

OpenPlatformTrustServices Server Setup Guide

2008/02/18

for OpenPlatformTrustServices v1.0

Copyright IBM Japan, Ltd. 2008

*) This work is sponsored by the Ministry of Economy, Trade and Industry, Japan (METI) under contract for the New-Generation Information Security R&D Program.

*) Linux is a trademark of Linus Torvalds. All trademarks, logos, service marks, and other materials used in this site are the property of IBM corp. or other entities.

1. Introduction.....	4
1.1 Overview	4
1.2 Composition	4
2. Required Packages.....	5
2.1 Preparing the Operating System	5
2.2 Java 6	5
2.3 PostgreSQL.....	5
2.4 Tomcat	5
3. Setup Database.....	6
3.1 Configuration.....	6
3.1.1 (OPTION) enable remote access	6
3.1.2 Start PostgreSQL service	6
3.1.3 Environment variable PGDATA.....	6
3.2 Account Creation	6
3.2.1 Create an administrator account	6
3.2.2 Create a user account	6
3.3 Create database	7
3.4 Inserting data	7
3.4.1 Creating the table schema.....	7
3.4.2 Setup Integrity Information Database of current host	8
3.4.3 Setup Vulnerability Database	9
3.5 Maintenance	9
3.5.1 Performance.....	9
3.5.2 Backup	10
3.5.3 Restore	10
4. Setup the Validation Server.....	10
4.1 Configuration.....	10
4.2 Start the server.....	10
5. Interface to the Database (option).....	10
5.1 GUI.....	10
5.2 Web interface by Ruby on Rails.....	11
5.2.1 Build and install Ruby.....	11
5.2.2 Install RubyGems.....	11
5.2.3 Install Rails and PostgreSQL library	11
5.2.4 Setup Ruby on Rails.....	11

5.2.5	Start Web server.....	13
5.2.6	Accessing to Web server.....	13

1. Introduction

1.1 Overview

This is a demonstration using KNOPPIX - CD bootable Operating System - for experiencing the Remote Attestation which is the fundamental capability provided by Trusted Computing Technology. This KNOPPIX supports Trusted Boot and client software for Remote Attestation, and can be validated by demo Validation Service on Internet. When the validation results in success without any known vulnerability, the client will be able to use a service like a demonstration of vulnerability search service.

This guide is for setting up the server which used in this experiment.

Any information you can share with us, including your test result and trouble, will be very helpful and appreciated. The following mailing-lists are available for such reporting.

Japanese <http://lists.sourceforge.jp/mailman/listinfo/openpts-jpusers>

English <http://lists.sourceforge.jp/mailman/listinfo/openpts-users>

1.2 Composition

This guidance shows about how to setup the server on the “Red Hat Enterprise Linux 4”.

Server OS	Red Hat Enterprise Linux 4
Database Server	PostgreSQL
HTTP Server	Apache
Application Server	Tomcat
Validation Program	Java 6
Client for verification	KNOPPIX

Table 1: Software Composition

Section 3 presents the construction of a database server. Section 4 describes about the application server. Section 5 describes about the interface to access the database.

