DJIGZO EMAIL ENCRYPTION

DJIGZO Gateway Quick Install Guide

May 21, 2012, Rev: 6838

Acknowledgments: Thanks goes out to Andreas Hödle for feedback and input on gateway security.

Contents

1 Introduction

This quick install guide explains how to install DJIGZO on Ubuntu, Debian, Red Hat and CentOS. The .deb and .rpm packages have been tested on Ubuntu 8.04, 10.04 & 12.04, Debian 5 & 6, RedHat 5 & 6 and CentOS 5 & 6. For installation on systems not supported by the .deb or .rpm packages, you are advised to use the manual installation guide. It is recommended to install DJIGZO on a dedicated and clean machine.

Requirements

- PostgreSQL
- Postfix
- OpenJDK 6
- ANT, ANT-optional
- Tomcat (or Jetty)

Note: commands that should be executed by the user are shown on lines starting with a *$* sign (the *$* sign is not part of the command to execute). It is recommended to copy and paste the commands directly to the command line.

WARNING do not install DJIGZO on a live email system!

2 Install DJIGZO on Ubuntu & Debian

This section explains how to install DJIGZO on Ubuntu and Debian.

Install required packages[1]

```
$ sudo apt-get install postgresql postfix openjdk-6-jre \
openjdk-6-jre-headless tzdata-java ant ant-optional \
mktemp wget libsasl2-modules
```

Note: during the installation of Postfix, select “No Configuration”.

2.1 Install DJIGZO packages

A full installation of DJIGZO requires the DJIGZO encryption back-end and the Web GUI front-end. The .deb packages can be downloaded from http://www.djigzo.com. The following two files are required: **djigzo_2.4.0-3_all.deb** and **djigzo-web_2.4.0-3_all.deb**. Note that the version can be different when a new version of DJIGZO is released.

[1] The sudo package is required by DJIGZO. Debian does not install sudo by default. If installing on Debian, sudo must be installed prior to installing DJIGZO.

Install the .deb files[2]

```
$ sudo dpkg -i djigzo_2.4.0-3_all.deb
$ sudo dpkg -i djigzo-web_2.4.0-3_all.deb
```

2.2 Configure Postfix

A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email. This requires some changes to the postfix configuration files. DJIGZO installs a pre-configured Postfix main and master configuration file which should be copied to the postfix configuration directory.

WARNING! THIS WILL OVERWRITE ALL SETTINGS IN THE ORIGINAL POSTFIX CONFIG FILES SO ONLY DO THIS IF THE ORIGINAL SETTINGS MAY BE OVERWRITTEN. IF EXISTING POSTFIX SETTINGS MUST BE KEPT YOU SHOULD MERGE THE REQUIRED CHANGES MANUALLY.

Copy postfix configuration files[3]

```
$ sudo cp /etc/postfix/djigzo-main.cf /etc/postfix/main.cf
$ sudo cp /etc/postfix/djigzo-master.cf /etc/postfix/master.cf
```

Update aliases Postfix uses /etc/alias as the alias file. Make sure that the alias file is available and up-to-date.

```
$ sudo newaliases
```

Restart postfix

```
$ sudo /etc/init.d/postfix restart
```

2.3 Install Tomcat[4]

If DJIGZO is installed on Ubuntu 8.04 or Debian 5, Tomcat 5 should be used. If DJIGZO is installed on Ubuntu 10.04 or greater or Debian 6 or greater, Tomcat 6 should be used.

2.3.1 Install Tomcat 5

Install the required Tomcat package (for Tomcat 6, see next section)

```
$ sudo apt-get install tomcat5.5
```

[2] DJIGZO depends on OpenJDK. If you need to use SUN JRE you should use the --ignore-depends parameter to skip installing OpenJDK.

[3] see the manual installation guide on how to configure Postfix if the current Postfix configuration files should not be overwritten.

[4] if you would like to use Jetty instead of Tomcat skip the installation of Tomcat. See Appendix B for instructions on installing Jetty.

Note for Debian users: Tomcat fails on Debian 5 because a suitable JDK cannot be found. See Appendix A for instructions on how to set the JDK path.

