EMC® Avamar®

Version 7.3

Product Security Guide

302-002-859

REV 02

EMC²

Published February 2018

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

CONTENTS

FIGURES

TABLES

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Some versions of the software or hardware currently in use do not support every function that this document describes. The product release notes provide the most up-to-date information on product features.

If a product does not function correctly or does not function as described in this document contact an EMC technical support professional.

Note

This document was accurate at publication time. Go to EMC Online Support (https://support.EMC.com) to find 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
01	April, 2016	GA release of Avamar 7.3
02	January, 2018	Remote management interface port information added

Related documentation

The following EMC publications provide additional information:

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

The following other publications also provide 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 to alert the reader to particular information.

NOTICE

The Notice convention emphasizes important information about the current topic.

Note

The Note convention addresses specific information that is related to the current topic.

Typographical conventions

In this document, EMC uses the typographical conventions that are shown in the following table.

Table 2 Typographical conventions

Convention	Example	Description
Bold typeface	Click **More Options**.	Use for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what a user specifically selects or clicks).
Italic typeface	*EMC Avamar Administration Guide*	Use for full titles of publications that are referenced in text.

Monospace font	`Event Type = INFORMATION`
	`Event Severity = OK`
	`Event Summary = New group created`
		Use for:
		• System code
		• System output, such as an error message or script
		• Pathnames, file names, prompts, and syntax
		• Commands and options

Monospace font with italic typeface	Type ***Avamar_server***, where *Avamar_server* is the DNS name or IP address of the Avamar server.	Use for variables.
Monospace font with bold typeface	Type **yes**.	Use for user input.
Square brackets	**[--domain=*String*(/)] --name=*String***	Square brackets enclose optional values.
Vertical bar	**[--domain=*String*(/)] | --name=*String***	Vertical bar indicates alternate selections - the bar means “or”.

Table 2 Typographical conventions (continued)

Convention	Example	Description
Braces	`{[--domain=String(/)] | --name=String}`	Braces enclose content that the user must specify.
Ellipses	`valid hfs ...`	Ellipses indicate nonessential information that is 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 upper 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 to supplement the information in 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 **Search** 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 **Search**.

Online communities
Go to 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 an EMC sales representative for details about obtaining a valid support agreement or with questions about an 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**.

Enhancing 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 sends configuration, capacity, and general system information to EMC Customer Support.

Comments and suggestions
Comments and suggestions help EMC to continue to improve the accuracy, organization, and overall quality of the user publications. Send comments and suggestions about this document to 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 to help address documentation issues

CHAPTER 1

Introduction

This chapter includes the following topics:

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 that the operating system's company (Red Hat or SUSE) released since the previous Avamar periodic security update. The updates also include relevant kernel-level and OS-level security patches and changes.

The periodic updates are cumulative. Install each periodic update that is issued for the Avamar system in order of release, starting with the first periodic update issued after the release of the 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 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 Installation 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 security features

Installation of Avamar software, or upgrade to Avamar server version 7.1 or newer, installs hardening and firewall packages that improve security capabilities on the Avamar server. These packages cannot be uninstalled.

Installation of the hardening package will not restrict supported server functionality. Installation of the firewall package will prevent unencrypted backups from running. If you are upgrading from versions prior to 7.1 and our scheduled backups are unencrypted, following instructions in Permitting unencrypted data-in-flight on page 68 to enable unencrypted backups. For some other tasks, EMC Support provides the steps and tools that are required to complete the task (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

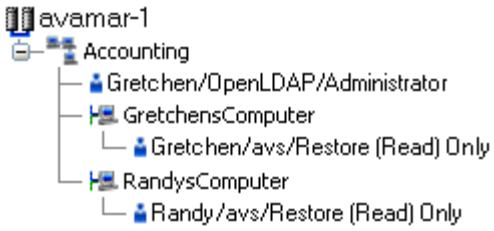
A user account in Avamar can administer a domain or client. The user account defines the authentication system that is used to grant users access to the Avamar server. It also defines the role for the user, which controls the operations that a user can perform.

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 an 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 a computer. However, the authentication system and role are 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

Use directory service authentication to authenticate and assign roles to Avamar users by using information from an existing directory service. Directory service authentication works with specific LDAP directory services and provides additional functionality when used with an OpenLDAP directory service. Directory service authentication also works with a Network Information Service (NIS), on its own or with one of the supported LDAP directory services.

Avamar products that use directory service authentication

The following Avamar products can use directory service authentication to authenticate and authorize users:

- Avamar Administrator
- Avamar Web Restore
- Avamar client web UI (Avamar Desktop/Laptop)

Avamar product that uses directory service client records

Avamar Client Manager does not use directory service authentication to authenticate and authorize user logins. However, Avamar Client Manager can use the directory service mechanism to obtain information about computers that are potential Avamar clients. Avamar Client Manager queries the directory service to obtain information about clients and, if available, directory service organizational units, such as directory domains, and directory groups.

Directory services types

Directory service authentication supports the following types of directory services:

Table 4 Supported directory service types

Type	Supported implementations
LDAP	• Active Directory for Windows Server 2003 • Active Directory Domain Services for Windows Server 2008

Table 4 Supported directory service types (continued)

Type	Supported implementations
	• Active Directory Domain Services for Windows Server 2012 • 389 Directory Server version 1.1.35
OpenLDAP	• SUSE OpenLDAP version 2.4
NIS	• Network Information Service

LDAP maps

Directory service authentication uses LDAP maps to form a group of Avamar domain users by using information from a directory service. Create LDAP maps to link Avamar authorization levels to mapped directory service user accounts. The Adding an LDAP map section provides more information.

NOTICE

Deleting an Avamar domain removes the LDAP maps that rely on that Avamar domain for access. However, removing LDAP maps does not affect the directory service groups or the directory service user records that are associated with the removed maps.

How Avamar authenticates users and assigns roles

To provide backward 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 try:

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 try 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 typed in the **Password** field.
   - Avamar domain, as typed 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 that is 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 responsible for maintaining the system.

You can only assign the role of administrator to user accounts at a domain level. Domain level includes the top-level (root) domain and any other domain or subdomain. You cannot assign the administrator 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 5 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 that are described in this guide. Administrators typically can only view or operate on objects in the domain. Any activity that would allow a domain administrator to view data outside the domain is disallowed. Access to server features of a global nature (for example, suspending or resuming scheduled operations or changing runtimes for maintenance activities) is disallowed. Domain administrators: • Cannot add or edit other subdomain administrators. • Cannot change their assigned role. • Can change their password.

Operator roles

Operator roles are generally implemented to allow 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 6 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. 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. Restore only operators can restore backup data and monitor activities in the assigned domain. • By default, restore only operators cannot perform restores to a different location or restores to multiple locations. To enable this, you must set the restore_admin_can_direct_restores attribute to true in the mcserver.xml file. • 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 operator, add the `noticketrequired` privilege by using the `avmgr chgv` command:**`avmgr chgv --acnt=`*`location`*** **`--u=`*`name`*** **`--ud=`*`auth`*** **`\ --pv="enabled,read,mclogin,noticketrequired"`** where *location* is the subdomain of the operator, *name* is the Avamar username of the user, and *auth* is the external authentication system used to authenticate the user.
Back up only operator	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. 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. Back up only operators can perform on-demand backups of a client or a group, as well as monitor activities in the assigned domain. • By default, back up only operators cannot perform restores to a different location or restores to multiple locations. To enable this, you must set the restore_admin_can_direct_restores attribute to true in the mcserver.xml file. • 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 `noticketrequired` privilege by using the `avmgr chgv` **`avmgr chgv --acnt=`*`location`*** **`--u=`*`name`*** **`--ud=`*`auth`*** **`\ --pv="enabled,read,mclogin,backup,noticketrequired"`**command: where *location* is the subdomain of the operator, *name* is the Avamar username of the user, and *auth* is the external authentication system used to authenticate the user.
Back up/restore operator	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. 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. Back up/restore operators can perform the following tasks in the assigned domain:

Table 6 Operator roles (continued)

Operator type	Description
	• Perform on-demand backups for a client or group. • Perform restores. • Monitor activities. By default, back up/restore operators cannot browse backups from the command line or by using the Avamar Web Restore interface, and cannot perform backups from the command line. To enable these activities, add the `noticketrequired` privilege by using the `avmgr chgv` command: **`avmgr chgv --acnt=`*`location`*` --u=`*`name`*` --ud=`*`auth`*` \ --pv="enabled,read,mclogin,backup,noticketrequired"`** where *location* is the subdomain of the operator, *name* is the Avamar username of the user, and *auth* is the external authentication system used to authenticate the user.
Activity operator	Activity operators are generally only allowed to monitor backup and restore activities and to create certain reports. 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. Activity operators can perform the following tasks in the assigned domain: • Monitor activities. • View the group status summary. • View the Activity Report. • View the Replication Report.

