Genesys Quality Management 8.0

Security Guide

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. 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

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 41. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

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: 80gqm_security_04-2011_8.0.480.00 v.1.00

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:

Document Purpose

This document describes the setup of the main security features in Genesys Quality Management 8.0.480. 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 - e.g. Genesys Call Recording Administrator Guide and related Whitepapers.

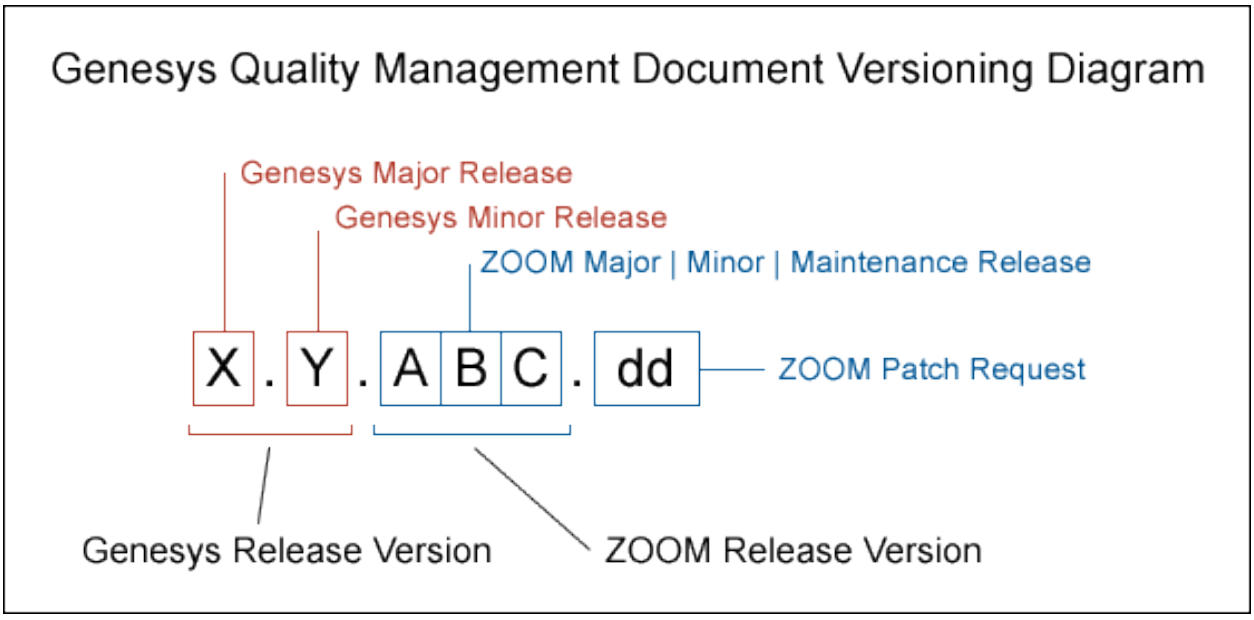
Audience

This document is intended for 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:



Related Documents

For other documents related to Genesys Call Recording please consult:

- *Genesys Call Recording 8.0 User Guide*
- *Genesys Quality Management 8.0 Installation Guide*
- *Genesys Call Recording 8.0 Administration Guide*
- *Genesys Quality Management 8.0 Planning Guide*

Conventions Used

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

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

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

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)
- Genesys Quality Management IP Ports (Screen Capture, Call Recording, Quality Manager, 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 Quality Management installation.

Chapter

3 PCI DSS Compliance

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.

The information in this chapter is divided into the following topics:

Genesys Quality Management 8.0.480 introduces full compliancy with the following relevant PCI DSS directives:

Control Objectives	PCI DSS Requirements	Quality Management 8.0.480
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	✓
	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

Genesys Quality Management PCI Compliance Checklist

Ensure that your Genesys Quality Management license includes the 'PCI Compliance' property, which enables the following features in Genesys Quality Management :

- 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 Genesys Quality Management features influencing PCI Compliancy are correctly configured within the Genesys Quality Management installation.

