TD DROITS D'ACCES

Question 1 :

```
root@Deb1:~# adduser john
Ajout de l'utilisateur « john » ...
Ajout du nouveau groupe « john » (1002) ...
Ajout du nouvel utilisateur « john » (1002) avec le groupe « john » (1002) ...
Création du répertoire personnel « /home/john » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Aucun mot de passe n'a été fourni.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour john
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
        NOM []:
        Numéro de chambre []:
        Téléphone professionnel []:
        Téléphone personnel []:
        Autre []:
Cette information est-elle correcte ? [O/n]
Ajout du nouvel utilisateur « john » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « john » au groupe « users » ...
root@Deb1:~#
```

```
root@Deb1:~# cd /home/john
root@Deb1:/home/john# _
```

```
root@Deb1:/home/john# touch test.txt
root@Deb1:/home/john# ls
test.txt
root@Deb1:/home/john# ls -l
total 0
-rw-r--r-- 1 root root 0 29 sept. 15:15 test.txt
root@Deb1:/home/john# chmod test.txt 640
chmod: mode incorrect : « test.txt »
Saisissez « chmod --help » pour plus d'informations
root@Deb1:/home/john# chmod 640 test.txt
root@Deb1:/home/john# ls -l
total 0
-rw-r----- 1 root root 0 29 sept. 15:15 test.txt
root@Deb1:/home/john#
```

John ne peut rien faire sur ce fichier car on a défini le droit des autres utilisateurs à 0 donc aucun et lecture/écriture pour l'utilisateur root et lecture pour le groupe root. A noter que les utilisateur et groupe propriétaire du fichier est root car c'est cet utilisateur qui a créé le fichier.

Question 2 :

```
  GNU nano 7.2                                                    .bashrc *
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
umask 001_
# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
```

ici dans « /home/john/.bashrc » « umask 001 »

Pour recharger le fichier « `root@Deb1:/home/john# source .bashrc` »

Pour tester «
```
root@Deb1:/home/john# umask
0001
```
»

test :

```
root@Deb1:/home/john# touch umask.txt
root@Deb1:/home/john# ls -l
total 0
-rw-r----- 1 root root 0 29 sept. 15:15 test.txt
-rw-rw-rw- 1 root root 0 29 sept. 15:25 umask.txt
```

On voit ici que le fichier créer a comme droit 666 donc john a comme droit lecture et ecriture

```
root@Deb1:/home/john# mkdir dossier_test
root@Deb1:/home/john/dossier_test# ls -ld
drwxrwxrw- 2 root root 4096 29 sept. 15:26 .
root@Deb1:/home/john/dossier_test#
```

Ici on voit que john a comme droit lecture et ecriture sur le dossier créer

Pour déterminer la valeur à utiliser pour la commande umask, vous devez soustraire la valeur des droits d'accès souhaités (au moyen de la valeur que vous définiriez pour la commande chmod) des droits d'accès par défaut en cours affectés aux fichiers.

Par exemple :

Pour les fichiers :

- Permissions par défaut : **666** (rw-rw-rw-)
- umask de **001** enlève **1** de la permission des "autres".
- **Résultat** : **666 - 001 = 666** (rw-rw-rw-)

  Donc ne change rien car droit par défaut 666

Pour les dossiers :

- Permissions par défaut : **777** (rwxrwxrwx)
- umask de **001** enlève **1** de la permission d'exécution pour "autres".
- **Résultat** : **777 - 001 = 776** (rwxrwxrw-)

Question 3 :

```
root@Deb1:/home/john# chmod +t ./
root@Deb1:/home/john# ls -ld
drwxrwxrwt 5 john john 4096 29 sept. 15:47 .
```

Ici on voit avec le T à la fin que le sticky bit est bien activé c'est-à-dire que seul le propriétaire du dossier ou root peut supprimer ou renommer un fichier dans ce dossier essayons :

```
theo@Deb1:/home/john$ ls
dossier_test  test  umask.txt
theo@Deb1:/home/john$ rm umask.txt
rm: impossible de supprimer 'umask.txt': Opération non permise
theo@Deb1:/home/john$
```

Malgré le fait que j'ai les permissions je ne peux pas supprimer le fichier.

