OpenPlatformTrustServices サーバー構築ガイド

2008/02/21

for OpenPlatformTrustServices v1.0

*) 本研究は、経済産業省 新世代情報セキュリティ研究開発事業の委託研究の一環として行っているものです。

1. はじめに

1.1 概要

CD 起動型 OS、KNOPPIX を利用した Trusted Computing の基本技術の 1 つである Remote Attestation を体験するデモです。実験で利用する KNOPPIX には Trusted Boot の機能と Remote Attestation のクライアント側の機能が組み込まれており、Remote Attestation 検証サーバーによる検証が可能です。検証が成功した場合（脆弱性や不正な改ざんがない場合）には、デモのサービス（脆弱性の検索サービス）を利用することが可能です。
本書は、実験で利用する Remote Attestation 検証サーバーの構築ガイドです。

お気づきになった情報は下記メーリングリストなどに寄せていただけると助かります。
日本語　http://lists.sourceforge.jp/mailman/listinfo/openpts-jpusers
英語　　http://lists.sourceforge.jp/mailman/listinfo/openpts-users

1.2 想定する構成

本ガイドは Red Hat Enterprise Linux 4 を利用してサーバーを構築する場合について記載してあります。

サーバーOS	Red Hat Enterprise Linux 4
データベースサーバー	PostgreSQL
HTTP サーバー	Apache
アプリケーションサーバー	Tomcat
検証プログラム動作環境	Java 6
検証対象クライアント	KNOPPIX

表 1: 想定する構成

データベースサーバーの構築については 3 章、アプリケーションサーバーについては 4 章、データベースへアクセスするインタフェースについては 5 章で説明します。

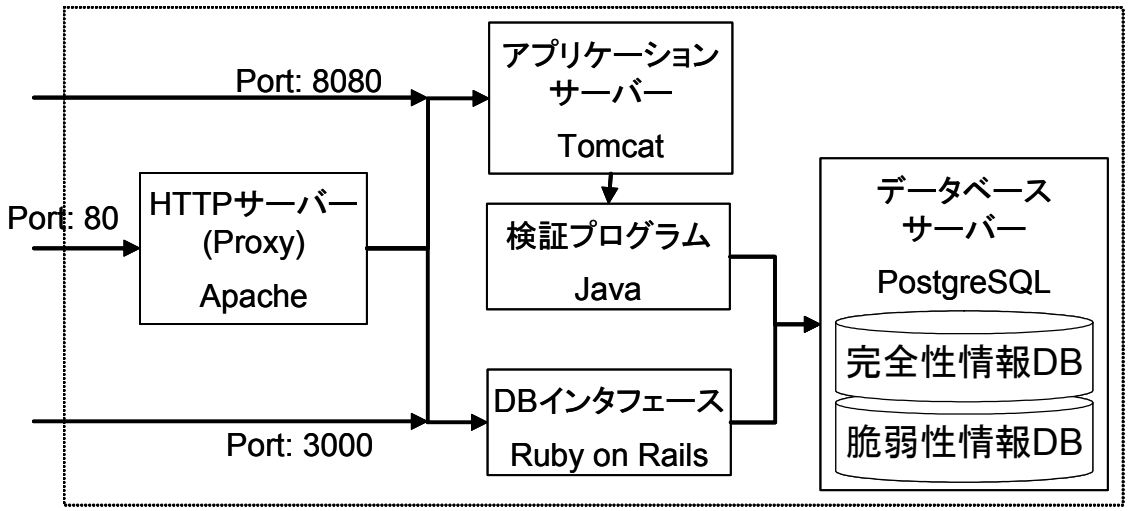

図 1： サーバー構成

2. 必要なパッケージ

2.1 OS 準備

Red Hat Enterprise Linux 4 をインストールします。

インストール後、Prelink 機能を OFF にするため、/etc/sysconfig/prelink ファイルを次のように編集します。

```
PRELINKING=no
```