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where DJIGZO Web GUI is stored. The property will be added to the Tomcat default config file.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/share/djigzo-web\"" >> /etc/default/tomcat5.5'
```

Configure Tomcat memory usage In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10's of thousands certificates) DJIGZO requires that Tomcat is setup with max. 256 MB heap space.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/default/tomcat5.5'
```

Disable Java security manager DJIGZO currently does not function properly when the Tomcat Java security manager is enabled. The Tomcat Java security manager should therefore be disabled.

```
$ sudo bash -c 'echo "TOMCAT5_SECURITY=no" >> /etc/default/tomcat5.5'
```

Allow reading and writing of SSL certificate DJIGZO Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL import page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

```
$ sudo chown tomcat55:djigzo /usr/share/djigzo-web/ssl/sslCertificate.p12
```

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by DJIGZO, it's advised to replace the existing Tomcat configuration file (/etc/tomcat5.5/server.xml) with the configuration file provided by DJIGZO.

```
$ sudo cp /usr/share/djigzo-web/conf/tomcat/server.xml /etc/tomcat5.5
```

Note: if you want to keep the existing server.xml file, you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/share/djigzo-web/djigzo.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat5.5/Catalina/localhost/djigzo.xml'
```

Note: if you want DJIGZO web admin to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml[5].

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/share/djigzo-web/djigzo-portal.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat5.5/Catalina/localhost/web.xml'
```

Restart Tomcat Tomcat should be restarted to make it use the new Tomcat configuration.

```
$ sudo /etc/init.d/tomcat5.5 restart
```

2.3.2 Install Tomcat 6

Install the required Tomcat package

```
$ sudo apt-get install tomcat6
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where DJIGZO Web GUI is stored. The property will be added to the Tomcat default config file.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/share/djigzo-web\"" >> /etc/default/tomcat6'
```

Configure Tomcat memory usage In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10's of thousands certificates) DJIGZO requires that Tomcat is setup with max. 256 MB heap space.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/default/tomcat6'
```

Disable Java security manager DJIGZO currently does not function properly when the Tomcat Java security manager is enabled. The Tomcat Java security manager should therefore be disabled.

```
$ sudo bash -c 'echo "TOMCAT6_SECURITY=no" >> /etc/default/tomcat6'
```

Allow reading and writing of SSL certificate DJIGZO Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL import page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

```
$ sudo chown tomcat6:djigzo /usr/share/djigzo-web/ssl/sslCertificate.p12
```

[5] the root context allows you to access DJIGZO using a URL of the form https://192.168.178.2/ instead of https://192.168.178.2/djigzo

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by DJIGZO, it's advised to replace the existing Tomcat configuration file (/etc/tomcat6/server.xml) with the configuration file provided by DJIGZO.

```
$ sudo cp /usr/share/djigzo-web/conf/tomcat/server-T6.xml \
/etc/tomcat6/server.xml
```

Note: if you want to keep the existing server.xml file, you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/share/djigzo-web/djigzo.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/djigzo.xml'
```

Note: if you want DJIGZO web admin to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml[6].

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/share/djigzo-web/djigzo-portal.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/web.xml'
```

Restart Tomcat Tomcat should be restarted to make it use the new Tomcat configuration.

```
$ sudo /etc/init.d/tomcat6 restart
```

2.3.3 Finish

Open the Web GUI DJIGZO should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL https://192.168.178.2:8443/djigzo[7] (change the IP address accordingly)

Note: DJIGZO comes with a pre-installed SSL certificate which is not by default trusted by your browser. You should therefore manually accept the SSL certificate.

Login Use the following login credentials:

username:	admin
password:	admin

[6] the root context allows you to access DJIGZO using a URL of the form https://192.168.178.2/ instead of https://192.168.178.2/djigzo

[7] if DJIGZO was installed as the root context, the URL should be https://192.168.178.2:8443

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If DJIGZO is not running, check the following log files for errors:

DJIGZO log

```
$ less /var/log/djigzo.log
```

Tomcat 5 log

```
$ sudo less /var/log/tomcat5.5/catalina.*.log
```

Tomcat 6 log

```
$ sudo less /var/log/tomcat6/catalina.out
```

Note: replace * with the current date to view the most recent log file.

3 Install DJIGZO on Red Hat 5 & CentOS 5

This section explains how to install DJIGZO on Red Hat 5.X and CentOS 5.X. It is assumed that all commands are run as root (i.e., the user is logged in as root).

Configure firewall Red Hat and CentOS by default block access to most ports. The firewall should therefore be configured to allow access to certain ports used by DJIGZO. The following ports should be remotely accessible: *25 (SMTP)* and *8443*[8]. The firewall can be configured with the system-config-securitylevel-tui tool.

