Genesys Quality Management 8.1

Security Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software - with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service - and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System’s Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support. Before contacting technical support, please refer to the ***Genesys Care Program Guide*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 81gqm_security_10-2012_v8.1.501.00

Table of Contents

Chapter

1 Introduction

This chapter provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information.

This chapter contains the following sections:

Document Purpose

This document describes the setup of the main security features in Genesys GQM. It is planned that it will be improved and enhanced with each release – we would welcome your comments regarding further topics you would like to see covered.

Advanced configuration, clustering and integration with third party applications are described in other documents - for example. Genesys Call Recording Administration Guide and related Whitepapers.

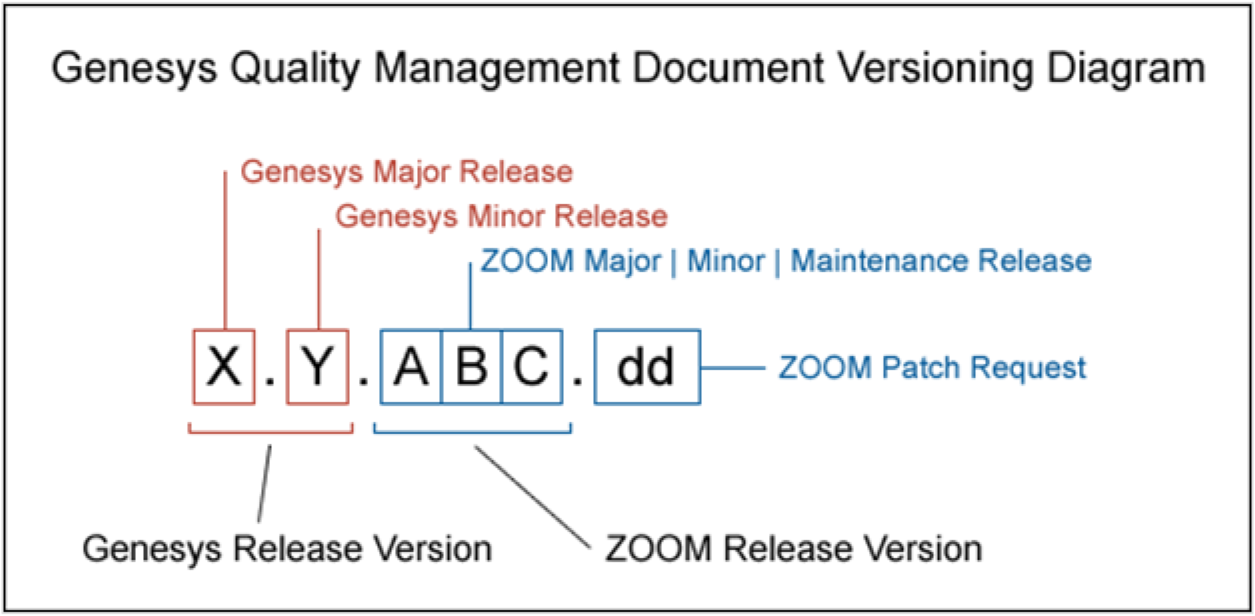
Audience

This document is intended for System Administrators and Architects.

Document Version

The Genesys Quality Management products are provided by a partnership between Genesys and ZOOM International. The Genesys Quality Management products use a versioning format that represents a combination/joining of the versions used by these two separate entities. Although the Genesys Quality Management products and documentation use this combined versioning format, in much of the software and logs you will see the ZOOM versioning alone. You need to be aware of this, for example, when communicating with Technical Support.

The version for this document is based on the structure shown in the following diagram:



Typographical Conventions

Names of functions and buttons are in bold. For example: **Upload**.

File names, file paths, command parameters and scripts launched from the command line are in `non-proportional font`.

Referred documents are in italics. For example: see the document *This is a Document* for more information.

```
Code is placed on a gray background and bordered
```

Hyperlinks are shown in blue and underlined: http://genesyslab.com/support/contact.

Expected Knowledge

Readers of this document are expected to have the following skills or knowledge:

- Basic knowledge of the Genesys Call Recording system features and functionality
- Knowledge of Red Hat Enterprise Linux or CentOS installation and configuration

- Unix system administration skills
- Network administration skills

Chapter

2 Security Guide Overview

This aim of this guide is to cover the most important procedures and best practices in order to ensure that your Genesys Quality Management products are secure and stable. The guide currently covers the following topics:

- PCI DSS Compliance (including secure user access, call data encryption)
- Secure Web Access (https)
- GQM IP Port Assignment (Call Recording, Quality Manager, Screen Capture, Live Monitor)

This guide does not introduce or cover these areas in great depth, but rather offers the Genesys administrator a fast-track reference to configure or apply the appropriate procedures and settings to a new or existing Genesys GQM installation.

Finally, the Appendix includes additional related topics, including the use of the call Encrypt Tool, provided to complement the Key Manager component.

Chapter

3 PCI DSS Compliance

The following Chapter describes PCI DSS Compliance and how each issue is addressed.

This chapter contains the following sections:

PCI DSS Compliance Overview

PCI DSS (Payment Card Industry Data Security Standard) is a worldwide information security standard defined by the Payment Card Industry Security Standards Council – an organization founded by the key electronic payment providers including American Express, Visa, Inc and MasterCard Worldwide. The standard aims to reduce or prevent credit card fraud by requiring that organizations in the payment card industry implement increased controls around cardholder data, thereby minimizing its exposure to compromise.

Certification as “PCI DSS compliant” is mandatory for large numbers of organizations in the credit card payment industry; the standard applies to all organizations that hold, process or exchange cardholder information from any card branded with the logo of one of the PCI SSC members.

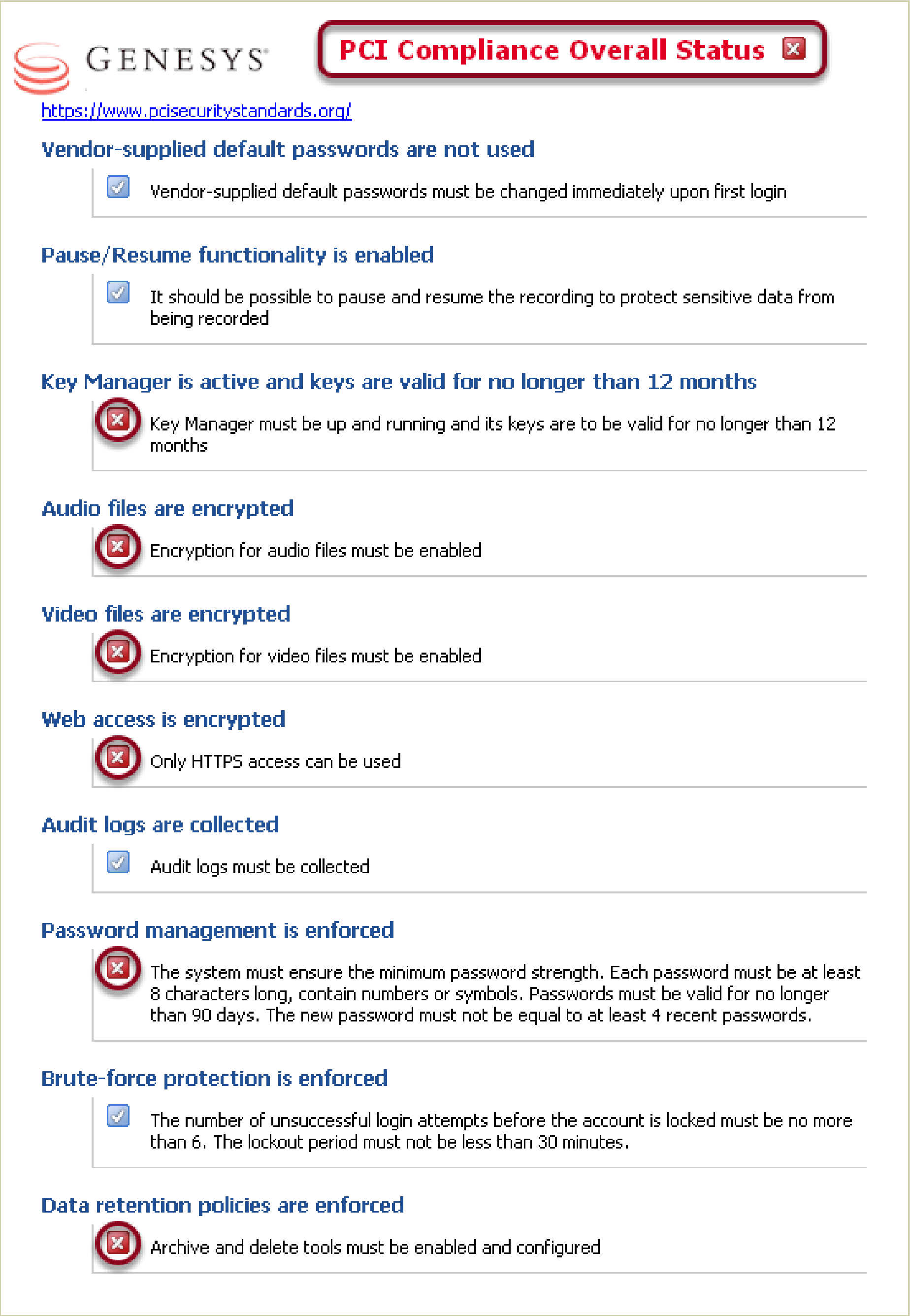
Genesys GQM 8.1.50x introduces full compliancy with the following relevant PCI DSS directives:

