LE WEB SINGLE SIGN-ON AVEC LEMONLDAP::NG

par Issam Mejri

L'authentification unique ou identification unique, en anglais Single Sign-On (SSO), est une méthode permettant à un utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques. Le Web SSO s'intéresse particulièrement à l'authentification unique pour les applications web et donc, toute application accessible à travers un navigateur web.

Lorsque l'on gère plusieurs applications web en Intranet ou des applications ouvertes à l'extérieur et que ces dernières nécessitent l'authentification de l'utilisateur, ainsi que l'attribution de ses droits, la gestion des utilisateurs et de leur identifiant et mot de passe devient de plus en plus compliquée (d'autant plus que leur nombre augmente considérablement au cours du temps).

En effet, vos utilisateurs doivent mémoriser les logins et mots de passe pour chacune de vos applications ; de plus, vous allez devoir attribuer les droits d'accès séparément pour chaque identifiant. Sans oublier les problèmes liés à l'administration de ces comptes en cas de perte ou d'oubli de mot de passe...

Une solution pour toutes ces contraintes : le Web SSO. Un logiciel libre nous permet de mettre en œuvre cette solution : LemonLDAP::NG.

1 Qu'est ce que LemonLDAP::NG ?

Concrètement LemonLDAP::NG est un portail d'accès qui va authentifier l'utilisateur à travers un annuaire LDAP où sont stockés les comptes utilisateurs. Une fois que cette phase est réalisée, l'utilisateur va pouvoir accéder aux applications de manière transparente et donc, sans se reconnecter à l'annuaire LDAP.

Le Web SSO se consacre à l'authentification unique pour les applications web, donc toute application client-serveur dont le client est un navigateur web. Le principe de fonctionnement est d'intercepter les requêtes entre le client et le serveur et d'indiquer que le client est bien authentifié. LemonLDAP utilise des en-têtes HTTP pour propager l'identité de l'utilisateur.

1.1 Les éléments de LemonLDAP::NG

LemonLDAP repose sur trois composants ou modules Perl :

- Le portail (*portal*) est une interface d'authentification, qui contient également le menu des applications protégées par LemonLDAP, ainsi que la ré-initialisation du mot de passe pour l'utilisateur authentifié.
- Le *handler* est un agent réalisant le contrôle des accès aux applications web.
- Le *manager* est l'interface de configuration et l'explorateur de sessions ; c'est une interface graphique permettant de configurer facilement LemonLDAP (interface réservée à l'administrateur).

Remarque

LemonLDAP a été créé en 2003 par Éric German pour le ministère des finances français. À partir de 2004, le projet a été progressivement repris par le Commandant Xavier Guimard (Gendarmerie Nationale) pour devenir en 2005 LemonLDAP::NG. Les 2 projets ont coexisté quelque temps avant l'abandon définitif du support de LemonLDAP. LemonLDAP::NG est aujourd'hui soutenu en grande partie par le groupe LINAGORA et la Gendarmerie Nationale.

1.2 Principe de fonctionnement

1.2.1 Phase d'authentification

LemonLDAP::NG découpe l'authentification en plusieurs étapes :

- vérification des sessions existantes ;
- récupération des données d'authentification ;
- recherche de l'utilisateur ;
- authentification ;
- stockage des données dans la session ;

- écriture du cookie ;
- redirection vers l'application initiale demandée par l'utilisateur (requête HTTP).

1.2.2 Bases de données employées par LemonLDAP::NG

Pour fonctionner, notre LemonLDAP::NG utilise trois bases de données :

- Une base de données d'authentification permettant le stockage des identifiants utilisateurs. On peut utiliser un *backend* parmi plusieurs (LDAP, DBI, OpenID, CAS...), dans notre cas et dans la suite de cet article, nous utilisons LDAP comme base de données.
- Une base de données de configuration, qui enregistre la configuration globale de LemonLDAP::NG. Elle est aussi gérable à travers le manager. Elle permet également de choisir le type de backend pour le stockage de la configuration (file configuration backend (par défaut), SQL, LDAP, SOAP).
- Une base de données de sessions que LemonLDAP::NG exploite pour la gestion et l'enregistrement des sessions utilisateurs.