User roles

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

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

The following table describes the four user roles.

Table 7 User roles

User type	Description
Back Up Only User	Users assigned this role can start backups directly from the client by using the `avtar` command line.
Restore (Read) Only User	Users assigned this role can start restores directly from the client by using the `avtar` command line or MCS web services.
Back Up/Restore User	Users assigned this role can start backups and restores directly from the client by using the `avtar` command line or MCS web services.
Restore (Read) Only/ Ignore File Permissions	Similar to the Restore (Read) Only User role except that operating system file permissions are ignored during restores. This user is allowed to restore any file that is stored for an Avamar client. This role is only available when users are authenticated by using Avamar internal authentication. To ensure trouble-free restores, Windows client user accounts should be assigned this role only when both of the following are true: • Users are authenticated using Avamar internal authentication.

Table 7 User roles (continued)

User type	Description
	• Users do not require access to the Avamar client web UI.

Default user accounts

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

Table 8 Avamar server Linux OS default user accounts

User account	Default password	Description
root	changeme	Linux OS root account on all Avamar nodes. **Note** The use of ssh to the root user is allowed: • Internally on all nodes (via localhost) • From the utility node to itself and to all data nodes.
admin	changeme	Linux OS account for Avamar administrative user.
dpn	changeme	Linux OS account for Avamar maintenance user.

Table 9 Avamar server software default user account

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

Table 10 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.

Table 10 MCS default user accounts (continued)

User account	Default password	Description
backuprestore	backuprestore1	Account for internal use by the MCS.
repluser	9RttoTriz	Account for internal use by the MCS for replication.

Table 11 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 12 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.

Changing server passwords and OpenSSH keys

Use the `change-passwords` utility to change the passwords for operating system user accounts and Avamar server user accounts. Also use `change-passwords` to create and modify SSH keys for those accounts.

The `change-passwords` utility guides you through the following operations:

- Changing passwords for the operating system accounts: admin, dpn, and root
- Changing passwords for the internal Avamar server accounts: root, MCUser, repluser, and viewuser
- Creating and changing SSH keys

Procedure

1. Suspend all scheduled operations:
   a. In Avamar Administrator, select **Tools** > **Manage Schedules**.
   b. On the **Manage All Schedules** window, click **Suspend All**.
2. Open a command shell and log in using one of the following methods:
   - To log in to a single-node server, log in to the server as root.
   - To log in to a multi-node server, log in to the utility node as root.
3. Start the utility by typing **`change-passwords`**.

   On a multi-node server, the output prompts you to specify whether to change passwords on all nodes or selected nodes.

4. Type `y` to change passwords on all nodes or `n` to change passwords on selected nodes, and then press **Enter**.

   The output prompts you to indicate whether you plan to specify SSH private keys that are authorized for root operations.

5. Type `n` and press **Enter**.

   The output prompts you to specify whether to change admin, dpn, or root operating system user account passwords.

6. Type `y` to change the passwords or `n` to skip the process of changing the passwords, and then press **Enter**.

7. If you typed `y` in the previous step, then follow the system prompts to change the passwords for one or more of the admin, dpn, or root operating system user accounts.

   The output prompts you to specify whether to change SSH keys.

8. Type `y` to change or create an SSH key, or type `n`, and then press **Enter**.

9. If you typed `y` in the previous step, then follow the system prompts to change or create the keys.

   The output prompts you to specify whether to change Avamar server passwords.

10. When prompted, type `y` to change the MCUser, Avamar root, repluser, and viewuser passwords, or if you do not want to change the passwords, type `n`, and then press **Enter**.

11. If you typed `y` in the previous step, then follow the system prompts to change the passwords.

    The output prompts you to accept or reject the changes that are made to passwords or SSH keys during this utility session.

12. Type `y` to accept the changes or type `n` to exit this utility session without changes, and then press **Enter**.

    The output provides the status of the operation.

13. When the operation completes, resume scheduled operations:

    a. In Avamar Administrator, select **Tools** > **Manage Schedules**.

    b. On the **Manage All Schedules** window, click **Resume All**.

CHAPTER 3

Client/Server Access and Authentication

This chapter includes the following topics:

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).

Session security features

Avamar session security features are provided by the Avamar installation, Avamar Virtual Edition configuration, and upgrade workflow packages as well as a standalone session security workflow.

Session security features include security improvements for communications between Avamar system processes.

The Avamar system secures all communications between Avamar system processes by using session tickets. A valid session ticket is required before an Avamar system process accepts a transmission from another Avamar system process.

The session tickets have the following general characteristics:

- The session ticket is encrypted and signed to protect against modification
- The session ticket is valid for a very short time
- Each session ticket contains a unique signature and is assigned to only one Avamar system process

- The integrity of a session ticket is protected by encryption
- Each Avamar system node separately verifies the session ticket signature
- When required, a session can be extended beyond the life of the session ticket

Avamar server authentication

After installing the session security features, the Avamar system acts as a private certification authority and generates a unique server certificate for the Avamar system.

The Avamar system installs the public key for the server certificate on every Avamar client that is registered with the Avamar server. Avamar clients use the public key to authenticate transmissions from the Avamar system.

For clients that are currently registered, the public key for the server certificate and other required certificate files are propagated to the client within an hour of the installation.

The Avamar system also automatically shares the Avamar server certificate with the Avamar storage nodes. Sharing the certificate allows the utility node and the storage nodes to provide the same certificate for authentication.

Avamar client authentication

Enable client authentication when installing the session security features to have the Avamar system act as a private certification authority and generate a unique client certificate for each Avamar client.

A client certificate is generated when the Avamar server registers an Avamar client.

After generating a client certificate, the Avamar system uses an encrypted connection with the Avamar client to install the certificate on the client. The Avamar system also stores the public key for the client certificate. The public key is used to authenticate the client in all subsequent communications.

Improved security for communications between Avamar system processes

Session security features are provided by the several workflow packages available in the 7.3 Avamar product, including installation, upgrade, and standalone session security workflows.

The security features include:

- Generation and propagation of certificates
- Authentication that is based on X.509 v3 certificates
- Certificate expiration

Installing the session security features

Session security can be implemented and configured during installation of Avamar 7.3 software, configuration of Avamar Virtual Edition 7.3 software, and upgrade of a previous version of Avamar software to 7.3. Session security also can be implemented post-installation or post-upgrade.

Install the session security features by running one of four workflows, whichever is appropriate to the Avamar system, including:

- Avamar Software Installation workflow
- Avamar Virtual Edition Configuration workflow

- Avamar Upgrade workflow
- Session Security Configuration workflow

Use the workflow's Security Settings tab to configure session security features. See the help file associated with each workflow in the Avamar Installation Manager for additional information about each option. In the Security Settings tab, you can:

- Select the type of communication desired between the Management Server and Avamar client agents.
- Select the type of communication desired between the Avamar clients and Avamar server.
- Select the authentication type to use between the server and client when communication is initiated:
  - Single - the client authenticates the server
  - Dual - both client and server authenticate each other
- Create and propagate server certificates on the Avamar server and data nodes which are used for server or client authentication (or both). This is done using the CA certificate installed in the keystore.
- Set a timeframe for the generated server certificates to expire.
- Run the mcrootca all command, which generates all new certificates for root, TLS, and EC root. This forces the creation of new server certificates

Note

If you wish to generate all new certificates for root, TLS, and EC root on an Avamar system, run the Session Security Configuration workflow and use the last option (Generate All New Certificates) on the Security Settings tab. Refer to the workflow's help file for complete instructions on the use of the workflow.

Requirements

Avamar session security features should not be used in an environment that includes unsupported operating systems, clients, plug-ins, or devices. Installing session security features will stop communication with Avamar processes on the unsupported operating systems, clients, plug-ins, and devices.