設定をすぐに有効にするため、以下のコマンドを実行します。

```
$ prelink -ua
```

2.2 Java 6

Java Runtime Environment Version 6 の RPM パッケージを入手し、インストールします。

http://www.java.com

2.3 PostgreSQL

postgresql-server をインストールします。

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

postgresql サーバーを起動し、postgres ユーザーのパスワードを設定します。

```
# /sbin/service postgresql start
# passwd postgres
```

2.4 Tomcat

Tomcat 5.5 を入手し、インストールします。
http://tomcat.apache.org/

本ガイドでは、RPM のパッケージではなく、単独インストールされている Tomcat について解説します。 4 章では、/opt/apache-tomcat-5.5.25 にインストールされているとして、解説します。

2.5 OpenPlatformTrustServices

OpenPlatformTrustServicesv0.1.1 を入手し、インストールします。
次の3つのパッケージのインストールが必要です。

- openpts (OpenPlatformTrustServices-0.1.1.tgz)
  - PTS 本体のパッケージ
- openpts-tools（OpenPlatformTrustServices-tools-0.1.1.tgz）
  - PTS 関連の各種ツールのパッケージ
- openpts-tcdemo（OpenPlatformTrustServices-tcdemo-0.1.1.tgz）
  - 構成証明 TTP 実証実験で利用しているデモパッケージ

ビルドとインストールの方法については以下を参照してください。
http://sourceforge.jp/projects/openpts/wiki/HowToBuildForRedHat

3. データベース構築

3.1 サーバーの設定

3.1.1 (OPTION) リモートアクセス有効化

/var/lib/pgsql/data/postgresql.conf を次のように編集します。

```
tcpip_socket = true
```

/var/lib/pgsql/data/pg_hba.conf を次のように編集します。

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

3.1.2 PostgreSQL サービスの起動

postgresql サーバーを起動します。

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

3.1.3 PGDATA 変数の設定

管理者権限でデータベースにログインし、環境変数を設定します。

```
> su postgres
Password: xxxxxxxx
> export PGDATA=/var/lib/pgsql/data
```

3.2 アカウント作成

3.2.1 管理者アカウント作成

管理者権限でデータベースにログインした状態で、新たな管理者名とパスワードを設定します。

```
> createuser -a -d -P ptsadmin
Enter password for new user: xxxxxxxx
Enter it again: xxxxxxxx
CREATE USER
```

3.2.2 ユーザーアカウント作成

管理者権限でデータベースにログインした状態で、ユーザー名とパスワードを設定します。

```
> createuser -A -D -P ptsuser
Enter password for new user: xxxxxxxx
Enter it again: xxxxxxxx
CREATE USER
```

3.3 データベース作成

完全性情報データベースと脆弱性情報データベースを作成します。ここでは、完全性情報データベースを knoppix 向けに”iidb_knoppix”という名前に、脆弱性情報データベースを”vul”という名前に設定します。

```
> createdb -E utf8 iidb_knoppix
CREATE DATABASE
> createdb -E utf8 vul
CREATE DATABASE
```

3.4 データ挿入

Open Platform Trust Services をインストールします。次の 2 つが必要です。

openpts (OpenPlatformTrustServices-0.1.1.tgz)

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

3.4.1 テーブルスキーマ作成

openpts-tools の /opt/OpenPlatformTrustServices/database/dbsetup.sh を起動します。

データベースの設定を確認し、必要があれば変更し、データベースを作成します。S) Setup New Databases で設定の変更と作成を行うことができます。

```
$ 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
```

3.23.3 の例であれば、以下のようになります。

DB type	:postgres
DB admin	:ptsadmin
DB user	:ptsuser
Vulnerability Database name	:vul
Integrity Information Database 0 name	:iidb_knoppix

3.4.2 完全性情報の挿入

データ取得対象クライアントである KNOPPIX を起動し、KNOPPIX 上で openpts-tools の /opt/OpenPlatformTrustServices/bin/deb-all.sh を実行します。次の例で ”*knoppix*” の部分は

ディレクトリ名です。このスクリプトは ”deb-meta.pl”, “deb-file.pl sha1”, “deb-file.pl md5” の 3 種類のスクリプトを実行します。