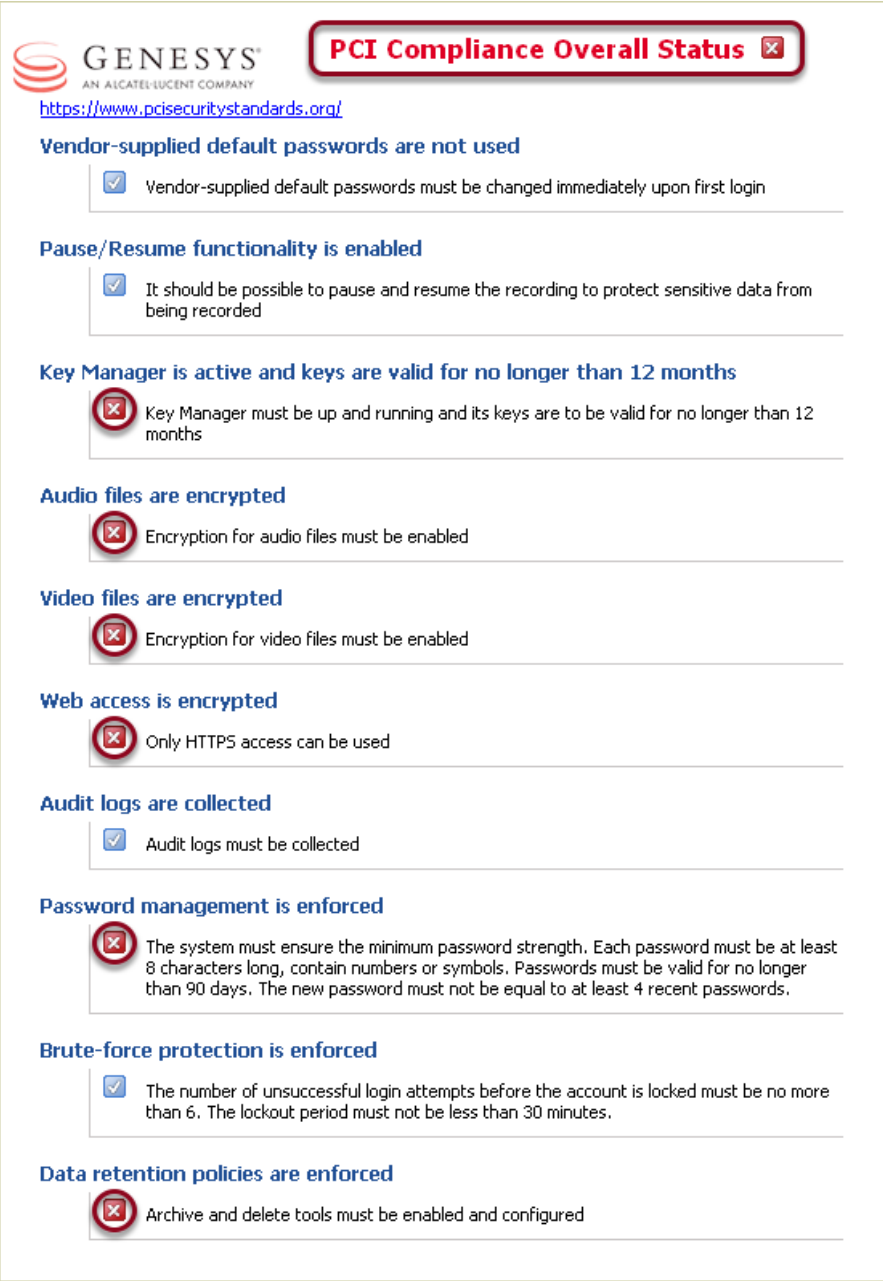


Figure 1: PCI DSS Compliance Status screen

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, he/she is required to change his/her password.

Resolution: ensure the system administrator is not (again) using the default password ('admin').

Pause/Resume functionality is enabled

This functionality is currently available via the RMI API for third party applications; it will be extended further in an upcoming service release, so at this time it is permanently enabled after installation.

