Pratique

Mise en œuvre d'une PKI avec OpenCA

Azzeddine Ramrami

Degré de difficulté

Cet article est la suite de l'article PKI et applications paru dans le numéro de juin 2007 de hakin9. Il explique comment mettre œuvre une PKI basée sur OpenCA avec un annuaire LDAP et une base de donnée MySQL.

OpenCA est une PKI Open Source qui s'appuie sur les composants technologiques suivants : OpenSSL, Perl, Javascript et sur un serveur Web. Il est disponible à l'adresse : *http://www.openca.org/.*

Le projet OpenCA a démarré dans les années 1999 avec comme idée de rassembler trois parties : une interface *Web en Perl*, *OpenSSL* comme moteur cryptographique et une base de données pour le stockage des certificats. La plus part des opérations sont réalisées au travers de l'interface Web. Nous commencerons par vous montrer les opérations réalisées par OpenCA.

Le logiciel est capable de réaliser les opérations suivantes :

- interface Publique,
- interface LDAP,
- interface RA,
- interface CA,
- interface SCEP,
- interface OCSP,
- filtrage par adresse IP de l'accès aux interfaces,
- accès par mot de passe,
- contrôle d'accès basé sur le Rôle,
- gestion flexible des certificats,
- gestion flexible des extensions des certificats,
- révocation par code PIN,
- révocation par signature numérique,
- gestion des CRL,
- alertes pour l'expiration des certificats,
- support de la quasi-totalité des navigateurs.

OpenCA a été conçu pour une infrastructure distribuée. Il est donc possible de définir plusieurs niveaux hiérarchiques. L'idée étant

Cet article explique...

- La notion de confiance.
- Les PKI et les certificats X509v3.
- Commandes Linux.

Ce qu'il faut savoir...

- Des notions sur la cryptologie.
- HTML, Javascript et MySQL.
- Perl.
- Les protocoles TCP/IP, SSLv3.

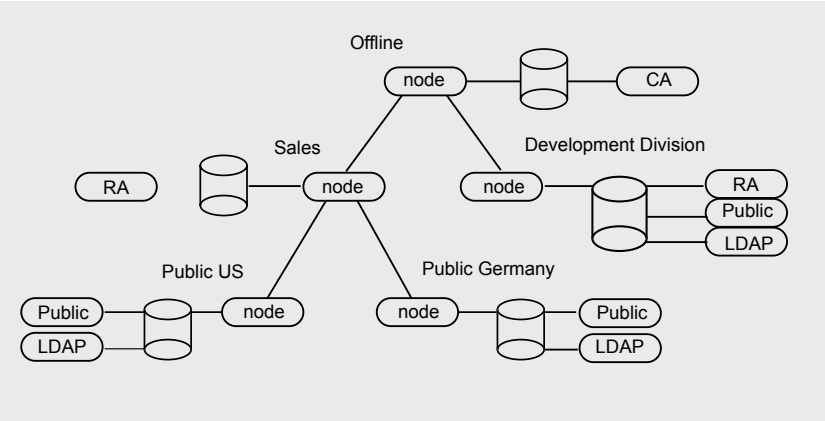

Figure 1. *Organisation logique d'OpenCA*

l'adaptation à des grandes organisations comme des universités ou de grandes entreprises.

Organisation logique

La Figure 1 donne l'organisation logique de la PKI OpenCA.

Derrière une PKI X.509, on trouve une forte organisation hiérarchique. Ainsi, lorsque nous envisageons de créer une architecture PKI distribuée, nous construisons en fait un arbre de bases de données. L'échange de données entre des bases aussi isolées est facile à gérer automatiquement si votre système de base de données est lui même distribué.

Dans le cas d'OpenCA, ce système complexe est géré au sein d'une seule base de données. S'il existe réellement une base de données isolée (comme par exemple une CA hors-ligne), on doit alors fournir la technique d'échange (ou synchronisation) des données et la gestion complète du nœud dans la hiérarchie. Ces fonctionnalités de gestion sont fournies sous la forme d'une interface appelée node.

Normalement, chaque serveur dans l'infrastructure du centre de confiance possède sa propre base de données pour des raisons de sécurité.

Présentation des différentes interfaces

Si on veut normalement concevoir un centre de confiance sérieux, on doit organiser l'ordonnancement de tâches comme le montre la Figure 1 sous la forme d'un ensemble d'interfaces distribuées.

Interface Node

Cette interface permet de gérer les bases de données ainsi que toutes les importations et exportations.

La base de données peut être initialisée depuis cette interface, cela signifie qu'OpenCA est capable de créer toutes les tables nécessaires, sans pour autant être capable de créer la base elle-même.

Cette interface inclut aussi des fonctionnalités de sauvegarde et de restauration. Mais en ce qui concerne la sauvegarde de la clé privée et du certificat de la CA, on doit avoir un autre dispositif. Il n'existe aucune méthode automatique pour sauvegarder la clé privée. Par contre la plupart des CA utilisent des cartes cryptographiques (modules matériels dédiés à la cryptographie, ou HSM) et nécessitent un système de sauvegarde différent et souvent propriétaire. L'importation et l'exportation sont aussi gérées par cette interface.