Les sessions sont stockées en utilisant le module Apache::Session. On peut également stocker les sessions dans une base SQL, par exemple MySQL, en fournissant le module Apache::Session::MySQL.

1.2.3 Environnement d'exécution

LemonLDAP::NG est développé principalement en Perl, donc ce langage de script doit être présent pour l'exécution des différents modules. Dans la majorité des distributions Linux, Perl est installé par défaut. D'autres briques de logiciels doivent être présentes :

- le serveur web Apache, ainsi que le module Perl associé (`mod_perl`) ;
- OpenLDAP installé et configuré, nous supposons ici que votre serveur LDAP est configuré pour l'authentification des utilisateurs (voir *Linux Pratique* n° 66) ;

Dans la suite de l'article, notre suffixe de base est `dc=pme,dc=com` et nos utilisateurs sont dans l'unité d'organisation `ou=people,dc=pme,dc=com`. Nous utiliserons par ailleurs : Ubuntu 10.04 LTS, Apache 2.2.14, LemonLDAP::NG 1.1.1, OpenLDAP 2.4.26, Perl 5.10.1.

Remarque **Le module Apache::Session**

Apache::Session est un framework qui est particulièrement utile pour le suivi des données de sessions entre les requêtes HTTP. Il est conçu pour fonctionner avec Apache et mod_perl et il peut également fonctionner avec d'autres serveurs web.

2 Passons à la pratique

2.1 Pré-requis et dépendances nécessaires

Pour faire fonctionner LL::NG, il faut que votre serveur web Apache soit éventuellement compilé avec le `mod_perl`. Le serveur Apache peut être exécuté avec l'un des binaires suivants : le MPM (*Multi-Processing Module*) Prefork ou le MPM Worker, ce dernier est celui utilisé par défaut dans le cas de l'installation du paquet `apache2`. L'utilisation du MPM Worker augmente considérablement les performances de LemonLDAP grâce à l'utilisation des threads.

Voici par ailleurs la liste des modules Perl utilisés dans LemonLDAP::NG. Les modules de base doivent être installés sur le système ; d'autres doivent être installés uniquement si vous avez prévu d'utiliser certaines fonctionnalités.

Les modules de base

Apache::Session, Net::LDAP, MIME::Base64, CGI, LWP::UserAgent, Cache::Cache, DBI, XML::Simple, CGI::Session, Regexp::Assemble, XML::LibXML, Crypt::Rijndael, IO::String, XML::LibXSLT, HTML::Template, SOAP::Lite, Config::IniFiles, JSON, Digest::HMAC, Crypt::OpenSSL::RSA, Crypt::OpenSSL::X509, Convert::PEM, Clone.

Tous ces modules peuvent être téléchargés à partir de http://www.cpan.org.

Les autres modules selon le fournisseur d'identité à utiliser (facultatifs)

- SAML 2 : Lasso, GLIB, LemonLDAP::NG::Portal::AuthSAML, LemonLDAP::NG::Portal::_SAML ;
- CAS : AuthCAS, LemonLDAP::Handlers::CAS ;
- OpenID : Net::OpenID::Consumer > 1.00, Net::OpenID::Server > 1.00.

Plus de détails sur ces modules sur le site http://lemonldap-ng.org.

Les autres dépendances

- Jquery (framework Javascript)

2.2 Installation des dépendances

Sur un système Debian-like, on exécutera :

```
sudo apt-get install apache2 libapache2-mod-perl2 libapache-session-perl libnet-ldap-perl libcache-cache-perl libdbi-perl perl-modules libwww-perl libcache-cache-perl libxml-simple-perl  libsoap-lite-perl libhtml-template-perl libregexp-assemble-perl libjs-jquery libxml-libxml-perl libcrypt-rijndael-perl libio-string-perl libxml-libxslt-perl libconfig-inifiles-perl libjson-perl libstring-random-perl libemail-date-format-perl libmime-lite-perl libcrypt-openssl-rsa-perl libdigest-hmac-perl libclone-perl libauthen-sasl-perl
```

2.3 Installation de LemonLDAP

Pour installer LL::NG à partir de votre gestionnaire de paquets, il faut ajouter ces deux lignes dans `/etc/apt/sources.list` :

```
deb http://lemonldap-ng.org/deb squeeze main
deb-src http://lemonldap-ng.org/deb squeeze main
```

On installera ensuite le paquet nommé `lemonldap-ng`. On pourra également choisir de compiler l'application à partir des sources disponibles sur : http://lemonldapng.org/download.

