perform analysis or maintenance, the customer can initiate a web conference using a web-based conferencing application such as WebEx.

Additionally, beginning with version 6.0, customers can install an EMC Secure Remote Support (ESRS) gateway to allow EMC Technical Support to access their systems without WebEx.

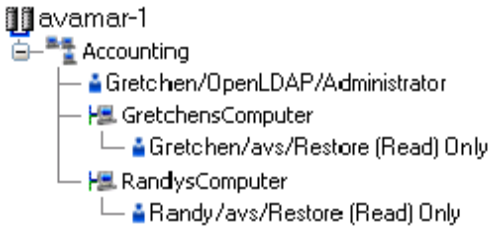
CHAPTER 2
User Authentication and Authorization

The following topics provide an overview of Avamar user accounts and the authentication and authorization mechanisms available for those accounts:

Domain and client users

In the Avamar system, user accounts can be added to domains or individual clients. Domain users administer the domain to which they belong and any subdomains beneath it. Individual client users perform backups and restores of the client to which they belong and access backups in the system that belong to the client.

In Avamar, user accounts are not reusable objects; they are simply entries in a domain or client access list. When you add a new user account to the Avamar system, you actually add a new entry to the domain or client user access list. Consider the following example:



User “Gretchen” has been added to both the Accounting domain and her computer. However, the authentication system (OpenLDAP in the Accounting domain and avs on the computer) and role (Administrator in the Accounting domain and Restore [Read] Only on the computer) are different. These are in fact two completely separate user accounts that happen to have the same user name.

Avamar user accounts comprise the following pieces of information:

- User name
- Authentication system
- Role

User name

The user name for a domain or client user account must be in the format that the selected authentication system accepts. For example, the internal Avamar authentication system uses case-sensitive user names, whereas Windows Active Directory user names are case-insensitive.

Note: User names cannot be longer than 31 characters.

Authentication system

An authentication system is a user name/password system that is used to grant domain and client users access to the Avamar server. Avamar supports its own internal authentication system (“avs”), as well as several external authentication systems, such as Network Information Service (NIS), Open Lightweight Directory Access Protocol (OpenLDAP) and Windows Active Directory. “Enterprise authentication” on page 32 provides details on supported external authentication systems and how to configure the Avamar system to use one of these systems.

Roles

Roles define various allowable operations for each user account. There are three basic categories of roles:

- Administrator roles
- Operator roles
- User roles

Administrator roles

Administrators are generally responsible for maintaining the system.

The role of administrator can only be assigned to user accounts at a domain level; this role cannot be assigned to user accounts at a client level. The role of administrator can be assigned to user accounts at the top-level (root) domain, or any other domain or subdomain.

Root administrators

Administrators at the top-level (root) domain have full control of the system. They are sometimes referred to as “root administrators.”

Domain administrators

Administrators at lower level domains (other than root) generally have access to most features, but typically can only view or operate on objects (backups, policy objects, and so forth) within that domain. Any activity that might allow a domain administrator to view data outside that domain is disallowed. Therefore, access to server features of a global nature (for example, suspending or resuming scheduled operations, changing run times for maintenance activities, and so forth) is disallowed.

Furthermore, domain administrators:

- Cannot add or edit other subdomain administrators
- Cannot change their assigned role
- Can change their password

Operator roles

Operator roles are generally implemented to allow limited access to certain areas of the system to perform backups and restores, or obtain status and run reports. These roles allow greater freedom in assigning backup, restore, and reporting tasks to persons other than administrators.

As with administrator roles, operator roles can only be assigned to user accounts at the domain level; these roles cannot be assigned to user accounts at the client level. Furthermore, to add the user account to subdomains, you must have administrator privileges on the parent domain or above.

There are four operator roles:

- Restore only operator
- Backup only operator
- Backup/restore operator
- Activity operator

Users who have been assigned an operator role do not have access to the entire Avamar Administrator application. Instead, following login, they are presented with a single window that provides easy access to the features that they are allowed to use.

Restore only operator

Restore only operators are generally only allowed to perform restores and to monitor those activities to determine when they complete and if they complete without errors.

As with roles assigned to other domain user accounts, restore only operators at the top-level (root) domain can perform restores for any client in the system; restore only operators at lower level domains (other than root) can only perform restores for clients within that domain.

To enforce these constraints, restore only operators do not have access to the full Avamar Administrator application. Instead, following login, restore only operators are presented with a window that provides easy access to the features that they are allowed to use.

Restore only operators can perform the following tasks within the allowable domain:

- Perform a restore
- Monitor activities

Backup only operator

Backup only operators are generally only allowed to perform backups and to monitor those activities to determine when they complete and if they complete without errors.

As with roles assigned to other domain user accounts, backup only operators at the top-level (root) domain can perform backups for any client or group in the system; backup only operators at lower level domains (other than root) can only perform backups for clients or groups within that domain.

To enforce these constraints, backup only operators do not have access to the full Avamar Administrator application. Instead, following login, backup only operators are presented with a window that provides easy access to the features that they are allowed to use.

Backup only operators can perform the following tasks within the allowable domain:

- Perform on-demand client backups
- Initiate on-demand group backups
- Monitor activities

Backup/restore operator

Backup/restore operators are generally only allowed to perform backups or restores, and to monitor those activities to determine when they complete and if they complete without errors.

As with roles assigned to other domain user accounts, backup/restore operators at the top-level (root) domain can perform backups and restores for any client or group in the system; backup/restore operators at lower level domains (other than root) can only perform backups and restores for clients or groups within that domain.

To enforce these constraints, backup/restore operators do not have access to the full Avamar Administrator application. Instead, following login, backup/restore operators are presented with a window that provides easy access to the features that they are allowed to use.

Backup/restore operators can perform the following tasks within the allowable domain:

- Perform on-demand client backups
- Initiate on-demand group backups
- Monitor activities
- Perform a restore

Activity operator

Activity operators are generally only allowed to monitor backup and restore activities and create certain reports.

Activity operators at the top-level (root) domain can view or create reports for backup and restore activities within the entire system (all domains and subdomains); activity operators at lower level domains (other than root) can only view or create reports for backup and restore activities within that domain.

To enforce these constraints, activity operators do not have access to the full Avamar Administrator application. Instead, following login, activity operators are presented with a window that provides easy access to the features that they are allowed to use.

Activity operators can perform the following tasks within the allowable domain:

- Monitor activities
- View the group status summary
- View the activity report
- View the replication report

User roles

User roles are always assigned to a user account for a specific client. As such, allowable operations are inherently constrained to that specific client.

Note: Users assigned any of the following roles cannot log in to Avamar Administrator.

There are four user roles:

- Backup only user

  Users with this role can initiate backups directly from the client using the **avtar** command line.

- Restore (read) only user

  Users with this role can initiate restores directly from the client using the **avtar** command line or Avamar Web Services.

- Backup/Restore user

  Users with this role can initiate backups and restores directly from the client using the **avtar** command line or Avamar Web Services.

- Restore (read) only/ignore file permissions

  This role is similar to the Restore (Read) Only User role except that operating system file permissions are ignored during restores, thereby effectively allowing this user to restore any file stored for that Avamar client.

  All Windows client user accounts should be assigned this role to ensure trouble-free restores.

  This role is only available when external authentication is used. "Enterprise authentication" on page 32 provides details on external authentication.

Managing domain and client users

You can add a new user to a client or to a domain, edit user information, or delete a user by using the Avamar Administrator Administration Window Account Management tab. The *EMC Avamar Administration Guide* provides details.

Default user accounts

The Avamar system uses the following default user accounts and passwords.

Table 2 Default user accounts

User Account	Default Password	Description/Remarks
Avamar Server Linux OS		
root	changeme	Linux OS root account on all Avamar nodes.
admin	changeme	Linux OS account for Avamar administrative user.
dpn	changeme	Linux OS account for Avamar maintenance user.
Avamar Server Software		
root	8RttoTriz	Avamar server software root user account.
Avamar Administrator		
MCUser	MCUser1	Default Avamar Administrator administrative user account.
backuponly	backuponly1	Account for internal use by the MCS.
restoreonly	restoreonly1	Account for internal use by the MCS.
backuprestore	backuprestore1	Account for internal use by the MCS.
repluser	9RttoTriz	Account for internal use by the MCS for replication.
MCS PostgreSQL Database		
admin		No password, logged in on local node only.
viewuser	viewuser1	Administrator server database view account.
EMS PostgreSQL Database		
admin		No password, logged in on local node only.
Proxy Virtual Machine Linux OS		
root	avam@r	Linux OS root account on all proxies deployed using the Avamar proxy appliance. This account is for internal use only.

Password encryption

Although Avamar passwords are typically entered as plain text, they are stored on each respective host filesystem in encrypted form.

All Avamar clients and utilities automatically detect whether a supplied password is plain text or encrypted. Either plain text or encrypted format will work.

The **avtar --encodepassword** command can be used to process a plain text password and output the correct encrypted string to stdout.

Encrypted passwords are host-specific. A password encrypted and stored on one host cannot be copied and used on another host.

Changing passwords for default user accounts with the change-passwords utility

The **change-passwords** utility enables you to change passwords for the following default user accounts:

- The admin, dpn, and root operating system user accounts
- The root and MCUser Avamar server user accounts

The **change-passwords** utility also enables you to create new admin and dpnid OpenSSH keys.

IMPORTANT

After using this utility to change the password for the MCUser account, use dpnctl to restart the Avamar Desktop/Laptop service, dtlt, as described in the administration guide. If the dtlt service is not restarted, Avamar client users will encounter session expired messages when they log in to the web UI.

To start the **change-passwords** utility:

1. Open a command shell and log in:
   - If logging into a single-node server, log in to the server as dpn.
   - If logging into a multi-node server, log in to the utility node as dpn.
2. Type:

   ```
   change-passwords
   ```

   The utility prompts you to change the operating system and Avamar server user accounts, as well as to create new admin and dpnid OpenSSH keys, if desired.

   You can choose to perform one or all of these tasks on all nodes including optional node types, or on utility node and storage nodes only.

   Keep in mind the following points about the utility:

   - If you are administering a multi-node server, you can choose whether to change the passwords on all nodes or only on selected nodes.
   - To change the password for either the MCUser or root Avamar server user accounts, you must specify the current password for the root account.
   - After changing the MCUser password using **change-passwords**, notify owners of hosts external to the Avamar server to update their Avamar Management Console Command Line Interface (MCCLI) configurations, as discussed in "Manually updating MCCLI passwords" on page 27.
   - If there were custom public keys in the authorized_keys2 files for the admin, dpn, or root operating system user accounts, then you may need to re-add the custom keys. The authorized_keys2 files are detailed in "SSH keys for operating system user accounts" on page 28.
   - Remember to resume all schedules by using Avamar Administrator.

Manually updating MCCLI passwords

The **change-passwords** utility does change the internal Avamar server MCUser password for the Avamar Management Console Command Line Interface (MCCLI), which generates events whenever cron maintenance activities run. However, **change-passwords** will not update any MCCLI configuration files located externally to the utility node. Therefore, any external MCCLI configuration files will need to be manually updated.

Note: Use of **change-passwords** to change the internal Avamar server MCUser password disables the MCCLI.

Edit the following files to manually update the MCUser password:

- ~admin/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml
- ~dpn/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml
- ~root/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml

To edit the mcclimcs.xml files for admin, dpn, and root to use the new MCUser password:

1. Open a command shell and log in:
   - If logging into a single-node server, log in to the server as admin.
   - If logging into a multi-node server, log in to the utility node as admin.
2. Open ~admin/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml in a UNIX text editor such as **vi** or **emacs**.
3. Locate the following entries:

```
<MCSConfig>
   <MCS
      mcsprofile="local"
      mcsaddr="AVAMARSERVER"
      mcsport="7778"
      mcsuserid="MCUser"
      mcspasswd="PASSWORD"
   />
   <!-- add more profiles if needed here and set default to select
default -->
</MCSConfig>
```

Note: This example has been simplified for clarity.

4. Change the mcspasswd=”PASSWORD” entry to the new password that you set with the **change-passwords** utility.
5. Save the changes.
6. Switch user to the dpn user account by typing:

```
su - dpn
```

7. When prompted for a password, type the dpn password and press **Enter**.
8. Load the dpn OpenSSH key by typing:

```
ssh-agent bash
ssh-add ~dpn/.ssh/dpnid
```

9. Open ~dpn/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml in a UNIX text editor.
10. Repeat steps 3–5.
11. Switch back to the admin user account by typing:

```
exit
exit
```

12. Switch user to root by typing:

```
su -
```

13. When prompted for a password, type the root password and press **Enter**.
14. Open ~root/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml in a UNIX text editor.

Note: The ~root/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml file might not be present on all servers. If this is the case, skip steps 14–15.

15. Repeat steps 3–5.
16. Switch back to the admin user account by typing:

```
exit
```

SSH keys for operating system user accounts

Access to the admin, dpn and root operating system user accounts is available through SSH login. SSH uses public and private encrypted keys to authenticate users logging in to those accounts. SSH login access can be obtained by supplying operating system account passwords or using either of two pre-authorized private keys, as described in the following table.

Table 3 SSH keys for operating system user accounts

Private key file name	Matching public key file name	Default passphrase	Authorizes access to	Where keys can be found
admin_key	admin_key.pub	P3t3rPan	Operating system admin account	~admin/.ssh/
dpnid	dpn_key.pub		Operating system admin and root accounts	~admin/.ssh ~dpn/.ssh/

On an Avamar server, use the **change-passwords** utility, discussed in "Changing passwords for default user accounts with the change-passwords utility" on page 26, to coordinate changes to private keys and corresponding authorizations across all nodes.

admin user account

The admin user account SSH v2 key configuration is controlled by the following files and directories in the home directory for admin.

Table 4 admin user account SSH key files

File/directory	Description
~admin/.ssh/	Private SSH directory. This directory must be fully protected and owned as follows: drwx------ 2 admin admin
~admin/.ssh/config	SSH configuration file. This file must contain the following entry: StrictHostKeyChecking=no This file must be fully protected and owned as follows: -r-------- 1 admin admin
~admin/.ssh/admin_key	Private RSA OpenSSH key file. This file must be fully protected and owned as follows: -r-------- 1 admin admin The admin user account SSH private and public keys must be named admin_key and admin_key.pub, respectively.
~admin/.ssh/admin_key.pub	Public RSA OpenSSH key file. This file is public and does not need to be protected. -r--r--r-- 1 admin admin
~admin/.ssh/dpnid	Private DSA OpenSSH key file. This file must be fully protected and owned as follows: -r-------- 1 admin admin
~admin/.ssh/id_rsa	Symbolic link to ~admin/.ssh/admin_key.
~admin/.ssh/authorized_keys2	Contains a list of public keys for users allowed to log in to the admin user account. This file must be fully protected and owned as follows: -r-------- 1 admin admin This file must contain public key entries for the admin and dpn user accounts: • The admin public key entry is an RSA key, prefixed with "ssh-rsa" and appended with the comment "dpn_admin_key." • The dpn public key entry is a DSA key, prefixed with "ssh-dss" and appended with the comment "dpn@dpn41s."

Any files not listed in the previous table can be ignored.

Use of the admin key requires a passphrase. The only method to change or remove a passphrase is to generate a new private/public key pair and modify the appropriate authorized_keys2 files accordingly. To ensure proper operation of the Avamar server, the admin user must authorize SSH access by way of the dpnid private key. This is

accomplished by including the matching public key (dpn_key.pub) in the authorized_keys2 file for the admin user. The dpnid private key must not require a passphrase.

dpn user account

The dpn user account SSH v2 key configuration is controlled by the following files and directories.

Table 5 dpn user account SSH keys

File/directory	Description
~dpn/.ssh/	Private SSH directory. This directory must be fully protected and owned as follows: drwx------ 2 dpn admin - or - drwx------ 2 dpn dpn
~dpn/.ssh/config	SSH configuration file. This file must contain the following entry: StrictHostKeyChecking=no This file must be fully protected and owned as follows: -r-------- 1 dpn admin - or - -r-------- 1 dpn dpn
~dpn/.ssh/dpnid	Private DSA OpenSSH key file. This file must be fully protected and owned as follows: -r-------- 1 dpn admin - or - -r-------- 1 dpn dpn The dpn user account SSH private and public keys must be named dpnid and dpn_key.pub, respectively.
~dpn/.ssh/dpn_key.pub	Public DSA OpenSSH key file. This file is public and does not need to be protected. -r--r--r-- 1 dpn admin - or - -r--r--r-- 1 dpn dpn
~dpn/.ssh/id_rsa	Symbolic link to ~dpn/.ssh/dpnid.
~dpn/.ssh/authorized_keys2	Contains a list of public keys for users allowed to log in to the admin user account. This file must be fully protected and owned as follows: -r-------- 1 dpn admin - or - -r-------- 1 dpn dpn This file is deliberately left empty to ensure that no one can log in as user dpn using SSH keys.

Any other files can be ignored.

The only way to log in as user dpn is to know the operating system dpn password. To ensure proper operation of the Avamar server, the public key for dpn must be in both the .ssh/authorized_keys2 file for both root and admin.

root user account

The root user account SSH v2 key configuration is controlled by the following files and directories.

Table 6 root user account SSH keys

File/directory	Description
.ssh/	Private SSH directory. This directory must be fully protected and owned as follows: drwx------ 2 root root
.ssh/config	SSH configuration file. This file must contain the following entry: StrictHostKeyChecking=no This file must be fully protected and owned as follows: -r-------- 1 root root
.ssh/authorized_keys2	Contains a list of public keys for users allowed to log in to the root user account. This file must be fully protected and owned as follows: -r-------- 1 root root This file must contain a public key entry for the dpn user accounts. As currently shipped, the dpn public key entry is a DSA key, prefixed with “ssh-dss” and appended with the comment “dpn@dpn41s.”

Any files not listed in the previous table can be ignored.

To log in as the root user requires the password for the root account or use of the pre-authorized dpnid private key. To ensure proper operation of the Avamar server, the root user must authorize SSH access by way of the dpnid private key. This is accomplished by including the matching public key (dpn_key.pub) in the authorized_keys2 file for the root user. The dpnid private key must not require a passphrase.