On peut configurer des règles différentes pour la synchronisation de nœuds à différents niveaux de la hiérarchie. Ceci comprend la configuration des objets et des états qui peuvent être échangées.

Interface CA

L'interface CA (autorité de certification), possède toutes les fonctionnalités pour créer les certificats et les listes de révocation de certificats (CRL). Elle inclut aussi toutes les fonctions destinées à modifier la configuration avec l'interface Web. Aucune autre interface Web ne permet de changer la configuration. Cette interface est aussi l'endroit d'ou peut lancer les taches les taches automatisées.

Interface RA

L'interface RA d'OpenCA est capable de gérer tous les types de requêtes. Entre autres :

- éditer des requêtes,
- approuver des requêtes,
- effacer des requêtes incorrectes et avertir les utilisateurs par e-mail,
- créer les clés privées avec des cartes à puce (*smart cards*).

Interface LDAP

L'interface LDAP a été implémentée pour séparer complètement la gestion de l'annuaire LDAP du reste du logiciel. Ceci a été rendu nécessaire car ces fonctionnalités, purement spécifiques aux administrateurs LDAP, ne sont utilisées que par un très petit nombre d'utilisateurs.

Interface Pub

L'interface Publique fournit toutes ces petites choses si précieuses aux utilisateurs. En voici quelques unes :

Tableau 1. *Matrice des logiciels installés*

Logiciel	CA Server	RA Server	RA Operator
Modules génériques Perl	x	x	x
Modules Perl pour OpenCA	x	x	-
Serveur WWW	x	x	x
Module SSL/TLS	x	x	x
Serveur LDAP (OpenLDAP)	-	-	x
OpenSSL	-	-	-

Tableau 3. *Modules Perl*

CGI::Session	3.95	pour le traitement des sessions OpenCA
Convert::ASN1	0.21	
Digest::HMAC	1.01	pour Authen::SASL
Digest::MD5	2.24	normalement inclus dans Perl
Digest::SHAI	2.02	pour OpenCA
Encode::Unicode	2.20	pour l'internationalisation d'OpenCA
IO::Socket::SSL	0.92	IO::stringy- 2.108
MIME::Based4	2.20	pour le codage/decodage Base64
MIME::Lite	3.01	pour le traitement des emails dans OpenCA
MIME-tools	1.58	pour le traitement des emails dans OpenCA
Net-Server	0.86	pour le demon OpenCA
Parse::RecDes-cent	1.94	Pour X500::DN
URI	1.23	
X500::DN	0.28	
XML::Twig	3.09	used for XML parsing Warning Please read the file README in the distribution of XML: :Twig which you use really carefully. There are several incompatibilities with some versions of XML::Parser and expat. The used version of Perl is heavily important too.
hbintl-perl	1.10	interface pour i18n stuff
perl-ldap	0.28	interface LDAP de Perl

- génération des Requêtes de Signature de Certificats pour Microsoft Internet Explorer.
- génération des CSR pour Firefox.
- génération des requêtes et des clés privées indépendantes du navigateur.
- réception des requêtes au format PKCS#10 ou PKCS#12 générées par certains serveurs.
- inscription des CRL.
- support de deux méthodes de révocation.
- recherche des certificats.
- test de certificats utilisateur dans des navigateurs : Microsoft Internet Explorer, Firefox.

Description des composants OpenCA

Contrairement à d'autres PKI, OpenCA n'est pas un logiciel monolithique, il utilise d'autres composants logiciels issus de la communauté Open Source comme : Apache, mod_ssl, OpenSSL, OpenLDAP, Perl. Le Tableau 1 donne la matrice des logiciels installés. L'ensemble a été déployé sur un système d'exploitation Linux Fedora Core 6 (Voir Tableau 2).

Environnement Perl

La PKI OpenCA repose sur plusieurs modules, certains sont écrits en PERL et d'autres en JavaScript.

Pour faire fonctionner les Scripts CGI écrits en PERL, nous avons besoin d'installer l'interpréteur PERL sur le serveur. Normalement la version Linux Fedora 6 est livrée en standard avec Perl 5.8.8.

Les modules perles suivants doivent être installés dans l'ordre suivant :

- `Convert::BER` c'est une classe d'objets perl qui implémentent pour coder et décoder les objets selon le standard ITU-T X.209 (ASN.1) en utilisant le BER (*Basic Encoding Rules*). Le nom du fichier est appelé *Convert-BER-1.26.tar.gz*,
- `MIME::Base64 and MIME::QuotedPrint` offre le codage/décodage base64 et le code/décodage des caractères spéciaux d'impression. Ces méthodes d'encodage sont décrits dans la RFC 2045 – MIME (*Multipurpose Internet Mail Extensions*). Le nom du fichier est *MIME-Base64-2.11.tar.gz*,
- `The URI` cette classe d'objet perl fournie les fonctionnalités conformément à l'Uniform Resource Identifier, comme spécifié dans la RFC 2396. Le nom du fichier est *URI-1.04.tar.gz*,
- `The Digest::*` cette classe d'objet perl fournie l'implémentation pour les fonctions de hachage MD5 (*RFC 1321*), MD2 (*RFC 1319*) et SHA-1 (*FIPS PUB 180-1*) . Une implémentation de la fonction HMAC (*RFC 2104*) est fournie. Le nom du fichier est *Digest-MD5-2.09.tar.gz*,