Question 4 :

```
root@Deb1:/home/john# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68248 23 mars   2023 /usr/bin/passwd
root@Deb1:/home/john# ls -l /etc/shadow
-rw-r----- 1 root shadow 969 29 sept. 15:13 /etc/shadow
root@Deb1:/home/john#
```

Oui il est activé, car Le programme **passwd** doit modifier des fichiers sensibles comme **/etc/shadow** (qui contient les mots de passe des utilisateurs), et ces fichiers ne sont accessibles qu'avec des privilèges **root**. Si le **SUID (Correspond au s dans les droits de passwd)** n'était pas activé, un utilisateur non privilégié ne pourrait pas modifier son propre mot de passe, car il n'aurait pas accès à ces fichiers critiques.

Question 5 :

```
root@Deb1:/home/john# chmod g+s ./dossier_test/
root@Deb1:/home/john# cd dossier_test
root@Deb1:/home/john/dossier_test# ls -ld
drwxrwsrw- 2 john john 4096 29 sept. 15:26 .
root@Deb1:/home/john/dossier_test#
```

On voit ici que le S a été rajouter dans la partie groupe donc le SGID est en place, c'est-à-dire que tous les fichiers créer dans ce dossier même si c'est un autre utilisateur aura comme groupe propriétaire celui du répertoire dans lequel il a été créer :

```
root@Deb1:/home/john/dossier_test# touch sgid.txt
root@Deb1:/home/john/dossier_test# ls -ls
total 0
0 -rw-rw-rw- 1 root john 0 29 sept. 16:10 sgid.txt
root@Deb1:/home/john/dossier_test# _
```

On voit ici que le fichier créer a bien hérité de l'utilisateur qui l'as créé mais pas de son groupe car le SGID est activer donc il a hérité du groupe propriétaire du dossier.

Question 6 :

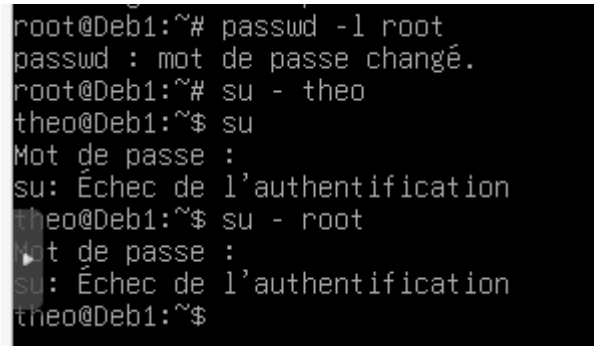
On peut faire cette commande « find / -perm -4000 -type f 2>/dev/null » :

Explication de la commande :

- **/** : Représente la recherche à partir de la racine du système de fichiers.
- **-perm -4000** : Cela filtre les fichiers qui ont le **bit SUID** activé (le chiffre 4000 correspond au bit SUID dans la notation octale).
- **-type f** : Limite la recherche aux fichiers (exclut les répertoires).
- **2>/dev/null** : Cette partie redirige les erreurs (comme celles provenant de répertoires auxquels vous n'avez pas accès) vers /dev/null, pour ne pas les afficher.

Question 7 :

Oui on peut le faire en faisant « passwd -l root » qui vas verrouiller le compte root, Lorsqu'un compte est verrouillé, cela rend impossible la connexion à ce compte, car le mot de passe est remplacé par une chaîne de caractères (généralement ! ou *) dans le fichier /etc/shadow (pour débloquer « passwd -u root ») :



Pour donner droit root à un autre compte il suffit de mettre ALL a l'utilisateur ops dans le fichier sudoers si le fichier n'est pas présent installé sudo "**apt install sudo**" :