Control Objectives	**PCI DSS Requirements**	**GQM 8.1.50x**
Build and Maintain a Secure Network	1. Install and maintain a firewall configuration to protect cardholder data.	N/A
	2. Do not use vendor-supplied defaults for system passwords and other security parameters.	✓
Protect Cardholder Data	3. Protect stored cardholder data.	✓
	4. Encrypt transmission of cardholder data across open, public networks.	✓
Maintain a Vulnerability Management Program	5. Use and regularly update anti-virus software on all systems commonly affected by malware.	N/A
	6. Develop and maintain secure systems and applications.	✓ (ongoing)
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need-to-know.	✓

Control Objectives	PCI DSS Requirements	GQM 8.1.50x
	8. Assign a unique ID to each person with computer access.	✓
	9. Restrict physical access to cardholder data.	N/A
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data.	✓
	11. Regularly test security systems and processes.	N/A
Maintain an Information Security Policy	12. Maintain a policy that addresses information security.	N/A

Table 1: PCI DSS Compliance

Genesys GQM PCI Compliance Checklist



PCI Compliance Overall Status ☒

https://www.pcisecuritystandards.org/

Vendor-supplied default passwords are not used

☑ Vendor-supplied default passwords must be changed immediately upon first login

Pause/Resume functionality is enabled

☑ It should be possible to pause and resume the recording to protect sensitive data from being recorded

Key Manager is active and keys are valid for no longer than 12 months

☒ Key Manager must be up and running and its keys are to be valid for no longer than 12 months

Audio files are encrypted

☒ Encryption for audio files must be enabled

Video files are encrypted

☒ Encryption for video files must be enabled

Web access is encrypted

☒ Only HTTPS access can be used

Audit logs are collected

☑ Audit logs must be collected

Password management is enforced

☒ The system must ensure the minimum password strength. Each password must be at least 8 characters long, contain numbers or symbols. Passwords must be valid for no longer than 90 days. The new password must not be equal to at least 4 recent passwords.

Brute-force protection is enforced

☑ The number of unsuccessful login attempts before the account is locked must be no more than 6. The lockout period must not be less than 30 minutes.

Data retention policies are enforced

☒ Archive and delete tools must be enabled and configured

Figure 1: PCI DSS Compliance Status Screen

Ensure that your Genesys GQM license includes the ‘PCI Compliance’ property, which enables the following features in GQM:

- Key Manager, for managing server and client encryption keys (more information below).
- The PCI Compliance Status page (in the Call Recording Web GUI at **Settings > PCI Compliance Status**), which clearly displays if the GQM features influencing PCI Compliancy are correctly configured within the GQM installation.

Important:

The **PCI Compliance Status** screen will not be visible in the Call Recording Web GUI until a valid license including the PCI Compliance feature has been uploaded and Call Recording restarted.

The following sub-topics cover how to achieve compliancy for each requirement displayed on the **PCI Compliance Status** page.

Vendor-supplied Default Passwords Are Not Used

By default after installation, the first time the system administrator logs in to the Genesys Call Recording Web GUI using the default login credentials, the administrator is required to change the administrator password.

Resolution: none required.

Pause/Resume Functionality Is Enabled

This functionality is currently available via the Pause/Resume and RMI API for third party applications please see the Developer API Guide.

Resolution: none required.

Key Manager Is Active and Keys Are Valid for no Longer than 12 Months

PCI-DSS Compliance requires authenticated, encrypted transmission of data across networks (see Appendix A Encrypt Tool)– which includes between clients and servers in distributed systems like Genesys GQM. One of the functions of the Key Manager is to manage this secure transmission, including automatic transparent renewal of authentication certificates when they expire.

Resolution: install authentication and encryption certificates and activate Key Manager. See Activating Key Manager.

Self-Signed or Commercial Certificates

For standard production environments, we recommend that you use **commercially signed authentication certificates** with Key Manager. “Commercial certificates” are authentication certificates that are signed by a trusted commercial CA (Certificate Authority, such as, Thawte or Verisign).

Self-signed certificates are quick to create; you can create them during GQM setup by answering ‘yes’ to the query "Do you want to create a self-signed certificate and keys for Key Manager?" (see the Implementation Guide).

However, self-signed certificates are not as secure or trusted as commercial certificates, so they can provoke warnings and security errors, particularly when used with web technologies (see the SSL section in this Guide). We only recommend that you use them for testing purposes.

Key Manager in Cluster Installations

To comply with PCI DSS recommendations, in cluster installations Key Manager must only be enabled on one server. Typically Key Manager is deployed on the server that runs Call Recording Core. The Key Manager service in the GQM is selected by default in the service list during setup so the Key Manager service must be deselected on all the other servers in the cluster.

The following security precautions must be taken:

- Remote access to the key store must not be possible.
- The directory where are the keys stored must be protected by file system permissions and should be only accessible for the Key Manager process and the key manager administrator.
- Keys for communication between Key Manager and it's clients should be distributed using safe transport, for example, distributed physically on a USB stick or in protected SSH session.

There is a tool for importing and exporting certificates into and out of the key store.

Activating Key Manager

Activate Key Manager using the following procedure:

Either:

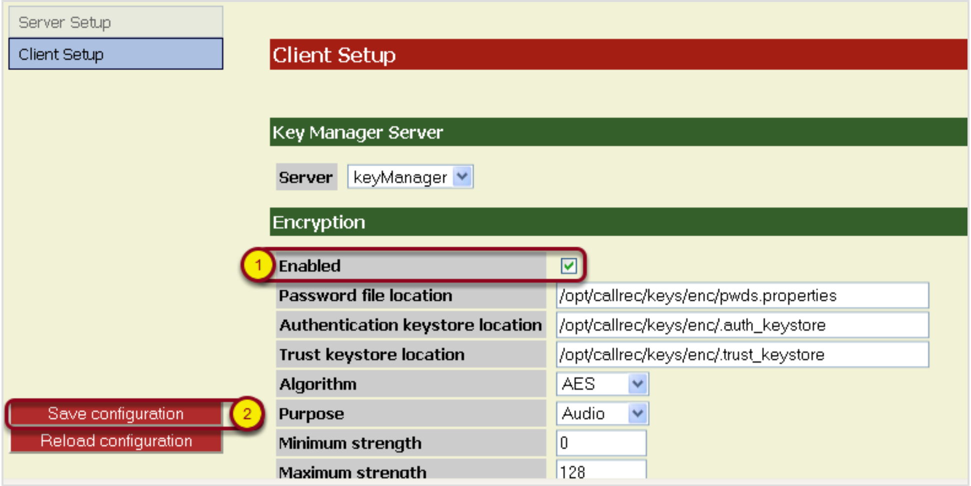
Opt to create **self-signed certificates and keys** during setup. These self-signed certificates are usually only used for test purposes during set up of the system. They are not recommended for use in a working environment.

Or:

Opt to use a **commercial certificate and keys**. In this case, do not create self-signed certificates and keys during setup, but after setup is complete, manually set up Key Manager with a commercial certificate and keys (see the Installing Commercially Signed Certificates section of this guide).

Enabling Encryption in Client Setup

Navigate to:**Settings > Configuration > Key Manager > Client Setup**.