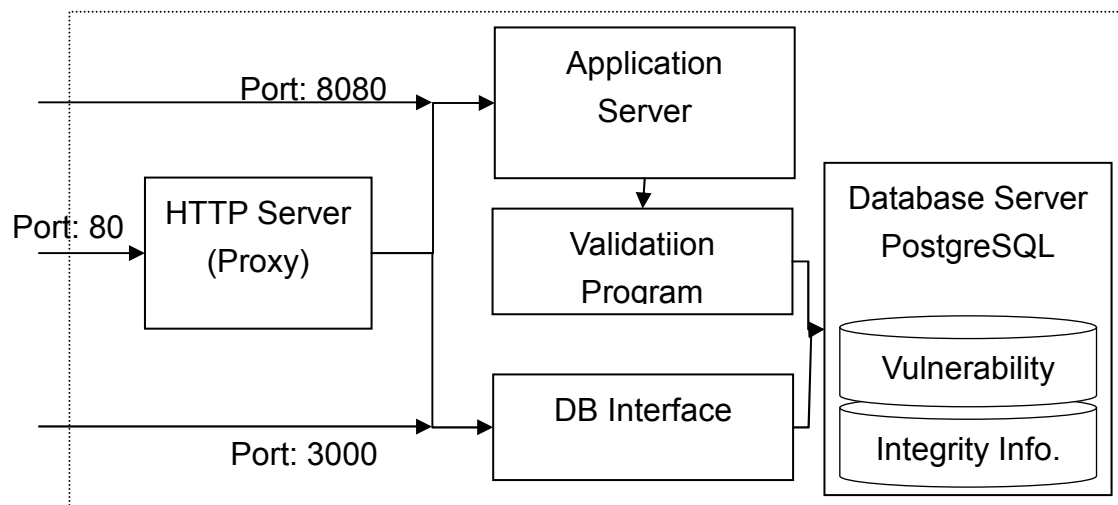


Figure 1: Server Architecture

2. Required Packages

2.1 Preparing the Operating System

Install a Red Hat Enterprise Linux 4.

After installation, you have to disable prelink function by modifying /etc/sysconfig/prelink file.

```
PRELINKING=no
```

In order to confirm a setup immediately, execute the following command.

```
$ prelink -ua
```

2.2 Java 6

Download a RPM package of Java Runtime Environment Version 6, and install it.

<http://www.java.com>

2.3 PostgreSQL

Install PostgreSQL Server RPM package.

```
$ rpm -q postgresql-server
```

Start postgresql server, and set the password for postgres user.

```
# /sbin/service postgresql start
```

```
# passwd postgres
```

2.4 Tomcat

Download a Tomcat 5.5, and install it.

<http://tomcat.apache.org/>

3. Setup Database

3.1 Configuration

3.1.1 (OPTION) enable remote access

Edit `/var/lib/pgsql/data/postgresql.conf`

```
tcpip_socket = true
```

Also edit `/var/lib/pgsql/data/pg_hba.conf`

```
host    all    all    127.0.0.1    255.255.255.255    password
local   all    all                                password
```

3.1.2 Start PostgreSQL service

Start the postgresql server.

```
# /sbin/service postgresql start
```

3.1.3 Environment variable PGDATA

Set the environment variable for the administrator of the database.

```
> su postgres
```

```
Password: xxxxxxxx
```

```
> export PGDATA=/var/lib/pgsql/data
```

3.2 Account Creation

3.2.1 Create an administrator account

To create an administrator account, login to the database by administrator privilege and enter a new password for an administrator.

```
> createuser -a -d -P ptsadmin
```

```
Enter password for new user: xxxxxxxx
```

```
Enter it again: xxxxxxxx
```

```
CREATE USER
```

3.2.2 Create a user account

To create a user account, login to the database by administrator privilege and enter a new password for a new user.

```
> createuser -A -D -P ptsuser
```

```
Enter password for new user: xxxxxxxx
```

Enter it again: xxxxxxxx

CREATE USER

3.3 Create database

Create two databases. One is an integrity information database for knoppix named "iidx_knoppix", and the other is a vulnerability database named "vul".

```
> createdb -E utf8 iidx_knoppix
```

CREATE DATABASE

```
> createdb -E utf8 vul
```

CREATE DATABASE

3.4 Inserting data

Install following two Open Platform Trust Services.

openpts (OpenPlatformTrustServices-0.1.1.tgz)

openpts-tools (OpenPlatformTrustServices-tools-0.1.1.tgz)

3.4.1 Creating the table schema

Run the script, /opt/OpenPlatformTrustServices/database/dbsetup.sh of openpts-tools.

Confirm the configuration of the database and modify them if needed. To create the database, select S) Setup New Databases.

```
$ sh /opt/OpenPlatformTrustServices/database/dbsetup.sh
```

S) Setup New Databases

C) Show Current Configuration

L) Show State

B) Backup Databases

D) Delete Databases

Q) Exit

When you use the same variables as examples 3.2 and 3.3, the setting becomes to the following values.

DB type	:postgres
DB admin	:ptsadmin
DB user	:ptsuser
Vulnerability Database name	:vul

Integrity Information Database 0 name :iidb_knoppix

3.4.2 Setup Integrity Information Database of current host

At first, run the KNOPPIX on the client platform to correct package information.

