



EMC<sup>®</sup> Avamar<sup>®</sup>

Version 7.1

## Product Security Guide

302-000-830

REV 03

EMC<sup>2</sup>

Copyright © 2001-2014 EMC Corporation. All rights reserved. Published in USA.

Published December, 2014

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

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

EMC<sup>2</sup>, EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

EMC Corporation  
Hopkinton, Massachusetts 01748-9103  
1-508-435-1000 In North America 1-866-464-7381  
[www.EMC.com](http://www.EMC.com)

# CONTENTS

|                  |  |           |
|------------------|--|-----------|
| <b>Figures</b>   |  | <b>7</b>  |
| <b>Tables</b>    |  | <b>9</b>  |
| <b>Preface</b>   |  | <b>11</b> |
| <b>Chapter 1</b> | <b>Introduction</b>  | <b>15</b> |
|                  | Security patches.....  | 16        |
|                  | Periodic security updates for multiple components.....                   | 16        |
|                  | Remediating security patch compatibility issues.....                     | 16        |
|                  | Email home notification using ConnectEMC.....                            | 16        |
|                  | Remote access.....   | 17        |
|                  | Avamar 7.1 security features.....  | 17        |
| <b>Chapter 2</b> | <b>User Authentication and Authorization</b>                             | <b>19</b> |
|                  | Overview of Avamar user accounts.....                                    | 20        |
|                  | Authentication systems.....  | 20        |
|                  | Avamar internal authentication.....                                      | 21        |
|                  | Directory service authentication.....                                    | 21        |
|                  | How Avamar authenticates users and assigns roles.....                    | 21        |
|                  | Roles.....   | 22        |
|                  | Administrator roles.....   | 22        |
|                  | Operator roles.....  | 23        |
|                  | User roles.....  | 25        |
|                  | Default user accounts.....   | 26        |
| <b>Chapter 3</b> | <b>Client/Server Access and Authentication</b>                           | <b>27</b> |
|                  | Network access control.....  | 28        |
|                  | Client/server authentication.....  | 28        |
|                  | Certificate acceptance workflow.....                                     | 29        |
|                  | One-way authentication.....  | 30        |
|                  | Requesting signed certificates using a Certificate Signing Request... .. | 30        |
|                  | Requesting signed certificates using an enrollment form.....             | 31        |
|                  | Using a private CA to sign Avamar node certificates.....                 | 32        |
|                  | Installing certificates in the Avamar system.....                        | 38        |
|                  | Configuring Avamar to use server authentication.....                     | 38        |
|                  | Importing a CA root certificate to Unix-like clients.....                | 39        |
|                  | Importing a CA root certificate to Windows clients.....                  | 40        |
|                  | Enforcing encrypted client/server communications.....                    | 41        |
|                  | Two-way authentication.....  | 41        |
|                  | Requesting client certificates using a Certificate Signing Request.....  | 42        |
|                  | Requesting client certificates using an enrollment form.....             | 43        |
|                  | Using a private CA to sign client certificates.....                      | 44        |
|                  | Configuring Avamar to use client authentication.....                     | 47        |
|                  | Installing a client certificate on a UNIX-like client.....               | 48        |
|                  | Installing a client certificate on a Windows client.....                 | 49        |

|                  |   |           |
|------------------|---|-----------|
|                  | Verify client/server authentication.....                                    | 49        |
|                  | Server authentication using Apache.....                                     | 49        |
|                  | Support for Subject Alternative Names.....                                  | 50        |
|                  | Create a private key for Apache.....  | 50        |
|                  | Generating a certificate signing request for Apache.....                    | 52        |
|                  | Obtain a public key certificate for Apache.....                             | 53        |
|                  | Configuring Apache to use a key and a root CA certificate.....              | 54        |
|                  | Server authentication using Tomcat.....                                     | 56        |
|                  | Deleting the default Tomcat key entry.....                                  | 57        |
|                  | Creating a custom Tomcat key entry.....                                     | 57        |
|                  | Generating a certificate signing request for Tomcat.....                    | 58        |
|                  | Obtaining a public key certification for Tomcat.....                        | 59        |
|                  | Importing a root certificate file or a chained certificate file.....        | 60        |
|                  | Importing the Tomcat public key certificate.....                            | 61        |
|                  | Restarting the Tomcat services.....   | 62        |
|                  | Changing the keystore password and Tomcat key entry password....            | 62        |
| <b>Chapter 4</b> | <b>Data Security and Integrity</b>  | <b>65</b> |
|                  | Data-in-flight encryption.....  | 66        |
|                  | Avamar 6.0 through 7.0 data-in-flight encryption.....                       | 66        |
|                  | Avamar 7.1 data-in-flight encryption.....                                   | 67        |
|                  | Unencrypted data-in-flight on new installs of Avamar 7.1 and newer<br>..... | 68        |
|                  | Client/server encryption behavior.....                                      | 69        |
|                  | Increasing Avamar server cipher strength .....                              | 69        |
|                  | Data-at-rest encryption.....  | 70        |
|                  | Internal data-at-rest encryption key management.....                        | 70        |
|                  | Avamar Key Manager.....   | 72        |
|                  | Data integrity.....   | 77        |
|                  | Data erasure.....   | 78        |
|                  | Requirements for securely deleting backups.....                             | 78        |
|                  | Securely deleting a backup.....   | 79        |
| <b>Chapter 5</b> | <b>System Monitoring, Auditing, and Logging</b>                             | <b>83</b> |
|                  | Client activity monitoring.....   | 84        |
|                  | Server monitoring.....  | 84        |
|                  | Monitoring server status.....   | 84        |
|                  | Monitoring system events.....   | 84        |
|                  | Event notification profiles.....  | 86        |
|                  | Email home notification.....  | 86        |
|                  | Auditing.....   | 86        |
|                  | Logs.....   | 87        |
| <b>Chapter 6</b> | <b>Server Security Hardening</b>  | <b>95</b> |
|                  | Overview.....   | 96        |
|                  | STIG compliance.....  | 96        |
|                  | Server security hardening levels.....                                       | 96        |
|                  | Level-1 security hardening.....   | 96        |
|                  | Advanced Intrusion Detection Environment (AIDE).....                        | 96        |
|                  | The auditd service.....   | 97        |
|                  | sudo implementation.....  | 97        |
|                  | Command logging.....  | 98        |
|                  | Locking down single-user mode on RHEL servers.....                          | 98        |

|                   |   |            |
|-------------------|---|------------|
|                   | Disabling Samba.....  | 99         |
|                   | Web server cipher suite hardening on pre-7.1 Avamar systems.....    | 100        |
|                   | Web server cipher suite hardening on Avamar server version 7.1 .... | 103        |
|                   | Removing suid bit from non-essential system binaries on RHEL.....   | 105        |
|                   | Preventing unauthorized access to GRUB configuration.....           | 105        |
|                   | Level-2 security hardening.....                                     | 106        |
|                   | Additional operating system hardening.....                          | 106        |
|                   | Additional password hardening.....                                  | 107        |
|                   | Additional firewall hardening (avfirewall).....                     | 109        |
|                   | Installing level-2 security hardening features.....                 | 110        |
|                   | Level-3 security hardening.....                                     | 113        |
|                   | Disabling Apache web server.....                                    | 113        |
|                   | Stopping the EMS.....   | 114        |
|                   | Disabling Dell OpenManage web server.....                           | 114        |
|                   | Stopping Avamar Desktop/Laptop.....                                 | 115        |
|                   | Disabling SSLv2 and weak ciphers.....                               | 115        |
|                   | Updating OpenSSH.....   | 118        |
|                   | Disabling SNMP.....   | 119        |
|                   | Disabling RPC.....  | 120        |
|                   | Configuring the firewall to block access to port 9443.....          | 120        |
|                   | Changing file permissions.....                                      | 121        |
|                   | Preparing for a system upgrade.....                                 | 122        |
| <b>Appendix A</b> | <b>Port Requirements</b>  | <b>125</b> |
|                   | Terminology.....  | 126        |
|                   | Avamar firewall.....  | 127        |
|                   | Controlling the firewall daemon.....                                | 127        |
|                   | Utility node ports.....   | 127        |
|                   | Utility node required inbound ports.....                            | 128        |
|                   | Utility node optional inbound ports.....                            | 131        |
|                   | Utility node required outbound ports.....                           | 132        |
|                   | Storage node ports.....   | 134        |
|                   | Storage node required inbound ports.....                            | 134        |
|                   | Storage node required outbound ports.....                           | 135        |
|                   | Avamar client ports.....  | 135        |
|                   | Avamar client required inbound ports.....                           | 136        |
|                   | Avamar client required outbound ports.....                          | 136        |
|                   | Avamar Downloader Service host ports.....                           | 137        |
|                   | Avamar Downloader Service host required inbound port.....           | 137        |
|                   | Avamar Downloader Service host required outbound ports.....         | 138        |
|                   | Ports when using a Data Domain system.....                          | 138        |
|                   | Required ports when using a Data Domain system.....                 | 138        |
| <b>Appendix B</b> | <b>Enterprise Authentication</b>                                    | <b>141</b> |
|                   | Enterprise authentication.....                                      | 142        |
|                   | Supported components and systems.....                               | 142        |
|                   | Configuring Enterprise authentication.....                          | 143        |
|                   | Configuring an LDAP interface.....                                  | 143        |
|                   | Configuring an NIS interface.....                                   | 146        |
| <b>Appendix C</b> | <b>IAO Information</b>  | <b>149</b> |
|                   | System-level accounts.....  | 150        |
|                   | Files with SUID bit and SGID bit.....                               | 150        |

## CONTENTS

# FIGURES

|   |                              |    |
|---|------------------------------|----|
| 1 | Users in Avamar domains..... | 20 |
|---|------------------------------|----|

## FIGURES

# TABLES

|    |   |     |
|----|---|-----|
| 1  | Revision history.....   | 11  |
| 2  | Typographical conventions.....  | 12  |
| 3  | Avamar user account information.....                                    | 20  |
| 4  | Administrator roles.....  | 23  |
| 5  | Operator roles.....   | 23  |
| 6  | User roles.....   | 25  |
| 7  | Avamar server Linux OS default user accounts.....                       | 26  |
| 8  | Avamar server software default user account.....                        | 26  |
| 9  | MCS default user accounts.....  | 26  |
| 10 | MCS PostgreSQL database default user accounts.....                      | 26  |
| 11 | Proxy virtual machine Linux OS default user account.....                | 26  |
| 12 | Cipher levels and associated OpenSSL suites.....                        | 67  |
| 13 | Options for installAKM.sh.....  | 74  |
| 14 | Error messages for installAKM.sh.....                                   | 74  |
| 15 | Critical files used by Avamar Key Manager.....                          | 76  |
| 16 | Component log files on a single-node Avamar system.....                 | 87  |
| 17 | Component log files on a utility node.....                              | 89  |
| 18 | Component log files on a storage node.....                              | 90  |
| 19 | Component log file on a spare node.....                                 | 91  |
| 20 | Component log files for the NDMP Accelerator.....                       | 91  |
| 21 | Component log files on an access node.....                              | 91  |
| 22 | Component log files on an Avamar Administrator client.....              | 92  |
| 23 | Component log files for an Avamar backup client.....                    | 93  |
| 24 | STIG requirements satisfied by AIDE.....                                | 96  |
| 25 | STIG requirements satisfied by the auditd service.....                  | 97  |
| 26 | STIG requirements satisfied by the implementation of sudo.....          | 97  |
| 27 | STIG requirements satisfied by the additional OS hardening package..... | 107 |
| 28 | STIG requirements satisfied by additional password hardening.....       | 108 |
| 29 | Cipher levels and associated OpenSSL suites.....                        | 116 |
| 30 | Required inbound ports on the utility node.....                         | 128 |
| 31 | Optional inbound ports on the utility node.....                         | 131 |
| 32 | Required outbound ports for the utility node.....                       | 132 |
| 33 | Required inbound ports on each storage node.....                        | 134 |
| 34 | Required outbound ports for each storage node.....                      | 135 |
| 35 | Required inbound ports on an Avamar client.....                         | 136 |
| 36 | Required outbound ports for an Avamar client.....                       | 137 |
| 37 | Required inbound port on an Avamar Downloader Service host.....         | 138 |
| 38 | Required outbound ports for an Avamar Downloader Service host.....      | 138 |
| 39 | Required ports when using a Data Domain system.....                     | 138 |
| 40 | Supported external authentication systems.....                          | 142 |



# 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 technical support professional if a product does not function properly or does not function as described in this document.

---

## Note

This document was accurate at publication time. Go to EMC Online Support (<https://support.emc.com>) 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   |
|----------|-------------------|---|
| 03       | December 15, 2014 | Service pack 1 release of Avamar 7.1.                           |
| 02       | August 15, 2014   | Added <a href="#">Avamar 7.1 security features on page 17</a> . |
| 01       | June 11, 2014     | Initial release of Avamar 7.1.                                  |

## Related documentation

The following EMC publications provide additional information:

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

The following publications provide additional information:

- *US Department of Defense (DoD) Security Technical Implementation Guide (STIG) for Unix*

## Special notice conventions used in this document

EMC uses the following conventions for special notices:

**NOTICE**

Addresses practices not related to personal injury.

---

---

**Note**

Presents information that is important, but not hazard-related.

---

**Typographical conventions**

In this document, EMC uses the typographical conventions shown in the following table:

**Table 2** Typographical conventions

|                         |   |
|-------------------------|---|
| <b>Bold</b>             | Use for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)   |
| <i>Italic</i>           | Use for full titles of publications referenced in text  |
| Monospace               | Use for: <ul style="list-style-type: none"> <li>• System code</li> <li>• System output, such as an error message or script</li> <li>• Pathnames, file names, prompts, and syntax</li> <li>• Commands and options</li> </ul> |
| <i>Monospace italic</i> | Use for variables   |
| <b>Monospace bold</b>   | Use for user input  |
| [ ]                     | 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 Support.

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.

### Online communities

Visit EMC Community Network at <http://community.EMC.com> for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners and certified professionals for all EMC products.

### Live chat

To engage EMC Customer Support 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 Support, 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 Support.

### 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 [DPAD.Doc.Feedback@emc.com](mailto:DPAD.Doc.Feedback@emc.com).

Please include the following information:

- Product name and version
- Document name, part number, and revision (for example, 01)
- Page numbers

- Other details that will help us address the documentation issue

# CHAPTER 1

## Introduction

This chapter includes the following topics:

- [Security patches](#).....16
- [Email home notification using ConnectEMC](#)..... 16
- [Remote access](#)..... 17
- [Avamar 7.1 security features](#)..... 17

## Security patches

Each Avamar release is available with a set of up-to-date security patches.

### Periodic security updates for multiple components

EMC periodically provides a security update for components of the Avamar system's host operating system. These periodic updates combine patches and updates released by the operating system's company (Red Hat or SuSE) since the previous Avamar periodic security update, and include relevant kernel-level and OS-level security patches and changes.

The periodic updates are cumulative. Install each periodic update issued for your Avamar system in order of release, starting with the first periodic update issued after the release of your Avamar system software.

EMC announces each periodic update through an EMC Security Advisory (ESA). The ESA provides details about the contents of the periodic update and installation instructions. Go to [https://support.emc.com/products/759\\_Avamar-Server](https://support.emc.com/products/759_Avamar-Server) to view these advisories and to register for email notifications.

EMC provides the periodic updates as Avamar update packages that can normally be installed through Avamar Enterprise Manager.

### Remedying security patch compatibility issues

If you separately install other security patches or security applications that are found to be incompatible with Avamar:

1. Remove the separately installed patches or applications.
2. Restore the Avamar system to its previous working configuration.
3. File a support case with EMC support that includes a specific description of the separately installed patches or applications.

---

#### Note

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 Customer 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 Customer 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 Customer Support to access their systems without WebEx.

## Avamar 7.1 security features

Installation of Avamar 7.1 software or upgrade to 7.1 installs, by default, hardening and firewall packages that improve security capabilities on the Avamar server. Installation of these packages does not restrict supported server functionality, and they cannot be uninstalled. In some cases, like future upgrades, EMC Support personnel will provide the steps and tools to enable necessary procedures (for instance, FTP capabilities for downloading packages to the server).



# CHAPTER 2

## User Authentication and Authorization

This chapter includes the following topics:

- [Overview of Avamar user accounts](#)..... 20
- [Authentication systems](#)..... 20
- [Roles](#)..... 22
- [Default user accounts](#)..... 26

## Overview of Avamar user accounts

A user account in Avamar can administer a domain or client. The user account defines the authentication system that is used to grant users access to the Avamar server. It also defines the role for the user, which controls the operations that a user can perform.

You can add user accounts to domains or individual clients. When you add a user account to a domain, the account can administer that domain and any subdomains beneath it. When you add a user account to an individual client, the account can perform backups and restores of that client, and access backups belonging to that client in the system.

In Avamar, users are entries in a domain or client access list. When you add a user account to the Avamar system, you are adding a new entry to a domain or client user access list.

In the following example, the user “Gretchen” has been added to both the Accounting domain and her computer. However, the authentication system and role are different. These are in fact two completely separate user accounts that happen to have the same username.

**Figure 1** Users in Avamar domains



The following table describes the information that comprises an Avamar user account.

**Table 3** Avamar user account information

| Information           | Description  |
|-----------------------|--|
| Username              | The username depends on the authentication system and must be in the format that the authentication system accepts. For example, the internal authentication system uses case-sensitive usernames, whereas Windows Active Directory usernames are case-insensitive. Usernames cannot be longer than 31 characters. |
| Authentication system | An authentication system is a username/password system that is used to grant users access to the Avamar server.  |
| Role                  | Roles define the allowable operations for each user account.   |

## Authentication systems

An authentication system is a username/password system that is used to grant domain and client users access to the Avamar server. Avamar supports its own internal authentication system (“Avamar authentication” or “avs”), as well as directory service authentication. Directory service authentication uses an existing LDAP v.3 directory service or an existing Network Information Service (NIS) to provide authentication.

## Avamar internal authentication

With Avamar internal authentication, you define the username and password for Avamar user accounts, and Avamar stores the information. Usernames are case-sensitive and cannot be longer than 31 characters.

No additional steps are required to use internal Avamar authentication to authenticate user accounts. You define the username and password for each account when you add the user in Avamar Administrator.

## Directory service authentication

When you use directory service authentication to authenticate and assign roles to Avamar users, you can take advantage of a directory service that already exists in an organization. You can use any LDAP v.3-compliant directory service, such as Microsoft Active Directory Domain Services. Also, you can use a Network Information Service (NIS) on its own or with the LDAP services.

### Avamar products that use directory service authentication

The following Avamar products use existing directory services to authenticate users:

- Avamar Administrator
- Avamar Web Restore
- Avamar client web UI

Avamar Web Restore requires the use of existing directory services to authenticate users. The other products also support other authentication methods.

#### NOTICE

When you delete an Avamar domain, Avamar removes the LDAP maps that rely on that Avamar domain for access. The directory service groups associated with the removed LDAP maps are not affected by the deletion.

### LDAP requirements

Avamar directory service authentication supports LDAP v.3-compliant directory services when the following conditions are met:

- LDAP server permits username bind through both of the following formats:
  - *username*
  - *username@domain.com*
- LDAP server permits searching for group membership by using a username.
- LDAP server permits searching for groups by using a search string.
- LDAP server account that is provided when adding an LDAP map has permission to run a nested `ldapsearch` command.

### Encrypted communication

Avamar's directory service authentication uses the Kerberos protocol for all communications with the Key Distribution Center. Avamar automatically encrypts usernames and passwords before sending them to port 88 on the Key Distribution Center.

## How Avamar authenticates users and assigns roles

To provide backwards compatibility with enterprise authentication and to account for the possibility of users in more than one LDAP mapped group, Avamar uses the following authentication and role assignment sequence for each login attempt:

1. When the username is in the format *user*, where *user* is a username without *@server* appended, then Avamar checks the internal Avamar authentication database. If the username, password, and domain match, then the login is successful and Avamar assigns the user a role in the Avamar database. If they do not match, then the login fails.
2. When the username is in the format *user@server*, where *user* is a username and *server* is the fully qualified domain name of the authentication server, then Avamar checks the login information by using enterprise authentication. If the username, password, and domain match, then the login is successful and Avamar assigns the user a role in the Avamar database. If there is no match, then the evaluation continues.
3. When the username is in the format *user@server* and authentication by using enterprise authentication fails, then Avamar checks the LDAP mapping system. The login attempt is checked against all mapped groups for a match of each of the following identifiers:
  - Username, the portion of the **User Name** field entry before the @ symbol.
  - Password, as entered in the **Password** field.
  - Avamar domain, as entered in the **Domain Name** field.
  - Directory service domain, the portion of the **User Name** field entry after the @ symbol.

When all identifiers match, the login is successful and Avamar assigns the user a role from the mapped group.

A user can be the member of mapped groups in different directory service domains. The role of the mapped group that matches the directory service domain provided during login is assigned to the user for that session.

When the user is a member of more than one mapped group in the same directory service domain, the role with the greatest authority is assigned.

4. When the login information does not meet the requirements of any of the previous steps, then the login fails and a failure message appears.

## Roles

Roles define the allowable operations for each user account.

There are three types of roles:

- Administrator roles
- Operator roles
- User roles

### Administrator roles

Administrators are generally responsible for maintaining the system.