Server	keyManager
Enabled	☑
Password file location	/opt/callrec/keys/enc/pwds.properties
Authentication keystore location	/opt/callrec/keys/enc/.auth_keystore
Trust keystore location	/opt/callrec/keys/enc/.trust_keystore
Algorithm	AES
Purpose	Audio
Minimum strength	0
Maximum strength	128

Figure 2: Activating Key Manager and Call Encryption

1. Select the **Enabled** checkbox in the Encryption section to enable Key Manager and call encryption.
2. Click **Save configuration**.

Important:

> The **Key Manager** settings tab will not be visible in the Call Recording Web GUI until a valid license including the PCI Compliance feature has been uploaded, certificates (self-signed or commercial) installed and Call Recording restarted using the `service callrec restart` command.

In both cases, the key validation expiration dates are determined when generating the server keys, using the `keygen` command line tool. In the case of self-signed certificates created during GQM setup, an expiration date of 365 days is set (the maximum allowable period for PCI Compliance).

Installing Commercially Signed Certificates

Commercially signed certificates are created and installed using the following process. It is assumed that a Certification Authority (CA) such as Thawte or Verisign is available to process certificate signing requests:

- Generate server, encoder and decoder private keys and certificates.
- Generate certificate signing request (.csr) files for each certificate and send these for signing to the CA.

- [Optional] Install a root (trust) certificate for the CA if required.
- Install the signed certificates from the CA in the server authorization store and encoder & decoder trust and authorization stores.
- Generate Key Manager encryption keys.

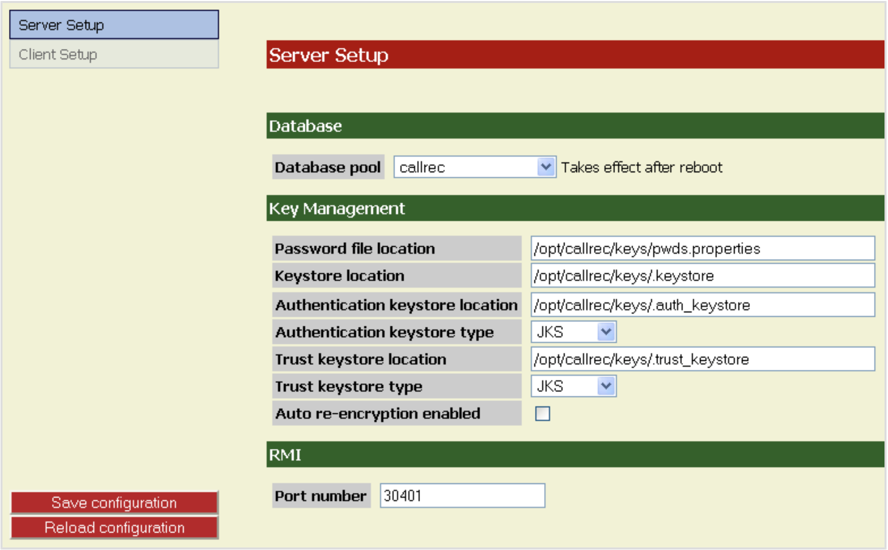
All of this is accomplished at the command line (with root privileges). See Appendix A: Installing Commercial Certificates for Key Manager for full details of the commands used.

Configuring Key Manager

After Key Manager has been activated through the installation of authentication keys and certificates.

Navigate to **Settings > Configuration > Key Manager > Server Setup**.

Server Setup



Server Setup

Database

Database pool	callrec — Takes effect after reboot

Key Management

Password file location	/opt/callrec/keys/pwds.properties
Keystore location	/opt/callrec/keys/.keystore
Authentication keystore location	/opt/callrec/keys/.auth_keystore
Authentication keystore type	JKS
Trust keystore location	/opt/callrec/keys/.trust_keystore
Trust keystore type	JKS
Auto re-encryption enabled	□

RMI

Port number	30401

Figure 3: Key Manager Configuration – Server Setup

The Server Setup section contains the following parameters:

Database

Database Pool: The database pool used by Key Manager – usually `callrec` for a single server installation.

Key Management

Password file location: The Key Manager server's key/certificate password lookup file. Key Manager uses this to manage the key stores in the event of authentication/encryption key expiration & re-creation.

Keystore location: The server key store, containing media encryption keys.

Authentication keystore location: Key Manager's authentication key store, containing the K.M. server's own private authentication key(s).

Trust keystore location: Key Manager's trust key store, containing public authentication keys of trusted clients (for example,encryption & decryption clients).

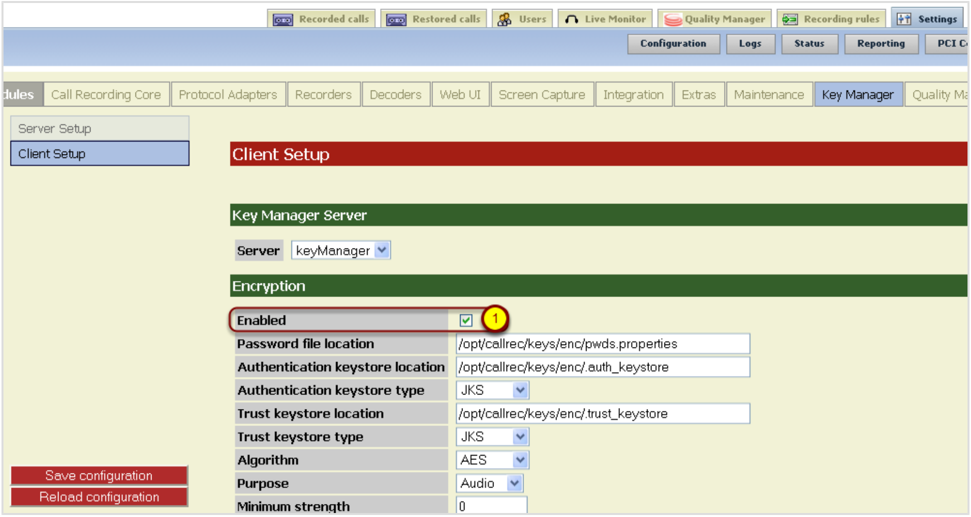
Auto re-encryption enabled: Encrypted files will be automatically re-encrypted when their certificates expire.

RMI

Port number: RMI port number used by Key Manager – typically `30401`.

Client Setup

Navigate to **Settings > Configuration > Key Manager > Client Setup**.



Client Setup

Key Manager Server

Server	keyManager

Encryption

Enabled	☑
Password file location	/opt/callrec/keys/enc/pwds.properties
Authentication keystore location	/opt/callrec/keys/enc/.auth_keystore
Authentication keystore type	JKS
Trust keystore location	/opt/callrec/keys/enc/.trust_keystore
Trust keystore type	JKS
Algorithm	AES
Purpose	Audio
Minimum strength	0

Figure 4: Key Manager Configuration – Client Setup

1. Select the **Enabled** checkbox to enable call and screen capture encryption.

The Client Setup section contains the following parameters:

Key Manager Server

Server: The Key Manager server (defined in Call Recording Core settings).

Encryption

Enabled: Enable call and screen capture encryption. This will only function after both the authentication keys and encryption keys have been configured, as described earlier in this document.

Password file location: The encryption client key/certificate password lookup. The client uses this to manage the key stores in the event of authentication/encryption key expiry and re-creation.

Authentication keystore location: The encryption client authentication key store, containing the client's own private authentication keys.

Trust keystore location: the encryption client trust key store, containing public authentication keys of the trusted servers.

Algorithm: The type of cipher used for encryption and decryption. Genesys uses AES as standard.

Purpose: Specify the key set to use for encryption and decryption. The key set's purpose is defined during key creation (audio is default).

Minimum strength: Lowest strength cipher to use if the server doesn't support stronger algorithms.

Maximum strength: The preferred (default) strength, used if server and client both support it. On a single server default installation this should always be used.

Decryption

Password file location: The decryption client key/certificate password lookup. The client uses this to manage the key stores in the event of authentication/encryption key expiration and re-creation.

Authentication keystore location: The decryption client authentication key store, containing the client's own private authentication keys.

Audio Files Are Encrypted

Once the Key Manager is activated, audio encryption is enabled automatically.

Resolution: none required.

Video Files Are Encrypted

Once the Key Manager is activated, video (Screen Capture) encryption is enabled automatically.

Resolution: none required.