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 – which includes between clients and servers in distributed systems like Genesys Quality Management . 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 as follows:

Self-Signed or Commercial Certificates

For standard production environments, it is recommended that **commercially signed authentication certificates** are used with Key Manager. "Commercial certificates" are in fact self-signed certificates that are signed by a commercial CA (Certificate Authority – such as Thawte or Verisign) to sign the authentication certificates that are created. The point at which this happens is indicated in the procedure below.

Self-signed certificates are quick to create; they can be created during Genesys Quality Management setup by answering 'yes' to the query 'Do you want to create a self-signed certificate and keys for Key Manager?' (see the *Quality Management Installation 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). They are therefore only recommended for testing purposes.

Key Manager Activation

The Key Manager is enabled if selected in the Genesys Quality Management service list during setup, and is activated using the following procedure:

1. **Either**:

   Opt to create **self-signed certificates and keys** during setup.

   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).

2. Enable Key Manager and call encryption in the Call Recording Web GUI (**Settings** > **Configuration** > **Key Manager** > **Client Setup**):

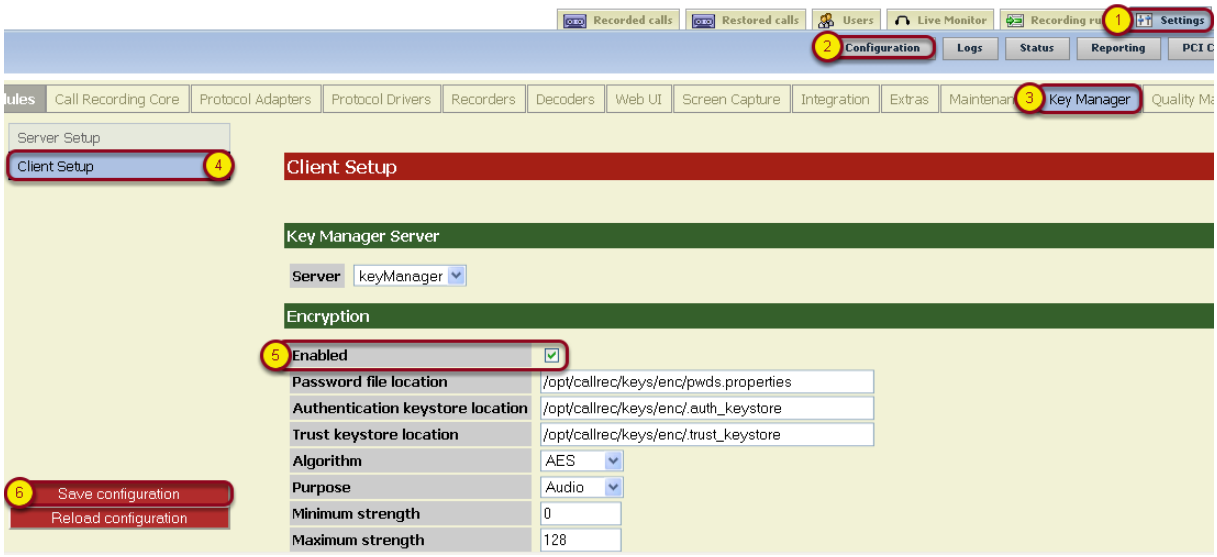


Figure 2: Activating Key Manager and call encryption

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 Genesys Quality Management 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 Chapter 6: Installing Commercial Certificates for Key Manager for full details of the commands used.

Key Manager Configuration

After Key Manager has been activated through the installation of authentication keys and certificates, the configuration parameters can be found in **Settings** > **Configuration > Key Manager**, in **Server** and **Client** sub-sections, as follows:

Server Setup

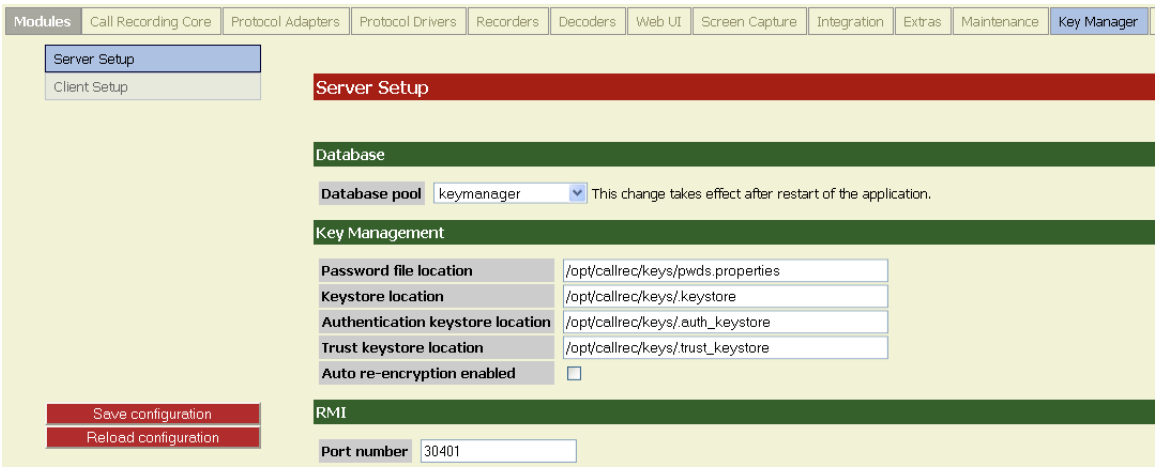


Figure 3: Key Manager configuration – Server Setup

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 (e.g. encryption & decryption clients)

Auto re-encryption enabled: encrypted files will be automatically re-encrypted when their certificates expire – experimental in version 8.0.480.

RMI

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

Client Setup

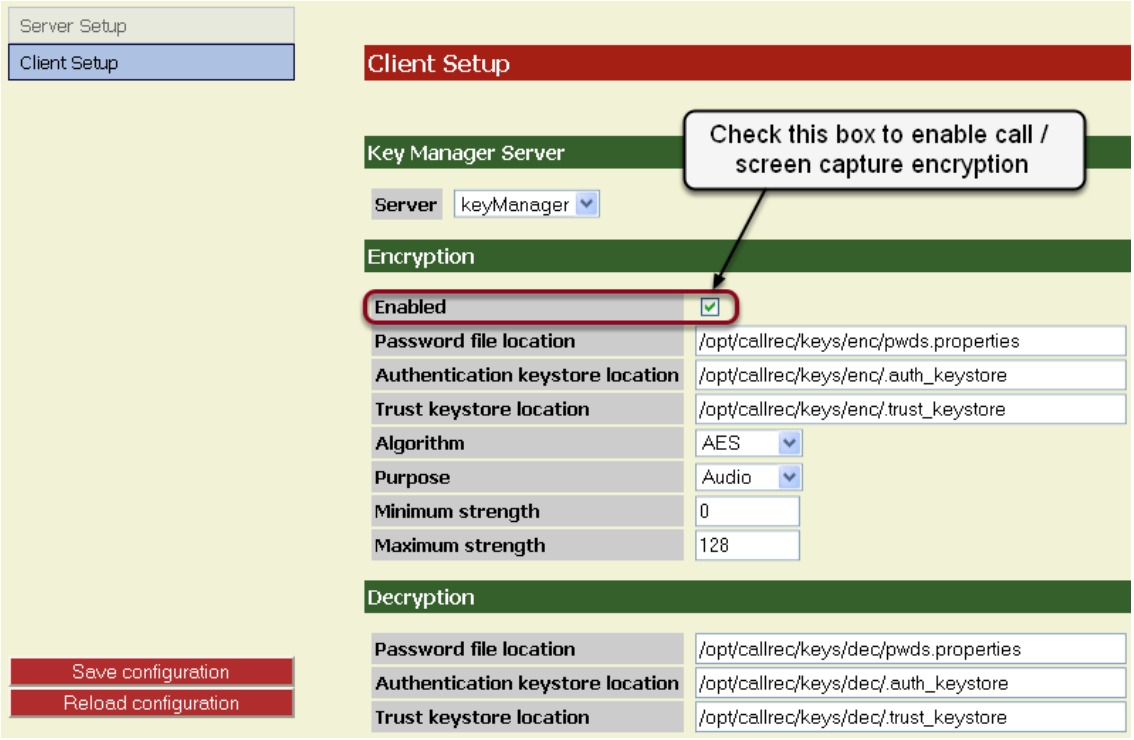


Figure 4: Key Manager configuration – Client Setup

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 expiration / re-creation.