The Avamar software versions requirements for the session security features are described in the following table.

Software	Minimum version
Avamar server	Avamar 7.1 Service Pack 1 on SUSE Linux Enterprise Server only
Avamar client	Avamar 7.1 Service Pack 1

Prepare multiple Avamar clients for the session security features by pushing Avamar client upgrades out with Avamar Client Manager. Individual Avamar clients can be prepared by downloading and running the Avamar client software installer.

The following table describes the port requirements for the session security features.

Port/Protocol	Source	Destination	Description
29000/TCP	Utility node	Storage node	GSAN using SSL
29000/TCP	Storage node	Utility node	GSAN using SSL
30001/TCP	Utility node	Storage node	MCS using SSL

Port/Protocol	Source	Destination	Description
30001/TCP	Storage node	Utility node	MCS using SSL
30002/TCP	Avamar system	Avamar client	Avamar client using SSL
30002/TCP	Avamar client	Avamar system	Avamar client using SSL
30003/TCP	Utility node	Storage node	MCS using SSL
30003/TCP	Storage node	Utility node	MCS using SSL

The Avamar session security features are subject to some limitations.

- Server operating system
  Session security features cannot be used with Avamar server running on the Red Hat Enterprise Linux operating system.
- Clients
  Session security features cannot be used with any of the following Avamar clients:
  - Avamar Client for HP-UX, on HP-UX 11i
  - Avamar Client for Solaris, on Solaris 11 x64
  - Avamar Client for Solaris, on Solaris 11 SPARC
  - Avamar Cluster Client for Solaris on Veritas Cluster Server
  - Avamar Client for Solaris in Solaris Clusters
  - Avamar Cluster Client for Windows Server 2008
- Plug-ins
  Session security features cannot be used with Avamar Plug-in for the Windows VSS (system state only)
- Devices
  Session security features cannot be used with the Avamar NDMP Accelerator device.
- Other products
  The use of NTP time synchronization of the Avamar server, Avamar clients, and the Data Domain system (if applicable) is strongly encouraged. If the time is not synchronized, it could result in registration and backup/restore failure based on certificate validity and expiration times. Changing the timezone on a host may have a similar impact and may require regeneration of certifications.

Generation and propagation of certificates

Session security-enabling workflow packages enable automatic generation and propagation of certificates.

The Avamar system acts as a private certification authority and generates the certificates that permit the authentication and encryption of communications between Avamar system processes, including processes running on:

- The Avamar utility node
- The Avamar storage nodes
- Avamar clients

The Avamar system also securely propagates the certificates and the public keys to the required locations on each involved computer.

Generating new certificates with Data Domain systems

After generating new certificates on the Avamar server, the following steps are required for Data Domain systems that are configured for Avamar backup storage. Session tickets are supported with Data Domain systems at release 5.6 or greater.

Procedure

1. Wait for the Data Domain server to be aware of the updated certificate.

   The Data Domain server will display a yellow status in the Avamar Administrator with the status message "Unable to retrieve ssh key file pair." This may take up to 30 minutes.

2. Open the Data Doman server in the Avamar Administrator:- In Avamar MCGUI, go to Server > Server Management, select the DD server, click on Edit Data Domain System icon and click on OK in the pop-up window

   a. In Avamar Administrator, click the **Server** launcher button.

      The **Server** window appears.

   b. Click the **Server Management** tab.

   c. Select the Data Domain system to edit.

   d. Select **Actions** > **Edit Data Domain System**.

      The **Edit Data Domain System** dialog box appears.

   e. Click **OK**.

      There is no need to make any changes to the Data Domain configuration.

3. Restart DDBoost on the Data Domain system:

   a. Log into the Data Domain System.

   b. Enter the following commands in the Data Domain CLI:

      ```
      ddboost disable

      ddboost enable
      ```

Results

If multiple Avamar systems are attached to a single Data Domain system and one of those Avamar systems is detached from the system, ddboost should be disabled and enabled again to ensure that backups from the otehr Avamar systems will be successful.

Authentication based on X.509 v3 certificates

The Avamar session security features use X.509 v3 certificates.

The X.509 v3 certificates used with the Avamar session security features have the following default characteristics:

- Key type: RSA
- Key length: 3072 bits
- Cryptographic hash function and digest: SHA256

Certificate expiration

To enhance security, the Avamar session security features include the regular expiration of certificates.

The following table describes the default expiration periods and regeneration methods for the certificates used by the Avamar session security features.

Certificate type	Default expiration period	Description
Root authentication keys	Five years	New certificates generated by running the workflow package to modify the session security features
Session ticket signing key	One month	New key generated automatically on a monthly cycle
Client certificates	Five years	New certificate generated by manually reregistering the client.

Network configuration changes

The Avamar session security features require changes to some network configuration tasks that are performed after installation.

For a session security-enabled Avamar system, the following actions require additional network configuration tasks:

- Changing the IP address or hostname of the Avamar server
- Replacing the utility node
- Replacing a storage node
- Adding a storage node

Those additional network configuration tasks are described in the following resources:

- *EMC Avamar Post-installation Network Configuration Avamar 5.x, 6.x, or 7.x Server Software* technical note, part number 300-015-091
- Procedure documentation that is generated by using the EMC Avamar SolVe Generator (SolVe Generator)

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.

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.

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 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 www.iso.org.
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: **corp-1.example.com**. Example wildcard FQDN for several computers: **corp-*.example.com**.
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.

   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 www.iso.org.
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	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 www.iso.org.
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: **corp-1.example.com**. Example wildcard FQDN for several computers: **corp-*.example.com**.
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
`chain.pem` exists	Append the information in the root certificate to the existing content of `chain.pem`.
`chain.pem` does not exist	Rename the root certificate to `chain.pem` and copy it to: `C:\Program Files\avs\etc\chain.pem`.

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 www.iso.org.
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: **corp-1.example.com**. Example wildcard FQDN for several computers: **corp-*.example.com**.
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. (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 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 www.iso.org.
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.

Field	Description
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: `corp-1.example.com`. Example wildcard FQDN for several computers: `corp-*.example.com`.
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 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 to the server as admin.
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.

Condition	Action
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.

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:
    a. Log in to the server as admin.

b. Switch user to root by typing `su -`.

c. For a multi-node server, load the rootid OpenSSH key by typing:

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

2. Type one of the following alternative commands.

Key type	Command
Private key without randomness and without a passphrase	`openssl genrsa -out` *server.key* `3072`
Private key with randomness and without a passphrase	`openssl genrsa -rand` *binary-files* `-out` *server.key* `3072`
Private key without randomness and with a passphrase	`openssl genrsa -aes128 -out` *server.key* `3072`
Private key with randomness and with a passphrase	`openssl genrsa -rand` *binary-files* `-aes128 -out` *server.key* `3072`

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:

a. Log in to the server as admin.

b. Switch user to root by typing `su -`.

c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 www.iso.org.
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: **corp-1.example.com**. Example wildcard FQDN for several computers: **corp-*.example.com**.
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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **`su -`**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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
   ```

CHAPTER 4

Data Security and Integrity

This chapter includes the following topics:

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.

Data-in-flight encryption in Avamar version 6.0 through Avamar version 7.0

To provide enhanced security during client/server data transfers, Avamar systems, with Avamar server version 6.0 through Avamar server version 7.0, support two levels of data-in-flight encryption: Medium and High. The exact encryption technology and bit strength that is used for a client/server connection depends on a number of factors, including the client platform and specific server version.

When you store Avamar version 6.0 through Avamar version 7.0 client backups on a Data Domain system, the connection between the Avamar client and the Data Domain system is not encrypted. Those Avamar versions use the Data Domain Distributed Deduplication Bandwidth Optimized OST (DDBOOST) SDK to access the Data Domain system. That SDK does not support data encryption between the client and the Data Domain system.

Avamar data-in-flight encryption

To provide enhanced security during client/server data transfers, Avamar version 7.1 and newer supports six levels of data-in-flight encryption: `cleartext`, `insecure`, `low`, `legacy`, `medium`, and `high`. The exact encryption technology and bit strength that is used for a 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 13 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 server-side only encryption.

The default Avamar cipher level is the `legacy` setting, which matches the behavior of Avamar version 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.

Avamar version 7.1 and newer clients support TLS encryption of the data-in-flight for backups that are stored on a Data Domain version 5.5 or newer system. However,

Avamar client cannot provide encryption of the data-in-flight for backups that are stored on a Data Domain version 5.4 or earlier system.

Unencrypted data-in-flight on new installs of Avamar

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

To prevent disruption of existing backup tasks, upgrading a pre-7.1 version Avamar system to Avamar server version 7.1 or newer does not block unencrypted data-in-flight. Upgrading to Avamar server version 7.1 or newer does not block existing backup policies that include transfer of unencrypted data-in-flight.

However, new installs of Avamar server version 7.1 and newer include firewall settings that block unencrypted data-in-flight. This firewall policy increases data security. To allow unencrypted data-in-flight on new installs of Avamar 7.1 and newer, manually change the firewall settings.

Permitting unencrypted data-in-flight

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

NOTICE

This task reduces the security of data-in-flight. Only perform this task to meet a specific business requirement.

Procedure

1. Open a command shell:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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/
   ```

