|  |  |
| --- | --- |
|  | **Authentification des clients** |

# Présentation

L’authentification HTTP de base est relativement simple à mettre en pratique pour qui connait un peu le serveur web Apache2.

Il existe plusieurs façons de générer des fenêtres avec formulaire d'authentification mais il existe également une solution simple soit htpasswd.   
htpasswd est un utilitaire dont la fonction est de stocker les mots de passe sous une forme chiffrée à utiliser par Apache dans les services d'authentification. Son utilisation permet d’accéder à certains répertoires ou fichiers uniquement à des utilisateurs identifiés.

# Authentification de base et Authentification Digest

Il existe de nombreux **schémas d'authentification disponibles**.

Ce tableau illustre la compatibilité des différents schémas d'authentification sur différents systèmes d'exploitation. Notez que si on utilise **HTTPS**, on suppose que le **certificat SSL est valide** (non auto-signé).

|  |  |  |
| --- | --- | --- |
|  | HTTP | HTTPS |
| Authentification de base | MacOS et Linux | MacOS, Linux et Windows |
| Authentification Digest | MacOS, Linux et Windows | MacOS, Linux et Windows |

**Authentification Digest**

Dans un premier temps il faut **générer le fichier** (appelé users.password) qui **emmagasine les mots de passe pour les utilisateurs**.   
Dans l'authentification Digest, il existe le **champ domaine qui agit comme un espace de noms pour les utilisateurs**.

Pour cet exemple, le premier utilisateur s'appellera tux.

Pour générer le fichier digest, il faut **installer les dépendances pour Apache** …  
**>>** sudo apt install apache2-utils

# Mise en place de l’authentification

## Ajout des utilisateurs

On ajoute des utilisateurs par la suite.

Pour **générer le fichier de mot de passe de l'utilisateur** ...  
**>>** sudo htdigest -c /etc/apache2/users.password secure tux

La commande possède la structure suivante …

* **-c** permet de créer le fichier ;
* **/etc/apache2/users.password** est le chemin absolu du fichier ;
* **secure** est le contexte dans lequel les noms d’usagers et mots de passe seront utilisés ;
* **tux** est le nom d'utilisateur.

Cela ajoute l'utilisateur tux au fichier de mot de passe.   
La **saisie d'un mot de passe est demandée** afin de créer le mot de passe pour l'utilisateur tux.

**Remarque** …  
Pour l'ajout ultérieur d'utilisateurs, il faudra **omettre le commutateur -c car le fichier de mots de passe est déjà créé**.

Voici un autre exemple d'ajout d'un utilisateur nommé squiddly …  
**>>** sudo htdigest /etc/apache2/users.password secure squiddly

**Remarque** …  
Pour afficher la liste des noms d’utilisateurs …  
**>>** sudo cat /etc/apache2/users.password

La prochaine étape est de **permettre à Apache de lire (utiliser) le fichier de mot de passe**.   
Il faut donc en modifier le propriétaire …  
**>>** sudo chown www-data:www-data /etc/apache2/users.password

Une fois le fichier de mot de passe créé, on doit apporter les **modifications qui suivent à la configuration dans /etc/apache2/sites-available/000-default.conf** …  
**>>** nano /etc/apache2/sites-available/000-default.conf

On ajoute les lignes suivantes au fichier …  
 **AuthType Digest  
 AuthName "secure"  
 AuthUserFile /etc/apache2/users.password  
 Require valid-user**

La version finale devrait ressembler à ceci (avec les commentaires supprimés).

**<VirtualHost \*:80>  
 ServerAdmin webmaster@localhost  
 DocumentRoot /var/www/html**

**<Directory /var/www/html>  
 AuthType Digest  
 AuthName "secure"  
 AuthUserFile /etc/apache2/users.password  
 Require valid-user  
 </Directory>**

**ErrorLog ${APACHE\_LOG\_DIR}/error.log  
 CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
</VirtualHost>**

**Quelques précisions** …

* Le **module mod\_auth** contient les **définitions des directives d'authentification** ;
* La **directive AuthType** indique à Apache que pour le répertoire /var/www/html, il devrait y avoir une **authentification utilisant le schéma Digest**.
* L'**authentification** **Digest** nécessite une **valeur pour le paramètre realm (AuthName)** défini comme secure.   
  Le **paramètre realm agit comme un espace de noms**.   
  Lorsque l'on a des utilisateurs portant le même nom, il est possible de les **différencier en utilisant des valeurs différentes pour le répertoire**.   
  L'utilisation de la **directive AuthName permet de définir la valeur du répertoire**.
* La **directive AuthUserFile** est utilisée afin d'indiquer l'emplacement du fichier de mot de passe.
* La **directive Require** indique que seuls les utilisateurs valides qui s'authentifient peuvent accéder à ce répertoire.

## Redémarrage d’Apache2

Enfin, il faut activer le module Digest et redémarrer le serveur pour que les paramètres prennent effet …  
**>>** sudo a2enmod auth\_digest  
**>>** sudo systemctl restart apache2.service

Lors de la prochaine connexion, Apache2 demande une identification afin d’accéder au site Web.

Une image contenant texte

Description générée automatiquement

# Autorisation d'accès à plusieurs personnes

Dans la plupart des cas, on devra autoriser l'accès à plusieurs personnes.   
C'est ici qu'intervient la **directive AuthGroupFile**.

Son contenu se présente comme suit ...  
<Nom de Groupe>: <Utilisateur1 Utilisateur 2 Utilisateur 3 Utilisateur 4>

On crée le fichier suivant …  
**>>** sudo nano "/etc/apache2/groups.password"

Son contenu est …  
**mascottes: tux squiddly kermit**

Il s'agit simplement une liste des membres du groupe sous la forme d'une ligne séparée par des espaces.

Pour ajouter un utilisateur au fichier de mots de passe préexistant …  
**>>** sudo htdigest /etc/apache2/users.password secure kermit

On obtiendra le même effet qu'auparavant, mais le mot de passe sera ajouté au fichier, plutôt que d'en créer un nouveau.

Comme prochaine étape, on devra modifier son fichier .htaccess ou la section <Directory> de la manière suivante …  
**<VirtualHost \*:80>  
 ServerAdmin webmaster@localhost  
 DocumentRoot /var/www/html**

**<Directory /var/www/html>  
 AuthType Digest  
 AuthName "secure"  
 AuthUserFile "/etc/apache2/users.password"  
 Require valid-user  
 AuthGroupFile "/etc/apache2/groups.password"  
 Require group "secure"  
 </Directory>**

**ErrorLog ${APACHE\_LOG\_DIR}/error.log  
 CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
</VirtualHost>**

Enfin, quiconque appartient au **groupe <Nom de Groupe>**, et possède une entrée dans le fichier password pourra accéder au répertoire s'il tape le bon mot de passe.

## Redémarrage d’Apache2

Enfin, il faut activer le module authz\_groupfile et redémarrer le serveur pour que les paramètres prennent effet …  
**>>** sudo a2enmod authz\_groupfile  
**>>** sudo systemctl restart apache2.service

**Remarque** …

Il existe une autre méthode moins contraignante pour autoriser l'accès à plusieurs personnes.   
Plutôt que de créer un fichier de groupes, il suffit d'ajouter la directive suivante …  
**Require valid-user  
Require tux**

Le remplacement de la ligne **Require user tux** par la ligne **Require valid-user** autorisera l'accès à quiconque possédant une entrée dans le fichier password, et ayant saisi le bon mot de passe.