Après l'installation, si bien sûr elle s'est déroulée correctement, vous aurez les répertoires suivants présents sur votre système :

- `/usr/share/doc/lemonldap-ng` est intuitivement le répertoire de documentation ;
- `/var/lib/lemonldap-ng` est le répertoire d'installation de `lemonldap-ng` ;
- `/var/lib/lemonldap-ng/conf` est le répertoire contenant la base de données de configuration ;
- `/var/lib/lemonldap-ng/sessions` est le répertoire contenant la base de données des sessions ;
- `/etc/lemonldap-ng` contient les fichiers de configuration de LemonLDAP, dont le fichier `lemonldap-ng.ini` permettant la configuration manuelle sans passer par l'interface graphique du manager.

3 Configuration d'Apache

L'installation de LL::NG a créé trois fichiers de configuration exploitables par le serveur web Apache : `handler-apache2.conf`, `manager-apache2.conf` et `portal-apache2.conf`. Ces fichiers sont placés lors de l'installation dans `/etc/apache2/sites-available`.

Pour activer cette configuration (hôtes virtuels), il suffit d'exécuter les commandes ci-dessous :

```
a2ensite handler-apache2.conf
a2ensite  manager-apache2.conf
a2ensite portal-apache2.conf
```

Puis on redémarre le serveur web :

```
sudo /etc/init.d/apache2 restart
```

3.1 Résolution adresse IP -> nom d'hôte

Si votre réseau local n'est pas configuré pour traiter des requêtes à destination d'un serveur de noms (DNS), il faut avoir recours au fichier local `/etc/hosts` et ajouter les lignes suivantes :

```
127.0.0.1 local.example.com
127.0.0.1 reload.example.com
127.0.0.1 test1.example.com test2.example.com auth.example.com
127.0.0.1 manager.example.com
```

L'URL local.example.com sera utilisée dans la suite de notre article lors de la création de notre propre hôte virtuel protégé par LL::NG.

3.2 Test de l'installation

Notre installation va donner naissance à un domaine et deux applications protégées : example.com, test1.example.com, test2.example.com, ainsi qu'un accès à la page de configuration manager.example.com. Commençons par ce dernier (Fig. 1).

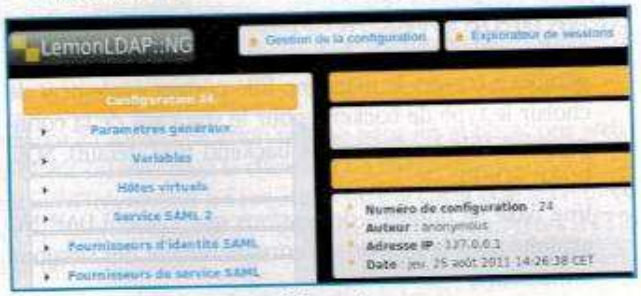

Figure 1

Le manager est non protégé par défaut (accès anonyme), c'est pourquoi par question de sécurité l'accès est réservé à l'administrateur. Je détaillerai dans la suite de cet article comment protéger l'accès au manager. Vérifiez ensuite que vous ne pouvez pas accéder à l'application test2.example.com (un alias à test1.example.com) et que vous êtes redirigé vers le portail d'authentification, dont l'adresse est auth.example.com, qui vous invite à saisir un nom d'utilisateur et un mot de passe (Fig. 2). Si tout fonctionne bien, on peut poursuivre notre chemin avec LL::NG.

