

Informative article

September, 2003.

## **Implementing your own PGP key server.**

stijn.huyghe@thti.telindus.be

### **Introduction**

In this article we are going to describe the setup and configuration of a basic PGP key server by using the OpenPGP Public Key Server (<http://pks.sourceforge.net>) on an UNIX platform (Red Hat 9).

The PGP key server is compatible with PGP 8.x from pgp.com.

This article is not going to explain how PGP works. For this, you might consult some online information like "How PGP works" from PGP International (<http://www.pgpi.org/doc/pgpintro/>).

### **Installation of the OpenPGP key server on Red Hat 9.**

Grab the OpenPGP public key server sources from <http://pks.sourceforge.net/downloads.html>. At this moment, the current release is "pks-0.9.6" (pks-0.9.6.tar.gz). Verify the signature on the sources before installing!

Copy the sources to your favorite directory and extract them:

- "gzip -d pks-0.9.6.tar.gz".
- "tar -xvf pks-0.9.6.tar.gz".

Go into the "pks-0.9.6" directory and take a look at the README file and the different MAN pages for the possible configurations and options!

Compiling the source code should not result in any problems at all. In order to do this, issue at the command prompt:

- "./configure (--prefix=).
- "make".
- "make install".

### **Basic configuration of the OpenPGP key server.**

The first thing your PGP server needs is a database to store its information. Suppose that you don't have a database or key ring at this time, create an empty database by using the command "pksclient".

To create the empty database, issue: "pksclient /var/pgpdbname create" (placing the database in the directory "/var/pgpdbname").

```
[root@anubis root]# pksclient
pksclient /db/path create [num_files]
pksclient /db/path recover
pksclient /db/path add filename [flags]
pksclient /db/path get userid [flags]
pksclient /db/path index userid [flags]
pksclient /db/path since time [flags]
pksclient /db/path delete userid [flags]
pksclient /db/path disable userid [flags]
```

Figure 1 "pksclient".

```
[root@anubis pgpbase]# ls
db_lock.share  db_mpool.share  keydb000  keydb002  num_keydb  wordddb
db_log.share   db_txn.share    keydb001  log.0000000001  timedb
```

Figure 2 Dbase directory.

The configuration file of the server daemon is pksd.conf, let's take a look at a simple configuration:

```
[root@anubis pks-0.9.6]# cat < pksd.conf
pks_bin_dir /usr/local/bin
### specify where you are storing your database.
db_dir /var/pgpbase
### www directory.
www_dir /usr/local/var
### Set www_port to the port on which HTTP requests should be accepted.
### If you do not want to process HTTP requests, set this to 0.
### we want to process HTTP requests!
www_port 8080
### Set www_readonly to 0 if you want to allow ADD requests over HTTP
### we want be able to add requests over HTTP.
www_readonly 0
socket_name /usr/local/var/pksd_socket
### Specify the envelope sender address as the -f argument to
### sendmail. This is the address which will receive any bounces.
### If you don't use sendmail, then change this to an equivalent command.
### If you do not want to process mail requests, leave this unset.
### we do not want to process mail requests!
#mail_delivery_client sendmail -t -oi -fmailer-daemon
### Set this to the address which should be displayed as the From:
### address in all outgoing email, and as the maintainer in the body
### of each message.
maintainer_email PGP Key Server Administrator <nobody>
mail_intro_file /usr/local/share/mail_intro
help_dir /usr/local/share
mail_dir /usr/local/var/incoming
### If you change this, make sure to put a corresponding help file in
### the help_dir named above
default_language EN
### This is the email address of this site. It will be inserted in all
```

```
### outgoing incremental messages, so it should match whatever the
### downstream sites use as syncsite in their pkzd.conf files.
# this_site pgp-public-keys@your-site
### Include a syncsite line for each site with which you are exchanging
### incremental requests.
# syncsite pgp-public-keys@pgp-server-1
# syncsite pgp-public-keys@pgp-server-2
### Set this to 0 to disable mailserver LAST requests completely, to a
### positive integer to limit LAST requests to that many days, or -1
### to allow any argument to LAST.
# max_last -1
### Set this to the maximum number of keys to return in the reply to
### a last query. Setting it to -1 will allow any size reply.
### WARNING: set the maximum allowed numbers of keys in a reply.
### choose a good value for your environment!
### In this lab setup, we allow any size reply.
max_last_reply_keys -1
### Set this to the maximum number of keys to return in the reply to
### an index, verbose index, or get query. Setting it to -1
### will allow any size reply.
### WARNING: set the maximum allowed number of keys to return in a
### reply for indexes/queries. Choose a good value for your environment!
### in this lab setup, we allow any size reply!
max_reply_keys -1
```

