

BAMBA N'GIANTCHAN ALLASSANE

DIAKITE ABDOUL JUNIOR

DJOMAN ANNE ANKRAN

FOFANA ABDOUL ISMAEL

RAPPORT : ACTIVER L'AUTHENTICATION À DEUX FACTEURS POUR SSH SOUS LINUX VIA GOOGLE AUTHENTICATOR

Sommaire

- 1 Présentation
- 2 Synchronisation horaire
- 3 Installation de Google Authenticator
- 4 Installation de Google Authenticator sur le smartphone
- 5 Recuperation de la cle d'installation
- 6 Configuration de l'authentification a deux (02) facteurs
- 7 Vérification de l'authentification a deux (02) facteurs

1 Présentation

Nous apprendrons à activer l'authentification à deux facteurs pour SSH sous Linux. Il est principalement utilisé par les administrateurs pour accéder et gérer en toute sécurité les ordinateurs distants, en particulier ceux qui exécutent Linux. Il prend en charge deux formes d'authentification très courantes l'authentification par mot de passe et l'authentification par clé publique. Il est toujours recommandé d'utiliser l'authentification par clé publique pour l'accès SSH car elle est plus sécurisée. Toutefois, si vous dépendez toujours de l'authentification par mot de passe, il est fortement recommandé d'activer l'authentification à deux facteurs

2 Synchronisation horaire

Avant d'activer 2FA, il est très important que les horloges de votre système Linux et de votre appareil mobile soient synchronisées, ce qui signifie que les deux doivent avoir la même heure dans leurs horloges. Cependant, une différence de temps nominale de quelques secondes (p. ex., 30 secondes ou moins) est généralement acceptable.

La meilleure façon de garder l'heure synchronisée est d'utiliser le protocole NTP (Network Time Protocol). Par défaut, Ubuntu utilise *timedatectl* / *timesyncd* pour la synchronisation de l'heure, et nous nous en tiendrons à la valeur par défaut. Pour vérifier si le service NTP est activé et si l'horloge du système est correctement synchronisée, utilisez la commande *timedatectl*.

```
abdoul@abdoul-virtual-machine:~/Desktop$ timedatectl
      Local time: Wed 2023-02-08 21:48:23 GMT
      Universal time: Wed 2023-02-08 21:48:23 UTC
      RTC time: Wed 2023-02-08 21:48:23
      Time zone: Africa/Abidjan (GMT, +0000)
System clock synchronized: yes
      NTP service: active
      RTC in local TZ: no
abdoul@abdoul-virtual-machine:~/Desktop$
```

Si vous voyez *Horloge système synchronisée : no* et *Service NTP : inactif*

Cette commande active le service NTP et synchronise l'horloge du système

```
Sudo timedatectl set-ntp true
```

. Répétez maintenant la commande *timedatectl* une fois de plus.

Vous remarquerez que l'horloge système est maintenant correctement synchronisée.

3 Installation de Google Authenticator

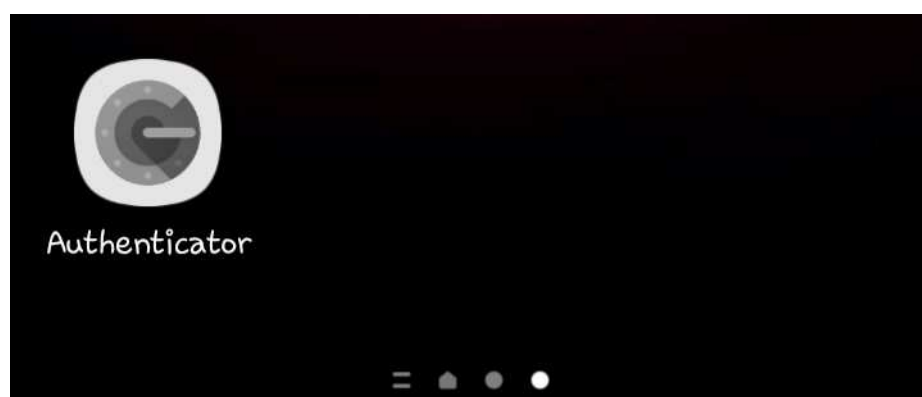
Après avoir synchronisé les horloges sur votre système Linux et votre smartphone, accédez à la console Linux et exécutez la commande suivante pour installer les packages PAM Google Authenticator :

```
sudo apt install libpam-google-authenticator -y
```

```
abdoul@abdoul-virtual-machine:~/Desktop$ sudo apt install libpam-google-authenti
cator -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libqrencode4
The following NEW packages will be installed:
  libpam-google-authenticator libqrencode4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 57.3 kB of archives.
After this operation, 190 kB of additional disk space will be used.
Get:1 http://ci.archive.ubuntu.com/ubuntu focal/universe amd64 libqrencode4 amd6
4 4.0.2-2 [23.6 kB]
Get:2 http://ci.archive.ubuntu.com/ubuntu focal/universe amd64 libpam-google-aut
henticator amd64 20170702-2 [33.7 kB]
Fetched 57.3 kB in 36s (1,572 B/s)
Selecting previously unselected package libqrencode4:amd64.
(Reading database ... 157220 files and directories currently installed.)
Preparing to unpack .../libqrencode4_4.0.2-2_amd64.deb
```