Enterprise authentication

Enterprise (or external) authentication enables users to use the same user ID and password to log in to multiple systems. The Avamar external authentication feature is not a single user ID/password login, fully integrated into an external authentication system on which users are created and managed. Instead, the same user ID must be created on both Avamar and external systems while the password is set and managed externally.

Avamar Login Manager provides access to the external authentication databases through the standard Pluggable Authentication Module (PAM) library of the Linux operating system.

Login Manager runs on the utility node and is installed and started during Avamar server installation and upgrade. It uses the domains configuration file to identify the supported domains.

Supported components and systems

External authentication is only available for specific Avamar components and two external systems.

Avamar components

Avamar Administrator, Avamar Enterprise Manager, and Avamar Web Access support external authentication for user accounts.

External authentication is *not* available for Avamar server-level administration user accounts, including:

- root, admin, and dpn operating system user accounts
- Special Avamar system administrative users like MCUser and root

External systems

Avamar supports the following categories of external authentication systems.

Table 7 Supported external authentication systems

Category	Description
Lightweight Directory Access Protocol (LDAP)	Hierarchical directory structure X.500 standard system such as: • Microsoft Active Directory Service (MS ADS) • Novell NDS and eDirectory
Network Information Service (NIS) SUN Yellow Pages (YP)	Flat workgroup-based database structure of user IDs, passwords, and other system parameters comparable to Microsoft Windows NT such as: • Master NIS Server - Primary Domain Controller (PDC) • Slave NIS Servers - Backup Domain Controllers (BDC)

Configuring enterprise authentication

To configure Avamar external authentication:

1. Back up the current configuration files.
2. Configure the LDAP or NIS interface, as discussed in "Configuring the LDAP interface" on page 33 or "Configuring the NIS interface" on page 36.
3. Use Avamar Administrator to create the users who require login access to Avamar. The *EMC Avamar Administration Guide* provides detailed instructions.

   The username must match exactly the user ID on the LDAP or NIS server. Create external users in the proper LDAP or NIS server domain location (for example, the root "/" or other directory like "/clients/"). When creating users, the external domain appears in the Authentication System list.
4. Confirm the creation of the external users by logging in to Avamar Administrator or Avamar Enterprise Manager as the external user.

   Log in according to the following rules:

   a. User ID followed by `@DOMAIN`.

      Where DOMAIN is the LDAP or NIS server domain that you specified when you edited the /etc/avamar/domains.cfg file while configuring the LDAP or NIS interface.

      For example: SueV@example.com

   b. User password same as entered in the external LDAP or NIS system.

   c. Domain path where external users reside (for example, "/clients/").
5. Back up the configuration files again.

Note: You also should back up the configuration files before you install future software upgrades because the process might overwrite them with default values. Resetting external authentication is fairly simple with backed up configuration files.

Configuring the LDAP interface

1. Collect the following server information and utilities.

Table 8 Information required to configure LDAP (page 1 of 2)

Category	Item
Information about external LDAP system	LDAP domain name
	IP address or fully-qualified domain/hostname of the LDAP authentication server
	Distinguished name (DN) of the user used for LDAP queries
	Password of DN used for LDAP queries
Information about the Avamar server	Linux operating system root user password
	Linux operating system admin user password
	Avamar system admin user name and password

Table 8 Information required to configure LDAP (page 2 of 2)

Category	Item
Utilities for testing and troubleshooting	ldapbrowser
	GetMyDN (Windows utility from Softerra)
	ldapsearch (/usr/bin directory)

2. Open a command shell and log in:
   - If logging into a single-node server, log in to the server as root.
   - If logging into a multi-node server, log in to the utility node as root.
3. Open /etc/avamar/domains.cfg in a UNIX text editor, such as **vi** or **emacs**.
4. Add the following entry in the Customer Specific Domains section, and then save the file:

   ```
   DOMAIN=ID
   ```

   where:
   - DOMAIN (format: example.com) is a unique customer-specific LDAP domain used for addressing PAM.
   - ID is an integer larger than 1. IDs 0 and 1 are reserved for Avamar internal use.

   Note: Step 5 requires the creation of a symbolic link for this entry. Instead of DOMAIN=ID, an existing ldap=3 is available for use (by uncommenting the line). If ldap=3 is used, skip step 5 because the symbolic link already exists.

   The DOMAIN part of the entry (either ldap or a unique LDAP domain) appears in the Avamar Administrator Authentication System list. Entering a unique DOMAIN clarifies which LDAP domain is used for external authentication.
5. Create a unique lm_ldap file and symbolically link to it by typing:

   ```
   ln -sf /etc/pam.d/lm_ldap /etc/pam.d/lm_NUMBER
   ```

   where NUMBER is the LDAP domain ID in step 4.
6. Log in to the server as admin.
7. Load the admin OpenSSH key by typing:

   ```
   ssh-agent bash
   ssh-add ~admin/.ssh/admin_key
   ```
8. When prompted, type the admin user account passphrase and press **Enter**.
9. Confirm that the systemname and lmaddr are set up correctly by typing:

   ```
   avmaint config --avamaronly | grep systemname
   avmaint config --avamaronly | grep lmaddr
   ```

   These commands display the hostname and IP address of the utility node, respectively.

10. As root, create a symbolic link from ldap.conf to ldap.conf.winad by typing:

    ln -sf /etc/ldap.conf.winad /etc/ldap.conf

11. Set correct group ownership and file permissions for ldap.conf by typing:

    chown root:root /etc/ldap.conf
    chmod 0600 /etc/ldap.conf

12. Confirm the symbolic link by typing:

    ls -l /etc/ldap.conf

    The following information appears in the command shell:

    /etc/ldap.conf -> /etc/ldap.conf.winad

13. In a UNIX text editor, open /etc/ldap.conf.
14. Modify the following entries, and then save the file:

    `host HN-IPADD`

    where HN-IPADD is the fully-qualified hostname or IP address of the LDAP server.

    `base dc=DOMAIN, dc=com`

    where DOMAIN is the first part of the LDAP domain name. For example: example.com would be displayed as dc=example, dc=com.

    `binddn cn=PROXYUSER, ou=PROXYUNIT, ou=PROXYORG, dc=DOMAIN, dc=com`

    where PROXYUSER, PROXYUNIT, PROXYORG, and DOMAIN comprise parts of the distinguished name of the user used to bind with the LDAP server. Components include:

    - cn - common name
    - ou - organizational or unit name
    - dc - domain

    For example: Distinguished name avamaruser.users.avamar.emc.com
    Components: cn=avamaruser, ou=users, ou=avamar, dc=emc, dc=com

    `bindpw PWD`

    where PWD is the password of the user used to bind with the LDAP server.

15. Restart Login Manager by typing:

    service lm restart

16. Confirm that configuration changes were accepted by typing:

    avmgr lstd

    All domains used in Avamar authentication are listed.

17. Confirm that the LDAP server can be queried by typing:

    ```
    ldapsearch -x -W -h HOSTNAME -b dc=DISTINGUISHED_NAME -D
     cn=VALID_USERNAME, cn=users,dc=DISTINGUISHED_NAME
    ```

    where:

    - HOSTNAME is the hostname or IP address of the LDAP server.
    - dc=DISTINGUISHED_NAME is the domain part of the distinguished name (the two "dc" components).
    - VALID_USERNAME is a valid user in the LDAP server domain.

    A success message or referral result should appear. A communication or authentication failure is a problem indication.

    For example:

    ```
    ldapsearch -x -W -h 10.0.100.21 -b dc=aelab01,dc=com -D
     cn=administrator,cn=users,dc=aelab01,dc=com
    ```

Note: Space limitations in this guide caused the previous commands to continue (wrap) to more than one line. The command must be entered on a single command line (no line feeds or returns allowed).

Configuring the NIS interface

1. Open a command shell and log in:
    - If logging into a single-node server, log in to the server as root.
    - If logging into a multi-node server, log in to the utility node as root.
2. Open /etc/avamar/domains.cfg in a UNIX text editor.
3. Add the following entry in the Customer Specific Domains section, and then save the file:

    ```
    DOMAIN=ID
    ```

    where:

    - DOMAIN (format: example.com) is a unique customer-specific NIS domain used for addressing PAM
    - ID is an integer larger than 1. IDs 0 and 1 are reserved for Avamar internal use.

    Note: Step 4 requires the creation of a symbolic link for this entry. Instead of DOMAIN=ID, an existing nis=2 is available for use (by uncommenting the line). If nis=2 is used, skip step 4 because the symbolic link already exists.

    The DOMAIN part of the entry (either nis or a unique NIS domain) appears in the Avamar Administrator Authentication System list. Typing a unique DOMAIN clarifies which NIS domain is used for external authentication.

4. Create a unique lm_nis file and symbolically link to it by typing:

   ```
   ln -sf /etc/pam.d/lm_nis /etc/pam.d/lm_NUMBER
   ```

   where NUMBER is the NIS domain ID in step 3.

5. Set correct group ownership and file permissions for the lm_nis file by typing:

   ```
   chown root:root /etc/pam.d/lm_NUMBER
   chmod 0600 /etc/pam.d/lm_NUMBER
   ```

   where NUMBER is the NIS domain ID in step 3.

6. Confirm the symbolic link by typing:

   ```
   ls -l /etc/pam.d/lm_NUMBER
   ```

   where lm_NUMBER is the file created in step 4.

   The following information appears in the command shell:

   ```
   /etc/pam.d/lm_NUMBER -> lm_nis
   ```

7. In a UNIX text editor, open lm_NUMBER (created in step 4).

8. Modify the following entries, and then save the file:

   ```
   auth required /lib/security/pam_nis.so domain=NISDOMAIN
   account required /lib/security/pam_nis.so domain=NISDOMAIN
   ```

   where NISDOMAIN is the NIS domain in step 3.

9. Log in to the server as admin.

10. Load the admin OpenSSH key by typing:

    ```
    ssh-agent bash
    ssh-add ~admin/.ssh/admin_key
    ```

11. When prompted, type the admin user account passphrase and press **Enter**.

12. Confirm the systemname and lmaddr are set up correctly by typing:

    ```
    avmaint config --avamaronly | grep systemname
    avmaint config --avamaronly | grep lmaddr
    ```

    These commands display the hostname and IP address of the utility node, respectively.

13. As root, restart Login Manager by typing:

    ```
    service lm restart
    ```

14. With keys loaded, confirm that configuration changes were accepted by typing:

    ```
    avmgr lstd
    ```

    All domains used in Avamar authentication are listed.

15. Open /etc/sysconfig/network in a UNIX text editor.

16. Add the following entry, and then save the file:

    ```
    NISDOMAIN=DOMAINNAME
    ```

    where DOMAINNAME is the NIS domain in step 3.

17. Open /etc/yp.conf in a UNIX text editor.
18. Add the following entry:

    ```
    domain NISDOMAIN server NISSERVERNAME_IP
    ```

    where:

    - NISDOMAIN is the NIS domain in step 3.
    - NISSERVERNAME_IP is the NIS server hostname or IP address.

    Examples:

    ```
    domain hq server 122.138.190.3
    domain hq server unit.example.com
    ```

19. Set **ypbind** to automatically start by typing:

    ```
    /sbin/chkconfig ypbind on
    ```

20. Confirm the previous settings by typing:

    ```
    /sbin/chkconfig --list ypbind
    ```

    The following information appears in the command shell:

    ```
    ypbind 0:off 1:off 2:off 3:on 4:on 5:on 6:off
    ```

    Numbers 3, 4, and 5 should be “on”. If not, type:

    ```
    /sbin/chkconfig --level NUMBERS ypbind on
    ```

    where NUMBERS is a comma-separated list of the numbers to set "on" (for example, /sbin/chkconfig --level 3,4 ypbind on).

21. Start the **ypbind** daemon by typing:

    ```
    service ypbind restart
    ```

    The following information appears in the command shell:

    ```
    Shutting down NIS services: [ OK or FAIL ]
    Binding to the NIS domain: [ OK ]
    Listening for NIS domain server:
    ```

    Note: Shutting down NIS services can fail if it has not started already. In that case, listening for the NIS domain server should fail because the default NIS domain has not yet been set up.

    A delay in the start() section is usually required between the **ypbind** and **ypwhich** (in next step) commands.

22. Confirm NIS configuration by typing:

    ```
    ypwhich
    ```

This command displays the IP address or the fully-qualified domain name of the NIS server.

```
ypcat -d NISDOMAIN passwd | grep USER-ID
```

where:

- NISDOMAIN is the NIS domain in step 3.
- USER-ID is the partial or whole name of a user registered in the external authentication system.

These commands verify that data can be retrieved from the NIS domain server by returning user login data from the NIS server.

CHAPTER 3
Client/Server Access and Authentication

The following topics provide details on Avamar client and server access and authentication:

Network access control

The following topics provide details on network access control in an Avamar environment:

Subnet and gateway assignments

Avamar client machines must be able to connect to every node in the Avamar environment directly, and each node in the environment must be able to connect to the client machines.

Assign a default gateway to the router in the Avamar environment.

DNS requirements

The Avamar environment requires a Domain Name System (DNS) server.

If you have a single-node Avamar server, then assign a forward mapping and optionally a reverse mapping to the server.

If you have a multi-node Avamar server, then assign a forward mapping and optionally a reverse mapping to the utility node.

An example of a forward mapping entry might be as follows in a Berkeley Internet Name Domain (BIND) environment:

```
avamar-1     A          10.0.5.5
```

A corresponding optional reverse mapping for a zone serving the 5.0.10.in-addr.arpa subnet in a BIND environment might be as follows:

```
5      PTR        avamar-1.example.com.
```

Remote access control

Protect all nodes and the switch in the Avamar server against unauthorized access. Use a Virtual Private Network (VPN) system if remote access to the Avamar server is required.

SNMP access configuration

Avamar supports system monitoring and event notification through the Simple Network Management Protocol (SNMP), as discussed in "Event notification mechanisms" on page 89.

Before Avamar release 4.1, SNMP was configured by default to provide read-only access through the public community. This community presents a medium-level security vulnerability.

In release 4.1 and later, the default community name is AvCom (Avamar Community), which provides a higher level of security.

You can change the SNMP configuration from the public community to the AvCom community in releases earlier than 4.1 by editing the SNMP configuration file, snmpd.conf, on each node in the Avamar system.

To change the community name to AvCom on each node in Avamar releases earlier than 4.1:

1. Open the /etc/snmp/snmpd.conf file in a UNIX text editor such as **vi** or **emacs**.
2. Go to the line "com2sec notConfigUser default public."
3. Change the community name from public to AvCom:

   ```
   com2sec notConfigUser default AvCom
   ```

4. Save the /etc/snmp/snmpd.conf file.
5. Restart the **snmpd** agent.
6. Repeat steps 1–5 on each node in the Avamar system.

NOTICE

Dell omreport actively uses SNMP. According to Dell, changing the public community string to a different value does not affect functionality.

Client/server authentication

Avamar clients and servers use Transport Layer Security (TLS) certificates and Public Key Infrastructure (PKI) for authentication and optional encryption of data in transit. TLS and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide secure communications on the Internet for activities such as web browsing, email, Internet faxing, instant messaging, and other data transfers. Although essentially the same, there are minor differences between SSL and TLS.

Avamar supports the X.509 v3 standard for formatting digital certificates. To sign the certificates, you can:

- Use a commercial certification authority (CA), such as Verisign.
- Generate your own root certificate and set up a private CA.
- Self-sign, although self-signing is not recommended in production environments, and therefore, is not discussed in detail in this guide.

Note: When the Avamar server is installed, a public/private key pair and a self-signed certificate are generated automatically in the /data01/home/admin directory on each Avamar server storage node and in the /usr/local/avamar/etc directory on the utility node. However, because self-signing is not recommended in production environments, you should generate and install a key and signed certificate from either a commercial or private CA.

You can configure the Avamar environment for one-way or two-way authentication between Avamar clients and the Avamar server:

- With one-way authentication, the Avamar client requests authentication from the Avamar server, and the server sends the appropriate certificate to the client. The client then validates the certificate. This is also called server-to-client authentication in this guide.
- With two-way authentication, the client requests authentication from the Avamar server, and then the Avamar server also requests authentication from the client. Client-to-server authentication can be set up in addition to server-to-client authentication to provide a stronger level of security.

One-way authentication typically provides sufficient security. However, in some cases, two-way authentication is required or preferred.

In both configurations, all network data can be encrypted. Encryption is discussed in "Encrypting data" on page 78.

The following topics provide details on how to configure Avamar for client/server authentication:

- "Configuring server-to-client authentication" on page 44
- "Configuring client-to-server authentication" on page 48
- "Setting up a private certification authority" on page 52
- "Verifying client/server authentication" on page 59

Configuring server-to-client authentication

With server-to-client (one-way) authentication, the Avamar client requests authentication from the Avamar server, and the server sends the appropriate certificate to the client. The client then validates the certificate.

Generating and installing a server certificate

To generate and install a unique server authentication certificate for each Avamar server node (both the utility node and all data nodes) for server-to-client authentication:

1. Generate a private key and Certificate Signing Request (CSR) for the certificate for each Avamar server node:

   Note: Ensure that the CSR that you create contains the Avamar server node IP address in the Alternative Subject Name field. If nodes use multiple IP addresses (multihomed servers, servers behind Network Address Translation (NAT), etc.), then ensure that each IP address is added to the Alternative Subject Name field.

   a. If you have not already done so, download and install OpenSSL on the system that will generate the certificates and CSRs.

      Note: OpenSSL is available for Linux, Windows, OpenBSD, and other operating systems. For maximum security, use the OpenBSD operating system as the host for the OpenSSL key and certificate utilities.

b. Using the same account that you used to install OpenSSL, open a command shell and type the following on a single command line:

```
openssl req -new -newkey rsa:1024 -keyform PEM
 -keyout avamar-1key.pem -nodes -outform PEM
 -out avamar-1req.pem
```

where:

- avamar-1 is the Avamar server name.
- avamar-1key.pem is the file name for the key.
- avamar-1req.pem is the file name for the CSR.

Note: The OpenSSL website at www.openssl.org provides additional details on **openssl req**.