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

引数として指定したディレクトリ内に data ディレクトリが作成され、Debian パッケージの md5, sha1 のリストファイルと metadata のファイルが含まれています。ここで取得したデータはサーバーで使用するため、サーバー上にコピーします。

次にサーバー上で、openpts の /opt/OpenPlatformTrustServices/bin/openpts を利用するため、データベースを設定します。以下はサーバー上での操作です。
/opt/OpenPlatformTrustServices/database/ibatis/sqlMapsConfig.properties.sample を sqlMapsConfig.properties という名前でコピーし、編集します。3.23.3 の例であれば、次のようになります。

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

openpts を実行し、先ほど取得したデータをデータベースに挿入します。最後の引数が Debian パッケージのデータのあるディレクトリ（data）を指します。

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

Debian パッケージではなく RPM パッケージの完全性情報を取得するためには、”deb” の部分を ”rpm” となっているファイルやコマンドを使用します。上記のコマンドの場合は “debimport”の代わりに “rpmimport”を用います。

3.4.3 脆弱性情報の挿入

脆弱性情報はインターネット経由で取得します。CVE 情報、DSA（Debian Security Advisory）の情報が必要です。CVE 情報源は nvdcve-2008.xml から nvdcve-2002.xml まであります。次のコマンドの引数を変えてそれぞれ実行してください。--outputdir は取得した xml ファイルを保存する場所です。

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

KNOPPIX では DSA 情報を用います。同様に 2008 の部分を 2000 まで引数を変えてそれぞれ

実行してください。

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

次に、各 DSA 情報に対する詳細情報を取得し、さらに Debian パッケージ情報のデータベースへ反映させます。--dbindex の数字は、sqlMapsConfig.properties で指定した url_iidb の番号を示します。url_iidb0 の場合は 0 です。

$ /opt/OpenPlatformTrustServices/bin/openpts dsadetail --outdir /tmp

$ /opt/OpenPlatformTrustServices/bin/openpts dsasync --dbindex 0

Debian パッケージではなく RPM パッケージを対象としている場合は、DSA 情報の代わりに OVAL 情報を用います。対象とする Red Hat のバージョンを –distribution 引数として指定します。

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

3.5 保守

3.5.1 メンテナンス処理

PostgreSQL では、VACUUM という処理を行うことで不要領域を回収することができます。

- AUTOVACUUM（自動化）
- VACUUM DELAY（負荷分散）
- VACUUM FULL

インデックス化されている項目については、REINDEX コマンドを用いてインデックスを再作成することができます。

3.5.2 バックアップ

データベースを SQL 文としてファイルに書き出します。

$ pg_dump データベース名 > バックアップファイル名

3.5.3 復元（リストア）

バックアップしたファイルをデータベースに戻します。以下どちらでも可能です。

$ psql -e データベース名 < バックアップファイル名

$ pg_restore –d データベース名 バックアップファイル名

4. 検証サーバーの構築

Open Platform Trust Services をインストールします。次の 2 つが必要です。

openpts (OpenPlatformTrustServices-0.1.1.tgz)

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

OpenPlatformTrustServices-tcdemo-0.1.1.tgz を展開してサーブレットをインストールします。
必要であれば Makefile にある CATALINA_HOME の設定を修正します。
（/opt/apache-tomcat-5.5.25 になっています）

Makefile にバグがありますので以下の修正をお願いします。
Makefile にある LIB_DIR を/opt/`OpenPlatformTrustServices/lib` に変更します。
Makefile の install-servlet で“`../openpts/database/ibatis`“ の記述を
“`/opt/OpenPlatformTrustServices/database/ibatis`“ に変更します。

以下のコマンドを実行し、検証サーバーのサーブレットを Tomcat に組み込みます。

```
$ tar xvfz OpenPlatformTrustServices-tcdemo-0.1.1.tgz
$ cd OpenPlatformTrustServices-tcdemo-0.1.1
$ sudo make setup-jars
$ make servlet
$ sudo make install-servlet
```