Authentication keystore location: the encryption client authentication key store, containing the client's own private authentication key(s)

Trust keystore location: the encryption client trust key store, containing public authentication key(s) of the trusted server(s)

Algorithm: the type of cipher used for encryption/decryption. Genesys uses AES as standard

Purpose: Specify the keyset to be used for encryption / decryption. The keyset'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 / re-creation.

Authentication keystore location: the decryption client authentication key store, containing the client's own private authentication key(s)

Trust keystore location: the decryption client trust key store, containing public authentication key(s) of the trusted server(s)

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 Genesys Quality Management 8.0.480. 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 Administrator Guide*). Similarly, the Quality Manager audit log can be viewed and exported in Excel format (see the *Quality Manager User Guide (CC Manager)*).

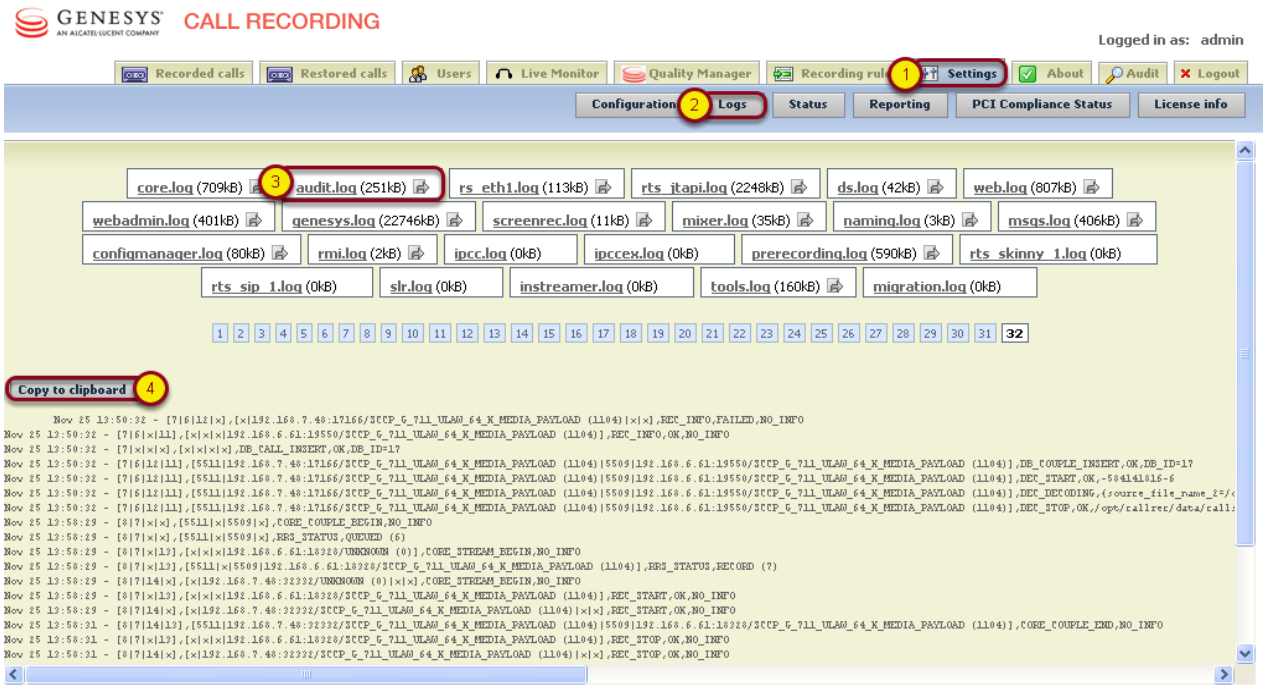


Figure 5: Copying Call Recording audit log data to the clipboard

Resolution: none required

Password management is enforced

Genesys Quality Management 8.0.480 includes advanced password management facilities, which are initially switched off by default, allowing weak passwords to be used.