7. Restart the Avamar firewall service.

- For a single-node server, type: **service avfirewall restart**
- For a multi-node server, type: **mapall --noerror --all+ --user=root 'service avfirewall restart'**

Client/server encryption behavior

Client/server encryption functional behavior in any given circumstance is dependent on a number of factors, including the `mcserver.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 server supports cipher strengths up to 128-bit. You can increase the cipher strength that is used by this server to 256-bit for communications on the following ports:

- Ports 7778 and 7779 for the Management Console Server (MCS)
- 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/mcserver.xml` in a plain text editor.
3. Locate the `rmi_cipher_strength` setting and change it to **high**.

   `rmi_cipher_strength=`**high**
4. Close `mcserver.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
```

Data-at-rest encryption

An Avamar server can be configured to encrypt the data that is stored on it. This configuration 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

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.

Note

Avamar server version 7.1 and newer systems perform encryption using the AES 256 CFB block cipher mode. Avamar server version 7.0 and earlier systems can continue to use another block cipher mode 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. Alternatively, enable it from the utility node command line after you complete an installation or upgrade. Also, use this method to rotate in a new data-at-rest encryption key.

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 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 success by typing:

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

   Information similar to the following appears in the 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.

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. Obtain the certificate of a certification authority (CA) to use when generating and signing a public/private key pair.

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

2. 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`

3. Change the file name of the CA certificate 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 an Avamar system when the internal key managed data-at-rest encryption is already enabled. The installation of Avamar Key Manager 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

Enclose options that include a space character within quotes.

Table 14 Options for `installAKM.sh`

Option	Description
-h --help	Displays command line help.
-i *dpm-server* --ip=*dpm-server*	Identifies the fully qualified domain name (FQDN) or IP address of the RSA Data Protection Manager computer, where *dpm-server* represents the fully qualified domain name, or IP address in dotted-quad format. This option is required.
-k *keyclass* --keyclass=*keyclass*	Provides the RSA Data Protection Manager key class to use, where *keyclass* 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. Providing a key class value is optional. The default key class is `Avamar AES 256 CFB`.
--updatepassword	Replaces the private key certificate password that Avamar Key Manager stores 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 15 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.
Error: *avmaint* is not executable	The `avmaint` utility cannot be started, where *avmaint* is the full path to the expected location. The default location is `/usr/local/avamar/bin/avmaint`.
Error: *akm.xml* is not writable	The `akm.xml` file cannot be opened for write access, where *akm.xml* is the full path to the expected location. The default location is `/usr/local/avamar/etc/akm/akm.xml`.
Error: *hosts* is not readable	The `hosts` file cannot be read, where *hosts* is the full path to the expected location. The default location is `/etc/hosts`.
Error: *mapall* is not executable	The `mapall` utility cannot be started, where *mapall* is the full path to the expected location. The default location is `/usr/local/avamar/bin/mapall`.
Error: *probe.xml* is not readable	The `probe.xml` file cannot be opened for reading, where *probe.xml* is the full path to the expected location. The default location is `/usr/local/avamar/var/probe.xml`.
Error: *akm_appreg.cfg* is not readable	The `akm_appreg.cfg` file cannot be opened for reading, where *akm_appreg.cfg* is the full path to the expected location. The default location is `/usr/local/avamar/bin/akm_appreg.cfg`.
Error: *akm_appreg.cfg.org* is not readable	The `akm_appreg.cfg.org` file cannot be opened for reading, where *akm_appreg.cfg.org* is the full path to the expected location. The default location is `/usr/local/avamar/bin/akm_appreg.cfg.org`.

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
     ```
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 16 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 `installAKM.sh` script.
Single-node Avamar server with Data Domain	Required files are automatically backed up to the Data Domain system through the Checkpoint Backup feature (`cpbackup`).

Table 16 Critical files used by Avamar Key Manager (continued)

Configuration	Backup considerations
Single-node Avamar server without Data Domain	Required files are not automatically backed up. Manually back up all files in `/usr/local/avamar/etc/akm`.

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
```

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 `securedelete` 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
```

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

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

where:

- *user* is the Avamar username.
- *auth* is 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:

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

where:

- *location* is 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 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 `securedelete delb` command. If interrupted, all data will not be securely deleted.

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