4 Installation de Google Authenticator sur le smartphone

L'étape suivante consiste à installer l'application Google Authenticator sur votre téléphone mobile ou votre tablette



5 Recuperation de la cle d'installation

09:59

4G



Configurer votre premier compte

Utilisez le code QR ou la clé de configuration dans les paramètres d'authentification à deux facteurs (de Google ou d'un service tiers). En cas de problème, consultez la page g.co/2sv



Scanner un code QR



Saisir une clé de configuration

[Importer des comptes existants ?](#)

Maintenant, exécutez l'application Google Authenticator sur l'appareil mobile. Cliquez sur le bouton *Démarrer* et vous verrez deux façons d'ajouter un appareil :

- Scanner un code QR
- Entrez une clé d'installation

Dans notre cas nous allons utiliser la deuxième option

Dans la console Linux, exécutez le package Google Authenticator que vous avez installé précédemment sur votre serveur Linux

```
google-authenticator -s ~/.ssh/google_authenticator
```

Cette commande exécutera le module PAM Google Authenticator.

L'option -s nous a permis d'enregistrer la clé secrète dans un emplacement non standard, et nous avons spécifié le répertoire ~/.ssh où les clés SSH sont conservées. Lorsque vous exécutez cette commande, vous verrez un code QR à l'écran si les bibliothèques de codes QR sont prises en charge. Ouvrez maintenant l'application Google Authenticator sur votre mobile, appuyez sur l'option Scanner un code QR et scannez le *code QR* affiché sur la console du serveur Linux.



Répondez à toute les questions par y

```
Your new secret key is: YWUV7A3C5ZS36J7LSKIU4506LQ
Your verification code is 205912
Your emergency scratch codes are:
45535235
59682043
64399787
76059057
31588138

? Do you want me to update your "/home/abdoul/.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
abdoul@abdoul-virtual-machine:~$
```

Si votre serveur Linux ne prend pas en charge les codes QR ou si votre appareil mobile ne peut pas scanner le code QR, vous pouvez utiliser l'option de *clé secrète*. Pour ce faire, appuyez sur l'option *Saisir une clé de configuration* dans l'application Google Authenticator et utilisez la clé secrète affichée sur la console Linux pour ajouter le compte dans l'application Authenticator.

Une fois votre compte ajouté à l'application Authenticator, tapez le TOTP à partir de l'application Authenticator dans la console Linux et appuyez sur Entrée. Lorsque le code est confirmé, vous verrez les codes de travail d'urgence (ou codes de secours) affichés sur la console Linux. Stockez ces codes de sauvegarde dans un endroit sûr, car vous pourriez en avoir besoin pour récupérer votre accès SSH si quelque chose ne va pas avec votre téléphone à l'avenir.