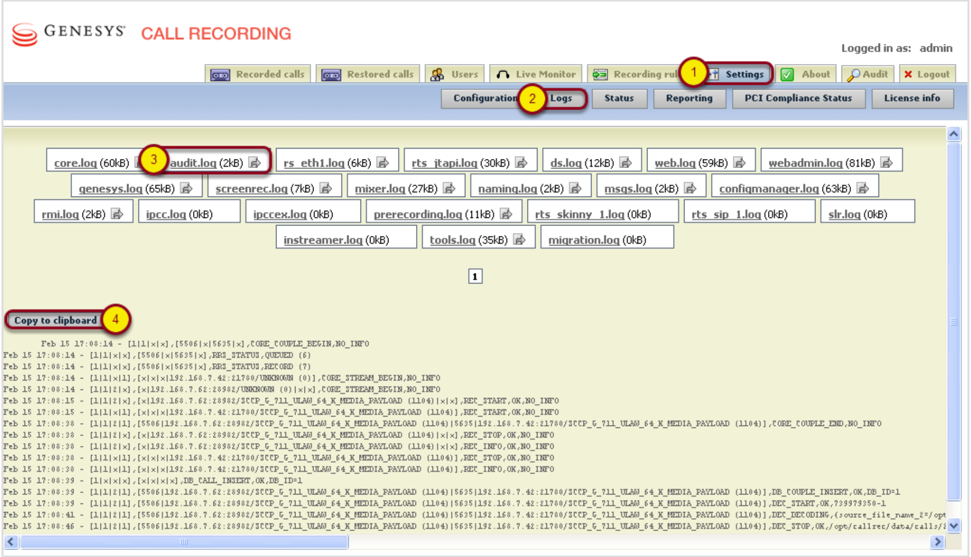
Web Access Is Encrypted

By default, the Tomcat web server installed and configured for the Call Recording Web GUI and Quality Manager applications does not have secure-socket layer (SSL) encryption enabled. This is a requirement for PCI Compliance. Instructions are given in the section Secure Web Access.

Resolution: Manual configuration of SSL security in the Tomcat web server.

Audit Logs Are Collected

By default, audit logs are collected in GQM 8.1.50x Call Recording audit logs are available in the following directory: `/opt/callrec/logs`. They can also be viewed in the Call Recording Web GUI (see screenshot and the Call Recording Administration Guide). Similarly, the Quality Manager audit log can be viewed and exported in Excel format (see the Quality Manager User Guide CC Manager).



```
      Feb 15 17:08:14 - [1|1|x|x],[5506|x|5635|x],CORE_COUPLE_BEGIN,NO_INFO
Feb 15 17:08:14 - [1|1|x|x],[5506|x|5635|x],RRS_STATUS,QUEUED (6)
Feb 15 17:08:14 - [1|1|x|x],[5506|x|5635|x],RRS_STATUS,RECORD (7)
Feb 15 17:08:14 - [1|1|x|1],[x|x|x|192.168.7.42:21780/UNKNOWN (0)],CORE_STREAM_BEGIN,NO_INFO
Feb 15 17:08:14 - [1|1|2|x],[x|192.168.7.62:28982/UNKNOWN (0)|x|x],CORE_STREAM_BEGIN,NO_INFO
Feb 15 17:08:15 - [1|1|2|x],[x|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|x|x],REC_START,OK,NO_INFO
Feb 15 17:08:15 - [1|1|x|1],[x|x|x|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],REC_START,OK,NO_INFO
Feb 15 17:08:38 - [1|1|2|1],[5506|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|5635|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],CORE_COUPLE_END,NO_INFO
Feb 15 17:08:38 - [1|1|2|x],[x|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|x|x],REC_STOP,OK,NO_INFO
Feb 15 17:08:38 - [1|1|2|x],[x|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|x|x],REC_INFO,OK,NO_INFO
Feb 15 17:08:38 - [1|1|x|1],[x|x|x|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],REC_STOP,OK,NO_INFO
Feb 15 17:08:38 - [1|1|x|1],[x|x|x|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],REC_INFO,OK,NO_INFO
Feb 15 17:08:39 - [1|x|x|x],[x|x|x|x],DB_CALL_INSERT,OK,DB_ID=1
Feb 15 17:08:39 - [1|1|2|1],[5506|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|5635|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],DB_COUPLE_INSERT,OK,DB_ID=1
Feb 15 17:08:39 - [1|1|2|1],[5506|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|5635|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],DEC_START,OK,739979350-1
Feb 15 17:08:41 - [1|1|2|1],[5506|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|5635|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],DEC_DECODING,{source_file_name_2=/opt
Feb 15 17:08:46 - [1|1|2|1],[5506|192.168.7.62:28982/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)|5635|192.168.7.42:21780/SCCP_G_711_ULAW_64_K_MEDIA_PAYLOAD (1104)],DEC_STOP,OK,/opt/callrec/data/calls/2
```

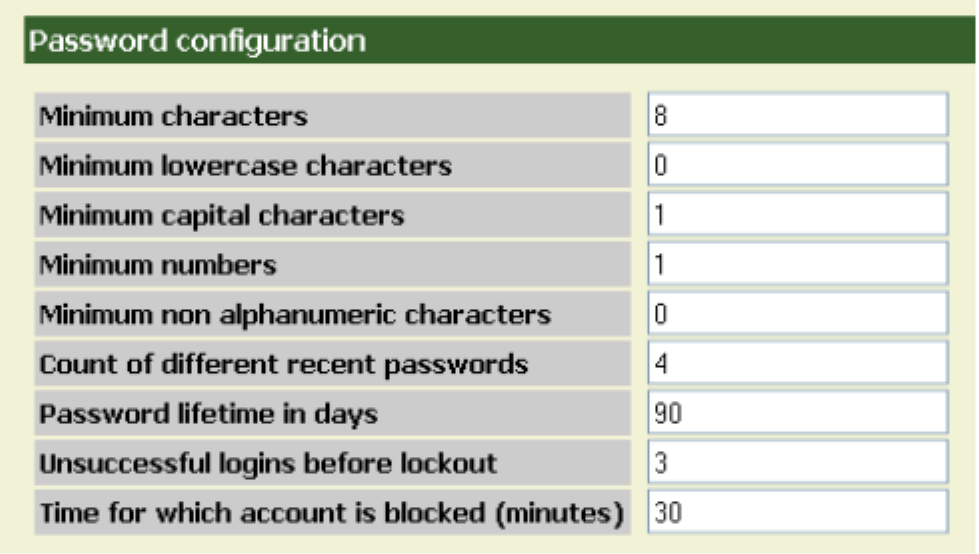
Figure 5: Copying Call Recording Audit Log Data to the Clipboard

Resolution: None required

Password Management is Enforced

GQM 8.1.49x includes advanced password management facilities, which are initially switched off by default, allowing weak passwords to be used. These settings also dictate the settings for Quality Manager. Where integration with external systems is used, the external system dictates password settings for external users.

The following settings are required to be modified from the default values in order for passwords to be marked as PCI Compliant. These are modified in the **Call Recording Web GUI > Settings > Configuration > Web UI > Web Interface > Password configuration** section.



Password configuration

Minimum characters	8
Minimum lowercase characters	0
Minimum capital characters	1
Minimum numbers	1
Minimum non alphanumeric characters	0
Count of different recent passwords	4
Password lifetime in days	90
Unsuccessful logins before lockout	3
Time for which account is blocked (minutes)	30

Figure 6: Minimum password configuration for PCI Compliance

- **Minimum characters**: At least 8
- **Minimum capital characters**: At least 1
- **Minimum numbers**: At least 1

See the screenshot for more details:

For more information on password configuration settings, see the Call Recording Administration Guide (User Interface Configuration section).

Resolution: Update the **Password configuration** settings in Call Recording Web UI.

Brute-force protection is enforced

In addition to the minimum password configuration settings above, PCI Compliance also requires protection against brute-force attacks, when a hacker makes use of automated password generation techniques to repeatedly attempt entry.

To safeguard against these attacks, the following two settings in the Password configuration section are required to be active (they are PCI Compliant by default):

- **Unsuccessful logins before lockout**: 6 or under
- **Time for which account is blocked (minutes)**: 30 or more

To change these settings navigate to Call Recording **Web GUI > Settings > Configuration > Web UI > Web Interface > Password configuration**.