## Running the key server.

If your configuration is ok, you can run the PGP key server by the command “pkzd your.config” (e.g. “./pkzd pkzd.conf”).  
When everything is ok, you should be able to visit your PGP key server with your web browser.



Figure 3 Browsing to the key server.

## Working with the key server.

In this section, we are going to perform some basic actions with the PGP key server and a PGP client, in our case, PGP 8.0.2 from PGP.com.

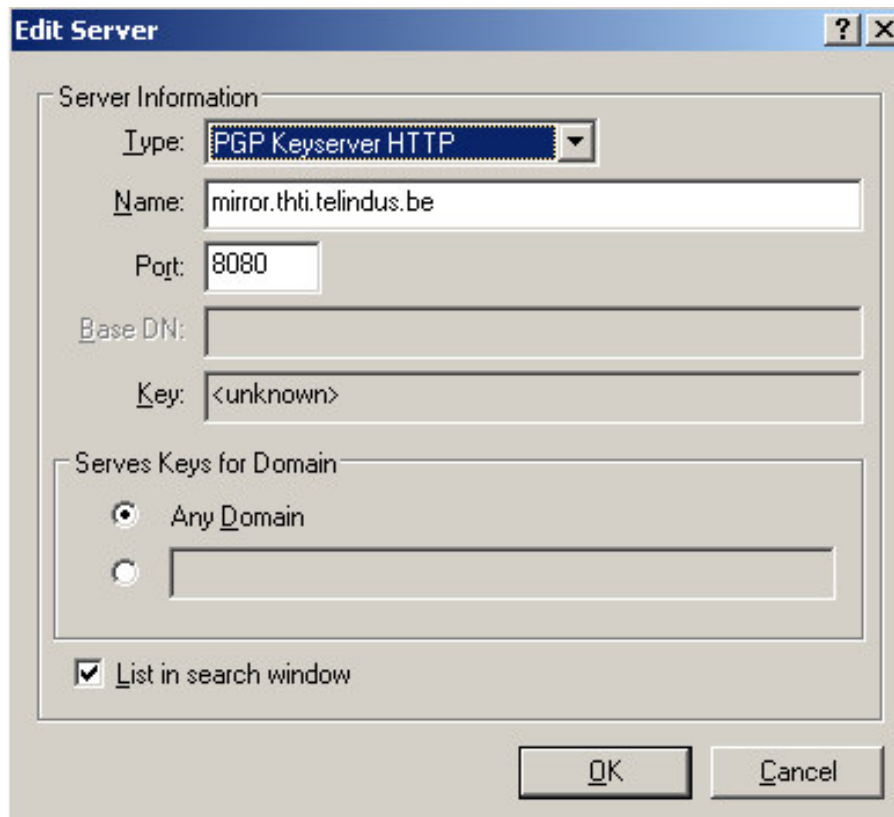
### Adding the PGP server to the PGP client software.

If you want to work with the PGP server, you also need add this server to your PGP client configuration so that it knows it is available.

For this, from your PGP task bar, choose “options” → “Servers” → “New”.

Below you can see an example configuration of our lab server.

Note that its type is “PGP Keyserver HTTP”!



**Figure 4 Adding the PGP server.**

### Submitting a key to the key server.

If you want to submit keys to the key server, you have two options:

- you can submit your keys online through the web interface (when allowed by the configuration).
- you can submit your keys through your PGP client software.

In order to do this, choose “PGPKeys” → Select your PGP key you want to upload → “Send to” → Choose your HTTP server.

