

Compte rendu Examen Final (Guali Taha)

1- Création des comptes utilisateur et des groupes.

a) Création des admin de notre serveur

Création de l'utilisateur taha

L'utilisateur taha a été créé autant qu'administrateur d'où j'ai mis --ingroup admin.

Le mot de passe de taha est : 0123456789.

```
admin@ip-172-31-6-161:~$ sudo adduser --ingroup admin taha
Adding user `taha' ...
Adding new user `taha' (1001) with group `admin' ...
Creating home directory `/home/taha' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for taha
Enter the new value, or press ENTER for the default
  Full Name []: Guali Taha
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
```

Cette commande n'est pas suffisante puisqu'il faut usermod
(La commande usermod permet de modifier toutes les options
fixées par la commande adduser).

```
admin@ip-172-31-6-161:~$ sudo usermod -aG admin taha
admin@ip-172-31-6-161:~$ getent group admin
admin:x:1000:taha
admin@ip-172-31-6-161:~$ █
```

Une fois que usermod est exécutée je vérifie que mon utilisateur taha fait bien partie du groupe admin.

Finalement puisque l'utilisateur taha est admin alors il doit disposer des droits d'un sudoer (exécuter sous sudo).

```
admin@ip-172-31-6-161:~$ sudo usermod -aG sudo taha
admin@ip-172-31-6-161:~$ getent group sudo | cut -d: -f4
admin,taha
admin@ip-172-31-6-161:~$
```

Création de l'utilisateur Alexis

Nb : cette ligne je l'ai juste ajouté pour prévenir que créer un utilisateur avec un A majuscule serait ne pas respecter la convention de nommage j'aurai pu faire --force-badname pour résoudre le problème, mais j'ai préféré respecter la convention.

```
admin@ip-172-31-6-161:~$ sudo adduser --ingroup admin Alexis
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX configuration variable. Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
```

En gros :

Votre nom d'utilisateur sera : alexis.

Le mot de passe : eilco.

```
admin@ip-172-31-6-161:~$ sudo adduser --ingroup admin alexis
Adding user `alexis' ...
Adding new user `alexis' (1002) with group `admin' ...
Creating home directory `/home/alexis' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alexis
Enter the new value, or press ENTER for the default
    Full Name []: Chebrek Alexis
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
```

C'est bon une fois l'utilisateur alexis est créé on fait usermod pour qu'il soit réellement ajouté au group d'admin et on lui donne le droit de sudo.

```
admin@ip-172-31-6-161:~$ sudo usermod -aG admin alexis
admin@ip-172-31-6-161:~$ getent group admin
admin:x:1000:taha,alexis
admin@ip-172-31-6-161:~$ sudo usermod -aG sudo alexis
admin@ip-172-31-6-161:~$ getent group sudo
sudo:x:27:admin,taha,alexis
admin@ip-172-31-6-161:~$
```

Après cette étape tout semble bon.

b) Création des invités de notre serveur

Création de l'utilisateur invite

Pour le compte d'invité :

J'ai préféré lui créer un groupe (invite) comme ça il n'est pas affecté au groupe par défaut.

Une fois le groupe est créé, on crée l'utilisateur invite dans le groupe invite.

```
admin@ip-172-31-6-161:~$ sudo groupadd -g 1001 invite
admin@ip-172-31-6-161:~$ sudo adduser --ingroup invite invite
Adding user `invite' ...
Adding new user `invite' (1003) with group `invite' ...
Creating home directory `/home/invite' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for invite
Enter the new value, or press ENTER for the default
    Full Name []: invite
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
```

Après on fait usermod pour qu'il soit réellement.

Nb cette utilisateur dispose de quelques droits (connexion, etc...)

Mais pas de droit de faire sudo ou de modifier dans notre serveur.

```
admin@ip-172-31-6-161:~$ sudo usermod -aG invite invite
admin@ip-172-31-6-161:~$ getent group invite
invite:x:1001:invite
admin@ip-172-31-6-161:~$
```

2- Création des comptes utilisateur

Pour l'utilisateur taha

Je me connecte autant que taha.

Je crée un répertoire .ssh c'est ce répertoire qui va contenir mes clés publiques qui vont me permettre de me connecter au serveur.