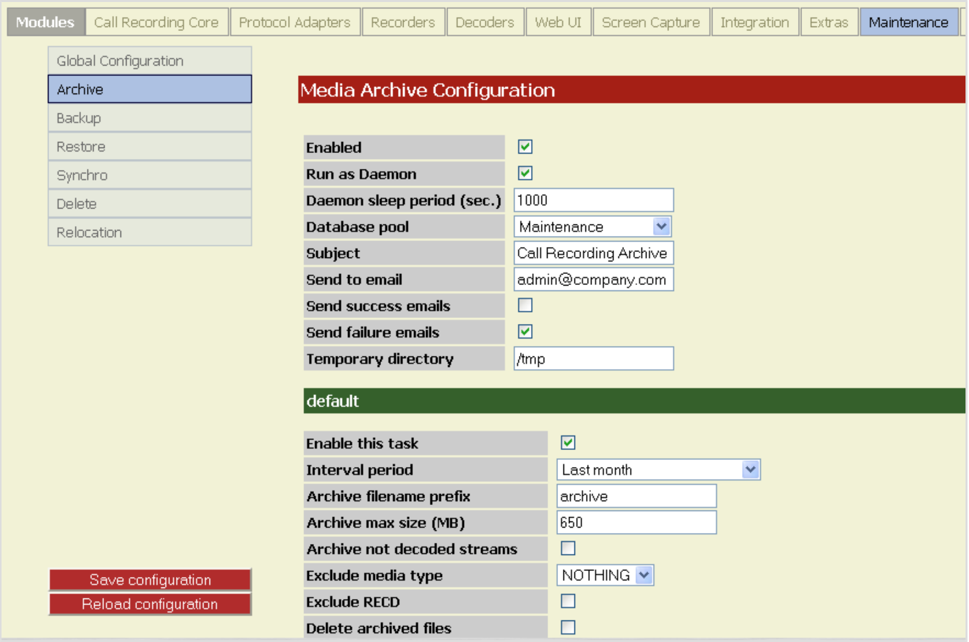
Resolution: None required if default settings are kept.

Data Retention Policies Are Enforced

For full PCI Compliance, both the **Archive** and **Delete** media lifecycle management (MLM) tools need to be configured and operational. Both of these can be enabled and configured in the **Maintenance** section of Call Recording Settings (**Call Recording Web GUI > Settings > Configuration > Maintenance**).

Sample settings for these tools are shown in the following screenshots. It is critical that settings are configured according to your MLM requirements – see the Call Recording Administration Guide (Maintenance section) for more details.

Archive Tool



Media Archive Configuration

Enabled	☑
Run as Daemon	☑
Daemon sleep period (sec.)	1000
Database pool	Maintenance
Subject	Call Recording Archive
Send to email	admin@company.com
Send success emails	□
Send failure emails	☑
Temporary directory	/tmp

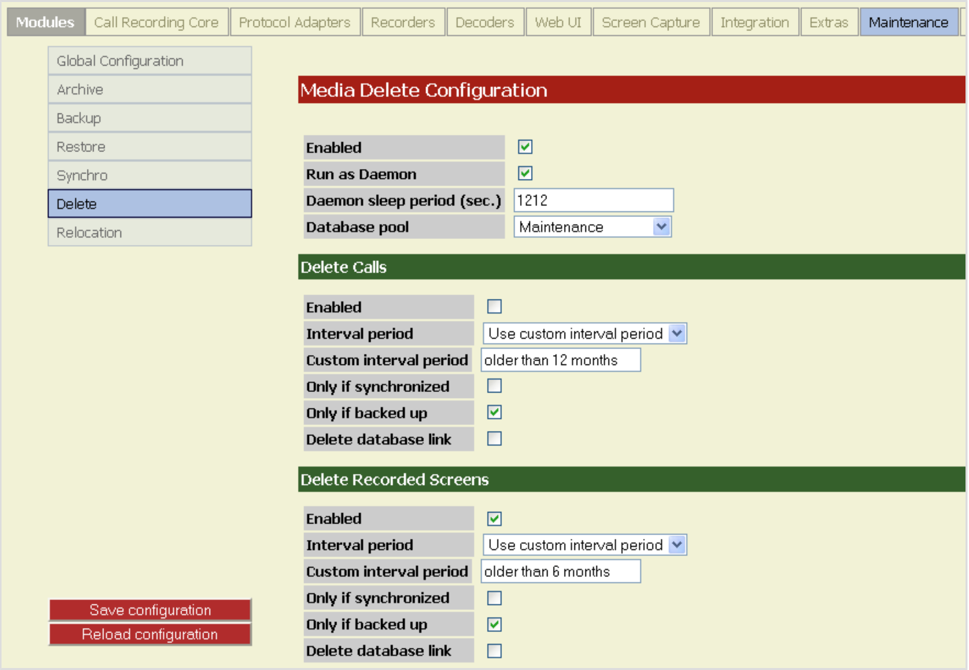
default

Enable this task	☑
Interval period	Last month
Archive filename prefix	archive
Archive max size (MB)	650
Archive not decoded streams	□
Exclude media type	NOTHING
Exclude RECD	□
Delete archived files	□

Figure 7: Maintenance Settings – Archive tool sample settings

Enable the **Archive** tool including **Daemon sleep period** and email settings (**Subject, Send to email** (address), **Send success mails** or **Send failure emails,** then add an archive task, including the Interval period. See the Call Recording Call Recording Administration Guide for more details.

Delete Tool



Media Delete Configuration

Enabled	☑
Run as Daemon	☑
Daemon sleep period (sec.)	1212
Database pool	Maintenance

Delete Calls

Enabled	□
Interval period	Use custom interval period
Custom interval period	older than 12 months
Only if synchronized	□
Only if backed up	☑
Delete database link	□

Delete Recorded Screens

Enabled	☑
Interval period	Use custom interval period
Custom interval period	older than 6 months
Only if synchronized	□
Only if backed up	☑
Delete database link	□

Figure 8: Maintenance Settings – Delete Tool Sample Settings

Enable the **Delete** tool including **Daemon sleep period** (set to a different value than for the Archive tool in this example), then add a **delete task**, including checking (enabling) the type of media to delete and **Interval period** for each. See the Call Recording Administration Guide for more details.

Resolution: Enable and configure the **Archive** and **Delete** MLM tools in Call Recording Maintenance settings.

Chapter

4 Secure Web Access

Genesys GQM installs a web server (Apache Tomcat 6.x) to run web-based applications such as Call Recording Web GUI and Quality Manager. By default, Tomcat is not configured to provide secure (HTTPS) access via a Secure Socket Layer (SSL) implementation, but this is required for PCI-DSS compliancy.

This chapter contains the following sections:

Component Compatibility

At present, not all GQM components are compatible with SSL connections, and require HTTP connectivity to be retained alongside secure HTTPS. Please review the following notes before deciding whether to deploy HTTPS only, or both HTTPS and HTTP protocols in parallel.

- **CUCM-based Prerecording**: Requires HTTP as well as HTTPS due to a CUCM limitation.
- **Live Monitor**: Works with HTTPS with no additional configuration (HTTP not required).
- **Screen Capture**: Currently requires HTTP as well as HTTPS. Although the Screen Capture Capture Client communicates via TLS to the Screen Capture Server (SRS), HTTP is required for communication from the Client to the Screen Capture Media Upload Server.

Configuration

Depending on your deployment, you may need to use a commercial CA (Certificate Authority – such as Thawte or Verisign) to sign the SSL certificates that are created. Using a commercial CA avoids browser warnings regarding ‘untrustworthy’ (self-signed) certificates, which is not an issue if only a small number of administrators need access to the web application, such as for small Call Recording-only deployments.

The following steps cover the procedure to configure secure web access using both commercially signed and self-signed SSL certificates. Tomcat 6.0 contains the Tomcat Native APR library, recommended for production use. However, usage of this library prevents the use of the java `keytool` utility for key & certificate generation; the OpenSSL utility must be used instead as covered here.