To get package information from current host, execute the script

/opt/OpenPlatformTrustServices/bin/deb-all.sh of openpts-tools. The argument is a directory name. In the following example, “*knoppix*” is the directory name to store the corrected information. This shell script runs “deb-meta.pl”, “deb-file.pl sha1” and “deb-file.pl md5”.

```
$ sh /opt/OpenPlatformTrustServices/bin/deb-all.sh knoppix
```

At the host using rpm packages, just run tools/package/rpm/rpm-all.sh in a similar manner as the debian host.

After running this command, we can get the data files in the directory.

The files are

```
packagelist.txt
tm3-data.txt
data/
    NAME_VERSION.metadata
    NAME_VERSION.md5.filelist
    NAME_VERSION.sha1.filelist
```

In order to import in the database, transport these data to the server.

To use the openpts command at /opt/OpenPlatformTrustServices/bin/openpts, setup the database configuration.

Copy /opt/OpenPlatformTrustServices/database/ibatis/sqlMapsConfig.properties.sample to sqlMapsConfig.properties and edit it according to your environment. When you use the same variables as examples 3.2 and 3.3, the following values are used in setting.

```
driver=org.postgresql.Driver
username=ptsadmin
password=xxxxxxx
url_vul=jdbc:postgresql://localhost/vul
url_iidb0=jdbc:postgresql://localhost/iidb_knoppix
```

To insert the data into Integrity Information Database, run the following command. The last argument is the data directory which storing the package information. The “—dbindex” is the

database index listed as `url_iidb` in `sqlMapsConfig.properties`. If you want to use the database of “`url_iidb0`”, add “`--dbindex 0`”.

```
$ /opt/OpenPlatformTrustServices/bin/openpts debimport --dbindex 0 --inputdir
~/knoppix/data/
```

3.4.3 Setup Vulnerability Database

Get the vulnerability information via Internet. We need CVE data and DSA (Debian Security Advisory) data to check the security of KNOPPIX.

CVE is released from 2002 to 2008 (from `nvdCVE-2002.xml` to `nvdCVE-2008.xml`). To setup `cve_definitions` table, execute the following command for each year. In this example, the xml files are saved at “`--outputdir /tmp`”.

```
$ /opt/OpenPlatformTrustServices/bin/openpts cve --xmlfile
http://nvd.nist.gov/download/nvdCVE-2008.xml --outputdir /tmp
```

To store the DSA data to `debian_security_advisories` table, execute the following command for each year from 2000 to 2008.

```
$ /opt/OpenPlatformTrustServices/bin/openpts dsainfo --url
http://www.debian.org/security/2008/ --outdir /tmp
```

Then, get the detail information for each DSA entry, and make it reflected to the database of package information.

```
$ /opt/OpenPlatformTrustServices/bin/openpts dsadetail --outdir /tmp
$ /opt/OpenPlatformTrustServices/bin/openpts dsasync --dbindex 0
```

If you use the RPM package of Red Hat, get OVAL information instead of DSA. In this case, the argument “`--distribution`” is for the version number of Red Hat.

```
$ /opt/OpenPlatformTrustServices/bin/openpts oval --dbindex 0 --xmlfile
https://www.redhat.com/security/data/oval/com.redhat.rhsa-all.xml --distribution rhel5
```

3.5 Maintenance

3.5.1 Performance

In PostgreSQL, you can recover an unnecessary domain by performing `VACUUM`.

- `AUTOVACUUM`

- VACUUM DELAY
- VACUUM FULL

REINDEX command re-create the INDEX which was created beforehand.

- REINDEX

3.5.2 Backup

You can write out the database to a file as SQL.

```
$ pg_dump database_name > file_name.sql
```

3.5.3 Restore

To restore the backup files to database,

```
$ psql -e database_name < file_name.sql  
$ pg_restore -d database_name file_name.sql
```

4. Setup the Validation Server

Install the following two Open Platform Trust Services.

```
openpts (OpenPlatformTrustServices-0.1.1.tgz)  
openpts-tcdemo (OpenPlatformTrustServices-tcdemo-0.1.1.tgz)
```

4.1 Configuration

After installing the openpts-tcdemo, jar files are located to webapps/pva/WEB-INF/lib/ of Tomcat directory.