You can only assign the role of administrator to user accounts at a domain level. This includes the top-level (root) domain or any other domain or subdomain. You cannot assign this role to user accounts at a client level.

You can assign the administrator role to users at the top-level (root) domain or to a specific domain or subdomain.

**Table 4** Administrator roles

| Administrator type    | Description   |
|-----------------------|---|
| 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 domains other than root generally have access to most of the features described in this guide, but typically can only view or operate on objects (backups, policy objects, and so forth) in 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 runtimes for maintenance activities, and so forth) is disallowed.<br><br>Furthermore, domain administrators: <ul style="list-style-type: none"> <li>• Cannot add or edit other subdomain administrators.</li> <li>• Cannot change their assigned role.</li> <li>• Can change their password.</li> </ul> |

## Operator roles

Operator roles are generally implemented to allow certain users 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.

You can only assign operator roles to user accounts at the domain level. You cannot assign these roles to user accounts at the client level. To add the user account to subdomains, you must have administrator privileges on the parent domain or above.

Users with an operator role do not have access to all features in Avamar Administrator. Instead, after login, they are presented with a single window that provides access to the features that they are allowed to use.

The following table describes the four operator roles.

**Table 5** Operator roles

| Operator type         | Description  |
|-----------------------|--|
| 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 completed without errors.<br><br>Restore only operators at the top-level (root) domain can perform restores for any client in the system. Restore only operators at a domain other than root can only perform restores for clients in that domain.<br><br>Restore only operators can restore backup data and monitor activities in the assigned domain.<br><br>By default, restore only operators cannot browse backups from the command line or the Avamar Web Restore interface. To enable these activities for a restore only |

**Table 5** Operator roles (continued)

| Operator type                   | Description   |
|---------------------------------|---|
|                                 | <p>operator, add the <code>noticketrequired</code> privilege by using the <code>avmgr chgv</code> command:</p> <pre>avmgr chgv --acct=<i>location</i> --u=<i>name</i> --ud=<i>auth</i> \ --pv="enabled,read,mclogin,noticketrequired"</pre> <p>where <i>location</i> is the subdomain of the operator, <i>name</i> is the Avamar username of the user, and <i>auth</i> is the external authentication system used to authenticate the user.</p>   |
| <p>Back up only operator</p>    | <p>Back up only operators are generally only allowed to perform backups and to monitor those activities to determine when they complete and if they completed without errors.</p> <p>Back up only operators at the top-level (root) domain can perform backups for any client or group in the system. Back up only operators at domains other than root can only perform backups for clients or groups in that domain.</p> <p>Back up only operators can perform on-demand backups of a client or a group, as well as monitor activities in the assigned domain.</p> <p>By default, back up only operators cannot perform backups from the command line. To enable command line backups for a back up only operator, add the <code>noticketrequired</code> privilege by using the <code>avmgr chgv</code> command:</p> <pre>avmgr chgv --acct=<i>location</i> --u=<i>name</i> --ud=<i>auth</i> \ --pv="enabled,read,mclogin,backup,noticketrequired"</pre> <p>where <i>location</i> is the subdomain of the operator, <i>name</i> is the Avamar username of the user, and <i>auth</i> is the external authentication system used to authenticate the user.</p>  |
| <p>Back up/restore operator</p> | <p>Back up/restore operators are generally only allowed to perform backups or restores and to monitor those activities to determine when they complete and if they completed without errors.</p> <p>As with roles assigned to other domain user accounts, back up/restore operators at the top-level (root) domain can perform backups and restores for any client or group in the system. Back up/restore operators at domains other than root can only perform backups and restores for clients or groups in that domain.</p> <p>Back up/restore operators can perform the following tasks in the assigned domain:</p> <ul style="list-style-type: none"> <li>• Perform on-demand backups for a client or group.</li> <li>• Perform restores.</li> <li>• Monitor activities.</li> </ul> <p>By default, back up/restore operators cannot browse backups from the command line or using the Avamar Web Restore interface, and cannot perform backups from the command line. To enable these activities, add the <code>noticketrequired</code> privilege by using the <code>avmgr chgv</code> command:</p> <pre>avmgr chgv --acct=<i>location</i> --u=<i>name</i> --ud=<i>auth</i> \ --pv="enabled,read,mclogin,backup,noticketrequired"</pre> <p>where <i>location</i> is the subdomain of the operator, <i>name</i> is the Avamar username of the user, and <i>auth</i> is the external authentication system used to authenticate the user.</p> |

**Table 5** Operator roles (continued)

| Operator type     | Description  |
|-------------------|--|
| Activity operator | <p>Activity operators are generally only allowed to monitor backup and restore activities and to create certain reports.</p> <p>Activity operators at the top-level (root) domain can view or create reports for backup and restore activities in all domains and subdomains. Activity operators at domains other than root can only view or create reports for backup and restore activities in that domain.</p> <p>Activity operators can perform the following tasks in the assigned domain:</p> <ul style="list-style-type: none"> <li>• Monitor activities.</li> <li>• View the group status summary.</li> <li>• View the Activity Report.</li> <li>• View the Replication Report.</li> </ul> |

## User roles

User roles limit the operations allowed for a user account to a specific client.

Users assigned to one of the user roles cannot log in to Avamar Administrator, Avamar Enterprise Manager, or the Avamar client web UI.

The following table describes the four user roles.

**Table 6** User roles

| User type                                   | Description  |
|---|--|
| Back Up Only User                           | Users assigned this role can initiate backups directly from the client by using the <code>avtar</code> command line.   |
| Restore (Read) Only User                    | Users assigned this role can initiate restores directly from the client by using the <code>avtar</code> command line or MCS web services.  |
| Back Up/Restore User                        | Users assigned this role can initiate backups and restores directly from the client by using the <code>avtar</code> command line or MCS web services.  |
| Restore (Read) Only/Ignore File Permissions | <p>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.</p> <p>This role is only available when you use internal authentication.</p> <p>Windows client user accounts should be assigned this role to ensure trouble-free restores, only if both of the following are true:</p> <ul style="list-style-type: none"> <li>• Users are authenticated using Avamar internal authentication.</li> <li>• The user will not access the Avamar client web UI.</li> </ul> |

## Default user accounts

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

**Table 7** Avamar server Linux OS default user accounts

| User account | Default password | Description                                      |
|--------------|------------------|--|
| root         | changeme         | Linux OS root account on all Avamar nodes.       |
| admin        | changeme         | Linux OS account for Avamar administrative user. |
| dpm          | changeme         | Linux OS account for Avamar maintenance user.    |

**Table 8** Avamar server software default user account

| User account | Default password | Description                               |
|--------------|------------------|---|
| root         | 8RttoTriz        | Avamar server software root user account. |

**Table 9** MCS default user accounts

| User account  | Default password | Description   |
|---------------|------------------|---|
| 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.                      |
| replonly      | 9RttoTriz        | Account for internal use by the MCS for replication.      |

**Table 10** MCS PostgreSQL database default user accounts

| User account | Default password | Description                                 |
|--------------|------------------|---|
| admin        |                  | No password, logged in on local node only.  |
| viewuser     | viewuser1        | Administrator server database view account. |

**Table 11** Proxy virtual machine Linux OS default user account

| User account | Default password | Description  |
|--------------|------------------|--|
| root         | avam@r           | Linux OS root account on all proxies deployed using the Avamar proxy appliance. This account is for internal use only. |

# CHAPTER 3

## Client/Server Access and Authentication

This chapter includes the following topics:

- [Network access control](#)..... 28
- [Client/server authentication](#).....28
- [One-way authentication](#)..... 30
- [Two-way authentication](#).....41
- [Verify client/server authentication](#).....49
- [Server authentication using Apache](#)..... 49
- [Server authentication using Tomcat](#).....56

## Network access control

Control of networking in the Avamar environment starts with awareness of several parts of the network.

### 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. Within the DNS domain, assign forward mapping to the Avamar utility node, or to the single-node Avamar server. Optionally, also assign reverse mapping to the utility node or single-node server.

For example, use the following forward mapping entry in a BIND environment:

```
avamar-1      A      10.0.5.5
```

Continuing the example, use the following optional reverse mapping for a zone serving the 5.0.10.in-addr.arpa subnet:

```
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 when accessing the Avamar system from a remote location.

### SNMP

Avamar provides support for system monitoring and event notification through the Simple Network Management Protocol (SNMP).

## Client/server authentication

Avamar clients and Avamar servers use Transport Layer Security (TLS) certificates and Public Key Infrastructure (PKI) for authentication and optional data-in-flight encryption.

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 a root certificate and set up a private CA.
- Use a self-signed certificate (not recommended in production environments and not discussed in detail in this guide).

### NOTICE

Installing Avamar server automatically generates a public/private key pair and a self-signed certificate in the `/data01/home/admin` directory on each Avamar server storage node and in the `/usr/local/avamar/etc` directory on the utility node. Use these self-signed certificates only for installation and testing. EMC does not recommend the use of self-signed certificates in production environments.

Configure the Avamar environment for one-way or two-way authentication between Avamar clients and the Avamar server:

- Use one-way authentication to have the Avamar client request authentication from the Avamar server, and the server send a certificate to the client. The client then validates the certificate. One-way authentication is also called server-to-client authentication in this guide.
- Use two-way authentication to have the client request authentication from the Avamar server, and have the Avamar server request authentication from the client. This client-to-server authentication combined with server-to-client authentication provides a stronger level of security.

In most cases, one-way authentication provides sufficient security. However, to provide more security, set up two-way authentication. Both configurations provide the capability of data-in-flight encryption.

## Certificate acceptance workflow

Avamar uses a specific workflow when a client validates a server certificate, and when a server validates a client certificate.

Avamar uses the following workflow when determining whether to accept a certificate:

1. Obtain the fully qualified domain name (FQDN) of the computer.
 

When connected to a computer through an IP address, use reverse-DNS to determine the FQDN of the computer.
2. Compare the FQDN to the value specified in the Common Name (CN) field of the certificate.
  - When the FQDN matches the value specified in the CN field, accept that the certificate validates the computer.
  - When the FQDN does not match, continue the workflow.
3. If the certificate has a wildcard character (\*) in the hostname portion of the value specified in the CN field, perform a simple wildcard match of the FQDN to the CN.
  - When the wildcard match is successful, accept that the certificate validates the computer.
  - When the match is unsuccessful, continue the workflow.

For example, the value “r\*.example.com” in the CN field of the certificate would match an FQDN such as: “real.example.com”, “right.example.com”, or “reality.example.com”; but would not match “alright.example.com”.
4. Compare the IP address of the computer to each IP address listed in the Subject Alternative Name (SAN) field of the certificate.
  - When the IP address of the computer matches an IP address in the SAN field, accept that the certificate validates the computer.
  - When the match is unsuccessful, reject the certificate and terminate the connection.

## One-way authentication

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, using the certificate acceptance workflow.

Obtain the certificates required by one-way authentication through one of the following alternative methods:

- Requesting signed certificates using a Certificate Signing Request  
This method does not normally result in a certificate that contains multiple IP addresses in the SAN field. To obtain certificates that include the SAN field, use one of the other methods.
- Requesting signed certificates using an enrollment form
- Signed certificates from a private CA

After obtaining signed certificates, complete the following tasks:

- Installing certificates in Avamar
- Configuring Avamar to use server authentication
- Importing a CA root certificate to Unix-like clients
- Importing a CA root certificate to Windows clients
- Enforcing encrypted client/server communications

## Requesting signed certificates using a Certificate Signing Request

A Certificate Signing Request (CSR) contains the basic information that a commercial CA uses to issue a certificate. Create separate CSRs for the utility node and for each storage node. Alternatively, create a single CSR that references several nodes through the CN field.

### Procedure

1. Download and install OpenSSL on the system that generates the CSRs.

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.

2. Using an account with write permission for the current working directory, type the following on a single command line:

```
openssl req -new -newkey rsa:3072 -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 web site at [www.openssl.org](http://www.openssl.org) provides information about the openssl req command.

---

3. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                    | Description   |
|--------------------------|---|
| Country Name             | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> .   |
| State or Province Name   | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.   |
| Locality Name            | City where the organization is located.   |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.  |
| Organizational Unit Name | Optional entry for more information about the organization, such as a department name.  |
| Common Name (CN)         | FQDN of the computer, or a wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. Example for single computer: <b>corp-1.example.com</b> . Example wildcard FQDN for several computers: <b>corp-*.example.com</b> . |
| Email Address            | Email address of the primary administrator of the computer or computers.  |
| Challenge password       | A password that must be provided before revoking the certificate. The password is only required if your certificate is compromised. Optional field.   |
| Company name             | Name for your company. The exact legal name is not required. Optional field.  |

OpenSSL creates the CSR and key in the current working directory.

4. Repeat these steps for another Avamar server node, or group of nodes sharing the CN field.
5. Submit the resulting CSRs to a commercial CA for signing.

## Requesting signed certificates using an enrollment form

Many commercial CAs provide signed certificates that include x509 v3 extensions, such as the Subject Alternative Name (SAN) field. The SAN extension permits the issuance of a certificate that applies to multiple IP addresses. Normally an enrollment form is used to request this type of certificate from these CAs.

When several IP addresses are included in the SAN field of a certificate, Avamar can use that certificate to authenticate:

- A multi-homed server, by using any one of its IP addresses.
- Several servers that share the certificate, by parsing the list of IP addresses.

### Procedure

1. Determine the FQDN of the multi-homed computer, or the wildcard FQDN that represents several computers.

2. Determine the IP addresses covered by the certificate.
3. Select a commercial CA and complete the certificate enrollment process.

The certificate request procedures used by commercial CAs vary. The certificate must meet the requirements in the following table.

| Attribute                      | Requirement  |
|--------------------------------|--|
| Key format                     | RSA  |
| Key size                       | 3072 bits  |
| Output format                  | PEM  |
| Private key format (keyout)    | PEM  |
| Private key format (nodes)     | Not encrypted  |
| File Name extension            | .pem   |
| Common Name (CN)               | FQDN of the computer, or wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value.                                      |
| Subject Alternative Name (SAN) | List of several IP addresses for a multi-homed computer, or a list of IP addresses for several computers sharing the certificate. A CIDR notation value can be used to refer to a range of IP addresses. |

## Using a private CA to sign Avamar node certificates

Create a private certification authority (private CA) within your company and use the private CA to sign the certificates for your Avamar nodes.

When creating and signing certificates, EMC recommends that you:

- Properly secure the private key associated with the root certificate.
- In a high-risk environment, use an air-gapped network 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.

Creating a private CA, and using the private CA to sign certificates for your Avamar nodes, requires the completion of several tasks. The following steps identify those tasks and the order in which to perform them. Many of the tasks describe the use of the OpenSSL software. Alternatively, other implementations of the SSL/TLS protocols can be used.

### Procedure

1. Generate a private CA root certificate and key.
2. Create a custom OpenSSL configuration file.
3. Create a CSR for each Avamar node.
4. Use the private CA to sign the certificates.

## Generating a private CA root certificate and key

Generate a private CA root certificate and key by using OpenSSL.

Complete this task to begin the process of using a private CA to sign certificates.

### Procedure

1. Download and install OpenSSL on the system that generates the CSRs.

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.

2. Log in to the private CA computer as root.
3. Change the working directory to the location where you want to store the private CA root certificate and key.

For example, you could store the private CA root certificate and key in `/etc/ssl/private`.

4. Type the following on a single command line:

```
openssl req -new -x509 -newkey rsa:3072 -keyform PEM -keyout
privateCAkey.pem -extensions v3_ca -outform PEM -out privateCAcert.pem -
days 3654
```

where:

- `privateCAkey.pem` is the file name of the private CA key
- `privateCAcert.pem` is the file name of the private CA certificate
- `3654` is the number of days the certificate is valid, here it is 3,654 days

---

### Note

Additional details on the `openssl req` command can be found on the OpenSSL web site at [www.openssl.org](http://www.openssl.org).

---

The program prompts for a passphrase.

5. Enter a passphrase for the key.

The passphrase should be memorable. It cannot be retrieved.

The program prompts for the same passphrase.

6. Re-enter the passphrase for the key.
7. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                  | Description   |
|------------------------|---|
| Country Name           | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> . |
| State or Province Name | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.                               |

| Field                    | Description  |
|--------------------------|--|
| Locality Name            | Name of the city where the organization is located.                                |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.             |
| Organizational Unit Name | Optional entry for additional organization information, such as a department name. |
| Common Name (CN)         | The display name for the root certificate  |
| Email Address            | Contact email address for all CA-related issues.                                   |

OpenSSL creates the private CA certificate and key in the current working directory.

8. Create back up copies of *privateCAcert.pem* and *privateCAkey.pem*.

### After you finish

Create a custom OpenSSL configuration file.

## Creating a custom OpenSSL configuration file

Modify the OpenSSL configuration file, `openssl.cnf`, to meet the requirements of your organization for server certificates.

### Before you begin

Generate a private CA root certificate and key.

Use the OpenSSL configuration file to provide the additional information that is required to create a server certificate that includes:

- FQDN that uses a wildcard
- Multiple IP addresses

Use this capability to create a single-server certificate to use with all of the nodes in an Avamar system.

### Procedure

1. Log in to the private CA computer as root.
2. Open `/etc/ssl/openssl.cnf` in a plain text editor.
3. For server and server-as-client certificates, add the following to the end of `openssl.cnf`:

```
[ server_ext ]
basicConstraints = CA:false
keyUsage = critical, digitalSignature, keyEncipherment
nsCertType = server,client
extendedKeyUsage = serverAuth, clientAuth
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 = MMM.MMM.MMM.MMM
DNS.0 = avamar00.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.
  - *avamar00.example.com* represents the FQDN of the server. An asterisk wildcard character can be used in the hostname portion of the FQDN to represent the hostnames of several computers.
  - *MMM.MMM.MMM.MMM* represents an extra IP address for the server.
4. (Optional) Add more IP keys and IP addresses to the `[alt_names]` section, using the following methods:
- Uncomment the `IP.1` key and replace *MMM.MMM.MMM.MMM* with an IP address. Use this format to add more keys and IP addresses as required.

For example:

```
[alt_names]
IP.0 = 192.168.100.21
IP.1 = 192.168.100.22
IP.2 = 192.168.99.16
```

- For any key, `IP.0` through `IP.n`, use a CIDR notation value to refer to a range of IP addresses.

For example:

```
[alt_names]
IP.0 = 192.168.100.21
IP.1 = 192.168.100.22
IP.2 = 192.168.99.16
IP.3 = 192.168.101.0/29
```

5. (Optional) Uncomment the `DNS.1` key to add an extra FQDN entry, or wildcard FQDN entry, to the `[alt_names]` section.

Use this format to add more keys and FQDN entries as required.

For example:

```
[alt_names]
...
DNS.0 = avamar0*.example0.com
DNS.1 = avamar0*.example1.com
DNS.2 = test.example.com
DNS.3 = node*.home.com
```

where the ellipsis represents IP keys not relevant to the example.

6. Save and close the file.

### After you finish

Create a CSR for your Avamar nodes.

## Creating a CSR for Avamar nodes

The Certificate Signing Request (CSR) provides the basic information required to create a certificate for an Avamar node.

### Before you begin

Create a custom OpenSSL configuration file for Avamar nodes.

Create a CSR for the utility node, and a separate CSR for each storage node. Alternatively, create a single CSR that references several nodes through the CN field, the SAN field, or both fields.

**Procedure**

1. Log in to the private CA computer as root.
2. Change the working directory to the location where you want to store the CSRs.  
For example, `/etc/ssl/private`.

3. Type the following, on a single command line:

```
openssl req -new -newkey rsa:3072 -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

4. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                    | Description   |
|--------------------------|---|
| Country Name             | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> .   |
| State or Province Name   | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.   |
| Locality Name            | City where the organization is located.   |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.  |
| Organizational Unit Name | Optional entry for more information about the organization, such as a department name.  |
| Common Name (CN)         | FQDN of the computer, or a wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. Example for single computer: <code>corp-1.example.com</code> . Example wildcard FQDN for several computers: <code>corp-*.example.com</code> . |
| Email Address            | Email address of the primary administrator of the computer or computers.  |
| Challenge password       | A password that must be provided before revoking the certificate. The password is only required if your certificate is compromised. Optional field.   |
| Company name             | Name for your company. The exact legal name is not required. Optional field.  |

OpenSSL creates the CSR and key in the current working directory.

5. Repeat these steps to create a CSR for another Avamar server node, or group of nodes.

**After you finish**

Use the private CA to sign certificates for the Avamar nodes.

**Signing Avamar node certificates by using a private CA**

Use a private CA to sign X.509-compliant certificates for your Avamar nodes.

**Before you begin**

Create at least one CSR for your Avamar nodes.

The procedure assumes the following:

- The CA certificate is in a file named `privateCAcert.pem`.
- The key for the CA certificate is in a file named `privateCAkey.pem`.
- A serial number seed file named `privateCA.srl` does not already exist.
- The default `openssl.cnf` file that is provided with OpenSSL is modified to include information specific to your organization.