Tableau 2. *Liste des logiciels installés*

Logiciel	Type	Version	Site
Linux	Système d'exploitation	Rhedat Core with Kernel 6, 2.6.18	www.fedora-fr.org
Apache	Serveur Web	2.2.3-5	www.apache.org
OpenSSL	Module de cryptographie	0.9.8b-8	www.openssl.org
OpenLDAP	Annuaire LDAPv3	02.03.27	www.openldap.org
OpenCA	Logiciel de l'Infrastructure	0.9.3-rc1	www.openca.org
Perl	Langage de Sript	5.8.8-10	www.perl.com

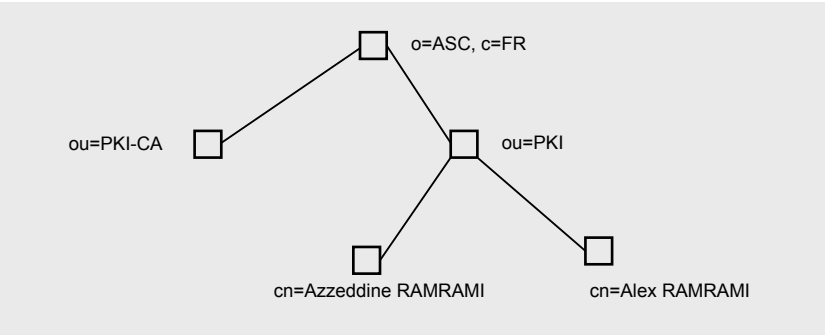

Figure 3. *Schéma de l'annuaire OpenLDAP*

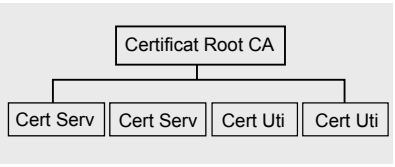

Figure 4. *Architecture de la PKI OpenCA*

- `perl-ldap` fourni les accès aux serveurs LDAP. Le module Convert::BER est un pré-requis pour installer `perl-ldap`. Le nom du fichier est *perl-ldap-0.13.tar.gz*,
- normalement le module perl suivant remplace le module `perl-ldap` *Net-LDAPapi-1.42.tar.gz*.

Pour installer les modules Perl il suffit de décompresser le fichier du module et de taper les commandes suivantes dans le répertoire du module :

```
perl Makefile.PL
make
make test
```

`make install` (si make test Ok à 100%)

Il faut aussi installer les modules spécifiques à OpenCA ci-dessous.

Ils permettent de passer les fichiers de configuration :

- e module perl OpenCA-Configuration est utilisé pour accéder aux fichiers de configuration de la PKI OpenCA. Actuellement les fichiers de configuration sont :

ca.conf
`raserver.conf`
`secure.cnf`

Le nom du fichier est *OpenCA-Configuration-1.2.tar.gz.*

- Le module perl OpenCA-TRISTateCGI est utilisé pour accéder aux variables de configuration qui ont trois états. Il est utilisé pour faciliter l'accès aux fichiers de configuration de la PKI OpenCA. Le nom du fichier est *OpenCA-TRIStateCGI-1.02.tar.gz*.

Le tableau 3 donne les modules Perl et les versions minimales.

Environnement Apache

Comme nous venons de le préciser au-dessus, OpenCA repose sur une architecture Web, dont certaines interfaces sont statiques avec du code HTML et d'autres dynamiques qui intègrent du code JavaScript et du Perl. Dans la version 2.2.3 d'Apache, on n'a pas besoin d'installer mod_ssl. Pour l'inclure, nous avons eu besoin de préciser l'option `=enabled-ssl-shared`

Environnement OpenSSL

OpenSSL est un ensemble d'outils cryptographiques qui implémente les protocoles réseau SSLv3.1 et TLS v1.0 sur lequel s'appuie OpenCA pour générer les clés et les certificats électroniques.

OpenSSL qui implémente les principaux algorithmes et formats standards cryptographiques .C'est un outil. Open source en ligne de commande qui utilise des diverses fonctions cryptographiques des librairies OpenSSL.

Il peut être utilisé pour des raisons suivantes :

- créer des clés RSA, Diffie-Hellman(DH) et DSA,
- signer des certificats X.509,
- générer des listes de révocation (CRL),
- calculer des empreintes numériques,
- chiffrer et déchiffrer des fichiers,
- tester des clients et des serveurs SSL/TLS (https, smtps, pop3s, imaps,..),
- etc.

Dans le cas de l'architecture PKI, nous avons utilisé la version 0.9.8e.

