Installation de l'application WebAuth

Version 1.0 – 21 mars 2004
Pierre David, Jean Benoit

Table des matières

1 Introduction

L'application WebAuth permet de s'affranchir de la gestion statique (via l'utilitaire `htpasswd`) des comptes sur un serveur Web. Elle vous permet de gérer facilement et efficacement les utilisateurs via une base d'authentification, exploitée par le serveur Web Apache et, par exemple, les applications Web développées au CRC.

Les principales caractéristiques de l'application sont :

- gestion centralisée de tous les comptes d'un serveur Web : tous les comptes sont dans une base d'authentification unique ;
- basée atuellement sur une base de données PostgreSQL : l'utilisation d'un annuaire LDAP est tout à fait possible[1] bien que l'utilisation de certaines fonctionnalités évoluées (telles que la recherche phonétique) ne soit alors pas garantie.
- distinction des domaines d'accès : un utilisateur peut appartenir à un ou plusieurs groupes, permettant ainsi à Apache d'offrir l'accès à une partie de l'arborescence à un ou plusieurs groupes ;
- facilement intégrable dans une application : chaque nouvelle application Web peut embarquer des fonctions de changement de mot de passe, de création ou de suppression d'utilisateur, etc.

L'objectif de ce document est de décrire l'installation de l'application WebAuth

Si vous lisez ce document, il est vraisemblable que vous êtes en train d'installer une autre des applications développées au CRC. Dans ce cas, il est conseillé de créer juste un utilisateur avec l'interface de WebAuth, et de créer les autres par l'intermédiaire de l'autre application que vous souhaitez installer.

2 Principe

2.1 Utilisateurs et groupes

L'application WebAuth permet de définir des utilisateurs et des groupes, avec une sémantique proche de `/etc/passwd` et `/etc/group` sur Unix :

- un utilisateur appartient à un ou plusieurs groupes
- un groupe regroupe seulement des utilisateurs (et pas des sous-groupes)

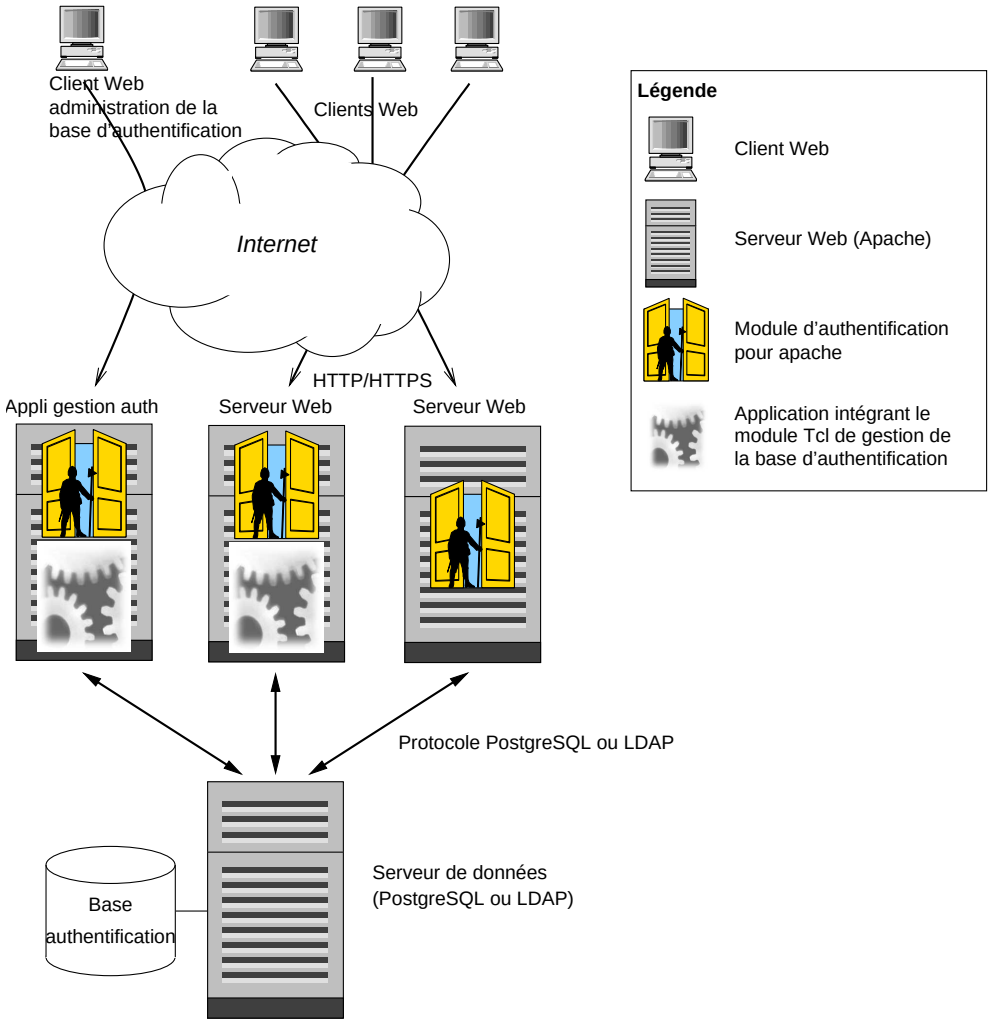
À la différence du modèle `/etc/passwd` et `/etc/group`, un utilisateur n'a pas de « groupe privilégié ».