**Procedure**

1. Log in to the private CA computer as root.
2. Type the following on a single command line:

```
openssl x509 -CA privateCAcert.pem -CAkey privateCAkey.pem -req -in
avamar-1req.pem -extensions server_ext -extfile openssl.cnf -outform PEM
-out avamar-1cert.pem -days 3654 -CAserial privateCA.srl -CAcreateserial
```

where:

- `privateCAcert.pem` is the full or relative path to the private CA certificate
- `privateCAkey.pem` is the full or relative path to the private CA certificate key
- `avamar-1req.pem` is the file name of the CSR
- `openssl.cnf` is the full or relative path to the OpenSSL configuration file
- `avamar-1cert.pem` is the file name of the resulting signed certificate
- `3654` is the number of days the certificate is valid, here it is 3,654 days
- `privateCA.srl` is a temporary serial number seed file

The program prompts for a passphrase for the private CA certificate key.

3. Type the passphrase for the certificate key.  
OpenSSL creates the signed certificate in the current working directory.
4. Repeat these steps for each CSR.
5. (Optional) Display the certificate content in text format by typing:

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

**After you finish**

Install the certificates in the Avamar system.

## Installing certificates in the Avamar system

Install certificates in the Avamar system by copying the certificates to the correct location on each node.

### Before you begin

Obtain certificates from a commercial CA or from your private CA.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Copy the certificate to the locations specified for the type of Avamar system.
  - Single-node system
    - Copy the certificate to: `/data01/home/admin/cert.pem`.
    - Copy the certificate to: `/usr/local/avamar/etc/cert.pem`.
  - Multi-node system
    - On each storage node, copy the certificate generated for that node to: `/data01/home/admin/cert.pem`.
    - On the utility node, copy the certificate generated for that node to: `/usr/local/avamar/etc/cert.pem`.
3. Copy the key associated with the certificate to the locations specified for the type of Avamar system.
  - Single-node system
    - Copy to the key to: `/data01/home/admin/key.pem`.
    - Copy to the key to: `/usr/local/avamar/etc/key.pem`.
  - Multi-node system
    - On each storage node, copy the key generated for that node to: `/data01/home/admin/key.pem`.
    - On the utility node, copy the key generated for that node to: `/usr/local/avamar/etc/key.pem`.
4. Stop and restart the Avamar server by typing the following commands:

```
dpnctl stop gsan
dpnctl start
```

### After you finish

Configure the Avamar system to use server authentication.

## Configuring Avamar to use server authentication

Configure the Management Console Server (MCS) to use server authentication.

### Before you begin

Obtain certificates from a commercial CA or from your private CA, and install the certificates in the Avamar system.

**Procedure**

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Open `/usr/local/avamar/var/mc/server_data/prefs/mcserver.xml` in a plain text editor.
3. In `mcserver.xml`, locate the `encrypt_server_authenticate` preference and change it as follows:
 

```
encrypt_server_authenticate=true
```
4. Save and close the file.
5. Stop and restart the MCS by typing:
 

```
dpnctl stop mcs
dpnctl start mcs
```
6. Use the following methods to enforce server authentication for all future client communications:
  - When using Avamar Administrator to create or edit a group, always select **Medium** or **High** from the **Encryption** method list.

---

**Note**

When you need to override this setting, refer to the EMC Avamar Administration Guide. That guide describes how to override the group encryption method for a specific client, for a specific backup, and for a specific restore.

- When using the `avtar` command, always include the `--encrypt=tls-sa` option, and either the `--encrypt-strength=medium` option or the `encrypt-strength=high` option.

**After you finish**

Configure all Avamar clients to accept the server certificates.

## Importing a CA root certificate to Unix-like clients

Allow a UNIX-like client to authenticate an Avamar server's certificate by copying the root certificate of the CA that signed the Avamar server's certificate to the UNIX-like client.

**Before you begin**

Do the following:

- Install server certificates in the Avamar system and configure the Avamar system to use server authentication.
- Determine the value of the `--sysdir` argument used when starting `avagent` on the client.

**Procedure**

1. Create the file `chain.pem` by using the correct method for the number of files in the root certificate:
  - When the root certificate is several files that form a certificate chain, use `cat` with the redirect and append operators to combine the certificates, by typing:

```
cat chain-cert-1> chain.pem
cat chain-cert-2>> chain.pem
cat chain-cert-3>> chain.pem
```

where *chain-cert-1*, *chain-cert-2*, and *chain-cert-3* represent the path to each certificate in the certificate chain.

The resulting combined file must be named `chain.pem`.

- When the root certificate is a single file, copy the root certificate to a file named `chain.pem`.
2. Copy `chain.pem` to the following location on the Unix-like client:

```
/path/chain.pem
```

where */path* is the value of the `--sysdir` argument. The default value is: `/usr/local/avamar/etc`.

For example, when the value of the `--sysdir` argument is the default, copy `chain.pem` to `/usr/local/avamar/etc/chain.pem`.

**After you finish**

Enforce encrypted communication between the Avamar server and its clients.

## Importing a CA root certificate to Windows clients

Allow a Windows client to trust the Avamar server's certificate by making the root certificate of the CA that signed the Avamar server's certificate available on the Windows client.

**Before you begin**

Install server certificates in the Avamar system and configure the Avamar system to use server authentication.

Client computers validate a server certificate based on a chain of trust between the server certificate and a trusted certificate installed on the client. A server certificate issued by a commercial CA is normally accepted by a Windows client because a chain of trust exists between the server certificate and trusted certificates pre-installed on the Windows client. When a public key certificate issued by a commercial CA is not trusted by a Windows client, or when the server certificate is issued by a private CA, import the CA's root certificate.

**Procedure**

1. Copy the root certificate to the Windows client.
2. Determine if the following file exists on the Windows client: `C:\Program Files\avs\etc\chain.pem`.
3. Depending on whether `chain.pem` exists, do one of the following:

| Condition                       | Action  |
|---------------------------------|---|
| <b>chain.pem exists</b>         | Append the information in the root certificate to the existing content of <code>chain.pem</code> .                      |
| <b>chain.pem does not exist</b> | Rename the root certificate to <code>chain.pem</code> and copy it to: <code>C:\Program Files\avs\etc\chain.pem</code> . |

**After you finish**

Enforce encrypted communication between the Avamar server and its clients.

**Enforcing encrypted client/server communications**

Configure the MCS to refuse plain-text communication from Avamar clients.

Completing this task forces Avamar clients to use the Avamar server's trusted public key to encrypt all communication sent to the Avamar server.

**Procedure**

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Open `/usr/local/avamar/var/mc/server_data/prefs/mcserver.xml` in a plain text editor.
3. Locate the `enforce_client_msg_encryption` preference and change it to the following:

```
enforce_client_msg_encryption=true
```

4. Save and close the file.
5. Restart the MCS by typing the following commands:

```
dpnctl stop mcs
dpnctl start mcs
```

**Two-way authentication**

When two-way authentication is enabled, the Avamar server provides authentication to the Avamar client and the Avamar client provides authentication to the Avamar server.

With two-way authentication, both of the following occur:

- 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, using the certificate acceptance workflow.
- The Avamar server requests authentication from the Avamar client, and the client sends the appropriate certificate to the server. The server then validates the certificate, using the certificate acceptance workflow.

**NOTICE**

Before beginning these tasks, enable one-way authentication.

Obtain the client certificates required for two-way authentication through one of the following alternative methods:

- Requesting client certificates using a Certificate Signing Request
 

This method does not result in a certificate that contains multiple IP addresses in the SAN field. To obtain certificates that include the SAN field, use one of the other methods.
- Requesting client certificates using an enrollment form
- Use a private CA to sign client certificates

After obtaining signed certificates, complete the following tasks:

- Configuring Avamar for client authentication
- Installing a client certificate on a UNIX-like client
- Installing a client certificate on a Windows client

## Requesting client certificates using a Certificate Signing Request

A Certificate Signing Request (CSR) contains the basic information that a commercial CA uses to issue a client certificate. Create a blanket CSR for all clients by using a wildcard FQDN in the CN field. To enhance security, create separate CSRs for each client.

### Procedure

1. Download and install OpenSSL on the system that generates the CSRs.

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.

2. Using an account with write permission for the current working directory, type the following on a single command line:

```
openssl req -new -newkey rsa:3072 -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.
3. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                    | Description   |
|--------------------------|---|
| Country Name             | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> .   |
| State or Province Name   | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.   |
| Locality Name            | City where the organization is located.   |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.  |
| Organizational Unit Name | Optional entry for more information about the organization, such as a department name.  |
| Common Name (CN)         | FQDN of the computer, or a wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. Example for single computer: <b>corp-1.example.com</b> . Example wildcard FQDN for several computers: <b>corp-*.example.com</b> . |
| Email Address            | Email address of the primary administrator of the computer or computers.  |

| Field              | Description   |
|--------------------|---|
| Challenge password | A password that must be provided before revoking the certificate. The password is only required if your certificate is compromised. Optional field. |
| Company name       | Name for your company. The exact legal name is not required. Optional field.  |

OpenSSL creates the CSR and key in the current working directory.

4. (Optional) When obtaining separate certificates for several groups of clients, or for several clients, repeat these steps for each required certificate.
5. Submit the resulting CSRs to a commercial CA for signing.

## Requesting client certificates using an enrollment form

Many commercial CAs provide signed certificates that include x509 v3 extensions, such as the Subject Alternative Name (SAN) field. The SAN extension permits the issuance of a certificate that applies to multiple IP addresses. Normally an enrollment form is used to request this type of certificate from these CAs.

When several IP addresses are included in the SAN field of a certificate, Avamar can use that certificate to authenticate:

- A multi-homed client, by using any one of its IP addresses.
- Several clients that share the certificate, by parsing the list of IP addresses.

### Procedure

1. Determine the FQDN of the multi-homed computer, or the wildcard FQDN that represents several computers.
2. Determine the IP addresses covered by the certificate.
3. Select a commercial CA and complete the certificate enrollment process.

The certificate request procedures used by commercial CAs vary. The certificate must meet the requirements in the following table.

| Attribute                      | Requirement   |
|--------------------------------|---|
| Key format                     | RSA   |
| Key size                       | 3072 bits   |
| Output format                  | PEM   |
| Private key format (keyout)    | PEM   |
| Private key format (nodes)     | Not encrypted   |
| File Name extension            | .pem  |
| Common Name (CN)               | FQDN of the computer, or wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. |
| Subject Alternative Name (SAN) | List of several IP addresses for a multi-homed computer, or a list of IP addresses for several  |

| Attribute | Requirement   |
|-----------|---|
|           | computers sharing the certificate. A CIDR notation value can be used to refer to a range of IP addresses. |

## Using a private CA to sign client certificates

Use your company's private certification authority (private CA) to sign client certificates.

### Before you begin

Generate a private CA root certificate and key.

Using a private CA to sign client certificates requires the completion of several tasks. The following steps identify those tasks and the order in which to perform them.

### Procedure

1. Create a custom OpenSSL configuration file for the clients.
2. Create a CSR for the clients
3. Sign client certificates by using a private CA.

### After you finish

Create a custom OpenSSL configuration file for the clients.

## Creating a custom OpenSSL configuration file for clients

Modify the OpenSSL configuration file, `openssl.cnf`, to meet the requirements of your organization for client certificates.

### Before you begin

Generate a private CA root certificate and key.

Use the OpenSSL configuration file to provide the additional information that is required to create a server certificate that includes:

- FQDN that uses a wildcard
- Multiple IP addresses

Use this capability to create a single client certificate to use with a group of Avamar clients.

### Procedure

1. Log in to the private CA computer as root.
2. Open `/etc/ssl/openssl.cnf` in a plain text editor.
3. For client certificates, add the following at the end of `openssl.cnf` (after the server entry):

```
[ 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 = MMM.MMM.MMM.MMM
DNS.0 = client00.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.
  - *client00.example.com* represents the FQDN of the client. An asterisk wildcard character can be used in the hostname portion of the FQDN to represent the hostnames of several computers.
4. (Optional) Add more IP addresses to the `[alt_names]` section, using the following methods:
    - Uncomment the `IP.1` key and replacing *MMM.MMM.MMM.MMM* with an IP address. Use this format to add more keys and IP addresses as required.
    - For any key, `IP.0` through `IP.n`, use a CIDR notation value to refer to a range of IP addresses.
  5. (Optional) Uncomment the `DNS.1` key to add an extra FQDN entry, or wildcard FQDN entry, to the `[alt_names]` section.

Use this format to add more keys and FQDN entries as required.

6. Save and close the file.

### After you finish

Create a CSR for your Avamar clients.

## Creating a CSR for Avamar clients

The Certificate Signing Request (CSR) provides the basic information required to create a certificate for an Avamar client.

### Before you begin

Create a custom OpenSSL configuration file for Avamar clients.

Create a separate CSR for each Avamar client. Alternatively, create a single CSR that references a group of Avamar clients through the CN field, the SAN field, or both fields.

### Procedure

1. Log in to the private CA computer as root.
2. Change the working directory to the location where you want to store the CSRs.  
For example, `/etc/ssl/private`.
3. Type the following, on a single command line:

```
openssl req -new -newkey rsa:3072 -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.
4. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                    | Description   |
|--------------------------|---|
| Country Name             | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> .   |
| State or Province Name   | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.   |
| Locality Name            | City where the organization is located.   |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.  |
| Organizational Unit Name | Optional entry for more information about the organization, such as a department name.  |
| Common Name (CN)         | FQDN of the computer, or a wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. Example for single computer: <code>corp-1.example.com</code> . Example wildcard FQDN for several computers: <code>corp-*.example.com</code> . |
| Email Address            | Email address of the primary administrator of the computer or computers.  |
| Challenge password       | A password that must be provided before revoking the certificate. The password is only required if your certificate is compromised. Optional field.   |
| Company name             | Name for your company. The exact legal name is not required. Optional field.  |

OpenSSL creates the CSR and key in the current working directory.

- Repeat these steps to create a CSR for additional clients, or groups of clients.

## Signing Avamar client certificates by using a private CA

Use a private CA to sign X.509-compliant certificates for your Avamar clients.

### Before you begin

Do the following:

- Generate a private CA root certificate and key
- Create a custom OpenSSL configuration file for clients
- Create a CSR for each client or group of clients

The procedure assumes the following:

- The CA certificate is in *privateCAcert.pem*.
- The key for the CA certificate is in *privateCAkey.pem*.
- The *privateCA.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.

### Procedure

1. Log in to the private CA computer as root.
2. Type the following on a single command line:

```
openssl x509 -CA privateCAcert.pem -CAkey privateCAkey.pem -req -in
avamarclientreq.pem -extensions client_ext -extfile openssl.cnf -outform
PEM -out avamarclientcert.pem -days 3654 -CAserial privateCA.srl -
CAcreateserial
```

where:

- *privateCAcert.pem* is the full or relative path to the private CA certificate
  - *privateCAkey.pem* is the full or relative path to the private CA certificate key
  - *avamarclientreq.pem* is the file name of the CSR
  - *openssl.cnf* is the full or relative path to the OpenSSL configuration file
  - *avamarclientcert.pem* is the file name of the resulting signed certificate
  - *3654* is the number of days the certificate is valid, here it is 3,654 days
  - *privateCA.srl* is a temporary serial number seed file
3. Type the passphrase for the certificate key and press Enter.  
OpenSSL creates the signed certificate in the current working directory.
  4. Repeat these steps for each CSR.
  5. (Optional) Display the certificate content in text format by typing:

```
openssl x509 -in avamarclientcert.pem -noout -text
```

## Configuring Avamar to use client authentication

Configure the Avamar system to authenticate client certificates.

### Before you begin

Obtain signed client certificates, and the signing authority's root certificate. The root certificate comes from either a commercial CA or your private CA.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Stop the Avamar server by typing:

```
dpnctl stop gsan
```

3. Determine if the file `chain.pem` exists in the following locations:

- Single-node system
  - `/data01/home/admin/chain.pem`
  - `/usr/local/avamar/etc/chain.pem`

- Multi-node system
  - On each storage node:  
/data01/home/admin/chain.pem
  - On the utility node:  
/usr/local/avamar/etc/chain.pem

4. Depending on whether chain.pem exists, do one of the following:

| Condition                | Action  |
|--------------------------|---|
| chain.pem exists         | Append the contents of the signing authority's root certificate to the contents of chain.pem in each location.          |
| chain.pem does not exist | Copy the signing authority's root certificate to chain.pem in the each of the locations specified in the previous step. |

5. Restart the Avamar server by typing:

```
dpnctl start
```

6. Enable client authentication by typing:

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

## Installing a client certificate on a UNIX-like client

Install a client certificate on a UNIX-like client. The Avamar server uses this certificate to authenticate the client when two-way authentication is enabled.

### Before you begin

Do the following:

- Configure the Avamar server for client authentication.
- Determine the value of the --sysdir argument used when starting avagent on the client.

### Procedure

1. Obtain a signed certificate and private key file for the client.
2. Copy the client's certificate to the following location:

```
/path/cert.pem
```

where /path is the value of the --sysdir argument. The default value is: /usr/local/avamar/etc.

For example, when the value of the --sysdir argument is the default, copy cert.pem to /usr/local/avamar/etc/cert.pem.

The certificate file must be named cert.pem.

3. Copy the private key file for the client's certificate to the following location:

```
/path/key.pem
```

where /path is the value of the --sysdir argument. The default value is: /usr/local/avamar/etc.

The key file must be named key.pem.

## Installing a client certificate on a Windows client

Install a client certificate on a Windows client. The Avamar server uses this certificate to authenticate the client when two-way authentication is enabled.

### Before you begin

Configure the Avamar server for client authentication.

### Procedure

1. Obtain a signed certificate and private key file for the client.
2. Copy the client's certificate to `C:\Program Files\avs\etc\cert.pem`.

The certificate file must be named `cert.pem`.

3. Copy the private key file for the client's certificate to `C:\Program Files\avs\etc\key.pem`.

The key file must be named `key.pem`.

## Verify client/server authentication

Verify an implementation of client/server authentication by running a test backup with server authentication enabled.

The test backup can be run by using either `avtar` from the command line or by using Avamar Administrator.

### Verify authentication with the `avtar` command

Use the `avtar` command to verify client/server authentication by running a backup and including the server authentication option `--encrypt=tls-sa`.

The server authentication option requires authentication of the Avamar server based on the trusted certificates installed on the Avamar client.

### Verify authentication with Avamar Administrator

To verify client/server authentication with Avamar Administrator, 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.

## Server authentication using Apache

Several Avamar web-based services use the Apache HTTP server (Apache) to supply a secure web browser-based user interface. Web browser connections with these applications use secure socket layer/transport layer security (SSL/TLS) to provide authentication and data security.

Apache handles the SSL/TLS sockets for Avamar web-based services when a connection is made on the default HTTP port. Apache redirects the connection request to an SSL/TLS socket and handles the encryption and authentication for that socket.

### Web browser authentication warning

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 HTTP 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 enable Apache to provide authentication, and prevent web browser authentication warnings, complete the following tasks:

- Create a private key for Apache
- Generate a certificate signing request for Apache
- Obtain a public key certificate for Apache
- Configure Apache to provide public key authentication

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

## Support for Subject Alternative Names

On an Avamar system, the Apache HTTP server (Apache), and each Apache Tomcat (Tomcat) web server, supports the X509 Version 3 (RFC 2459, section 4.2.1.7) extension. This extension provides support for certificates that include the Subject Alternative Name (SAN) field.

Apache and Tomcat can use a certificate with several IP addresses in the SAN field to provide authentication for:

- A multi-homed server, by using any one of its IP addresses.
- Several servers that share the certificate, by parsing the list of IP addresses.

Not all combinations of browser and OS support Subject Alternative Names. Test a SAN certificate with the browser and OS combinations used by your company before installing the certificate on a production system.

## Create a private key for Apache

The public key infrastructure (PKI) private key for an Avamar system's Apache HTTP server (Apache) can be generated using various levels of security.

Use the private key generation method that is appropriate for the level of security required by your organization.

The methods for generating a private key are:

- Create a private key without randomness and without a passphrase
- Create a private key with randomness and without passphrase
- Create a private key with passphrase and without randomness
- Create a private key with randomness and with a passphrase

When a passphrase-protected private key is used, Apache prompts for the passphrase every time the Apache process starts. The Apache configuration setting `SSLPassPhraseDialog` can be used to obtain the passphrase from a script. For more information, refer to Apache documentation available through the Apache web site at [www.apache.org](http://www.apache.org).

## Creating a private key for Apache

Create a public key infrastructure (PKI) private key for the Avamar system's Apache HTTP server (Apache).

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Type one of the following alternative commands.

| Key type  | Command   |
|---|---|
| Private key without randomness and without a passphrase | <code>openssl genrsa -out server.key 3072</code>                            |
| Private key with randomness and without a passphrase    | <code>openssl genrsa -rand binary-files -out server.key 3072</code>         |
| Private key without randomness and with a passphrase    | <code>openssl genrsa -aes128 -out server.key 3072</code>                    |
| Private key with randomness and with a passphrase       | <code>openssl genrsa -rand binary-files -aes128 -out server.key 3072</code> |

where:

- *server.key* is a pathname you provide for the private key.
- *binary-files* is a colon-separated list of paths to two or more binary files that OpenSSL uses to generate randomness.

3. (Key with passphrase) At the prompt, type a passphrase.

4. (Key with passphrase) At the prompt, retype the passphrase.

## Generating a certificate signing request for Apache

Create a certificate signing request (CSR) for the Apache HTTP server (Apache) on an Avamar system.

### Before you begin

Generate a private key for Apache.

A commercial certification authority (CA) uses the CSR when issuing a trusted private key certificate.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. 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.

3. (Key with passphrase) Type the passphrase for the private key and press **Enter**.

4. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

| Field                    | Description   |
|--------------------------|---|
| Country Name             | The two-letter ISO abbreviation for the country. The list of abbreviations is available on the ISO web site at <a href="http://www.iso.org">www.iso.org</a> . |
| State or Province Name   | In countries where it is applicable, the state or province where the organization is located. This entry cannot be abbreviated.                               |
| Locality Name            | City where the organization is located.   |
| Organization Name        | The exact legal name of the company. This entry cannot be abbreviated.  |
| Organizational Unit Name | Optional entry for more information about the organization, such as a department name.  |

| Field              | Description   |
|--------------------|---|
| Common Name (CN)   | FQDN of the computer, or a wildcard FQDN for several computers. The wildcard character (*) must only appear once, and only in the hostname portion of the FQDN value. Example for single computer: <code>corp-1.example.com</code> . Example wildcard FQDN for several computers: <code>corp-*.example.com</code> . |
| Email Address      | Email address of the primary administrator of the computer or computers.  |
| Challenge password | A password that must be provided before revoking the certificate. The password is only required if your certificate is compromised. Optional field.   |
| Company name       | Name for your company. The exact legal name is not required. Optional field.  |

OpenSSL creates the CSR and key in the current working directory.

#### After you finish

Use the CSR to obtain a trusted public key certificate from a commercial CA.

## Obtain a public key certificate for Apache

Obtain a public key certificate for the Avamar system's Apache HTTP server (Apache) from a commercial CA.

Provide a commercial CA with the CSR that was generated for Apache and complete any other requirements specific to that CA. After its requirements are met, the CA provides a public key certificate for Apache in the form of an electronic file, usually with the `.crt` filename extension.

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.

### Combining a multiple file certificate chain

Commercial certification authorities sometime provide a multiple file certificate chain that links the private key certificate to a trusted root CA certificate. Use this procedure to combine those files into a single file.

#### Before you begin

From a commercial CA, obtain a multiple file trusted root CA certificate chain.

#### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. 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.

### Results

The `cat` command with the redirect and append operators combines all of the files into a single file.

## Configuring Apache to use a key and a root CA certificate

Configure the Avamar system's Apache HTTP server (Apache) to use a private key, a public key certificate, and a trusted root CA certificate.

### Before you begin

Place in a temporary directory on the Avamar system's utility node the following:

- Private key for Apache
- Public key certificate for Apache
- Trusted root CA certificate for the public key certificate used by Apache

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as `admin`.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as `admin`.
    - b. Load the `dpnid` OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Change the working directory to the temporary location of the certificate, key, and certificate chain file.
3. Use the correct command sequence to move the certificate, key, and certificate chain file to the OS-specific default locations.
  - 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 Linux Enterprise 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
```

#### NOTICE

Custom locations can be specified for these files by changing the Apache SSL configuration file. However, the Apache SSL configuration file is overwritten during Avamar system upgrades. Restore that file after a system upgrade.