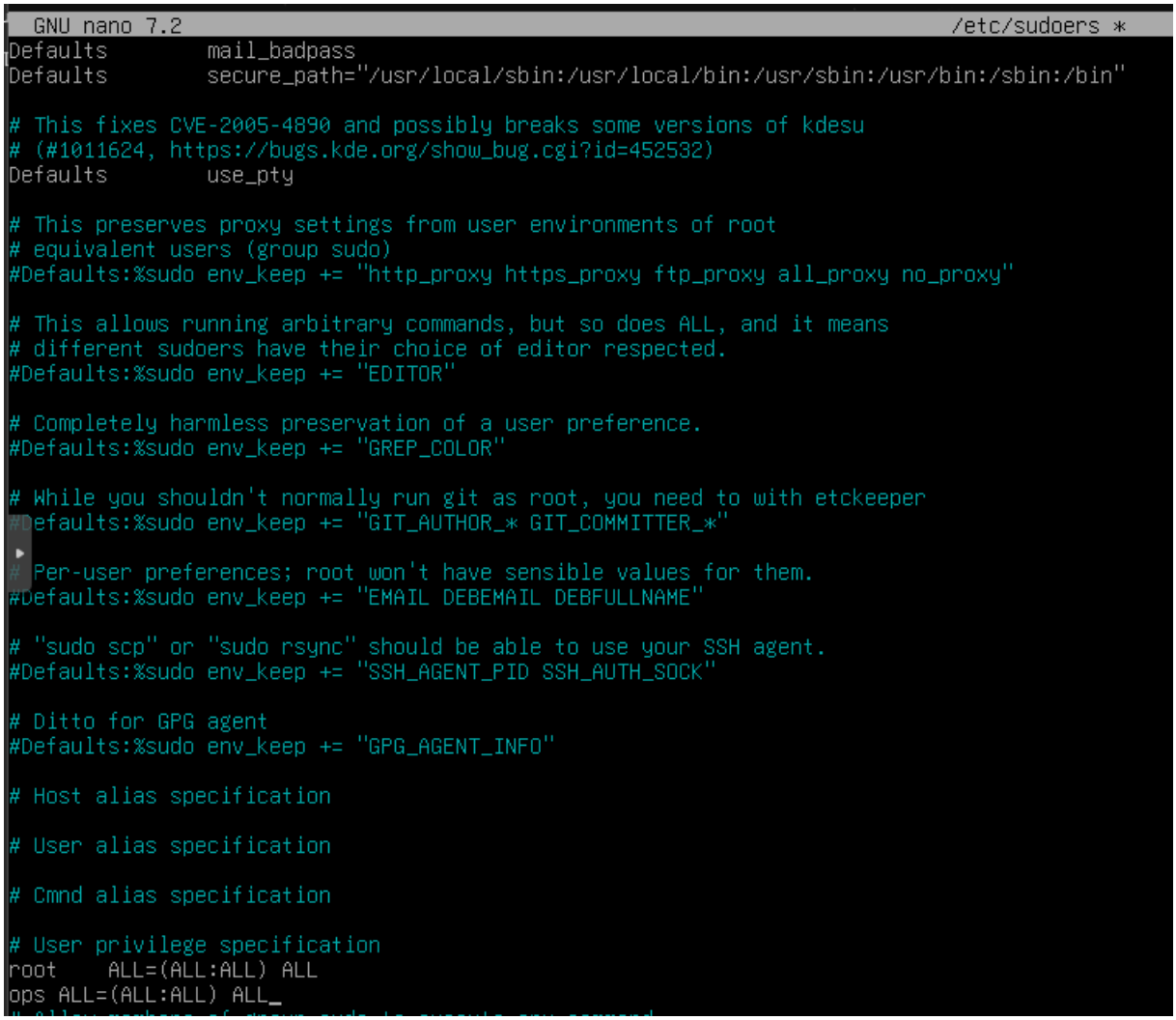
ALL (1er) :

Cette première mention de ALL signifie que l'utilisateur peut exécuter des commandes en tant qu'utilisateur root **sur n'importe quelle machine**.

(ALL:ALL) :

La première partie ALL avant le deux-points signifie que ops peut exécuter des commandes **en tant que n'importe quel utilisateur**. Cela inclut non seulement root, mais aussi d'autres utilisateurs s'il le souhaite.
La seconde partie ALL après le deux-points fait référence au groupe. Cela signifie que ops peut aussi exécuter des commandes en tant que n'importe quel groupe.