```
1 Request succeeded
```

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

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

5. If an error is encountered:
   - Search the knowledge base 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

You can monitor client backup, restore, and validation activity to verify that 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.

Acknowledgment 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 acknowledgment 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 that are based on filtering rules that are 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 selected 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 that users start 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 that is associated with the action
- The ID and role of the user that started the action
- The product and component from which the action was started
- 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 more information about 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 that are created by components of a single-node Avamar system.

Table 17 Component log files on a single-node Avamar system

Component	Pathname
Avamar Administrator	/usr/local/avamar/var/mc/server_log/flush.log /usr/local/avamar/var/mc/server_log/restore.log /usr/local/avamar/var/mc/server_log/mcserver.log.# /usr/local/avamar/var/mc/server_log/mcserver.out /usr/local/avamar/var/mc/server_log/pgsql.log /usr/local/avamar/var/mc/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log /usr/local/avamar/var/mc/server_data/mcs_data_dump.sql
Avamar EM (Server)	/usr/local/avamar/var/em/server_log/flush.log /usr/local/avamar/var/em/server_log/restore.log /usr/local/avamar/var/em/server_log/emserver.log.# /usr/local/avamar/var/em/server_log/emserver.out /usr/local/avamar/var/em/server_log/pgsql.log /usr/local/avamar/var/em/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log /usr/local/avamar/var/em/server_data/emt_data_dump.sql
Maintenance	/usr/local/avamar/var/cron/clean_emdb.log /usr/local/avamar/var/cron/dpn_crontab.log /usr/local/avamar/var/cron/cp.log /usr/local/avamar/var/cron/gc.log /usr/local/avamar/var/cron/hfscheck.log /usr/local/avamar/var/cron/ntpd_keepalive_cron.log /usr/local/avamar/var/cron/ntpd_keepalive_cron.log.# /usr/local/avamar/var/cron/suspend.log
avw_install utility	/usr/local/avamar/var/avw_cleanup.log /usr/local/avamar/var/avw_install.log /usr/local/avamar/var/avw-time.log /usr/local/avamar/var/log/dpnavwinstall-VERSION.log
axion_install utility	/usr/local/avamar/var/axion_install_DATE_TIME.log

Table 17 Component log files on a single-node Avamar system (continued)

Component	Pathname
Avamar File System (AvFS)	/usr/local/avamar/var/axionfs.log
change-passwords utility	/usr/local/avamar/var/change-passwords.log
ddrmaint utility	/usr/local/avamar/var/log/ddrmaint.log
dpnctl utility	/usr/local/avamar/var/log/dpnctl.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutil-version.log /usr/local/avamar/var/log/dpnnetutil.log* /usr/local/avamar/var/log/dpnnetutilbgaux.log /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
permctl utility	/usr/local/avamar/var/log/permctl.log
resite utility	/usr/local/avamar/var/dpnresite-version.log /usr/local/avamar/var/mcspref.log /usr/local/avamar/var/nataddr.log /usr/local/avamar/var/smtphost.log
timedist utility	/usr/local/avamar/var/timedist.log
timesyncmon program	/usr/local/avamar/var/timesysncmon.log
Avamar Replicator	/usr/local/avamar/var/cron/replicate.log
Avamar license server	/usr/local/avamar/var/ascd-PORT.log
Storage server	/data01/cur/err.log /data01/cur/gsan.log

Utility node log files

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

Table 18 Component log files on a utility node

Component	Pathname
Avamar Administrator	/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

Table 18 Component log files on a utility node (continued)

Component	Pathname
	/usr/local/avamar/var/mc/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log /usr/local/avamar/var/mc/server_data/mcs_data_dump.sql
Avamar EM (Server)	/usr/local/avamar/var/em/server_log/flush.log /usr/local/avamar/var/em/server_log/restore.log /usr/local/avamar/var/em/server_log/emserver.log.# /usr/local/avamar/var/em/server_log/emserver.out /usr/local/avamar/var/em/server_log/pgsql.log /usr/local/avamar/var/em/server_data/postgres/data/pg_log/postgresql-DATE_TIME.log /usr/local/avamar/var/em/server_data/emt_data_dump.sql
Maintenance	/usr/local/avamar/var/cron/clean_emdb.log /usr/local/avamar/var/cron/dpn_crontab.log /usr/local/avamar/var/cron/cp.log /usr/local/avamar/var/cron/gc.log /usr/local/avamar/var/cron/hfscheck.log /usr/local/avamar/var/cron/ntpd_keepalive_cron.log /usr/local/avamar/var/cron/ntpd_keepalive_cron.log.# /usr/local/avamar/var/cron/suspend.log
avw_install utility	/usr/local/avamar/var/avw_cleanup.log /usr/local/avamar/var/avw_install.log /usr/local/avamar/var/avw-time.log /usr/local/avamar/var/log/dpnavwinstall-VERSION.log
axion_install utility	/usr/local/avamar/var/axion_install_DATE_TIME.log
Avamar File System (AvFS)	/usr/local/avamar/var/axionfs.log
change-passwords utility	/usr/local/avamar/var/change-passwords.log
ddrmaint utility	/usr/local/avamar/var/log/ddrmaint.log
dpnctl utility	/usr/local/avamar/var/log/dpnctl.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutil-version.log /usr/local/avamar/var/log/dpnnetutil.log* /usr/local/avamar/var/log/dpnnetutilbgaux.log /usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log
permctl utility	/usr/local/avamar/var/log/permctl.log
timedist utility	/usr/local/avamar/var/timedist.log
timesyncmon program	/usr/local/avamar/var/timesyncmon.log

Table 18 Component log files on a utility node (continued)

Component	Pathname
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 that an Avamar storage node creates.

Table 19 Component log files on a storage node

Component	Pathname
Storage server log	/data01/cur/err.log /data01/cur/gsan.log
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log /usr/local/avamar/var/log/dpnnetutilbgaux.log
Maintenance 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 20 Component log file on a spare node

Component	Pathname
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log /usr/local/avamar/var/log/dpnnetutibgaux.log

Avamar NDMP Accelerator log files

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

Table 21 Component log files for the NDMP Accelerator

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

Table 21 Component log files for the NDMP Accelerator (continued)

Component	Pathname
	/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 22 Component log files on an access node

Component	Pathname
dpnnetutil utility	/usr/local/avamar/var/log/dpnnetutilbgaux-stdout-stderr.log /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 23 Component log files on an Avamar Administrator client

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

Backup client log files

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

Table 24 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 C:\Program Files\avs\var\clientlogs\{WORKORDER-ID}.log
Avamar Client for Windows tray applet	C:\Program Files\avs\var\avscc.log

Table 24 Component log files for an Avamar backup client (continued)

Component	Pathname
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

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 25 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 26 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 27 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

Table 27 STIG requirements satisfied by the implementation of sudo (continued)

Requirement ID	Requirement title
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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **`su -`**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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
   _SHA,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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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,TL
   S_DHE_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
    _SHA,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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:

   ```
   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

1. Open a command shell:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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

   ```
   ciphers="TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AE
   S_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_EC
   DSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA3
   84,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_25
   6_CBC_SHA384"
   ```

5. Save the changes and close the file.
6. Restart the Tomcat server:

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

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

   ```
   ciphers="TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AE
   S_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_EC
   DSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA3
   84,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDH_RSA_WITH_AES_25
   6_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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 server version 6.0 and newer systems on supported versions of SLES.

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

Note

Installation of Avamar software, or upgrade to Avamar server version 7.1 or newer, installs hardening and firewall packages that improve security capabilities on the Avamar server. Installation of the hardening package will not restrict supported server functionality. Installation of the firewall package will prevent unencrypted backups from running. If you are upgrading from versions prior to 7.1 and our scheduled backups are unencrypted, following instructions in Permitting unencrypted data-in-flight on page 68 to enable unencrypted backups. For some other tasks, EMC Support provides the steps and tools that are required to complete the task (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 28 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
GEN001560	Home Directories Files Permissions
GEN002420	User Filesystems Not Mounted With NoSUID
GEN002580	Permissive umask Documentation
GEN002680	Audit Logs Accessiblity
GEN002700	Audit Logs Permissions
GEN002960	Cron Utility Accessibility
GEN002980	The cron.allow Permissions
GEN003000	Cron Executes World Writable Programs
GEN003020	Cron Executes Programs in World Writable Directories
GEN003040	Crontabs Ownership
GEN003080	Crontab Files Permissions
GEN003100	Cron and Crontab Directories Permissions
GEN003160	Cron Logging
GEN003180	Cronlog Permissions
GEN003200	cron.deny Permissions
GEN003400	The at Directory Permissions
GEN003520	Core Dump Directory Ownership and 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 29 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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **`su -`**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 a pre-7.1 Avamar server to permit Avamar Client Manager management

Installing the level-2 firewall hardening package on a pre-7.1 Avamar server prevents Avamar Client Manager from managing clients that are associated with the server. To remedy this problem, allow a range of IP addresses to access the server through port 5555.

Specify a limited range of IP addresses that includes the IP addresses of the Avamar system that is running Avamar Client Manager.

Procedure

1. Open a command shell:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.

c. For a multi-node server, load the rootid OpenSSH key by typing:

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

2. Open `/etc/firewall.base` in a plain text editor.
3. Define one or more IP addresses as `M_SUBNET` by typing the following entry after the `BEGIN UTILITY NODE SPECIFIC RULES` comment line and before the `END UTILITY NODE SPECIFIC RULES` comment line:

   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 the entry defining the value of`M_SUBNET`, type 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 the 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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 EMT

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 EM Tomcat server by typing `dpnctl stop emt`.

Results

Although the EMT is stopped, it restarts 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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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`

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 servers to use strong ciphers

Complete this task to enforce the use of strong ciphers on Avamar systems with Avamar server version 7.1 or newer.

Procedure

1. Open a command shell:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

2. Type the following command:

   `avmaint config sslciphers=`*`level`* `--ava`

   where *level* is the Avamar cipher level in the following table.

Table 30 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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **su -**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 EMT

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. Start the EM Tomcat server by typing `dpnctl start emt`.

CHAPTER 7

Intelligent Platform Management Interface

This chapter includes the following topics:

IPMI subsystem security

Avamar system computer hardware can contain manufacturer-specific implementations of the Intelligent Platform Management Interface (IPMI). The IPMI subsystem provides out-of-band management of a computer system. A comprehensive plan to secure an Avamar system includes tasks that secure the IPMI subsystem.

IPMI software interacts with the hardware through the baseboard management controller (BMC). IPMI provides management and monitoring of the computer through a subsystem that is separate from the computer's operating system, CPU, and firmware.

On July 26, 2013 the United States Computer Emergency Response Team (US-CERT) released an alert that is entitled: "Risks of Using the Intelligent Platform Management Interface (IPMI)" (TA13-207A). In the alert US-CERT warns that:

> Attackers can use IPMI to essentially gain physical-level access to the server. An attacker can reboot the system, install a new operating system, or compromise data, bypassing any operating system controls.

To secure the IPMI subsystem of an Avamar system, complete the tasks that are described in the following table.

Table 31 Descriptions of security tasks for the IPMI subsystem

Task	Description
Find all channels with the "802.3 LAN" media type	Channels with the "802.3 LAN" media type provide access to the IPMI subsystem from the LAN. LAN access is a known attack vector.
Disable privileges for Cipher Suite 0	IPMI subsystems provide Cipher Suite 0 as an option that permits unauthenticated access for the designated privilege level. Prevent unauthenticated access for all privilege levels by setting the privilege level of this cipher suite to 0.
Secure anonymous logins	IPMI subsystems reserve the account with user ID 1 for anonymous log in. Secure anonymous logins by: • Disabling the anonymous account for Serial over LAN access • Placing the privileges for the account at the lowest level • Disabling IPMI support for the account
Create strong passwords for each baseboard management controller (BMC) account	Strong passwords reduce the possibility of unauthorized access to the IPMI subsystem.
Isolate the LAN port that is used for BMC management	Limit access to the BMC management LAN port.

Table 31 Descriptions of security tasks for the IPMI subsystem (continued)

Task	Description
Disable remote media redirection	Disable BMC access to remote media. Only allow access to remote media during the time it is required to perform a valid IPMI task.
Disable the keyboard/video/monitor (KVM) functionality of the BMC	Disable the KVM functionality of the BMC. Only allow KVM functionality during the time it is required to perform a valid IPMI task.
Prevent access to the BIOS and POST serial interfaces	Isolate the BIOS and POST serial interfaces within the corporate LAN.
Disable boot from USB and boot from CD/DVD	Prevent the possibility of the computer starting from unauthorized media by changing the computer BIOS settings to prevent boot from USB and boot from CD/DVD.
Redirect all incoming HTTP packets sent to Port 80 to the HTTPS port	Force encryption of all HTTP packets by redirecting HTTP sockets to the HTTPS port.

Finding all LAN channels

Channels with the "802.3 LAN" media type provide access to the IPMI subsystem from the LAN. LAN access is a known attack vector. Find all LAN channels to help manage LAN access to the IPMI subsystem.

Before you begin

Obtain console access to the Avamar system computers.

Procedure

1. At the Avamar system utility node console, log in as root.
2. Type the following command for each channel ID:

   `ipmitool channel info` ***`channel_id`***

   where *channel_id* is each of the following channel ID hexadecimal values: 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, and 0x0D.

   Each time the command is typed, the system displays information for the specified channel ID.
3. Record the value of the **Channel Medium Type** field for each channel ID.

   When the value of the **Channel Medium Type** field is `802.3 LAN` the channel is accessible from the LAN.
4. Repeat these steps for each storage node.

Results

This task creates a record of all IPMI subsystem channels that can be accessed from the LAN.

For example, to determine whether channel with the ID value of 0x01 is accessible from the LAN, type the following command:

ipmitool channel info 0x01

The system returns the following information:

```
Channel 0x1 info:
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 1
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : disabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : disabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
```

Disabling privileges for Cipher Suite 0

IPMI subsystems provide Cipher Suite 0 as an option that permits access without authentication, without integrity checks, and without encryption to ensure confidential communication. Prevent unauthenticated access for all privilege levels by setting the privilege level of this cipher suite to 0.

Before you begin

Find all channels with the "802.3 LAN" media type.

Procedure

1. At the Avamar system utility node console, log in as root.
2. Type the following command:

   ipmitool raw 0x0C 0x02 ***channel_id*** **0x18 0x00 0x00**

   where *channel_id* is a channel ID hexadecimal value for a channel that has the "802.3 LAN" media type.

   For example, for channel 0x01, type:

   ipmitool raw 0x0C 0x02 0x01 0x18 0x00 0x00
   The following response appears:

   ```
   11 00 44 44 44 44 44 44 44 44 44 44
   ```

   The system returns a string of 12 half-bytes. The value of the third half-byte indicates the privilege level that is assigned to Cipher Suite 0. In this example, the value of the third half-byte, 44, indicates that the administrator privilege level is assigned to Cipher Suite 0. Change this value to 40 to disable privileges for Cipher Suite 0.
3. Type the following command:

   ipmitool raw 0x0C 0x01 ***channel_id*** **0x18 0x00 0x40 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44**

where *channel_id* is the channel ID hexadecimal value that is used in the previous step. The value `0x40` in the command represents Cipher Suite 0 with privilege level 0.

4. Type the following command to verify the change:

   ipmitool raw 0x0C 0x02 *channel_id* 0x18 0x00 0x00

   For example, for channel 0x01, type:

   ipmitool raw 0x0C 0x02 0x01 0x18 0x00 0x00
   The following response appears:

   ```
   11 00 40 44 44 44 44 44 44 44 44 44
   ```

   The value of the third half-byte is 40 which means that the Cipher Suite 0 privilege level is set to 0 (no privileges) for the specified channel.
5. Repeat these steps for each channel that has the "802.3 LAN" media type.
6. Repeat these steps for each Avamar storage node.

Results

The IPMI subsystem prohibits unauthenticated LAN access.

Securing anonymous logins

IPMI subsystems reserve the account with user ID 1 for anonymous log in. Secure anonymous logins by disabling the anonymous account for Serial over LAN access, placing the privileges for the account at the lowest level, and disabling IPMI support for the account.

Before you begin

Find all channels with the "802.3 LAN" media type.

Procedure

1. At the Avamar system utility node console, log in as root.
2. Type the following command:

   ipmitool sol payload disable *channel_id* 1

   where *channel_id* is a channel ID hexadecimal value for a channel that has the "802.3 LAN" media type.

   The ipmitool disables anonymous user logins through Serial over LAN for the specified channel.
3. Repeat the previous step for each channel that has the "802.3 LAN" media type.
4. Type the following command:

   ipmitool channel setaccess *channel_id* 1 callin=off ipmi=off link=off privilege=1

   where *channel_id* is a channel ID hexadecimal value for a channel that has the "802.3 LAN" media type.

   The ipmitool puts the anonymous user at the lowest privilege level for the specified channel.
5. Repeat the previous step for each channel that has the "802.3 LAN" media type.

6. Type the following command:

   ipmitool user disable 1

   The ipmitool disables support for the BMC anonymous user account.
7. Repeat these steps for each Avamar storage node.

Results

The IPMI subsystem secures anonymous logins.

Creating strong passwords for BMC accounts

Identify the existing baseboard management controller (BMC) accounts and create a strong password for each account. Strong passwords reduce the possibility of unauthorized access to the IPMI subsystem.

Procedure

1. At the Avamar system utility node console, log in as root.
2. Type the following command:

   ipmitool user list

   The system displays a list that has columns of information about each BMC user account.
3. For each user account, type the following command:

   ipmitool user set password *user_ID new_password*

   where *user_ID* is the integer value that is listed in the ID column for the user account and *new_password* is the new strong password for the account.
4. Repeat these steps for each Avamar storage node.

Results

The BMC requires the strong passwords for BMC account access.

For example, type:

ipmitool user list

The following response appears:

```
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv Limit
2   root             false   false      true       ADMINISTRATOR
3   admin            true    true       true       ADMINISTRATOR
```

Change the password for the root account, by typing the following:

ipmitool user set password 2 *new_password*

Change the password for the admin account by typing the following:

ipmitool user set password 3 *new_password*

Additional BMC security tasks

Limit access to the baseboard management controller (BMC) by completing these additional tasks.

Refer to the hardware manufacturer's documentation for information about the additional security tasks that are described in the following sections.

Isolate the BMC management LAN port
The BMC provides a management interface through a dedicated NIC that opens a LAN port on channel 4. Restrict access to this port by the following:

- Never expose the port to internet access
- Never expose the port to access from outside of the corporate LAN
- Assign a static private address to the port
- Only allow access to the port from the subnet

Disable remote media redirection
By default, Avamar systems have remote media redirection disabled. Only enable this BMC feature when it is required.

Disable the KVM functionality
By default, Avamar systems have keyboard/video/monitor (KVM) functionality of the BMC disabled. Only enable this BMC feature when it is required, and only with authentication and strong passwords.

Prevent access to the BIOS and POST serial interfaces
The BMC management port provides BIOS and POST serial interfaces. Do not connect the management port to a device that permits BIOS and POST serial access from outside of the corporate LAN.

Disable boot from USB and boot from CD/DVD
Disable boot from USB and boot from CD/DVD in the BIOS settings of the Avamar system computers to prevent starting the computers from remote media. Do not put the USB interface in the boot path.

Redirect HTTP packets to the HTTPS port
Help secure the BMC management web UI by redirecting traffic sent to the web UI from port 80 (HTTP) to port 443 (HTTPS). Also, improve authentication by configuring the BMC management web UI to use a certification authority-issued trusted public key certificate.

APPENDIX A

Port Requirements

This appendix includes the following topics:

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:
   a. Log in to the server as admin.
   b. Switch user to root by typing **`su -`**.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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
   ```

Editing the Firewall in Avamar

Edit the status of the Avamar firewall.