Figure 2

4 Configuration de l'authentification

Dans le paragraphe 1.2.3, j'ai mentionné que notre serveur LDAP (OpenLDAP) est installé et configuré pour gérer la base `dc=pme,dc=com` et que la branche `ou=people,dc=pme,dc=com` est créée au niveau de l'annuaire. Cette branche contiendra nos utilisateurs. La figure 3 nous montre la liste des utilisateurs

Figure 3

présents dans l'annuaire. Chaque utilisateur s'authentifie avec les attributs `uid` et `userpassword`. LL::NG dispose de plusieurs modules d'authentification, pour choisir LDAP il faut passer par le manager.

4.1 LDAP comme module d'authentification

Dans la gestion de la configuration du manager, il faut renseigner les champs suivants : dans ***Paramètres généraux***, choisir ***Modules d'authentification***, puis dans la liste déroulante à droite, choisir « LDAP ». Dans ***Module d'utilisateurs*** et ***Module de mot de passe***, choisir également « LDAP ».

Pour que LL::NG localise notre serveur LDAP ainsi que l'emplacement de nos utilisateurs, il faut remplir les champs du menu ***Paramètres LDAP*** : ***Hôte*** > localhost, ***Port*** > 389, ***Base de recherche des utilisateurs*** > ou=people,dc=pme,dc=com. Ceci fait, cliquez sur ***Sauver*** pour enregistrer les modifications.

Pour vérifier que nous sommes sur le bon chemin, essayons de nous connecter à http://test1.example.com avec un compte d'utilisateur présent dans l'annuaire. Nous visualisons alors la page de test illustrée par la figure 4, tout va bien.

Ne négligez pas cette page de test, car elle est pleine d'informations utiles telles que l'entête HTTP créée suite à la requête émise par le client, ainsi que toutes les variables d'environnement d'exécution liées à l'ouverture de la session.

4.2 Un coup d'œil sur l'explorateur de sessions et la base de données de configuration

Une fois qu'un utilisateur est connecté à une application, il n'a pas besoin de s'authentifier une deuxième fois lorsqu'il accède à d'autres applications gérées par le manager, sauf restriction explicite de l'administrateur et c'est le principe même du single sign on.

Un fichier de session est créé dans le répertoire `/var/lib/lemonldap/session` :

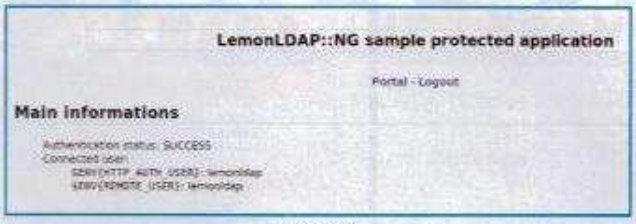

Figure 4

```
ls -l /var/lib/lemonldap/session/
-rw-r--r-- 1 www-data www-data  523 2011-08-28 13:36
17f519a29f7ca9e1462106a33b659db0
drwxr-xr-x 2 www-data www-data 4096 2011-08-28 13:36 lock
```

Ce fichier porte le même nom que l'ID de session de l'utilisateur connecté, qu'on pourra vérifier soit à partir du navigateur web (Fig. 5), soit à partir de l'explorateur de sessions. La session est enregistrée dans l'explorateur de sessions (Fig. 6, page suivante).