Creating the Key/Certificate

- Generate an RSA private key:

```
$ openssl genrsa 1024 > localhost.key
$ chmod 400 localhost.key
```

- **EITHER**: Create a self-signed certificate
  - Answer all questions for certificate data:

Important:

The Common Name certificate parameter must contain the FQDN name of your server, for example,`callrec.mycompany.com`

```
openssl req -new -x509 -nodes -sha1 -days 365 -key localhost.key >
localhost.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:California
Locality Name (eg, city) [Newbury]:San Francisco
Organization Name (eg, company) [My Company Ltd]:MyCompany, Inc.
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname)
[]:callrec.mycompany.com
Email Address []:it-callrec@mycompany.com
```

- **OR**: Obtain a commercially signed certificate
  - Create the certificate signing request file (`cert.csr` in PEM format); answer all questions (including the required challenge password for identification):

```
$ openssl req -new -nodes -sha1 -key localhost.key > cert.csr
```

- Send the certificate signing request file `cert.csr` to your CA.
- After receiving back the signed certificate, save it as `localhost.crt` on the server in the same location as the private key.
- Copy key and certificate into place and change file ownership:

```
$ cp localhost.key /opt/callrec/web/conf
$ cp localhost.crt /opt/callrec/web/conf
$ chown callrec.callrec /opt/callrec/web/conf/localhost.*
```

Converting the Certificate

The signed certificate can be converted from an alternative format to PEM format (`.crt`, `.cer` filetypes) using openssl, for example, the following converts a DER encoded certificate file (`cert.cer`) into PEM format (`localhost.crt`):

```
openssl x509 -inform der -in cert.cer -out localhost.crt
```

For further information and conversion examples, see the OpenSSL documentation: http://www.openssl.org/docs/apps/x509.html and the helpful SSL Shopper site: https://www.sslshopper.com/ssl-converter.html.

Configure Tomcat

- Edit the config file at `/opt/callrec/web/conf/server.xml` to include the following `<Connector>` port node definition (paste within the `<Service name=”Catalina”>` node service definition):

```
<Connector port="8443" maxHttpHeaderSize="8192" maxThreads="150"
           enableLookups="false" disableUploadTimeout="true"
           acceptCount="100" scheme="https" secure="true"
           SSLEnabled="true"
           SSLCertificateFile="${catalina.base}/conf/localhost.crt"
           SSLCertificateKeyFile="${catalina.base}/conf/localhost.key" />
```

Important:

If you wish to specify the version of the SSL protocol used, you can add the following option into the Connector port configuration (see http://tomcat.apache.org/tomcat-6.0-doc/apr.html#HTTPS for details):

```
SSLProtocol="SSLv3"
```

- If you wish to disable unsecured HTTP access, comment out the http connector pointing to port 8080 in the file `/opt/callrec/web/conf/server.xml`:

```
<!--
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
-->
```

Restart The Call Recording Web Service

- After completing configuration, restart the Call Recording web service:

```
$ /opt/callrec/bin/rc.callrec_web restart
```

- Observe the web server log at `/var/log/callrec/web.log` for any errors.
- If the web server restarts successfully, and no serious errors are apparent in the server log:
  - Log in to the Call Recording Web GUI using the secure URL address, of the form:
    `https://<FQDN>:8443/callrec`
    The `<FQDN>` is the Fully Qualified Domain Name for your Call Recording Web server. It must be the same as that entered for the Common Name parameter of the `localhost.crt` certificate earlier, for example, `callrec.mycompany.com`.
- If the web server is not accessible, try to access using the original non-secure http URL; if necessary re-enabling non-secure access if it was disabled earlier. Troubleshoot the `/var/log/callrec/web.log` log file for further indication of any issues.

Add Localhost Certificate to Java CA Certificates

- Use the Java `keytool` utility to add the new `localhost.crt` certificate to the collection of trusted Certification Authorities (CA). Change the `–alias` parameter value (`callrecssl`) if required:

```
keytool -keystore /usr/java/jdk1.6.0_35/jre/lib/security/cacerts -alias
callrecssl -importcert -file
/opt/callrec/web/conf/localhost.crt
```

- Enter the default keystore password `changeit`.
- Ensure the displayed certificate information is correct and type `y` to trust the certificate.
- For more information on the `keytool` utility, including how to change the keystore password, see: http://download.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html.

Add Port Redirect to IP Tables

At this point, SSL access is functional, but a port (:8443) is always required in the Call Recording server URL. Adding an SSL port redirect rule to the Linux IP Tables configuration via the following procedure removes this requirement:

- Add redirect rule to existing IP Tables (replace `10.9.8.7` with your Call Recording server IP address):

```
iptables -t nat -A PREROUTING -d 10.9.8.7 -p tcp --dport 443 -j REDIRECT --
to-ports 8443
```

- List (and note) updated IP Tables:

```
iptables -t nat -L -v -n
```

- Save updated IP Tables records:

```
/etc/init.d/iptables save
```

- Restart IP Tables:

```
/etc/init.d/iptables restart
```

- Check (and compare) updated IP Tables:

```
iptables -t nat -L -v -n
```

- Restart web server, cleaning out the server cache too:

```
/opt/callrec/bin/rc.callrec_web stop
rm -rf /opt/callrec/web/work/Catalina/localhost/*
/opt/callrec/bin/rc.callrec_web start
```

- The Call Recording web server should now be accessible at the URL: `https://<SERVER_IP>` without a port being specified; for example, `https://10.9.8.7`

Configure Quality Manager Stream URL Setting

- When secure access to the Call Recording Web GUI is finalized, the Quality Manager **URL to Call Recording stream** parameter must be updated in the Call Recording **Web GUI > Settings > Configuration > Quality Manager > Basic Setup** section to allow Quality Manager to correctly play media over the secure connection.
- The Call Recording stream parameter will be the same URL as used to access the Call Recording Web GUI over https, for example: `https://<FQDN>/callrec`

At this point, SSL access should be working for all GQM Tomcat-based web applications.

More information on setting up SSL in Apache Tomcat: http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html#Troubleshooting.

Chapter

5 Password Storage in GQM

To meet PCI DSS requirements for password storage, from GQM version 8.0.47x onwards, passwords are stored in the Call Recording database as follows:

- A unique password salt is created for each user and stored in the database.
- The user's password is hashed with the salt using approx. 1000 passes of the SHA-1 encryption algorithm.

This procedure provides protection against brute force and rainbow table attacks - see the references below for more information.

References:

- Wikipedia entry for cryptographic salts: http://en.wikipedia.org/wiki/Salt_(cryptography)
- Wikipedia entry for the SHA-1 cryptographic hash function: http://en.wikipedia.org/wiki/Sha-1
- Wikipedia entry for Brute Force attacks: http://en.wikipedia.org/wiki/Brute-force_attack
- Wikipedia entry for Rainbow Tables: http://en.wikipedia.org/wiki/Rainbow_table

Chapter

6 Request Technical Support

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact http://genesyslab.com/support/contact Genesys Technical Support.

Appendix

A Reference

This appendix contains further advanced information or procedures that may be necessary for your GQM installation.

Knowledge of Linux administration commands and techniques is assumed. Please contact http://genesyslab.com/support/contact if you need any support or further technical information.

This chapter contains the following sections:

GQM Port Usage Guide

The single server installation uses the following ports:

Port Number	TCP	UDP	Use
22	✓		SSH – distant access
80	✓		GUI – http (internally redirected to port 8080)
111	✓	✓	NFS (for replay synchro)
389	✓		LDAP
443	✓		GUI – https (internally redirected to port 8443)
2049	✓	✓	NFS (for replay synchro)
4001 – 4004	✓	✓	NFS (for replay synchro)
5060	✓	✓	SLR default SIP port
5432	✓		PostgreSQL (for replay synchro)
7003	✓		Screen Capture Server (also TLS)
8080	✓		GUI – http (see port 80)
8443	✓		GUI – https (see port 443)
16384 - 17183.		✓	RTP streams to SLR
30100	✓		Skinny sniffer
30200	✓		SIP sniffer
30300	✓		JTAPI sniffer
30350	✓		MSR SLR sniffer

Port Number	TCP	UDP	Use
30400	✓		Default RMI port
30401	✓		Key Manager
30500	✓		Configuration service (allow it for Live Monitor)
30501	✓		Configuration service (allow it for Live Monitor)
30600	✓		Core (allow it for Live Monitor)
30601	✓		Core (allow it for Live Monitor)
37000 - 37100		✓	Datagrams ports (allow it for Live Monitor)

Table 2: Single Server Port Usage Guide

Genesys default ports for MSR/EPR/GIM

Port Number	TCP	UDP	Use
2020	✓		Genesys Configuration Service
3000	✓		T-Server communication

Table 3: Genesys Default Ports for MSR/EPR/GIM

Tip:

RMI communications between modules uses random ports from range: 1024 – 65535 (TCP).