4. Restart Apache by typing:

```
website restart
```

## Restoring the Apache SSL configuration file

The Apache SSL configuration file is overwritten during Avamar system upgrades. This also overwrites custom paths for the certificate, key, and certificate chain file. To use custom paths restore the Apache SSL configuration file from the backup copy made during the upgrade.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Back up the latest version of the Apache SSL configuration file.

- On Red Hat Enterprise Linux:

```
cd /etc/httpd/conf.d/
cp ssl.conf ssl.conf.orig
```

- On SUSE Linux Enterprise Server:

```
cd /etc/apache2/vhosts.d/
cp vhost-ssl.conf vhost-ssl.conf.orig
```

3. Change the current working directory.

```
cd /usr/local/avamar/var/avi/server_data/package_data/  
UPGRADE_FROM_VERSION/ConfigureApacheSsl/
```

where `UPGRADE_FROM_VERSION` is the name of the directory created during the latest upgrade.

4. Extract the previous version backup copy of the Apache SSL configuration file, by typing:

```
tar -xzf node_0.s_*.*.*.tgz -C /
```

5. Restart Apache, by typing:

```
website restart
```

## Server authentication using Tomcat

The Apache HTTP server (Apache) handles access to Avamar web-based services. Apache provides server authentication information and creates an encrypted channel between the requestor and the Apache Tomcat (Tomcat) server port of the service. However, when a connection request directly addresses the Tomcat server, Apache is bypassed and Tomcat handles the authentication and encryption.

---

### Note

This topic describes a non-standard method of access Avamar web-based services. The tasks described by this topic are not required when using the standard web address of Avamar Enterprise Manager or Avamar Web Restore.

To bypass Apache and access the Tomcat servers for Avamar Enterprise Manager and Avamar Web Restore directly, use web addresses of the form:

- Avamar Enterprise Manager

```
https://AVAMARSERVER:8543/cas
```

- Avamar Web Restore

```
https://AVAMARSERVER:8444/dt1t/home.html
```

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

By using addresses of this form, Apache is bypassed and the Tomcat server handles authentication and encryption.

### Web browser authentication warning

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.

By default, the Tomcat servers on Avamar systems use a self-signed certificate not a trusted public key certificate. The Tomcat servers can encrypt communication using this certificate but cannot provide adequate authentication.

To enable the Tomcat servers to authenticate using a trusted public key certificate, and avoid web browser authentication warnings, complete the following tasks:

- Delete the default key entry
- Create a custom key entry
- Generate a certificate signing request
- Obtain a public key certificate
- Import chained or root certificates
- Import the public key certificate
- Restart the Tomcat services

- Change the root keystore password and Tomcat key entry password

## Deleting the default Tomcat key entry

Delete the default Apache Tomcat (Tomcat) key entry from the keystore on an Avamar system.

Complete this task as the first step in installing a trusted public key certificate for Tomcat.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Run the `keytool -delete` command by typing:

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

3. At the password prompt, type the root keystore password.

The default keystore password is **changeit**.

#### **NOTICE**

After completing all of the tasks associated with installing a trusted public key certificate for Tomcat, change the root keystore password.

### After you finish

Create a custom Tomcat key entry in the keystore.

## Creating a custom Tomcat key entry

After deleting the default Apache Tomcat (Tomcat) key entry from the keystore on an Avamar system, create a custom Tomcat key entry in the keystore.

### Before you begin

Delete the default Tomcat key entry from the keystore on an Avamar system.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.

b. Load the dpnid OpenSSH key by typing:

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

c. When prompted, type the dpnid passphrase and press **Enter**.

2. Run the `keytool -genkeypair` command by typing:

```
$JAVA_HOME/bin/keytool -genkeypair -keysize 1024 -alias tomcat -
keyalg RSA
```

3. At the password prompt, type the root keystore password.

The default keystore password is **changeit**.

**NOTICE**

After completing all of the tasks associated with installing a trusted public key certificate for Tomcat, change the root keystore password.

4. At each prompt, type the information described in the following table. Press **Enter** after each entry.

For optional fields, you can provide an empty value by typing a period (.) and pressing **Enter**.

**Note**

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

| Field               | Description   |
|---------------------|---|
| First and last name | Fully qualified domain name of the Avamar utility node or single-node server.     |
| Organizational unit | Organizational unit within the company that has authority over the Avamar system. |
| Organization        | Name of the company.  |
| City                | City in which the Avamar system is located.                                       |
| State               | State in which the Avamar system is located.                                      |
| Country             | Country in which the Avamar system is located.                                    |

**Results**

The `keytool -genkeypair` command creates a custom Tomcat key entry.

**After you finish**

Generate a certificate signing request (CSR) for Tomcat.

## Generating a certificate signing request for Tomcat

After creating a custom Apache Tomcat (Tomcat) key entry in the keystore of an Avamar system, generate a certificate signing request (CSR) for Tomcat.

**Before you begin**

Create a custom Tomcat key entry in the keystore on an Avamar system.

A public certification authority uses the CSR when creating a public key certificate for Tomcat.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Generate a CSR by typing:

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

3. At the password prompt, type the root keystore password.

The default keystore password is `changeit`.

#### NOTICE

After completing all of the tasks associated with installing a trusted public key certificate for Tomcat, change the root keystore password.

---

### Results

The `keytool` utility creates a CSR named `tomcat.certrequest` in root's home directory.

### After you finish

Obtain a public key certificate for Tomcat.

## Obtaining a public key certification for Tomcat

Use a certificate signing request (CSR) to request a public key certificate for the Apache Tomcat (Tomcat) services on an Avamar system. After a public certification authority (CA) provides the public key certificate, save it in the home directory of root on the Avamar system.

### Before you begin

Generate a CSR for Tomcat.

### Procedure

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.

When the public key certificate requirements are satisfied, the CA provides a public key certificate. The CA normally also provides a root certificate file or a chained certificate file to provide chain-of-trust authentication for the public key certificate.

2. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the `dpnid` passphrase and press **Enter**.
3. Save the public key certificate as `tomcat.cert` in the home directory of root.

#### After you finish

Import the root certificate file or a chained certificate file provided by the CA into the root keystore on the Avamar system.

## Importing a root certificate file or a chained certificate file

To configure chain-of-trust authentication for the public key certificate of the Apache Tomcat (Tomcat) on an Avamar system, import the root certificate file or chained certificate file provided by the public certification authority (CA).

#### Before you begin

Obtain a public key certificate for Tomcat and an associated root certificate file or chained certificate file.

A CA normally provides a root certificate file or chained certificate file along with the public key certificate. If the CA did not provide one of these files, skip this topic.

#### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Open the root certificate file or chained certificate file in a plain text editor.
3. Separate the contents of the root certificate file or chained certificate file into individual certificate files, using the `BEGIN CERTIFICATE` and `END CERTIFICATE` designations as the boundaries for each individual certificate file.
4. Save each individual certificate file with a distinguishing file name in the home directory of root.

5. Import an individual certificate file by typing:

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

where *individual\_cert* is the name of one of the individual certificate files derived from the root certificate file or chained certificate file provided by the CA.

6. At the password prompt, type the root keystore password.

The default keystore password is **changeit**.

#### NOTICE

After completing all of the tasks associated with installing a trusted public key certificate for Tomcat, change the root keystore password.

---

7. Repeat the previous two steps for each of the individual certificate files.

### After you finish

Import the Tomcat public key certificate.

## Importing the Tomcat public key certificate

Import a public key certificate for Apache Tomcat (Tomcat) into the root keystore on an Avamar system.

### Before you begin

Obtain a public key certificate for Tomcat from a public certification authority (CA) and save it

This task assumes that the public key certificate file was saved as `tomcat.cert` in the home directory of root. When performing the task, replace `tomcat.cert` with the name of the public key certificate file.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Import the certificate by typing:

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

3. At the password prompt, type the root keystore password.

The default keystore password is **changeit**.

**NOTICE**

After completing all of the tasks associated with installing a trusted public key certificate for Tomcat, change the root keystore password.

---

**Results**

The `keytool` command incorporates the public key certificate in the keystore entry shared by the Tomcat services on the Avamar system. Reference the entry using the alias: `tomcat`.

**After you finish**

Restart the Tomcat services.

## Restarting the Tomcat services

To make an imported public key certificate for Apache Tomcat (Tomcat) available for web browser authentication requests, restart the Tomcat services.

**Before you begin**

Import a public key certificate for Apache Tomcat (Tomcat) into the root keystore on an Avamar system.

**Procedure**

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the `dpnid` OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Restart the Tomcat services by typing:

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

**Results**

The Tomcat services provide the public key certificate in response to web browser authentication requests.

## Changing the keystore password and Tomcat key entry 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, change the keystore password. The password for the private key entry for the Tomcat servers must be identical to the

keystore password. When changing the root keystore password, also change the password for the Tomcat server private key entry.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Change the root keystore password by typing:

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

3. When prompted, type the old password, and then type the new password at each subsequent prompt.

4. Change the password of the Tomcat server private key entry by typing:

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

#### NOTICE

The new Tomcat key entry password that you provide in the next step must be identical to the new keystore password.

---

5. When prompted, type the old password, and then type the new password at each subsequent prompt.

6. Set the Tomcat server for Avamar Enterprise Manager to use the new password.

- a. Open `/usr/local/avamar-tomcat/conf/server.xml` in a plain-text editor.
- b. Find the Connector that contains the attribute: `port="8543"`.
- c. In that Connector, add the `keystorePass` attribute with the new password by typing:

```
keystorePass="newpassword"
```

where *newpassword* is the new password provided for the root keystore and for the Tomcat server key entry.

- d. Save and close the file.

7. Set the Tomcat server for Avamar Web Restore to use the new password.

- a. Open `/usr/local/avamar-dtlt-tomcat/conf/server.xml` in a plain-text editor.
- b. Find the Connector that contains the attribute: `port="8444"`.
- c. In that Connector, add the `keystorePass` attribute with the new password by typing:

```
keystorePass="newpassword"
```

where *newpassword* is the new password provided for the root keystore and for the Tomcat server key entry.

d. Save and close the file.

8. Restart the services by typing:

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

# CHAPTER 4

## Data Security and Integrity

This chapter includes the following topics:

- [Data-in-flight encryption](#).....66
- [Data-at-rest encryption](#).....70
- [Data integrity](#).....77
- [Data erasure](#).....78

## Data-in-flight encryption

Avamar can encrypt all data sent between Avamar clients and the Avamar server during transmission (data-in-flight encryption). Encryption methodology and levels are different depending on the Avamar system version.

You specify the default encryption method to use for client/server data transfers 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.

You can also 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

Higher cipher levels result in slower Avamar system performance.

---

## Avamar 6.0 through 7.0 data-in-flight encryption

To provide enhanced security during client/server data transfers, Avamar 6.0 through 7.0 supports two levels of data-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.

**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.

## Avamar 7.1 data-in-flight encryption

To provide enhanced security during client/server data transfers, Avamar 7.1 supports six levels of data-in-flight encryption: cleartext, insecure, low, legacy, 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.

Each cipher level maps to a specific set of OpenSSL suites as shown in the following table.

**Table 12** Cipher levels and associated OpenSSL suites

| Avamar cipher level | OpenSSL suites   |
|---------------------|--|
| cleartext           | NULL-SHA   |
| insecure            | ALL:NULL-SHA   |
| low                 | EDH-DSS-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA   |
| legacy              | EDH-DSS-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA   |
| medium              | ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA:AECDH-AES128-SHA |
| high                | ECDHE-ECDSA-AES256-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-GCM-SHA384:AES256-SHA  |

The insecure and low settings provide serve-side only encryption.

The Avamar 7.1 default is the legacy setting, which matches the behavior of Avamar 7.0 in the following ways:

- When you use the `avtar` command with the `--encrypt-strength=medium` option or you include `-encrypt-strength=medium` in `/usr/local/avamar/var/avtar.cmd`, the shared cipher is AES128-SHA.
- When you use the `avtar` command with the `--encrypt-strength=high` option or you include `-encrypt-strength=high` in `/usr/local/avamar/var/avtar.cmd`, the shared cipher is AES256-SHA.

**Note**

If you store Avamar 7.1 client backups on a Data Domain 5.5 system, TLS encryption of data in transit is supported. Avamar 7.1 backups on a Data Domain 5.4 or earlier system do not support encryption in transit.

## Unencrypted data-in-flight on new installs of Avamar 7.1 and newer

On new installs of Avamar 7.1 and newer, the Avamar firewall blocks unencrypted data-in-flight.

Upgrading a pre-7.1 version Avamar system to Avamar server version 7.1 or newer does not block unencrypted data-in-flight. Existing backup policies that are configured without encryption continue to work after an upgrade.

However, to increase security, all new installs of Avamar 7.1 and newer enforce firewall settings that block unencrypted data-in-flight. To permit unencrypted data-in-flight on new installs of Avamar 7.1 and newer, these firewall settings must be manually changed.

### Permitting unencrypted data-in-flight on new installs of Avamar 7.1 and newer

Make changes to the Avamar firewall settings to permit unencrypted data-in-flight on new installs of Avamar 7.1 and newer.

#### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Open in a plain text editor a new file named `/usr/local/avamar/lib/admin/security/gsan-port`.

3. Add the following line to the new file:

```
GSAN_PLAIN_TEXT='27000,'
```

4. Save and close the file.

5. (Multi-node systems only) Use `mapall` to copy the file to the storage nodes, by typing:

```
mapall --user=root copy /usr/local/avamar/lib/admin/security/gsan-port
```

6. (Multi-node systems only) Use `mapall` to move the file, by typing:

```
mapall --user=root mv gsan-port /usr/local/avamar/lib/admin/security/
```

## Client/server encryption behavior

Client/server encryption functional behavior in any given circumstance is dependent on a number of factors, including the `mcservers.xml` `encrypt_server_authenticate` value, and the `avtar` encryption settings used during that activity.

The `encrypt_server_authenticate` value is set to `true` when you configure server-to-client authentication.

During backup and restore activities, you control client/server encryption by specifying an option flag pair: `--encrypt` and `--encrypt-strength`. The `--encrypt-strength` option takes one of three values: `None`, `Medium`, or `High`.

## Increasing Avamar server cipher strength

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 Server (MCS)
- Ports 8778 and 8779 for the Enterprise Manager Server (EMS)
- Port 9443 for the Management Console Web Services

### Increasing cipher strength for the MCS

#### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as `admin`.
  - For a multi-node server, log in to the utility node as `admin`.
2. Open `/usr/local/avamar/var/mc/server_data/prefs/mcservers.xml` in a plain text editor.
3. Locate the `rmi_cipher_strength` setting and change it to **high**.
 

```
rmi_cipher_strength=high
```
4. Close `mcservers.xml` and save your changes.
5. Download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6:
  - a. In a web browser, go 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.
6. Restart the MCS and the scheduler by typing:

```
dpnctl stop mcs
dpnctl start mcs
dpnctl start sched
```

## Increasing cipher strength for the EMS

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Open `/usr/local/avamar/var/mc/server_data/prefs/emserver.xml` in a UNIX text editor.
3. Locate the `rmi_cipher_strength` setting and change it to **high**.
 

```
rmi_cipher_strength=high
```
4. Download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6:
  - a. In a web browser, go 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.
5. Close `emserver.xml` and save your changes.
6. Restart the EMS by typing:
 

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

## Data-at-rest encryption

An Avamar server can be configured to encrypt the data that is stored on it. This is called data-at-rest encryption.

Avamar provides two choices for managing data-at-rest encryption keys:

- Internal key management using the `avmaint` command
- External key management using the Avamar Key Manager program

---

### Note

You must commit to using one key management system at a time. Do not simultaneously use both internal and external key management systems on the same Avamar server as this could cause data loss.

---

## Internal data-at-rest encryption key management

When you enable data-at-rest encryption with Avamar’s internal key management 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.

The internal 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 are, over time, converted to use the 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.

## Enabling data-at-rest encryption using internal key management

Normally you enable data-at-rest encryption using internal key management during the installation or upgrade workflow. You can also enable it from the utility node command line after you complete an installation or upgrade.

---

### Note

Do not enable data-at-rest encryption during the maintenance window or when backups are running. For best results, perform this task during the backup window with the scheduler disabled.

---

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Type the following commands:

```
avmaint atrestencryption --restpassword='password' --avamaronly
avmaint atrestencryption --restsalt='salt' --avamaronly
```

where:

- *password* is a new data-at-rest encryption password.
  - *salt* is a user-defined character string (salt), which is used to create a new secure encryption key. *salt* can contain any character (ASCII as well as multibyte character sets are supported) and should be enclosed in single quotes. Empty strings are not allowed.
3. Wait for these commands to complete.
  4. Verify that data-at-rest encryption has been successfully enabled by typing:

```
avmaint nodelist --xmlperline=9999 | grep atrestencryption
```

Information similar to the following appears in your command shell:

```
<atrestencryption-status enabled="true" nr-salts="1"/>
```

where:

- `enabled="true" nr-salts="1"` indicates that data-at-rest encryption is enabled.
- `enabled="false" nr-salts="0"` indicates that data-at-rest encryption is not enabled.

## Rotating in a new data-at-rest encryption internal key

Use the utility node command line to supply a new salt to use create a new encryption key.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Type the following command:

```
avmaint atrestencryption --restsalt='salt' --avamaronly
```

where *salt* is a new user-defined character string (salt), which is used to create a new secure encryption key. *salt* can contain any character (ASCII as well as multibyte character sets are supported) and should be enclosed in single quotes. Empty strings are not allowed.

## Changing the salt table password

The Avamar system uses a password to access the salt table in the persistent store. Use the utility node command line to supply a new password.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Type the following command:

```
avmaint atrestencryption --restpassword='password' --avamaronly
```

where *password* is the new salt table password. *password* can contain any character (ASCII as well as multibyte character sets are supported) and should be enclosed in single quotes. Empty strings are not allowed.

## Avamar Key Manager

An alternative to internal key management for data-at-rest encryption is to use external key management by enabling Avamar Key Manager. Avamar Key Manager acts as a client of RSA Data Protection Manager to allow external key management through RSA Data Protection Manager.

When you install Avamar Key Manager it configures data-at-rest encryption on all Avamar nodes and registers with RSA Data Protection Manager. Avamar Key Manager then acts as a client of RSA Data Protection Manager and permits RSA Data Protection Manager to handle all key management tasks for Avamar data-at-rest encryption.