Note: Space limitations in this guide caused the previous command to continue (wrap) to more than one line. Type the command on a single line (no line feeds or returns allowed).

The following information appears in the command shell:

```
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.++++++
...++++++
writing new private key to 'avamar-1key.pem'
-----
```

c. When prompted, type the information described in the following table, and press **Enter** after each entry.

Table 9 Server certificate information (page 1 of 2)

Name field	Description
Distinguished Name (DN)	Unique name for this particular server node. For example: `avamar-1.node-1`
Country Name	The two-letter ISO abbreviation for the country. For example: `US` The list of abbreviations is available on the ISO website at www.iso.org.
State or Province Name	In countries where it is applicable, the state or province where the organization is located. For example: `California` **Note:** This entry cannot be abbreviated.
Locality Name	City where the organization is located. For example: `Los Angeles`

Table 9 Server certificate information (page 2 of 2)

Name field	Description
Organization Name	The exact legal name of the company. For example: `Example, Inc.` **Note:** This entry cannot be abbreviated.
Organizational Unit Name	Optional entry for additional organization information, such as a department name.
Common Name	A name for the certificate. For example: `example.com Certificate Authority`
Email Address	Primary email address for this server. For example: `avamar-1-admin@example.com`
Challenge password	A password that must be provided before the certificate can be revoked by the CA. Only used if your certificate is compromised. This is an optional field. To skip this field enter a period character.
Company name	Name for your company. The exact legal name is not required. This is an optional field. To skip this field enter a period character.

The information that you specify is incorporated into the CSR.

Note: If you type a period (.) and press **Enter** for an entry, the entry is left blank.

The output from avamar-1req.pem is similar to the following:

```
-----BEGIN CERTIFICATE REQUEST-----
ABCDEF...
...XYZ=
-----END CERTIFICATE REQUEST-----
```

The output from avamar-1key.pem is similar to the following:

```
-----BEGIN RSA PRIVATE KEY-----
ABCDEF...
...XYZ=
-----END RSA PRIVATE KEY-----
```

2. Repeat step 1 on page 44 for every node on the Avamar server.
3. Do one of the following:
   - If using a commercial CA, such as Verisign, to sign certificates, submit the CSRs, such as avamar-1req.pem, to the commercial CA to be signed.

- If you want to set up a private CA to sign certificates:
  - Generate the root certificate and key as discussed in "Generating a root certificate and key" on page 53.
  - Sign the server certificates with the root certificate and key by performing the steps in "Signing certificates" on page 55 once for each server certificate.

Note: Self-signing certificates is possible but not recommended in production environments, and therefore, is not discussed in detail in this guide.

4. Copy the signed server certificate to /data01/home/admin/cert.pem and the and private key to /data01/home/admin/key.pem on each Avamar server storage node.
5. Copy the cert.pem and key.pem files to /usr/local/avamar/etc on the utility node.
6. Stop and restart the Avamar server by typing:

```
dpnctl stop gsan
dpnctl start
```

7. Configure the Management Console Server (MCS):
   a. Set the encrypt_server_authenticate value in the /usr/local/avamar/var/mc/server_data/prefs/mcserver.xml file by typing:

```
encrypt_server_authenticate=true
```

   b. Restart the MCS by typing:

```
dpnctl stop mcs
dpnctl start
```

8. Select either a **Medium** or **High** encryption level for future client communication:
   - When you create and edit groups with Avamar Administrator, select **Medium** or **High** from the Encryption method list.

     Note: You also can override the group encryption method for a specific client on the Client Properties tab of the Edit Client dialog box, for a specific backup on the On Demand Backup Options dialog box, or for a specific restore on the Restore Options dialog box. The *EMC Avamar Administration Guide* provides details.

   - When you use the **avtar** command, use the **--encrypt=tls-sa** option and either the **--encrypt-strength=medium** option or the **--encrypt-strength=high** option.

     Note: If Avamar 4.0 or earlier is installed on the Avamar client, then use the **avtar** command with the **--encrypt=sslverify** option.

"Encrypting data" on page 78 provides additional details on encryption of Avamar data.

Copy or import the root or intermediate CA certificates to the client

You must also copy or import CA certificates to the client. Instructions for this vary depending on the platform involved.

Copy the root or intermediate CA certificates on Linux

To copy the root or intermediate CA certificates on Linux:

1. Create a file on the client called /usr/local/avamar/etc/chain.pem
2. Copy the content of the rootca.pem (and any intermediate CAs that signed the server certificate) into the chain.pem file.

Import the root and intermediate CA certificates on Windows

On Windows systems, for any non-commercial (i.e., private root) CA, and any intermediate CAs that signed the server certificate, you must import the rootca.pem to the client. To import the file, follow step 2 through step 14 in the section "Installing a client certificate on a windows client" on page 50.

Configuring client-to-server authentication

With client-to-server authentication, the Avamar server requests authentication from the client. Client-to-server authentication is used in a two-way authentication environment, where it is configured in addition to server-to-client authentication.

To configure the Avamar environment for client-to-server authentication:

1. Configure server-to-client authentication as discussed in "Configuring server-to-client authentication" on page 44.
2. Generate a unique private key (key.pem) and a single generic client authentication certificate (cert.pem) for use on all clients, as discussed in "Generating a client certificate" on page 48. For stricter client validation, repeat the process for each client, generating and distributing the appropriate keys to client.
3. Configure the Avamar server to request the client certificate, as discussed in "Configuring the Avamar server to request a client certificate" on page 50.
4. Install the client certificate on the client, as discussed in the following topics:
   - "Installing a client certificate on a windows client" on page 50
   - "Installing a client certificate on a UNIX client" on page 52

Generating a client certificate

To generate a unique private key and a single generic client authentication certificate for use on all clients:

1. Using the same account that you used to install OpenSSL, open a command shell and type the following on a single command line to generate a unique private key and CSR:

   ```
   openssl req -new -newkey rsa:1024 -keyform PEM
    -keyout avamarclientkey.pem -nodes -outform PEM
    -out avamarclientreq.pem
   ```

   where:
   - avamarclientkey.pem is the file name for the key.
   - avamarclientreq.pem is the file name for the CSR.

Note: Space limitations in this guide caused the previous command to continue (wrap) to more than one line. Type the command on a single line (no line feeds or returns allowed).

The following information appears in the command shell:

```
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.++++++
...++++++
writing new private key to 'avamarclientkey.pem'
-----
```

2. When prompted, type the information in the following table, and press **Enter** after each entry.

Table 10 Client certificate information

Name Field	Description
Country Name	The two-letter ISO abbreviation for the country. For example: `US` The list of abbreviations is available on the ISO website at www.iso.org.
State or Province Name	In countries where it is applicable, the state or province where the organization is located. For example: `California` **Note:** This entry cannot be abbreviated.
Locality Name	City where the organization is located. For example: `Los Angeles`
Organization Name	The exact legal name of the company. For example: `Example, Inc.` **Note:** This entry cannot be abbreviated.
Organizational Unit Name	Optional entry for additional organization information, such as a department name.
Common Name	A name for the certificate. For example: `Generic Avamar Backup Client`
Email Address	Contact email address for all CA-related issues. For example: `CA-admin@example.com`
Challenge Password	A password that all users of this certificate must know in order to be authenticated.
Optional Company Name	Optional entry.

The information that you specify is incorporated into the CSR.

Note: If you type a period (.) and press **Enter** for an entry, the entry is left blank.

The output from avamarclientreq.pem is similar to the following:

```
-----BEGIN CERTIFICATE REQUEST-----
ABCDEF
..XYZ=
-----END CERTIFICATE REQUEST-----
```

The output from avamarclientkey.pem is similar to the following:

```
-----BEGIN RSA PRIVATE KEY-----
ABCDEF
..XYZ=
-----END RSA PRIVATE KEY-----
```

3. Do one of the following:
   - If you use a commercial CA, such as Verisign, to sign certificates, submit the CSR (avamarclientreq.pem in the example in this procedure) to the commercial CA to be signed.
   - If you set up a private CA to sign certificates, sign the client certificate with the root certificate and key by performing the steps in "Signing certificates" on page 55.

Note: Self-signing certificates is possible but not recommended in production environments, and therefore, is not discussed in detail in this guide.

Configuring the Avamar server to request a client certificate

To configure the Avamar server to enforce a requirement for client certificates:

1. Stop the Avamar server by typing:

   ```
   dpnctl stop gsan
   ```

2. Append the certificate (from the server's certificate signer) to the chain.pem file on the utility node, and each storage node.

   The utility node chain.pem file is located in /usr/local/avamar/etc/chain.pem; the storage node chain.pem files are located in /data01/home/admin/chain.pem.

   If any of the chain.pem files do not exist, copy the certificate to chain.pem.

3. Restart the Avamar server by typing:

   ```
   dpnctl start
   ```

4. Enable client authentication by typing:

   ```
   avmaint config verifypeer=yes --avamaronly
   ```

Installing a client certificate on a windows client

To install a client authentication certificate on a Windows client:

1. Combine the key and signed client certificate into a pkcs#12 format file suitable for importing into a Microsoft Certificate Store by typing:

   ```
   openssl pkcs12 -in avamarclientcert.pem
    -inkey avamarclientkey.pem -export -out avamarclientcert.p12
    -name "Avamar Trusted Client"
   ```

where:

- avamarclientcert.pem is the file name of the signed certificate.
- avamarclientkey.pem is the file name of the key.
- avamarclientcert.p12 is the file name of the resulting pkcs#12 file.

Note: Space limitations in this guide caused the previous command to continue (wrap) to more than one line. Type the command on a single command line (no line feeds or returns allowed).

The following information appears in the command shell:

```
Loading 'screen' into random state - done
Enter Export Password: mypassword
Verifying - Enter Export Password: mypassword
```

2. Log in to the Windows client computer by using an account with local administrator privileges.
3. Open the Microsoft Management Console:
   a. Open the Windows **Start** menu and select **Run**.

      The Run dialog box appears.

   b. Type **mmc** and press **Enter**.

      The Microsoft Management Console appears.
4. From the **File** menu, select **Add/Remove Snap-in**.

   The Add/Remove Snap-In dialog box appears.
5. On the **Standalone** tab, click **Add**.

   If installing on Windows Vista, perform the following steps:

   a. Click **Add**.

   b. Select Computer Account and press **Enter** twice.

   c. Click **OK**.

   The Add Standalone Snap-in dialog box appears.
6. Select Certificates from the list and click **Add**.

   The Certificates snap-in dialog box appears.
7. Select **Computer account**, and then click **Next**.

   The Select Computer dialog box appears.
8. Leave the default selection of **Local computer**, and then click **Finish**.
9. Click **Close** on the Add Standalone Snap-in dialog box.
10. Click **OK** on the Add/Remove Snap-in dialog box.

    The Certificates (Local Computer) Management console is visible in the tree.
11. Expand the following nodes in the console tree: **Certificates (Local Computer)** › **Trusted Root Certification Authorities** › **Certificates**.

12. Right-click the **Certificates** node and select **All tasks** › **Import**.

    The Certificate Import Wizard appears.

13. Click **Next**, and then click **Browse**.
14. Navigate to the location of the file holding the client trusted root certificate and click **Open**.
15. Click **Next** and proceed through the remainder of the wizard.

Installing a client certificate on a UNIX client

To install a signed client certificate on a UNIX client:

1. Copy the key and signed client authentication certificate to SYSDIR (/usr/local/avamar/etc).

   The generated client key file must be named key.pem, and the client certificate file must be named cert.pem. The root CA's certificate must be in the chain.pem file.

2. Append the certificate from the server's certificate signer to the chain.pem file on the Avamar client.

   Note: The chain.pem file is located in SYSDIR (/usr/local/avamar/etc) on the Avamar client.

   If chain.pem does not exist, copy the certificate to chain.pem.

Setting up a private certification authority

You can sign both server and client authentication certificates either by using a commercial CA, such as Verisign, or by setting up a private CA.

Note: fSelf-signing certificates, although possible, is not recommended in production environments, and therefore, is not discussed in detail in this guide.

There are multiple ways to set up a private CA. One way is to use OpenSSL tools.

To use OpenSSL tools to set up a private CA to sign certificates, perform the following steps:

1. Generate a root certificate and key, as discussed in "Generating a root certificate and key" on page 53.
2. Sign the server and/or client certificates using the steps in "Signing certificates" on page 55.

   Note: You do not need to perform the steps in these topics if you use a commercial CA to sign certificates.

Generating a root certificate and key

If you plan to set up a private CA to sign authentication certificates, then you first must generate a root certificate and key by using OpenSSL tools.

When creating and signing certificates, EMC recommends that you:

- Properly secure the private key associated with the root certificate.
- Use an air-gapped network in a high-risk environment for signing operations and creating keys, CSRs, and other security-related artifacts. (An air-gapped network is completely physically, electrically, and electromagnetically isolated.)
- Use a hardware Random-number Generator (RNG) to efficiently and quickly generate random numbers with adequate characteristics for cryptographic use.
- For maximum security, use the OpenBSD operating system as the host for the OpenSSL key and certificate utilities.

Note: You do not need to generate a root certificate and key if you use a commercial CA, such as Verisign, to sign certificates.

To generate a root certificate and key with openssl req:

1. If you have not done so already, download and install OpenSSL and a Perl interpreter on the system that will generate the certificate.

   Note: OpenSSL and Perl interpreters are available for Linux, Microsoft Windows, OpenBSD, and other operating systems.

2. Using the same account that you used to install OpenSSL, open a command shell and type:

   ```
   openssl req -new -x509 -newkey rsa:1024 -keyform PEM
    -keyout examplekey.pem -extensions v3_ca -outform PEM
    -out exampleca.pem -days 3654
   ```

   where the **-days 3654** option certifies the certificate for 3,654 days. You can set the **-days** option to any period of time.

   Note: Space limitations in this guide caused the previous command example to continue (wrap) to more than one line. Type the command on a single line (no line feeds or returns allowed).

   Note: The OpenSSL website at www.openssl.org provides additional details on **openssl req**.

   The following prompt appears:

   ```
   Enter PEM pass phrase
   ```

3. Enter a pass phrase for the key.

   The pass phrase should be memorable. It cannot be retrieved.

   The following prompt appears:

   ```
   Verifying - Enter PEM pass phrase
   ```

4. Re-enter the pass phrase for the key.
5. When prompted, type the information as described in the following table, and press **Enter** after each entry.

Table 11 Root certificate with openssl req information

Name Field	Description
Country Name	The two-letter ISO abbreviation for the country. For example: `US` The list of abbreviations is available on the ISO website at www.iso.org.
State or Province Name	In countries where it is applicable, the state or province where the organization is located. For example: `California` **Note:** This entry cannot be abbreviated.
Locality Name	City where the organization is located. For example: `Los Angeles`
Organization Name	The exact legal name of the company. For example: `Example, Inc.` **Note:** This entry cannot be abbreviated.
Organizational Unit Name	Optional entry for additional organization information, such as a department name.
Common Name	The name of the certificate. For example: `example.com Certificate Authority`
Email Address	Contact email address for all CA-related issues. For example: `CA-admin@example.com`

   Note: If you type a period (.) and press **Enter** for an entry, the entry is left blank.

   The files with the CA certificate (exampleca.pem) and certificate key (examplekey.pem) are created.

6. Back up exampleca.pem and examplekey.pem.

Signing certificates

After you generate a root certificate and key as described in "Generating a root certificate and key" on page 53, you can create signed X.509 certificates for servers and clients.

NOTICE

You do *not* need to generate a self-signed x509 certificate if you use a commercial CA, such as Verisign, to sign the server certificates.

The procedure assumes the following:

- The CA certificate is in exampleca.pem.
- The key for the CA certificate is in examplekey.pem.
- The example.srl serial number seed file does not already exist.
- The default openssl.cnf file that is provided with OpenSSL is modified to include information specific to your organization.

To modify openssl.cnf for your organization:

1. For server certificates, add the following at the end of openssl.cnf:

```
[ server_ext ]
basicConstraints = CA:false
keyUsage = critical, digitalSignature, keyEncipherment
nsCertType = server
extendedKeyUsage = serverAuth
nsComment = "OpenSSL-generated server certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always, issuer:always
subjectAltName = @alt_names
[alt_names]
IP.0 = NNN.NNN.NNN.NNN
# add ip for multihomed server or NAT
#IP.1 = 1.2.3.4
DNS.0 = dnsserver.example.com
#add hostnames for multihomed server or NAT
#DNS.1 = natavds.example.com
```

   where:

   - NNN.NNN.NNN.NNN represents an IP address for the server .
   - dnsserver.example.com represents a hostname for the server.

2. For client certificates, add the following after the server entry made in the previous step.

```
[ client_ext ]
basicConstraints = CA:false
keyUsage = critical, digitalSignature, keyEncipherment
nsCertType = client
extendedKeyUsage = clientAuth
nsComment = "OpenSSL-generated client certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always, issuer:always
subjectAltName = @alt_names
[alt_names]
IP.0 = NNN.NNN.NNN.NNN
# add ip for multihomed server or NAT
#IP.1 = 1.2.3.4
DNS.0 = dnsserver.example.com
#add hostnames for multihomed server or NAT
#DNS.1 = natavds.example.com
```

   where:

   - NNN.NNN.NNN.NNN represents an IP address for the client
   - dnsserver.example.com represents a hostname or IP address for the DNS server.

To sign a server certificate request and generate the signed certificate:

1. Type the following command on a single line:

```
openssl x509 -CA expampleca.pem -CAkey examplekey.pem
 -req -in avamar-1req.pem -extensions server_ext
 -extfile openssl.cnf -outform PEM -out avamar-1cert.pem
 -days 3650 -CAserial example.srl -CAcreateserial
```

   where:

   - expampleca.pem is the file name for the CA certificate, examplekey.pem is the certificate key.
   - avamar-1req.pem is the file name of the CSR.
   - avamar-1cert.pem is the file name of the resulting signed certificate.

   Note: Space limitations in this guide cause the previous command example to continue (wrap) to more than one line. Type the command on a single command line (no line feeds or returns allowed).