4.1 サーバー設定

下記設定ファイルを修正＆作成します。

/opt/apache-tomcat-5.5.25/webapp/pva/WEB-INF/classes/sqlMapsConfig.properties

```
driver=org.postgresql.Driver
url_vul=jdbc:postgresql://localhost/vul
url_iidb0=jdbc:postgresql://localhost/iidb_redhat
url_iidb1=jdbc:postgresql://localhost/iidb_centos
url_iidb2=jdbc:postgresql://localhost/iidb_knoppix
url_iidb3=jdbc:postgresql://localhost/iidb_ubuntu
url_iidb4=jdbc:postgresql://localhost/iidb
url_iidb5=jdbc:postgresql://localhost/iidb
url_iidb6=jdbc:postgresql://localhost/iidb
url_iidb7=jdbc:postgresql://localhost/iidb
username=ptsadmin
password=XXXXXXXX
```

注意）KNOPPIX の検証デモを行う場合は、`url_iidb2` に `iidb_knoppix` を配置してください。

4.2 サーバー起動

```
$ sudo make start-tomcat
```

ブラウザから http://localhost:8080/pva/ にアクセスし、Platform Validation Authority Demo の画面が表示されれば OK です。

4.3 サーバー起動（手動、デバッグ用）

```
$ cd /opt/apache-tomcat-5.5.25/bin
$ ./catarina.sh run
```

コンソールにログが表示されます。

5. データベースへのインタフェース (option)

5.1 GUI

以下のようなツールを用いることで、GUI 経由で PostgreSQL のデータベースを閲覧することができます。

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

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

5.1.1 phpPgAdmin のインストール

5.2 で説明する Rails が使えない場合に、phpPgAdminを使ってデモサーバーを構築することも可能です。

```
# yum install php-pgsql
# cd /opt
# tar xvfj phpPgAdmin-4.1.3.tar.bz2
# ln -s /opt/phpPgAdmin-4.1.3 /var/www/html/phpPgAdmin
# chown -R apache.apache /var/www/html/phpPgAdmin
```

TODO,認証周りでトラぶってうまく動きません。

5.2 Ruby on Rails による Web インタフェース構築

（※ 現在、生成プログラムは未公開です。）

5.2.1 Ruby のインストール

Ruby のバージョンは 1.8.5 以上である必要があります。最新版を入手し、コンパイルおよびインストールしてください。ftp://ftp.ruby-lang.org/pub/ruby/1.8/

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

5.2.2 RubyGems のインストール

RubyGems の RPM パッケージを入手し、インストールしてください。
http://rubyforge.org/projects/rubygems/

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

5.2.3 Ruby on Rails インストール

Ruby on Rails とそれに依存するファイル、そして PostgreSQL を利用するために必要なライブラリをインストールします。

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

5.2.4 Ruby on Rails インタフェースの設定

openpts-rails（仮称 ※現在、生成プログラムは未公開です。）を入手し、インストールします。
以下は、これまでの例を用い、openpts-server という名前のプロジェクトを作成する場合の入力例です。
”URL Path - PATHNAME for http://hostname/PATHNAME/” は複数の Ruby on Rails サーバーを起動するとき、パスを分けるために必要です。例えば knoppix という名前を指定すると、サーバーにアクセスするために http://localhost:3000/knoppix/ というURL を利用することになります。単一サーバーの場合は必要ありません。指定しない場合は、http://localhost:3000/ になります。
Web login の項目はインタフェースへのアクセス制御を行うための項目です。これらは後に変更することができます。変更するには、作成されたプロジェクトディレクトリ内の .htpasswd（Web login name for administrator）, .htpasswd_user（Web login name for guest） を編集します。アクセス制御を使用しない場合は、同ディレクトリ内の app/controllers/{package_controller.rb, package_user_controller.rb, measurements_controller.rb, measurement_user_controller.rb} から htpasswd :file=> の行を # でコメントアウトしてください。

設定内容の確認後、select [S/E/Q/H]:E を指定すると、プロジェクトが作成されます。プロジェクトを削除する場合は、作成されたディレクトリごと消してください。

```
$ 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 Administator 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 Web サーバーの起動

作成したプロジェクトで Ruby on Rails サーバーを起動するためには、以下のコマンドを実行してください。

