

Introduction

Dans cette documentation on va mettre en place le MFA pour l'accès ssh sur un serveur Linux, sa permettra de garantir un haut niveau de sécurité, même si le mdp est trouvé il y'aura une autre barrière à fournir.

Mise en place

Sa sera du MFA google, en utilisant l'appli authenticator

On fait une petite update de la liste des paquets et on installe ce paquet qui est un module à PAM pour MFA selon son nom

```
root@sadek:~# apt-get install libpam-google-authenticator
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libpam-google-authenticator
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 45.5 kB of archives.
After this operation, 138 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 libpam-google-authentic
or amd64 20191231-2 [45.5 kB]
Fetched 45.5 kB in 1s (84.6 kB/s)
Selecting previously unselected package libpam-google-authenticator.
(Reading database ... 97138 files and directories currently installed.)
Preparing to unpack ../libpam-google-authenticator_20191231-2_amd64.deb ...
Unpacking libpam-google-authenticator (20191231-2) ...
Setting up libpam-google-authenticator (20191231-2) ...
Processing triggers for man-db (2.9.4-2) ...
root@sadek:~#
```

Ensuite exécuter cette commande : google-authenticator

Ensuite une question

La première pour savoir si on veut que le mdp soit temporaire

Ensuite un QR code s'affiche qu'il faut scanner avec son appli

```
root@adek:~# google-authenticator
Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to Google:
https://www.google.com/chart?chs=200x200&chld=M|0&cht=sqr6chl=otpauth://totp/root@adek.oVhK3Fsecret%3D1C2U4KAE8T2MMFPG4MWN3UME%26issuer%3Dadek.oVh
[QR code]
Your new secret key is: [redacted]
Enter code from app (-1 to skip):
```

Le code temporaire qui s'affiche sur notre appli doit être saisi en bas

Ensuite un code de secours s'affiche au cas ou

```
Code confirmed
Your emergency scratch codes are:
790
529
2174
4113
6904
```

Ensuite on nous demande si on veut mettre à jour le fichier. google_authenticator , on dit oui.

Ensuite on nous demande si on veut désactiver l'authentification à plusieurs users avec le même token, on dit oui

Ensuite on dit oui pour un token toute les 30 sec c'est amplement suffisant

Ensuite on active le blocage pendant 30 sec en cas de tentative 3 fois non réussis

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

By default, a new token is generated every 30 seconds by the mobile app.
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. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) y

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) y
root@sadek:~#
```

Ensuite il faut modifier ce fichier

/etc/pam.d/sshd

Et on rajoute cette ligne à la fin du fichier

```
auth required pam_google_authenticator.so
```

Ensuite il faut modifier le fichier de conf du serveur sshd

Ici il faut faire passer de « no » à « yes »

```
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication yes
```

Test

Ensuite on redémarre le service ssh on garde notre connexion et on ouvre une nouvelle fenêtre de connexion

Et voila

```
PS C:\Users\A.sadek-ext> ssh -p root@smtp.sadek.ovh
(root@smtp.sadek.ovh) Password:
(root@smtp.sadek.ovh) Verification code:
Linux sadek.ovh 5.10.0-19-cloud-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Thu Dec 21 15:58:41 2023 from 90.83.123.116
root@sadek:~#
```

Remarque

Avec mobaXterm sa ne fonctionne pas mais client ssh basique oui.