   The following information appears in the command shell:

```
Loading 'screen' into random state - done
Signature ok
subject=/C=US/ST=California/L=Los Angeles/O=Example,
Inc./OU=Dept55/CN=avamar-1.example.com/emailAddress=avamar-1-admin@
example.com
Getting CA Private Key
Enter pass phrase for examplekey.pem:
```

2. Type the passphrase for the certificate key and press **Enter**.

   The content of signed certificate looks similar to the following output:

```
-----BEGIN CERTIFICATE-----
ABCDEF...
...XYZ=
-----END CERTIFICATE-----
```

3. Display the certificate content in text by typing:

   openssl x509 -in avamar-1cert.pem -noout -text

   The following information appears in the command shell:

```
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number:
         9f:3a:d1:2d:93:2d:3d:92
      Signature Algorithm: sha1WithRSAEncryption
      Issuer: C=US, ST=California, O=Example, Inc., OU=Dept55,
       CN=example.com
       Certificate Authority/emailAddress=avamar-1.example.com
      Validity
         Not Before: May 16 20:21:12 2008 GMT
         Not After : May 16 20:21:12 2009 GMT
      Subject: C=US, ST=California, L=Los Angeles, O=Example, Inc.,
       OU=Dept55,

CN=avamar-1.example.com/emailAddress=avamar-1-admin@example.com
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:c2:e2:f9:b8:77:9a:06:fe:6d:1d:c8:9d:04:3a:
               7d:75:aa:1e:8d:4a:57:34:f7:a6:4e:30:73:80:ca:
               c0:38:be:e9:e5:04:1b:05:42:79:b1:07:40:59:b7:
               3f:7f:79:21:2d:95:74:96:6f:25:ce:16:b8:ae:72:
               b1:b4:76:e7:fd:45:28:87:50:fd:76:b2:fe:c3:c2:
               cd:20:ee:54:40:2a:56:55:ca:d4:f4:df:ae:29:6b:
               4b:84:18:98:b7:ff:be:04:4e:bf:b5:9a:a7:39:ba:
               2e:87:3e:ea:d0:ae:8a:ec:d4:6a:7c:f3:cb:79:0b:
               b9:a9:83:28:67:80:e2:e1:dd
            Exponent: 65537 (0x10001)
      X509v3 extensions:
         X509v3 Basic Constraints:
            CA:FALSE
```

```
            X509v3 Key Usage: critical
               Digital Signature
            Netscape Cert Type:
               SSL Client
            X509v3 Extended Key Usage:
               TLS Web Client Authentication
            Netscape Comment:
               OpenSSL-generated server certificate
            X509v3 Subject Key Identifier:

A5:29:93:8E:98:E1:FB:4E:7A:2A:5A:A0:AB:76:A6:C5:18:F1:78:0A
            X509v3 Authority Key Identifier:

keyid:DA:27:CF:99:D1:EB:C2:2C:93:50:9D:09:B7:20:E0:31:7E:D6:84:09

DirName:/C=US/ST=California/O=example.com/OU=Dept55/CN=example.com
                Certificate
Authority/emailAddress=avamar-1@example.com
               serial:DA:2D:59:E2:4F:E2:91:F8
   Signature Algorithm: sha1WithRSAEncryption
       9e:10:07:a7:1a:e8:7e:5c:b1:87:0d:81:5a:70:49:2c:86:e6:
       4c:36:93:31:4e:bf:f6:bf:de:02:52:66:25:c0:67:e9:a5:dc:
       5d:bf:9c:10:b6:77:c4:ce:a8:18:8d:6f:1d:e2:32:e5:01:56:
       20:86:f8:c3:9d:01:e6:dc:f4:0d:56:fc:22:dc:f7:be:64:42:
       cf:1e:ca:cb:7d:18:7b:8e:c0:ca:64:33:a1:aa:e5:1a:b6:1b:
       9f:f0:c8:19:55:c4:88:c1:77:bb:16:da:58:63:22:7d:ba:ff:
       9e:bc:c8:11:3f:37:cb:5e:a9:8d:dd:3b:f3:e6:cd:56:2f:2a:
       47:e9
   f3:f8
```

4. Verify authentication as described in "Verifying client/server authentication" on page 59.

To sign a client certificate request and generate the signed certificate:

1. Type the following command on a single line:

   ```
   openssl x509 -CA exampleca.pem -CAkey examplekey.pem
    -req -in client0req.pem -extensions client_ext
    -extfile openssl.cnf -outform PEM -out client0cert.pem
    -days 3650 -CAserial example.srl -CAcreateserial
   ```

   where:

   - exampleca.pem is the file name for the CA certificate.
   - examplekey.pem is the certificate key.
   - client0req.pem is the file name of the CSR.
   - client0cert.pem is the file name of the resulting signed certificate.

   Note: Space limitations in this guide cause the previous command example to continue (wrap) to more than one line. Type the command on a single command line (no line feeds or returns allowed).

The following information appears in the command shell:

```
Loading 'screen' into random state - done
Signature ok
subject=/C=US/ST=California/L=Los Angeles/O=Example,
Inc./OU=Dept55/CN=client0.example.com/emailAddress=client0-admin@ex
ample.com
Getting CA Private Key
Enter pass phrase for examplekey.pem:
```

2. Type the passphrase for the certificate key and press **Enter**.

   The content of signed certificate looks similar to the following output:

```
-----BEGIN CERTIFICATE-----
ABCDEF...
...XYZ=
-----END CERTIFICATE-----
```

3. Verify authentication as described in "Verifying client/server authentication" on page 59.

Verifying client/server authentication

To verify authentication, run a test backup. Use either the **avtar** command from the command line or Avamar Administrator.

Verifying authentication with the avtar command

To verify client/server authentication by using the **avtar** command with an encryption option:

- For Avamar clients running 4.1 or later, use the **avtar** command with the **--encrypt=tls-sa** option.
- For Avamar clients running 4.0 or before, use the **avtar** command with the **--encrypt=sslverify** option.

The **--encrypt=tls-sa** and **--encrypt=sslverify** options verify the identity of the Avamar server to the Avamar client.

Verifying authentication with Avamar Administrator

To verify client/server authentication with Avamar Administrator 4.1 or later, run a backup and select medium or high from the Encryption method list. The Encryption method list appears on both the On Demand Backup Options dialog box and the Restore Options dialog box.

The *EMC Avamar Administration Guide* provides more information on how to run a backup with the Avamar Administrator.

Note: If you block non-TLS (port 27000) traffic to Avamar with a firewall, then only authenticated clients can connect to the server. To connect to the server, Avamar 4.1 or later clients must use the **--encrypt=tls** option, and clients with an earlier release must use the **--encrypt=ssl** option. All clients also must use properly signed certificates to authenticate themselves to the server.

Web browser authentication using Apache

Avamar Enterprise Manager and Avamar client web UI each use the Apache web server to provide a secure web browser-based user interface. Web browser connections for these applications use secure socket layer/transport layer security (SSL/TLS) to provide authentication and data security.

When a web browser accesses a secure web page from an unauthenticated web server the SSL/TLS protocol causes it to display an authentication warning. An unauthenticated web server is one that does not authenticate itself using a trusted public key certificate.

The Apache web server provided with Avamar is installed with a self-signed certificate not a trusted public key certificate. The self-signed certificate is sufficient to establish an encrypted channel between web browsers and the server, but it cannot be used for authentication.

To provide server authentication, and thereby prevent web browser warnings complete the following tasks:

- "Create a private key" on page 60
- "Generate a certificate signing request" on page 62

The tools used in these tasks are part of the OpenSSL toolkit. OpenSSL is provided with Avamar.

Alternative authentication method

Authentication of the Tomcat server used by Avamar Enterprise Manager should normally be handled by Apache as described in this section. Then the Tomcat server is not required to provide server authentication and does not require a separate trusted public key certificate.

However, the Tomcat server does listen on port 8543 and is configured to provide SSL/TLS authentication on that port. If your organization connects to Avamar Enterprise Manager directly on port 8543 you may want to install a trusted public key certificate for the Tomcat server.

"RMI and Tomcat server authentication" on page 65 describes how to replace the Tomcat server's default self-signed certificate with a trusted public key certificate.

Create a private key

A private key can be generated with pass phrase protection and without pass phrase protection. It can also be generated using a random key generation algorithm. Use the method that is appropriate for the level of security required by your organization.

Note: When a password protected private key is used, Apache prompts for the passphrase at startup. The configuration setting SSLPassPhraseDialog can be used to obtain the passphrase from a script. For more information, refer to Apache documentation available through the Apache website at www.apache.org.

To create a private key without a passphrase and without additional randomness:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   su -

3. Create the private key by typing:

   openssl genrsa -out server.key 2048

   where server.key is a name you provide for the private key.

   The private key is created in the current working directory.

To create a private key using a random key generation algorithm:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   su -

3. Create the private key by typing:

   openssl genrsa -rand binary-files -out server.key 2048

   where server.key is a name you provide for the private key and binary-files is a colon-separated list of paths to two or more binary files.

   The private key is created in the current working directory.

To create a pass phrase protected private key:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   su -

3. Create the private key by typing:

   openssl genrsa -des3 -out server.key 2048

   where server.key is a name you provide for the private key.

   The following prompt appears:

   ```
   Enter pass phrase for server.key:
   ```

4. Enter a pass phrase and press **Enter**.

   The following prompt appears:

   ```
   Verifying - Enter pass phrase for server.key:
   ```

5. Re-enter the pass phrase and press **Enter**.

   The private key is created in the current working directory.

Generate a certificate signing request

Apply for a public key certificate from a Commercial CA, by sending the CA a certificate signing request (CSR).

To generate a CSR:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Generate the CSR by typing:

   ```
   openssl req -new -key server.key -out server.csr
   ```

   where:

   - server.key is a name you provide for the private key.
   - server.csr is a name you provide for the CSR.

4. (Pass phrase protected private key only) Enter the pass phrase for the private key and press **Enter**.
5. Provide the Distinguished Name (DN) information as requested and press **Enter**.

   The tool prompts for DN information. At each prompt, type the information described in the following table, and press **Enter** after each entry.

Table 12 Certificate signing request distinguished name information (page 1 of 2)

Name field	Description
Country Name	The two-letter ISO abbreviation for the country. For example: `US` The list of abbreviations is available on the ISO website at www.iso.org.
State or Province Name	In countries where it is applicable, the state or province where the organization is located. For example: `California` **Note:** This entry cannot be abbreviated.
Locality Name	City where the organization is located. For example: `Los Angeles`
Organization Name	The exact legal name of the company. For example: `Example, Inc.` **Note:** This entry cannot be abbreviated.

Table 12 Certificate signing request distinguished name information (page 2 of 2)

Name field	Description
Organizational Unit Name	Optional entry for additional organization information, such as a department name.
Common Name	The fully qualified domain name of the Avamar server (single-node) or utility node (multi-node). For example: `avamar-1.example.com`
Email Address	Primary email address for this server. For example: `avamar-1-admin@example.com`

The following prompt appears:

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

6. Type a password or type **.** and press **Enter**.

   A password is optional. If provided, the certificate cannot be revoked without first entering the password. To skip this step type **.** and press **Enter**.

   The following prompt appears:

   ```
   An optional company name []:
   ```

7. Enter an alternative form of the company name or type **.** and press **Enter**.

   The CSR is created in the current working directory.

Request a public key certificate

Request a public key certificate from a commercial CA. Include the CSR as part of the request.

After its criteria are met, the CA provides a public key certificate, in the form of an electronic file, usually with a file name ending in crt.

The CA may also provide a certificate chain. A certificate chain is a series of certificates that link the public key certificate you receive to a trusted root CA certificate. Combine the certificate chain into a single file.

To combine the certificate chain into a single file:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Use cat with the redirect and append operators to combine the certificates by typing:

```
cat chain-cert-1 > cachain.crt
cat chain-cert-2 >> cachain.crt
cat chain-cert-3 >> cachain.crt
cat chain-cert-4 >> cachain.crt
cat chain-cert-5 >> cachain.crt
```

   where chain-cert-1 through chain-cert-5 represent the path to each certificate in the certificate chain and cachain.crt is a name you provide for the combined file.

Configure Apache to use the key and certificates

Configure Apache to use the private key, public key certificate, and the certificate chain file. Then restart Apache.

IMPORTANT

The certificate, key, and certificate chain file must be installed in the default locations specified in the Apache SSL configuration file. The Apache SSL configuration file is overwritten during upgrades and custom locations will be lost.

To configure Apache to use the certificate, key, and certificate chain file:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

```
su -
```

3. Change the working directory to the temporary location of the certificate, key, and certificate chain file.
4. Move the certificate, key, and certificate chain file to the default location.
   - On Red Hat Enterprise Linux:

```
mv server.crt /etc/httpd/conf/ssl.crt/server.crt
mv server.key /etc/httpd/conf/ssl.key/server.key
mv cachain.crt /etc/httpd/conf/ssl.crt/ca.crt
```

   - On SUSE Enterprise Linux Server:

```
mv server.crt /etc/apache2/ssl.crt/server.crt
mv server.key /etc/apache2/ssl.key/server.key
mv cachain.crt /etc/apache2/ssl.crt/ca.crt
```

5. Restart Apache by typing:

```
website restart
```

RMI and Tomcat server authentication

Several Avamar services are distributed object applications that use Java remote method invocation (RMI). The RMI channels are transported through SSL/TLS sockets. By default these Avamar services share a self-signed certificate for the SSL/TLS sockets. A self-signed certificate is sufficient for encrypted data channels but does not provide adequate server authentication.

The Avamar services that utilize RMI and share a certificate are:

- Enterprise Manager server
- Management Console server
- Management Console command line interface
- Avamar Administrator
- Avamar Administrator via webstart

To provide server authentication for these services, install a trusted public key certificate as described in "Installing a trusted public key certificate" on page 66.

The certificate shared by these services is also shared by Avamar Enterprise Manager's Tomcat server. This is described in "Avamar Enterprise Manager" on page 65.

A trusted public key certificate is installed and stored with other certificates in the root keystore. This keystore is protected by a password, but the default password is commonly known and insecure. To protect the integrity of the keystore's contents, change the password, as described in "Changing the root keystore password" on page 73.

Avamar Enterprise Manager

Avamar Enterprise Manager uses an Apache Tomcat servlet container (Tomcat server) to provide a pure Java, web browser-based, user interface. Web browser connections with the Tomcat server use SSL/TLS to provide authentication and data security.

The SSL/TLS sockets are normally handled by Apache, "SSL/TLS through Apache" on page 65, but they can also be handled directly by the Tomcat server, "SSL/TLS through Tomcat" on page 66.

SSL/TLS through Apache

In most cases Avamar Enterprise Manager SSL/TLS sockets are handled by Apache. This occurs when a connection is made to Avamar Enterprise Manager using a web address of the form:

`http://AVAMARSERVER/em`

where AVAMARSERVER is the resolvable hostname or IP address of the utility node or single-node server.

Apache redirects requests for this address to an SSL/TLS socket and handles that socket. Server authentication for this method is provided by installing a trusted public key certificate for Apache.

Installing a trusted public key certificate for Apache is described in "Web browser authentication using Apache" on page 60.

SSL/TLS through Tomcat

Avamar Enterprise Manager's Tomcat server can also be directly accessed using a web address of the form:

`https://AVAMARSERVER:8543/cas`

where AVAMARSERVER is the resolvable hostname or IP address of the utility node or single-node server.

When an address of this form is used, SSL/TLS sockets are handled by the Tomcat server instead of Apache.

The Tomcat server used by Avamar Enterprise Manager shares a certificate with several Avamar services that utilize RMI. By default this certificate is self-signed. A self-signed certificate is sufficient to establish an encrypted channel between the Tomcat server and web browsers, but it cannot be used for authentication.

When a web browser accesses a secure web page that is served by an unauthenticated web server the SSL/TLS protocol causes the browser to display an authentication warning. An unauthenticated web server is one that does not authenticate itself using a trusted public key certificate.

To provide server authentication when directly accessing the Tomcat server, obtain and install a trusted public key certificate as described in "Installing a trusted public key certificate" on page 66. Using a trusted public key certificate provides valid authentication of Avamar Enterprise Manager and prevents web browser warnings.

Note: This procedure is not required for Avamar Enterprise Manager when it is accessed through Apache as described in "SSL/TLS through Apache" on page 65.

Installing a trusted public key certificate

Install a trusted public key certificate to provide authentication of the Avamar services that utilize RMI and the Avamar Enterprise Manager Tomcat server. This certificate is shared by the Avamar services and the Tomcat server.

The tasks involved in installing a trusted public key certificate are:

1. "Deleting the default key entry" on page 67
2. "Creating a new key entry" on page 67
3. "Generating a certificate signing request" on page 68
4. "Obtaining a public key certificate" on page 70
5. "Importing chained or root certificates" on page 71
6. "Importing the public key certificate" on page 72

Deleting the default key entry

You must delete the default "tomcat" key entry from the keystore before you can create a new key entry that contains your company's information.

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Run the **keytool -delete** command by typing:

   ```
   $JAVA_HOME/bin/keytool -delete -alias tomcat
   ```

4. At the password prompt, type the keystore password and press **Enter**:

   ```
   Enter keystore password: PASSWORD
   ```

   where PASSWORD is the keystore password. The default is "changeit".

   Note: After the trusted public key certificate is installed, change the default password, as described in "Changing the root keystore password" on page 73.

Creating a new key entry

After you delete the default "tomcat" key entry, create a new one that contains your company's information.

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Run the keytool -genkeypair command by typing the following on a single command line:

   ```
   $JAVA_HOME/bin/keytool -genkeypair -keysize 2048 -alias tomcat
   -keyalg RSA
   ```

4. At the password prompt, type the keystore password and press **Enter**:

   ```
   Enter keystore password: PASSWORD
   ```

   where PASSWORD is the keystore password. The default is "changeit".

5. At each prompt, type the information described in the following table, and press **Enter** after each entry.

   Note: To accommodate individuals, the **keytool -genkeypair** command prompts for first and last name. However, in a corporate environment, this prompt should be answered with the fully qualified domain name (FQDN) of the Avamar utility node.

Table 13 Tomcat key fully qualified domain name information

Name field	Description
First and last name	Fully qualified domain name of the Avamar utility node.
Organizational unit	Organizational unit within the company that has authority over the host.
Organization	Name of the company.
City	City in which the host is located.
State	State in which the host is located.
Country	Country in which the host is located.