Chmod 700 : - Lecture, écriture, exécution juste pour le propriétaire(taha). Pour les autres et le group ils ont aucun accès (r-w-x)

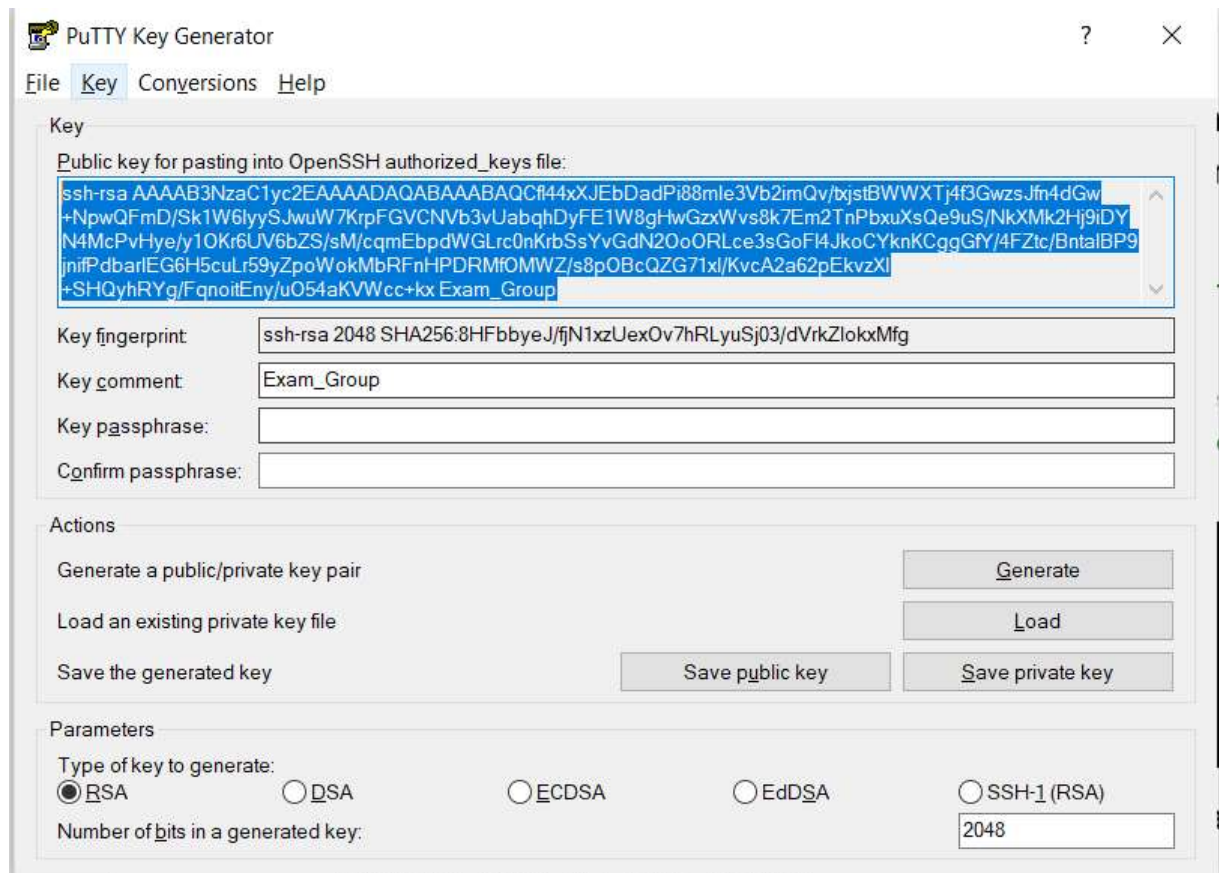
Chmod 600 : - Lecture, écriture, juste pour le propriétaire(taha). Pour les autres et le group ils ont aucun accès (r-w-x). Ici même taha n'as pas le droit d'exécuter.

```
admin@ip-172-31-6-161:~$ su -l taha
Password:
taha@ip-172-31-6-161:~$ mkdir .ssh
taha@ip-172-31-6-161:~$ chmod 700 .ssh
taha@ip-172-31-6-161:~$ touch .ssh/authorized_keys
taha@ip-172-31-6-161:~$ chmod 600 .ssh/authorized_keys
```

J'ouvre le fichier authorized_keys ou je vais stocker ma clé publique qui est Exam_Group.

```
taha@ip-172-31-6-161:~/ssh$ nano ~/.ssh/authorized_keys
```

Pour être sûr que j'ai copié la bonne clé je la charge dans puttygen et je prends la clé publique que je colle sur mon fichier `authorized_keys`.