Before you begin

Firewall edit functionality allows the user to open and close nondependent ports for customized data transfer and to modify associated rules. Rules and ports can be initiated, edited, and terminated through manual configuration of a designated text file, executing those changes, and then restarting the firewall on the Avamar server. Editing the firewall is essentially understanding the content of the config file, editing that content, and then executing those changes.

Procedure

1. Log in to the utility node (or single node server) as root.

   Provide the appropriate password.

2. Change the working directory to the following: `/usr/local/avamar/lib/admin/security`.
3. Open `avfwb_custom_config.txt` in a plain text editor.

   See section below for config file example and how to edit the file.
4. Save and close the file.
5. Run the following command:`manage-custom-rules.sh -execute-rules`.

   This command copies the new firewall rules to all nodes in the system and restarts the firewall.
6. Exit the command session.

The firewall customization lines you add to the `avfwb_custom_config.txt` file must be structured in a pipe-delimited fashion such as the following:

Source IP | Source Port | Destination IP | Destination Port | Protocol | ICMP-type | Target | Chain | Node type

where:

Section	Description
Source IP	Source specification - address can be a network IP address (with /mask) or a plain IP address.
Source Port	Port of origin for traffic.
Destination IP	IP address of destination machine.
Destination Port	Destination port or port range specification.
Protocol	TCP, UDP, or ICMP.
ICMP-type	If ICMP is entered for Protocol, enter the type.
Target	ACCEPT, REJECT, DROP, or LOGDROP.
Chain	INPUT, OUTPUT, or LOGDROP
Node type	ALL (all nodes), DATA (data nodes only), or UTILITY (only applies to the utility node).

If a field does not apply, leave the field blank.

Miscellaneous information

To delete all firewall rules, delete the rules in `avfwb_custom_config.txt` and run `manage-custom-rules.sh --execute-rules` again.

For diagnostic purposes, the log file is located in /var/log/custom-firewall.

To view the current state of the firewall iptable on the utility node or a single node server, run the following command: `iptables -L` (for ipv4) or `ip6tables -L` (for ipv6).

To view the current state of the firewall iptable on all of the nodes of a multi-node server, run the following command: `mapall --all+ --user=root iptables -L`.

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 32 Required inbound ports on the utility node

Port	Protocol	Service name	Source computer	Additional information
22	TCP	SSH	• Administrator computers • Other Avamar server nodes	Secure shell access.
69	TCP	TFTP	Internal switch	
80	TCP	HTTP	• Web browser clients • Reverse proxy web server • AvInstaller • Avamar Downloader Service host	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.
161	TCP	SNMP	Data Domain system	Getter/setter port for SNMP objects from a Data Domain system.

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
				Required when storing Avamar client backups on a Data Domain system.
443	TCP	HTTP protocol over TLS/SSL	• Web browser clients • Reverse proxy web server • AvInstaller • Avamar Downloader Service host • Avamar Key Manager	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	• Web browser clients • Reverse proxy web server	
703	TCP	AKM service	Avamar server nodes	Used for key management.
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 that are used during software installation. 1. Notice: Close this port when installation of the Avamar software is complete. Avamar services do not listen on port 1234.

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
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	• Utility node running Avamar Client Manager • PostgreSQL administrator client computers	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	• localhost • Other Avamar utility nodes • Avamar Extended Retention computers • EMC Backup and Recovery Manager computers	RabbitMQ is a message broker who is 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.

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
7543	HTTP/SSL	Update Manager	Web browser clients	Web browser clients use this port to create HTTPS connections to Avamar Installation Manager. Limit access to trusted administrator computers.
7544	TCP	Update Manager	Jetty socket clients	Jetty socket clients use this port to send a shutdown signal to its Jetty web server. Limit access to trusted administrator computers.
7580	HTPP	Update Manager	Web browser clients	Used for connections from the Avamar Downloader Service computer, and for access Update Manager from other web browser clients.
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.
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

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
				node or single-node server.
8580	TCP	AvInstaller	Web browser clients	Used for connections from Avamar Downloader Service computer, and for access to AvInstaller from other web browser clients.
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.
26500	TCP/UDP	GSAN	Avamar server nodes	GSAN communication.
27000	TCP	Avamar server	• Avamar client computers • Avamar server nodes • Avamar nodes acting as a replicator source	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	• Avamar server nodes • Avamar nodes acting as a replicator source	GSAN communication.
28001	TCP	• Avamar server CLI • MCS • Avagent	• Avamar client computers • VMware proxy • Replication source • Replication target	• CLI commands from client computers. • (Avamar server version 7.1 and earlier) Avagent to MCS communication.

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
				• (Avamar server version 7.1.1 and newer) Bi-directional communication between avagent and MCS on the replication source Avamar server and the replication destination Avamar server to permit authentication key exchange.
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.
28810-28819	TCP	ddrmaint	localhost	Internal use only for token-based authentication when connecting to Data Domain; only localhost can use it.
29000	TCP	Avamar server SSL	Avamar client computers	GSAN communication.
30001	TCP	MCS	• Avamar client computers • VMware proxy	• 2-way secure socket communication. • Avagent to MCS communication. Avamar server version 7.2 and newer.
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

Table 32 Required inbound ports on the utility node (continued)

Port	Protocol	Service name	Source computer	Additional information
				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 33 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.
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 34 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.

Table 34 Required outbound ports for the utility node (continued)

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	• vSphere API • TCP	• VMware vCenter • Avamar Key Manager	
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	• localhost • Other Avamar utility nodes • Avamar Extended Retention computers	RabbitMQ messaging. RabbitMQ is a message broker used to enhance asynchronous interprocess communication.

Table 34 Required outbound ports for the utility node (continued)

		• EMC Backup and Recovery Manager computers	
7443	TCP	Media Access node that hosts Avamar Extended Retention	Only required when using the Avamar Extended Retention feature.
7444	TCP	VMware vCenter	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.
7543	HTTP/SSL	Update Manager	Web browser clients use this port to create HTTPS connections to Avamar Installation Manager. Limit access to trusted administrator computers.
7544	TCP	Update Manager	Jetty socket clients use this port to send a shutdown signal to its Jetty web server. Limit access to trusted administrator computers.
7580	HTPP	Update Manager	Used for connections from the Avamar Downloader Service computer, and for access Update Manager from other web browser clients.
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.
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.

Table 34 Required outbound ports for the utility node (continued)

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.
28001	TCP	Replication source system and replication target system	Replication on Avamar server version 7.1.1 and newer systems requires bi-directional access between the replication source Avamar server and the replication destination Avamar server to permit authentication key exchange.
28002 - 28009	TCP	VMware proxy	Avagent paging port. Unsecured communication with VMware proxy. Avamar server version 7.0 only.
28009	TCP	VMware proxy	• Avagent paging port. Unsecured communication with VMware proxy. Avamar server version 7.1 only. • MCS access to proxy logs. Avamar server version 7.1 and newer.
28011	TCP	Avamar Extended Retention Media Access Node	The firewall rules open this port when you install support for Avamar Extended Retention.
30002	TCP	Avamar client computers	Communication with avagent.
30102	TCP	VMware proxy	Avagent paging port. Secure communication with VMware proxy. Avamar server version 7.2 and newer.
61617	TCP	Media Access node that hosts Avamar Extended Retention	Only required when using the Avamar Extended Retention feature.

Table 34 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 35 Required inbound ports on each storage node

Port	Protocol	Service name	Source	Additional information
22	TCP	SSH	• Administrator computers • Other Avamar server node	Secure shell access.
123	TCP/UDP	NTP	• NTP time servers • Avamar utility node	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.