```
$ system-config-securitylevel-tui
```

Note: port numbers should be postfixed with *:tcp*. For example, to open port 8443, add *8443:tcp* to the port configuration.

RPM signing keys The RPM packages are signed with a GPG key. Unless RPM and yum are forced not to check signatures, RPM refuses to install packages with invalid or untrusted signatures. It is therefore advised to download and import the GPG key from http://www.djigzo.com/downloads/DJIGZO_SIGNING_GPG.key.

```
$ wget http://www.djigzo.com/downloads/DJIGZO_SIGNING_GPG.key
$ rpm --import DJIGZO_SIGNING_GPG.key
```

Note: alternatively, if you do not want to import the GPG key you can skip checking the signature by adding `--nogpgcheck`.

3.1 Install DJIGZO packages

A full installation of DJIGZO requires the DJIGZO encryption back-end and the Web GUI front-end. The RPM packages can be downloaded from http://www.djigzo.com. The RPM packages should be installed with yum to make sure that all required packages are installed as well (sudo is required and should be installed manually[9])

```
$ yum install djigzo-2.4.0-3.noarch.rpm
$ yum install djigzo-web-2.4.0-3.noarch.rpm
```

Note: the current version can be different when a new version of DJIGZO is released.

[8] See Appendix E.1 for an overview of all ports used by DJIGZO.

[9] This will be fixed on future versions

3.2 Configure PostgreSQL

With the default install of PostgreSQL on RedHat/CentOS, the *autovacuum* service is not enabled. The PostgreSQL vacuum command must be run on a regular basis to keep the database in optimal shape and to make sure that disk space occupied by updated or deleted rows is automatically recovered.
The *autovacuum* service should be enabled by uncommenting and changing the following settings in the PostgreSQL main configuration file.

```
$ vi /var/lib/pgsql/data/postgresql.conf
```

```
stats_start_collector = on
stats_row_level = on
autovacuum = on
autovacuum_naptime = 60
```

After changing these settings, PostgreSQL should be restarted.

```
$ /sbin/service postgresql restart
```

3.3 Configure Postfix

A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email. Red Hat/CentOS installs Sendmail by default. Because DJIGZO requires Postfix we must switch the default MTA from Sendmail to Postfix.

```
$ yum install system-switch-mail
$ system-switch-mail
```

Optionally, if Sendmail is no longer required, Sendmail can be removed.

```
$ yum remove sendmail
```

Configure SELinux If SELinux is enabled (which is by default) Postfix is not allowed to bind to port 10026 (which is used by DJIGZO as the Postfix "reinjection" port). SELinux should be configured to allow Postfix to bind to port 10026. This can be done by creating a file `djigzo.te` with the following content:

```
module djigzo 1.0;

require {
        type postfix_master_t;
        type port_t;
        class tcp_socket name_bind;
}

allow postfix_master_t port_t:tcp_socket name_bind;
```

The SELinux module should be compiled and loaded

```
$ checkmodule -M -m -o djigzo.mod djigzo.te
$ semodule_package -o djigzo.pp -m djigzo.mod
$ semodule -i djigzo.pp
```

Note: alternatively, you can disable SELinux with the system-config-securitylevel-tui tool if you have troubles getting SELinux to work with Postfix and DJIGZO.

Copy Postfix config A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email. This requires some changes to the postfix configuration files. DJIGZO installs a pre-configured Postfix main and master configuration file which should be copied to the postfix configuration directory.

WARNING! THIS WILL OVERWRITE ALL SETTINGS IN THE ORIGINAL POSTFIX CONFIG FILES SO ONLY DO THIS IF THE ORIGINAL SETTINGS MAY BE OVERWRITTEN. IF EXISTING POSTFIX SETTINGS MUST BE KEPT YOU SHOULD MERGE THE REQUIRED CHANGES MANUALLY.

Copy postfix config files[10]

```
$ cp /etc/postfix/djigzo-main.cf /etc/postfix/main.cf
$ cp /etc/postfix/djigzo-master.cf /etc/postfix/master.cf
```

Update aliases Postfix uses /etc/alias as the alias file. Make sure that the alias file is available and up-to-date.

```
$ newaliases
```

Restart postfix

```
$ /sbin/service postfix restart
```

Make mail logs readable The mail logs should be readable by user *djigzo*.

```
$ chmod +r /var/log/maillog
```

3.4 Install Tomcat

```
$ yum install tomcat5
```

[10] see the manual installation guide on how to configure Postfix if current Postfix config files must be kept

Add xalan to endorsed jars Djigzo-web requires xalan jars in the Tomcat endorsed directory.