Une fois la clé est collée sur mon fichier je peux vérifier le contenu de mon fichier à l'aide de la commande `cat`.

```
taha@ip-172-31-6-161:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACf144xXJEbDadPi88mle3Vb2imQv/txjstBWWXTj4f3GwzsJfn4dGw+NpwQFmD/Sk1W6lyySjWuW7KrpFGVCNVb3vUabqhDyFE1W8gHwGzxWvs8k7Em2TnPbxuXsQe9uS/NkXMk2Hj9iDYN4McPvHye/y1OKr6UV6bZS/sM/cqmEbpdWGLrc0nKrbSsYvGdN2OoORLce3sGoFl4JkoCYknKCggGfY/4FZtc/BntalBP9jnifPdbarIEG6H5cuLr59yZpoWokMbRfNHPDRMfOMWZ/s8pOBcQZG71xI/KvcA2a62pEkvzXl+SHQyhRYg/FqnoitEny/u054aKVWcc+kx Exam_Group
taha@ip-172-31-6-161:~$
```


Pour l'utilisateur alexis

Je me connecte autant que alexis.

Je crée un répertoire .ssh c'est ce répertoire qui va contenir mes clés publiques qui vont me permettre de me connecter au serveur.

Chmod 700 : - Lecture, écriture, exécution juste pour le propriétaire(taha). Pour les autres et le group ils ont aucun accès (r-w-x). cette commande concerne le répertoire .ssh .

Chmod 600 : - Lecture, écriture, juste pour le propriétaire(taha). Pour les autres et le group ils ont aucun accès (r-w-x). Ici même taha n'as pas le droit d'exécuter. Cette commande concerne le fichier authorized_keys.

J'ouvre le fichier authorized_keys ou je vais stocker ma clé publique qui est Exam_Group.

Une fois la clé est collée sur mon fichier je peux vérifier le contenu de mon fichier à l'aide de la commande cat.

```
taha@ip-172-31-6-161:~$ su -l alexis
Password:
alexis@ip-172-31-6-161:~$ mkdir .ssh
alexis@ip-172-31-6-161:~$ chmod 700 .ssh
alexis@ip-172-31-6-161:~$ touch .ssh/authorized_keys
alexis@ip-172-31-6-161:~$ nano ~/.ssh/authorized_keys
alexis@ip-172-31-6-161:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACfl44xXJEBDadPi88mle3Vb2imQv/txjstBWWXTj4f3GwzsJfn4dGw+
NpwQFmD/SklW6lYySJWuW7KrpFGVCNVb3vUabqhDyFE1W8gHwGzxWvs8k7Em2TnPbxuXsQe9uS/NkXMk2Hj9iDYN4McPv
Hye/y1OKr6UV6b2S/sM/cqmEbpdWGLrc0nKrbSsYvGdN2OoORLce3sGoFl4JkoCYknKCggGfY/4FZtc/BntalBP9jnifP
dbarIEG6H5cuLr59yZpoWokMbRfNHPDRMfOMWZ/s8pOBcQZG71xI/KvcA2a62pEkvzXl+SHQyhRYg/FqnoitEny/uO54a
KvWcc+kx Exam_Group
alexis@ip-172-31-6-161:~$
```

Pour être sûr que j'ai copié la bonne clé je la charge dans puttygen et je prends la clé publique que je colle sur mon fichier authorized_keys.

PutTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACf44xXJEbDadPi88mle3Vb2imQv/bxstBWWXTj4f3GwzsJfn4dGw
+NpwQFmD/Sk1W6llyS.JwuW7KrpFGVCNVb3vUabqhDyFE1W8gHwGzxWvs8k7Em2TnPbxuXsQe9uS/NkXMk2Hj9iDY
N4McPvHye/y1OKr6UV6bZS/sM/cqmEbpdWGLrc0nKrbSsYvGdN2OoORLce3sGoFI4JkoCYknKCggGY/4FZtc/BntalBP9
jnfPdbarlEG6H5culr59yZpoWokMbRFnHPDRMfOMWZ/s8pOBcQZG71xl/KvcA2a62pEkvzXl
+SHQyhRYg/FqnoitEny/uO54aKvWcc+kx Exam_Group
```

Key fingerprint: ssh-rsa 2048 SHA256:8HFbbyeJ/fjN1xzUexOv7hRLyuSj03/dVrkZllokxMfg

Key comment: Exam_Group

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

Type of key to generate:

☒ RSA ☐ DSA ☐ ECDSA ☐ EdDSA ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

Test de connexion

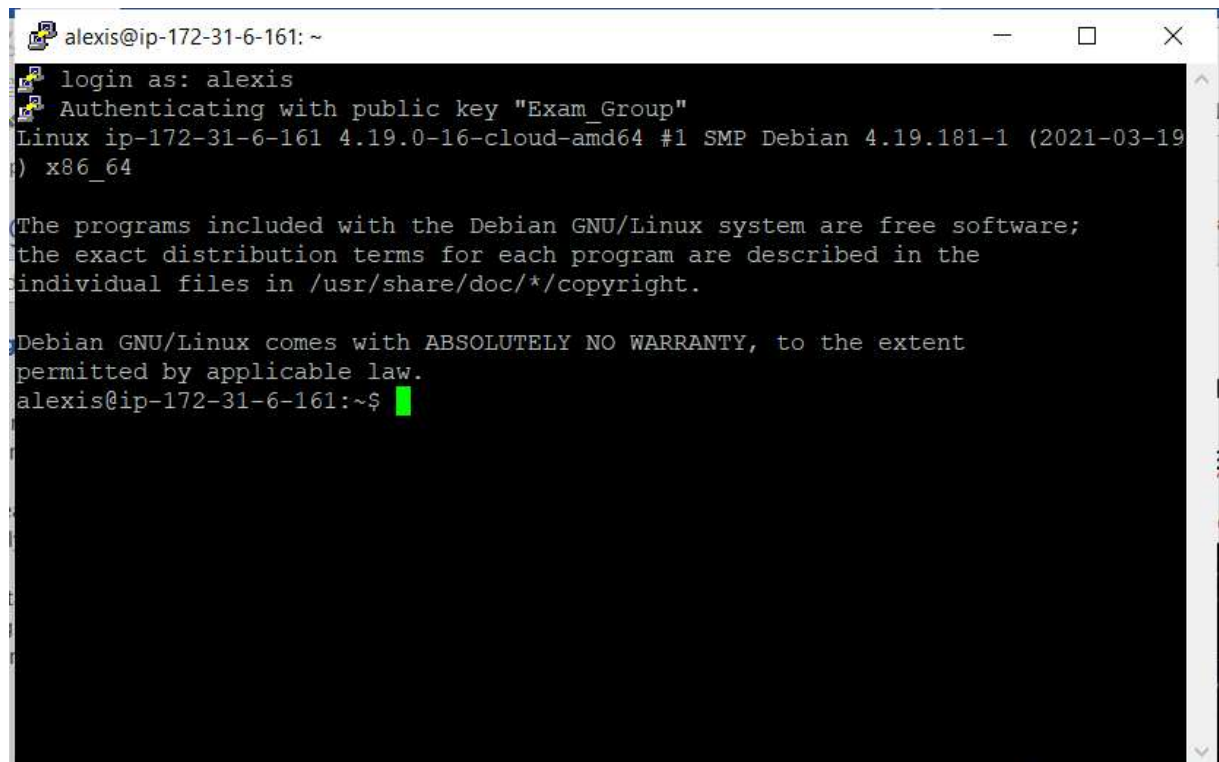
Pour taha :

```
taha@ip-172-31-6-161: ~
login as: taha
Authenticating with public key "Exam_Group"
Linux ip-172-31-6-161 4.19.0-16-cloud-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) 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.
taha@ip-172-31-6-161:~$
```

Pour alexis :



```
alexis@ip-172-31-6-161: ~  
login as: alexis  
Authenticating with public key "Exam_Group"  
Linux ip-172-31-6-161 4.19.0-16-cloud-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19)  
) 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.  
alexis@ip-172-31-6-161:~$
```


3- Connexion via SSH mot de passe et clé.

Après avoir ajouté une clé ssh pour tous les utilisateurs il faut configurer notre fichier sshd_config afin que l'utilisateur ne se connecte qu'après avoir chargé la bonne clé ssh et introduit le bon mot de passe.

C'est une restriction de connexion il doit obligatoire avoir les deux pour pouvoir accéder à notre serveur.

Pour configurer le fichier sshd_config il faut qu'un des utilisateurs sudo le fasse :

Rappel des users sudo :

```
admin@ip-172-31-6-161:~$ sudo usermod -aG sudo alexis
admin@ip-172-31-6-161:~$ getent group sudo
sudo:x:27:admin,taha,alexis
```

Il faut ajouter cette dans le fichier sshd_config : **AuthenticationMethods "publickey,password"**.

Cette ligne signifie que les deux méthodes d'authentification sont

- 1-avoir une public key qui est valable et une fois que la clé est valide.
- 2- l'utilisateur est obligé d'introduire son mot de passe sinon il se verra refuser l'accès à notre serveur.

Comme ça on sécurise l'accès à notre serveur par une double authentification.

Cette option n'est pas suffisante il faut le paramètre **PubKeyAuthentication** soit activé : comme ça notre serveur requiert la clé publique.

Finalement le paramètre **PasswordAuthentication** doit activer pour demander le mot de passe.

Cette configuration est illustrée dans l'images en bas.