Table 35 Required inbound ports on each storage node (continued)

Port	Protocol	Service name	Source	Additional information
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	• Avamar client computers • Avamar nodes acting as a replicator source	GSAN communication. 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.
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 36 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.

Table 36 Required outbound ports for each storage node (continued)

Port	Protocol	Destination	Additional information
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 37 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 38 Required outbound ports for an Avamar client

Port	Protocol	Destination	Additional information
53	TCP/UDP	DNS	Required for name resolution and DNS zone transfers.

Table 38 Required outbound ports for an Avamar client (continued)

Port	Protocol	Destination	Additional information
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.
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 39 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 40 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.

Table 40 Required outbound ports for an Avamar Downloader Service host (continued)

Port	Protocol	Destination	Additional information
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.

Also to the port requirements described in this section, implement the additional Data Domain system port requirements that are 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 41 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.
22	TCP	Utility node	Data Domain system	SSH	Secure shell communication with the Data Domain system.
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.
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.

Table 41 Required ports when using a Data Domain system (continued)

Port	Protocol	Source	Destination	Service	Additional information
3008	TCP	Avamar client	Data Domain system	Active archive service	Only required when the active archive feature is enabled.

Remote management interface ports

The remote management interface on Avamar utility, storage, and accelerator nodes has specific port requirements both for inbound and outbound ports.

The remote management interface depends on the type of ADS platform:

- The Gen4T platform uses the Baseboard Management Controller (BMC) Web Console
- The Gen4S platform uses the Remote Management Module 4 (RMM4)
- The Gen4 platform uses the Integrated Dell Access Controller 6 (iDRAC6)

The tables in this section list the inbound port requirements for the remote management interface on all the nodes. The ports that must be opened to network transmissions from specified source computers are based on your network environment.

NOTICE

It is recommended to isolate the management network.

Connection to the remote management interfaces depends on the type of ADS platform and is made through the relevant BMC Web Console, RMM4, or iDRAC6 IP address. Do not use the backup interface for this purpose.

NOTICE

The dedicated port and shared port cannot be the same IP address. If the IP address is the same, set the IP address of the shared port to 0.0.0.0. Also, connection of the dedicated port through a switch may require gratuitous ARP to be turned on.

Remote management interface inbound ports

The following table describes the inbound ports that should be open on the remote management interface of all Gen4T-based Avamar nodes. The actual ports that should be open depend on your network environment. For every port listed in this table, the remote management interface on the node is the destination and the source is listed in the Source computer column.

Table 42 Inbound ports for the remote management interface on all Gen4T-based nodes

Port	Protocol	Service name	Source computer	Additional information
80	TCP	HTTP	Administrator computers	HTTP access

Table 42 Inbound ports for the remote management interface on all Gen4T-based nodes (continued)

Port	Protocol	Service name	Source computer	Additional information
443	TCP	HTTP protocol over TLS/SSL	Administrator computers	HTTPS access
2068	TCP	Virtual console and media redirection	Administrator computers	Virtual console keyboard/mouse, virtual media server, virtual media secure service, and virtual console video

The following table describes the inbound ports that should be open on the remote management interface of all Gen4S-based Avamar nodes. The actual ports that should be open depend on your network environment. For every port listed in this table, the remote management interface on the node is the destination and the source is listed in the Source computer column.

Table 43 Inbound ports for the remote management interface on all Gen4S-based nodes

Port	Protocol	Service name	Source computer	Additional information
80	TCP	HTTP	Administrator computers	HTTP access
443	TCP	HTTPS	Administrator computers	HTTPS access
5120	TCP	CDROM media redirection	Administrator computers	
5123	TCP	Floppy/USB media redirection	Administrator computers	
7578	TCP	Keyboard, video, mouse	Administrator computers	

The following table describes the inbound ports that should be open on the remote management interface of all Gen4-based Avamar nodes. The actual ports that should be open depend on your network environment. For every port listed in this table, the remote management interface on the node is the destination and the source is listed in the Source computer column.

Table 44 Inbound ports for the remote management interface on all Gen4-based nodes

Port	Protocol	Service name	Source computer	Additional information
22	TCP	SSH	Administrator computers	Secure shell access
23	TCP	TELNET	Administrator computers	Remote login service

Table 44 Inbound ports for the remote management interface on all Gen4-based nodes (continued)

Port	Protocol	Service name	Source computer	Additional information
80	TCP	HTTP	Administrator computers	HTTP access
443	TCP	HTTPS	Administrator computers	HTTPS access
623	TCP	RMCP/RMCP+	Administrator computers	
5900	TCP	Virtual console and media redirection	Administrator computers	Console Redirection keyboard/mouse, Virtual Media Service, Virtual Media Secure Service, Console Redirection video

Remote management interface outbound ports

The following table describes the outbound ports that should be accessible to network packets that are sent from the remote management interface on all Avamar nodes. The actual ports that should be open depend on your network environment. By default, none of these outbound ports are configured to be in use. You must modify the configuration to use those protocols. For each row, the node is the source computer that must have outgoing access to the listed port on the listed destination computer.

Table 45 Outbound ports for the remote management interface on all Avamar nodes

Port	Protocol	Destination computer	Additional information
25	TCP	Administrator computers	Required to make an SMTP connection with Administrator computers.
53	TCP/UDP	DNS server	Required for DNS queries.
68	UDP	Administrator computers	Required for DHCP-assigned IP address.
69	UDP	Administrator computers	Required for trivial file transfers (TFTP).
162	UDP	Administrator computers	Required to send SNMP traps.
636	TCP/UDP	LDAPS server	Required to make Secure LDAP queries.
3269	TCP /UDP	LDAPS server	Required for LDAPS global catalog (CG).

APPENDIX B

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

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
uuidd
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
```

Permissions within /var folder

Many components of the Avamar system write to the /var folder.

Permissions on the /var folder of an Avamar node are world writeable because many components of the Avamar system write files such as logs there. On physical Avamar servers, the folder in question is /usr/local/avamar/var; on virtual Avamar servers, it is /space/avamar/var. This security exception is necessary for the operation of the product.

APPENDIX C

Enterprise Authentication

This appendix includes the following topics:

Enterprise authentication

Enterprise (or external) authentication enables users to use the same user ID and password to log in to multiple systems.

NOTICE

For backward compatibility, this appendix preserves information about the deprecated Enterprise authentication method. The functionality of this method is replaced, and improved on, 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 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 that are described in the following table.

Table 46 Supported external authentication systems

Category	Description
Lightweight Directory Access Protocol (LDAP)	Hierarchical directory structure, X.500-standard, system such as: • Microsoft Active Directory Service (MS ADS) • Novell NDS and eDirectory

Table 46 Supported external authentication systems (continued)

Category	Description
Network Information Service (NIS) SUN Yellow Pages (YP)	Flat, workgroup-based, database structure of user IDs, passwords, and other system parameters comparable to Microsoft Windows NT such as: • Master NIS Server - Primary Domain Controller (PDC) • Slave NIS Servers - Backup Domain Controllers (BDC)

Configuring Enterprise authentication

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 or Configuring an NIS interface.
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:
   a. Log in to the server as admin.
   b. Switch user to root by typing `su -`.
   c. For a multi-node server, load the rootid OpenSSH key by typing:

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

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 that is 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. Typing 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 system name 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.con`.
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 account that is 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 account that is used to bind with the LDAP server.

14. Restart Login Manager by typing:

```
service lm restart
```

15. Confirm acceptance of the configuration changes, by typing:

    ```
    avmgr lstd
    ```

    All of the Avamar authentication domains 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 in to a single-node server, log in to the server as root.
   - If logging in to 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 that is 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. Typing 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 that is 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 that the system name 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 acceptance of the configuration changes by typing:

    ```
    avmgr lstd
    ```

All Avamar authentication domains 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 start automatically by typing:

    `/sbin/chkconfig ypbind on`
20. Confirm the previous settings by typing:

    `/sbin/chkconfig --list ypbind`

    The following information appears in the command shell:

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

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

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

    where *NUMBERS* is a comma-separated list of the numbers to set "on" (for example, `/sbin/chkconfig --level 3,4, ypbind on`).
21. Start the ypbind daemon by typing:

    `service ypbind restart`

    The following information appears in the command shell:

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

Note

If NIS services has not started, shutting down NIS services can fail. 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:

    ```
    ypwich
    ```

    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 who is 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.