La notion de groupe est utilisée, en particulier sur le serveur Apache, pour donner l'accès à un ensemble de pages Web (statiques, ou produites par des scripts CGI). En ce sens, la notion de groupe correspond à un « domaine Web » que l'utilisateur peut consulter, sans avoir à donner son mot de passe de multiples fois.

[1] Nous cherchons un ou des volontaires pour effectuer le portage. Merci de vous manifester si vous êtes intéressé.

2.2 Architecture de l'application

Le principe général de fonctionnement de l'application est résumé sur la figure ci-après :



Dans cette figure, l'administrateur accède à la base d'authentification par l'intermédiaire de l'application WebAuth, pour ajouter, supprimer ou modifier des utilisateurs ou des groupes. Le serveur Apache utilise le module approprié (PostgreSQL ou LDAP) pour contrôler l'accès à cette application. WebAuth utilise le module Tcl pour toutes les fonctions de gestion des utilisateurs.

Les autres applications souhaitant utiliser l'authentification peuvent le faire selon deux méthodes :

- soit, sans modification de l'application existante, en ne faisant que contrôler l'accès à certaines parties de l'arborescence Web. Ceci est réalisé simplement par une configuration du serveur Apache, et convient bien aux pages statiques (pages à accès restreint) ou aux applications qu'il serait trop complexe de modifier ;
- soit en modifiant l'application pour y intégrer le module Tcl de gestion des comptes, ce qui est le cas des applications distribuées par le CRC. L'application peut alors dynamiquement créer, supprimer ou modifier des comptes, et les utilisateurs peuvent changer leur mot de passe.

3 Structure de la distribution

La distribution de l'application WebAuth est organisée comme suit :

`doc/`	documentation
`dump/`	répertoire de sauvegarde quotidienne de la base
`expl/`	scripts de maintenance et d'exploitation de la base
`htg/`	générateur de pages Web (répertoire commun à toutes les applications développées au CRC)
`inst/`	scripts d'installation de la base
`pkgtcl/`	paquetages Tcl utilisés par les divers scripts (répertoire commun à toutes les applications développées au CRC)
`www/`	les 2 applications web proprement dites
`www/auth/`	l'application de gestion des utilisateurs
`www/auth/bin`	l'application Web elle-même : les scripts CGI
`www/auth/lib`	fichiers utilisés par les scripts, y compris les pages HTML à trous
`www/passwd/`	l'application de changement de mot de passe
`www/passwd/bin`	l'application Web elle-même : les scripts CGI
`www/passwd/lib`	fichiers utilisés par les scripts, y compris les pages HTML à trous

4 Pré-requis

Cette section décrit les pré-requis avant d'entamer l'installation.

4.1 Composants nécessaires

Les composants logiciels nécessaires pour l'application WebAuth sont décrits ci-après.

4.1.1 Apache

Le premier composant indispensable est un serveur Web. En théorie, tout serveur Web disposant d'une interface CGI est utilisable. En pratique, l'application a été testée avec Apache (versions 1 et 2). Par ailleurs, il est conseillé d'utiliser une extension SSL pour bénéficier d'un bon niveau de sécurité.

Disponibilité : `http://httpd.apache.org/`

4.1.2 Tcl

Le langage utilisé par l'application et nombre de scripts auxiliaires est le langage Tcl (Tool Control Language) développé à l'origine par John Ousterhout. WebAuth a été développé avec la version 8.4.

Il faut noter que Tcl est souvent associé à la boîte à outils graphique Tk. Cette dernière n'est pas utilisée par WebAuth. Cependant, il est possible que la compilation de PostgreSQL (voir ci-après) nécessite Tk.

Disponibilité : `http://tcl.activestate.com/`

4.1.3 PostgreSQL

Le moteur de base de données utilisé est PostgreSQL. Il faut utiliser une version supérieure ou égale à 7.4.