```
GNU nano 3.2 /etc/ssh/sshd_config

#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
AuthenticationMethods "publickey,password"
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# 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

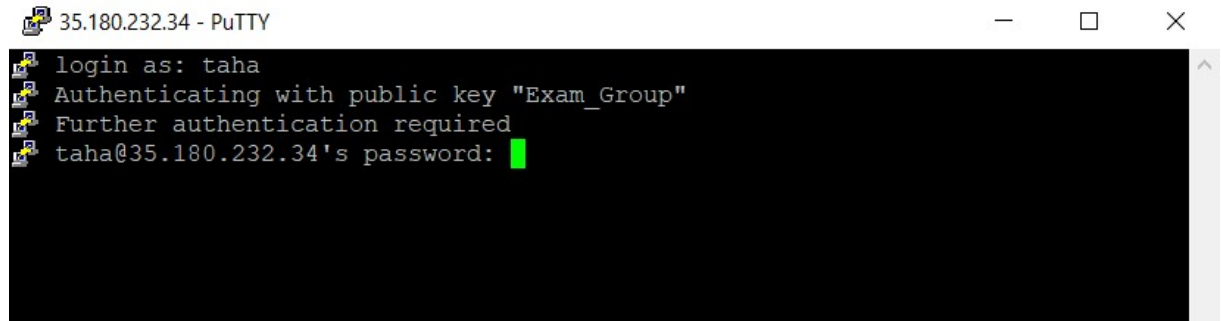
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
```

Une fois que tout est bien configuré il suffit de sauvegarder le fichier et de relancer le serveur SSH sinon les changements ne prendront pas effet.

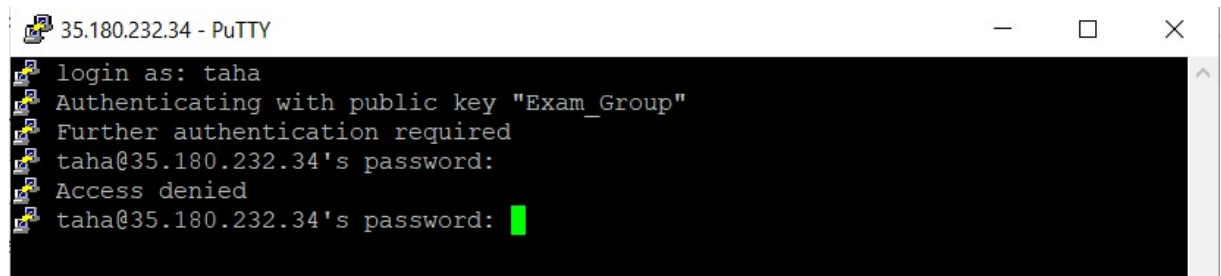
Test d'authentification

Une fois qu'on essaye de lancer le serveur dans cas la clé est bonne alors il demande le mot de passe pour se connecter.

A terminal window titled "35.180.232.34 - PuTTY" showing a successful login process. The user 'taha' enters their password, and the system authenticates using the public key "Exam_Group".