Avamar Key Manager uses public-key cryptography to secure all communications with RSA Data Protection Manager. As preparation for using external key management, you install a private key for Avamar Key Manager and a public key certificate for RSA Data Protection Manager on the Avamar system. Also the RSA Data Protection Manager administrator installs the public key for Avamar Key Manager on RSA Data Protection Manager.

---

**Note**

Data-at-rest encryption through Avamar Key Manager cannot be reversed. Data encrypted by this process can only be read using Avamar Key Manager's decryption algorithms and through keys that are stored in the RSA Data Protection Manager database. Avamar files that are required by this process are stored in `/usr/local/avamar/etc/akm`. Do not delete these files. The RSA Data Protection Manager database must be backed up as described in that product's documentation.

---

## Preparing the server for external key management

Avamar Key Manager uses public-key cryptography to secure all communications with RSA Data Protection Manager. Create and install certificates for this cryptography method.

**Procedure**

1. Generate a public/private key pair for the Avamar system.

The resulting files must be:

- Private key and certificate that is contained in a PKCS#12 file named `client.p12`
- Public key certificate that is contained in a Base64 encoded file named `client.pem`

2. Obtain the CA certificate of the certification authority that is used to sign the public/private key pair in the previous step.

The certificate must be in a Base64 encoded file. The certification authority can be commercial or private.

3. Change the name of the CA certificate file to `rt.pem`.
4. Copy `client.p12` and `rt.pem` to the `/usr/local/avamar/etc/akm` directory on the Avamar utility node or single-node server.
5. Set the permissions of these files to read-only, `0444 (-r--r--r--)`.
6. Use the Appliance Management Console of RSA Data Protection Manager to add `client.pem` and `rt.pem` into an Identity.
7. In RSA Data Protection Manager, associate the Identity for the Avamar system with the Identity group and Keyclass assigned to the Avamar system.

## Switching from internal to external key management

You can switch data-at-rest encryption from internal key management to external key management. During server maintenance, crunched stripes are converted to use the new external key.

---

**Note**

Before enabling external key management using Avamar Key Manager, ensure that old internal encryption keys are automatically stored in a secure manner so that data stripes encrypted with previous keys can always be decrypted and read.

---

## installAKM.sh

The `installAKM.sh` script installs, configures, and starts Avamar Key Manager.

**General information**

You can install Avamar Key Manager on any Avamar system that does not have data-at-rest encryption enabled. You can also install Avamar Key Manager on an Avamar system that already has internal key managed data-at-rest encryption enabled. When you do this it changes the data-at-rest encryption key to one that is managed through RSA Data Protection Manager.

**Usage**

The following table lists the options that are available for `installAKM.sh`.

**Note**

Options that include a space character must be enclosed by quotes.

**Table 13** Options for `installAKM.sh`

| Option  | Description   |
|---|---|
| -h<br>--help                                      | Displays command line help.   |
| -i <i>dpm-server</i><br>--ip= <i>dpm-server</i>   | Identifies the fully qualified domain name (FQDN) or IP address of the RSA Data Protection Manager computer, where <i>dpm-server</i> represents the fully qualified domain name, or IP address in dotted-quad format. This option is required.  |
| -k <i>keyclass</i><br>--keyclass= <i>keyclass</i> | Provides the RSA Data Protection Manager key class to use, where <i>keyclass</i> represents the name that identifies the key class in RSA Data Protection Manager. The key class value must be in quotes when the name includes a space character. This is optional. The default key class is Avamar AES 256 CFB. |
| --updatepassword                                  | Replaces the private key certificate password stored by Avamar Key Manager with a new one. When the password of the private key certificate for the Avamar server is changed, use this option to provide Avamar Key Manager with the new password.  |

**Error messages**

The following table provides information about error messages that can appear when you run `installAKM.sh`.

**Table 14** Error messages for `installAKM.sh`

| Message                                      | Description   |
|--|---|
| Error: No version(s) of dpnakm are installed | The Avamar Key Manager rpm is not installed. Install the required rpm by installing or upgrading to Avamar server version 7.1.0 or later. |

**Table 14** Error messages for `installAKM.sh` (continued)

| Message  | Description   |
|--|---|
| Error: <i>avmaint</i> is not executable          | The <i>avmaint</i> utility cannot be started, where <i>avmaint</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/bin/avmaint</code> .  |
| Error: <i>akm.xml</i> is not writable            | The <i>akm.xml</i> file cannot be opened for write access, where <i>akm.xml</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/etc/akm/akm.xml</code> .                         |
| Error: <i>hosts</i> is not readable              | The <i>hosts</i> file cannot be read, where <i>hosts</i> is the full path to the expected location. The default location is <code>/etc/hosts</code> .   |
| Error: <i>mapall</i> is not executable           | The <i>mapall</i> utility cannot be started, where <i>mapall</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/bin/mapall</code> .   |
| Error: <i>probe.xml</i> is not readable          | The <i>probe.xml</i> file cannot be opened for reading, where <i>probe.xml</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/var/probe.xml</code> .                            |
| Error: <i>akm_appreg.cfg</i> is not readable     | The <i>akm_appreg.cfg</i> file cannot be opened for reading, where <i>akm_appreg.cfg</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/bin/akm_appreg.cfg</code> .             |
| Error: <i>akm_appreg.cfg.org</i> is not readable | The <i>akm_appreg.cfg.org</i> file cannot be opened for reading, where <i>akm_appreg.cfg.org</i> is the full path to the expected location. The default location is <code>/usr/local/avamar/bin/akm_appreg.cfg.org</code> . |

## Enabling data-at-rest encryption using Avamar Key Manager

Run `installAKM.sh` to enable data-at-rest encryption using Avamar Key Manager.

### Before you begin

Install RSA Data Protection Manager on a separate computer and ensure that the Avamar utility node can communicate with that computer. By default, Avamar firewall opens the required ports on the utility node and on the storage nodes.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server, log in to the server as `admin`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the admin OpenSSH key by typing:

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

- c. When prompted, type the `admin_key` passphrase and press **Enter**.

2. Change the working directory by typing `cd /usr/local/avamar/bin`.

3. Type the following command:

```
./installAKM.sh -i dpm-server -k keyclass
```

where:

- *dpm-server* is the fully qualified domain name, or the IP address, of the RSA Data Protection Manager computer.
- *keyclass* is the name of the key class in RSA Data Protection Manager.

---

#### Note

The key class name must be the valid name of an existing RSA Data Protection Manager key class.

---

4. At the Please enter the DPM Password prompt, type the password for the Avamar system's private key certificate.

#### Results

The script starts the Avamar Key Manager service, registers as a client of RSA Data Protection Manager, and enables data-at-rest encryption.

## Backing up critical Avamar Key Manager files

Avamar Key Manager relies on files stored in `/usr/local/avamar/etc/akm`. Without these files Avamar Key Manager cannot function normally and data cannot be encrypted or decrypted.

To protect these files, back them up after the first time you run `installAKM.sh`. In most cases, a backup is made automatically. The following table describes the possible configurations and the associated backup considerations.

**Table 15** Critical files used by Avamar Key Manager

| Configuration                                 | Backup considerations   |
|---|---|
| Multi-node Avamar server                      | Required files are automatically backed up to the storage nodes by the <code>installAKM.sh</code> script.                             |
| Single-node Avamar server with Data Domain    | Required files are automatically backed up to the Data Domain system through the Checkpoint Backup feature ( <code>cpbackup</code> ). |
| Single-node Avamar server without Data Domain | Required files are not automatically backed up. Manually back up all files in <code>/usr/local/avamar/etc/akm</code> .                |

---

#### Note

The RSA Data Protection Manager database stores the keys used to encrypt Avamar's at rest data. Back up this database to assure continued access to Avamar's at rest encrypted data. If an encryption key is lost then the encrypted data cannot be decrypted.

---

## Avamar Key Manager unavailability

After enabling data-at-rest encryption using Avamar Key Manager, Avamar server relies on communication with RSA Data Protection Manager to perform the following:

- Restart the `gsan` process on any node
- Restart the `gsan` process during a restart of the Avamar server

- Rotate in a new data-at-rest encryption external key
- Run an HFS check

When Avamar Key Manager cannot contact RSA Data Protection Manager, running these tasks fails and the Avamar server logs the events.

## Changing the private key certificate password

If you change the private key certificate password, provide Avamar Key Manager with the new password.

---

### Note

Do not update the private key certificate password during the maintenance window or when backups are running. For best results, perform this task during the backup window with the scheduler disabled.

---

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server, log in to the server as admin.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the admin OpenSSH key by typing:

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

c. When prompted, type the `admin_key` passphrase and press **Enter**.

2. Change the working directory by typing `cd /usr/local/avamar/bin`.

3. Type the following command:

```
./installAKM.sh --updatepassword -i dpm-server
```

where *dpm-server* is the fully qualified domain name, or the IP address, of the RSA Data Protection Manager computer.

4. Ensure that the storage server (also known as GSAN) is idle.

The GSAN is idle when it is not doing any of the following: backups, restores, and maintenance activities.

5. Type **y** and then press **Enter** to confirm that the storage server is idle.

6. At the `Please enter the DPM Password` prompt, type the new private key certificate password, and then press **Enter**.

### Results

The password update is complete and Avamar Key Manager stores the new password.

## Data integrity

Checkpoints are server 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 deleting server checkpoints. *EMC Avamar Administration Guide*

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.

## Requirements for securely deleting backups

### Avamar requirements

- 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.

### Other requirements

- 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.

## Securely deleting a backup

The `securedelb` program enables you to securely erase a backup on the Avamar server.

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 Customer Support for any questions regarding the effect of company procedures on the Avamar server software.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server, log in to the server as `admin`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the admin OpenSSH key by typing:

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

- c. When prompted, type the `admin_key` passphrase and press **Enter**.

2. Locate the backups to securely delete by typing the following command:

```
securedelb getb --id=user@auth --password=password --
account=domain/client
```

where:

- `user` is the Avamar username.
- `auth` is the authentication system used by that user (the default internal authentication domain is `avamar`).
- `password` is the password for the `user@auth` account.
- `domain/client` is the full location of the client machine.

3. Locate the backup to delete in the list, and then note the date in the **created** field.

4. Securely delete the backup by typing the following command:

```
securedelb delb --account=location --date=date --id=user@auth --
password=password
```

where:

- *location* is the location of the backup, expressed as a file path relative to the current working directory. However, if the first character is a slash (/), the value is treated as an absolute file path.
- *date* is the backup date noted in the previous step.
- *user* is the Avamar username.
- *auth* is the authentication system used by that user (the default internal authentication domain is avamar).
- *password* is the password for the *user@auth* account.

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

---

#### Note

Do not interrupt `securedelb delb` command. If interrupted, all data will not be securely deleted.

If successful, the `securedelb delb` command returns the following response:

```
1 Request succeeded
```

If unsuccessful, the `securedelb delb` command returns the following response:

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

5. If an error is encountered:
  - Search the knowledge base on EMC Online Support, for the specific error code.
  - If the required information is not found, engage EMC Customer Support using Live Chat, or create a Service Request.
6. 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*'
```

7. If any such messages are present:
  - Search the knowledge base on EMC Online Support, for the specific error code.
  - If the required information is not found, engage EMC Customer Support using Live Chat, or create a Service Request.
8. 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.

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
```

9. If backups were performed before the most recent checkpoint was taken, roll the server back to the most recent checkpoint, and then attempt to securely delete the backup again.

10. Repeat the previous step for all applicable checkpoints.
11. Repeat this entire procedure on all other Avamar servers to which this Avamar server replicates backups.



# CHAPTER 5

## System Monitoring, Auditing, and Logging

This chapter includes the following topics:

- [Client activity monitoring](#).....84
- [Server monitoring](#).....84

## 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

Avamar systems provide monitoring of several items on the Avamar server.

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 backups, 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.

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 these 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.

#### **NOTICE**

For maximum security, EMC recommends implementing remote syslog monitoring as described in the *EMC Avamar Administration Guide*.

---

#### **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.

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 EMC Data Domain System 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 Customer 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

Avamar systems provide an email home feature.

When fully configured and enabled, the email home feature automatically emails the following information to EMC Customer Support 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.

## 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.

Log information is organized into tables for each Avamar component. For additional information on log files, refer to the Avamar guide for the specific component.

### Single-node system log files

The following table lists the pathnames for the log files created by components of a single-node Avamar system.

**Table 16** Component log files on a single-node Avamar system

| Component                          | Pathname  |
|------------------------------------|---|
| Avamar Administrator               | <pre> /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 </pre> |
| Avamar Enterprise Manager (Tomcat) | <pre> /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 </pre>  |
| Avamar Enterprise Manager (Server) | <pre> /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 </pre> |
| Maintenance                        | <pre> /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.# </pre>   |

**Table 16** Component log files on a single-node Avamar system (continued)

| Component                 | Pathname  |
|---------------------------|---|
|                           | <code>/usr/local/avamar/var/cron/suspend.log</code>   |
| avw_install utility       | <code>/usr/local/avamar/var/avw_cleanup.log</code><br><code>/usr/local/avamar/var/avw_install.log</code><br><code>/usr/local/avamar/var/avw-time.log</code><br><code>/usr/local/avamar/var/log/dpnavwinstall-VERSION.log</code>                                   |
| axion_install utility     | <code>/usr/local/avamar/var/axion_install_DATE_TIME.log</code>  |
| Avamar File System (AvFS) | <code>/usr/local/avamar/var/axionfs.log</code>  |
| change-passwords utility  | <code>/usr/local/avamar/var/change-passwords.log</code>   |
| ddrmaint utility          | <code>/usr/local/avamar/var/log/ddrmaint.log</code>   |
| dpnctl utility            | <code>/usr/local/avamar/var/log/dpnctl.log</code>   |
| dpnnetutil utility        | <code>/usr/local/avamar/var/log/dpnnetutil-version.log</code><br><code>/usr/local/avamar/var/log/dpnnetutil.log*</code><br><code>/usr/local/avamar/var/log/dpnnetutilbgaux.log</code><br><code>/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log</code> |
| permctl utility           | <code>/usr/local/avamar/var/log/permctl.log</code>  |
| resite utility            | <code>/usr/local/avamar/var/dpnresite-version.log</code><br><code>/usr/local/avamar/var/mcspref.log</code><br><code>/usr/local/avamar/var/nataddr.log</code><br><code>/usr/local/avamar/var/smtphost.log</code>   |
| timedist utility          | <code>/usr/local/avamar/var/timedist.log</code>   |
| timesyncmon program       | <code>/usr/local/avamar/var/timesyncmon.log</code>  |
| Avamar Replicator         | <code>/usr/local/avamar/var/cron/replicate.log</code>   |
| Avamar license server     | <code>/usr/local/avamar/var/ascd-PORT.log</code>  |
| Storage server            | <code>/data01/cur/err.log</code><br><code>/data01/cur/gsan.log</code>   |

## Utility node log files

The following table lists the pathnames for the log files created by components of the utility node.

**Table 17** Component log files on a utility node

| Component                          | Pathname   |
|------------------------------------|--|
| Avamar Administrator               | <pre> /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 </pre> |
| Avamar Enterprise Manager (Tomcat) | <pre> /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 </pre>   |
| Avamar Enterprise Manager (Server) | <pre> /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 </pre>  |
| Maintenance                        | <pre> /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 </pre>   |
| avw_install utility                | <pre> /usr/local/avamar/var/avw_cleanup.log /usr/local/avamar/var/avw_install.log /usr/local/avamar/var/avw-time.log </pre>  |

**Table 17** Component log files on a utility node (continued)

| Component                 | Pathname  |
|---------------------------|---|
|                           | /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<br>/usr/local/avamar/var/log/dpnnetutil.log*<br>/usr/local/avamar/var/log/dpnnetutilbgaux.log<br>/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/timesyncmon.log   |
| Avamar Replicator         | /usr/local/avamar/var/cron/replicate.log  |
| Avamar license server     | /usr/local/avamar/var/ascd-PORT.log   |
| switch_monitoring utility | /usr/local/avamar/var/log/switch_monitoring.log   |

## Storage node log files

The following table lists the pathnames for the log files created by the storage node.

**Table 18** Component log files on a storage node

| Component          | Pathname  |
|--------------------|---|
| Storage server log | /data01/cur/err.log<br>/data01/cur/gsan.log                 |
| dpnnetutil utility | /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log |

**Table 18** Component log files on a storage node (continued)

| Component           | Pathname                                       |
|---------------------|--|
|                     | /usr/local/avamar/var/log/dpnnetutilbgaux.log  |
| Maintenance task    | /usr/local/avamar/var/ntpd_keepalive_cron.log* |
| timesyncmon program | /usr/local/avamar/var/timesyncmon.log*         |

## Spare node log file

The following table lists the pathname for the spare node log file.

**Table 19** Component log file on a spare node

| Component          | Pathname   |
|--------------------|--|
| dpnnetutil utility | /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log<br>/usr/local/avamar/var/log/dpnnetutilbgaux.log |

## Avamar NDMP Accelerator log files

The following tables list the pathnames for the log files created by the Avamar NDMP Accelerator.

**Table 20** Component log files for the NDMP Accelerator

| Component          | Pathname   |
|--------------------|--|
| avndmp log         | /usr/local/avamar/var/{FILER-NAME}/*.avndmp.log  |
| dpnnetutil utility | /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log<br>/usr/local/avamar/var/log/dpnnetutilbgaux.log |

## Access node log files

The following table lists the pathname for the log files created by an access node.

**Table 21** Component log files on an access node

| Component          | Pathname   |
|--------------------|--|
| dpnnetutil utility | /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log<br>/usr/local/avamar/var/log/dpnnetutilbgaux.log |

## Avamar Administrator client log files

The following tables list the pathnames for the log files created by the Avamar Administrator client.

**Table 22** Component log files on an Avamar Administrator client

| Component  | Operating system                               | Pathname  |
|--|--|---|
| Avamar Administrator management console                        | Windows 7<br>Windows Vista Windows XP<br>Linux | C:\Users\USERNAME<br>\.avamardata\var\mc\gui_log<br>C:\Documents and Settings<br>\USERNAME\.avamardata\var<br>\mc\gui_log<br>\$HOME/.avamardata/var/mc/<br>gui_log/mcclient.log.0 |
| Avamar Administrator management console command line interface | UNIX   | \$HOME/.avamardata/var/mc/<br>gui_log/mccli.log.0   |

## Backup client log files

The following table lists the pathnames for the log files created by Avamar components on an Avamar backup client.

**Table 23** Component log files for an Avamar backup client

| Component                             | Pathname   |
|---------------------------------------|--|
| Client avagent process (all clients)  | C:\Program Files\avs\var\avagent.log   |
| Client avtar process (all clients)    | C:\Program Files\avs\var\clientlogs\{WORKORDER-ID}.alg<br>C:\Program Files\avs\var\clientlogs\{WORKORDER-ID}.log |
| Avamar Client for Windows tray applet | C:\Program Files\avs\var\avsccl.log  |
| Avamar Plug-in for DB2                | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |
| Avamar Exchange Client                | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |
| Avamar NDMP Accelerator               | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |
| Avamar Client for NetWare             | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |
| Avamar Plug-in for Oracle             | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |
| Avamar Plug-in for SQL Server         | /usr/local/avamar/var/client/{WORKORDER-ID}.log  |



# CHAPTER 6

## Server Security Hardening

This chapter includes the following topics:

- [Overview](#)..... 96
- [Level-1 security hardening](#)..... 96
- [Level-2 security hardening](#)..... 106
- [Level-3 security hardening](#)..... 113

## Overview

Avamar 6.0 and later servers running the SUSE Linux Enterprise Server (SLES) operating system can implement various server security hardening features.

## 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.

## Server security hardening levels

The server security hardening features are grouped in increasingly more secure levels. Select a level of security appropriate for your organization, and make the changes in that level and any level beneath it. For example, level-3 security requires all changes described in level-1 and level-2 in addition to those described in level-3.

## Level-1 security hardening

Many Level-1 security hardening features are part of the base SUSE Enterprise Linux Server (SLES) operating system on Gen4 and later Avamar Data Stores.

## 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 STIG requirements in the following table.

**Table 24** STIG requirements satisfied by AIDE

| Requirement ID | Requirement title  |
|----------------|--|
| 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/aide` cron job.

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

## The auditd service

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 writes log output in `/var/log/audit/audit.log`.

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 STIG requirements in the following table.

**Table 25** STIG requirements satisfied by the `auditd` service

| Requirement ID | Requirement title   |
|----------------|---|
| 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   |

## 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 STIG requirements in the following table.

**Table 26** STIG requirements satisfied by the implementation of `sudo`

| Requirement ID | Requirement title            |
|----------------|------------------------------|
| 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.

## Locking down single-user mode on RHEL servers

For RHEL servers, limit access in single-user mode to the root user. This task is not required on SLES servers.