Disponibilité : `http://www.postgresql.org/`

Attention : la compilation de PostgreSQL doit être réalisée avec le support de Tcl. Sous FreeBSD, cela signifie qu'il faut charger les ports de `postgresql` et `postgresql-tcltk`.

Pour vérifier si le support de Tcl est bien compilé, faire :

```
$ tclsh8.4      # ou "tclsh" simplement
% package require Pgtcl
1.4
```

Si vous rencontrez une erreur à la place d'un numéro de version (ici 1.4), le support Tcl pour PostgreSQL n'est pas correctement installé.

Vous devez également compiler le support du langage « PL/Tcl ». Sous FreeBSD, cela signifie qu'il faut charger le port de `postgresql-pltcl`. Pour tester si cela fonctionne, utilisez la commande `createlang` fournie avec PostgreSQL :

```
$ createlang pltcl
```

Si vous rencontrez une erreur, le support de PL/Tcl n'est pas correctement installé.

4.1.4 mod_auth_pgsql

L'authentification des utilisateurs dans l'application est réalisée par le serveur Apache. Pour le moment, la seule authentification définie repose sur une base PostgreSQL, ainsi que sur le module `mod_auth_pgsql`.

Disponibilité : `http://www.giuseppetanzilli.it/mod_auth_pgsql/` (pour Apache 1)

Disponibilité : `http://www.giuseppetanzilli.it/mod_auth_pgsql2/` (pour Apache 2)

4.1.5 pwgen

Le programme `pwgen` permet de générer des mots de passe.

Disponibilité : `http://www.tricknology.org/ports/pwgen-1.15.tar.gz`

4.1.6 LaTeX

Pour accéder aux impressions (génération de fichiers PDF), il faut également une distribution LaTeX contenant au moins le programme `pdflatex` ainsi que les paquetages `babel`, `fontenc`, `palatino`, `geometry`, et `supertabular`.

Tout ceci figure dans l'excellente distribution teTeX que nous vous recommandons par ailleurs.

Disponibilité : `http://www.tug.org/teTeX/`

Si vous ne souhaitez pas installer teTeX, vous ne pourrez pas accéder aux impressions. Nous vous conseillons alors de modifier les pages Web pour ne pas afficher les boutons correspondants.

4.2 Contexte système

4.2.1 Activer les mots de passe PostgreSQL

À moins que votre serveur Web ne soit hébergé sur une machine sans compte utilisateur (autre que les administrateurs), il est souhaitable de configurer PostgreSQL pour que les connexions au moteur nécessitent un mot de passe.

Pour cela, il faut éditer le fichier `~pgsql/data/pg_hba.conf` installé avec PostgreSQL, et modifier les lignes :

```
local all all                           trust
host  all all 127.0.0.1 255.255.255.255 trust
host  all all ::1       ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff trust
```

en :

```
local all all                           password
host  all all 127.0.0.1 255.255.255.255 password
host  all all ::1       ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff password
```

Il faut également modifier certains des scripts de démarrage, et attribuer des mots de passe aux utilisateurs déjà créés à l'aide de l'utilitaire `psql` et de la directive SQL « `ALTER...USER` ». Par exemple :

```
ALTER USER pda WITH PASSWORD 'toto' ;
```

Consultez `http://www.postgresql.org/docs/current/static/client-authentication.html` pour plus d'informations.

Vous pouvez néanmoins procéder au reste de l'installation sans activer la protection par mot de passe, au prix d'une sécurité amoindrie.

4.2.2 Utilisateurs PostgreSQL

Il est conseillé de créer des comptes PostgreSQL (avec l'utilitaire `createuser`) pour les utilisateurs pouvant intervenir lors des opérations d'administration de la base. La création de comptes spécifiques simplifie les opérations.

En outre, les divers scripts et exemples de configuration nécessitent la création d'un utilisateur PostgreSQL nommé « `auth` », avec le minimum de droits (en particulier, pas le droit de créer une base ou de nouveaux utilisateurs). Toutes les opérations d'authentification ainsi que les opérations effectuées par l'application le seront sous cette identité. Il n'y a pas besoin de créer un compte Unix pour `auth`.

Une fois créés, vous pouvez affecter des mots de passe à ces utilisateurs.