```
$ rebuild-jar-repository /var/lib/tomcat5/common/endorsed \
xalan-j2-2.7.0.jar

$ rebuild-jar-repository /var/lib/tomcat5/common/endorsed \
xalan-j2-serializer-2.7.0.jar
```

Update Javamail Red Hat/CentOS by default installs an older version of Javamail. The newer version of Javamail provided by DJIGZO will be added as a new alternative.

```
$ alternatives --install /usr/share/java/javamail.jar javamail \
/usr/share/djigzo/lib/mail/mail.jar 20000
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where DJIGZO Web GUI is stored. The property will be added to the Tomcat default config file.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/share/djigzo-web\"" >> /etc/sysconfig/tomcat5
```

Configure Tomcat memory usage In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10‘s of thousands certificates) DJIGZO requires that Tomcat is set to at least 256 MB of heap space.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/sysconfig/tomcat5
```

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by DJIGZO, it's advised to replace the existing Tomcat configuration file (/etc/tomcat5/server.xml) with the configuration file provided by DJIGZO.

```
$ cp /usr/share/djigzo-web/conf/tomcat/server.xml /etc/tomcat5
```

Note: if you want to keep the existing server.xml file, you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ echo "<Context docBase=\"/usr/share/djigzo-web/djigzo.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat5/Catalina/localhost/djigzo.xml
```

Note: if you want DJIGZO web admin to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml[11].

[11] the root context allows you to access DJIGZO using a URL of the form https://192.168.178.2/ instead of https://192.168.178.2/djigzo

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ echo "<Context docBase=\"/usr/share/djigzo-web/djigzo-portal.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat5/Catalina/localhost/web.xml
```

Allow reading and writing of SSL certificate DJIGZO Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL import page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

```
$ chown tomcat:djigzo /usr/share/djigzo-web/ssl/sslCertificate.p12
```

Make Tomcat start at reboot Tomcat should be automatically started at reboot.

```
$ /sbin/chkconfig tomcat5 on
```

3.5 Finalize

Start services

```
$ /sbin/service djigzo restart
$ /sbin/service tomcat5 restart
```

Open the Web GUI DJIGZO should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL https://192.168.178.2:8443/djigzo[12] (change the IP address accordingly)

Note: DJIGZO comes with a pre-installed SSL certificate which is not by default trusted by your browser. You should therefore manually accept the SSL certificate.

Login Use the following login credentials:

username:	admin
password:	admin

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If DJIGZO is not running, check the following log files for errors:

DJIGZO log

```
$ less /var/log/djigzo.log
```

[12] if DJIGZO was installed as the root context, the URL should be https://192.168.178.2:8443

Tomcat log

```
$ less /var/log/tomcat5/catalina.out
```

4 Install DJIGZO on Red Hat 6 & CentOS 6

This section explains how to install DJIGZO on Red Hat 6.X and CentOS 6.X. It is assumed that all commands are run as root (i.e., the user is logged in as root).

Configure firewall Red Hat and CentOS by default block access to most ports. The firewall should therefore be configured to allow access to certain ports used by DJIGZO. The following ports should be remotely accessible: SMTP (25) and 8443[13]. The firewall can be configured with the system-config-firewall-tui tool.

```
$ yum install system-config-firewall-tui
$ system-config-firewall-tui
```

Note: Port 25 can be opened by selecting *Mail (SMTP)* in the *Trusted Services* list. Port 8443 with protocol tcp should be added to the "Other Ports".

4.1 Install PostgreSQL

```
$ yum install postgresql-server
```

A PostgreSQL should be initialized and started.

```
$ /sbin/service postgresql initdb
$ /sbin/service postgresql restart
```

4.2 Install DJIGZO packages

RPM signing keys The DJIGZO RPM packages are signed with a GPG key. Unless RPM and yum are forced not to check signatures, RPM refuses to install packages with invalid or untrusted signatures. It is therefore advised to download and import the GPG key from http://www.djigzo.com/downloads/DJIGZO_SIGNING_GPG.key.

```
$ wget http://www.djigzo.com/downloads/DJIGZO_SIGNING_GPG.key
$ rpm --import DJIGZO_SIGNING_GPG.key
```

Note: alternatively, if you do not want to import the GPG key you can skip checking the signature by adding `--nogpgcheck`.