**Procedure**

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as `admin`.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the `dpnid` OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Create a backup copy of `/etc/inittab`:

- Single-node server:
 

```
cp -p /etc/inittab /etc/inittab.backup
```

- Multi-node server:  

```
mapall --all --user=root "cp /etc/inittab /etc/inittab.backup"
```
3. Open `/etc/inittab` in a plain text editor.
  4. Add the following entry:  
Change:  

```
# System initialization
si::sysinit:/etc/rc.d/rc.sysinit
```

To:  

```
# System initialization
si::sysinit:/etc/rc.d/rc.sysinit
ss:S:respawn:/sbin/sulogin
```
  5. Close `inittab` and save your changes.
  6. (Multi-node system only) Copy the changes made to `/etc/inittab` to all nodes by typing:  

```
cd /etc
mapall --all --user=root copy inittab
mapall --all --user=root "cp /root/inittab /etc/inittab"
mapall --all --user=root "rm -f /root/inittab"
```

## Disabling Samba

For RHEL servers, and SLES servers with the optional Samba packages installed, disabling Samba prevents the use of Samba commands to obtain valid local and domain usernames and to obtain the Avamar server's browse list. The browse list is a list of the computers nearest to the Avamar server.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the `dpnid` OpenSSH key by typing:  

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Disable Samba:
  - Single-node server:  

```
service smb stop
chkconfig smb off
```
  - Multi-node server:  

```
mapall --all --user=root "service smb stop"
mapall --all --user=root "chkconfig smb off"
```

## Results

Samba is disabled and will not start when the Avamar system boots.

## Web server cipher suite hardening on pre-7.1 Avamar systems

Harden the cipher suite used by the web servers on pre-7.1 Avamar systems to prevent intrusions.

### NOTICE

The tasks listed in this section only apply to pre-7.1 Avamar server versions.

To help prevent security intrusions that take advantage of weaker default cipher suites on pre-7.1 Avamar server versions, complete the following tasks:

- Force strong ciphers for Apache on pre-7.1 Avamar server versions
- Force strong ciphers for Java on pre-7.1 Avamar server versions
- Force strong ciphers for Tomcat on pre-7.1 Avamar server versions
- Configure IE8 to use strong encryption

### Forcing strong ciphers for Apache on pre-7.1 Avamar server versions

Harden the cipher suite used by the Apache HTTP server (Apache) on pre-7.1 Avamar systems.

#### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the dpnid passphrase and press **Enter**.
2. Make a backup copy of the SSL configuration file for Apache.
  - On SLES, make a copy of `/etc/apache2/vhosts.d/vhost-ssl.conf`.
  - On RHEL, make a copy of `/etc/httpd/conf.d/ssl.conf`.
3. Open the SSL configuration file in a plain text editor.
  - On SLES, open `/etc/apache2/vhosts.d/vhost-ssl.conf`.
  - On RHEL, open `/etc/httpd/conf.d/ssl.conf`.
4. In the SSL configuration file, move the line that reads `SSLHonorCipherOrder On` before the line that starts with `SSLCipherSuite`.
5. In the SSL configuration file, replace the existing `SSLCipherSuite` line with the following:

```
SSLCipherSuite DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:AES256-SHA:AES128-SHA:DES-CBC3-SHA
```

6. Save the changes and close the SSL configuration file.
7. Restart the Apache web server by typing:

- On SLES:
 

```
/etc/init.d/apache2 stop
/etc/init.d/apache2 start
```
- On RHEL:
 

```
/etc/init.d/httpd stop
/etc/init.d/httpd start
```

## Forcing strong ciphers for Java on pre-7.1 Avamar server versions

Harden the cipher suite used by the Java on pre-7.1 Avamar systems.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Stop the MCS by typing `dpnctl stop mcs`.
3. Open `/usr/local/avamar/var/mc/server_data/prefs/mcserver.xml` in a plain text editor.
4. Replace the existing `cipher_suite_128` line with the following :

```
<entry key="cipher_suite_128"
value="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SH
A,TLS_DHE_DSS_WITH_AES_128_CBC_SHA"/>
```

5. Save the changes, and close `mcserver.xml`.
6. Start the MCS and the scheduler by typing:

```
dpnctl start mcs
dpnctl start sched
```

## Forcing strong ciphers for Tomcat on pre-7.1 Avamar server versions

Harden the cipher suite used by the Apache Tomcat (Tomcat) servers on pre-7.1 Avamar systems.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Stop the Tomcat components by typing

```
emwebapp.sh --stop
```

3. Open `/usr/local/avamar-tomcat/conf/server.xml` in a plain text editor.
4. Find and change the cipher suite in the SSL connector, as follows:

```
<Connector SSLEnabled="true" Server="Avamar" ciphers="
TLS_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_D
HE_DSS_WITH_AES_128_CBC_SHA" clientAuth="false"
maxKeepAliveRequests="1" maxThreads="150" port="8543"
protocol="org.apache.coyote.http11.Http11NioProtocol"
scheme="https" secure="true" sslProtocol="TLS"/>
```

5. Save your changes, and close `server.xml`.
6. Start the Tomcat components by typing `emwebapp.sh --start`.
7. Return the active account to admin by typing `su admin`.
8. Stop the EMS web components by typing `emwebapp.sh --stop`.
9. Open `/usr/local/avamar/var/mc/server_data/prefs/emserver.xml` in a plain text editor.
10. Find and change the cipher suite in the `cipher_suite_128` setting, as follows:

```
<entry key="cipher_suite_128"
value="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SH
A,TLS_DHE_DSS_WITH_AES_128_CBC_SHA " />
```

11. Save the changes, and close `emserver.xml`.
12. Start the EMS web components by typing `emwebapp.sh --start`.

## Configuring IE8 to use strong encryption

After changes are made to prohibit SSL 2.0 connections, Internet Explorer 8 (IE8) must be configured to use only SSL 3.0 protocols, and the TLS 1.0, 1.1, and 1.2 protocols. This change can be accomplished by pushing out a new domain group policy or by manually changing the setting in each web browser.

This task describes how to manually change IE8 to use strong encryption.

### Procedure

1. Start IE8.
2. On the menu bar, click **Tools** > **Internet Options**.
3. Click the **Advanced** tab and complete the following settings:
  - Clear **Use SSL 2.0**.
  - Select **Use SSL 3.0**.
  - Select **Use TLS 1.0**.
  - Select **Use TLS 1.1**.
  - Select **Use TLS 1.2**.

Click **OK**.

## Web server cipher suite hardening on Avamar server version 7.1

Harden the cipher suite used by the web servers on an Avamar version 7.1 system to prevent Browser Exploit Against SSL/TLS (BEAST) attacks and other intrusions.

### NOTICE

Installing or upgrading to Avamar server version 7.1.1 and newer automatically installs hardened cipher suites for the system's web servers. The tasks listed in this section are not required for Avamar server version 7.1.1 and newer systems.

To help prevent security intrusions that take advantage of weaker default cipher suites on Avamar server version 7.1, complete the following tasks:

- Force strong ciphers for Apache on Avamar server version 7.1
- Force strong ciphers for Tomcat on Avamar server version 7.1

### Forcing strong ciphers for Apache on Avamar server version 7.1

Modify the cipher suite that is used by the Apache HTTP server (Apache) to force the use of strong ciphers on Avamar server version 7.1 systems.

#### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

c. When prompted, type the dpnid passphrase and press **Enter**.

2. Make a backup copy of the SSL configuration file for Apache.

- On SLES, make a copy of `/etc/apache2/vhosts.d/vhost-ssl.conf`.
- On RHEL, make a copy of `/etc/httpd/conf.d/ssl.conf`.

3. Open the SSL configuration file in a plain text editor.

- On SLES, open `/etc/apache2/vhosts.d/vhost-ssl.conf`.
- On RHEL, open `/etc/httpd/conf.d/ssl.conf`.

4. In the SSL configuration file, move the line that reads `SSLHonorCipherOrder` On before the line that starts with `SSLCipherSuite`.

5. In the SSL configuration file, replace the existing `SSLCipherSuite` line with the following:

```
SSLCipherSuite DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:AES256-SHA:AES128-SHA:DES-CBC3-SHA
```

6. Save the changes and close the SSL configuration file.

- Restart the Apache web server by typing:

```
service apache2 restart
```

## Forcing strong ciphers for Tomcat on Avamar server version 7.1

Modify the cipher suite that is used by the Apache Tomcat (Tomcat) servers to force the use of strong ciphers on Avamar server version 7.1 systems.

### Procedure

- Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

- Make a backup copy of `/usr/local/avamar-tomcat/conf/server.xml`.
- Open `/usr/local/avamar-tomcat/conf/server.xml` in a plain text editor.
- Change the `ciphers` line to the following:

```
ciphers="TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384"
```

- Save the changes and close the file.
- Restart the Tomcat server:

```
dpnctl stop mcs
dpnctl start mcs
emwebapp.sh -stop
emwebapp.sh -start
```

- Make a backup copy of `/usr/local/avamar-dtlt-tomcat/conf/server.xml`.
- Open `/usr/local/avamar-dtlt-tomcat/conf/server.xml` in a plain text editor.
- Change the `ciphers` line to the following:

```
ciphers="TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384"
```

```
8_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384"
```

10. Save the changes and close the file.

11. Restart the Desktop/Laptop Tomcat server, by restarting the DTLT service:

```
dpnctl stop dtlt
dpnctl start dtlt
```

## Removing suid bit from non-essential system binaries on RHEL

On RHEL systems, remove the suid bit from non-essential system binaries to prevent them from running with elevated permissions.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Type the following commands:

```
chmod u-s /sbin/pam_timestamp_check
chmod u-s /opt/dell/srvadmin/oma/bin/omcliproxy
chmod u-s /usr/lib64/squid/pam_auth
```

## Preventing unauthorized access to GRUB configuration

Changes to the configuration file of GNU GRUB bootloader (GRUB) can change the startup configuration of the Avamar system. Install an encrypted password to prevent unauthorized changes to this file.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Start the encryption application.
  - On SLES, type `/usr/sbin/grub-md5-crypt`.
  - On RHEL, type `/sbin/grub-md5-crypt`.
3. When prompted, type the GRUB password.  
The MD5 hash of the password appears.
4. Copy and save the MD5 hash.
5. Open `/boot/grub/menu.1st` in a plain text editor.
6. Add the following entry below the `timeout` entry:

```
password --md5 hash
```

where *hash* is the MD5 hash.

7. Close `menu.1st` and save your changes.
8. (Multi-node system only) Push the change to the storage nodes by typing the following commands:
 

```
cd /boot/grub
mapall --all --user=root copy menu.1st
mapall --all --user=root "cp /root/menu.1st /boot/grub/menu.1st"
mapall --all --user=root "rm -f /root/menu.1st"
```

## Level-2 security hardening

Level-2 security hardening features can be installed on a feature-by-feature basis.

All level-2 security hardening features can be installed on Avamar 6.0 and later servers running supported versions of SLES.

Additional password and firewall hardening can be installed on supported versions of Red Hat Enterprise Linux (RHEL).

---

### Note

Beginning with Avamar 7.1, software installation and any upgrade to 7.1 automatically installs, by default, the operating system hardening and firewall hardening packages. This does not restrict supported server functionality, and therefore, those packages cannot be uninstalled. In some cases, like future upgrades, EMC Support personnel will provide steps and tools to enable necessary procedures (for instance, FTP capabilities for downloading packages to the server).

---

## 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, or manually after server software installation. This package satisfies the STIG requirements in the following table.

**Table 27** STIG requirements satisfied by the additional OS hardening package

| Requirement ID | Requirement title                                       |
|----------------|---|
| 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 and RHEL operating systems 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 without a login

---

#### Note

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, or manually after server software installation.

Additional password hardening satisfies the STIG requirements in the following table.

**Table 28** STIG requirements satisfied by additional password hardening

| Requirement ID | Requirement title                   |
|----------------|-------------------------------------|
| 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                 |

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

- Account lockout
- Password aging
- Password complexity, length, and reuse

#### Account lockout

All local Avamar server operating system accounts must log in at least once every 35 days.

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

---

**Note**

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 three of the following four character sets:
  - Two or more lowercase characters.
  - Two or more uppercase characters.
  - Two or more numeric characters.
  - Two or more special (non-alphanumeric) characters.
- Minimum length is determined by complexity:
  - If you use any three character sets, the password must be at least 14 characters.
  - If you use all four character sets, the password must be at least 11 characters.
- Passwords must contain at least three 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 and RHEL operating systems 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, or manually after server software installation.

Additional server firewall hardening satisfies the GEN006580 - Access Control Program STIG requirement.

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

The output for `avfirewall` is logged to `/var/log/firewall` on SLES servers only. 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`.

---

**Note**

If you are backing up a Hyper-V or Microsoft SQL plug-in to a server running the `avfirewall` service and the encryption method for the backup is set to None, the backup will fail with errors indicating a problem connecting to the server. Set the encryption method to Medium or High.

---

## Installing level-2 security hardening features

Level-2 security hardening features can be installed during Avamar server software installation. The *Avamar SLES Installation Workflow Guide* provides information about installing and enabling security hardening features. This guide is available during installation when you click the help icon in Avamar Installation Manager. If you did not install level-2 security hardening features during Avamar server software installation, you can manually install them after server software installation is complete.

### Manually installing level-2 hardening packages on SLES

---

#### Note

This topic is applicable to Avamar 7.0.x and earlier servers.

---

#### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Change directory to where the install packages reside by typing:

```
cd /usr/local/avamar/src/SLES11_64/
```

3. If installing on a multi-node server, copy one or more level-2 hardening packages to all other server nodes by typing the following commands:

```
mapall --all+ --user=root copy avhardening-version.x86_64.rpm
mapall --all+ --user=root copy avpasswd-version.x86_64.rpm
```

where *version* is the specific version you are installing.

If you are not installing a particular level-2 hardening feature, omit the command to copy that install package.

4. Install the hardening packages by doing one of the following:

- If installing on a single-node server, type:
 

```
rpm -Uvh avhardening-version.x86_64.rpm
rpm -Uvh avpasswd-version.x86_64.rpm
rpm -Uvh avfwb-version.x86_64.rpm
```

- If installing on a multi-node server, type:

```
mapall --all+ --user=root "rpm -Uvh avhardening-
version.x86_64.rpm"
```

```
mapall --all+ --user=root "rpm -Uvh avpasswd-version.x86_64.rpm"
mapall --all+ --user=root "rpm -Uvh avfwb-version.x86_64.rpm"
```

where *version* is the specific version you are installing.

If you are not installing a particular level-2 hardening feature, omit the command to copy that install package.

5. If installing on a multi-node server, delete the install packages by typing:

```
mapall --user=root "rm -f avhardening*"
mapall --user=root "rm -f avpasswd*"
mapall --user=root "rm -f avfwb*"
```

If you did not copy a particular install package, omit the command to delete that package.

## Manually installing level-2 hardening packages on RHEL

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Change directory to where the install packages reside by typing:

```
cd /usr/local/avamar/src/RHEL4_64/
```

3. If installing on a multi-node server, copy one or more level-2 hardening packages to all other server nodes by typing:

```
mapall --all+ --user=root copy avpasswd-version.x86_64.rpm
mapall --all+ --user=root copy avfwb-version.x86_64.rpm
```

where *version* is the specific version you are installing.

4. Install the hardening packages by doing one of the following:

- If installing on a single-node server, type:

```
rpm -Uvh avpasswd-version.x86_64.rpm
rpm -Uvh avfwb-version.x86_64.rpm
```

- If installing on a multi-node server, type:

```
mapall --all+ --user=root "rpm -Uvh avpasswd-version.x86_64.rpm"
mapall --all+ --user=root "rpm -Uvh avfwb-version.x86_64.rpm"
```

where *version* is the specific version you are installing.

If you are not installing a particular level-2 hardening feature, omit the command to copy that install package.

5. If installing on a multi-node server, delete the install packages by typing:

```
mapall --user=root "rm -f avpasswd*"
mapall --user=root "rm -f avfwb*"
```

If you did not copy a particular install package, omit the command to delete that package.

## Configuring replication for level-2 firewall hardening

Implementing level-2 firewall hardening can cause replication to fail unless TLS encryption is enabled on the destination server.

### Configuring policy-based replication for level-2 firewall hardening

Installing the level-2 firewall hardening package might cause policy-based replication to fail. If this occurs, enable TLS encryption on the destination server by including the `--dstencrypt=tls` option with each `avrepl` command.

The *EMC Avamar Administration Guide* provides additional information about policy-based replication and the `avrepl` command.

### Configuring cron-based replication for level-2 firewall hardening

Installing the level-2 firewall hardening package might cause cron-based replication to fail. If this occurs, modify the `repl_cron.cfg` file to enable TLS encryption on the destination server.

#### Procedure

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

```
--dstencrypt=tls
```
4. Close `repl_cron.cfg` and save your changes.

## Configuring the firewall to support Avamar Client Manager

Installing the level-2 firewall hardening package will prevent Avamar Client Manager from managing clients associated with the firewall protected server. To remedy this, you must define a range of IP addresses that are allowed to access server data port 5555.

#### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Open `/etc/firewall.base` in a plain text editor.
3. Define one or more IP addresses as `M_SUBNET` by adding the following entry:
 

```
M_SUBNET=IP-address-range
```

 where *IP-address-range* is range of IP addresses that are allowed to access port 5555, in one of the following formats:
  - A single IP address. For example:  
`192.25.113.29`
  - A comma-separated list of IP addresses. For example:  
`192.25.113.29,192.25.113.50`
  - A CIDR notation address range. For example:  
`192.25.113.0/24`
4. Beneath entry defining the value of `M_SUBNET`, add the following if/then statement to allow the IP addresses defined by `M_SUBNET` to have access to port 5555.
 

```
if [ $THISNODE == "$UTILITY" ]; then  
$IPT -A INPUT -p tcp -m multiport --dport 5555 -s $M_SUBNET -j  
ACCEPT  
fi
```
5. Close `firewall.base` and save your changes.
6. Restart the `avfirewall` service by typing the following commands:
 

```
service avfirewall stop  
service avfirewall start
```
7. (Multi-node servers only) Perform the following steps on each server storage node:
  - a. Open a command shell and log in to the storage node as `admin`.
  - b. Switch user to root by typing `su -`.
  - c. Modifying `/etc/firewall.base` as described previously.
  - d. Restart the `avfirewall` service as described previously.

## Level-3 security hardening

Level-3 security hardening disables all web-based services and reduces other services to the minimum required to manage and use the Avamar system.

Level-3 security hardening features can be applied to a running, fully functional Avamar server

---

### Note

Level-1 and level-2 security hardening must be completely implemented prior to implementing level-3 security hardening.

---

## Disabling Apache web server

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Turn off the Apache web server by typing `website stop`.
  3. Disable the Apache web server by typing `chkconfig apache2 off`.

### Results

The Apache web server is disabled and will not automatically run when the Avamar server is restarted.

## Stopping the EMS

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Stop the EMS by typing `dpnctl stop ems`.

### Results

Although the EMS is stopped, it will restart when the server is restarted. Repeat this task each time the Avamar server is restarted.

## Disabling Dell OpenManage web server

Disabling the web server for Dell OpenManage prevents web browser access to that service. The Dell OpenManage services remain available at the console.

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Stop the Dell OpenManage web server.
  - On multi-node servers, type:  
`mapall --all+ --user=root "service dsm_om_connsvc stop"`
  - On single-node servers, type:  
`service dsm_om_connsvc stop`
3. Disable the Dell OpenManage web server.
  - On multi-node servers, type:  
`mapall --all+ --user=root "chkconfig dsm_om_connsvc off"`
  - On single-node servers, type:  
`chkconfig dsm_om_connsvc off`
4. (Optional) Verify that the Dell OpenManage web server is not running.
  - On multi-node servers, type:  
`mapall --all+ --user=root "chkconfig dsm_om_connsvc --list"`
  - On single-node servers, type:  
`chkconfig dsm_om_connsvc -list`

## Stopping Avamar Desktop/Laptop

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Stop Avamar Desktop/Laptop by typing `dpnctl stop dtlt`.

### Results

Although Avamar Desktop/Laptop is stopped, it will restart when the server is restarted. Repeat this task each time the Avamar server is restarted.

## Disabling SSLv2 and weak ciphers

Configure the Avamar server to disallow the use of SSL v.2 and weak ciphers in communication between server nodes and backup clients.

---

### Note

Enforcing the use of strong ciphers prevents clients that do not support strong ciphers from connecting with Avamar server. For example, clients running any of the following OS versions that do not support strong ciphers are blocked by this configuration: Microsoft Windows NT, Microsoft Windows 2000, and Microsoft Windows 2003 (without strong cipher patches).

---

## Configuring Avamar 7.1 servers to use strong ciphers

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.

- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
  - c. When prompted, type the dpnid passphrase and press **Enter**.

2. Type the following command:

```
avmaint config sslciphers=level --ava
```

where *level* is the Avamar cipher level in the following table.

**Table 29** Cipher levels and associated OpenSSL suites

| Avamar cipher level | OpenSSL suites   |
|---------------------|--|
| cleartext           | NULL-SHA   |
| insecure            | ALL:NULL-SHA   |
| low                 | EDH-DSS-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA   |
| legacy              | EDH-DSS-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA:AES128-SHA:AES256-SHA   |
| medium              | ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA:AECDH-AES128-SHA |
| high                | ECDHE-ECDSA-AES256-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-GCM-SHA384:AES256-SHA  |

3. Repeat these steps on each server node.

## Configuring Avamar 6.1 through 7.0 servers to use strong ciphers

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

c. When prompted, type the dpnid passphrase and press **Enter**.