4.2.3 Accès à PostgreSQL depuis le serveur Apache

Si vous localisez le serveur Apache et le serveur PostgreSQL sur des machines différentes, prenez bien soin à autoriser l'accès du serveur Apache sur le serveur PostgreSQL. Pour cela, modifiez le fichier `pg_hba.conf` et insérez la ligne suivante :

```
host auth auth 192.168.1.2 255.255.255.255 password
```

Cette ligne autorise l'accès à la base `auth` par l'utilisateur `auth` depuis la machine d'adresse IPv4 192.168.1.2. Bien sûr, si l'accès se fait par IPv6, vous remplacerez l'adresse et le masque par les valeurs appropriées.

5 Personnalisation des pages HTML et LaTeX

Parmi les possibilités de personnalisation de l'application WebAuth figure en bonne place la faculté de modifier les pages HTML et LaTeX de l'application.

En effet, la présentation est largement indépendante de la logique des scripts, et repose sur le concept de « page à trous ».

5.1 Principe des « pages à trous »

Lorsqu'un script CGI souhaite afficher une information, il la place dans un fond de page HTML à un endroit déterminé par un « trou ».

Par exemple, une page servant à confirmer l'enregistrement d'une demande pourrait se limiter à :

```
<HTML>
<TITLE>Demande enregistrée</TITLE>
<BODY>
    <H1>Demande enregistrée</H1>
    <STRONG>%MESSAGE%</STRONG>
    <P>
    <A HREF="%HOMEURL%">Revenir au menu principal</A>
</BODY>
</HTML>
```

Cette page contient deux trous : à la place du premier, « `%MESSAGE%` », les scripts de l'application insèrent le contenu du message de confirmation. À la place du second, « `%HOMEURL%` », les scripts insèrent l'adresse Web de l'application.

Ce principe s'applique également aux fichiers LaTeX utilisés pour générer les fichiers PDF, lors des demandes d'impression.

L'annexe A décrit les fichiers HTML et les trous que les scripts CGI de l'application utilisent.

5.2 Utilisation du générateur « htg »

Les pages HTML, au CRC, sont générées automatiquement à partir d'un mini-langage (HTG, pour HTML Generator). L'outil HTG, qui permet de transformer un source HTG en fichier HTML, est fourni (répertoire `./htg`) ainsi que sa documentation (répertoire `./htg/doc`).

L'intérêt de l'utilisation de HTG est de dissocier complètement le fond de la forme. L'utilisateur de HTG ne spécifie que le fond, et HTG s'occupe de la forme par le biais de *modèles*. Trois types de modèles sont fournis :

- modèle « crc » : utilisé au Centre Réseau Communication de l'Université Louis Pasteur de Strasbourg ;
- modèle « csi » : utilisé au Centre de Services Informatiques de l'Université de Versailles/St-Quentin-en-Yvelines ;
- modèle « vide » : utilisé comme exemple, avec le minimum de fioritures d'affichage.

Pour compiler « `htg` », allez dans le répertoire `./htg/src` :

- éditez le fichier `Makefile` et lancez la compilation avec `make` ;
- éditez le fichier `htg` pour modifier, en première ligne, la localisation exacte du fichier `htgtcl` compilé dans la précédente étape.

Ensuite, sélectionnez le type de modèle que vous préférez en changeant le lien `./htg/modeles`.

5.3 Quelles pages HTML utiliser ?

Plusieurs solutions s'offrent à vous :

- utiliser HTG : dans ce cas, il vous suffit de reprendre les pages du CRC, en les modifiant éventuellement pour indiquer des titres plus représentatifs de votre service.
  Concernant le choix du modèle, vous pouvez alors :
  - reprendre le modèle « crc »
  - reprendre le modèle « vide »
  - concevoir un nouveau modèle (cette option n'est pas documentée ici)
- utiliser des fonds de page HTML que vous aurez conçus vous-même.

La section 7.4 fournit plus de détails sur l'installation de l'application.

6 Installation de la base d'authentification

Vous n'avez pas besoin des droits de « `root` » pour effectuer les opérations décrites dans cette section.

6.1 Installation de la base PostgreSQL

Ce chapitre décrit la mise en place minimale de la base de données utilisée comme support de l'application WebAuth. Il s'agit de créer la base et d'y introduire le minimum d'information nécessaire pour pouvoir effectuer le paramétrage dans le menu d'administration.

6.1.1 Création de la base

Examinez le script `./inst/creer-base`. Dans ce script :

- modifiez votre mot de passe PostgreSQL ;
- mettez en commentaire la ligne « `exit 0` » située vers le début du fichier. Lorsque vous aurez exécuté le script, remettez le `#` qui vous protégera ainsi d'une maladresse si vite arrivée !
- modifiez les logins des utilisateurs privilégiés. Ces utilisateurs (PostgreSQL) doivent pouvoir réaliser toutes les opérations dans la base. Pour cela, tous les droits sont donnés aux tables de l'application.