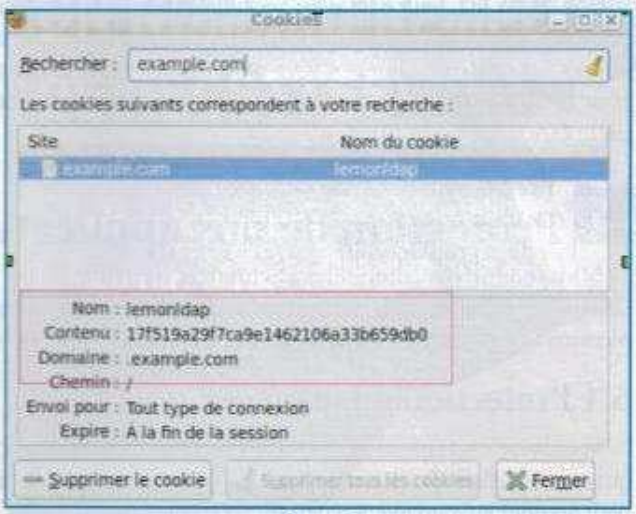

Figure 5

4.3 Base de données de configuration

La base de données de configuration, comme celle des sessions, est un simple répertoire. Chaque modification de la configuration génère une nouvelle version du fichier de configuration, donc le répertoire de la base de données comportera le fichier de configuration associé au numéro de version correspondant.

```
# ls -l /var/lib/lemonldap/conf/
-rw-r----- 1 www-data www-data 3356 2011-08-03 17:17 lmConf-1
-rw-r----- 1 www-data www-data 8154 2011-08-17 01:51 lmConf-2
-rw-r----- 1 www-data www-data 8337 2011-08-17 01:53 lmConf-3
-rw-r----- 1 www-data www-data 8410 2011-08-17 02:07 lmConf-4
-rw-r----- 1 www-data www-data 8469 2011-08-17 02:17 lmConf-5
------------------ Affichage
tronqué----------------------------------
 -rw-r----- 1 www-data www-data 8469 2011-08-17 02:17 lmConf-24
```

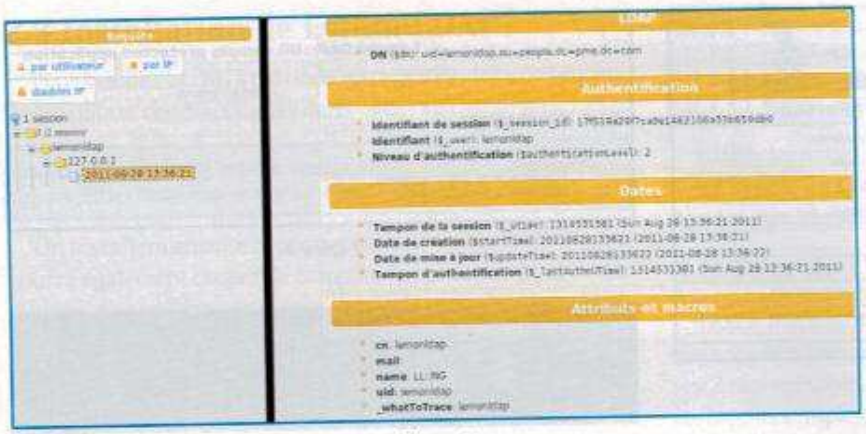

Figure 6

Nous sommes à la version 24 du fichier de configuration (comme indiqué dans la gestion de configuration du manager).

Ce type de configuration de la base de données se limite à un seul serveur LemonLDAP::NG : pour partager la configuration à travers le réseau entre plusieurs serveurs, il faut utiliser d'autres moyens de stockage de la configuration. On peut utiliser une base de données SQL, un annuaire LDAP ou le protocole SOAP. Idem pour la configuration de sessions. L'utilisation de ce type de configuration sort du périmètre de notre article.

5 Protection de nos applications

Maintenant que nous avons compris le principe de fonctionnement de LemonLDAP::NG pour la protection des applications et la gestion des sessions, passons à la protection de notre propre application (site web).

5.1 Protection du manager

Vous avez certainement constaté que l'accès au manager n'est pas protégé et qu'aucune authentification n'est demandée, par conséquent, n'importe quel utilisateur peut accéder au manager à partir de son poste client. En effet, lors de l'installation, le manager n'est pas protégé afin de nous permettre la configuration de LL::NG.