Configuration d'OpenSSL

Pour configurer OpenSSL il suffit de modifier le fichier suivant : */usr/local/ssl/openssl.cf* Afin de tester le fonctionnement d'OpenSSL procéder de la manière suivante :

- taper : `mkdir /certificat`
- taper : `cd /certificat`
- ensuite créer un certificat en tapant la commande : `openssl req -x509 -newkey rsa:1024 -out cacert.pem -outform PEM -days 730`

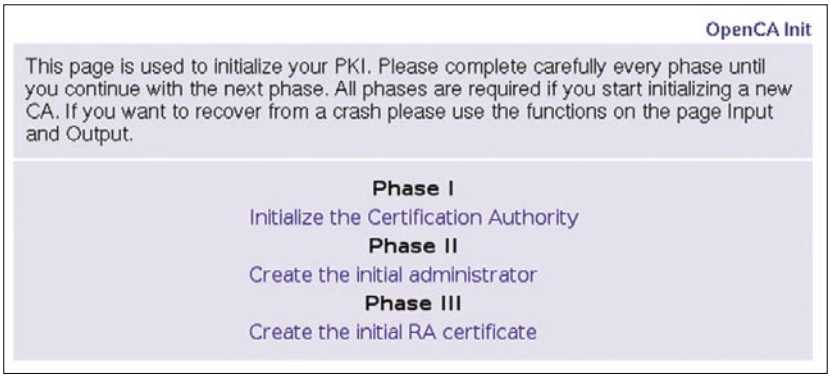

Figure 5. *Initialisation de la CA*

Cette commande crée un certificat en utilisant le format PEM et crée une paire de clé privée et publique en utilisant l'algorithme RSA et une clef de 1024 bits.

- sous */certificat*, on trouve : *cacert.pem*
- sous */certificate*, on trouve la clé privée : *privkey.pem*

Une fois ces différents tests réussis, on est sur que le module de Cryptographie OpenSSL va fonctionner sous OpenCA.

Environnement OpenLDAP

Nous allons vous présenter le schéma de l'annuaire. Comme nous l'avons présenté sur les chapitres précédents, LDAP est un protocole qui permet de définir une méthode d'accès aux données de l'annuaire.

Ces données de la base sont organisées sous une forme logique d'arbre. Dont chaque nœud de l'arbre constitue les différentes entrées de la base. Nous utilisons ici OpenLDAP comme annuaire de notre infrastructure de clés publique dans laquelle seront exportés les différents certificats.

La Figure 3 donne le schéma de l'annuaire LDAP.

Installation des composants OpenLDAP

Pour stocker les données de l'annuaire, nous avons associé à l'annuaire une base de données externe non Relationnelle dont Berkeley DB dont l'accès aux données ne se fait pas au moyen des requêtes SQL mais au Moyen d'un couple de (clé, donnée).Ce type de base est beaucoup plus conseillé car elle offre un service rapide de recherche de donnée comparativement à la base de données standard (MySQL, Oracle, etc.). Afin de contrôler l'accès aux données, nous allons coupler deux méthodes d'authentification : une authentification SASL (*login+ passoir+md5*) & une méthode d'authentification par certificat avec Le protocole TLS.

Berkeley DB

La version que nous utilisons est : *BerkeleyDB-4.5.20*. Cette installa-

Listing 1. *Installation de Cyrus SASL*

```
tar -xvzf cyrus-sasl-2.1.22.tar.gz
cd ./ cyrus-sasl-2.1.22 /dist
./configure -with-bdb-libdir=/usr/local/BerkeleyDB.4.5/lib/
--with-bdb-ncdir=/usr/local/BerkeleyDB.4.5/include/ --with-openssl=/usr/
                    local/ssl/
--with-ldap=/root/openldap-2.3.35
make
make install
ln -s /usr/local/lib/sasl2 /usr/lib/sasl2
ln -s /usr/local/BerkeleyDB.4.5/lib/libdb-4.5.so /usr/lib/libdb-4.5.so
ln -s /usr/local/BerkeleyDB.4.5/lib/libdb-4.5.so /usr/lib/libdb-4.so
ln -s /usr/local/BerkeleyDB.4.5/lib/libdb-4.5.so /usr/lib/libdb.so
ln -s /usr/local/BerkeleyDB.4.5/lib/libdb-4.5.a /usr/lib/libdb
```

Listing 2. *Installation de l'annuaire OpenLDAP*

```
export CPPFLAGS= "-I/usr/local/BerkeleyDB.4.1/include -I/usr/local/include"
export LDFLAGS= "-L/usr/local/BerkeleyDB.4.1/lib -L/usr/local/lib"
tar -xzvf openldap-2.1.31.tar.gz
cd /home/openldap-2.1.31
./configure --prefix=/usr/local/etc/openldap --enable-referrals --enable-ipv6
--with-cyrus-sasl --enable-cleartext --enable-crypt --enable-spasswd --
                    enable-bdb
--enable-dnssrv --enable-dbm --enable-rewrite --enable-ldap --enable-meta
--enable-monitor --enable-slurpd
make depend
make
make test
make install
```

Listing 3. *Installation de l'interface CA*

```
cd openca-0.9.3-rc1
./configure --prefix=/srv/ca --with-node-prefix=online_node -with-web-
                    host=ca.ASC.fr
      --with-httpd-user=apache --with-httpd-group=apache -with-ca-
                    organization=ASC
      --with-ca-locality=Paris --with-ca-country=FR -with-openssl-prefix=/usr/
                    local/ssl
      --with-mailprogram="sendmail t" --with-hierarchy-level=ca -with-web-
                    host=localhost
      --enable-dbi --with-db-type=mysql --with-db-name=openca -with-db-
                    host=localhost
      --with-db-port=3306 --with-db-user=openca --with-db-passwd=secret
make install-ca
```

