ATEXO
Trust Business Integration

ADULLACT

Local Trust PKI

GUIDE D'INSTALLATION

IDENTITE DU DOCUMENT	
Client	ADULLACT
Affaire	Local Trust PKI
Titre	GUIDE D'INSTALLATION
Référence	ATEXO – ADULLACT – DT002 – 20031112
Etat	Final
Version	1.0
Du	16 décembre 2003
Dernière page	13

EVOLUTION DU DOCUMENT			
Date	**Version**	**Rédacteur**	**Commentaires**
16-12-2003	1.0	G.NAGGEAR	Création du document

APPROBATION DE LA VERSION		
Entreprise ou Service	**Nom**	**Visa**
ATEXO	P.FAU	

DIFFUSION DE LA VERSION				
Entreprise ou Service	**Destinataires**	**Fonction**	**Pour action**	**Pour info**
ADULLACT	P.FEYDEL		**x**	

1. INSTALLATION

1.1 Packages à installer

1.1.1 Serveur Web

- httpd-1.3.28 (serveur Apache 1.3.28)
- mod_ssl-2.8.15-1.3.28 (module *HTTP over SSL*)

1.1.2 Outil cryptographique

- Openssl 0.9.7c

1.1.3 Langage php

- php-4.3.3 (serveur php)

1.1.4 MySQL

- MySql-4.0.15-0(serveur MySQL)
- MySql-Max-4.0.15-0
- MySql-devel-4.0.15

1.1.5 LDAP

- openldap-2.1.23

1.1.6 Serveur de mail

- sendmail-8.12.5-7

1.2 Installation des différents packages

Les différents logiciels décrits dans la section précédente sont à télécharger au format de compression `*.tar.gz` sauf les fichiers concernant MySQL et sendmail qui sont à télécharger au format `*.rpm`.

Le système d'exploitation utilisé est RedHat 8.0

L'ensemble des fichiers doit être copié dans le répertoire `/usr/src/` de Linux.

Ensuite, il faut décompresser chaque fichier à l'aide de la commande suivante :

```
gzip -d -c logiciel.tar.gz | tar xvf -
Exemple pour Apache :
gzip -d -c apache 1.3.28.tar.gz | tar xvf -
```

Une fois les fichiers décompressés, on peut alors passer à la compilation des fichiers source de chaque logiciel.

Note : Pour l'exécution des différentes commandes exécutées ci-après, le répertoire de base est `/usr/src/`.

1.3 OpenSSL

Voici la succession de commande à effectuer pour compiler OpenSSL.

```
cd openssl-0.9.7
./config
make
make test
cd ..
```

La compilation de OpenSSL peut prendre quelques minutes et les détails de compilation doivent s'afficher à l'écran.

1.4 MySQL

L'installation de MySQL Server et de MySQL Max se fait de la manière suivante :

```
rpm -ivh MySQL-4.0.15-0.i386.rpm
rpm -ivh MySQL-Max-4.0.15-0.i386.rpm
rpm -ivh MySQL-server-4.0.15-0.rpm
rpm -ivh MySQL-devel-4.0.15-0.rpm
```

1.5 ModSSL

Il faut configurer modSSL de façon à ce qu'il communique avec Apache. Cela se fait par la commande suivante :

```
cd mod ssl-2.8.12-1.3.28
./configure \
--with-apache=../apache_1.3.28
cd ..
```

1.6 Apache

La compilation des fichiers sources d'Apache se fait de la manière suivante:

```
cd apache_1.3.28
SSL BASE="../openssl-0.9.7c" \
./configure \
--with-layout=Apache \
--prefix=/usr/local/apache \
--enable-module=ssl \
--activate-module=src/modules/php4/libphp4.a \
--enable-module=php4 \
```

1.7 OpenLdap

Il faut télécharger le module OpenLdap de RedHat sous forme de RPM. L'installation de OpenLdap se fait de la manière suivante:

```
tar -zvxf openldap-2.1.23.tar.gz
cd openldap-2.1.23
./configure
make depend
make
make test
make install
cd ..
```

1.8 PHP

Tout d'abord il faut pré-configurer Apache pour PHP à l'aide de la commande suivante :

```
cd apache 1.3.28
./configure \
--prefix=/usr/local/apache
cd ..
```

La configuration (`./configure`) de PHP avec les modules adéquats et la compilation des sources se fait de la manière suivante :

```
cd php-4.3.3
./configure   --with-apache=../apache_1.3.28 --with-mysql=/usr --with-zlib=/usr
--with-ldap=/usr/local --with-gettext --with-mhash=/usr --with-mcrypt=/usr --with-curl=/usr
gmake
gmake install
cd ..
```

1.9 Sendmail

Il faut télécharger le module Sendmail de RedHat sous forme de RPM. L'installation se fait de la manière suivante:

```
rpm -ivh sendmail-8.12.5-7.src.rpm
```

2 CONFIGURATION

Les fichiers servant à l'installation de Local Trust PKI sont les suivants: `pki_web.tgz` et `pki_conf.tgz .`

Soit `path_to_pki` le répertoire racine du serveur web. Ex: `/usr/local/apache/htdocs/IGC`.

Exécuter la commande :

```
tar -zvxf pki_web.tgz -C path_to_pki.
```

Les répertoires créés dans `path_to_pki` lors de l'exécution de cette commande sont: `Admin, AC_aut, AE, Titulaire` et `pki_scripts.`

Soit `root_dir` le répertoire de configuration de la PKI. EX: `/etc/pki/localtrust`.

Exécuter la commande :

```
tar -zvxf pki_conf.tgz -C root_dir.
```

2.1 Apache

Ci-joint les éléments essentiels qui doivent figurer dans httpd.conf (inclus dans <VirtualHost *default*:443>):

```
#IGC AE
# Authentification mutuelle SSL
<Directory /usr/local/apache/htdocs/IGC/AE>
SSLExportClientCertificates
SSLRequireSSL
SSLVerifyClient require
SSLVerifyDepth 2
</Directory>

#IGC AC aut
# Authentification mutuelle SSL
<Directory /usr/local/apache/htdocs/IGC/AC_aut>
SSLExportClientCertificates
SSLRequireSSL
SSLVerifyClient require
SSLVerifyDepth 2

SSLRequire %{SSL_CLIENT_S_DN_OU} in { "AUTORITE ADMINISTRATION", "AUTORITE
ENREGISTREMENT","TITULAIRE",}
</Directory>

#IGC Admin
# Authentification mutuelle SSL
<Directory /usr/local/apache/htdocs/IGC/Admin>
```

```
SSLExportClientCertificates
SSLRequireSSL
SSLVerifyClient require
SSLVerifyDepth 2
</Directory>

#IGC Titulaire/secure
# Authentification SSL du serveur
<Directory /usr/local/apache/htdocs/IGC/Titulaire/secure>
SSLExportClientCertificates
SSLRequireSSL
SSLVerifyClient optional no ca
SSLVerifyDepth 2
</Directory>
```

2.2 MySQL

Pour configurer les tables MySQL, un script d'initialisation est fourni dans path_to_pki/pki_scripts/*mysql_script.sql*

Afin de l'utiliser, taper la commande:

```
cd root_dir/pki_scripts
mysql –u root –p < mysql_script.sql
```

Ci-dessous le contenu de ce script:

```
##Création de la base
CREATE DATABASE ltrustpki;


## Creation de l'utilisateur avec les droits
GRANT USAGE ON * . * TO ltrustpki@localhost IDENTIFIED BY "secret" WITH MAX_QUERIES_PER_HOUR
0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 ;
GRANT LOCK TABLES ON * . * TO ltrustpki@localhost WITH MAX QUERIES PER HOUR 0
MAX CONNECTIONS PER HOUR 0 MAX UPDATES PER HOUR 0 ;
GRANT SELECT , INSERT ,UPDATE ,REFERENCES ,INDEX ,CREATE TEMPORARY TABLES ON ltrustpki . *
TO ltrustpki@localhost;




USE ltrustpki;
#
# Structure de la table `EURequest`
#
```

```
CREATE TABLE `EURequest` (
  `id` int(11) NOT NULL auto increment,
  `EU_id` int(11) NOT NULL default '-1',
  `EA_id` int(11) NOT NULL default '0',
  `CA id` int(11) NOT NULL default '0',
  `demande rev id` int(11) NOT NULL default '-1',
  `rev id` int(11) NOT NULL default '-1',
  `pkcs10` longtext,
  `spkac` mediumtext,
  `date_req` datetime default NULL,
  `date ea` datetime default NULL,
  `date ca` datetime NOT NULL default '0000-00-00 00:00:00',
  `x509` longtext,
  `date_x509` datetime default NULL,
  `md5_x509` varchar(32) default NULL,
  `revoked` enum('0','1','2','3') NOT NULL default '0',
  `status` enum('0','1','2','3','4','5','6') NOT NULL default '0',
  `reject reason` tinytext,
  `answer_1` varchar(50) default NULL,
  `answer_2` varchar(50) default NULL,
  `answer_3` varchar(50) default NULL,
  `answer 4` varchar(50) default NULL,
  `retrieval code` varchar(16) NOT NULL default '',
  `passphrase` varchar(16) default NULL,
  `is_Published` tinyint(4) NOT NULL default '0',
  PRIMARY KEY  (`id`),
  KEY `EE id` (`EU id`),
  KEY `md5 x509` (`md5 x509`)
) TYPE=InnoDB COMMENT='Demande de certificat d''un ''EndUser''';


# --------------------------------------------------------

#
# Structure de la table `EndUser`
#

CREATE TABLE `EndUser` (
  `id` int(11) NOT NULL auto increment,
  `first name` varchar(40) NOT NULL default '',
  `last name` varchar(40) NOT NULL default '',
  `email` varchar(50) NOT NULL default '',
  `organization_unit` varchar(40) NOT NULL default '',
  `organization` varchar(40) NOT NULL default '',
```

```
  `city` varchar(50) NOT NULL default '',
  `country` char(2) NOT NULL default '',
  `is_AE` tinyint(4) default '0',
  `is_AC` tinyint(4) NOT NULL default '0',
  PRIMARY KEY  (`id`),
  UNIQUE KEY `email` (`email`)
) TYPE=InnoDB COMMENT='Informations sur le EndUser de la PKI';


# --------------------------------------------------------

#
# Structure de la table `Logs`
#

CREATE TABLE `Logs` (
  `id` bigint(20) NOT NULL auto_increment,
  `date action` datetime NOT NULL default '0000-00-00 00:00:00',
  `author` bigint(20) NOT NULL default '0',
  `action` varchar(255) NOT NULL default '',
  `req_id` int(20) NOT NULL default '-1',
  `eu id` int(20) NOT NULL default '-1',
  `author addr` varchar(15) default NULL,
  `is AE` tinyint(4) NOT NULL default '0',
  `is_AC` tinyint(4) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=InnoDB COMMENT='logs all actions';
##Insertion du pemier element vide pour les Logs
INSERT INTO `EndUser` VALUES (1, '', '', '', '', '', '', '', 0, 0);
```

Enfin, il est nécessaire de configurer les fichier `sqlRignts.inc.php` des répertoires `Admin`, `AE`, `AC_aut` et `Titulaire` pour leur permettre de communiquer avec la base de donnée MySQL.

Ci dessous le contenu de ce fichier:

```
<?
//Nom ou adresse de la machine qui héberge MySql
$sqlHost    ='localhost';
//Login de l'utilisateur de la PKI
$sqlLogin   ='ltrustpki';
//Password de l'utilisateur
$sqlPasswd  ='secret';
//Nom de la base de donnée
```

```
$sqlDatabase='ltrustpki';
?>
```

2.3 OpenLdap

Les lignes suivantes devront être ajoutées au fichier slapd.conf:

```
include         /usr/local/etc/openldap/schema/core.schema
include         /usr/local/etc/openldap/schema/cosine.schema
include         /usr/local/etc/openldap/schema/inetorgperson.schema

database        bdb
suffix          "dc=mycompany dc=com"
rootdn          "cn=Manager,dc=mycompany,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          secret
```