Vous pouvez à présent exécuter le script (après avoir changé de répertoire dans `./inst`).

Pour vérifier si tout s'est bien passé, vous pouvez utiliser « `psql` » pour passer les deux commandes `\dt` (afficher les tables) et `\q` (sortir) :

```
$ psql auth
auth=# \dt
     List of relations
 Schema |     Name     | Type  | Owner
--------+--------------+-------+-------
 public | config       | table | pda
 public | groupes      | table | pda
 public | membres      | table | pda
 public | utilisateurs | table | pda
(4 rows)

auth=# \q
```

6.1.2 Installation des données minimales

Le script `./inst/init-base` contient le minimum, c'est à dire l'insertion d'un groupe (`authadmin`) et d'un utilisateur dans ce groupe.

Dans ce script :

- modifiez le login, le mot de passe, le nom et le prénom par vos coordonnées. Il s'agit ici d'insérer le minimum pour pouvoir ajouter les autres utilisateur par l'application Web. Le mot de passe doit être chiffré : reprenez par exemple celui qui est dans votre fichier `/etc/passwd` ou équivalent. Si vous laissez le mot de passe par défaut, pensez à le changer le plus rapidement possible lorsque l'application sera opérationnelle.
- modifiez également la ligne affectant votre nom de login au groupe `authadmin`.
- mettez en commentaire la ligne « `exit 0` » située vers le début du fichier. Lorsque vous aurez exécuté le script, remettez le # qui vous protégera ainsi d'une maladresse si vite arrivée !

Lancez le script.

Pour vérifier si l'installation s'est bien passée, vous pouvez utiliser « `psql` » pour consulter ce qui a été mis dans la base :

```
$ psql auth
auth=# \encoding latin9

auth=# SELECT * FROM groupes ;
  groupe   |              descr
-----------+---------------------------------
 authadmin | Administrateurs de la base Auth
(1 row)

auth=# SELECT login, password, nom, prenom, phnom, phprenom FROM utilisateurs ;
 login |   password   |  nom  | prenom | phnom | phprenom
-------+--------------+-------+--------+-------+----------
 pda   | xxxxxxxxxxxx | DAVID | Pierre | D930  | P600
(1 row)

auth=# SELECT * FROM membres ;
 login |  groupe
-------+-----------
 pda   | authadmin
(1 row)

auth=# \q
```

Vous pouvez constater que les colonnes `phnom` et `phprenom` ont été mises à jour automatiquement : elles contiennent une chaîne représentant la forme phonétique du nom et du prénom, ce qui permettra ultérieurement des recherches par approximation.

6.1.3 Regénération de votre mot de passe

Si vous perdez votre mot de passe, vous pouvez le regénérer simplement avec la commande `psql` :

```
$ psql auth
auth=# UPDATE utilisateurs SET password = 'xxxx' WHERE login = 'pda' ;
UPDATE 1
```

```
auth=# \q
```

Dans cet exemple, la chaîne « xxxx » représente le mot de passe chiffré tel qu'il figure dans le fichier /etc/master.passwd ou équivalent.

6.2 Installation de la base LDAP

Cette section est à écrire lorsque l'application sera portée sur LDAP.

7 Installation de l'application de gestion des utilisateurs

Hormis le paramétrage du serveur Apache, vous n'avez pas besoin des droits de « root » pour effectuer les opérations décrites dans cette section.

7.1 Compilation du programme trpw

Le programme trpw affiche sur la sortie standard la forme chiffrée d'un mot de passe donné en argument. Pour le compiler, faire :

```
$ cd ./pkgtcl
$ cc -o trpw trpw.c -lcrypt
```

Suivant le type et la version de système que vous utilisez, vous devrez éventuellement changer la librairie.

Une fois compilé, vérifiez que le programme fonctionne bien :

```
$ cd ./pkgtcl
$ ./trpw toto
G(aXj3KElrHaU
$
```

La chaîne affichée correspond au mot de passe chiffré, qui varie car le « sel » est généré aléatoirement.

7.2 Configuration du package d'authentification

Modifiez les deux lignes du fichier ./pktcl/auth.tcl :

```
variable trpw       "/local/services/www/pkgtcl/trpw"
variable genpw      "/usr/local/bin/pwgen --numerals 8"
```

pour refléter la localisation des commandes :

- trpw (voir 7.1, page 10)
- pwgen (voir 4.1.5, page 5)

7.3 Configuration du package des applications Web

Modifiez les lignes du fichier ./pktcl/webapp.tcl :

```
variable pdflatex   /usr/local/bin/pdflatex
variable sendmail   {/usr/sbin/sendmail -t}
```

pour refléter la localisation des commandes :

- `pdflatex` (voir 4.1.6, page 5)
- `sendmail`, utilisé pour envoyer un mail lorsqu'un mot de passe est réinitialisé.