```
login as: taha
Authenticating with public key "Exam_Group"
Further authentication required
taha@35.180.232.34's password: 
```

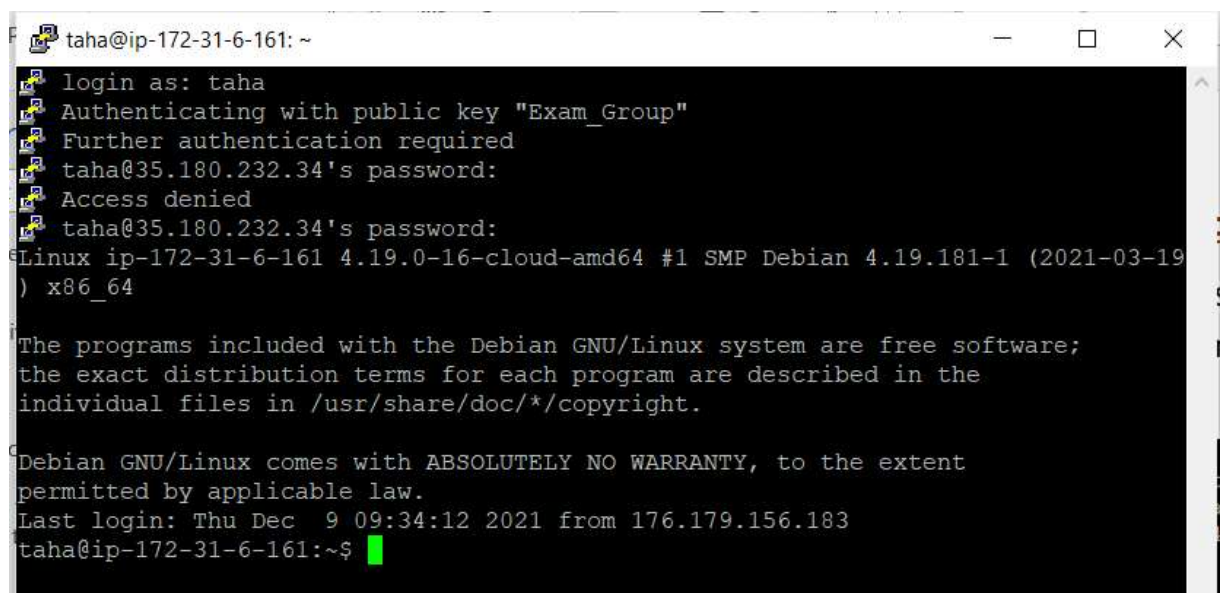
Si le mot de passe est incorrect on se voit refuser l'accès.

A terminal window titled "35.180.232.34 - PuTTY" showing an unsuccessful login attempt. After entering the password, the system responds with "Access denied".

```
login as: taha
Authenticating with public key "Exam_Group"
Further authentication required
taha@35.180.232.34's password:
Access denied
taha@35.180.232.34's password: 
```

Si le mot de passe et la clé sont correctes :

On peut accéder au serveur.

A terminal window titled "taha@ip-172-31-6-161: ~" showing a successful login process. The user 'taha' enters their password, and the system authenticates using the public key "Exam_Group". The terminal then displays the Linux boot information and the Debian GNU/Linux system's warranty disclaimer.

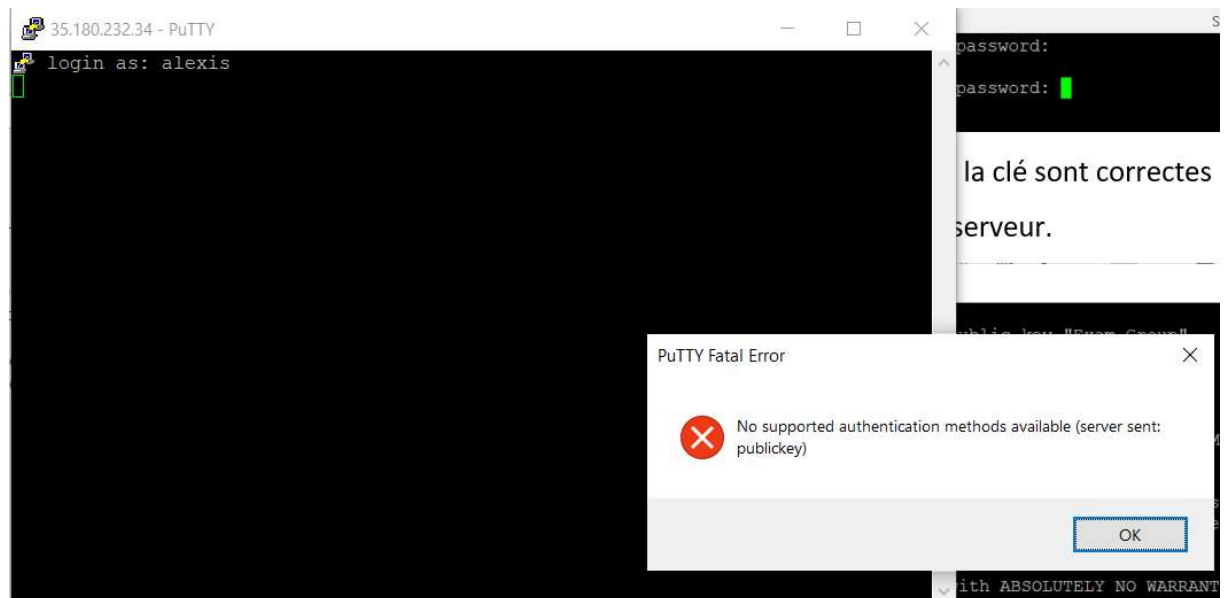
```
login as: taha
Authenticating with public key "Exam_Group"
Further authentication required
taha@35.180.232.34's password:
Access denied
taha@35.180.232.34's password:
Linux ip-172-31-6-161 4.19.0-16-cloud-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19)
) x86_64