Il faudra aussi créer un fichier init.ldif contenant les lignes suivantes:

```
dn: dc=mycompany, dc=com
objectClass: dcObject
objectClass: organization
dc: mycompany
o: mycompany

dn: cn=Manager,dc=mycompany,dc=com
objectClass: OrganizationalRole
cn: Manager
description: manages the ldap directory

dn: ou=people,dc=mycompany,dc=com
objectClass: organizationalUnit
ou: people

dn: cn=greg,ou=people,dc=mycompany,dc=com
objectClass: person
cn: user
sn: user
```

Pour ajouter les champs précédents dans l'annuaire LDAP, taper la commande:

```
ldapadd init.ldif
```

2.4 Local Trust PKI

2.4.1 Paires de clés et OpenSSL

Pour initialiser la PKI, il est nécessaire de créer la clé privée et le certificat de l'autorité de certification, respectivement `ca_pkey.pem` et `ca.pem` et copier ces fichiers dans les répertoires, respectivement `CA_PRIV_KEY` et `CA_CERT` (cf. crypto.inc.php ci-dessous)

Les modules **Admin**, **AE** et **Titulaire** doivent avoir une paire de clé (signée par l'autorité créée précédemment) pour s'authentifier auprès du module **AC_aut**. Le champ `OU` du certificat du module **Admin** doit être égal à "AUTORITE ADMINISTRATION", celui du module **AE** doit être égal à "AUTORITE ENREGISTREMENT" et celui du module **Titulaire** doit être égal à "TITULAIRE".

Les chemins vers ces bi-clés sont déterminés dans les fichiers de configuration `/include/misc.inc.php` des répertoires `Admin` et `AE`, dans la rubrique `access to curl keys`.

Exemple de fichier de configuration `crypto.inc.php` se trouvant dans `AC_aut`

```
<?//##crypto.inc.php
//chemin de la commande openssl
define('OPENSSL', '/usr/local/ssl/bin/openssl');
//chemin du fichier openssl.cnf
define('OPENSSL_CONFIG', '/usr/local/ssl/openssl.cnf');
//section contenant les informations nécessaires à la génération des certificats pour les
utilisateurs
define('USR CERT SECTION', 'usr cert');
//nombre de jours de validité du certificat lors de sa création
define('CERT_DAYS', '365');
//chemin vers la racine de notre application
define('ROOT DIR', '/etc/pki/localtrust');
//chemin du certificat de l'AC root
define('CA CERT', ROOT DIR."/certs/ca.pem");
//chemin de la clé privée de l'AC racine
define('CA_PRIV_KEY', ROOT_DIR."/private/ca_pkey.pem");
//chemin du fichier serial
define('CA SERIAL',  ROOT DIR."/serial");
//chemin de la CRL
define('CRL OUT FILE', ROOT DIR."/crls/crl.crl");
//chemin de la base de données de openssl
define('OPENSSL_DATABASE', ROOT_DIR."/index.txt");
?>
```

Ci-dessous la section de openssl.conf contenant les informations nécessaires à la génération des certificats *"[usr_cert]"*.

```
[ usr cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here
# This is typical in keyUsage for a client certificate.
 keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
```

2.4.2 Module cron

La notification des utilisateurs par mail, la génération des LCR ainsi que la sauvegarde de la base de données se fait par un cron (à créer) dont le contenu est le suivant:

```
SHELL=/bin/bash
PATH=/usr/local/bin:/usr/bin:/bin
MAILTO=""
HOME=path to pki/AC
# run-parts
#Notification des EndUser
* * * * * php notifier eu.php
#génération périodique de la liste de révocation
* * * * * php generer crl.php
```

2.4.3 Création du premier administrateur de l'AC

Enfin, il est nécessaire de créer un premier administrateur de la PKI. Pour ce faire, le "futur" administrateur devra générer une demande de certificat via le module **Titulaire** et lancer le script:

```
php path_to_pki/AC_aut/generer_cert-NOAdmin.php
```

Pour plus de sécurité, il est recommandé de supprimer ce fichier dès que le certificat de l'administrateur est créé.