Listing 4. *Installation de l'interface RA*

```
./configure --prefix=/srv/ca --with-node-prefix=online_node -with-web-
                    host=ra.ASC.fr
      --with-httpd-user=apache --with-httpd-group=apache -with-ca-
                    organization=ASC
      --with-ca-locality=Paris --with-ca-country=FR -with-openssl-prefix=/usr/
                    local/ssl
      --with-mailprogram="sendmail t" --with-hierarchy-level=ca -with-web-
                    host=localhost
      --enable-dbi --with-db-type=mysql --with-db-name=openca -with-db-
                    host=localhost
      --with-db-port=3306 --with-db-user=openca --with-db-passwd=secret
make
make install-ext
```

tion est faite après celle d'OpenSSL de la manière suivante :

- récupérer la distribution sur Internet en tapant la commande : *http://www.oracle.com/technology/software/products/berkeley-db/index.html*,
- se placer dans le répertoire: */temp*,
- taper les commandes :

tar –xvzf db-4.5.20.tar.gz

Figure 6 *Initialisation de la phase 1 de la CA*

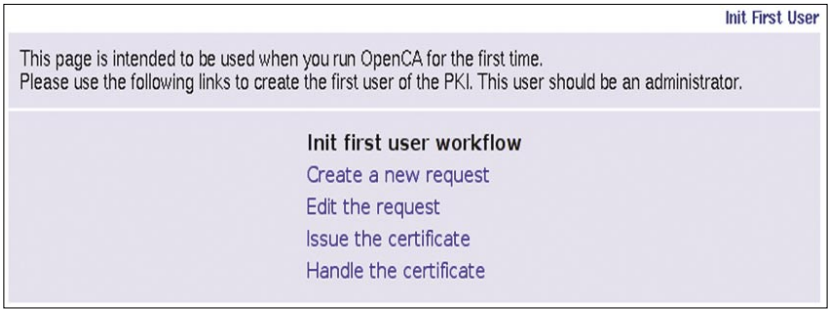

Figure 7. *Initialisation de la phase 2 de la CA*

Figure 8. *Initialisation de la phase 3 de la CA*

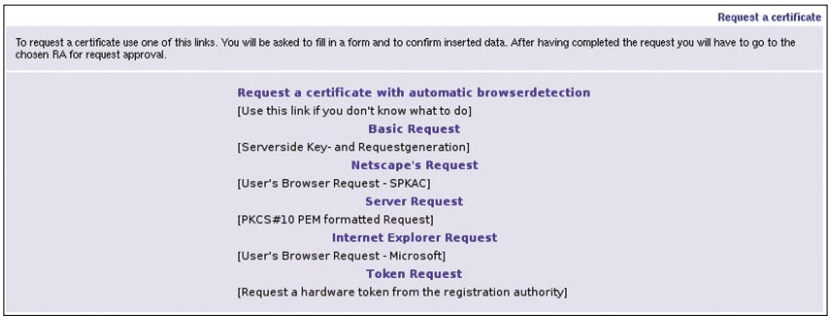

Figure 9. *Génération d'un certificat utilisateur*

```
cd db-4.2.25/build_unix
../dist/configure
make
make install
```

Cyrus SASL

La version que nous utilisons est : *cyrus-sasl-2.1.22* (*cyrus-sasl-2.1.22-4.i386.rpm*).

Taper la commande pour vérifier si le package db4 est installé :

```
rpm -q cyrus-sasl
```

Sinon on peut le construire de la manière suivante :

- récupérer la distribution sur Internet : *http://cyrusimap.web.cmu.edu/download.html#sasl*,
- se placer dans le répertoire : */temp*,
- taper les commandes (Cf. Listing 1).

OpenLDAP

La version que nous utilisons est : *openldap-2.3.35*

- récupérer la nouvelle version sur le site : *www.openldap.org*,
- taper ensuite les commandes suivantes (Cf. Listing 2).

Configuration de l'annuaire OpenLDAP

La configuration de l'annuaire OpenLDAP repose sur 3 fichiers qui s'appuient sur la vue logique du schéma de la base que nous avons défini au-dessus :

Pour la parie serveur : */etc/openldap/slapd.conf*

Pour la définition du nouveau schéma de la base : */etc/openldap/schema/pki.schema*. Voici le contenu de ce fichier :

```
objectclass ( 1.2.840.113533.7.67.4
      NAME 'uniquelyIdentifiedUser'
      SUP top AUXILIARY
      MUST ( serialNumber ) )
```

Ce fichier doit être rajouté, il décrit la structure logique des éléments de l'annuaire que sont les tables.

Ces tables sont vues ici comme

des classes d'objet car on est environnement objet.

Installation de la PKI OpenCA

Le tutorial fourni dans le CD-ROM fournit une explication détaillée pour installer la PKI OpenCA avec les packages nécessaires ainsi que les différents écrans de configuration.