Important:

Do not change Port settings directly in configurations files without consulting Genesys Support it is better to change these settings through the Admin User Interface. Ensure you have a backup of all configuration files before changing port numbers.

Installing Commercial Certificates for Key Manager

The following steps assume that you have not installed self-signed certificates for Key Manager (that is, selected **No** to the Do you want to create self-signed certificates... prompt during setup). However, if self-signed certificates are already installed, it is advised to remove them before attempting to install commercial certificates as follows, to avoid confusion:

Important:

You need to be logged in via SSH as the root user, since administrative privileges are required.

Basic Linux administration experience is required, including editing text files, creating and managing files and directories.

```
rm -rf /opt/callrec/keys
/opt/callrec/bin/rc.callrec_keymanager restart
Stopping CallREC Key Manager: ..........                   [  OK  ]
Starting CallREC Key Manager: ....                         [  OK  ]
```

Create keys directory, private keys and certificate request files.

1. Copy the following commands into a text file named `/home/admin/genkeys1.sh`, then modify the `CERTIFICATES_PASS` and `CERTIFICATES_PROPERTIES` information regarding password and organization details respectively.

```
#!/bin/sh
#
# Set up and create request files (.csr) for commercially signed
# certificates for Key Manager
# Genesys Labs, Inc. - GQM    8.1.50x
#
####### Modify as required #######
# Password for all certificate stores
CERTIFICATES_PASS=callrec
# Organizational details for certificates
# [first and last name, organizational unit, organization, city or
```

```
locality,
#  state or province, two-letter country code]
CERTIFICATES_PROPERTIES="CN=Administrator, OU=Dept, O=Company, L=City,
S=State, C=US"
##################################
######## Standard CallREC defaults #######
CALLREC_HOME=/opt/callrec
ERR_FILE=/tmp/installcerts.err
KEYTOOL=/usr/java/default/bin/keytool
KEYS_DIR=$CALLREC_HOME/keys
ENC_DIR=$KEYS_DIR/enc
DEC_DIR=$KEYS_DIR/dec
PWDS_FILE=$KEYS_DIR/pwds.properties
##########################################

# Create CallREC keys directory if it doesn't exist
# Creating /opt/callrec/keys directory tree including pwds.properties
files
if [ ! -e $KEYS_DIR ] ; then
mkdir -p $KEYS_DIR
fi
if [ ! -e $ENC_DIR ] ; then
mkdir -p $ENC_DIR
fi
if [ ! -e $DEC_DIR ] ; then
mkdir -p $DEC_DIR
fi

# Generating content of PWDS file
echo "authpwd=$CERTIFICATES_PASS" > $PWDS_FILE
echo "trustpwd=$CERTIFICATES_PASS" >> $PWDS_FILE
echo "keystorepwd=$CERTIFICATES_PASS" >> $PWDS_FILE
echo "keyentriespwd=$CERTIFICATES_PASS" >> $PWDS_FILE
cp $PWDS_FILE $ENC_DIR
cp $PWDS_FILE $DEC_DIR

# Generating content of PWDS file
echo "authpwd=$CERTIFICATES_PASS" > $PWDS_FILE
echo "trustpwd=$CERTIFICATES_PASS" >> $PWDS_FILE
echo "keystorepwd=$CERTIFICATES_PASS" >> $PWDS_FILE
echo "keyentriespwd=$CERTIFICATES_PASS" >> $PWDS_FILE
cp $PWDS_FILE $ENC_DIR 2>&1 >> $ERR_FILE
cp $PWDS_FILE $DEC_DIR 2>&1 >> $ERR_FILE

# Create private certificates for server and encoder, decoder clients,
#   then generate certificate signing request files (server.csr,
encoder.csr,
#   decoder.csr) in the /home/admin directory
# NOTE: To export existing certificates instead, replace the '-certreq'
#       parameter with '-exportcert', which will export a .cer type
#       certificate file, e.g. server.cer.
# Server
$KEYTOOL -genkeypair -alias server -keyalg rsa -keysize 2048
-validity 365
-keypass $CERTIFICATES_PASS -keystore $KEYS_DIR/.auth_keystore -storetype
jks
-storepass $CERTIFICATES_PASS -dname "$CERTIFICATES_PROPERTIES"
2>&1 >> $ERR_FILE

$KEYTOOL -certreq -alias server -file /home/admin/server.csr
-keystore
$KEYS_DIR/.auth_keystore -storetype jks -storepass $CERTIFICATES_PASS
2>&1 >> $ERR_FILE

# Encoder
$KEYTOOL -genkeypair -alias encoder -keyalg rsa -keysize 2048 |
-validity 365
-keypass $CERTIFICATES_PASS -keystore $ENC_DIR/.auth_keystore -storetype jks
-storepass $CERTIFICATES_PASS -dname "$CERTIFICATES_PROPERTIES"
2>&1 >> $ERR_FILE

$KEYTOOL -certreq -alias encoder -file /home/admin/encoder.csr
-keystore
$ENC_DIR/.auth_keystore -storetype jks -storepass $CERTIFICATES_PASS
2>&1 >> $ERR_FILE

# Decoder
$KEYTOOL -genkeypair -alias decoder -keyalg rsa -keysize 2048
-validity 365
-keypass $CERTIFICATES_PASS -keystore $DEC_DIR/.auth_keystore -storetype jks
-storepass $CERTIFICATES_PASS -dname "$CERTIFICATES_PROPERTIES"
2>&1 >> $ERR_FILE

$KEYTOOL -certreq -alias decoder -file /home/admin/decoder.csr
-keystore
$DEC_DIR/.auth_keystore -storetype jks -storepass $CERTIFICATES_PASS
2>&1 >> $ERR_FILE

# Set permissions
# Changing key file ownership to callrec/callrec
chown -R callrec:callrec $KEYS_DIR 2>&1 >> $ERR_FILE
```

2. Execute the following commands to run the file. Three '`.csr`' certificate signing request files (`server.csr`, `encoder.csr`, `decoder.csr`) will be created in the `/home/admin` directory.

```
chmod 755 /home/admin/genkeys1.sh
/home/admin/genkeys1.sh
```

Obtain Signed Certificates

3. Send the three certificate request files in the `/home/admin` directory to your chosen Certificate Authority (CA) and receive signed certificate files in return – upload them also to the `/home/admin` directory and rename them (if necessary) to `server.cer`, `encoder.cer`, `decoder.cer`.
4. **[OPTIONAL]** Install CA certificate file if CA is not include in the cacerts Java keystore.
5. Check for the existence of your CA in the `cacerts` keystore with the following command that lists all CA owner names (default password is `changeit`):

```
/usr/java/default/bin/keytool -list -v -keystore
/usr/java/default/jre/lib/security/cacerts | grep "Owner:"
```

6. To install a CA certificate, first modify the `-alias` and `-file` parameters in the following command to reflect a suitable reference name and location of certificate file before running it for certificate installation:

```
/usr/java/default/bin/keytool -importcert -alias myCA -file
/home/admin/myCA.cer
-keystore /usr/java/default/jre/lib/security/cacerts -storepass changeit
```