ALL (dernier) :
Cette dernière mention de ALL signifie que l'utilisateur ops peut exécuter **n'importe quelle commande** avec les droits définis ci-dessus. Cela accorde des droits d'accès illimités à toutes les commandes du système.



Questions 8 :

En faisant chmod 646 n'importe qui pourra écrire et lire dans se fichier donc Jim pourra lire et écrire dans le fichier, mais faire ceci n'est pas une bonne idée surtout pour la sécurité, la meilleure solution est d'utiliser les ACL (Pour installer "apt install ACL") :

```
root@DebianClient:~# setfacl -m u:jim:rw /home/john/text.txt
root@DebianClient:~# getfacl /home/john/text.txt
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/john/text.txt
# owner: john
# group: john
user::rw-
user:jim:rw-
group::r--
mask::rw-
other::r--
```

On voit bien sur la capture que jim a bien un droit de lecture et d'écriture qui ont été rajouté au fichier text.txt.

`setfacl` : C'est la commande utilisée pour configurer (set) les **Access Control Lists (ACL)** sur un fichier ou un répertoire.

`-m` : Cela signifie "modifier" (modify). Cette option permet de modifier ou d'ajouter une nouvelle entrée ACL.

`u:jim:rw` :

`u:jim` : Ce segment spécifie que l'entrée ACL est pour l'utilisateur (**u**) nommé **Jim**.

`rw` : Ce sont les permissions attribuées à **Jim**. Ici, `rw` signifie que **Jim** a le droit de lire (`r`) et d'écrire (`w`) dans le fichier **text.txt**.

`/home/john/text.txt` : C'est le chemin absolu du fichier sur lequel les droits ACL sont appliqués. Ici, il s'agit du fichier **text.txt** situé dans le répertoire personnel de **John**.

Questions 9 :

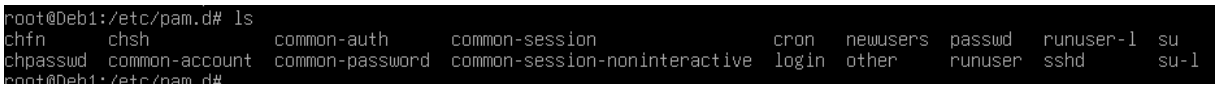
```
root@Deb1:~# usermod -s /bin/rbash jim
```

On peut changer le shell de l'utilisateur jim comme ceci "sudo usermod -s

/bin/rbash jim", de cette manière l'utilisateur ne pourra pas de déplacer et sera obligé de rester dans son répertoire.

Questions 10 :

PAM sert a demandé une authentification pour des divers services ou applications.



/etc/pam.d/common-auth :

Rôle : Définit les méthodes d'authentification pour tous les services utilisant PAM.

/etc/pam.d/common-account :

Rôle : Gère les politiques d'accès des utilisateurs. Il détermine si un utilisateur a le droit de se connecter ou d'utiliser le système.

/etc/pam.d/common-session :

Rôle : Définit les actions à réaliser lors de l'ouverture d'une session utilisateur.

/etc/pam.d/common-password :

Rôle : Gère les politiques de gestion des mots de passe, comme les exigences de complexité ou de changement de mot de passe.

/etc/security/access.conf :

Rôle : Utilisé par pam_access.so, il permet de contrôler l'accès au système en fonction de l'adresse IP, du nom d'utilisateur ou d'autres critères.

Exemples de règles : Autoriser ou interdire l'accès à certains utilisateurs à partir de certains hôtes.

/etc/pam.d/sshd : Configuration spécifique pour le service SSH.

/etc/pam.d/login : Configuration pour les connexions locales.

/etc/pam.d/su : Configuration pour la commande su, permettant de passer à un autre utilisateur.