Interface Node

L'architecture logique déployée d'OpenCA à la Figure 4.

Installation de l'AC Root de la PKI OpenCA

D'abord il faut créer deux répertoires */srv* et */srv/ca* et ensuite de suivre les étapes suivantes :

- se connecter en tant que : *root*
- taper la commande : `cd /temp`
- récupérer la dernière version OpenCA sur le site internet : *openca.org*
- taper la commande : `tar -xvzf openca-0.9.3-rc1.tar.gz`
- compiler et installer la CA en tapant (Listing 3).

Installation de l'interface RA

Pour l'installation de la RA et des autres *Interfaces* (PUB, RA, SCEP, LDAP) on procède de la manière suivante :

- taper la commande : `cd /temp/openca-0.9.3-rc1`
- compiler et installer les autres interfaces (RA, PUB, LDAP, SCEP) en tapant (Listing 4).

Remarque : Avant d'installer la RA et les autres interfaces, il est conseillé de modifier le *Makefile* dans la partie *install-ext* en enlevant le commentaire sur SCEP sinon cette interface ne sera pas installée.

Finalisation de la configuration d'OpenCA

Exécuter le script suivant : `/srv/ca/OpenCA/etc/configure_etc.sh`

Ce script va préparer l'interface web d'OpenCA. On va créer un fichier *apache.conf* qui sera stocké dans le répertoire */srv/ca*. Ce fichier contient (Listing 5).

Inclure la ligne suivante dans le fichier de configuration Apache : *httpd.conf* qui se trouve dans le répertoire */etc/http/conf/* : `include /srv/ca/apache.conf`

Exploitation et utilisation de la PKI OpenCA

Nous allons procéder à l'initialisation des interfaces CA, RA de la PKI OpenCA ainsi que l'utilisation de cette PKI pour générer des certificats utilisateurs et serveurs. Pour tester toutes les interfaces de la PKI, il est mieux de procéder à la génération des certificats que nous avons définis lors de la mise en en œuvre de la PKI, c'est à dire procéder de la façon suivante :

- générer un certificat auto signé pour la ROOT CA qui va signer tout le reste des autres certificats,
- générer un certificat pour un opérateur CA,
- générer un certificat pour un opérateur RA,
- générer un certificat pour un utilisateur standard,
- générer un certificat pour un serveur Web (Apache),
- générer un certificat pour un serveur de Messagerie (Sendmail),
- générer un certificat pour un serveur VPN (VPN SSL OpenVPN).

Initialisation de la CA et la RA

Taper : *http://www.ASC.fr/ca*. On obtient l'écran de la Figure 5.

Initialisation de la CA Phase 1

Choisissez ensuite *Initialize the Certification Authority*, puis *Initialize Database*. On obtient l'écran de la Figure 6.

Revenez sur la page précédente, puis sélectionnez *Generate new CA secret Key* et remplissez les champs avec *des3*, *2048*, puis entrez votre mot de passe pour la CA. Cliquez maintenant sur le lien : `Generate new CA Certificate`