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.

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

See the screenshot for more details:

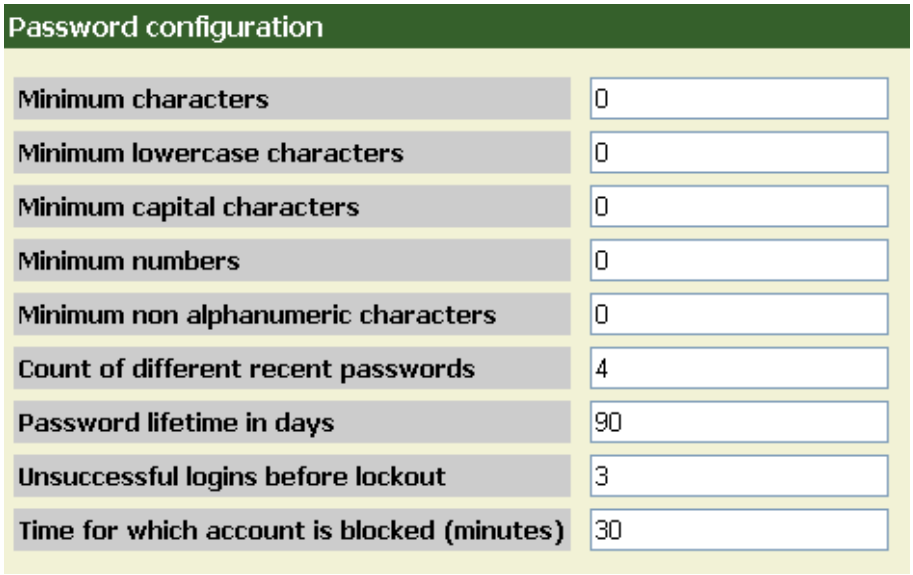


Figure 6: Minimum password configuration for PCI Compliance

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

Resolution: update 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

These settings are found in the Password configuration section at **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; however, it is critical that these are configured according to your MLM requirements – see the *Call Recording Administrator Guide* (*Maintenance* section) for more details.

Archive Tool

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 Administrator Guide* for more details.

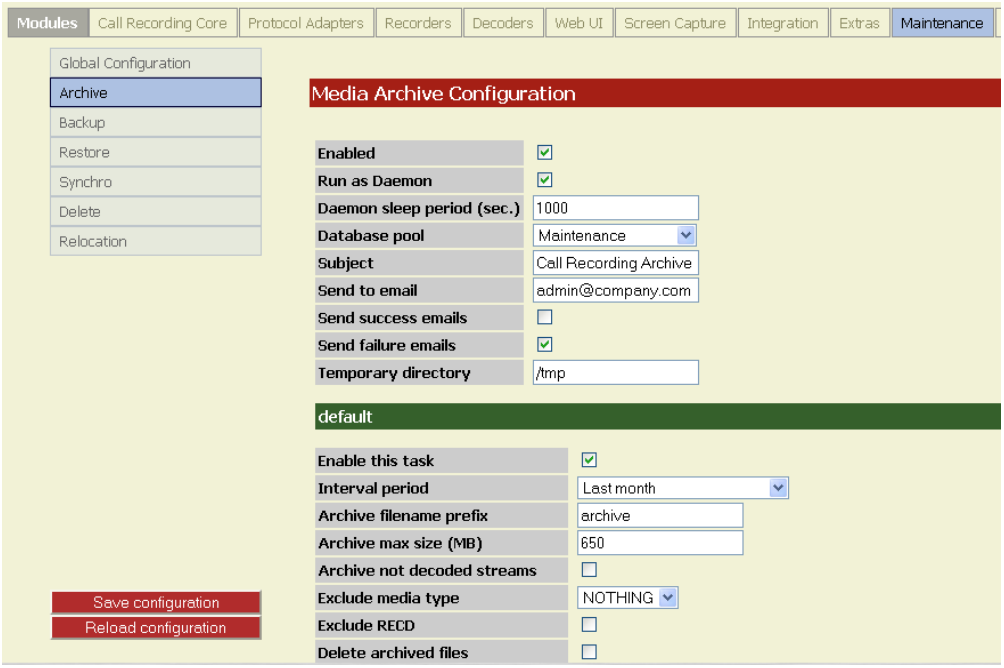


Figure 7: Maintenance settings – Archive tool sample settings

Delete Tool

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 Administrator Guide* for more details.