2. Type the following command:

```
avmaint config --ava sslciphers='TLSv1+HIGH:!SSLv2:!aNULL:!eNULL:@STRENGTH'
```

3. Repeat these steps on each server node.

## Configuring Avamar 6.0.x servers to use strong ciphers

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:
 

```
ssh-agent bash
ssh-add ~admin/.ssh/dpnid
```
    - c. When prompted, type the dpnid passphrase and press **Enter**.
2. Open `/usr/local/avamar/etc/stunnel/stunnel.conf` in a plain text editor.
3. Add the following entries:

Change:

```
foreground = no
client = no
cert = /usr/local/avamar/etc/stunnel/stunnel.pem
pid = /usr/local/avamar/var/stunnel.pid
[axionssl]
accept = 29000
connect = 27000
```

To:

```
foreground = no
client = no
cert = /usr/local/avamar/etc/stunnel/stunnel.pem
pid = /usr/local/avamar/var/stunnel.pid
options = NO_SSLv2
ciphers = ALL:+HIGH:!LOW:!EXP
[axionssl]
accept = 29000
connect = 27000
```

4. Close `stunnel.conf` and save your changes.
5. Stop and start stunnel by typing:

```
stunctl stop
stunctl start
```

## Configuring the NDMP accelerator to use strong ciphers

### Procedure

1. Open a command shell and log in to the accelerator as admin.
2. Switch user to root by typing `su -`.

3. Open `/usr/local/avamar/var/avtar.cmd` in a plain text editor.
4. Add the following entries:
 

```
--encrypt=tls
--encrypt-strength=high
```
5. Close `avtar.cmd` and save your changes.

## Configuring replication to use strong ciphers

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server, log in to the server as admin.
  - For a multi-node server, log in to the utility node as admin.
2. Switch user to root by typing `su -`.
3. Open `/usr/local/avamar/etc/repl_cron.cfg` in a plain text editor.
4. Add the following entries:
 

```
--avtar=--encrypt=tls
--avtar=--encrypt:1=tls
--dstavmgr=--encrypt=tls
--dstavmaint=--encrypt=tls
--encrypt-strength=high
```
5. Close `repl_cron.cfg` and save your changes.

## Updating OpenSSH

### Before you begin

Contact your EMC Customer Support professional to obtain and install the latest Avamar platform security rollup package. The platform security rollup package installs the latest version of OpenSSH.

Updating to the latest version of OpenSSH and performing this task configures OpenSSH to:

- Deny empty passwords
- Log at INFO level
- Use protocol 2
- Harden for security audit vulnerabilities

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as admin.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as admin.
    - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Open `/etc/ssh/sshd_config` in a plain text editor.
3. Add the following entries:

```
PermitEmptyPasswords no
LogLevel INFO
Protocol 2
Ciphers cipher_suite
```

where *cipher\_suite* is one of the following:

- For SLES:
 

```
aes128-ctr, aes192-ctr, aes256-ctr, arcfour256, arcfour
```
- For RHEL:
 

```
arcfour, aes128-ctr, aes192-ctr, aes256-ctr
```

4. Close `sshd_config` and save your changes.
5. Restart the `sshd` service by typing `service sshd restart`.
 

Restarting the `sshd` service can cause current SSH sessions to terminate.

## Disabling SNMP

### Procedure

1. Open a command shell and log in by using one of the following methods:
  - For a single-node server:
    - a. Log in to the server as `admin`.
    - b. Switch user to root by typing `su -`.
  - For a multi-node server:
    - a. Log in to the utility node as `admin`.
    - b. Load the `dpnid` OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.
2. Stop the `snmpd` service by typing `service snmpd stop`.
3. Disable the `snmpd` service on startup by typing `chkconfig snmpd off`.
4. Open `/etc/init.d/dataeng` in a plain text editor.
5. Edit the following entry:

Change:

```
OS_SNMP_SVCNAME="snmpd"
```

To:

```
OS_SNMP_SVCNAME=""
```

6. Close `dataeng` and save your changes.

7. Reboot the server by typing `reboot`.
8. (Optional) After the system is up, search `/var/log/messages` for the following warning:

```
dataeng: warning: not started. must be started to manage
this system using SNMP
```

This warning means that `snmpd` is disabled.

## Disabling RPC

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as `admin`.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the `dpnid` OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Stop the RPC service.
  - On SLES, type `service rpcbind stop`.
  - On RHEL, type `service portmap stop`.
3. Disable the RPC service at startup.

- On SLES, type:
 

```
chkconfig nfs off
chkconfig rpcbind off
```
- On RHEL, type `chkconfig portmap off`.

4. Repeat these steps on each server node.

## Configuring the firewall to block access to port 9443

Avamar Management Console Web Services normally use Port 9443 for Java Remote Method Invocation (RMI). Configure iptables to block port 9443.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as `admin`.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.

- b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Open `/etc/firewall.default` in a plain text editor.

3. Add the following entries:

```
$IPT -A INPUT -p tcp -m tcp --dport 9443 -j DROP
$IPT -A INPUT -p udp -m udp --dport 9443 -j DROP
```

4. Close `firewall.default` and save your changes.

5. Restart the `avfirewall` service by typing the following commands:

```
service avfirewall stop
service avfirewall start
```

## Changing file permissions

Use the `chmod o-w` command to prevent users in the Others group from writing to specific folders and files.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as `admin`.
  - b. Switch user to `root` by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Type the following commands:

```
chmod o-w -R /etc/openldap
chmod o-w -R /root/
chmod o-w /data01/avamar/var
chmod o-w /data01/avamar/var/change-passwords.log
chmod o-w /data01/avamar/var/local
chmod o-w /data01/avamar/var/local/ziptemp
chmod o-w /data01/avamar/var/p_*dat
chmod o-w /opt/dell/srvadmin/iws/config/keystore.db.bak
chmod o-w /tmp/replicate
chmod o-w /usr/local/avamar/bin/benchmark
chmod o-w /.avamardata/var/mc/cli_data/prefs/mcclimcs.xml
chmod o-w /.avamardata/var/mc/cli_data/prefs/mccli_logging.properties
chmod o-w /.avamardata/var/mc/cli_data/prefs/prefs.tmp
chmod o-w /.avamardata/var/mc/cli_data/prefs/mccli.xml
```

```

chmod o-w /data01/home/admin/.avamardata/var/mc/cli_data/prefs/
mccli.xml
chmod o-w /data01/home/admin/.avamardata/var/mc/cli_data/prefs/
mcclimcs.xml
chmod o-w /data01/home/admin/.avamardata/var/mc/cli_data/prefs/
mccli_logging.properties
chmod o-w /data01/home/admin/.avamardata/var/mc/cli_data/prefs/
prefs.tmp
chmod o-w /data01/home/dpn/.avamardata/var/mc/cli_data/prefs/
mccli.xml
chmod o-w /data01/home/dpn/.avamardata/var/mc/cli_data/prefs/
mcclimcs.xml
chmod o-w /data01/home/dpn/.avamardata/var/mc/cli_data/prefs/
mccli_logging.properties
chmod o-w /data01/home/dpn/.avamardata/var/mc/cli_data/prefs/
prefs.tmp
chmod o-w /data01/avamar/var/mc/server_log/mcddrsnmp.out

```

## Preparing for a system upgrade

To permit a successful system upgrade, some of the level-3 security hardening changes must be temporarily reversed. After the system upgrade is complete, reapply those changes.

### Enabling the Apache web server

#### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

```

ssh-agent bash
ssh-add ~admin/.ssh/dpnid

```

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Enable the Apache web server by typing the following command:

```
chkconfig --add apache2
```

3. Start the Apache web server by typing the following command:

```
website start
```

### Starting the EMS

#### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server, log in to the server as admin.

- For a multi-node server, log in to the utility node as admin.
2. (Single-node server only) Enable the EMS by typing `dpnctl enable ems`.
  3. Start the EMS by typing `dpnctl start ems`.



# APPENDIX A

## Port Requirements

This appendix includes the following topics:

- [Terminology](#)..... 126
- [Avamar firewall](#)..... 127
- [Utility node ports](#).....127
- [Storage node ports](#).....134
- [Avamar client ports](#)..... 135
- [Avamar Downloader Service host ports](#)..... 137
- [Ports when using a Data Domain system](#)..... 138

# Terminology

This appendix uses specific terms to refer to network concepts that concern Avamar systems.

The following terms are used in this appendix.

**Source**

Computer that originates a network transmission. The source computer transmits network packets through a network interface, over a network connection, and to a specific port on a target computer.

**Target**

Computer that receives a network transmission. The target computer receives transmitted network packets on the port that the source computer specified. A service on the target computer that is listening on the specified port processes the packets. Processing may include a response sent to the source computer or the establishment of two-way communication with the source computer.

**Inbound**

Direction of travel of network packets that are sent from another computer to a referenced Avamar computer. The referenced Avamar computer is the target and the other computer is the source. The referenced Avamar computer receives inbound network packets on an inbound port. The inbound port is a port on the referenced Avamar computer with a specific service for receiving and handling those network packets. The inbound port is also known as a listening port.

**Outbound**

Direction of travel of network packets that an Avamar computer sends to a destination computer. The referenced Avamar computer is the source and the other computer is the target. The outbound port is the port on which the other computer listens for the transmissions from the referenced Avamar computer.

**Required ports**

Inbound and outbound ports that must be open to allow the Avamar system to perform its core functions. Relevant routers, switches, and firewalls must allow the network packets to reach these required ports. Core functionality is reduced when a process listening on a required target port cannot receive packets from a source computer.

---

**Note**

When an Avamar server undergoes security hardening some of the required ports are intentionally closed. Security hardening provides an increase in security in exchange for a loss of some functionality.

---

**Optional ports**

Inbound and outbound ports that are used by the Avamar system to provide additional functionality. Closing these ports reduces or eliminates the additional functionality but does not prevent the Avamar system from performing its core functions.

## Avamar firewall

The Avamar firewall daemon runs on every Avamar node. The Avamar firewall daemon controls access to all inbound ports on each node and controls transmissions sent from each node.

The Avamar firewall daemon is called `avfirewall`. When a change is made to a firewall rule, restart `avfirewall` to load the new configuration.

The Avamar firewall daemon uses the rules in `/etc/firewall.base`. Use the symlink: `/ect/firewall.default` to access the rules file.

## Controlling the firewall daemon

Stop, start, restart, and check the status of the Avamar firewall daemon.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as `admin`.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as `admin`.
  - b. Load the `dpnid` OpenSSH key by typing:

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

- c. When prompted, type the `dpnid` passphrase and press **Enter**.

2. Stop the firewall daemon by typing:

```
service avfirewall stop
```

3. Start the firewall daemon by typing:

```
service avfirewall start
```

4. Restart the firewall daemon by typing:

```
service avfirewall restart
```

5. Check the status of the firewall daemon by typing:

```
service avfirewall status
```

## Utility node ports

The Avamar utility node has specific port requirements both for inbound and outbound ports.

The tables in this section list the following port requirements for the utility node:

- Required inbound ports

Ports on the utility node that must be open to network transmissions from specified source computers.

- Optional inbound ports  
Ports on the utility node that can be optionally opened to network transmissions from specified source computers to enable a specific feature.
- Required outbound ports  
Ports on another computer that the utility node must be allowed to access.

## Utility node required inbound ports

The following table describes the inbound ports that must be open on an Avamar utility node. For every port listed in this table, the Avamar utility node is the destination and the source is listed in the Source computer column.

**Table 30** Required inbound ports on the utility node

| Port | Protocol | Service name               | Source computer  | Additional information  |
|------|----------|----------------------------|--|---|
| 22   | TCP      | SSH                        | <ul style="list-style-type: none"> <li>• Administrator computers</li> <li>• Other Avamar server nodes</li> </ul>   | Secure shell access.  |
| 69   | TCP      | TFTP                       | Internal switch  |   |
| 80   | TCP      | HTTP                       | <ul style="list-style-type: none"> <li>• Web browser clients</li> <li>• Reverse proxy web server</li> <li>• AvInstaller</li> <li>• Avamar Downloader Service host</li> </ul>                               | Provides web browser access to Avamar services. A reverse proxy web server can be used to limit access to this port.                            |
| 123  | TCP/UDP  | NTP                        | NTP time servers   | Provides clock synchronization from network time protocol servers.  |
| 137  | UDP      | NETBIOS Name Service       | Avamar proxy   | Used for Avamar proxy communication.  |
| 138  | UDP      | NETBIOS Datagram Service   | Avamar proxy   | Used for Avamar proxy communication.  |
| 139  | TCP      | NETBIOS Session Service    | Avamar proxy   | Used for Avamar proxy communication.  |
| 161  | TCP      | SNMP                       | Data Domain system   | This is the getter/setter port for SNMP objects from a Data Domain system. Required when storing Avamar client backups on a Data Domain system. |
| 443  | TCP      | HTTP protocol over TLS/SSL | <ul style="list-style-type: none"> <li>• Web browser clients</li> <li>• Reverse proxy web server</li> <li>• AvInstaller</li> <li>• Avamar Downloader Service host</li> <li>• Avamar Key Manager</li> </ul> | Provides web browsers with HTTPS access to Avamar services. A reverse proxy web server can be used to limit access to this port.                |
| 700  | TCP/UDP  | Login Manager              | <ul style="list-style-type: none"> <li>• Web browser clients</li> <li>• Reverse proxy web server</li> </ul>  |   |
| 703  | TCP      | AKM service                | Avamar server nodes  | Used for key management.  |

**Table 30** Required inbound ports on the utility node (continued)

| Port | Protocol | Service name                      | Source computer   | Additional information  |
|------|----------|-----------------------------------|---|---|
| 1080 | TCP      | 3ware RAID management             | Web browser clients   | All nodes with legacy Axion-M or Axion-E hardware only. Only allow access from trusted administrator computers.   |
| 1234 | TCP      | Avamar installation utility HTTPS | Web browser clients   | Only open this port for installation of the Avamar software. Only permit access from trusted administrator computers used during software installation.<br><br>1. Notice: Close this port when installation of the Avamar software is complete. Avamar services do not listen on port 1234. |
| 2888 | TCP      | AVDTO                             | Avamar Extended Retention Media Access Node   | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 5555 | TCP      | PostgreSQL administrator server   | <ul style="list-style-type: none"> <li>Utility node running Avamar Client Manager</li> <li>PostgreSQL administrator client computers</li> </ul>   | Only open this port if you manage the Avamar server using Avamar Client Manager or if you must manage the PostgreSQL database from a remote computer. Limit access to trusted administrator computers.  |
| 5568 | TCP      | PostgreSQL                        | Avamar Extended Retention Media Access Node   | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 5671 | TCP      | RabbitMQ                          | <ul style="list-style-type: none"> <li>localhost</li> <li>Other Avamar utility nodes</li> <li>Avamar Extended Retention computers</li> <li>EMC Backup and Recovery Manager computers</li> </ul> | RabbitMQ is a message broker used to enhance asynchronous interprocess communication.   |
| 6667 | TCP      | Archive Service Event             | Avamar Extended Retention Media Access Node   | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 7000 | TCP      | Apache Tomcat                     | Avamar Extended Retention Media Access Node   | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 7443 | TCP      | Apache Tomcat                     | Avamar Extended Retention Media Access Node   | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 7778 | TCP      | RMI                               | Avamar Administrator management console   | Limit access to trusted administrator computers.  |
| 7779 | TCP      | RMI                               | Avamar Administrator management console   | Limit access to trusted administrator computers.  |
| 7780 | TCP      | RMI                               | Avamar Administrator management console   | Limit access to trusted administrator computers.  |
| 7781 | TCP      | RMI                               | Avamar Administrator management console   | Limit access to trusted administrator computers.  |
| 8105 | TCP      | Apache Tomcat                     | Avamar client computers   | Used by Avamar Desktop/Laptop.  |

**Table 30** Required inbound ports on the utility node (continued)

| Port  | Protocol | Service name                                    | Source computer                    | Additional information   |
|-------|----------|---|------------------------------------|--|
| 8109  | TCP      | Apache Tomcat                                   | Avamar client computers            | Used by Avamar Desktop/Laptop.   |
| 8181  | TCP      | Apache Tomcat                                   | Avamar client computers            | Connections from Avamar client computers and from AvInstaller hosts are redirected to this port.   |
| 8444  | TCP      | Apache Tomcat                                   | Web browser clients                | Web browser connections from Avamar Desktop/Laptop client computers are redirected to this port.   |
| 8505  | TCP      | Apache Tomcat                                   | Utility node or single-node server | Avamar Desktop/Laptop uses this port to send a shutdown command to its Apache Tomcat server. Limit access to the utility node or single-node server.                         |
| 8543  | TCP      | Apache Tomcat HTTPS                             | Web browser clients                | Web browser clients use this port to create HTTPS connections to Avamar Enterprise Manager and Avamar Installation Manager. Limit access to trusted administrator computers. |
| 8580  | TCP      | AvInstaller                                     | Web browser clients                | Used for connections from Avamar Downloader Service computer, and for access to AvInstaller from other web browser clients.  |
| 8778  | TCP      | RMI - Avamar Enterprise Manager                 | Utility node or single-node server | Any utility node that has Avamar Enterprise Manager installed. Limit access to the utility node or single-node server.   |
| 8779  | TCP      | RMI - Avamar Enterprise Manager login_server    | Utility node or single-node server | Any utility node with Avamar Enterprise Manager installed. Limit access to the utility node or single-node server.   |
| 8780  | TCP      | RMI - Avamar Enterprise Manager service_context | Utility node or single-node server | Any utility node with Avamar Enterprise Manager installed. Limit access to the utility node or single-node server.   |
| 8781  | TCP      | RMI - Avamar Enterprise Manager node_context    | Utility node or single-node server | Any utility node with Avamar Enterprise Manager installed. Limit access to the utility node or single-node server.   |
| 9443  | TCP      | RMI - Avamar Management Console web services    | Web browser clients                |  |
| 19000 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 19500 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 20000 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 20500 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 25000 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 25500 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |
| 26000 | TCP/UDP  | GSAN  | Avamar server nodes                | GSAN communication.  |

**Table 30** Required inbound ports on the utility node (continued)

| Port          | Protocol | Service name        | Source computer  | Additional information   |
|---------------|----------|---------------------|--|--|
| 26500         | TCP/UDP  | GSAN                | Avamar server nodes  | GSAN communication.  |
| 27000         | TCP      | Avamar server       | <ul style="list-style-type: none"> <li>Avamar client computers</li> <li>Avamar server nodes</li> <li>Avamar nodes acting as a replicator source</li> </ul> | GSAN communication. This port is blocked by default for new installs of Avamar server 7.1 or newer. Open this port to allow unencrypted backups. |
| 27500         | TCP      | Avamar server       | <ul style="list-style-type: none"> <li>Avamar client computers</li> <li>Avamar server nodes</li> <li>Avamar nodes acting as a replicator source</li> </ul> | GSAN communication.  |
| 28001         | TCP      | Avamar server CLI   | Avamar client computers  | CLI commands from client computers.  |
| 28002 - 28011 | TCP      |                     | Avamar Extended Retention Media Access Node  | The firewall rules open this port when you install support for Avamar Extended Retention.  |
| 28009         | TCP      | avagent             | VMware proxy   | Unsecure communication with VMware proxy.  |
| 29000         | TCP      | Avamar server SSL   | Avamar client computers  | GSAN communication.  |
| 30001         | TCP      | MCS                 | Avamar client computers  | 2-way secure socket  |
| 30003         | TCP      | MCS                 | Avamar client computers  |  |
| 30102 - 30109 | TCP      | avagent             | VMware proxy   | Secure communication with VMware proxy.  |
| 61617         | TCP      | Apache ActiveMQ SSL | Avamar Extended Retention Media Access Node  | The firewall rules open this port when you install support for Avamar Extended Retention.  |

## Utility node optional inbound ports

The following table describes the recommended, but optional, inbound ports for an Avamar utility node. For every port listed in this table, the Avamar utility node is the destination and the source is listed in the Source computer column.

**Table 31** Optional inbound ports on the utility node

| Port | Protocol | Service name | Source computer                         | Additional information   |
|------|----------|--------------|---|--|
| 514  | UDP      | syslog       | Utility node or single-node server      | Avamar server connects to this port to communicate events to syslog.   |
| 5556 | TCP      | PostgreSQL   | PostgreSQL client computer              | Avamar server node running Avamar Enterprise Manager. Limit access to computers that require access to the Avamar Enterprise Manager database. |
| 5557 | TCP      | PostgreSQL   | Avamar Enterprise Manager host computer | Avamar server node with metadata search feature installed. Facilitates metadata search in Avamar Enterprise Manager.                           |

**Table 31** Optional inbound ports on the utility node (continued)

| Port | Protocol | Service name  | Source computer                    | Additional information  |
|------|----------|---------------|------------------------------------|---|
| 8509 | TCP      | Apache Tomcat | Utility node or single-node server | The Apache JServ Protocol (AJP) uses port 8509 to balance the work load for multiple instances of Tomcat. |

## Utility node required outbound ports

The following table describes the outbound ports that must be accessible to network packets that are sent from an Avamar utility node. For each row, the utility node is the source computer that must have outgoing access to the listed port on the listed destination computer.