7.4 Installation des fichiers de l'application

Choisissez un répertoire pour placer les pages Web et les scripts CGI de l'application de gestion des utilisateurs, qui ne doit pas être le répertoire dans lequel vous avez détarré l'application. Dans l'installation par défaut, ce répertoire est nommé `/local/services/www/applis/auth/`.

- si vous souhaitez utiliser HTG (voir 5.2, page 7), utilisez les fichiers « `.htgt` » du répertoire `./www/auth/lib/` en modifiant éventuellement les parties « bannière », « titrepage » et « bandeau » ;
- si vous souhaitez concevoir des nouvelles pages à trous, installez-les dans le répertoire `./www/auth/lib/`. Vous devrez supprimer chaque fichier « .htgt » et le remplacer par un fichier « .html » équivalent, en respectant le nom des trous que les scripts CGI s'attendent à trouver (voir annexe A). Vous prendrez soin également à adapter le fichier LaTeX `utiliste.tex`.

Rendez-vous ensuite dans le répertoire `./www/auth` et éditez le fichier `Makefile`. Modifiez en particulier les variables :

Variable	Signification
`TCLSH`	localisation de l'exécutable `tclsh`
`AUTH`	paramètres d'accès à la base d'authentification
`HOMEURL`	chemin relatif à la racine de l'arborescence Web
`DESTDIR`	localisation de l'application dans l'arborescence Web
`PKGTCL`	localisation des packages Tcl inclus avec l'application
`HTG`	localisation de l'exécutable `htg`
`ROOT`	utilisateurs habilités à intervenir en mode « maintenance »
`NOLOGIN`	nom du fichier à créer pour rentrer en mode « maintenance »

Puis, lancez `make` (dans le répertoire `./www/auth`) pour installer tous les fichiers de l'application dans l'arborescence Web.

7.5 Configuration du serveur Apache

Le serveur Apache doit être configuré pour :

- autoriser l'accès en consultation à `/local/services/www/applis/auth/`
- autoriser l'accès en exécution CGI à `/local/services/www/applis/auth/bin`
- interdire tout accès à `/local/services/www/applis/auth/lib`

7.5.1 Avec l'authentification PostgreSQL

Ceci peut être réalisé grâce aux quelques lignes suivantes (voir première partie du fichier `./inst/httpd.conf`) dans le fichier `httpd.conf` de configuration d'Apache, que vous prendrez soin d'adapter :

```
ScriptAlias "/applis/auth/bin/" "/local/services/www/applis/auth/bin/"
```

```
<Directory /local/services/www/applis/auth>
    #
    # Ces lignes peuvent astucieusement être mises en commun
    # en les déclarant dans le répertoire racine de votre
    # serveur Web.
    #
    AuthName     "Intranet CRC"
    Auth_PG_host            localhost
    Auth_PG_port            5432
    Auth_PG_database        auth
    Auth_PG_user            auth
    Auth_PG_pwd             mot-de-passe-en-clair-de-auth
    Auth_PG_pwd_table       utilisateurs
    Auth_PG_uid_field       login
    Auth_PG_pwd_field       password
    Auth_PG_grp_table       membres

    # Attention : version avec mod_auth_pgsql
    #Auth_PG_gid_field       groupe
    # Attention : version avec mod_auth_pgsql2
    Auth_PG_grp_group_field groupe
    Auth_PG_grp_user_field  login

    #
    # Fin des lignes pouvant être mises en commun dans le
    # répertoire racine de votre serveur Web.
    #

    AuthType            Basic
    require             group authadmin

    # si vous avez une page prévue pour signaler les erreurs, mettez-la ici
    ErrorDocument       401 /errauth/erreur.html
</Directory>

<Directory /local/services/www/applis/auth/lib>
    order deny,allow
    deny from all
</Directory>

Alias "/applis/auth" "/local/services/www/applis/auth"

#
# Pour effectuer en une seule opération
# - l'accès via l'url /applis/auth, qui redirige en réalité vers un script
# - les redirections vers HTTPS
#
RedirectMatch permanent ^/applis/auth/$ \
                https://www-crc.u-strasbg.fr/applis/auth/bin/accueil
RedirectMatch permanent ^/applis/auth/index.html$ \
                https://www-crc.u-strasbg.fr/applis/auth/bin/accueil
```

7.5.2 Avec l'authentification LDAP

Cette section est à écrire lorsque l'application sera portée sur LDAP.