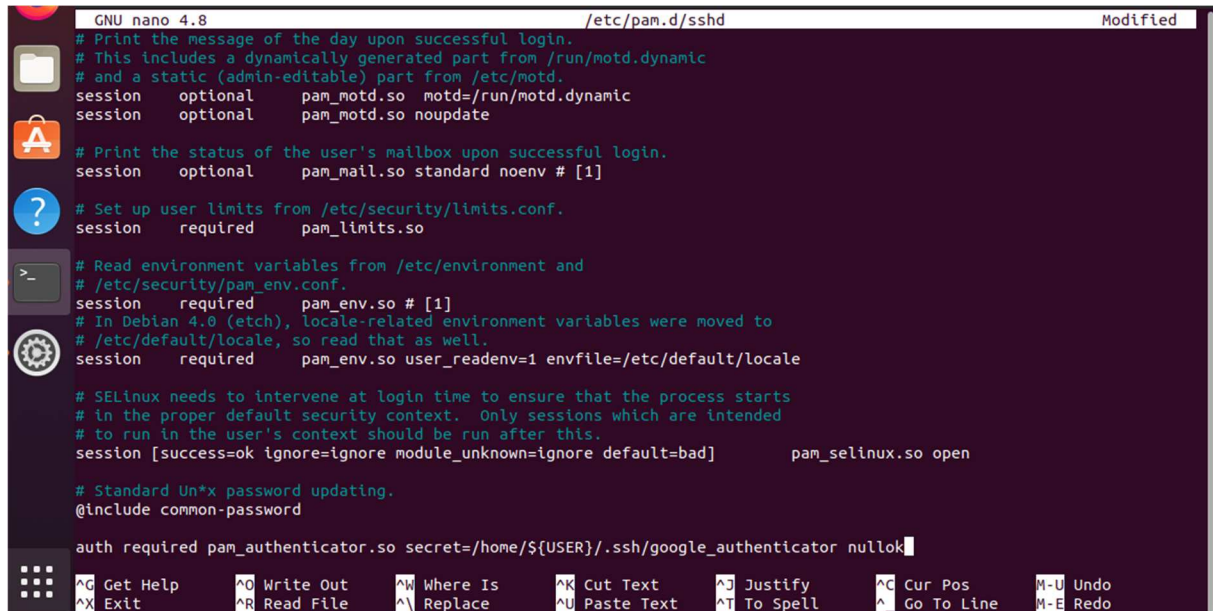
6 Configuration de l'authentification a deux (02) facteurs

L'étape suivante consiste à modifier le fichier de configuration PAM pour le démon SSH (*/etc/pam.d/sshd*). Exécutez la commande suivante pour ouvrir le fichier de configuration :

```
abdoul@abdoul-virtual-machine:~$ sudo nano /etc/pam.d/sshd
abdoul@abdoul-virtual-machine:~$
```


Faites défiler jusqu'à la fin du fichier et ajoutez la ligne de texte suivante

```
auth required pam_google_authenticator.so  
secret=/home/${USER}/.ssh/google_authenticator nullok
```



```
GNU nano 4.8 /etc/pam.d/ssh Modified  
# Print the message of the day upon successful login.  
# This includes a dynamically generated part from /run/motd.dynamic  
# and a static (admin-editable) part from /etc/motd.  
session optional pam_motd.so motd=/run/motd.dynamic  
session optional pam_motd.so noudate  
  
# Print the status of the user's mailbox upon successful login.  
session optional pam_mail.so standard noenv # [1]  
  
# Set up user limits from /etc/security/limits.conf.  
session required pam_limits.so  
  
# Read environment variables from /etc/environment and  
# /etc/security/pam_env.conf.  
session required pam_env.so # [1]  
# In Debian 4.0 (etch), locale-related environment variables were moved to  
# /etc/default/locale, so read that as well.  
session required pam_env.so user_readenv=1 envfile=/etc/default/locale  
  
# SELinux needs to intervene at login time to ensure that the process starts  
# in the proper default security context. Only sessions which are intended  
# to run in the user's context should be run after this.  
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open  
  
# Standard Unix password updating.  
@include common-password  
  
auth required pam_authenticator.so secret=/home/${USER}/.ssh/google_authenticator nullok
```