Generating a certificate signing request

To generate a certificate signing request (CSR) to send to a public certification authority (CA):

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Create the CSR by typing the following command on a single command line:

   ```
   $JAVA_HOME/bin/keytool -certreq -keyalg RSA -alias tomcat -file
   tomcat.certrequest
   ```

4. At the password prompt, type the keystore password and press **Enter**:

   ```
   Enter keystore password: PASSWORD
   ```

   where PASSWORD is the keystore password. The default is “changeit”.

   A CSR named tomcat.certrequest is created in root’s home directory.

The contents of tomcat.certrequest appear similar to the following example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
JIICpDCCAawCAQAwfzELMAkGA1UEBhMCVVMxETAPBgNVBAgTCENvbG9yYWRvMRMwEQY
DVQQHEwpM
b3Vpc3ZpbGxlMRgwFgYDVQQKEw9FTUMgQ29ycG9yYXRpb24xDzANBgNVBAsTBkF2YW1
hcjEdMBsG
A1UEAxMUbGF2YTIwMjIubHGzLmVtYy5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwA
wggEKAoIB
AQCYRPWEDUB5VtSyAIrIFxzd1/5b1tjgMZlHEX+JEAvfTfT+ujNHhgnOWQQmCYEw+A6
/vyVL3ntS
EmAL5ukARTQBsm1PvQR5yKtpGgKdiaNkdDkltC8VXJNifThHZ6ttOKehauE4lp3kJu/
NnZ4B0XlT
60B3hbhQuP97UUBB+fj4h0pKAu0NT/kp3QoGkEnBHRhkfKQUocKP/E1IK6oxXweoEVE
7BJedckot
h5ThP1wSXGWm/MdyTG+wPbWHyNtyc8sZa8p5XiqrlPsIt2t6T4+moModWHF6ggn6+z2
Ok6u3FxST
dyU1r5xcr/0235zPuzekAwRUR1qBVkN3470odwydAgMBAAGgADANBgkqhkiG9w0BAQU
FAAOCAQEA
X0MDz/gG433PnJ9jlZsfXtX2RwBmbU+C1tHxZksa9Sr6tLAPyu0OQh02ZOJ9PONBNNT
6gmH0YMcO
4KiGHEC/6xn1WKpuP+6ErG4h4KF7/H+v49qer1YXQFhWwR466uQcqCWM/iveoowIesw
AlEsvl5So
tHf5Nl1Jnripu+N2874eycAKEFbpmMrfFGBeAsXNeWLa311+VhLHllORTV7lRO46zcr
bmxdUTgat
3WRXIX2XDco8S0Lyf+Od5pASaRTc8SGWS6p8KSbqKrmdPVH5y0GonJp13va1iuY9vNN
SQYM22+po
rdVX0O/ULtuz9lJ2OA+9wAtYqN5Q8CEe18Vwlg==
-----END NEW CERTIFICATE REQUEST-----
```

Obtaining a public key certificate

To apply for a public key certificate through a CA:

1. Contact a CA and apply for the public key certificate.

   The CA requests a copy of the CSR (tomcat.certrequest). The CA also requires the approval of the domain registrant listed for the Avamar utility node's domain. The domain registrant can be determined by using a domain lookup tool on the web.

   After you complete the CA application requirements, the CA provides a public key certificate that looks similar to the following:

   ```
   -----BEGIN CERTIFICATE-----
   JIIEJDCCA42gAwIBAgIRAOiW1j5MGIPZ8zeLbNdSUPgwDQYJKoZIhvcNAQEFBQAw
   gYwxGjAYBgNVBAoTEVJTQSBTZWN1cml0eSBJbmMuMRUwEwYDVQQLEwxLQ0EgU2Vy
   dmljZXMxIDAeBgNVBAMTF1JTQSBDb3Jwb3JhdGUgU2VydmVyIENBMRAwDgYDVQQH
   EwdCZWRmb3JkMRYwFAYDVQQIAw1NYXNzYNNodXNldHRzMQswCQYDVQQGEwJVUzAe
   Fw0xMDAzMjQxMjQ5MjZaFw0xMjAzMjUxMSQ5MDdaMH8xCzAJBgNVBAYTAlVTMREw
   DwYDVQQIEwhDb2xvcmFkbzETMBEGA1UEBxMKTG91aXN2aWxsZTEYMBYGA1UEChMP
   RU1DIENvcnBvcmF0aW9uMQ8wDQYDVQQLExZBdmFtYXIxHTAbBgNVBAMTFGxhdmEy
   MDIyLmxzcy5lbWMuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCCvAfC
   D39UeAmNLoUx6IqjCXoNIticPQOftlwjgohYIvXKE6eWDF1S3HfWtkoexXMt/1hF
   xcREXqQOMcmi9ZHcFRCs1wti6jNh6Qc9O+E/cWKTneEgOWPigbTrB4lmWkKofKz/
   thC1p01ueQwUHMIsT6opIRsXHicsvsbj+Bj70QIDAQABo4IBkDCCAYwwDgYDVR0P
   AQH/BAQDAgO4MBEGCWCGSAGG+EIBAQQEAwIGQDAfBgNVHREEGDAWghRsYXZhMjAy
   Mi5sc3MuZW1jLmNvbTAfBgNVHSMEGDAWgBSJz061oIdefkN40ZlJaAd8hAczVjCB
   kgYDVR0gBIGKMIGHMIGEBgkqhkiG9w0FBzUwdzAuBggrBgEFBQcCARYiaHR0cDov
   L2NhLnJzYXNlY3VyaXR5LmNvxS9DUFMuaHRtbDBFBggrBgEFBQcCAjA5MBgWEVJT
   QSBTZWN1cml0eSBJbmMuMAMCAQEaHUNQUyBJbmNvcnBvcmF0ZWQgYnkgcmVmZXJl
   bmNlMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAdBgNVHQ4EFgQUXGqU
   d5if2l0PyMxNL9errGa+vEQwUgYDVR0fBEswSTBHoEWgQ4ZBaHR0cDovL2NybC5y
   c2FzZWN1cml0eS5jb206ODAvOlNBJTIwQ29ycG9yYXRlJTIwU2VydmVyJTIwQ0Et
   Mi5jcmwwDQYJKoZIhvcNAQEFBQADgYEAFPBGQjw2l2H4Vnvahbas2rdFSN8OFnbh
   VcYBnWAuiA1aOgc1u9ZtmMX1JrGVrS8qCThrUZoGHsLsbDF8wuyFeloelHcdZSY1
   GorkhdbcBR5NVGq5UHB7sbKiDvbMuEf6Gwbier0mps7oEOMU8uh8v2rMTsXEuhtK
   csWTe/IxkOk=
   -----END CERTIFICATE-----
   ```

2. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.

3. Open a root shell by typing:

   su -

4. Save the public key certificate in root's home directory as tomcat.cert.

Importing chained or root certificates

You normally receive a chained or root certificate file along with the public key certificate.

Note: If you do not receive a chained or root certificate file with the public key certificate, skip this topic and proceed to "Importing the public key certificate" on page 72.

To import the chained or root certificate:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Open the chained or root certificate file in a text editor.

   The file contents are similar to the following chained certificate example, which contains two certificates:

   ```
   -----BEGIN CERTIFICATE-----
   MIIDdDCCAt2gAwIBAgIQJyzRkL6Balz4Y8X3iFwiFjANBgkqhkiG9w0BAQUFADCB
   uzEkMCIGA1UEBxMbVmFsaUNlcnQgVmFsaWRhdGlvbiBOZXR3b3JrMRcwFQYDVQQK
   Ew5WYWxpQ2VydCwgSW5jLjE1MDMGA1UECxMsVmFsaUNlcnQgQ2xhc3MgMyBQb2xp
   Y3kgVmFsaWRhdGlvbiBBdXRob3JpdHkxITAfBgNVBAMTGGh0dHA6Ly93d3cudmFs
   aWNlcnQuY29tLzEgMB4GCSqGSIb3DQEJARYRaW5mb0B2YWxpY2VydC5jb20wHhcN
   MDUwNTAyMTczNDQ4WhcNMTkwNDMwMDkyNDAwWjBsMRowGAYDVQQKExFSU0EgU2Vj
   dXJpdHkgSW5jLjEeMBwGA1UEAxMVUlNBIFB1YmxpYyBSb290IENBIHYxMS4wLAYJ
   KoZIhvcNAQkBFh9yc2FrZW9ucm9vdHNpZ25AcnNhc2VjdXJpdHkuY29tMIGfMA0G
   CSqGSIb3DQEBAQUAA4GNADCBiQKBgQC0dSVa2EQ740GxqiIKsHATT4f8XYwUU23p
   zRe5W6IVpt4jkwDWkgnvTcP6M8PD4OfK6Ima14hAH/c3K/dUIH7YyQZRGAE5Y27G
   5klZYKidcFlrfautgVES170MLKFmJqym2FWAfVOibXatRcw7B0S+mP9404jID/Ma
   IpRXkH5JjwIDAQABo4HGMIHDMB0GA1UdDgQWBBT1TDF6UQM/LNeLl5lvqHGQq3g9
   mzBsBgNVHR8EZTBjMGGgX6BdhltodHRwOi8vd3d3LnJzYXNlY3VyaXR5LmNvbS9w
   cm9kdWN0cy9rZW9uL3JlcG9zaXRvcnkvY2VydGlmaWNhdGVfc3RhdHVzL1ZhbGlj
   ZXJ0X1Jvb3RfQ0EuY3JsMAwGA1UdEwQFMAMBAf8wDgYDVR0PAQH/BAQDAgGGMBYG
   A1UdIAQPMA0wCwYJKoZIhvcNBQYBMA0GCSqGSIb3DQEBBQUAA4GBAJsZWGBPVGmc
   xmwH5vM9xqt6r1jjQE44zOFNgwlXp0YR605ss5SjkVlyx4WtjKzSrI4hPLhVJN5f
   69F7NxNQMf658Mkkx3Vv6+orEvHFqIw/Hx4uqmdBRpHy/cckaBcEqhJfew7IUFS+
   4KRrACEZFnBeaZQlTH8J7UqTThT7By2x
   -----END CERTIFICATE-----
   -----BEGIN CERTIFICATE-----
   MIIC5zCCAlACAQEwDQYJKoZIhvcNAQEFBQAwgbsxJDAiBgNVBAcTG1ZhbGlDZXJ0
   IFZhbGlkYXRpb24gTmV0d29yazEXMBUGA1UEChMOVmFsaUNlcnQsIEluYy4xNTAz
   BgNVBAsTLFZhbGlDZXJ0IENsYXNzIDMgUG9saWN5IFZhbGlkYXRpb24gQXV0aG9y
   aXR5MSEwHwYDVQQDExhodHRwOi8vd3d3LnZhbGljZXJ0LmNvbS8xIDAeBgkqhkiG
   9w0BCQEWEWluZm9AdmFsaWNlcnQuY29tMB4XDTk5MDYyNjAwMjIzM1oXDTE5MDYy
   NjAwMjIzM1owgbsxJDAiBgNVBAcTG1ZhbGlDZXJ0IFZhbGlkYXRpb24gTmV0d29y
   azEXMBUGA1UEChMOVmFsaUNlcnQsIEluYy4xNTAzBgNVBAsTLFZhbGlDZXJ0IENs
   YXNzIDMgUG9saWN5IFZhbGlkYXRpb24gQXV0aG9yaXR5MSEwHwYDVQQDExhodHRw
   Oi8vd3d3LnZhbGljZXJ0LmNvbS8xIDAeBgkqhkiG9w0BCQEWEWluZm9AdmFsaWNl
   cnQuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjmFGWHOjVsQaBalfD
   cnWTq8+epvzzFlLWLU2fNUSoLgRNB0mKOCn1dzfnt6td3zZxFJmP3MKS8edgkpfs
   2Ejcv8ECIMYkpChMMFp2bbFc893enhBxoYjHW5tBbcqwuI4V7q0zK89HBFx1cQqY
   JJgpp0lZpd34t0NiYfPT4tBVPwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFa7AliE
   Zwgs3x/be0kz9dNnnfS0ChCzycUs4pJqcXgn8nCDQtM+z6lU9PHYkhaM0QTLS6vJ
   n0WuPIqpsHEzXcjFV9+vqDWzf4mH6eglkrh/hXqu1rweN1gqZ8mRzyqBPu3GOd/A
   PhmcGcwTTYJBtYze4D1gCCAPRX5ron+jjBXu
   -----END CERTIFICATE-----
   ```

4. Separate the chained or root certificate into individual certificate files using the BEGIN CERTIFICATE and END CERTIFICATE designations as the boundaries of each certificate. Save each with a distinguishing file name in root's home directory.

   For example, split the chained certificate shown in the previous step into two files, tomcat_chain1 and tomcat_chain2, and save each file in root's home directory.

5. Import the certificate by typing the following command on a single command line:

   ```
   $JAVA_HOME/bin/keytool -importcert -trustcacerts -noprompt -file
   ~/chained_certN -alias chained_certN
   ```

   wherechained_certN is a certificate file saved from the chained or root certificate that was received from the CA and N represents an integer identifier indicating that more than one certificate file was saved from the chained or root certificate.

6. At the password prompt, type the keystore password and press **Enter**:

   ```
   Enter keystore password: PASSWORD
   ```

   where PASSWORD is the keystore password. The default is "changeit".

   A message appears:

   ```
   Certificate was added to keystore
   ```

7. Repeat step 5 and step 6 for each individual certificate file derived from the chained or root certificate file.

Importing the public key certificate

To import the public key certificate that was saved as tomcat.cert in "Obtaining a public key certificate" on page 70:

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Import the certificate by typing the following command on a single command line:

   ```
   $JAVA_HOME/bin/keytool -importcert -trustcacerts -noprompt -file
   ~/tomcat.cert -alias tomcat
   ```

4. At the password prompt, type the keystore password and press **Enter**:

   ```
   Enter keystore password: PASSWORD
   ```

   where PASSWORD is the keystore password. The default is "changeit".

   A message appears:

   ```
   Certificate reply was installed in keystore
   ```

The trusted public key certificate is incorporated into the private key entry shared by the Avamar services that utilize RMI and Avamar Enterprise Manager's Tomcat server. It can be referenced using the "tomcat" alias.

Restarting services

Restart Avamar Enterprise Manager to make the public key certificate available for browser requests.

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Use dpnctl to restart the Avamar Enterprise Manager processes:

   ```
   dpnctl stop ems
   dpnctl start ems
   ```

Changing the root keystore password

The default password of the root keystore is “changeit” and is commonly known. To secure the keystore and preserve the integrity of its keys the keystore password should be changed.

The password for a Tomcat server’s private key entry must be identical to the keystore password. After changing the root keystore password, the password for the Tomcat server private key entry, created in “Installing a trusted public key certificate” on page 66, should be changed to the same password.

1. Log in to the Avamar server (single-node) or utility node (multi-node) using an account that is authorized to su to root.
2. Open a root shell by typing:

   ```
   su -
   ```

3. Change the root keystore password by typing the following command on a single command line:

   ```
   $JAVA_HOME/bin/keytool -storepasswd
   ```

4. When prompted, type the old password and then the new password twice.
5. Change the Tomcat server private key entry’s password by typing the following command on a single command line:

   ```
   $JAVA_HOME/bin/keytool -keypasswd -alias tomcat
   ```

6. When prompted, type the old password and then the new password twice.

   The new key entry password must be identical to the keystore password.

7. Delete the mcssl certificate from the keystore by typing:

   ```
   $JAVA_HOME/bin/keytool -delete -alias mcssl
   ```

8. Export a backup mcssl certificate from the rmi_ssl_keystore by typing the following on a single command line:

   ```
   $JAVA_HOME/bin/keytool -export
    -keystore /usr/local/avamar/lib/rmi_ssl_keystore
    -alias mcssl -file /tmp/mcssl.crt
   ```

   The default password for rmi_ssl_keystore is changeme. Use this password if it has not been changed.

9. Import the mcssl certificate file to the root keystore by typing the following on a single command line:

   ```
   $JAVA_HOME/bin/keytool -import -alias mcssl -file /tmp/mcssl.crt
   ```

10. Set the Tomcat server for Avamar Enterprise Manager to use the new password by editing its associated server.xml and emserver.xml files.

    a. Open server.xml in a plain-text editor such as vi:

       ```
       vi /usr/local/avamar-tomcat/conf/server.xml
       ```

    b. In server.xml, find the Connector element for port=8543 and change the keystorePass attribute in that element to the new password:

       ```
       keystorePass=newpassword
       ```

       where newpassword is the same as the root keystore password and the Tomcat server key entry password.

    c. Save and close server.xml.

    d. Open emserver.xml in a plain-text editor such as vi:

       ```
       vi /usr/local/avamar/var/em/server_data/prefs/emserver.xml
       ```

    e. Find the entry for the "trust_keystore_ap" key and edit it to replace the old password with the new password:

       ```
       <entry key="trust_keystore_ap" value="newpassword" />
       ```

       where newpassword is the same as the root keystore password and the Tomcat server key entry password.

    f. Save and close emserver.xml.

11. Use dpnctl to restart the Avamar Enterprise Manager processes:

    ```
    dpnctl stop ems
    dpnctl start ems
    ```

SSH authentication with Data Domain

If you store Avamar client backups on a Data Domain system, the Avamar Management Console Server (MCS) issues commands to a Data Domain system by using Secure Shell (SSH) commands. The commands retrieve information about the system, including serial number, disk capacity, CPU utilization, and so on.

The Data Domain system includes an SSH interface named DDSSH that allows commands to be issued remotely. DDSSH requires login credentials to establish a secure connection.

You can avoid the caching of a username and password for DDSSH by creating public/private keys on the Avamar server and exchanging the keys between the Data Domain system and the Avamar server for use by the MCS.

To generate an SSH public/private key pair and send the public key to the Data Domain system:

1. Open a command shell and log in to the utility node of the Avamar server as admin.
2. Change to the .ssh directory by typing:

   cd ~/.ssh

3. Generate a public/private key pair by typing:

   ssh-keygen -t rsa -N "" -f DDR_KEY

   where DDR_KEY is the file name for the key. There is no passphrase for the key.

4. Log in to the Data Domain system by typing:

   ssh AVAMAR_USER@DD_SYSTEM

   where:

   - AVAMAR_USER is the name of the Avamar user on the Data Domain system.
   - DD_SYSTEM is the name of the Data Domain system.

5. Add the SSH public key to the SSH authorized keys file on the Data Domain system by typing:

   sysadmin@DD_SYSTEM# adminaccess add ssh-keys user AVAMAR_USER

   where:

   - DD_SYSTEM is the name of the Data Domain system.
   - AVAMAR_USER is the name of the Avamar user on the Data Domain system.

6. Copy and paste the public key, which is the contents of the file ddr_key.pub, in /home/admin/.ssh:

   a. Open a second command shell and log in to the utility node of the Avamar server as admin.

   b. Change to the .ssh directory by typing:

      cd ~/.ssh

c. Display the ddr_key.pub file by typing:

```
cat ddr_key.pub
```

d. Select and copy the contents of the file.

e. Return to the first command shell window.

f. Paste the contents of the file in /home/admin/.ssh.

7. Enter the key by pressing Ctrl+D.
8. Log in to the Avamar server as root.
9. Change directory to /usr/local/avamar/lib by typing:

```
cd /usr/local/avamar/lib/
```

10. Copy the private key to /home/admin/.ssh/ddr_key, which is the path and name specified by ddr_ssh_key_path_name in the mcserver.xml file, by typing:

```
cp /home/admin/.ssh/DDR_KEY .
```

where DDR_KEY is the file name for the key.

11. Change the ownership of the key to the admin group by typing:

```
chown root:admin DDR_KEY
```

where DDR_KEY is the file name for the key.

12. Change the perfmissions for the key to 440 by typing:

```
chmod 440 DDR_KEY
```

where DDR_KEY is the file name for the key.

13. Test that you can log in to the Data Domain system without providing a password by typing:

```
ssh -i PATH/DDR_KEY AVAMAR_USER@DD_SYSTEM
```

where:

- PATH/DDR_KEY is the path and file name of the key.
- AVAMAR_USER is the name of the Avamar user on the Data Domain system.
- DD_SYSTEM is the name of the Data Domain system.

CHAPTER 4
Data Security and Integrity

The following topics provide details on the options to provide security and ensure the integrity of data in the Avamar system:

Encrypting data

Avamar can encrypt all data sent between clients and the server "in flight." Each individual Avamar server can also be configured to encrypt data stored on the server "at rest."

"In-Flight" encryption

To provide enhanced security during client/server data transfers, Avamar supports two levels of "in-flight" encryption: Medium and High. The exact encryption technology and bit strength used for any given client-server connection depends on a number of factors, including the client platform and Avamar server version. "Client/server encryption behavior" on page 79 provides details.

You specify the default encryption method to use for client/server data transfers (None, Medium, or High) when you create and edit groups. You also can override the group encryption method for a specific client on the Client Properties tab of the Edit Client dialog box, for a specific backup on the On Demand Backup Options dialog box, or for a specific restore on the Restore Options dialog box. The *EMC Avamar Administration Guide* provides details.

To enable encryption of data in transit, the Avamar server data nodes each require a unique public/private key pair and a signed X.509 certificate that is associated with the public key.

When the Avamar server is installed, a public/private key pair and a self-signed certificate are generated automatically in the /data01/home/admin directory on each Avamar server storage node and in the /usr/local/avamar/etc directory on the utility node. However, because self-signing is not recommended in production environments, you should generate and install a key and signed certificate from either a commercial or private CA. "Client/Server Access and Authentication" on page 41 provides instructions on how to do this, as well as how to configure both Windows and UNIX clients to validate the certificates from the Avamar server.

Note: You also can configure Avamar for two-way authentication, where the client requests authentication from the Avamar server, and then the Avamar server also requests authentication from the client. One-way, or server-to-client, authentication typically provides sufficient security. However, in some cases, two-way authentication is required or preferred.

The following steps detail the encryption and authentication process for client/server data transfers in a server-to-client authentication environment:

1. The Avamar client requests authentication from the Avamar server.
2. The server sends the appropriate certificate to the client. The certificate contains the public key.
3. The client verifies the server certificate and generates a random key, which is encrypted using the public key, and sends the encrypted message to the server.

4. The server decrypts the message by using its private key and reads the key generated by the client.
5. This random key is then used by both sides to negotiate on a set of temporary symmetric keys to perform the encryption. The set of temporary encryption keys is refreshed at a regular interval during the backup session.

Note: If you store Avamar client backups on a Data Domain system, the connection between the Avamar client and the Data Domain system is not encrypted. The Data Domain Distributed Deduplication Bandwidth Optimized OST (DDBOOST) SDK, which Avamar uses to access the Data Domain system, does not support data encryption between the client and the Data Domain system.

"At-Rest" encryption

In addition to encrypting client/server data transfers, each server can be configured to encrypt data stored residing on it. This is called "at-rest" encryption.

When encryption is enabled, the server accepts a user-defined salt that is then used to generate an encryption key. The salt is stored on the Avamar server for subsequent encryption/decryption activities.

Key management is completely automatic:

- Old encryption keys are automatically stored in a secure manner so that data stripes encrypted with previous keys can always be decrypted and read.
- During server maintenance, crunched stripes will over time be converted to use the most current key.

Note that since any reads/writes from disk require encryption processing with this feature enabled, there is a performance impact to the Avamar server of approximately 33 percent.

Beginning with version 6.1, encryption is performed using AES 128 CFB. Older systems can continue to use 128-bit Blowfish until the salt is changed.

Client/server encryption behavior

Client-server encryption functional behavior in any given circumstance is dependent on a number of factors, including Avamar server version, client version, the mcserver.xml encrypt_server_authenticate value, and the **avtar --encrypt** option used during that activity.

Note: You set the encrypt_server_authenticate value to true when you configure server-to-client authentication, as discussed in "Client/Server Access and Authentication" on page 41.

In Avamar 4.1 and later, you specify an option flag pair: encrypt and encrypt-strength. The encrypt-strength option takes one of three values: None, Medium, or High.

In Avamar releases before 4.1, you could request 256-bit or 128-bit encryption strength and SHA digests by using option flags, including the examples in the following list:

- ssl:AES256-SHA
- ssl:AES128-SHA
- sslverify:AES256-SHA
- sslverify:AES128-SHA

Note: Avamar supports other types of encryption in addition to the ones listed.

Avamar 4.1 and later deprecates this notation for option flags. Deprecated versions of option flags that still exist for clients running Avamar 4.1 or later are ignored. A pre-4.1 option flag, such as ssl:AES256-SHA, translates into an encrypt and encrypt-strength option flag pair for Avamar 4.1 and later.

For example, if server authentication is *not* requested, then the option flag pair for ssl:AES256-SHA is specified as follows:

```
--encrypt=tls
--encrypt-strength=high
```

If server authentication *is* requested, then the option flag pair for ssl:AES256-SHA is specified as follows:

```
--encrypt=tls-sa
--encrypt-strength=high
```

The following table documents various encryption behaviors and strengths.

Table 14 Client/server encryption behaviors and strengths

Encryption setting	mcserver.xml encrypt_server_authenticate setting	avtar setting	Behavior/description
None	FALSE	--encrypt=proprietary --encrypt-strength=cleartext	Unencrypted “clear” text.
	TRUE	Not supported.	Error Event - job failed due to options incompatibility.
Medium	FALSE	--encrypt=tls --encrypt-strength=medium	AES-128 encryption.
	TRUE	--encrypt=tls-sa --encrypt-strength=medium	AES-128 encryption.
High	Either TRUE or FALSE	--encrypt=tls --encrypt-strength=high	AES-256 encryption.

Increasing cipher strength used by Avamar servers

By default, the Management Console and Enterprise Manager servers support cipher strengths up to 128-bit. You can increase the cipher strength used by these servers to 256-bit for communications on the following ports:

- Ports 7778 and 7779 for the Management Console.
- Ports 8778 and 8779 for the Enterprise Manager.
- Port 9443 for the Management Console Web Services.

Increasing cipher strength for the Management Console

To increase the cipher strength used by the Management Console, do the following:

1. Set the rmi_cipher_strength parameter to high in the /usr/local/avamar/var/mc/server_data/prefs/mcserver.xml file:

   ```
   rmi_cipher_strength=high
   ```

2. Download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6:

   a. In a web browser, navigate to http://java.sun.com.

   b. Search for "Java Cryptography Extension."

   c. Download the file associated with Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6 (jce_policy-6.zip).

   d. Unzip the jce_policy-6.zip file in a temporary folder and follow the instructions in the README.txt file to install.

3. Restart the Management Console server by typing:

   ```
   dpnctl stop mcs
   dpnctl start
   ```

Increasing cipher strength for the Enterprise Manager

To increase the cipher strength used by the Enterprise Manager, do the following:

1. Set the rmi_cipher_strength parameter to high in the /usr/local/avamar/var/mc/server_data/prefs/emserver.xml file:

   ```
   rmi_cipher_strength=high
   ```

2. Download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6:

   a. In a web browser, navigate to http://java.sun.com.

   b. Search for "Java Cryptography Extension."

   c. Download the file associated with Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6 (jce_policy-6.zip).

   d. Unzip the jce_policy-6.zip file in a temporary folder and follow the instructions in the README.txt file to install.

3. Restart the Management Console server by typing:

```
dpnctl stop ems
dpnctl start ems
```

Data integrity

Checkpoints are system-wide backups taken for the express purpose of assisting with disaster recovery. Checkpoints are typically scheduled twice daily and validated once daily (during the maintenance window). You also can create and validate additional server checkpoints on an on-demand basis. The *EMC Avamar Administration Guide* provides details on creating, validating, and and deleting server checkpoints.

Checkpoint validation, which is also called an Avamar Hash Filesystem check (HFS check), is an internal operation that validates the integrity of a specific checkpoint. Once a checkpoint has passed an HFS check, it can be considered reliable enough to be used for a system rollback.

The actual process that performs HFS checks is **hfscheck**; it is similar to the UNIX **fsck** command.

You can schedule HFS checks by using Avamar Administrator. You also can manually initiate an HFS check by running **avmaint hfscheck** directly from a command shell.

An HFS check might take several hours depending on the amount of data on the Avamar server. For this reason, each validation operation can be individually configured to perform all checks (full validation) or perform a partial "rolling" check which fully validates all new and modified stripes, then partially checks a subset of unmodified stripes.

Initiating an HFS check requires significant amounts of system resources. To reduce contention with normal server operation, an HFS check can be throttled.

Additionally, during this time, the server is placed in read-only mode. Once the check has been initiated, normal server access is resumed. You can also optionally suspend command dispatches during this time, although this is not typically done.

If HFS check detects errors in one or more stripes, it automatically attempts to repair them.

Data erasure

When you manually delete a backup using Avamar Administrator or you automatically delete a backup when its retention policy expires and garbage collection runs, data is marked as deleted but is left on disk.

You can permanently and securely delete backups from an Avamar server in a manner that satisfies stringent security requirements by overwriting the data that is unique to a backup with random data. The following topics provide details on securely deleting backups from an Avamar server:

- "Requirements to securely delete backups" on page 83
- "How to securely delete backups" on page 84

Requirements to securely delete backups

Consider the following requirements for secure deletion of backups:

- You must be familiar with basic- to intermediate-level Avamar server terminology and command-line administration.
- Some steps to securely delete backups might require the use of third party tools such as the open-source srm or GNU shred utilities. The documentation for those utilities provides additional information regarding proper use, capabilities, and limitations of those utilities.
- Use of any non-certified storage hardware, including RAID controllers and disk storage arrays, might impact the effectiveness of the secure backup deletion. Consult the manufacturers of those devices for information about disabling or clearing write caches, or about any other features that impact data transfer to the storage media.
- The following conditions must be met in the Avamar environment:
  - All nodes must be in the ONLINE state, and no stripes should be in the OFFLINE state. This can be checked using the **status.dpn** command.
  - The most recent checkpoint must have been successfully validated.
  - Pending garbage collection operations can increase the time needed to complete the secure deletion process, or can cause extra data to be overwritten. Therefore, you should run garbage collection until all pending non-secure deletions have successfully completed. No errors should be reported by the garbage collection process.
  - The server should be idle:
    - There should be no backups in progress, nor should the server be running garbage collection or HFS checks.
    - The backup scheduler and maintenance windows scheduler should be stopped for the duration of the secure deletion process, so that no new backups or maintenance activities are initiated.
  - Avamar storage node ext3 file systems should not be configured to operate in data=journal mode. If this is the case, data might persist on the disk after the secure deletion process has completed.

How to securely delete backups

The **securedelete** program enables you to securely erase selected backups on the Avamar server.

Note: This procedure can be used in conjunction with the existing procedures at a company to securely delete data from other parts of the operating system or hardware. Contact EMC Technical Support for any questions regarding the effect of company procedures on the Avamar server software.

To securely delete backups from an Avamar server with the **securedelete** program:

1. Open a command shell and log in:
   - If logging into a single-node server, log in to the server as admin.
   - If logging into a multi-node server, log in to the utility node as admin, then load the admin OpenSSH key by typing:

     ```
     ssh-agent bash
     ssh-add ~admin/.ssh/admin_key
     ```

     When prompted, type the admin_key passphrase and press **Enter**.
2. Locate the backups to securely delete by typing the following on a single command line:

   ```
   securedelete getb --id=USER@AUTH --password=PASSWORD
    --account=DOMAIN/CLIENT
   ```

   where:
   - USER is the Avamar user name.
   - AUTH is the authentication system used by that user (the default internal authentication domain is “avamar”).
   - PASSWORD is the password for the --id=USER@AUTH account.
   - DOMAIN/CLIENT is the full location of the client machine.
3. Locate the backup to delete in the list, and note the date in the created field.
4. Securely delete the backup by typing the following on a single command line:

   ```
   securedelete delb --id=USER@AUTH --password=PASSWORD --secure
    --date=DATE
   ```

   where:
   - USER is the Avamar user name.
   - AUTH is the authentication system used by that user (the default internal authentication domain is “avamar”).
   - PASSWORD is the password for the --id=USER@AUTH account.
   - DATE is the backup date noted in step 3.

   This operation typically takes several minutes to complete while the server securely overwrites data.

Note: Do not interrupt the securedelete delb command. If interrupted, all data will not be securely deleted.

If successful, the **securedelete delb** command returns the following response:

```
1 Request succeeded
```

If unsuccessful, the **securedelete delb** command returns the following response:

```
0 ERROR! Exit code 0: Request failed.
```

5. If an error is encountered:
   - Search the knowledge base at the EMC Online Support website, https://support.emc.com, for the specific error code.
   - If the required information is not found, engage EMC Support using Live Chat or create a Service Request as described in "Where to get help" on page 11.
6. Repeat steps 2–5 for all other backups that are to be securely deleted.
7. Check the server logs for any ERROR or WARN messages that might indicate a failure of the secure deletion operation by typing:

   mapall --noerror 'grep "ERROR\|WARN" /data01/cur/gsan.log*'
8. If any such messages are present:
   - Search the knowledge base at the EMC Online Support website, https://support.emc.com, for the specific error code.
   - If the required information is not found, engage EMC Support using Live Chat or create a Service Request as described in "Where to get help" on page 11.

   If any stripes on the system have been repaired or rebuilt due to data corruption, then the bad versions remain on disk. Overwrite or securely delete these files by using an appropriate third-party tool.
9. Locate these stripes by typing:

   mapall --noerror 'ls /data??/cur/*.bad*'

   Information similar to the following appears in the command shell:

   ```
   /data06/cur/0000000300000016.0000000300000016.bad1240015157
   /data06/cur/0000000300000016.cdt.bad1240015157
   /data06/cur/0000000300000016.chd.bad1240015157
   /data06/cur/0000000300000016.wlg.bad1240015157
   ```
10. If backups were performed before the most recent checkpoint was taken, roll the server back to the most recent checkpoint, and repeat steps 2–9.
11. Repeat step 10 for all applicable checkpoints.
12. Repeat this entire procedure on all other Avamar servers to which this Avamar server replicates backups.

CHAPTER 5
System Monitoring, Auditing, and Logging

The following topics discuss the features available to monitor the Avamar environment and audit the operations performed. It also provides a list of log files that are available for each feature on each component in the system:

Client activity monitoring

You can monitor client backup, restore, and validation activity to verify backups are successfully completing and that no abnormal activity is occurring.

The Activity Monitor tab on the Activity window in Avamar Administrator provides details on client activity, including the type, status, start and end time, error code (if applicable), and other details for each client activity.

The *EMC Avamar Administration Guide* provides details on how to access the Activity Monitor tab and filter the activities that appear in the tab.

Server monitoring

There are several features available to assist you in monitoring the Avamar environment, including server status and system events.

Monitoring server status

You can monitor the status of the following items on the Avamar server:

- Overall Avamar server status
- Capacity usage
- Modules
- Nodes
- Partitions
- Checkpoints
- Garbage collection
- Maintenance activities

If you use a Data Domain system as storage for Avamar client backkups, you also can monitor CPU, disk activity, and network activity for each node on the Data Domain system.

This status information is provided on the tabs in the Avamar Server window in Avamar Administrator. The *EMC Avamar Administration Guide* provides details on how to access the Avamar Server window and the information available on each tab.

Monitoring system events

All Avamar system activity and operational status is reported as various events to the MCS. Examples of various Avamar events include client registration and activation, successful and failed backups, hard disk status, and others.

Events are listed in the Event Management tab in the Administration window of Avamar Administrator. The *EMC Avamar Administration Guide* provides details on how to access the Event Management tab and filter the events that appear in the tab.

Event notification mechanisms

You can also configure Avamar to notify you when events occur. There are several features and functions available.

Pop-up alerts

Events can be configured on an event-by-event basis to generate a graphical pop-up alert each time one of those events occurs. One significant limitation of this feature is that Avamar Administrator software must be running in order for the pop-up alerts to be displayed.

Acknowledgement required list

Events can be configured on an event-by-event basis such that when events of this type occur, an entry is added to a list of events that requires interactive acknowledgement by the Avamar system administrator.

Email messages

Events can be configured on an event-by-event basis to send an email message to a designated list of recipients. Email notifications can be sent immediately or in batches at regularly-scheduled times.

Syslog support

Events can be configured on an event-by-event basis to log information to local or remote syslog files based on filtering rules configured for the syslog daemon receiving the events. Third-party monitoring tools and utilities capable of examining log entries can access the syslog files and process them to integrate Avamar event information into larger site activity and status reports.

SNMP support

Simple Network Management Protocol (SNMP) is a protocol for communicating monitoring and event notification information between an application, hardware device or software application, and any number of monitoring applications or devices. The Avamar SNMP implementation provides two distinct ways to access Avamar server events and activity completion status:

- SNMP requests provide a mechanism for SNMP management applications to “pull” information from a remote SNMP-enabled client (in this case, the Avamar server).
- SNMP traps provide a mechanism for the Avamar server to “push” information to SNMP management applications whenever designated Avamar events occur. Events can be configured on an event-by-event basis to output SNMP traps.

Note: Avamar also can collect and display data for health monitoring, system alerts, and capacity reporting on a configured Data Domain system by using SNMP. The *EMC Avamar and Data Domain Integration Guide* provides details on how to configure SNMP for Avamar with Data Domain.

ConnectEMC support

Events can be configured on an event-by-event basis to send a notification message directly to EMC Technical Support using ConnectEMC.

The *EMC Avamar Administration Guide* provides details on how to configure each of these notification mechanisms.

Event notification profiles

Profiles are a notification management feature that are used to logically group certain event codes together and specify which notifications should be generated when these events occur. You can create custom profiles to organize system events and generate the desired notifications when any of those events occur. The *EMC Avamar Administration Guide* provides details on how to create and manage profiles.

Email home notification

When fully configured and enabled, the "email home" feature automatically emails the following information to EMC Customer Service twice daily:

- Status of the daily data integrity check
- Selected Avamar server warnings and information messages
- Any Avamar server errors
- Any RAID errors (single-node servers only)

By default, these email messages are sent at 6 a.m. and 3 p.m. each day (based on the local time on the Avamar server). The timing of these messages is controlled by the Notification Schedule.

The *EMC Avamar Administration Guide* provides details on how to enable and schedule the email home feature.

Auditing

The Avamar Audit Log provides details on the operations initiated by users in the Avamar system. The data in this log allows enterprises that deploy Avamar to enforce security policies, detect security breaches or deviation from policies, and hold appropriate users accountable for those actions. The audit log includes the following information for each operation:

- The date and time the action occurred
- The event code number associated with the action
- The ID and role of the user that initiated the action
- The product and component from which the action was initiated
- The severity of the action
- The domain in which the action occurred

The Audit Log is available in Avamar Administrator as a subtab of the Event Management tab in the Administration window. The *EMC Avamar Administration Guide* provides details on how to access the Audit Log and filter the events that appear in the log.

Gen4 and later Avamar Data Stores running the SUSE Linux Enterprise Server (SLES) operating system implement improved auditing features, such as Advanced Intrusion Detection Environment (AIDE) and the **auditd** service. "Advanced Intrusion Detection Environment (AIDE)" on page 98 and "Auditing service (auditd)" on page 99 for detailed information about those features.

Logs

Avamar software includes log files for server and client components, maintenance tasks, various utilities, and backup clients. These log files enable you to examine various aspects of the Avamar system.

The following sections includes log file information organized in tables for each Avamar component. For additional information on log files, refer to the Avamar guide for the specific component.

Single-node server

The following table lists single-node server log files.

Table 15 Single-node server log files (page 1 of 3)

Feature/function	Log file locations
Avamar Administrator server	/usr/local/avamar/var/mc/server_log/flush.log
	/usr/local/avamar/var/mc/server_log/restore.log
	/usr/local/avamar/var/mc/server_log/mcserver.log.#
	/usr/local/avamar/var/mc/server_log/mcserver.out
	/usr/local/avamar/var/mc/server_log/pgsql.log
	/usr/local/avamar/var/mc/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log
	/usr/local/avamar/var/mc/server_data/mcs_data_dump.sql
Avamar Enterprise Manager - Tomcat	/usr/local/avamar/var/em/webapp_log/admin.DATE.log
	/usr/local/avamar/var/em/webapp_log/catalina.DATE.log
	/usr/local/avamar/var/em/webapp_log/catalina.out
	/usr/local/avamar/var/em/webapp_log/host-manager.DATE.log
	/usr/local/avamar/var/em/webapp_log/localhost.DATE.log
	/usr/local/avamar/var/em/webapp_log/manager.DATE.log

Table 15 Single-node server log files (page 2 of 3)

Feature/function	Log file locations
Avamar Enterprise Manager - Server	/usr/local/avamar/var/em/server_log/flush.log
	/usr/local/avamar/var/em/server_log/restore.log
	/usr/local/avamar/var/em/server_log/emserver.log.#
	/usr/local/avamar/var/em/server_log/emserver.out
	/usr/local/avamar/var/em/server_log/pgsql.log
	/usr/local/avamar/var/em/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log
	/usr/local/avamar/var/em/server_data/ems_data_dump.sql
Maintenance tasks	/usr/local/avamar/var/cron/clean_emdb.log
	/usr/local/avamar/var/cron/dpn_crontab.log
	/usr/local/avamar/var/cron/cp.log
	/usr/local/avamar/var/cron/gc.log
	/usr/local/avamar/var/cron/hfscheck.log
	/usr/local/avamar/var/cron/ntpd_keepalive_cron.log
	/usr/local/avamar/var/cron/ntpd_keepalive_cron.log.#
	/usr/local/avamar/var/cron/suspend.log
avw_install utility	/usr/local/avamar/var/avw_cleanup.log
	/usr/local/avamar/var/avw_install.log
	/usr/local/avamar/var/avw-time.log
	/usr/local/avamar/var/log/dpnavwinstall-VERSION.log
axion_install utility	/usr/local/avamar/var/axion_install_DATE_TIME.log
Avamar File System (AvFS)	/usr/local/avamar/var/axionfs.log
change-passwords utility	/usr/local/avamar/var/change-passwords.log
ddrmaint utility	/usr/local/avamar/var/log/ddrmaint.log
dpnctl utility	/usr/local/avamar/var/log/dpnctl.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutil-version.log
	/usr/local/avamar/var/log/dpnnetutil.log*
	/usr/local/avamar/var/log/dpnnetutilbgaux.log
	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
permctl utility	/usr/local/avamar/var/log/permctl.log
resite utility	/usr/local/avamar/var/dpnresite-version.log
	/usr/local/avamar/var/mcspref.log
	/usr/local/avamar/var/nataddr.log
	/usr/local/avamar/var/smtphost.log

Table 15 Single-node server log files (page 3 of 3)

Feature/function	Log file locations
timedist utility	/usr/local/avamar/var/timedist.log
timesyncmon program	/usr/local/avamar/var/timesysncmon.log
Avamar Replicator	/usr/local/avamar/var/cron/replicate.log
Avamar license server	/usr/local/avamar/var/ascd-PORT.log
Storage server log	/data01/cur/err.log
	/data01/cur/gsan.log

Utility node

The following table lists utility node log files.

Table 16 Utility node log files (page 1 of 2)

Feature/function	Log file locations
Avamar Administrator server	/usr/local/avamar/var/mc/server_log/flush.log
	/usr/local/avamar/var/mc/server_log/restore.log
	/usr/local/avamar/var/mc/server_log/mcddrssh.log
	/usr/local/avamar/var/mc/server_log/mcddrsnmp.out
	/usr/local/avamar/var/mc/server_log/mcddrsnmp.log
	/usr/local/avamar/var/mc/server_log/mcserver.log.#
	/usr/local/avamar/var/mc/server_log/mcserver.out
	/usr/local/avamar/var/mc/server_log/pgsql.log
	/usr/local/avamar/var/mc/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log
	/usr/local/avamar/var/mc/server_data/mcs_data_dump.sql
Avamar Enterprise Manager - Tomcat	/usr/local/avamar/var/em/webapp_log/admin.DATE.log
	/usr/local/avamar/var/em/webapp_log/catalina.DATE.log
	/usr/local/avamar/var/em/webapp_log/catalina.out
	/usr/local/avamar/var/em/webapp_log/host-manager.DATE.log
	/usr/local/avamar/var/em/webapp_log/localhost.DATE.log
	/usr/local/avamar/var/em/webapp_log/manager.DATE.log

Table 16 Utility node log files (page 2 of 2)

Feature/function	Log file locations
Avamar Enterprise Manager - Server	/usr/local/avamar/var/em/server_log/flush.log
	/usr/local/avamar/var/em/server_log/restore.log
	/usr/local/avamar/var/em/server_log/emserver.log.#
	/usr/local/avamar/var/em/server_log/emserver.out
	/usr/local/avamar/var/em/server_log/pgsql.log
	/usr/local/avamar/var/em/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log
	/usr/local/avamar/var/em/server_data/ems_data_dump.sql
Maintenance tasks	/usr/local/avamar/var/cron/clean_emdb.log
	/usr/local/avamar/var/cron/dpn_crontab.log
	/usr/local/avamar/var/cron/cp.log
	/usr/local/avamar/var/cron/gc.log
	/usr/local/avamar/var/cron/hfscheck.log
	/usr/local/avamar/var/cron/ntpd_keepalive_cron.log
	/usr/local/avamar/var/cron/ntpd_keepalive_cron.log.#
	/usr/local/avamar/var/cron/suspend.log
avw_install utility	/usr/local/avamar/var/avw_cleanup.log
	/usr/local/avamar/var/avw_install.log
	/usr/local/avamar/var/avw-time.log
	/usr/local/avamar/var/log/dpnavwinstall-VERSION.log
axion_install utility	/usr/local/avamar/var/axion_install_DATE_TIME.log
Avamar File System (AvFS)	/usr/local/avamar/var/axionfs.log
change-passwords utility	/usr/local/avamar/var/change-passwords.log
ddrmaint utility	/usr/local/avamar/var/log/ddrmaint.log
dpnctl utility	/usr/local/avamar/var/log/dpnctl.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutil-version.log
	/usr/local/avamar/var/log/dpnnetutil.log*
	/usr/local/avamar/var/log/dpnnetutilbgaux.log
	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
permctl utility	/usr/local/avamar/var/log/permctl.log
timedist utility	/usr/local/avamar/var/timedist.log
timesyncmon program	/usr/local/avamar/var/timesysncmon.log
Avamar Replicator	/usr/local/avamar/var/cron/replicate.log
Avamar license server	/usr/local/avamar/var/ascd-PORT.log

Storage node

The following table lists storage node log files.

Table 17 Storage node log files

Feature/function	Log file locations
Storage server log	/data01/cur/err.log
	/data01/cur/gsan.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
	/usr/local/avamar/var/log/dpnnetutilbgaux.log
Maintenance tasks	/usr/local/avamar/var/ntpd_keepalive_cron.log*
timesyncmon program	/usr/local/avamar/var/timesyncmon.log*

Spare node

The following table lists spare node log files.

Table 18 Spare node log files

Feature/function	Log file locations
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
	/usr/local/avamar/var/log/dpnnetutilbgaux.log

Avamar NDMP Accelerator

The following table lists Avamar NDMP Accelerator log files.

Table 19 Avamar NDMP Accelerator log files

Feature/function	Log file locations
avndmp log	/usr/local/avamar/var/{FILER-NAME}/*.avndmp.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
	/usr/local/avamar/var/log/dpnnetutilbgaux.log

Access node

The following table lists access node log files.

Table 20 Access node log files

Feature/function	Log file locations
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
	/usr/local/avamar/var/log/dpnnetutilbgaux.log

Avamar Administrator client network host

The following table lists Avamar Administrator client network host log files.

Table 21 Avamar Administrator client network host log files

Feature/function	Operating system	Log file locations
Avamar Administrator management console	Windows 7	C:\Users\USERNAME\.avamardata\var\mc\gui_log
	Windows Vista Windows XP	C:\Documents and Settings\USERNAME\.avamardata\var\mc\gui_log
	Linux	$HOME/.avamardata/var/mc/gui_log/mcclient.log.0
Avamar Administrator management console command line interface	UNIX	$HOME/.avamardata/var/mc/gui_log/mccli.log.0

Backup client network host

The following table lists backup client network host log files.

Table 22 Backup client network host log files

Feature/function	Log file locations
Client **avagent** process (all clients)	C:\Program Files\avs\var\avagent.log
Client **avtar** process (all clients)	C:\Program Files\avs\var\{WORKORDER-ID}.alg
	C:\Program Files\avs\var\{WORKORDER-ID}.log
Avamar Client for Windows tray applet	C:\Program Files\avs\var\avscc.log
Avamar Plug-in for DB2	/usr/local/avamar/var/{WORKORDER-ID}.log
Avamar Exchange Client	/usr/local/avamar/var/{WORKORDER-ID}.log
Avamar NDMP Accelerator	/usr/local/avamar/var/{WORKORDER-ID}.log
Avamar Client for NetWare	/usr/local/avamar/var/{WORKORDER-ID}.log
Avamar Plug-in for Oracle	/usr/local/avamar/var/{WORKORDER-ID}.log
Avamar Plug-in for SQL Server	/usr/local/avamar/var/{WORKORDER-ID}.log

CHAPTER 6
Server Hardening

The following topics describe various server security hardening features, which are available for Avamar 6.0 and later servers running the SUSE Linux Enterprise Server (SLES) operating system:

Overview

STIG compliance

Beginning with version 6.0, Avamar servers running the SLES operating system offer a number of improved security features, which are primarily targeted for customers needing to comply with US Department of Defense (DoD) *Security Technical Implementation Guide (STIG) for Unix* requirements.

Tiered implementation

Server hardening features are implemented using a tiered two-level architecture:

Level-1

Level-1 security hardening features are part of the base SLES operating system on Gen4 and later Avamar Data Stores. Level-1 features are:

- "Advanced Intrusion Detection Environment (AIDE)" on page 98
- "Auditing service (auditd)" on page 99
- "sudo implementation" on page 99
- "Command logging" on page 101

Level-2

Level-2 security hardening features are additional features, which can be installed during Avamar server software installation:

- "Additional operating system hardening" on page 101
- "Additional password hardening" on page 102
- "Additional firewall hardening (avfirewall)" on page 104

Level-2 security hardening features are only available on Avamar 6.0 and later servers running supported versions of the SLES operating system.

Installing level-2 security hardening features

Level-2 security hardening features can be installed during Avamar server software installation. For information about installing and enabling security hardening features refer to the *Avamar SLES Installation Workflow Guide*. During installation this PDF guide is available by clicking the help icon in Avamar Installation Manager.

Advanced Intrusion Detection Environment (AIDE)

The Advanced Intrusion Detection Environment (AIDE) is a SLES feature that is used to take a snapshot of an Avamar server configuration for purposes of establishing a reliable system baseline reference.

AIDE is a level-1 hardening feature that is implemented as part of the base SLES operating system on Gen4 and later Avamar Data Stores.

AIDE satisfies the following STIG requirements:

- GEN000140 - Create and maintain system baseline
- GEN000220 - System baseline for system libraries and binaries checking
- GEN002260 - System baseline for device files checking
- GEN002380 - SUID files baseline
- GEN002400 - System baseline for SUID files checking
- GEN002440 - SGID files baseline
- GEN002460 - System baseline for SGID files checking

The system baseline snapshot is stored in /var/lib/aide/aide.db.

AIDE reports are run weekly as part of the /etc/cron/weekly cron job.

AIDE output is logged to /var/log/secure.

Auditing service (auditd)

The **auditd** service is a SLES feature that implements a CAPP-compliant (Controlled Access Protection Profiles) auditing feature, which continually monitors the server for any changes that could affect the server's ability to perform as intended.

The **auditd** service is a level-1 hardening feature that is implemented as part of the base SLES operating system on Gen4 and later Avamar Data Stores.

The **auditd** service feature satisfies the following STIG requirements:

- GEN002660 - Configure and implement auditing
- GEN002680 - Audit logs accessibility
- GEN002700 - Audit Logs Permissions
- GEN002720 - Audit Failed File and Program Access Attempts
- GEN002740 - Audit File and Program Deletion
- GEN002760 - Audit Administrative, Privileged, and Security Actions
- GEN002800 - Audit Login, Logout, and Session Initiation
- GEN002820 - Audit Discretionary Access Control Permission Modifications
- GEN002860 - Audit Logs Rotation

auditd output is logged to /var/log/audit/audit.log.

sudo implementation

The **sudo** command is an alternative to direct root login. On Gen4 and later Avamar Data Stores, the admin and dpn user accounts are automatically added to the sudoers file. This enables admin and dpn users to execute commands that would otherwise require operating system root permission.

Implementation of the **sudo** command for admin and dpn users is a level-1 hardening feature that is implemented as part of the base SLES operating system on Gen4 and later Avamar Data Stores.

Implementation of the **sudo** command for admin and dpn users satisfies the following STIG requirements:

- GEN000260 - Shared Account Documentation
- GEN000280 - Shared Account Direct Logon
- GEN001100 - Encrypting Root Access
- GEN001120 - Encrypting Root Access

Prefixing commands with "sudo"

Instead of switching user to root with the **su** command, admin and dpn users can directly issue commands normally requiring root permissions by prefixing each command with **sudo**. For example, the following command installs MyPackage.rpm:

```
sudo rpm -ivh MyPackage.rpm
```

If prompted for a password, type the password and press **Enter**.

You might be periodically prompted to retype your admin or dpn password when prefixing other commands with **sudo**. This is normal.

Spawning a sudo Bash subshell

If you need to execute several commands normally requiring root permissions, you can also spawn a persistent **sudo** Bash subshell by typing **sudo bash**.

Commands normally requiring root permissions can now be typed directly with no additional modifications to the command line syntax. For example:

```
sudo bash
rpm -ivh MyPackage1.rpm
rpm -ivh MyPackage2.rpm
rpm -ivh MyPackage3.rpm
exit
```

Command logging

Gen4 and later Avamar Data Stores log all Bash shell commands issued by any user.

Bash command logging is a level-1 hardening feature that is implemented as part of the base SLES operating system on Gen4 and later Avamar Data Stores.

Bash command logging does not satisfy any particular STIG requirements. It is intended to be used as a generalized debugging and forensics tool.

Additional operating system hardening

The additional Operating System (OS) hardening package provides the following capabilities for Avamar 6.0 and later servers running supported versions of SLES:

- Setting terminal timeout at 15 minutes
- Applying read-only permission to root home directory
- Removal of world read permissions on log files
- Removal of world read permissions on cron files
- Lockdown of some important /etc system configuration files
- Removal of world read permissions from admin, dpn and gsan home directories
- Removal of unnecessary default accounts and groups
- Disabling of ssh v1 protocol
- Removal of unnecessary tomcat directories
- Changing system and user umask settings to 077
- Removing unowned files
- Enabling cron logging in syslog

The additional OS hardening package is a level-2 hardening feature that can be installed during Avamar server software installation.

The additional OS hardening package satisfies the following STIG requirements:

- GEN000460 - Unsuccessful Login Attempts - Account Disabled
- GEN000480 - Unsuccessful Login Attempts - Fail Delay
- GEN000500 - Terminal Lockout
- GEN000980 - Root Console Access
- GEN001000 - Remote Consoles Defined
- GEN001020 - Direct Root Login
- GEN001120 - Encrypting Root Access
- GEN001160 - Unowned Files
- GEN001240 - System Files, Programs, and Directories Group Ownership
- GEN001260 - Log File Permissions

- GEN001480 - User Home Directory Permissions
- GEN001500 - Home Directory Permissions
- GEN001260 - Log File Permissions
- GEN001560 - Home Directories Files Permissions
- GEN002420 - User Filesystems Not Mounted With NoSUID
- GEN002580 - Permissive umask Documentation
- GEN003160 - Cron Logging
- GEN003180 - Cronlog Permissions

Additional password hardening

Avamar 6.0 and later servers running supported versions of SLES operating system can be configured to provide additional password hardening features such as:

- Aging — how long a password can be used before it must be changed
- Complexity — required number and type of characters in passwords
- Reuse — number of previously used passwords that can be recycled
- Lockout — denial of login after a specified number of unsuccessful login attempts
- Account lockout after 35 days of no logins

NOTICE

Password hardening is not appropriate for all customers. Successful implementation of this feature requires structures and policies that enforce changes to all operating system user accounts every 60 days, and require users to log into those accounts at least once every 35 days. Failure to implement proper structures and policies before installing the password hardening feature might cause you to be locked out of your Avamar server.

Additional password hardening is a level-2 hardening feature that can be installed during Avamar server software installation.

Additional password hardening satisfies the following STIG requirements:

- GEN000540 - Password Change 24 Hours
- GEN000560 - Password Protect Enabled Accounts
- GEN000580 - Password Length
- GEN000600 - Password Character Mix
- GEN000620 - Password Character Mix
- GEN000640 - Password Character Mix
- GEN000660 - Password Contents
- GEN000680 - Password Contents
- GEN000700 - Password Change Every 60 Days
- GEN000740 - Password Change Every Year

- GEN000760 - Inactive Accounts are not locked
- GEN000780 - Easily Guessed Passwords
- GEN000800 - Password Reuse
- GEN000820 - Global Password Configuration Files
- GEN000840 - Root Account Access

Hardened password rules

Following successful installation and configuration, the following rules are enforced for all local Avamar server operating system user accounts and passwords:

Account lockout

All local Avamar server operating system accounts must be logged into at least once every 35 days.

Furthermore, after 3 unsuccessful login attempts, that account will be administratively locked-out.

NOTICE

The SLES operating system allows expired root passwords to be used for logins until a new password is set. This is done to prevent inadvertent root lockouts. This is a feature of the SLES operating system and cannot be overridden.

Password aging

All local Avamar server operating system accounts must have their passwords changed every 60 days. Once a password is changed, it cannot be changed again for at least 24 hours.

Password complexity, length, and reuse

All local Avamar server operating accounts are required to have passwords with the following characteristics:

- Password complexity requires that you use at least 3 of the following 4 character sets:
  - 2 or more lowercase characters
  - 2 or more uppercase characters
  - 2 or more numeric characters
  - 2 or more special (non-alphanumeric) characters
- Minimum length is determined by complexity:
  - If you use any 3 character sets, the password must be at least 14 characters
  - If you use all 4 character sets, the password must be at least 11 characters
- Passwords must contain at least 3 characters that are different from the last password
- The previous 10 passwords cannot be reused

Additional firewall hardening (avfirewall)

Avamar 6.0 and later servers running supported versions of SLES operating system can be configured to use Linux IPTABLES.

Additional firewall hardening is a level-2 hardening feature that can be installed during Avamar server software installation.

Additional server firewall hardening satisfies the following STIG requirements:

- GEN006580 - Access Control Program

This feature is implemented by way of the **avfirewall** service.

avfirewall output is logged to /var/log/firewall on SLES servers. The /var/log/firewall file is not available on RHEL servers. However, firewall logging can be implemented using syslog on RHEL servers. The *EMC Avamar Administration Guide* provides details about implementing syslog.

Additional firewall configuration to support replication

Installing the **avfirewall** firewall hardening package will cause replication to fail until the following additional configuration is performed:

1. Open a command shell and log in using one of the following methods:
   - To log in to a single-node server, log in to the server as admin.
   - To log in to a multi-node server, log in to the utility node as admin.
2. Open /usr/local/avamar/etc/repl_cron.cfg in a UNIX text editor.
3. Add the following entries:

   ```
   --dstavmgr=--encrypt=tls
   --dstavmaint=--encrypt=tls
   ```

4. Save your changes.

Uninstalling level-2 hardening packages

To manually uninstall any level-2 security hardening package (operating system, password and firewall), perform the following:

1. Open a command shell and log in using one of the following methods:
   - To log in to a single-node server, log in to the server as root.
   - To log in to a multi-node server:
     a. Log in to the utility node as root, then load the dpnid OpenSSH key by typing:

        ```
        ssh-agent bash
        ssh-add /home/dpn/.ssh/dpnid
        ```

     b. When prompted, type the dpnid passphrase and press **Enter**.
2. Uninstall the hardening package by doing one of the following:
   - If uninstalling a package on a single-node server, type:

     ```
     rpm -e PACKAGENAME
     ```

     where PACKAGENAME is **avhardening.rpm** for the operating system hardening package, **avpasswd.rpm** for the password hardening packing, or **avfwb.rpm** for the firewall hardening package, respectively.
   - If uninstalling a package on a multi-node server, type:

     ```
     mapall --user=root --all rpm -e PACKAGENAME
     ```

     where PACKAGENAME is **avhardening.rpm** for the operating system hardening package, **avpasswd.rpm** for the password hardening packing, or **avfwb.rpm** for the firewall hardening package, respectively.
3. Repeat step 2 to uninstall additional level-2 security hardening packages.

APPENDIX A
Port Usage and Firewall Requirements

This appendix lists data port usage and firewall requirements for the Avamar system.

Avamar data port listing

Configure unobstructed client-server communication over the following data ports for all applicable firewalls.

Table 23 Data port listing (page 1 of 5)

Port/ Protocol	Purpose	Source	Destination	Remarks
7/ECHO	Java-related for Data Domain system communication			Only required if Data Domain is used to store Avamar client backups.
21/TCP	FTP	User-defined Microsoft Windows host that runs the Avamar Downloader Service	ftp.avamar.com	Optional, but recommended. Used to obtain update files. The following executable must be allowed bidirectional communication on port 21: AvamarDownloaderService.exe
22/TCP	SSH	Utility node and trusted administrator hosts	All nodes	Required.
53/UDP	DNS name resolution	DNS resolving name servers	All nodes	Optional, but recommended. Might restrict sources to specific name servers.
53/UDP	DNS name resolution	All nodes	DNS resolving name servers	Optional, but recommended. Might restrict destinations to specific name servers.
69/TCP	TFTP	Internal switch	Utility node	Required.
80/TCP	HTTP	User-defined web client hosts or reverse proxy web server	Utility node	Required. Permit access from all Avamar clients or only from reverse proxy web server (recommended).
	Avamar Downloader Service / web server / AvInstaller HTTP communication	User-defined Microsoft Windows host that runs the Avamar Downloader Service	Utility node or single-node server	Required. Permit access to and from specific hosts.
	Web browser / web server / AvInstaller HTTP communication	User-defined host with a web browser	Utility node or single-node server	Required. Permit access to and from specific hosts.
111/TCP	RPC portmapper	Local host	Data Domain system	RPC/NFS Portmapper. Only required if Data Domain is used to store Avamar client backups.
111/UDP	RPC portmapper	Local host	Data Domain system	RPC/NFS Portmapper. Only required if Data Domain is used to store Avamar client backups.
123/UDP	NTP	NTP time servers	All nodes	Required. Might restrict sources to specific time servers.
123/UDP	NTP	All nodes if external time servers are used	NTP time servers	Required. Might restrict destinations to specific time servers.
131	Related to the network environment			Required when storing backups on a Data Domain system.
137/UDP	Avamar proxy communication	Avamar proxy	Utility node or single-node server	

Table 23 Data port listing (page 2 of 5)

Port/ Protocol	Purpose	Source	Destination	Remarks
138/UDP	Avamar proxy communication	Avamar proxy	Utility node or single-node server	
139/TCP	Avamar proxy communication	Avamar proxy	Utility node or single-node server	
161/TCP	Getter/setter port for SNMP objects from Data Domain	Data Domain system	Utility node	Only required if Data Domain is used to store Avamar client backups. The port is used by Avamar to collect Data Domain system information by using the SNMP protocol.
163/TCP	SNMP traps from Data Domain	Utility node	Data Domain system	Only required if Data Domain is used to store Avamar client backups. The port receives SNMP traps from Data Domain systems.
443/TCP	HTTPS for Implements web restore, docs and downloads features	User-defined web client hosts	Utility node	Required. Permit access from all Avamar clients or only from reverse proxy web server (recommended).
	Avamar Downloader Service / web server / AvInstaller HTTP communication	User-defined Microsoft Windows host that runs the Avamar Downloader Service	Utility node or single-node server	Required. Permit access to and from specific hosts.
	Web browser / web server / AvInstaller HTTP communication	User-defined host with a web browser	Utility node or single-node server	Required. Permit access to and from specific hosts.
	Avamar proxy client communication	Avamar proxy	VMware vCenter	Required. Permit access to and from specific hosts.
	HTTPS for Avamar client web browser UI	User-defined client with a web browser	Utility node or single-node server	Required. Permit HTTPS access to and from Avamar client web UI.
514/UDP	Syslog	Utility node	Utility node	Optional. Logs Avamar server events to syslog.
902/TCP	Avamar proxy client communication	Avamar proxy	VMware ESX server	Required. Permit access to and from specific hosts.
1080/TCP	3ware RAID management	User-defined web client hosts	All nodes for Axion-M and Axion-E	Only required for legacy Axion-M and Axion-E hardware. Recommend only permitting access from trusted administrative hosts.
1234/TCP	HTTPS for avw_install utility	Trusted web client hosts	Utility node	Port 1234 must be open during the initial installation of Avamar software. After a successful installation, no Avamar service should be listening on port 1234. Permit access only to trusted hosts which are used to for the initial installation of Avamar software.
2049/TCP	NFS	Local host	Data Domain system	NFS daemon. Only required if Data Domain is used to store Avamar client backups.
2049/UDP	NFS	Local host	Data Domain system	NFS daemon. Only required if Data Domain is used to store Avamar client backups.

Table 23 Data port listing (page 3 of 5)

Port/ Protocol	Purpose	Source	Destination	Remarks
2052/TCP	NFS mountd	Local host	Data Domain system	NFS mountd (port statically assigned on Data Domain system). Only required if Data Domain is used to store Avamar client backups.
3008/TCP	Active/Archive communication	Local host	Data Domain system	Avamar Client to Data Domain communication on Active/Archive systems. Only required if Data Domain system is used to store Avamar client backups.
5555/TCP	Connection to administrator server PostgreSQL database	User-defined PostgreSQL client hosts	Utility node	Optional for connecting to PostgreSQL database from outside the module. Recommend only permitting access from hosts requiring access to administrator server database.
5556/TCP	Avamar Enterprise Manager server PostgreSQL database (emdb)	User-defined PostgreSQL client hosts	Avamar Enterprise Manager server node	Optional for connecting to PostgreSQL database from outside the module. Recommend only permitting access from hosts requiring access to administrator server database.
5557/TCP	Metadata search PostgreSQL database	Avamar Enterprise Manager	Access node (where metadata search database is installed)	Optional. Only required if metadata search feature is installed.
7778/TCP	RMI - Avamar Administrator server	Avamar Administrator management console	Utility node	Required. Recommend only permitting access from trusted administrative hosts.
7779/TCP	RMI - Avamar Administrator server.	Avamar Administrator management console	Utility node	Required. Recommend only permitting access from trusted administrative hosts.
7780/TCP	RMI - Avamar Administrator server	Avamar Administrator management console	Utility node	Required. Recommend only permitting access from trusted administrative hosts.
7781/TCP	RMI - Avamar Administrator server	Avamar Administrator management console	Utility node	Required. Recommend only permitting access from trusted administrative hosts.
8105/TCP	Tomcat server shutdown port for Avamar client web browser UI	Local host	Utility node or single-node server	Optional, but recommended. The /usr/local/avamar-tomcat/bin/shutdown.sh script makes a connection on port 8105, and sends a shutdown command to the running instance of tomcat. This connection can only be made from the local host. The server.xml file contains the definition for port 8105: ‹Server port="8105" shutdown="SHUTDOWN"› Do not modify this definition.
8109/TCP	Tomcat connector port for Avamar client web browser UI	Utility node	Utility node	Optional, but recommended. The Apache JServ Protocol (AJP) uses port 8109 to balance the work load for multiple instances of Tomcat. AJP can be turned off by removing the following element from the server.xml file: ‹Connector port="8109" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" /›

Table 23 Data port listing (page 4 of 5)

Port/ Protocol	Purpose	Source	Destination	Remarks
8181/TCP	Redirect port for Avamar client / AvInstaller communication	Avamar client	Utility node or single-node server	Required. Permit access to and from specific hosts.
8444/TCP	Redirect port for Tomcat HTTPS for Avamar client	Any network host running web browser	Utility node or single-node server	Required. Permit access to and from Avamar client web UI services.
8505/TCP	Tomcat server shutdown port	Local host	Utility node	Optional, but recommended. The /usr/local/avamar-tomcat/bin/shutdown.sh script makes a connection on port 8505, and sends a shutdown command to the running instance of tomcat. This connection can only be made from the local host. The server.xml file contains the definition for port 8505: ‹Server port="8505" shutdown="SHUTDOWN"› Do not modify this definition.
8509/TCP	Tomcat connector port	Utility node	Utility node	Optional, but recommended. The Apache JServ Protocol (AJP) uses port 8509 to balance the work load for multiple instances of Tomcat. AJP can be turned off by removing the following element from the server.xml file: ‹Connector port="8509" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" /›
8543/TCP	Redirect port for Tomcat HTTPS for Avamar Enterprise Manager	Any network host running web browser	Utility node	Optional, but recommended in order to use Avamar Enterprise Manager.
	Avamar Installation Manager Web GUI	Any network host running Avamar Administrator	Utility node or single-node server	Required. Permits adminstrative access to the Avamar Installation Manager Web GUI at URL: https://*servername*:8543/avi/avigui.html
8580/TCP	Redirect port for Avamar Downloader Service / AvInstaller communication	User-defined Microsoft Windows host that runs the Avamar Downloader Service	Utility node or single-node server	Required. Permit access to and from specific hosts.
	Redirect port for AvInstaller / Avamar Downloader Service communication	Utility node or single-node server	User-defined Microsoft Windows host that runs the Avamar Downloader Service	Required. Permit access to and from specific hosts.
	Redirect port for Web browser / AvInstaller communication	User-defined host with a web browser	Utility node or single-node server	Required. Permit access to and from specific hosts.
8778/TCP	RMI - Avamar Enterprise Manager	Utility node	Utility node (where Avamar Enterprise Manager is installed)	Required. Recommend only permitting access from the local host.

Table 23 Data port listing (page 5 of 5)

Port/ Protocol	Purpose	Source	Destination	Remarks
8779/TCP	RMI - Avamar Enterprise Manager login_server	Utility node	Utility node (where Avamar Enterprise Manager is installed)	Required. Recommend only permitting access from the local host.
8780/TCP	RMI - Avamar Enterprise Manager service_context	Utility node	Utility node (where Avamar Enterprise Manager is installed)	Required. Recommend only permitting access from the local host.
8781/TCP	RMI - Avamar Enterprise Manager node_context	Utility node	Utility node (where Avamar Enterprise Manager is installed)	Required. Recommend only permitting access from the local host.
9443/TCP	RMI - Avamar Management Console Web Services	User-defined host with a web browser	Utility node or single-node server	Required.
27000/TCP	Avamar client communications with Avamar server	Avamar client network hosts	All nodes	Required.
27000/TCP	Avamar server communications with Replicator target server (Avamar proprietary communication)	All nodes	Replicator target server	Required if server is used as Replicator source.
28001/TCP	Avamar client communications with administrator server	Avamar clients	Utility node	Required.
28002/TCP	Administrator server communications with Avamar client	Utility node	Avamar clients	Optional for browsing clients and cancelling backups from Avamar Administrator management console.
29000/TCP	Avamar client Secure Sockets Layer (SSL) communications with Avamar server	Avamar clients	All nodes	Required.
29000/TCP	Avamar server SSL communications with Replicator target server	All nodes	All Replicator target server nodes	Required if server is Replicator source.

Data Domain port usage

To enable communication between Avamar and a Data Domain system, review and implement the port usage and firewall requirements in "Port Requirements for Allowing Access to Data Domain System Through a Firewall," which is available on the Data Domain Support Portal at https://my.datadomain.com.

There are also several additional ports listed in "Avamar data port listing" on page 108 that must be open when you store Avamar backups on a Data Domain system.

INDEX

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X