Questions 11 :

Pour modifier la stratégie de mot de passe nous allons aller dans le fichier suivant :

```
/etc/pam.d/common-password
```

Et nous allons modifier la ligne suivante :

```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3
```

Et y rajouter :

```
minlen=3
```

Ce qui détermine le nombre de caractère minimum obligatoire pour le mot de passe *(3 ne sera pas du tout sécurisé dans un cadre strict mais dans le cadre du Td ça sera suffisant)*

Pour changer la date d'expiration du compte Linux nous allons utiliser la commande chage :
-E : permet de spécifier la date d'expiration
Nous avons donc défini l'expiration du compte le 10 janvier 2025 :

```
john@debianSRV:/$ sudo chage -E 2025-01-10 pinux
```

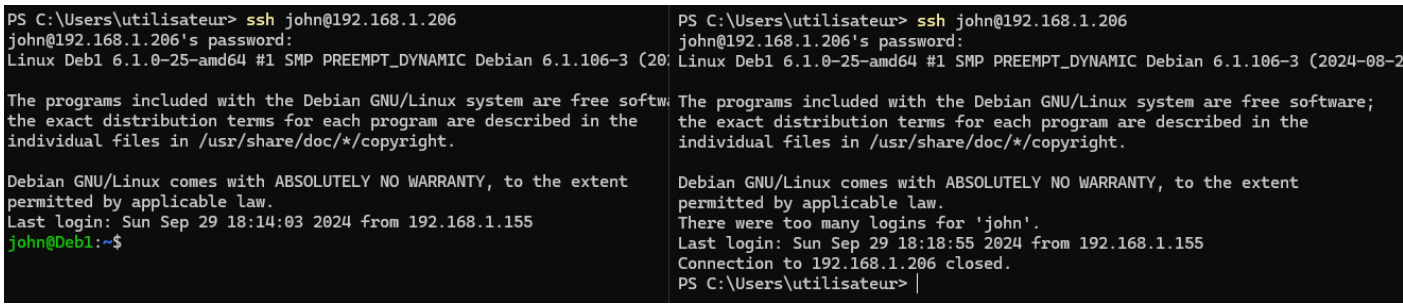
Questions 12 :

```
nano /etc/security/limits.conf_
```

Max logins pour tous les utilisateurs = 1

```
* hard maxlogins 1
```

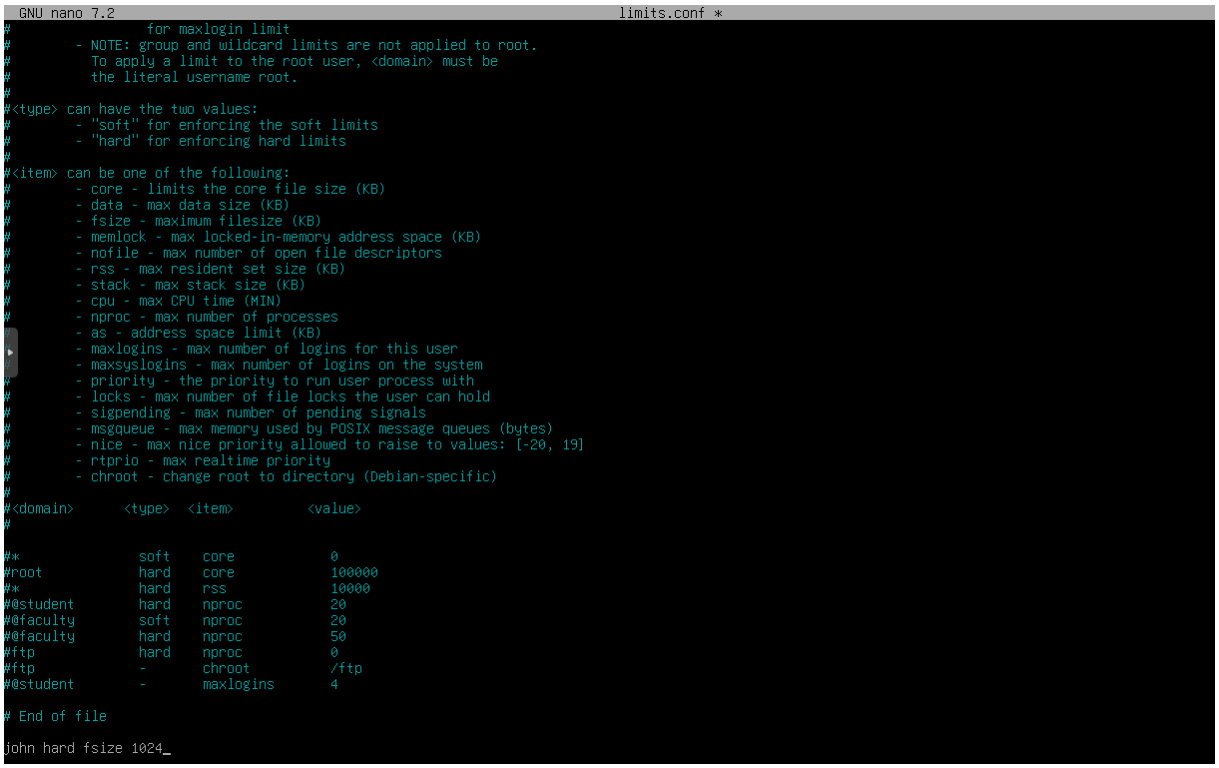
test :