Nous allons donc restreindre l'accès au manager à l'unique utilisateur **issam** qui en est l'administrateur. La protection se base sur l'authentification LDAP d'Apache exploitant le module **authnz_ldap**. Pour réaliser ceci, il suffit d'ajouter ces lignes dans le fichier d'hôte virtuel du manager (**/etc/apache2/sites-enables/manager-apache2.conf**) :

```
Protection du manager par apache
# <Directory /var/lib/lemonldap-ng/manager/>
#   AuthzLDAPAuthoritative On
#   AuthName "LL::NG Manager"
#   AuthType Basic
#   AuthBasicProvider ldap
#   AuthLDAPBindDN "cn=manager,dc=pme,dc=com"
#   AuthLDAPBindPassword "portos"
#   AuthLDAPUrl "ldap://localhost:389/ou=people,dc=pme,dc=com?uid"
#   Require ldap-user issam
#   Options +ExecCGI
# </Directory>
```

Essayons de nous connecter au manager (http://manager.example.com) en tant qu'utilisateur « issam ». Remarquez que l'auteur de la configuration est non plus « anonymous » mais « issam ». Et voilà, maintenant nous sommes en sécurité !

5.2 Protection de notre hôte virtuel

La première étape consiste à ajouter une directive à la configuration de notre hôte virtuel, cette directive permettant la protection de notre hôte par LemonLDAP::NG :

```
PerlHeaderParserHandler My::Package
```

Le fichier suivant (**local.conf**) représente la configuration de notre hôte virtuel : ce fichier est à placer dans le répertoire **/etc/apache2/sites-enabled/**.

```
### Hôte virtuel de test
<Virtualhost *:80>
### Protection par LemonLdap
  PerlHeaderParserHandler My::Package
###### FQDN du site à protéger ######
  Servername local.example.com
  Serveradmin root@exampel.com
  DocumentRoot /var/www/local
  ErrorLog /var/log/apache2/local.log
  CustomLog /var/log/apache2/local_access.log
  combined
<Directory /var/www/local>
  Options Indexes FollowSymlinks Multiviews
  DirectoryIndex index.html admin.html
  Order allow,deny
  Allow from all
</Directory>
</Virtualhost>
```

La deuxième étape nécessite l'ajout de l'hôte virtuel dans la configuration du manager comme suit : dans le menu de configuration, cliquez sur ***Hôtes virtuels***, puis dans le cadre de gauche, cliquez sur ***Nouvel hôte virtuel*** ; saisissez local.example.com, cliquez sur ***Appliquer***, puis ***Sauver***.

Finalement, il ne reste que quelques modifications au niveau des règles de restriction d'accès. La force de LemonLDAP réside dans la gestion des règles d'accès aux applications. En effet, pour chaque application, on peut définir

une ou plusieurs règle(s) qui détermine(nt) les droits des utilisateurs authentifiés. Pour notre hôte virtuel, nous allons définir quatre règles :

Commentaire	Expression	Règle
Accès pour lemonldap	^/pub/	`$mail eq 'lemonldap@pme.com' or $uid eq 'dorra'`
Logout	^/logout	`logout_sso http://auth.example.com`
Restriction rep admin	^/admin/	`$uid eq "issam"`
Règle par défaut	default	`accept`

Tableau 1

Explications :

- L'accès au répertoire **pub/** de notre application, c'est-à-dire l'URL http://local.example.com/pub, n'est autorisé qu'aux utilisateurs dont l'e-mail est équivalent à lemonldap@pme.com ou repéré par l'UID « dorra » dans l'annuaire LDAP ;
- Lorsque l'utilisateur se déconnecte, il est redirigé vers le portail d'authentification ;
- L'accès à l'URL local.example.com/admin est réservé à l'unique utilisateur dont l'UID est « issam » dans l'annuaire LDAP ;
- La règle par défaut est de protéger l'application par LemonLDAP::NG.

Voyons le comportement de l'application si l'utilisateur « lemonldap » tente de se connecter à l'URL local.example.com/admin/ avec ses identifiants : l'accès est refusé (Fig. 7). C'est prévisible, puisque l'accès au répertoire **admin/** est réservé à l'unique utilisateur « issam ».

Figure 7

Conclusion

Comme vous l'avez constaté tout au long de cet article, LemonLDAP::NG est très riche en fonctionnalités. Cet article ne suffit pas à toutes les présenter. Vous pourrez adapter ce logiciel à vos besoins pour votre application ou dans le cadre d'une infrastructure d'entreprise. Pour continuer avec LemonLDAP::NG, rendez-vous sur le site officiel du logiciel : http://lemonldap-ng.org. ■