## Submitting a new key to the server

Here is how to add a key to the server's keyring:

1. Cut-and-paste an ASCII-armored version of your public key into the text box.
2. Press "Submit".

That is it! The keyserver will process your request immediately. If you like, you can check that y

---

Enter ASCII-armored PGP key here:



**Figure 5** Submitting keys through the web interface.

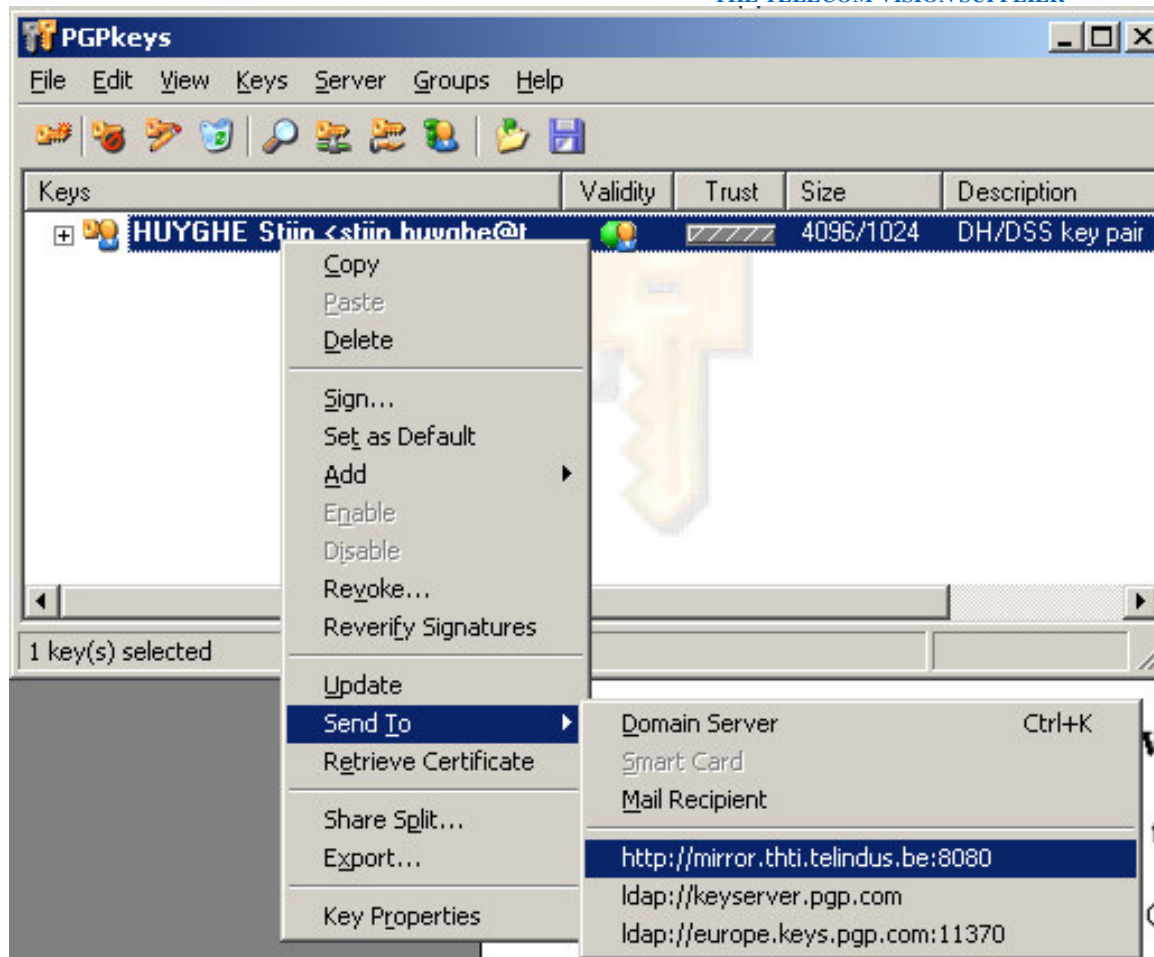
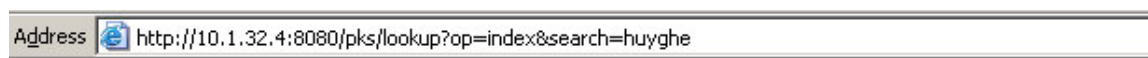


Figure 6 Submitting keys to the server.