A full installation of DJIGZO requires the DJIGZO encryption back-end and the Web GUI front-end. The RPM packages can be downloaded from http://www.djigzo.com. The RPM packages should be installed with yum to make sure that all required packages are installed as well.

```
$ yum install djigzo-2.4.0-3.noarch.rpm
$ yum install djigzo-web-2.4.0-3.noarch.rpm
```

[13] See Appendix E.1 for an overview of all ports used by DJIGZO.

Note: the current version can be different when a new version of DJIGZO is released.

4.3 Configure Postfix

A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email.

Configure SELinux If SELinux is enabled (which is by default) Postfix is not allowed to bind to port 10026 (which is used by DJIGZO as the Postfix "reinjection" port). SELinux should be configured to allow Postfix to bind to port 10026. This can be done by creating a file `djigzo.te` with the following content:

```
module djigzo 1.0;

require {
        type postfix_master_t;
        type port_t;
        class tcp_socket name_bind;
}

allow postfix_master_t port_t:tcp_socket name_bind;
```

The SELinux module should be compiled and loaded

```
$ checkmodule -M -m -o djigzo.mod djigzo.te
$ semodule_package -o djigzo.pp -m djigzo.mod
$ semodule -i djigzo.pp
```

Note: alternatively, you can disable SELinux.

Copy Postfix config A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email. This requires some changes to the postfix configuration files. DJIGZO installs a pre-configured Postfix main and master configuration file which should be copied to the postfix configuration directory.

WARNING! THIS WILL OVERWRITE ALL SETTINGS IN THE ORIGINAL POSTFIX CONFIG FILES SO ONLY DO THIS IF THE ORIGINAL SETTINGS MAY BE OVERWRITTEN. IF EXISTING POSTFIX SETTINGS MUST BE KEPT YOU SHOULD MERGE THE REQUIRED CHANGES MANUALLY.

Copy postfix config files[14] RedHat/CentOS 6 comes with Postfix 2.6 and therefore requires an updated master config file.

```
$ cp /usr/share/djigzo/conf/system/master-2.6.cf /etc/postfix/djigzo-master.cf
$ cp /etc/postfix/djigzo-master.cf /etc/postfix/master.cf
$ cp /etc/postfix/djigzo-main.cf /etc/postfix/main.cf
```

Update aliases Postfix uses /etc/alias as the alias file. Make sure that the alias file is available and up-to-date.

```
$ newaliases
```

Restart postfix

```
$ /sbin/service postfix restart
```

Make mail logs readable The mail logs should be readable by user *djigzo*.

```
$ chmod +r /var/log/maillog
```

Configure logrotate By default mail logs are rotated with the date appended to the filename (see *dateext* setting). DJIGZO however expects the rotated log files to be appended with an increasing number. To allow multiple mail log files to be read, modify the logrotate setting for maillog. Create a separate rotate rule for maillog by removing the default rule and appending the following lines to `/etc/logrotate.d/syslog`[15].

```
/var/log/maillog
{
  nodateext
  compress
  delaycompress
  sharedscripts
  postrotate
    /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
  endscript
}
```

Note: don't forget to remove the original `/var/log/maillog` line.

4.4 Install Tomcat

```
$ yum install tomcat6
```

[14] see the manual installation guide on how to configure Postfix if current Postfix config files must be kept

[15] This can be skipped if it is sufficient to only show the most recent log file (/var/log/maillog) directly from the mail log GUI page

Update Javamail Red Hat/CentOS by default installs an older version of Javamail. The newer version of Javamail provided by DJIGZO will be added as a new alternative.

```
$ alternatives --install /usr/share/java/javamail.jar javamail \
/usr/share/djigzo/lib/mail/mail.jar 20000
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where DJIGZO Web GUI is stored. The property will be added to the Tomcat default config file.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/share/djigzo-web\"" >> /etc/sysconfig/tomcat6
```

Configure Tomcat memory usage In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10‘s of thousands certificates) DJIGZO requires that Tomcat is set to at least 256 MB of heap space.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/sysconfig/tomcat6
```

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by DJIGZO, it’s advised to replace the existing Tomcat configuration file (/etc/tomcat6/server.xml) with the configuration file provided by DJIGZO.

```
$ cp /usr/share/djigzo-web/conf/tomcat/server-T6.xml /etc/tomcat6/server.xml
```

Note: if you want to keep the existing server.xml file, you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ echo "<Context docBase=\"/usr/share/djigzo-web/djigzo.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/djigzo.xml
```

Note: if you want DJIGZO web admin to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml[16].

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ echo "<Context docBase=\"/usr/share/djigzo-web/djigzo-portal.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/web.xml
```

[16] the root context allows you to access DJIGZO using a URL of the form https://192.168.178.2/ instead of https://192.168.178.2/djigzo

Allow reading and writing of SSL certificate DJIGZO Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL import page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

```
$ chown tomcat:djigzo /usr/share/djigzo-web/ssl/sslCertificate.p12
```

Make Tomcat start at reboot Tomcat should be automatically started at reboot.

```
$ /sbin/chkconfig tomcat6 on
```

4.5 Finalize

Start services

```
$ /sbin/service djigzo restart
$ /sbin/service tomcat6 restart
```

Open the Web GUI DJIGZO should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL https://192.168.178.2:8443/djigzo[17] (change the IP address accordingly)

Note: DJIGZO comes with a pre-installed SSL certificate which is not by default trusted by your browser. You should therefore manually accept the SSL certificate.

Login Use the following login credentials:

username:	admin
password:	admin

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If DJIGZO is not running, check the following log files for errors:

DJIGZO log

```
$ less /var/log/djigzo.log
```

Tomcat log

```
$ less /var/log/tomcat6/catalina.out
```

[17] if DJIGZO was installed as the root context, the URL should be https://192.168.178.2:8443

A Configure Tomcat on Debian 5

Tomcat on Debian 5 cannot start because a suitable JDK is not found:

```
no JDK found - please set JAVA_HOME failed!
```

The JDK path should be set in /etc/default/tomcat:

```
$ sudo bash -c 'echo "JAVA_HOME=/usr/lib/jvm/java-6-openjdk" >> \
/etc/default/tomcat5.5'
```

B Using Jetty 6

This appendix will explain how to configure Jetty 6 for DJIGZO. This guide will only explain how to install Jetty on Ubuntu. For installation instructions on installing Jetty on non-Ubuntu systems please see http://jetty.codehaus.org/jetty/. Configuration of Jetty for DJIGZO should be similar for all Jetty installations.

Note: the latest .deb releases of Jetty can only be installed on Ubuntu 8.10 and newer versions of Ubuntu. If you need to install Jetty 6 on a previous version of Ubuntu you either need to use an older version of Jetty (for example 6.1.17) or use the non-deb version.

Install the required packages

```
$ sudo apt-get install libservlet2.5-java
```

Download Jetty[18]

```
$ wget http://www.djigzo.com/downloads/jetty6_6.1.22-1_all.deb
$ wget http://www.djigzo.com/downloads/libjetty6-java_6.1.22-1_all.deb
```

Install the deb files

```
$ sudo dpkg -i jetty6_*_all.deb libjetty6-java_6.*_all.deb
```

Enable automatic startup By default Jetty is not automatically started at reboot. To make sure that Jetty is started at system startup replace `NO_START=1` with `NO_START=0` in file /etc/default/jetty6.

```
$ sudo sed s/NO_START\s*=\s*1/NO_START=0/ /etc/default/jetty6 --in-place
```

[18] other Jetty releases can be downloaded from http://dist.codehaus.org/jetty/

Configure Jetty Copy the required Jetty configuration files

```
$ sudo cp /usr/share/djigzo-web/conf/jetty/djigzo-jetty-ssl.xml \
/etc/jetty6/

$ sudo cp /usr/share/djigzo-web/conf/jetty/djigzo-jetty-context.xml \
/etc/jetty6/contexts/
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where DJIGZO Web GUI is stored. The property will be added to the Jetty default config file.

```
$ sudo bash -c 'echo "JAVA_OPTIONS=\"\$JAVA_OPTIONS -Ddjigzo-web.home=\
/usr/share/djigzo-web\"" >> /etc/default/jetty6'
```

Load SSL config

```
$ sudo sed $\a\/etc/jetty6/djigzo-jetty-ssl.xml \
/etc/jetty6/jetty.conf --in-place
```

Allow reading and writing of SSL certificate DJIGZO Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL import page. To support this functionality, Jetty should be allowed to read and write the SSL certificate.

```
$ sudo chown jetty /usr/share/djigzo-web/ssl/sslCertificate.p12
$ sudo chmod 660 /usr/share/djigzo-web/ssl/sslCertificate.p12
```

Restart Jetty

```
$ sudo /etc/init.d/jetty6 restart
```

Open the Web GUI DJIGZO should now be running (wait some time for Jetty to startup). The login page can be accessed using the following URL https://192.168.178.2:8443 (change the IP address accordingly)

Note: DJIGZO comes with a pre-installed SSL certificate which is not by default trusted by your browser. You should therefore manually accept the SSL certificate.

Login Use the following login credentials:

username:	admin
password:	admin

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

C Adding Tomcat HTTPS connector

DJIGZO uses the following Tomcat server.xml configuration files.

C.1 Tomcat 5

```
<Server>
    <Service name="Catalina">
        <Connector port="8443" maxHttpHeaderSize="8192"
                   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
                   enableLookups="false" disableUploadTimeout="true"
                   acceptCount="100" scheme="https" secure="true"
                   clientAuth="false" sslProtocol="TLS"
                   keystoreFile="/usr/share/djigzo-web/ssl/sslCertificate.p12"
                   keystorePass="djigzo"
                   keystoreType="PKCS12"
                   ciphers=
                      "TLS_RSA_WITH_AES_128_CBC_SHA,
                       TLS_RSA_WITH_AES_256_CBC_SHA,
                       SSL_RSA_WITH_3DES_EDE_CBC_SHA,
                       SSL_RSA_WITH_RC4_128_SHA"
        />

        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" unpackWARs="false"/>
        </Engine>
    </Service>
</Server>
```

C.2 Tomcat 6

```
<?xml version="1.0" encoding="UTF-8"?>
<Server>
    <Service name="Catalina">
        <Connector port="8443" maxHttpHeaderSize="8192"
                   maxThreads="150"
                   minSpareThreads="25"
                   maxSpareThreads="75"
                   enableLookups="false"
                   disableUploadTimeout="true"
                   acceptCount="100"
                   scheme="https"
                   secure="true"
                   clientAuth="false"
                   SSLEnabled="true"
                   sslProtocol="TLS"
                   protocol="HTTP/1.1"
                   keystoreFile="/usr/share/djigzo-web/ssl/sslCertificate.p12"
                   keystorePass="djigzo"
                   keystoreType="PKCS12"
                   ciphers=
                      "TLS_RSA_WITH_AES_128_CBC_SHA,
                       TLS_RSA_WITH_AES_256_CBC_SHA,
                       SSL_RSA_WITH_3DES_EDE_CBC_SHA,
                       SSL_RSA_WITH_RC4_128_SHA"
        />

        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" unpackWARs="false"/>
        </Engine>
    </Service>
</Server>
```

Note: If an existing server.xml should be used, the Connector for port 8443 should be added to the existing server.xml.

D Memory usage

Because of some limitations of Javamail, DJIGZO requires a lot of memory when it needs to encrypt large messages. By default DJIGZO is setup with a maximum heap size of 512 MB which should be enough for sending messages up to 50 MB. If you need to send larger messages, you need to increase the maximum heap size by modifying the property **wrapper.java.maxmemory** in file /etc/djigzo/djigzo.wrapper.conf. Alternatively, you can set the memory based on the total available memory by adding the following lines to /etc/default/djigzo[19]:

```
RESERVE=384
# On 32 bits Linux Java max mem is little less than 2048.
# on 64 bits Linux you can use much more memory.
# Set DJIGZO_MAX_MEM to the memory you want DJIGZO to use
DJIGZO_MAX_MEM=$(( $(free -m | grep "Mem:" | awk '$2 > (2024 + \
'$RESERVE') { $2 = (2024 + '$RESERVE') } { print $2 }') - $RESERVE ))

echo "* DJIGZO max memory setting: $DJIGZO_MAX_MEM MB"

# Additional options that are passed to the Daemon.
WRAPPER_OPTS=$WRAPPER_OPTS" wrapper.java.maxmemory=$DJIGZO_MAX_MEM"
```

This will ensure that the maximum heap size of DJIGZO is set to the total available memory minus 384 MB (you can change the value of `RESERVE` if other processes on the system require more memory)

[19] /etc/default/djigzo that comes with DJIGZO already contains these lines. They are however commented out

E Securing the gateway

E.1 Port usage

DJIGZO uses the following ports:

external → internal

Port	Service	Description
22	SSH	Console access
25	SMTP	Send/Receive email
8080	HTTP	Web manager
8443	HTTPS	Web manager
9000	SOAP (HTTP)	Back end*

* By default the back-end SOAP service is only accessible from localhost (i.e., it is bound to localhost)

internal → external

Port	Service	Description
25	SMTP	Send/Receive email
80	HTTP	CRL download
139	SMB/CIFS	remote backup and restore
398	LDAP	CRL download
443	HTTPS	CRL download
445	SMB/CIFS	remote backup and restore

When the encryption back-end and Web GUI front-end are installed on the same machine, remote access to port 9000 is not required. It is advised to block remote access to all ports which are not used by DJIGZO.