```
$ cd openpts-server
$ ruby script/server
```

デフォルトのポート番号は 3000 です。異なる番号を使用する場合は、-p 3001 のように –p オプションをつけてください。サーバーをデーモンとして起動するためには –d オプションをつけてください。オプションのヘルプを表示するためには -h オプションをつけてください。

5.2.6 Web サーバーへのアクセス

Guest アカウントから、http://localhost:3000/{package_user, measurement_user}、Administrator アカウントから、http://localhost:3000/{packages, measurements}にアクセスできます。

6. HTTP サーバー（Apache）との併用(option)

フロントに Apache を配してプロキシをして構成します。その結果 Tomcat、Rails は Apache の背後で動作するようになります。HTTPS の設定などはフロントの Apache で行います。

6.1 HTTP サーバーのプロキシ設定

/etc/httpd/conf を以下のように編集します

```
<snip>
ProxyRequests Off
ProxyPass /pva/ http://localhost:8080/pva/
ProxyPassReverse /pva/ http://localhost:8080/pva/
ProxyPass /knoppix/ http://localhost:3000/knoppix/
ProxyPassReverse /knoppix/ http://localhost:3000/knoppix/
ProxyPass /redhat/ http://localhost:3000/redhat/
ProxyPassReverse /redhat/ http://localhost:3000/redhat/
ProxyPass /centos/ http://localhost:3000/centos/
ProxyPassReverse /centos/ / http://localhost:3000/centos/
ProxyPass /ubuntu/ http://localhost:3000/ubuntu/
ProxyPassReverse /ubuntu/ / http://localhost:3000/ubuntu/
<snip>
```

6.2 HTTPS の設定

TBD

6.3 HTTPS の再起動

```
# /sbin/service httpd restart
```

ブラウザから http://localhost/pva/ にアクセスし、Platform Validation Authority Demo の画面が表示されれば OK です。

7. Geeks を使ったサーバーの動作検証

KNOPPIX5.1.1 for Trusted Computing Geeks を使ったサーバーの動作検証の方法を示します。
今回作成したサーバーを利用するためには、デフォルトの設定を修正する必要があります。

以下は、すでに一度 Geeks をセットアップした環境に対する変更作業です。

7.1 サーバーの起動

第 2 章から 6 章の設定が完了しているサーバーを再起動した場合、

```
$ cd /opt/apache-tomcat-5.5.25/bin/
$ sudo ./catarina.sh start

$ cd /opt/rails
$ sudo ./startup_server.sh

$ ifconfig
```

サーバーの IP アドレスを確認しておきます（この例では 192.168.0.2）。

7.2 HTTP でサーバーに接続する場合のクライアントの設定変更

まず、/opt/OpenPlatformTrustServices/tcdemo.properties ファイルの修正を行います。サーバーのアドレスを設定してください（この例では 192.168.0.2）。

```
<snip>
pts.hostname=192.168.0.2
pts.port=80
pts.protocol=servlet
pts.ssl=no
<snip>
service.num=2
service.0.name=Platform Validation Authority - Listing packages
service.0.url=http://192.168.0.2/knoppix/package_user
service.0.authtype=BASIC AUTH
service.0.account=guest
service.0.password=given
service.0.extend=/usr/lib/iceweasel/firefox-bin
service.1.name=Platform Validation Authority - Listing measurement
service.1.url=http://192.168.0.2/knoppix/measurement_user
service.1.authtype=BASIC AUTH
service.1.account=guest
service.1.password=given
service.1.extend=/usr/lib/iceweasel/firefox-bin
<snip>
```

次にクライアントの構成を修正します。

```
$ cd /opt/OpenPlatformTrustServices/tcdemo
$ sudo make start-client-admin-gcj
```

で Identity Setup タブの“Setup local settings for User”をクリックして、上記の変更を反映してください。その後、TCDEMO UserTool を起動し検証を行います。同時に表示されるコンソールにサーバーのアドレスが表示されていますので、確認してください。