7.6 Paramétrage de l'application

Une fois les étapes précédentes effectuées, vous devriez être en mesure d'accéder à l'URL de votre application.

7.6.1 Paramètres de configuration

La première étape consiste à rentrer dans le module « administration » (modification des paramètres) afin de finaliser les paramétrages.

L'aide en ligne intégrée (lien correspondant à chaque paramètre) fournit des exemples que vous pouvez facilement adapter.

7.6.2 Configuration des groupes et des utilisateurs

Un groupe correspond à un ensemble de pages Web accessibles sur votre serveur Web. En ce sens, la notion de groupe correspond à un « domaine Web »[2].

Avant de créer un groupe, il faut donc réfléchir au périmètre auquel les utilisateurs de ce groupe auront accès.

Le premier groupe créé correspond aux utilisateurs ayant droit à accéder à l'application WebAuth, et donc par là à gérer les comptes.

Le deuxième groupe que vous allez créer correspondra vraisemblablement aux utilisateurs de l'autre application du CRC que vous souhaitez installer (voir introduction).

L'ajout des utilisateurs peut être réalisé soit par l'interface Web (conseillé), soit par l'écriture d'un script comparable à celui fourni pour le remplissage initial (voir fichier `./inst/init-base`).

7.7 Script auxiliaire de maintenance de la base

L'application WebAuth est complétée par un script auxiliaire, lancé par `cron` pour réaliser les opérations de maintenance et de sauvegarde de la base PostgreSQL. Ce script, `./expl/quotidien`, effectue une sauvegarde dans le répertoire `./dump`, ainsi qu'un « `VACUUM` » (spécifique PostgreSQL) sur la base.

De plus, il permet également de créer une copie de la base d'exploitation dans une base de développement, mais ceci n'est pas activé par défaut.

Après l'avoir modifié selon vos besoins, vous pouvez le lancer toutes les nuits par `cron`, de préférence avant minuit pour avoir des noms de fichiers de sauvegarde représentatifs du jour sauvegardé. Voici, pour exemple, une ligne de `crontab` possible (voir fichier `./expl/crontab.auth`) :

```
30 22 * * * $HOME/expl/quotidien
```

8 Installation de l'application de changement de mot de passe

L'application WebAuth vient avec un exemple d'application de changement de mot de passe, accessible par n'importe quel utilisateur. Si vous choisissez de l'installer, voici comment procéder. La démarche est très similaire à l'installation de l'application de gestion des utilisateurs.

[2]Tant que le nom du « royaume d'authentification, correspondant à la directive Apache « `AuthName` », est le même, l'utilisateur n'aura pas à re-saisir son mot de passe.

8.1 Installation des fichiers de l'application

Choisissez un répertoire pour placer les pages Web et les scripts CGI de l'application, qui ne doit pas être le répertoire dans lequel vous avez détarré l'application. Dans l'installation par défaut, ce répertoire est nommé `/local/services/www/applis/passwd/`.

Comme précédemment (voir 7.4, page 11), choisissez la manière dont vous voulez construire les pages HTML, puis éditez le fichier `Makefile` du répertoire `./www/passwd` (les variables sont les mêmes).

Enfin, lancez `make` (dans le répertoire `./www/passwd`) pour installer tous les fichiers de l'application dans l'arborescence Web.

8.2 Configuration du serveur Apache

Comme précédemment (voir 7.5, page 11), le serveur Apache doit être configuré pour :

– autoriser l'accès en consultation à `/local/services/www/applis/passwd/`
– autoriser l'accès en exécution CGI à `/local/services/www/applis/passwd/bin`
– interdire tout accès à `/local/services/www/applis/passwd/lib`