**Table 32** Required outbound ports for the utility node

| Port | Protocol | Destination computer   | Additional information  |
|------|----------|--|---|
| 7    | TCP      | Data Domain system   | Required to register a Data Domain system for storing Avamar client backups.  |
| 23   | TCP      | Internal   | Required for communication with internal switches and for firmware upgrades.  |
| 25   | TCP      | EMC Customer Support   | Required to allow ConnectEMC to make an SMTP connection with EMC Customer Support.  |
| 53   | TCP/UDP  | DNS  | Required for name resolution and DNS zone transfers. VMware proxy nodes require the TCP connection to DNS.                              |
| 88   |          | Key Distribution Center (KDC)  | Required for access to Kerberos authentication system.  |
| 111  | TCP/UDP  | RPC port mapper service on Data Domain system  | Only required when backups are stored on a Data Domain system. Access to RPC and NFS port mapper functionality on a Data Domain system. |
| 123  | TCP/UDP  | NTP time servers   | Provides synchronization of system time from network time protocol servers.   |
| 163  | TCP      | SNMP service on Data Domain system   | Only required when backups are stored on a Data Domain system.  |
| 389  | TCP/UDP  | LDAP   | Provides access to directory services.  |
| 443  | TCP      | <ul style="list-style-type: none"> <li>VMware vCenter proxy service</li> <li>Avamar Key Manager</li> </ul> |   |
| 464  | TCP      | Key Distribution Center (KDC)  | Required for access to the Kerberos Change/Set password.  |
| 902  | TCP      | VMware ESX server proxy service  |   |
| 2049 | TCP/UDP  | NFS daemon on Data Domain system   | Only required when backups are stored on a Data Domain system.  |
| 2052 | TCP/UDP  | NFS mountd process on Data Domain system   | Only required when backups are stored on a Data Domain system. Outbound communication must be open for both protocols: TCP and UDP.     |
| 5671 | TCP      | <ul style="list-style-type: none"> <li>localhost</li> </ul>  | RabbitMQ messaging. RabbitMQ is a message broker used to enhance asynchronous interprocess communication.                               |

**Table 32** Required outbound ports for the utility node (continued)

|               |         |  |   |
|---------------|---------|--|---|
|               |         | <ul style="list-style-type: none"> <li>• Other Avamar utility nodes</li> <li>• Avamar Extended Retention computers</li> <li>• EMC Backup and Recovery Manager computers</li> </ul> |   |
| 7443          | TCP     | Media Access node that hosts Avamar Extended Retention   | Only required when using the Avamar Extended Retention feature.   |
| 7444          | TCP     | vCenter server   | For utility node configurations that also run the VMware Backup Appliance this port is opened by an if/then clause in the firewall rules. Otherwise, this port is not required. Used to test vCenter credentials.         |
| 8080          | TCP     | NetWorker server   | For utility node configurations that also run the VMware Backup Appliance this port is opened by an if/then clause in the firewall rules. Otherwise, this port is not required. Used to register with a NetWorker server. |
| 8543          | TCP     | AvInstaller service running on other Avamar servers  | Used by the AvInstaller service to control AvInstaller services on other Avamar servers.  |
| 8580          | TCP     | Computer running Avamar Downloader Service   | Used to make requests for package downloads from the Avamar Downloader Service computer.  |
| 9443          | TCP     | Managed Avamar servers   | Avamar Management Console web services use this outbound port for RMI communication via a dynamically assigned port on managed Avamar servers.  |
| 19000         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 19500         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 20000         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 20500         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 25000         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 25500         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 26000         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 26500         | TCP/UDP | Avamar server nodes  | GSAN communication.   |
| 27000         | TCP     | Avamar server nodes  | GSAN communication.   |
| 28002 - 28011 | TCP     | Avamar Extended Retention Media Access Node  | The firewall rules open this port when you install support for Avamar Extended Retention.   |
| 28009         | TCP     | VMware proxy   | Unsecure communication with VMware proxy.   |
| 30002         | TCP     | Avamar client computers  | Communication with avagent.   |
| 30102 - 30109 | TCP     | VMware proxy   | Secure communication with VMware proxy.   |
| 61617         | TCP     | Media Access node that hosts Avamar Extended Retention   | Only required when using the Avamar Extended Retention feature.   |

**Table 32** Required outbound ports for the utility node (continued)

|       |     |   |  |
|-------|-----|---|--|
| 61619 | TCP | Computer running EMC Backup and Recovery Manager. | Required to permit communication with EMC Backup and Recovery Manager. |
|-------|-----|---|--|

## Storage node ports

Avamar storage nodes have specific port requirements both for inbound and outbound ports.

The tables in this section list the following port requirements for storage nodes:

- Required inbound ports  
Ports on each storage node that must be open to network transmissions from specified source computers.
- Required outbound ports  
Ports on another computer that each storage node must be allowed to access.

## Storage node required inbound ports

The following table describes the inbound ports that must be open on each Avamar storage node. For every port listed in this table, the Avamar storage node is the destination and the source is listed in the Source computer column.

**Table 33** Required inbound ports on each storage node

| Port  | Protocol | Service name          | Source  | Additional information  |
|-------|----------|-----------------------|---|---|
| 22    | TCP      | SSH                   | <ul style="list-style-type: none"> <li>• Administrator computers</li> <li>• Other Avamar server node</li> </ul> | Secure shell access.  |
| 123   | TCP/UDP  | NTP                   | <ul style="list-style-type: none"> <li>• NTP time servers</li> <li>• Avamar utility node</li> </ul>             | Permits clock synchronization from network time protocol servers (exochronous) and from the utility node (isochronous). |
| 1080  | TCP      | 3ware RAID management | Web browser clients   | Nodes with legacy Axion-M or Axion-E hardware only. Only allow access from trusted administrator computers.             |
| 19000 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 19500 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 20000 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 20500 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 25000 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 25500 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 26000 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 26500 | TCP/UDP  | GSAN                  | Avamar server nodes   | GSAN communication.   |
| 27000 | TCP      | Avamar server         | <ul style="list-style-type: none"> <li>• Avamar client computers</li> </ul>                                     | GSAN communication. GSAN communication. This port is blocked by default for new installs                                |

**Table 33** Required inbound ports on each storage node (continued)

| Port  | Protocol | Service name      | Source   | Additional information  |
|-------|----------|-------------------|--|---|
|       |          |                   | <ul style="list-style-type: none"> <li>Avamar nodes acting as a replicator source</li> </ul> | of Avamar server 7.1 or newer. Open this port to allow unencrypted backups. |
| 29000 | TCP      | Avamar server SSL | Avamar client computers  | GSAN communication.   |

## Storage node required outbound ports

The following table describes the outbound ports that must be accessible to network packets that are sent from each Avamar storage node. For each row, the storage node is the source computer that must have outgoing access to the listed port on the listed destination computer.

**Table 34** Required outbound ports for each storage node

| Port  | Protocol | Destination                                  | Additional information  |
|-------|----------|--|---|
| 53    | TCP/UDP  | DNS  | Required for name resolution and DNS zone transfers. TCP connection to DNS is required by VMware proxy nodes.           |
| 123   | TCP/UDP  | NTP time servers and the Avamar utility node | Permits clock synchronization from network time protocol servers (exochronous) and from the utility node (isochronous). |
| 703   | TCP      | Utility node                                 | Permits access to the AKM service on the utility node.  |
| 19000 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 19500 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 20000 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 20500 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 25000 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 25500 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 26000 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 26500 | TCP/UDP  | Avamar server nodes                          | GSAN communication.   |
| 27000 | TCP      | Avamar server nodes                          | GSAN communication.   |

## Avamar client ports

Avamar clients have specific port requirements both for inbound and outbound ports.

The tables in this section list the following port requirements for Avamar clients:

- Required inbound ports
 

Ports on an Avamar client that must be open to network transmissions from specified source computers.
- Required outbound ports
 

Ports on another computer that an Avamar client must be allowed to access.

## Avamar client required inbound ports

The following table describes the inbound ports that must be open on an Avamar client. For every port listed in this table, an Avamar client is the destination and the source is listed in the Source computer column.

**Table 35** Required inbound ports on an Avamar client

| Port  | Protocol | Service name | Source              | Additional information                                       |
|-------|----------|--------------|---------------------|--|
| 28002 | TCP      | avagent      | Avamar server       | Provides management functionality from Avamar Administrator. |
| 30001 | TCP      | MCS          | Avamar utility node | 2-way secure socket  |
| 30002 | TCP      | avagent      | Avamar utility node |  |

## Avamar client required outbound ports

The following table describes the outbound ports that must be accessible to network packets that are sent from an Avamar client. For each row, the Avamar client is the source computer that must have outgoing access to the listed port on the listed destination computer.

**Table 36** Required outbound ports for an Avamar client

| Port  | Protocol | Destination                                  | Additional information  |
|-------|----------|--|---|
| 53    | TCP/UDP  | DNS  | Required for name resolution and DNS zone transfers.  |
| 80    | TCP      | Avamar server HTTP service                   | Required to use the web browser UI of Avamar Desktop/Laptop and the web browser UI of Avamar Web Restore. |
| 123   | UDP      | NTP time servers                             | Provides clock synchronization from network time protocol servers.  |
| 443   | TCP      | Avamar server HTTPS service                  | Required to use the web browser UI of Avamar Desktop/Laptop and the web browser UI of Avamar Web Restore. |
| 3008  | TCP      | Active archive service on Data Domain system | Only required when backups are stored on a Data Domain system, and the active archive feature is enabled. |
| 8105  | TCP      | Avamar server                                | Used by Avamar Desktop/Laptop.  |
| 8109  | TCP      | Avamar server                                | Used by Avamar Desktop/Laptop.  |
| 8181  | TCP      | Avamar server HTTP redirect port             | Required to use the web browser UI of Avamar Desktop/Laptop and the web browser UI of Avamar Web Restore. |
| 8444  | TCP      | Avamar server HTTPS redirect port            | Required to use the web browser UI of Avamar Desktop/Laptop and the web browser UI of Avamar Web Restore. |
| 27000 | TCP      | Avamar server                                | GSAN communication.   |
| 27500 | TCP      | Avamar server                                | GSAN communication.   |
| 28001 | TCP      | Avamar server                                | CLI commands from client computers.   |
| 29000 | TCP      | Avamar server                                | GSAN communication.   |
| 30001 | TCP      | Avamar utility node                          | MCS   |
| 30003 | TCP      | Avamar utility node                          | MCS   |

## Avamar Downloader Service host ports

An Avamar Downloader service host has specific port requirements both for inbound and outbound ports.

The tables in this section list the following port requirements for an Avamar Downloader service host:

- Required inbound port  
Port on an Avamar Downloader service host that must be open to network transmissions from specified source computers.
- Required outbound ports  
Ports on another computer that an Avamar Downloader service host must be allowed to access.

### Avamar Downloader Service host required inbound port

The following table describes the inbound port that must be open on an Avamar Downloader Service host. For the port listed in this table, an Avamar Downloader Service host is the destination and the source is listed in the Source computer column.

**Table 37** Required inbound port on an Avamar Downloader Service host

| Port | Protocol | Service name              | Source        | Additional information   |
|------|----------|---------------------------|---------------|--|
| 8580 | TCP      | Avamar Downloader Service | Avamar server | Avamar server connects to this port to access the Avamar Downloader Service. |

## Avamar Downloader Service host required outbound ports

The following table describes the outbound ports that must be accessible to network packets that are sent from an Avamar Downloader Service host. For each row, an Avamar Downloader Service host is the source computer that must have outgoing access to the listed port on the listed destination computer.

**Table 38** Required outbound ports for an Avamar Downloader Service host

| Port | Protocol | Destination                 | Additional information  |
|------|----------|-----------------------------|---|
| 21   | TCP      | EMC FTP server              | Provides the Avamar Downloader Service with FTP access to updates, security rollup packages, hotfixes, and patches provided by EMC. |
| 53   | TCP/UDP  | DNS                         | Required for name resolution and DNS zone transfers.  |
| 80   | TCP      | Avamar server HTTP service  | Provides HTTP access to the AvInstaller service.  |
| 123  | UDP      | NTP time servers            | Provides clock synchronization from network time protocol servers.  |
| 443  | TCP      | Avamar server HTTPS service | Provides HTTPS access to the AvInstaller service.   |

## Ports when using a Data Domain system

An Avamar system that is deployed with a Data Domain system as a storage target has specific port requirements.

In addition to the port requirements described in this section, implement the additional Data Domain system port requirements described in the Knowledgebase article: *Port Requirements for Allowing Access to Data Domain System Through a Firewall*. This article is available from: <https://support.EMC.com>.

### Required ports when using a Data Domain system

The following table describes the general port requirements when an Avamar system is deployed with a Data Domain system as a storage target

**Table 39** Required ports when using a Data Domain system

| Port | Protocol | Source       | Destination        | Service                 | Additional information   |
|------|----------|--------------|--------------------|-------------------------|--|
| 7    | TCP      | Utility node | Data Domain system | ECHO                    | Required to register a Data Domain system for storing Avamar client backups. |
| 111  | TCP/UDP  | Utility node | Data Domain system | RPC port mapper service | Access to RPC and NFS port mapper functionality on a Data Domain system.     |

**Table 39** Required ports when using a Data Domain system (continued)

| Port | Protocol | Source             | Destination        | Service                | Additional information   |
|------|----------|--------------------|--------------------|------------------------|--|
| 161  | TCP      | Data Domain system | Utility node       | SNMP                   | This is the getter/setter port for SNMP objects from a Data Domain system. |
| 163  | TCP      | Utility node       | Data Domain system | SNMP                   | none   |
| 2049 | TCP/UDP  | Utility node       | Data Domain system | NFS daemon             | none   |
| 2052 | TCP/UDP  | Utility node       | Data Domain system | NFS mountd process     | Outbound communication must be open for both protocols: TCP and UDP.       |
| 3008 | TCP      | Avamar client      | Data Domain system | Active archive service | Only required when the active archive feature is enabled.                  |



# APPENDIX B

## Enterprise Authentication

This appendix includes the following topics:

- [Enterprise authentication](#) ..... 142
- [Configuring Enterprise authentication](#) ..... 143

## Enterprise authentication

Enterprise (or external) authentication enables users to use the same user ID and password to log in to multiple systems.

### NOTICE

For backwards compatibility, this appendix preserves information about the deprecated Enterprise authentication method. The functionality of this method is replaced, and improved upon, by the directory service authentication method. Information about the directory service authentication method is available in the EMC Avamar Administration Guide.

The Avamar Enterprise 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

Enterprise authentication is only available for specific Avamar components. Enterprise authentication supports two external authentication systems.

### Avamar components

Avamar Administrator, Avamar Enterprise Manager, and Avamar Web Access support the use of Enterprise authentication for user accounts.

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

- Operating system user accounts: root, admin, and dpn.
- Special Avamar system administrative user accounts, for example MCUser and root.

### External systems

Avamar supports the external authentication systems described in the following table.

**Table 40** Supported external authentication systems

| Category  | Description  |
|---|--|
| Lightweight Directory Access Protocol (LDAP)            | Hierarchical directory structure, X.500-standard, system such as: <ul style="list-style-type: none"> <li>• Microsoft Active Directory Service (MS ADS)</li> <li>• Novell NDS and eDirectory</li> </ul> |
| 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:  |

**Table 40** Supported external authentication systems (continued)

| Category | Description  |
|----------|--|
|          | <ul style="list-style-type: none"> <li>• Master NIS Server - Primary Domain Controller (PDC)</li> <li>• Slave NIS Servers - Backup Domain Controllers (BDC)</li> </ul> |

## Configuring Enterprise authentication

Configuring Enterprise authentication involves the completion of a series of tasks, including configuring either an LDAP or an NIS interface.

Complete the sequence of tasks outlined below to complete Enterprise authentication configuration.

### Procedure

1. Back up the current configuration files.
2. Configure an LDAP or an NIS interface.

Complete the steps described in either [Configuring an LDAP interface on page 143](#) or [Configuring an NIS interface on page 146](#).

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 ability of the external users to log in to Avamar Administrator.

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/domain.cfg` file while configuring the LDAP or NIS interface.

For example: `SueV@example.com`.

- b. User password as used in the external LDAP or NIS system.
  - c. Domain path where external users reside (for example, "/clients/").
5. Back up the configuration files again.

As a best practice, back up configuration files before installing software upgrades to prevent the possibility of configuration files being overwritten with default values.

## Configuring an LDAP interface

Configure an LDAP interface on the Avamar system to use with Enterprise authentication.

### Before you begin

Gather the following information:

- LDAP information: LDAP domain name, IP address or FQDN of LDAP authentication server, and distinguished name (DN) of the account to use for LDAP queries.

- Avamar system information: OS root password, OS admin password, and Avamar system admin password.

### Procedure

1. Open a command shell and log in by using one of the following methods:

- For a single-node server:
  - a. Log in to the server as admin.
  - b. Switch user to root by typing `su -`.
- For a multi-node server:
  - a. Log in to the utility node as admin.
  - b. Load the dpnid OpenSSH key by typing:

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

- c. When prompted, type the dpnid passphrase and press **Enter**.

2. Open `/etc/avamar/domain.cfg` in a plain 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 LDAP domain used for addressing PAM.
- ID is an integer larger than 1. IDs 0 and 1 are reserved for Avamar internal use.

---

### Note

The next step creates a symbolic link for this entry. However, the Avamar system provides an existing symbolic link when you uncomment the line:

```
ldap=3
```

If you use `ldap=3`, skip the next step.

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 can help clarify which LDAP domain is used for external authentication.

---

4. 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 used in the previous step.

5. Log in to the server as admin.

6. Load the admin OpenSSH key by typing:

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

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

8. 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.

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

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

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

```
chown root:root /etc/ldap.conf
```

```
chmod 0600 /etc/ldap.conf
```

11. 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`

12. In a UNIX text editor, open `/etc/ldap.conf`.

13. 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.

14. Restart Login Manager by typing:

```
service lm restart
```

15. Confirm that configuration changes were accepted by typing:

```
avmgr lstd
```

All domains used in Avamar authentication are listed.

16. Confirm that the LDAP server can be queried by typing the following command:

```
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 appears.

For example:

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

### After you finish

Confirm the ability to log in to Avamar Administrator as an external user.

## Configuring an NIS interface

Configure an NIS interface on the Avamar system to use with Enterprise authentication.

### Procedure

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

The next step creates a symbolic link for this entry. However, the Avamar system provides an existing symbolic link when you uncomment the line:

```
nis=2
```

If you use `nis=2`, skip the next step.

The *DOMAIN* part of the entry (either `nis` or a unique NIS domain) appears in the Avamar Administrator Authentication System list. Entering a unique *DOMAIN* can help clarify which NIS domain is used for external authentication.

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

```
ls -sf /etc/pamd/lm_nis /etc/pam.d/lm_NUMBER
```

 where *NUMBER* is the NIS domain ID used in the previous step.
  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.
  6. Confirm the symbolic link by typing:
 

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

where `lm_NUMBER` is the file created earlier.

The following information appears in the command shell:

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

7. In a UNIX text editor, open `lm_NUMBER`.

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
```

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 confi --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.

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.
- *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 ypbinding on
```

20. Confirm the previous settings by typing:

```
/sbin/chkconfig --list ypbinding
```

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 the 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 password | grep USER-ID
```

where:

- *NISDOMAIN* is the NIS domain.
- *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.

### After you finish

Confirm the ability to log in to Avamar Administrator as an external user.

# APPENDIX C

## IAO Information

US Department of Defense (DoD) Security Technical Implementation Guide (STIG) for UNIX mandates information that should be disclosed to an Information Assurance Officer (IAO).

This appendix includes the following topics:

- [System-level accounts](#) .....150
- [Files with SUID bit and SGID bit](#)..... 150

## System-level accounts

Pursuant to the disclosure requirements of STIG compliance rule GEN000360, the following lists contains the names of accounts that are system-level, and are not privileged-user-level:

```
at
mysql
admin
dnsmasq
messagebus
polkituser
suse-ncc
uuuid
wwwrun
stunnel
```

## Files with SUID bit and SGID bit

Pursuant to the disclosure requirements of STIG compliance rule GEN002440, the following list contains the pathnames for files that have the set user ID (SUID) bit and the set group ID (SGID) attributes enabled:

```
/data01/connectemc/archive
/data01/connectemc/failed
/data01/connectemc/history
/data01/connectemc/logs
/data01/connectemc/output
/data01/connectemc/poll
/data01/connectemc/queue
/data01/connectemc/recycle
/lib64/dbus-1/dbus-daemon-launch-helper
/opt/dell/srvadmin/oma/bin/omcliproxy
/usr/bin/lockfile
/usr/bin/slocate
/usr/bin/ssh-agent
/usr/bin/vlock
/usr/bin/wall
/usr/bin/write
/usr/lib/PolicyKit/polkit-explicit-grant-helper
/usr/lib/PolicyKit/polkit-grant-helper
/usr/lib/PolicyKit/polkit-grant-helper-pam
/usr/lib/PolicyKit/polkit-read-auth-helper
/usr/lib/PolicyKit/polkit-revoke-helper
/usr/lib/PolicyKit/polkit-set-default-helper
/usr/lib/vte/gnome-pty-helper
/usr/sbin/lockdev
/usr/sbin/postdrop
/usr/sbin/postqueue
```

```
/usr/sbin/sendmail.sendmail  
/usr/sbin/utempter  
/usr/sbin/zypp-refresh-wrapper
```