Notez comment le chemin d'accès de la clé secrète `google_authenticator` est fourni à l'aide de `secret=/home/${USER}/.ssh/google_authenticator`. Si vous stockez votre fichier de clé secrète dans un emplacement différent, assurez-vous de fournir le chemin correct ici. Notez également que l'option **nullok** permettra aux utilisateurs de se connecter avec uniquement le nom d'utilisateur et le mot de passe jusqu'à ce qu'ils terminent la configuration 2FA. Une fois la configuration 2FA terminée avec succès pour tous les utilisateurs, il est judicieux de supprimer l'option **nullok** pour imposer l'utilisation de 2FA à tout le monde. Si vous êtes le seul utilisateur sur ce système, vous pouvez ignorer l'ajout de l'option **nullok**

Maintenant, ouvrez le fichier de configuration du démon SSH (`/etc/ssh/sshd_config`) à l'aide de la commande suivante :

```
abdoul@abdoul-virtual-machine:~$ sudo nano /etc/ssh/sshd_config  
abdoul@abdoul-virtual-machine:~$
```

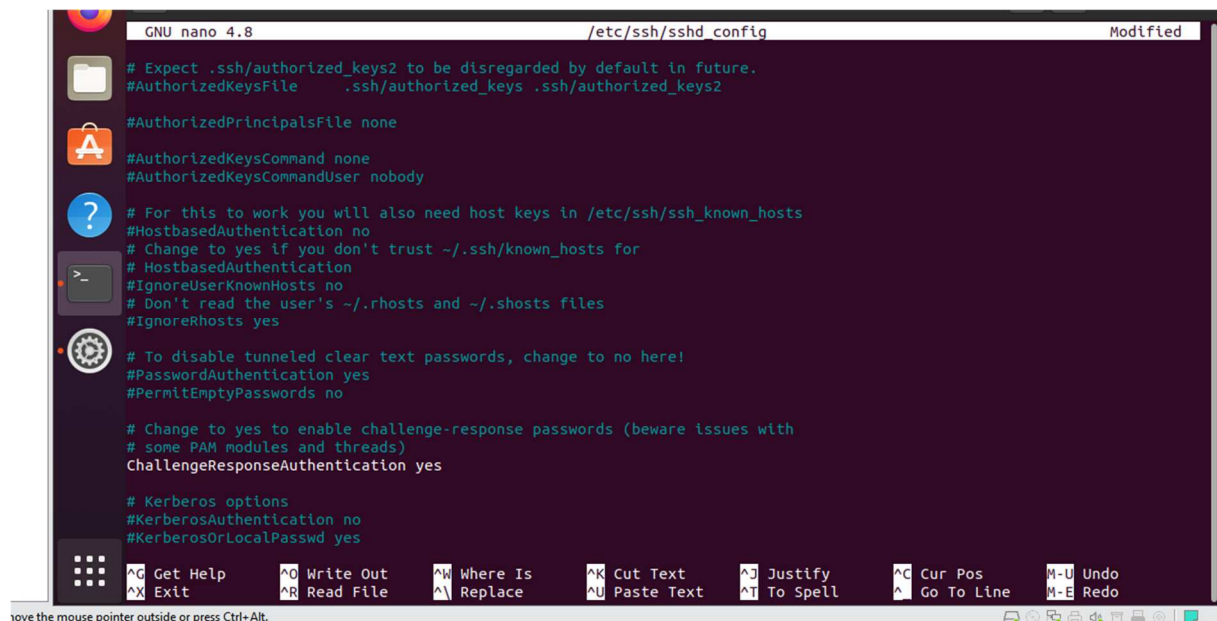
Si vous exécutez Ubuntu 22.04 ou version ultérieure, ajoutez la ligne suivante dans le fichier de configuration `sshd`:

KbdInteractiveAuthentication oui

Si vous utilisez Ubuntu 21.10 ou une version antérieure, ajoutez la ligne suivante à la place :