8.2.1 Avec l'authentification PostgreSQL

Ceci peut être réalisé grâce aux quelques lignes suivantes (voir deuxième partie du fichier `./inst/httpd.conf`) dans le fichier `httpd.conf` de configuration d'Apache, que vous prendrez soin d'adapter :

```
ScriptAlias "/applis/passwd/bin/" "/local/services/www/applis/passwd/bin/"

<Directory /local/services/www/applis/passwd>
    #
    # Ces lignes peuvent astucieusement être mises en commun
    # en les déclarant dans le répertoire racine de votre
    # serveur Web.
    #
    AuthName     "Intranet CRC"
    Auth_PG_host            localhost
    Auth_PG_port            5432
    Auth_PG_database        auth
    Auth_PG_user            auth
    Auth_PG_pwd             mot-de-passe-en-clair-de-auth
    Auth_PG_pwd_table       utilisateurs
    Auth_PG_uid_field       login
    Auth_PG_pwd_field       password
    Auth_PG_grp_table       membres

    # Attention : version avec mod_auth_pgsql
    #Auth_PG_gid_field       groupe
    # Attention : version avec mod_auth_pgsql2
    Auth_PG_grp_group_field groupe
    Auth_PG_grp_user_field  login

    #
    # Fin des lignes pouvant être mises en commun dans le
    # répertoire racine de votre serveur Web.
    #

    AuthType            Basic
    require             valid-user
```

```
    # si vous avez une page prévue pour signaler les erreurs, mettez-la ici
    ErrorDocument       401 /errauth/erreur.html
</Directory>

<Directory /local/services/www/applis/passwd/lib>
    order deny,allow
    deny from all
</Directory>

Alias "/applis/passwd" "/local/services/www/applis/passwd"

#
# Pour effectuer en une seule opération
# - l'accès via l'url /applis/passwd, qui redirige en réalité vers un script
# - les redirections vers HTTPS
#
RedirectMatch permanent ^/applis/passwd/$ \
                https://www-crc.u-strasbg.fr/applis/passwd/bin/passwd
RedirectMatch permanent ^/applis/passwd/index.html$ \
                https://www-crc.u-strasbg.fr/applis/passwd/bin/passwd
```

8.2.2 Avec l'authentification LDAP

Cette section est à écrire lorsque l'application sera portée sur LDAP.

9 Conclusion

Si ça marche, n'oubliez pas d'envoyer une bouteille de champagne aux auteurs...

A Pages à trous de l'application de gestion des utilisateurs

Fichier	Trou	Signification
(tous)	%HOMEURL%	adresse relative de la page d'accueil par rapport à la racine du serveur Web
(beaucoup)	%URL%	URL du script de gestion des utilisateurs
actionok.html	%TITREACTION%	titre de l'action effectuée
	%COMPLEMENT%	texte apportant un complément d'information le cas échéant.
admparliste.html	%TAB%	tableau contenant les paramètres éditables de l'application
erreur.html	%MESSAGE%	message d'erreur
grpajout.html	%GROUPES%	liste des groupes existant dans la base
grpconsult.html	%GROUPES%	liste des groupes existant dans la base
grpmodif.html	%MENUGROUPE%	menu HTML permettant de sélectionner le groupe à modifier
grpsuppr.html	%MENUGROUPE%	menu HTML permettant de sélectionner le groupe à modifier
grptraitemodif.html	%GROUPE%	groupe en cours de modification
	%VRAISMEMBRES%	liste actuelle des membres du groupe en cours de modication
	%DESCR%	description du groupe en cours de modification
	%LISTETOUS%	liste HTML des utilisateurs hors du groupe dans le groupe
	%LISTEMEMBRES%	liste HTML des utilisateurs membres du le groupe
utichoix.html	%MESSAGE%	message affiché au cas où plusieurs utilisateurs ont été trouvés
	%LISTEUTILISATEURS%	liste des utilisateurs trouvés
	%AUCUN%	formulaire de re-sélection si aucun utilisateur n'a été trouvé
utiliste.html et utiliste.tex	%NBUTILISATEURS%	nombre d'utilisateurs trouvés
	%S%	« s » si plusieurs utilisateurs ont été trouvés
	%DATE%	date
	%HEURE%	heure
	%TABLEAU%	tableau HTML présentant la liste des utilisateurs
utimodif.html	%TITRE%	nom de l'action qui va être effectuée
	%ACTION%	action à effectuer à la suite de la modification des paramètres
	%ETAT%	état
	%PARAMUTILISATEUR%	formulaire d'édition des paramètres de l'utilisateur
utipasswd.html	%LOGIN%	nom de login de l'utilisateur
	%PRENOM%	prénom de l'utilisateur
	%NOM%	nom de l'utilisateur
utisel.html	%MESSAGE%	message préalable aux critères de sélection
	%ACTION%	action à effectuer après la sélection
	%CRITERES%	formulaire de saisie des critères de sélection
utisuppr.html	%LOGIN%	nom de login de l'utilisateur

B Pages à trous de l'application de changement de mot de passe

Fichier	Trou	Signification
(tous)	%HOMEURL%	adresse relative de la page d'accueil par rapport à la racine du serveur Web
erreur.html	%MESSAGE%	message d'erreur
pwdchoix.html	(aucun)	(aucun)
pwdok.html	(aucun)	(aucun)