

Sécuriser les connexions SSH avec Google Authenticator

Testé sur Debian 8

Installation

Installer les packages pour compiler les outils nécessaires

```
apt-get install build-essential make automake m4 autotools-dev libtool  
libpam0g-dev libpam0g unzip qrencode
```

(Ici unzip sert à décompresser l'archive, qrencode sert à générer le QR pour configurer l'application mobile plus facilement. Le reste sert à la compilation.)

Télécharger et décompresser les sources de libpam authenticator sur Github (voir lien "Source 3" en bas de page)

```
cd /usr/local/src  
wget https://github.com/google/google-authenticator/archive/master.zip -O  
googleauth.zip  
unzip googleauth.zip && rm googleauth.zip  
cd google-authenticator-master/libpam
```

Penser à adapter l'URL pour la dernière version et donc le nom du dossier qui est extrait

Lancer la compilation

```
./bootstrap.sh  
./configure  
make  
make install
```

Configuration

Cette étape génère un fichier de configuration pour l'utilisateur courant seulement ! Pensez à faire des liens symboliques pour d'autres utilisateurs, ou bien lancer la commande pour ces utilisateurs pour obtenir leur propre fichier de configuration.

La connexion en root est non recommandée avec cette méthode. Créer la configuration pour un utilisateur standard, qui servira d'accès au compte root.

Saisir la commande

google-authenticator

Cinq questions vont être posées, les réponses YES sont les plus sécurisées, à adapter à vos besoins

```
Do you want authentication tokens to be time-based (y/n)
```

Yes

Vous obtenez une adresse de QRCode ou une clé de configuration pour Google Authenticator sur votre mobile. Gardez aussi en sécurité les 5 codes de secours.

```
Do you want me to update your "/root/.google_authenticator" file (y/n)
```

Yes

Le fichier sera généré pour l'utilisateur courant SEULEMENT !

```
Do you want to disallow multiple uses of the same authentication token? This restricts you to one login about every 30s, but it increases your chances to notice or even prevent man-in-the-middle attacks (y/n)
```

Yes

```
By default, tokens are good for 30 seconds and in order to compensate for possible time-skew between the client and the server, we allow an extra token before and after the current time. If you experience problems with poor time synchronization, you can increase the window from its default size of 1:30min to about 4min. Do you want to do so (y/n)
```

Yes

```
If the computer that you are logging into isn't hardened against brute-force login attempts, you can enable rate-limiting for the authentication module. By default, this limits attackers to no more than 3 login attempts every 30s. Do you want to enable rate-limiting (y/n)
```

Yes

Un QR sera affiché, et/ou une clé secrète, pour configurer votre Google Authenticator.

Configurer PAM

Editer le fichier de configuration PAM

```
/etc/pam.d/sshd
```

Ajouter à la fin

```
auth required pam_google_authenticator.so nullok
```

Le paramètre nullok permet à un compte utilisateur, sans configuration de double authentification de se connecter normalement en SSH. Enlever nullok pour forcer la double authentification sur TOUT les comptes.

Configuration SSH

Editer le fichier suivant

```
/etc/ssh/sshd_config
```

Chercher "ChallengeResponseAuthentication" et remplacer "no" par "yes"

Redémarrer le service ssh

```
/etc/init.d/ssh restart
```

Tester la connexion

Vérifier la configuration dans "/root/.google_authenticator" ou "/home/user/.google_authenticator"

Tester la connexion en SSH.

Si problème, vérifier "/var/log/auth.log".

[Source 1](#) - [Source 2](#) - [Source 3](#)

From:

<https://wiki.dureuil.info/> - **GD-WIKI**

Permanent link:

https://wiki.dureuil.info/doku.php/linux:ssh_gauth?rev=1456009943

Last update: **2020/07/24 22:03**