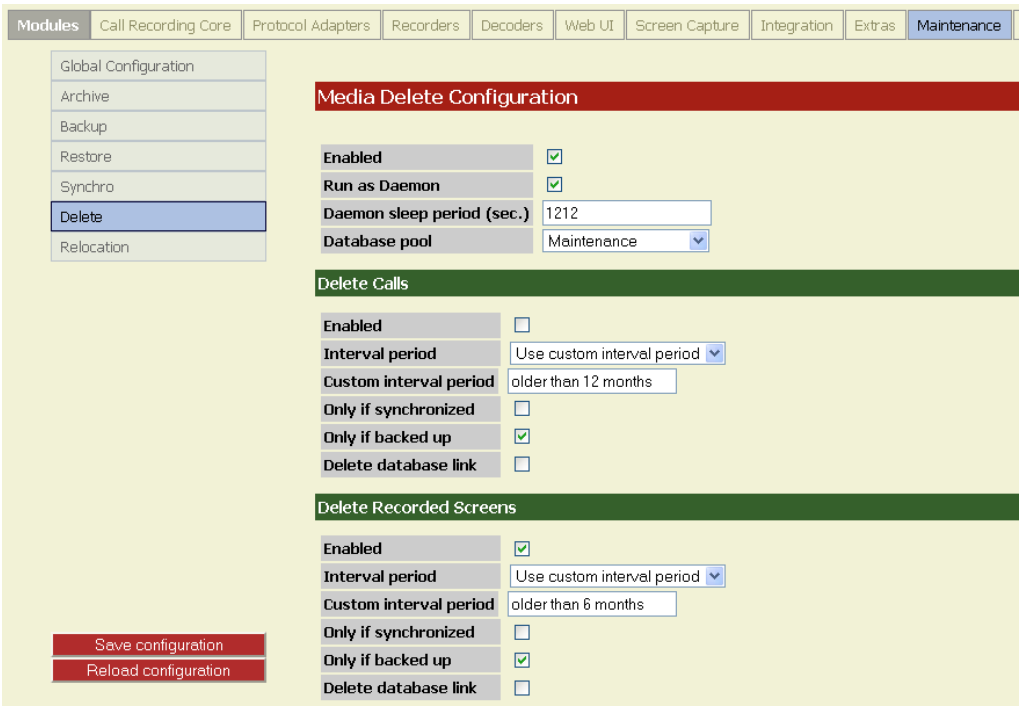


Figure 8: Maintenance settings – Delete tool sample settings

Resolution: enable and configure the Archive and Delete MLM tools in Call Recording Maintenance settings

Chapter

4 Secure Web Access

Genesys Quality Management release 8.0.480 and higher 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.

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 follows:

Key/Certificate Creation

- 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, e.g. `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) [AU]: US
State or Province Name (full name) [Some-State]: California
Locality Name (eg, city) []: San Francisco
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
MyCompany Inc.
Organizational Unit Name (eg, section) []: Call Center
Common Name (eg, YOUR name) []: callrec.mycompany.com
Email Address []: callcenter@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.*
```

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" />
```