Install signed certificates and create encryption/decryption certificates

7. Copy the following commands into a second text file named `/home/admin/genkeys2.sh`, then modify the `CERTIFICATES_PASS` to match the value you used for it in the earlier `genkeys1.sh` script.

```
#!/bin/sh
#
# Install signed certificates in Key Manager for encryption/decryption
# Genesys Labs, Inc. - GQM   8.1.50x
#
####### Modify as required #######
# Password for all certificate stores
```

```
CERTIFICATES_PASS=callrec
##################################
######## Standard CallREC defaults #######
CALLREC_HOME=/opt/callrec
ERR_FILE=/tmp/installcerts.err
KEYTOOL=/usr/java/default/bin/keytool
KEYS_DIR=$CALLREC_HOME/keys
ENC_DIR=$KEYS_DIR/enc
DEC_DIR=$KEYS_DIR/dec
PWDS_FILE=$KEYS_DIR/pwds.properties
CACHED_CFG_SERVER_IP=localhost
DEFAULT_PORT="30400"
##########################################
# OPTIONAL: Import CA certificates (only required if CA is not included
# in java CACERTS keystore)
# View current CACERTS entries like this (default password: changeit)
#/usr/java/default/bin/keytool -list -v -keystore
#/usr/java/jdk1.6.0_35/jre/lib/security/cacerts | grep "Owner:"
#
# To install a CA certificate, uncomment the following line, and modify
# the -alias and -file parameters to reflect a suitable reference name and
# location of certificate file:
#/usr/java/default/bin/keytool -importcert -alias myCA -file
#/home/admin/myCA.cer -keystore /usr/java/jdk1.6.0_
35/jre/lib/security/cacerts
#-storepass changeit

# Import signed cerficates recieved from your Certificate Authority (CA)
# Assumes that certificates are named server.cer, encoder.cer, decoder.cer
# in the /home/admin directory
# Server
$KEYTOOL -importcert -noprompt -trustcacerts -alias server -file
/home/admin/server.cer -keystore $KEYS_DIR/.trust_keystore -storepass
$CERTIFICATES_PASS 2>&1 >> $ERR_FILE

# Encoder (assumes CACERT certificate file is at $KEYS_DIR/.auth.cer)
$KEYTOOL -importcert -noprompt -trustcacerts -alias encoder -file
/home/admin/encoder.cer -keystore $KEYS_DIR/.trust_keystore -storepass
$CERTIFICATES_PASS 2>&1 >> $ERR_FILE

$KEYTOOL -importcert -noprompt -trustcacerts -alias server -file
/home/admin/server.cer -keystore $ENC_DIR/.trust_keystore -storepass
$CERTIFICATES_PASS 2>&1 >> $ERR_FILE

# Decoder (assumes CACERT certificate file is at $KEYS_DIR/.auth.cer)
$KEYTOOL -importcert -noprompt -trustcacerts -alias decoder -file
/home/admin/decoder.cer -keystore $KEYS_DIR/.trust_keystore -storepass
```

```
$CERTIFICATES_PASS 2>&1 >> $ERR_FILE

$KEYTOOL -importcert -noprompt -trustcacerts -alias server -file
/home/admin/server.cer -keystore $DEC_DIR/.trust_keystore -storepass
$CERTIFICATES_PASS 2>&1 >> $ERR_FILE


# Set permissions
# Changing key file ownership to callrec/callrec
chown -R callrec:callrec $KEYS_DIR 2>&1 >> $ERR_FILE

# Restart Key Manager
/opt/callrec/bin/rc.callrec_keymanager restart

# Create encryption/decryption keys using QM Suite genkeys utility
# Default activation date = today (or format: dd-mm-yyyy)
ACTIVATION_DATE=`date "+%d.%m.%Y"`
# Default expiration date = today + 365 days (or format: dd-mm-yyyy)
EXPIRATION_DATE=`date -d "+365 days" "+%d.%m.%Y"`
$CALLREC_HOME/bin/genkeys -activationDate $ACTIVATION_DATE
-algorithm AES
-expirationDate $EXPIRATION_DATE -purpose Audio -strength 128 -config
"//$CACHED_CFG_SERVER_IP:$DEFAULT_PORT/pci_compliance" 2>&1 >> $ERR_FILE
```

8. Execute the following two commands to run the file. Note the output below the commands. If you see something similar to the sample output below, certificate installation was successful. Otherwise check the default error file at `/tmp/installcerts.err`.

```
chmod 755 /home/admin/genkeys2.sh
/home/admin/genkeys2.sh
```

Sample output:

```
Certificate was added to keystore
Certificate was added to keystore
Certificate was added to keystore
Certificate was added to keystore
Certificate was added to keystore
0 [main] INFO cz.zoom.callrec.keyman.client.cmd.KeyGeneratorClient
- Fetched remote KeyVaultAdmin
287 [main] INFO cz.zoom.callrec.keyman.client.cmd.KeyGeneratorClient
- Generated key, uuid=87639aff-716f-41f3-a304-47594125edfe, algorithm=AES,
strength=128
```

```
287 [main] INFO cz.zoom.callrec.keyman.client.cmd.KeyGeneratorClient
- Key generation completed successfully
```

9. Switch on call encryption in the Call Recording Web GUI (see Client Encryption).
10. Restart Key Manager

```
/opt/callrec/bin/rc.callrec_keymanager restart
Stopping CallREC Key Manager: ..........                     [  OK  ]
Starting CallREC Key Manager: ..                             [  OK  ]
```

More information on keys, certificates and the Java keytool utility: Java SE keytool reference

Troubleshooting Key Errors

- If call encryption has been enabled in the Call Recording Web GUI, but calls are represented by a warning icon with the message **Decoder error (IO failure)**, check the decoder error log at `/opt/callrec/logs/ds.error.log`.
- If an exception containing text similar to: **cz.zoom.callrec.keyman.KeyVaultException: No key with these parameters can be found**, there is an issue with the encryption keys, which is preventing the decoder working. They should be reinstalled as follows:

  Remove the existing keys and certificates: `rm -f /opt/callrec/keys`

  1. Stop Call Recording: `service callrec stop`
  2. Run GQM setup again, selecting options to create self-signed certificates if required: `/opt/callrec/bin/callrec-setup`
  3. Follow the earlier instructions to install commercial certificates if required, and enable call encryption again
  4. If you repeatedly get the same key errors, please contact http://genesyslab.com/support/contact

Encrypt Tool

The encrypt tool (found at `/opt/callrec/bin/encrypt` on a default Call Recording server installation) is used to encrypt un-encrypted media files, or re-encrypt compromised media files (the encryption keys are no longer valid or safe).

There is an optional parameter `-r` allowing re-encryption of encrypted files. If run without this parameter, the tool will ALWAYS only encrypt non-encrypted files.

Parameters

`-config pci_compliance`: Mandatory parameter, which points to PCI compliance related parameters in the Configuration Service.

`-r`: Optional re-encryption mode parameter. If specified, only encrypted (compromised) files will be re-encrypted, otherwise only non-encrypted files will be encrypted.

`-date`: Optional parameter, which specifies a time window filter ('from' date and 'to' date) for files to encrypt. Date format: `hh/dd/mm/yyyy`. For example, `–date 23/04/05/2011 00/05/05/2011` would process all files created between 11pm of May 4th 2011 and midnight of May 5th 2011.
If no date is provided, the tool will display a message similar to the following:
`WARNING! No time range has been specified. Processing may take a while and can cause a significant load on the server.`

`-logger`: Optional parameter, which is provided with the path to a log4j properties file, for a customized debug log. More information about setting up log4j property files is given in the Switching On Debug Logs section of this Appendix.

Examples:

1. Encrypt all non-encrypted files:

```
/opt/callrec/bin/encrypt -config pci_compliance -logger
</path/to/log4j/properties/file>
```

2. Encrypt all non-encrypted files within given 1-hour time window:

```
/opt/callrec/bin/encrypt -config pci_compliance -date 20/04/05/2011
00/04/05/2011 -logger </path/to/log4j/properties/file>
```

3. Re-encrypt all encrypted files:

```
/opt/callrec/bin/encrypt -config pci_compliance -r -logger
</path/to/log4j/properties/file>
```

4. Re-encrypt all encrypted files with compromised key in given time window:

```
/opt/callrec/bin/encrypt -config pci_compliance -r -date date1 date2 -logger
</path/to/log4j/properties/file>
```

Switching On Debug Logs

Sometimes the default debug output of a Call Recording tool or script is not enough to pinpoint the cause of the error; therefore you need to switch on more granular error reporting. This can be accomplished as follows (and is similar for virtually any other component in the Genesys Quality Management product, since all use the same ‘log4j’ logging API):

1. Create a log configuration file with the following content using vi or other text editor and save it as `/opt/callrec/etc/mydebuglog.log4j.properties` (modify the `/var/log/callrec/mydebuglog.log` output log location as required):

```
log4j.rootLogger=TRACE, file
# file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.MaxFileSize=2500MB
log4j.appender.file.MaxBackupIndex=0
log4j.appender.file.File=/var/log/callrec/mydebuglog.log
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %-5p [%t]
%c - %m\n
```

2. Run the tool or script, using the `logger` parameter to specify the location of the configuration file you have created.
   For example, the following is how the `encrypt` tool is given the `logger` parameter:

```
/opt/callrec/bin/encrypt -logger
/opt/callrec/etc/mydebuglog.log4j.properties
```

3. View the output log at the location you specified and search for errors and exceptions in the detailed output:

```
less /var/log/callrec/mydebuglog.log
```