ChallengeResponseAuthentication oui

Dans les deux cas, si la ligne suggérée existe déjà dans le fichier de configuration `sshd`, assurez-vous de la décommenter et de la remplacer par « yes ». Ce faisant, vous activez essentiellement l'authentification interactive au clavier (ou réponse au défi) pour le démon SSH.



```
GNU nano 4.8 /etc/ssh/sshd_config Modified
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

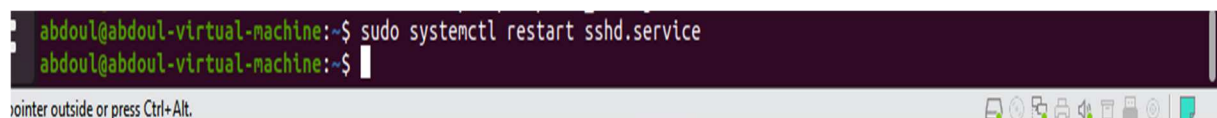
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes

Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  M-U Undo
Exit      Read File  Replace  Paste Text  To Spell  Go To Line M-E Redo
```

Enfin, redémarrez le démon SSH à l'aide de la commande suivante :

sudo systemctl restart sshd.service



```
abdoul@abdoul-virtual-machine:~$ sudo systemctl restart sshd.service
abdoul@abdoul-virtual-machine:~$
```

7 Vérification de l'authentification a deux (02) facteurs

À ce stade, votre système Ubuntu Linux est prêt avec l'authentification à deux facteurs. Pour vérifier si tout fonctionne, lancez un nouveau terminal et essayez d'accéder au système Ubuntu à l'aide de SSH. En supposant que vous ayez déjà une session SSH ou console ouverte sur votre système Ubuntu, vous pouvez exécuter la commande `sudo tail -f / var / log / auth.log` pour afficher les journaux d'accès SSH en même temps.

```
abdoul@abdoul-virtual-machine: ~/Desktop
abdoul@abdoul-virtual-machine:~/Desktop$ sudo tail -f /var/log/auth.log
[sudo] password for abdoul:
Feb 9 00:14:14 abdoul-virtual-machine polkitd(authority=local): Registered Authentication Agent for unix-session:2 (system bus name :1.79 [/usr/bin/gnome-shell], object path /org/freedesktop/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8)
Feb 9 00:14:27 abdoul-virtual-machine gdm-launch-environment]: pam_unix(gdm-launch-environment:session): session closed for user gdm
Feb 9 00:14:27 abdoul-virtual-machine systemd-logind[694]: Session c1 logged out. Waiting for processes to exit.
Feb 9 00:14:27 abdoul-virtual-machine systemd-logind[694]: Removed session c1.
Feb 9 00:14:28 abdoul-virtual-machine polkitd(authority=local): Unregistered Authentication Agent for unix-session:c1 (system bus name :1.42, object path /org/freedesktop/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8) (disconnected from bus)
Feb 9 00:14:29 abdoul-virtual-machine dbus-daemon[674]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)
Feb 9 00:17:01 abdoul-virtual-machine CRON[1938]: pam_unix(cron:session): session opened for user root by (uid=0)
Feb 9 00:17:01 abdoul-virtual-machine CRON[1938]: pam_unix(cron:session): session closed for user root
Feb 9 00:18:05 abdoul-virtual-machine sudo:    abdoul : TTY=pts/0 ; PWD=/home/abdoul/Desktop ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
Feb 9 00:18:05 abdoul-virtual-machine sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
```

À côté, ouvrez une nouvelle session SSH sur le système Ubuntu et vous remarquerez qu'après avoir entré votre nom d'utilisateur et votre mot de passe, vous serez maintenant invité à entrer un *code de vérification*

Vous devez utiliser le code de validation généré dans l'application Google Authenticator sur votre appareil mobile pour une authentification réussie. Si vous entrez le mauvais code de vérification, il vous demandera à nouveau un mot de passe.

Lorsque le code de vérification correct est entré, l'authentification réussit et la session SSH est ouverte.

```
abdou@abdou-virtual-machine: ~  
do you want to enable full connectivity? (y/n) y  
abdou@abdou-virtual-machine:~/Desktop$ sudo ssh abdou@127.0.0.1  
Password:  
Verification code:  
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-60-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Introducing Expanded Security Maintenance for Applications.  
  Receive updates to over 25,000 software packages with your  
  Ubuntu Pro subscription. Free for personal use.  
  
    https://ubuntu.com/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Your Hardware Enablement Stack (HWE) is supported until April 2025.
```

L'accès SSH notre système Ubuntu Linux est maintenant protégé avec deux facteurs d'authentification