Enable Ubuntu firewall Ubuntu can be protected by installing the "Uncomplicate Firewall" (UFW) with the following commands:

```
$ sudo apt-get install ufw
$ sudo ufw allow smtp/tcp
$ sudo ufw allow ssh/tcp
$ sudo ufw allow 8443/tcp
$ sudo ufw allow 8080/tcp
$ sudo ufw enable
```

Red Hat/CentOS already comes with a pre-installed firewall.

E.2 Passwords

Database By default, DJIGZO creates a database user *djigzo* with the password *djigzo*. If a different password should be used, the database password for user *djigzo* should be changed (see PostgreSQL documentation). The database password in the database configuration file /usr/share/djigzo/conf/hibernate.cfg.xml should be changed accordingly.

Back-end The front-end (Web GUI) communicates with the back-end (encryption engine) using password authenticated SOAP messages. If the back-end and front-end are not installed on the same machine, it is advised to change the SOAP password.

For the back-end, the password can be changed by modifying the property **protected.system.soap.password** in file /usr/share/djigzo/conf/djigzo.properties. If the password for the back-end is changed, the password used by the front-end should be changed accordingly. The password for the front-end can be changed by adding a property **soap.password** with the password as the property value to /etc/default/tomcat5.5[20] in a similar way as **djigzo-web.home** was set (see 4.4).

E.3 SSL certificate

Access to the administration page is protected with an encrypted HTTPS connection. DJIGZO comes with a default SSL certificate. It is advised to install a new SSL certificate using the “SSL certificate manager” from the DJIGZO Web GUI.

E.4 Prevent spoofing the From header

DJIGZO uses the *From* header as the identity of the sender. If the DJIGZO gateway is used for sending email to external recipients (i.e., relaying email), make sure that internal users cannot ‘spoof’ the *From* header.

E.5 Securing the database

Unless a “Hardware Security Module” (HSM) is used, all private keys used for signing and decrypting of email are stored in the database. The database therefore has to be protected against unauthorized access. If DJIGZO and PostgreSQL are installed on the same machine, the djigzo database user should only be allowed to access the database locally. This is done by making sure that only localhost (127.0.0.1) can login with the username *djigzo*. The PostgreSQL config file `pg_hba.conf` should contain a line similar to:

```
host  djigzo  djigzo  127.0.0.1/32  md5
```

E.6 Block access to pages

If the PDF reply functionality is used, external access to the gateway should be granted to all external IP addresses (otherwise the recipients of the encrypted PDF message cannot open the reply page). It is advised to only allow access to the PDF reply pages and block access to all other pages.

[20] If Jetty is used instead you should add the property to /etc/default/jetty6. If Tomcat 6 is used add the property to /etc/default/tomcat6

Access to the following URLs should be granted for all IP addresses: `https://192.168.178.24:8443/web/portal/*` (the IP address should be the external IP address and * means that access should be granted to all parent URLs). There are multiple ways to block access to most of the gateway pages while allowing access to the PDF reply page:

Block access with a firewall If a firewall is used and the firewall is capable of blocking access at the HTTP(s) level, a rule should be added to block access to all URLs with the exception of the PDF reply page URL.

Use Apache as a front-end Use Apache as a front-end to the gateway. Apache will handle all HTTP(s) access. Apache can be setup to only allow access to certain URLs. Add a rule to block access to all URLs except to the PDF reply page URL.

Enable the built-in IP filter DJIGZO can be setup to only allow access to the management pages from certain IP ranges. A property **djigzo.ipfilter.network** with the IP filter as the property value should be added to /etc/default/tomcat6[21] in a similar way as **djigzo-web.home** was set (see 4.4).

Example: -Ddjigzo.ipfilter.network=192.168.178.20

Filter The IP filter should be a comma separated list if IP ranges. Some example filters:

a) 192.168.*
b) 192.168.*, 127.*, 222.0.0.0/8

[21] For Red Hat/CentOS the property should be added to /etc/sysconfig/tomcat5. If Jetty is used you should add the property to /etc/default/jetty6