4.2 Start the server

```
$ /etc/init.d/tomcat start
```

5. Interface to the Database (option)

5.1 GUI

These tools are the viewer for the PostgreSQL database.

pgAdmin III - <http://www.pgadmin.org/>

phpPgAdmin - <http://phppgadmin.sourceforge.net/>

5.2 Web interface by Ruby on Rails

(* This generating program is not published yet.)

5.2.1 Build and install Ruby

For Ruby on Rails, the version of Ruby has to be upper than 1.8.5. Download Ruby's source package from <ftp://ftp.ruby-lang.org/pub/ruby/1.8/>, and compile it. Latest version is preferable. We have installed ruby-1.8.6-p111.

```
> cd ~/ruby-1.8.6-p111
> ./configure
> make
> su
# make install
```

5.2.2 Install RubyGems

Next, download RubyGems RPM package from <http://rubyforge.org/projects/rubygems/> and install it. We have installed RubyGems-0.9.4.

```
# cd ~/rubygems-0.9.4
# ruby setup.rb
```

5.2.3 Install Rails and PostgreSQL library

Install the Ruby on Rails and related packages, and database library for PostgreSQL.

```
# gem install rails --remote --include-dependencies
# gem install postgres-pr
```

5.2.4 Setup Ruby on Rails

Downloading openpts-rail (* not ready now), and install it.

In the following example, we make "openpts-server" project using examples used before.

Run a setup_project.sh script to make the Ruby on Rails project.

The value of "URL Path - PATHNAME for <http://hostname/PATHNAME/>" is needed to run many projects. For example, if you enter the name "knoppix" for PATHNAME, the server will be located at <http://localhost:3000/knoppix/>. This PATHNAME is not needed in running a single server. The address for the default server is <http://localhost:3000/>.

"Web login" items are variables for access control to this web interface. These can be changed later. To change them, edit .htpasswd (for Web login name for administrator) and .htpasswd_user (for Web login name for guest) in the project directory. If you don't use

access control, edit the app/controllers/{package_controller.rb, package_user_controller.rb, measurements_controller.rb, measurement_user_controller.rb} and comment out the line of "htpasswd :file=>..." by "#".

After confirming configurations and selecting E, the project will be created. When you delete a project, please erase the created whole directory.

```
$ sh setup_project.sh
```

S) Setup/Show Configuration

E) Execute

Q) Quit

H) Help

```
select [S/E/Q/H]:S
```

Interactive Setup

(To erase the optional selection, please type '-'.)

Project name []: openpts-server

Database SQL type, postgres or mysql []: postgres

Integrity Information Database name []: iidb_knoppix

Vulnerability Database name []: vul

Database User name []: ptsuser

Database User password []: xxxxxxxx

Database Administrator name []: ptsadmin

Database Administrator password []: xxxxxxxx

URL Path - PATHNAME for http://hostname/PATHNAME/ (optional)[]: -

Web login name for guest []: guest

Web login password for guest []: xxxxxxxx

Web login name for administrator []: admin

Web login password for administrator []: xxxxxxxx

save config ...

Current Configurations

Project name: openpts-server

Database>

Sql type: postgres

Integrity Information Database name: iidb_knoppix

```
Vulnerability Database name:      vul
User name:                        ptsuser
User password:                    xxxxxxxx
Administrator name:               ptsadmin
Administrator password:          xxxxxxxx
```

Web>

```
URL Path (optional):
Login name for guest:             guest
Login password for guest:        xxxxxxxx
Login name for administrator:    admin
Login password for administrator: xxxxxxxx
```

Change? [N/y]:N

S) Setup/Show Configuration

E) Execute

Q) Quit

H) Help

select [S/E/Q/H]:E

5.2.5 Start Web server

After a setup is completed, start the Web server as following.

```
$ cd openpts-server
```

```
$ ruby script/server
```

The “ruby script/server” has some options. The default port number is 3000. To use another number, add `-p` option like “`-p 3001`”. When run the server as a daemon, add the `-d` option. To see the help, add `-h` option.

5.2.6 Accessing to Web server

We can access `http://localhost:3000/{package_user, measurement_user}` by the guest account, and `http://localhost:3000/{packages, measurements}` by the administrator account.