Note: 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
```

  …and observe `/var/log/callrec/web.log` for any errors.

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_21/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 Sun's documentation.

Configure Quality Manager Stream URL Setting

- 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, e.g. `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.

- When secure access to the Call Recording Web GUI is functional, 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, e.g.:
  `https://<FQDN>:8443/callrec`

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

  More information on setting up SSL in Apache Tomcat: Tomcat SSL page

Chapter

5 IP Port Use

Warning: 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.

The single server installation uses the following ports:

Port Number	TCP	UDP	Used for
111	X	X	NFS (for replay synchro)
2049	X	X	NFS (for replay synchro)
4001 – 4004	X	X	NFS (for replay synchro)
5432	X	X	PostgreSQL (for replay synchro)
22	X	X	SSH – distant access
80	X	X	GUI – http redirect to 8080
8080	X	X	GUI – http
8443	X	X	GUI – https
443	X		Quality Manager (GUI access)
30400	X		Default RMI port
30300	X		JTAPI Sniffers
389	X	X	LDAP
30500	X		Configuration service (allow it for Live Monitor)
30501	X		Configuration service (allow it for Live Monitor)
30600	X		Core (allow it for Live Monitor)
30601	X		Core (allow it for Live Monitor)
37000-37100		X	Datagrams ports (allow it for Live Monitor)

Note: RMI inter-module communications use random ports in the TCP range: 1024 – 65535.

Chapter

6 Installing Commercial Certificates for Key Manager

The following steps assume that you have not installed self-signed certificates for Key Manager (i.e. replied 'No' to the query during setup). You need to be logged in via SSH as the root user.

If self-signed certificates were installed, it is advised to remove them before attempting to install commercial certificates, to avoid confusion:

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

1. Create keys directory, private keys and certificate request files.

    a. 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
      #
      # Create Self-Signed certificates for Key Manager
      # ZOOM International - QM Suite 4.8
      #
      ####### 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=ZOOM International, OU=ZOOM
International, O=ZOOM International, L=Prague, S=Prague, C=CZ"
      ###################################
  ######## 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
  # 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
```

b. Secondly, 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
```

2. Send the three certificate request files in the `/home/admin` directory to 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`

3. **[OPTIONAL]** Install CA certificate file if CA is not include in the `cacerts` Java keystore.

   a. 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/jdk1.6.0_21/jre/lib/security/cacerts | grep "Owner:"
```

   b. 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/jdk1.6.0_21/jre/lib/security/cacerts -storepass changeit
```

4. Install signed certificates and create encryption/decryption certificates

    a. 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
      #
      # Create Self-Signed certificates for Key Manager - 2
      # ZOOM International - QM Suite 4.8
      #
      ####### 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
  ##########################################


  # 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_21/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_21/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

  ###################################################################
  # WARNING – the following tool has been updated in 8.0.480, so
parameters are different. Please see newer documentation or contact
Support for sample code.
  ###################################################################

  # 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"`
      KEYMAN_PORT="30401"
  $CALLREC_HOME/bin/genkeys -activationDate $ACTIVATION_DATE -algorithm
AES -expirationDate $EXPIRATION_DATE -authStore $ENC_DIR/.auth_keystore
-authStorePassword $CERTIFICATES_PASS -port $KEYMAN_PORT -purpose Audio
-strength 128 -trustStore $ENC_DIR/.trust_keystore -trustStorePassword
$CERTIFICATES_PASS
```

b. Secondly, execute the following two commands to run the file. Note the output below the commands – if you see something similar, certificate installation was successful. Otherwise check the default error file at `/tmp/installcerts.err`.

```
chmod 755 /home/admin/genkeys2.sh
/home/admin/genkeys1.sh
```

Sample output:

```
Certificate stored in file </home/admin/server.cer>
Certificate stored in file </home/admin/encoder.cer>
Certificate stored in file </home/admin/decoder.cer>
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
```

5. [OPTIONAL] Restart Key Manager

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

6. Switch on call encryption in the Call Recording Web GUI (see Client Encryption).

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 CallREC 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:

1. Remove the existing keys and certificates: `rm -f /opt/callrec/keys`
2. Stop CallREC: `service callrec stop`
3. Run CallREC setup again, selecting options to create self-signed certificates if required: `/opt/callrec/bin/callrec-setup`
4. Follow the earlier instructions to install commercial certificates if required, and enable call encryption again
5. If you repeatedly get the same key errors, please contact Support.

Chapter

7 Requesting Technical Support

Prior to contacting Genesys technical support, please refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

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 at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
India	000-800-100-7136 (toll-free) +91-(022)-3918-0537	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp