

DJIGZO EMAIL ENCRYPTION

Djigzo Gateway Quick Install Guide



Author: Martijn BRINKERS

March 29, 2010, Rev: 4097

Copyright © 2008-2010, Martijn Brinkers.

Acknowledgments: I would like to thank Andreas Hödle for feedback and input on gateway security and Christine Karman for input, feedback and proof-reading.

Contents

1	Introduction	3
2	Install Djigzo on Ubuntu	3
2.1	Install Djigzo packages	3
2.2	Configure Postfix	4
2.3	Install Tomcat	4
3	Install Djigzo on Red Hat/CentOS	7
3.1	Install Djigzo packages	7
3.2	Configure Postfix	7
3.3	Install Tomcat	9
A	Configure Tomcat on Debian 5	12
B	Using Jetty 6	12
C	Adding Tomcat HTTPS connector	14
D	Memory usage	15
E	Securing the gateway	16
E.1	Port usage	16
E.2	Passwords	16
E.3	SSL certificate	17
E.4	Prevent spoofing the From header	17
E.5	Securing the database	17
E.6	Block access to pages	17

1 Introduction

This quick install guide will tell you how to install Djigzo on Ubuntu, Red Hat or CentOS. The .deb and .rpm packages have been tested on Ubuntu 8.04, Debian 5 and RedHat/CentOS 5.4. For installation on systems not supported by the .deb or .rpm packages you are advised to use the manual installation guide. You are 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). You can copy and paste the commands to the command line.

WARNING do not install Djigzo on a live email system!

2 Install Djigzo on Ubuntu

This section explains how to install Djigzo on Ubuntu 8.04 and Debian.

Install required packages

```
$ 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 you need to choose a configuration. Select “No Configuration”.

2.1 Install Djigzo packages

A full installation of Djigzo requires you to install the Djigzo engine and the web administration manager. The debian packages can be downloaded from <http://www.djigzo.com>. You need to download the following two files: **djigzo_1.3.2-1_all.deb** and **djigzo-web_1.3.2-1_all.deb**. Note that the current version can be different in your case.

Install the .deb files¹

```
$ sudo dpkg -i djigzo_1.3.2-1_all.deb  
$ sudo dpkg -i djigzo-web_1.3.2-1_all.deb
```

¹Djigzo depends on OpenJDK. If you need to use SUN JRE you should use the --ignore-depends parameter to skip installing OpenJDK.

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 comes with a modified postfix main and master config file which should be copied to the postfix config 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 files²

```
$ 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/aliases` 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 Tomcat³

Install the required Tomcat package

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

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` must reference the location where Djigzo web application 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 set to at least 256 MB of heap space.

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

²see the manual installation guide on how to configure Postfix if current Postfix config files must be kept

³if you would like to use Jetty instead of Tomcat skip the installation of Tomcat. See Appendix B for instructions on installing Jetty.

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 If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, Tomcat should be allowed to read and write the SSL certificate.

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

Adding a HTTPS connector A HTTPS connector must be added to the Tomcat server configuration. If the Tomcat installation is only used for Djigzo, you are 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` you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding a context A Djigzo context must now be added to Tomcat.

```
$ 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 to use the root context save the file to `ROOT.xml` instead of `djigzo.xml`⁴.

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

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

Open the Web Admin page Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL⁵ <https://192.168.178.2:8443/djigzo> (change the IP address accordingly)

Note: Djigzo comes with a default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

Login Use the following login credentials:

```
username: admin
password: admin
```

⁴the 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`

⁵if Djigzo was installed as the root context the URL should be `https://192.168.178.2:8443`

Note: it can take some time to login after a restart because the web application need to be initialized upon first login.

Log output In case Djigzo is not running you can check the log files for errors.

Djigzo log

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

Tomcat log⁶

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

⁶replace * with the current date to view the most recent log file.

3 Install Djigzo on Red Hat/CentOS

This section explains how to install Djigzo on Red Hat 5.4 and CentOS 5.4. It is assumed that all commands are run as root (ie. the user is logged in as root).

Configure firewall Red Hat and CentOS by default blocks access to most ports. The firewall must be configured to allow access to certain ports used by Djigzo. The following ports must be remotely accessible: 25 (*SMTP*) and 8443⁷. The firewall can be configured with the `system-config-securitylevel-tui` tool.

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

Note: ports, like port 8443, must be postfixed with `:tcp`. For example: `8443:tcp`.

RPM signing keys The RPM packages are signed with a GPG key. Unless RPM and yum are told not to check signatures, RPM refuses to install a package when the signature is invalid or not trusted. You are therefore advised to download and import the GPG key from http://www.djigzo.com/downloads/MARTIJN_BRINKERS_GPG.key.

```
$ wget http://www.djigzo.com/downloads/MARTIJN_BRINKERS_GPG.key
$ rpm --import MARTIJN_BRINKERS_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 you to install the Djigzo engine and the web administration manager. 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-1.3.2-1.noarch.rpm
$ yum install djigzo-web-1.3.2-1.noarch.rpm
```

Note: the current version can be different in your case.

3.2 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.

```
$ system-switch-mail
```

Optionally if Sendmail is no longer required you can remove Sendmail.

```
$ yum remove sendmail
```

⁷See Appendix E.1 for an overview of all ports used by Djigzo.

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 “re-injection” 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 must now 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 Djigzo requires some changes to the postfix configuration files. Djigzo comes with a modified postfix main and master config file which should be copied to the postfix config 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 files⁸

```
$ 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/aliases` as the alias file. Make sure that the alias file is available and up-to-date.

```
$ newaliases
```

Restart postfix

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

⁸see the manual installation guide on how to configure Postfix if current Postfix config files must be kept

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

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

3.3 Install Tomcat

```
$ yum install tomcat5.i386
```

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
```

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

Set djigzo-web.home The system property **djigzo-web.home** must reference the location where Djigzo web application 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 a HTTPS connector A HTTPS connector must be added to the Tomcat server configuration. If the Tomcat installation is only used for Djigzo, you are 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 you need to manually add the HTTPS Connector. See Appendix C for more information.

Adding a context A Djigzo context must now be added to Tomcat.

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

Note: if you want Djigzo to use the root context save the file to ROOT.xml instead of djigzo.xml⁹.

Allow reading and writing of SSL certificate If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, 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
```

Start Tomcat

```
$ /sbin/service tomcat5 start
```

Open the Web Admin page Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL¹⁰ <https://192.168.178.2:8443/djigzo> (change the IP address accordingly)

Note: Djigzo comes with a default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

Login Use the following login credentials:

```
username: admin
password: admin
```

Note: it can take some time to login after a restart because the web application need to be initialized upon first login.

Log output In case Djigzo is not running you can check the log files for errors.

Djigzo log

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

Tomcat log

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

⁹the 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>

¹⁰if Djigzo was installed as the root context the URL should be <https://192.168.178.2:8443>

Note: OpenJDK installed by Red Hat/CentOS 5.4 already contains the “unlimited strength JCE policy files”.

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 higher. 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 libervlet2.5-java
```

Download Jetty¹¹

```
$ 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
```

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/
```

¹¹other Jetty releases can be downloaded from <http://dist.codehaus.org/jetty/>

Set djigzo-web.home The system property **djigzo-web.home** must reference the location where Djigzo web application 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 If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, 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 Admin page 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 default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

Login Use the following login credentials:

```
username: admin
password: admin
```

Note: it can take some time to login after a restart because the web application will be initialized upon first login.

C Adding Tomcat HTTPS connector

The version of Tomcat server.xml that comes with Djigzo:

```
<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,
        TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
        TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
        TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
        TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
        TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
        TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA"
    />

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

If you want to use an existing server.xml you must add the Connector for port 8443 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`¹²:

```
RESERVE=320
# 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 320 MB (you can change the value of `RESERVE` if other processes on the system require more memory)

¹²`/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 for external → internal (listening ports)

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

and for 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

With a default setup of Djigzo, ie. the encryption back-end and web front-end are running on the same local machine, remote access to port 9000 is not required. You are advised to block remote access to all ports that need not be externally accessible.

Enable Ubuntu firewall Ubuntu can be protected by installing the “Uncomplicated 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 you would like to change the password you should change the database password for user *djigzo* (see PostgreSQL documentation) and change the password in the file `/user/share/djigzo/conf/hibernate.cfg.xml`.

Back-end The front-end (Web application) communicates with the back-end (encryption engine) using password authenticated SOAP messages. If the back-end and front-end are running on separate systems you are advised to change

the SOAP password. For the back-end the password should be changed in the file `/usr/share/djigzo/conf/djigzo.properties` (see property **protected.system.soap.password**). You can change the password for the front-end by adding a property **soap.password** with the password as the property value to `/etc/default/tomcat5.5`¹³ in a similar way as **djigzo-web.home** was set (see 3.3).

E.3 SSL certificate

Access to the administration page is protected with an encrypted HTTPS connection. Djigzo comes with a default SSL certificate. You are advised to install your own SSL certificate using the “SSL certificate manager” from the Djigzo web administration page.

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 (ie. relaying email) you should 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 the database are on the same machine the djigzo database user should not be allowed to access the database across the network. 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). You are advised to only allow access to the PDF reply pages and block access to all other pages. Access to the following URLs should be granted for all IP addresses: <https://192.168.178.24:8443/external/pdf/>* (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 your firewall is capable of blocking access at the HTTP(s) level you should add a rule to block access to all URLs except to the PDF reply page URL.

¹³If Jetty is used instead you should add the property to `/etc/default/jetty6`

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/tomcat5.5`¹⁴ in a similar way as `djigzo-web.home` was set (see 3.3).

Example: `-Ddjigzo.ipfilter.network=192.168.178.20`

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

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

Use a dedicate WAR Another option is to install an alternative Web Application Archive (WAR) with only support for the PDF reply functionality. Instead of using `djigzo.war` in the web context (see section 2.3) `djigzo-external.war` should be used instead.

¹⁴For 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`