### Searching for keys.

Again, two options:

- Online, through the web interface.
- Through the PGP client GUI (Choose "PGPKeys" → "Server" → "Search").



## Public Key Server -- Index ``huyghe ''

Type	bits	/keyID	Date	User ID
pub	1024D/	<a href="#">0027B56E</a>	2003/09/03	HUYGHE Stijn < <a href="mailto:stijn.huyghe@thti.telindus.be">stijn.huyghe@thti.telindus.be</a> >

Figure 7 Searching keys through the web interface.



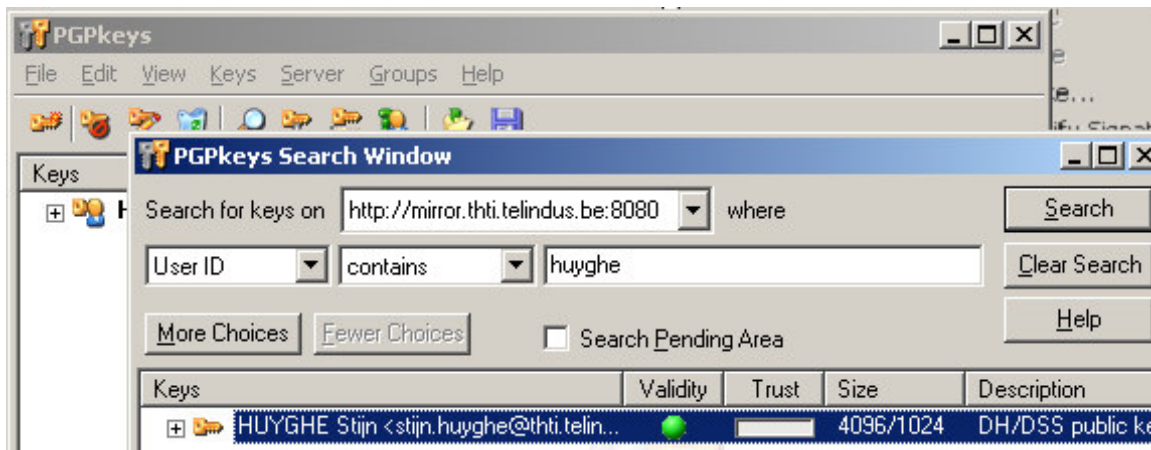


Figure 8 Searching keys through the PGP client GUI.

## Problems?

If you have problems, always consult your log messages because they contain often valuable information! You can find them in "/var/log/messages".

Example of PGP server actions:

```
Sep 11 10:08:43 anubis pksd[1631]: pksd: listener [www]: new www connection from 10.1.32.35
Sep 11 10:08:43 anubis pksd[1631]: pksd: reader [www]: request received: GET /
Sep 11 10:22:15 anubis pksd[1631]: pksd: listener [www]: new www connection from 10.1.32.35
Sep 11 10:22:15 anubis pksd[1631]: pksd: reader [www]: request received: GET
/pks/lookup?op=index&search=huyghe
Sep 11 10:22:15 anubis pksd[1631]: pksd: kd_index: userid="huyghe", flags=0
Sep 11 10:22:15 anubis pksd[1631]: pksd: kd_index: completed successfully
Sep 11 10:23:50 anubis pksd[1631]: pksd: listener [www]: new www connection from 10.1.32.35
Sep 11 10:23:50 anubis pksd[1631]: pksd: reader [www]: request received: GET
/pks/lookup?op=get&exact=off&search=huyghe
Sep 11 10:23:50 anubis pksd[1631]: pksd: kd_get: userid="huyghe", flags=0
Sep 11 10:23:50 anubis pksd[1631]: pksd: kd_get: completed successfully
```

## References:

- PGP Corporation, <http://www.pgp.com>.
- The International PGP Home Page, <http://www.pgpi.org>.
- OpenPGP Public Key Server, <http://sourceforge.net/projects/pks> or <http://pks.sourceforge.net>.
- OpenPGP Message Format, <http://www.ietf.org/rfc/rfc2440.txt>.