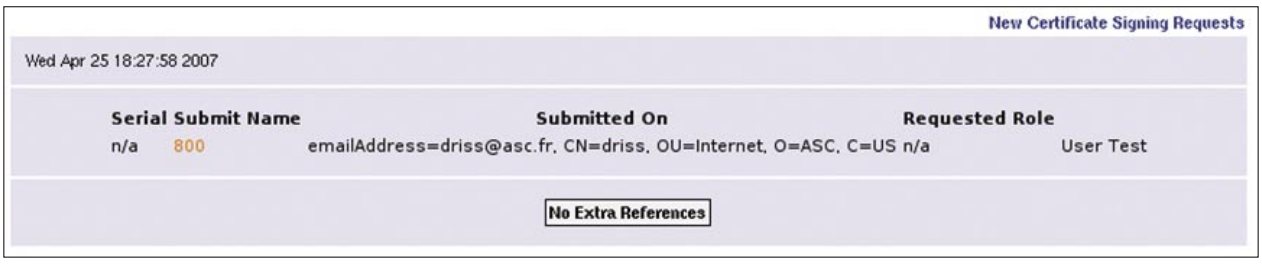

Figure 10. *Validation de la demande des certificats utilisateurs*

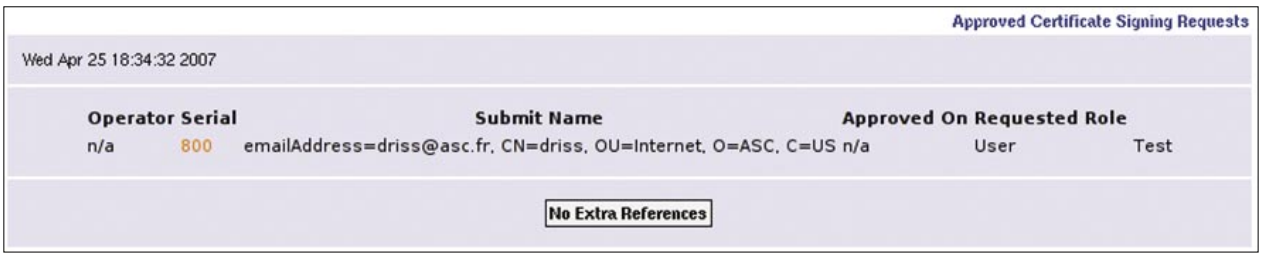

Figure 11. *La RA approuve la demande de certificat*

`Request (use generated secret key)` et remplissez les valeurs adéquates pour notre exemple : `aramrami@ASC.fr, Azzeddine RAMRAMI, ASC PKI, ASC, FR`. Choisissez ensuite : *Self Signed CA Certificate (from already generated request)* avec un nombre de jours égal à 730, puis donnez votre mot de passe de la CA. Revenez à l'écran initial et sélectionnez *Rebuild CA Chain*. Insérez ensuite une disquette vierge dans votre lecteur et assurez vous que nobody a le droit d'écrire dans : */dev/fd0/* (disquette).

Si ce n'est pas le cas, vous pouvez soit, rajouter nobody dans le groupe d'utilisateur autorisés à écrire sur la disquette ou changer les permissions de la disquette pour que tout le monde puisse écrire dedans (`chmod o+w /dev/fd0/`). Cliquez ensuite sur *Export Configuration.*

Initialisation de la CA Phase 2

Pendant cette phase on crée les certificats des opérateurs CA et RA. Revenez sur la page de départ (*http://www.ASC.fr/ca/*) puis cliquez sur *Create the initial administrator* : On obtient l'écran de la Figure 7.

Ensuite cliquez sur *Create a new request*. Vous aurez une certaine quantité d'informations à rentrer concernant le certificat de la personne qui jouera le rôle d'administrateur de vos CA et RA. Voici les valeurs que nous avons rentrées : *aramrami@ASC.fr* azzeddine RAMRAMI, PKI, RA Operator, Trustcenter itself, PIN Code, PIN Code, 1024. Confirmez, puis validez à deux reprises. Cliquez ensuite sur *Edit the request*, puis *OK* et issue the certificate, puis donnez le mot de passe de la CA. Ensuite sélectionnez *Handle the certificate*. Puis dans le champ *Certificate and Keypair*, préférez PKCS#12 à SSLeay (mod_ssl) puis stockez-le en lui donnant un nom avec l'extension p12. Taper la pahsephrase lors de la génération de la clef privée.

Initialisation de la CA Phase 3

Revenez sur la page de départ (*https://www.ASC.fr/ca*) puis cliquez sur *Create the initial RA Certificate*. On obtient l'écran de la Figure 8.

- cliquer sur *Create a new request,*
- cliquer sur *Continue,*
- revenez en arrière et cliquer sur *Edit the Request,*
- cliquer sur *Issue Certificate*. Et taper le mot de passe de la CA,
- revenez en arrière et ensuite cliquer *Handle the certificate*. Puis dans le champ *Certificate and Keypair,*
- préférez PKCS#12 à SSLeay (mod_ssl) puis stockez-le en lui donnant un nom avec l'extension p12 :
- maintenant la CA et la RA sont prêtes on peut passer à l'exploitation de la PKI ainsi installée en générant des certificats utilisateurs et serveurs.

Lancement du serveur de la PKI en SSL

Afin de travailler de manière sécurisée sur le serveur Web de la PKI, il faut commencer par lancer le serveur en mode sécuriser en tapant : `apachectl startssl`. Avant il faut configurer le serveur apache pour fonctionner en mode SSL. Puis importer le certificat de l'autorité de confiance ROOT CA dans le client web Internet Explorer afin qu'il soit reconnu comme autorité de confiance. Il suffit de suivre la procédure suivante pour le faire :

- cliquer sur *Get CA Certificate*.
- puis en choisissant *CA-certificate in format CRT...*
- enregistrer sous : *cacert.crt*
- cliquer ensuite sur le fichier *cacert.crt* pour effectuer l'installation du certificat. De la même façon on peut procéder à l'importation d'un des certificats utilisateurs mais ce n'est pas nécessaire car on n'a juste besoin de s'assurer que l'authentification du serveur Web dont le certificat est signé par la CA ROOT de notre PKI. Et c'est CA ROOT qui signe tous les certificats (utilisateurs, Serveurs,..).

Désormais la PKI repose sur un serveur WEB sécurisé. Pour accéder à

Listing 5. *Fichier apache.conf pour l'interface CA*

```
#www.ASC.FR
#From /srv/ca/apache.conf
<VirtualHost www.ASC.fr:80>
   ServerAdmin aramrami@asc.fr
   DocumentRoot /srv/ca/apache/htdocs
   ServerName www.ASC.fr
   <Directory "/srv/ca/apache/htdocs">
      Options Indexes FollowSymlinks MultiViews
      AllowOverride None
      Order allow,deny
      Allow from all
   </Directory>
   ScriptAlias /cgi-bin/ "/srv/ca/apache/cgi-bin/"
   <Directory "/srv/ca/apache/cgi-bin">
      AllowOverride None
      Options None
      Order allow,deny
      Allow from all
   </Directory>
</VirtualHost>
```

Listing 6. *Fichier /etc/openldap/slapd.conf*

```
uri ldaps://www.ASC.fr
   LDAPv3
   Ssl start_tls
   TLS_CACERT /srv/ca/OpenCA/var/crypto/cacerts/acert.pem
   TLS_KEY /srv/ca/OpenCA/var/crypto/certs/serv_lap.pem
```

toutes les interfaces, il suffit de taper : *https://www.asc.fr/interface*. Avec `Interface=(ca, ra, pub, ldap)`

Pour générer un certificat utilisateur, il faut passer par l'interface publique en procédant de la manière suivante :

Génération d'un certificat utilisateur

taper : *https://www.ASC.fr/pub*

On obtient l'écran de la Figure 9.

