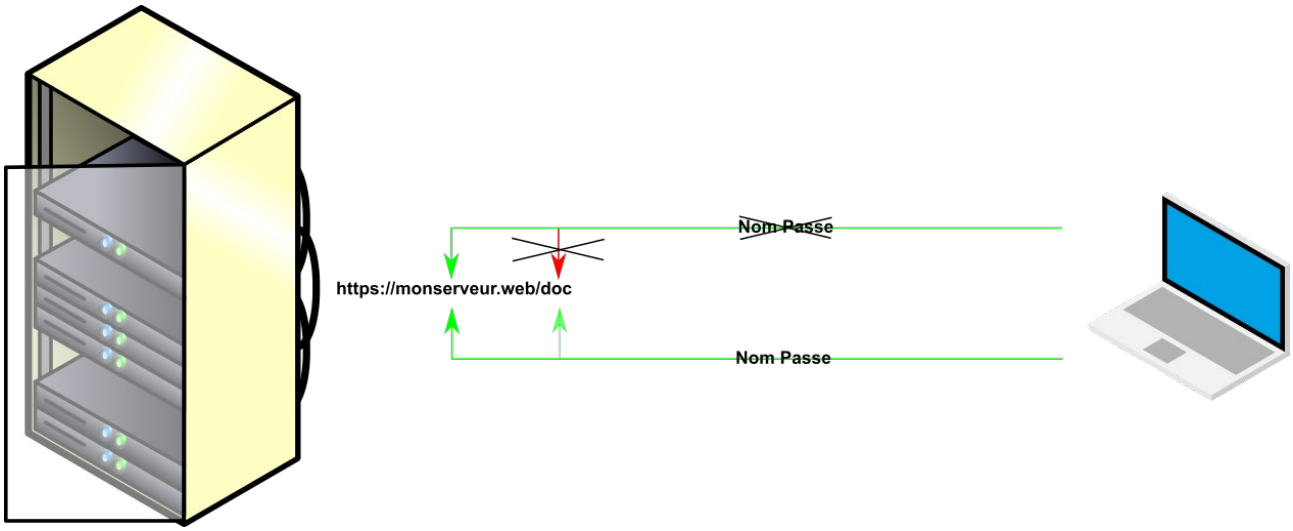
PROTÉGER L'ACCÈS A UN RÉPERTOIRE PAR NOM D'UTILISATEUR ET MOT DE PASSE

Table des matières

OBJET DU DOCUMENT

Il s'agit pour un site web public de bloquer l'accès à un dossier de l'arborescence de ce dernier. Mais en autorisant des comptes avec un nom utilisateur et un mot de passe prédéfinis.



Cette mise en place s'effectuant sur un serveur d'hébergement mis en place avec « Yunohost » et son application « mywebapps ». Sachant que ce serveur fourni une application web sous « nginx ». Cette solution a été mise en place à partir d'échanges entre des membres de la communauté « Yunohost » que je remercie.

LIENS UTILES

TITRE	LIEN UTILES
Yunohost	https://yunohost.org/#/whatsyunohost_fr
Forum yunohost	https://forum.yunohost.org/c/announcement/8
Mywebapps	https://github.com/YunoHost-Apps/my_webapp_ynh
Éditeur nano	https://debian-facile.org/doc:editeurs:nano
Ligne de commande cp	http://www.gogolplex.org/?linux-manipuler-fichiers-et-dossiers
Ligne de commande cp	https://manpages.debian.org/buster/coreutils/cp.1.en.html
Documentation nginx	https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/
Créer le mot de passe	https://www.geek17.com/fr/content/nginx-comment-proteger-un-dossier-ou-une-page-par-un-mot-de-passe-34#:~:text=Cr%C3%A9ation%20du%20fichier%20de%20protection,le%20mot%20de%20passe%20hash%C3%A9.
Mot de passe openssl	https://www.mkssoftware.com/docs/man1/openssl_passwd.1.asp
Mise en place dans différents contextes	https://geekflare.com/fr/protect-page-with-password/

DÉMARCHE

Après s'être **connecté en ssh** et sous « root ».

Il faut **générer** l'utilisateur et le mot de passe dans un fichier.

`.htpasswd`

que je choisis de mettre dans le répertoire à protéger.

Puis nous allons **sauvegarder** le fichier :

`/etc/nginx/conf.d/nom-de-domaine.d/webpapp_*.conf`

que nous allons modifier

Enfin nous **adapterons** le nouveau fichier :

`webpapp_*.conf`

GÉNÉRER LE COMPTE

Pour ne pas afficher en clair notre mot de passe dans un fichier de configuration, nous allons commencer par générer un mot de passe au format « APR-1 ».
Pour générer ce mot de passe, nous allons utiliser « openssl » avec la commande suivante qui demandera de taper un mot de passe, puis de le confirmer.

```
openssl passwd -apr1
Password:
Verifying - Password:
$apr1$EndwnhD8$LPy5If9aTsK7ykmlrfh/61
```

La chaine $apr1$EndwnhD8$LPy5If9aTsK7ykmlrfh/61 est le mot de passe hashé.
Copier cette chaîne pour la coller dans le fichier qui nous allons créer ci-dessous.

Imaginons que nous voulons protéger le répertoire /admin. Nous allons donc créer le ficher .htpasswd dans ce dossier : il contiendra le nom d'utilisateur et le mot de passe hashé.

```
sudo nano /www/doc/admin/.htpasswd
```

Dans ce nouveau fichier, nous collons le mot de passe généré précédemment. Puis ajoutons le nom de l'utilisateur devant, en utilisant « : » en séparateur entre les 2 chaînes. Par exemple, avec l'utilisateur admin.

```
admin:$apr1$EndwnhD8$LPy5If9aTsK7ykmlrfh/61
```

SAUVEGARDER LE FICHIER .CONF

Sauvegarde l'ancien fichier avec la commande « cp ». Le fichier est dans le répertoire : `/etc/nginx/conf.d/nom-de-domaine.d/webpapp_*.conf`

```
cp -a webpapp_*.conf webpapp_*.conf.backup
```

ADAPTER LE FICHIER .CONF

Nous allons ouvrir ce fichier :

```
nano webpapp_*.conf
```

et à la fin y ajouter :

```
location /api {
    auth_basic          “Zone d’identification”;
    auth_basic_user_file /www/doc/admin/.htpasswd;
}
```

La directive `auth_basic` donne le nom de la zone qui sera affichée dans la boîte de dialogue d'identification.
Après la directive `auth_basic_user_file` il faut indiquer le chemin vers le fichier contenant les identifiants créés.