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.
Last login: Thu Dec  9 09:34:12 2021 from 176.179.156.183
taha@ip-172-31-6-161:~$ 
```

Si la clé n'est pas correcte :

On se voit refuser l'accès au serveur.



4- Fail2ban.

Pour installer Fail2ban Il suffit de lancer cette commande

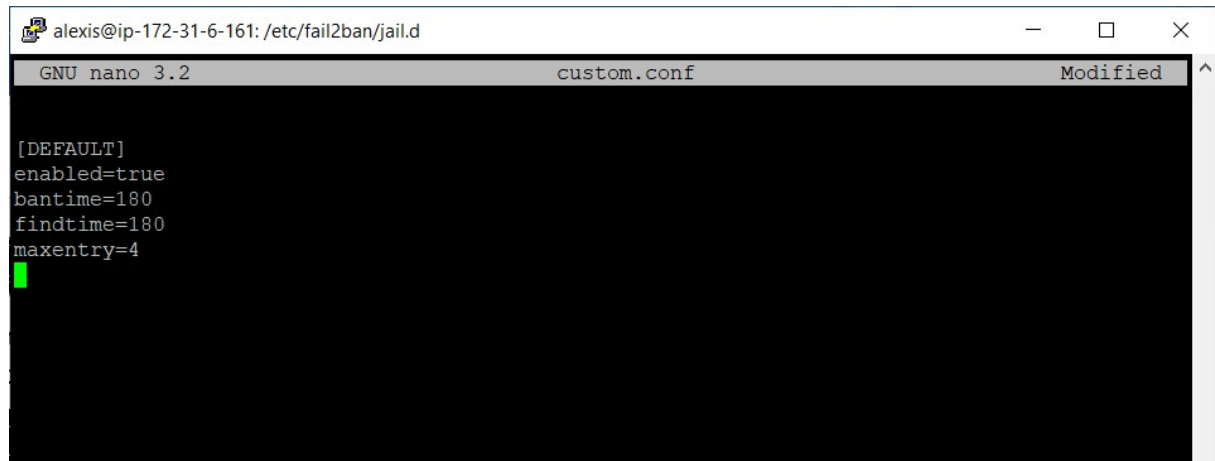
```
alexis@ip-172-31-6-161:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify python3-systemd whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify python3-systemd whois
0 upgraded, 4 newly installed, 0 to remove and 47 not upgraded.
Need to get 527 kB of archives.
After this operation, 2560 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://cdn-aws.deb.debian.org/debian buster/main amd64 fail2ban all 0.10.2-2.1 [385 kB]
Get:2 http://cdn-aws.deb.debian.org/debian buster/main amd64 python3-pyinotify all 0.9.6-1 [26.9 kB]
Get:3 http://cdn-aws.deb.debian.org/debian buster/main amd64 python3-systemd amd64 234-2+b1 [37.2 kB]
Get:4 http://cdn-aws.deb.debian.org/debian buster/main amd64 whois amd64 5.4.3 [77.8 kB]
Fetched 527 kB in 0s (16.1 MB/s)
```

Une fois fail2ban installé il faut démarrer fail2ban avec la commande suivante :

```
alexis@ip-172-31-6-161:~$ sudo service fail2ban start
alexis@ip-172-31-6-161:~$
```

Après on configure un fichier de configuration annexe qui est custom.conf

```
alexis@ip-172-31-6-161:~$ sudo service fail2ban start
alexis@ip-172-31-6-161:~$ cd /etc/fail2ban
alexis@ip-172-31-6-161:/etc/fail2ban$ cd jail.d
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo nano custom.conf
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo nano custom.conf
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo nano custom.conf
```



```
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d
GNU nano 3.2 custom.conf Modified
[DEFAULT]
enabled=true
bantime=180
findtime=180
maxentry=4
```

Alors et je paramètre le temps de ban à 180 s qui est de 3 min (je pouvais aussi faire que 3m), l'utilisateur sera banni après 4 essais comme maximum possible.

Pour le findtime qui définit en secondes le temps depuis lequel une anomalie est recherchée dans les logs. Je l'ai paramétré a 180 s.

Là on remarque que fail2ban est active :