- puis cliquez sur *Request Certificate* puis *A certificate with automatic Browser detection,*
- cliquer sur *Continue*,
- cliquer sur *Continue*. Un écran s'affiche pour demande un mot de passe et une fois tapé la génération de la clef démarre,

On constate que la requête généré a un numéro de série : 800,

Pour valider la génération du certificat on va utiliser l'interface RA, en tapant : *https://www.ASC.fr/ra* puis *Active CSRs*. Puis en cliquer sur *New,* On obtient l'écran de la Figure 10.

En cliquant le numéro de série (800), on obtient :

- ensuite cliquer sur : *Approve Request without signing*,
- ensuite utilise utiliser la CA pour signer la demande certificat en tapant: *http://www.ASC.fr/ca*
- en suite sélectionner *Approved*.

On obtient l'écran de la Figure 11.

Ensuite cliquer sur le numéro de série 800 on obtient et cliquer sur *Issued Certificate*.

Maintenant que le certificat utilisateur a été signé et chiffré par la CA, il faut taper : *https://www.ASC.fr/pub*.

Et sélectionner *Certificates*. Ensuite il suffit de cliquer sur le serial du certificat valide et le télécharger. Une fois le certificat chargé il suffit de l'installer au niveau du client : Internet Explorer ou Firefox.

Génération d'un certificat serveur OpenLDAP

La procédure est la même que sur les deux précédentes, on peut utiliser l'annuaire du serveur PKI pour tester les fonctionnalités de TLS et SSL.

Coté serveur

Arrêter le service ldap en tapant : `service ldap stop`

Modifier le fichier de configuration : */etc/openldap/slapd.conf* en rajoutant les directives suivantes (Cf Listing 6). Relancer le service ldap en tapant :

```
slapd -d 5 H ldaps://www.asc.fr:9009/
      /etc/openldap/slapd.conf
```

Coté client

Utiliser un client linux tel que :

```
ldapserach -P 3 -d 4 -H ldaps
      www.asc.fr:9009/ -s base
      -b "o=ASC,c=FR" " azzeddine"
```

Pour tester un client Windows (récupérer un browser ldap libre sur internet/sourceforge), intégrer le certificat de la ROOT CA, ensuite tester l'accès à l'annuaire en mode sécurisé en tapant :

ldaps://www.asc.fr:9009/" nom de la base"

Ne pas oublier que le lien vers la CRL qui liste tous les certificats révoqués et non valide est : *http://www.asc.fr/cacrl.crl*.

Conclusion

L'article a expliqué comment démarrer avec une PKI OpenCA et créer au moins les interfaces de base : CA, RA et PUB. On s'est arrêté à la génération des certificats pour le CA, RA et on a généré un certificat utilisateur et un certificat serveur pour OpenLDAP. À vous d'exploiter cette PKI pour l'utiliser dans les applications suivantes :

- serveur VPN SSL avec Open VPN,
- serveur Apache avec SSL.

La PKI OpenCA offre aussi la possibilité de créer des CA filles afin de créer une PKI multi-filiale. Elle offre les modules pour s'interfacer avec les systèmes HSM et des tokens matériels, elle offre les interfaces OCSP, SCEP. ●

Terminologie

- CA : Certificte Authorithty.
- RA : Registration Autorithy.
- Algorithme de hachage : 3DES, AES, MD5, SHA-1, etc.
- Diffie-Hellman : Algorithme à clef publique inventé en 1976 par Whitfield Diffie et Martin Hellman, c'est le plus ancien crypto système à chiffrement asymétrique.

À propos de l'auteur

L'auteur travaille depuis 15 ans en tant que Consultant Expert en sécurité des systèmes d'information dans la société Sogeti (Paris) ; de plus, il a crée des applications sécurité (chiffrement, Client Checking) pour Windows et Linux et une appliance VPN SSL et VoIP (IP PABX) sous FreeBSD intégrant une PKI complète. L'auteur a mis en œuvre plusieurs projets PKI dans le secteur bancaire et assurance. Son adresse email : *azzeddine.ramrami@free.fr.*

Sur Internet

- OpenCA Web site : *http://www.openca.org*
- OpenCA LiveCD : *http://www.dartmouth.edu/~deploypki/CA/InstallOpenCALiveCD.html*
- Dartmouth PKI Lab : *http://www.dartmouth.edu/~pkilab/*
- Feodra Core 6 : *http://www.fedora.org*
- Norme RSA PKCS : *http://www.rsa.com/rsalabs/node.asp?id=2124*