```
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-12-09 09:55:47 UTC; 12min ago
     Docs: man:fail2ban(1)
  Main PID: 9868 (fail2ban-server)
    Tasks: 3 (limit: 557)
   Memory: 13.6M
    CGroup: /system.slice/fail2ban.service
            └─9868 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Dec 09 09:55:47 ip-172-31-6-161 systemd[1]: Starting Fail2Ban Service...
Dec 09 09:55:47 ip-172-31-6-161 systemd[1]: Started Fail2Ban Service.
Dec 09 09:55:49 ip-172-31-6-161 fail2ban-server[9868]: Server ready
Dec 09 09:55:49 ip-172-31-6-161 systemd[1]: /lib/systemd/system/fail2ban.service:12: PIDFile=
lines 1-14/14 (END)
```


Pour le démarrage automatique faudrait faire :

```
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sy
sv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$
```

5- Installation et configuration du serveur web

Pour installer notre serveur web apache il est obligatoire de vérifier est notre serveur doit être à jour :

```
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo apt-get update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://cdn-aws.deb.debian.org/debian buster InRelease
Get:3 http://cdn-aws.deb.debian.org/debian buster-updates InRelease [51.9 kB]
Get:4 http://cdn-aws.deb.debian.org/debian buster-backports InRelease [46.7 kB]
Fetched 98.6 kB in 0s (256 kB/s)
Reading package lists... Done
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo apt-get upgrade
```

Installation du serveur apache :

```
alexis@ip-172-31-6-161:/etc/fail2ban/jail.d$ sudo apt-get install apache2
```

Vérifier le statut du serveur apache :

```
alexis@ip-172-31-6-161:~$ sudo /etc/init.d/apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-12-09 10:15:15 UTC; 17s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 15744 (apache2)
    Tasks: 55 (limit: 557)
   Memory: 4.9M
   CGroup: /system.slice/apache2.service
           └─15744 /usr/sbin/apache2 -k start
             └─15746 /usr/sbin/apache2 -k start
               └─15747 /usr/sbin/apache2 -k start

Dec 09 10:15:15 ip-172-31-6-161 systemd[1]: Starting The Apache HTTP Server...
Dec 09 10:15:15 ip-172-31-6-161 systemd[1]: Started The Apache HTTP Server.
alexis@ip-172-31-6-161:~$
```

Configurer le serveur apache :

Dans ce cas moi j'ai changé le fichier par défaut /var/www à /home/invite/ car je vais créer une répertoire www qui contiendra mon fichier html qui sera héberger sur mon serveur par la suite.

```
<Directory /home/invite/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Nb il faut que ce fichier soit accessible c'est pour cela que je l'ai créé chez le compte invité car taha, alexis c'est les seuls qui peuvent lire le contenu de leurs fichiers d'où ça va créer un problème.

```
invite@ip-172-31-6-161:~$ mkdir www
invite@ip-172-31-6-161:~$ cd www/
invite@ip-172-31-6-161:~/www$ nano index.html
```

Le contenu du fichier html qui sera affiché ressemble à ça :

```
invite@ip-172-31-6-161: ~/www
GNU nano 3.2 index
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>

  <body>
    <h1> Je vais valider la matière de réseau avec une bonne note c'est sur</h1>
  </body>
</html>
```

La partie finale c'est de dire au fichier config par défaut des sites actifs que la source ou je vais récupérer tous mes fichiers c'est /home/invite/www ou j'ai créé mon fichier html.

```
GNU nano 3.2 /etc/apache2/sites-enabled/000-default.conf
VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /home/invite/www

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Je redémarre mon serveur et que je vérifie que tout est bon

```
alexis@ip-172-31-6-161:~$ sudo /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
alexis@ip-172-31-6-161:~$ sudo /etc/init.d/apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-12-09 10:49:41 UTC; 24s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 17910 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 17914 (apache2)
    Tasks: 55 (limit: 557)
   Memory: 4.8M
   CGroup: /system.slice/apache2.service
           └─17914 /usr/sbin/apache2 -k start
             └─17917 /usr/sbin/apache2 -k start
               └─17918 /usr/sbin/apache2 -k start

Dec 09 10:49:41 ip-172-31-6-161 systemd[1]: Starting The Apache HTTP Server...
Dec 09 10:49:41 ip-172-31-6-161 systemd[1]: Started The Apache HTTP Server.
alexis@ip-172-31-6-161:~$
```

Pour qu'une fois je Ping sur mon adresse IP publique je vois ma page html se charger. Et comme ça j'ai pu héberger mon site sur mon serveur apache.

```
< > 35.180.232.34
Je vais valider la matière de réseau avec une bonne note c'est sur
```