On voit qu'on ne peut pas faire plus d'une connexion ssh, « there were too many logins for 'john' »

Questions 13 :

```
root@Deb1:/etc/security# nano limits.conf_
```



john hard fsize 1024

hard :

- Cela indique le type de limite. Il y a deux types de limites :
  - **hard** : Ce sont des limites imposées par le système qui ne peuvent pas être dépassées. L'utilisateur ne peut pas changer ces limites.
  - **soft** : Ce sont des limites qui peuvent être augmentées par l'utilisateur jusqu'à un certain maximum défini par le système.

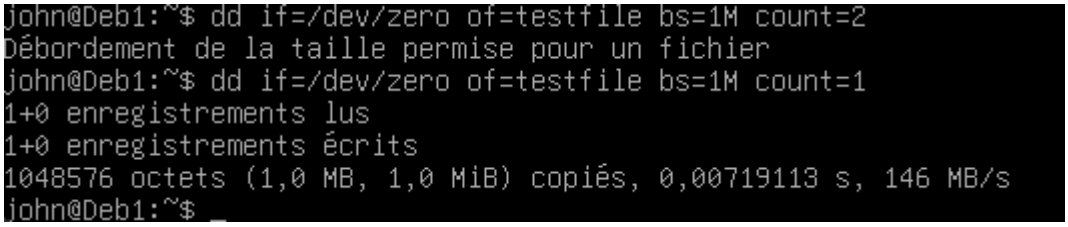
fsize :

- Cela fait référence à la taille maximale des fichiers que l'utilisateur peut créer ou modifier.

1024 :

- C'est la valeur de la limite, exprimée en blocs. Dans de nombreux systèmes Linux, un bloc est généralement de 1 Ko (1024 octets). Donc, 1024 ici signifie que l'utilisateur ne peut pas créer ou modifier des fichiers dont la taille totale dépasse **1 Mo** (1024 Ko).

Test :



dd :

- C'est une commande Unix/Linux utilisée pour copier et convertir des fichiers.

- « if=/dev/zero » :

- if signifie "input file" (fichier d'entrée). Dans ce cas, l'entrée provient de /dev/zero, un fichier spécial dans les systèmes Linux qui produit une série infinie de zéros (des octets de valeur 0). Cela signifie que le contenu qui sera écrit dans le fichier de sortie (of) sera constitué uniquement de zéros.

  - « of=testfile » :

  of signifie "output file" (fichier de sortie). Ici, le fichier de sortie sera nommé testfile. C'est le fichier dans lequel les données seront écrites.

  - « bs=1M » :

  bs signifie "block size" (taille de bloc). Cela définit la taille des blocs de données qui seront lus à partir de l'entrée et écrits dans la sortie. Dans ce cas, 1M signifie que chaque bloc a une taille de **1 Mo** (1 mégaoctet). Cela permet de remplir rapidement le fichier, car il écrit des blocs de 1 Mo à la fois.

  - « count=2 « :

  count spécifie le nombre de blocs à écrire dans le fichier de sortie. Ici, count=2 signifie que deux blocs de 1 Mo seront écrits, ce qui donnera un fichier de **2 Mo** au total.

Questions 14 :

```
root@Deb1:/etc/security# nano limits.conf_
```

```
john hard fsize 1024
john hard nproc 2
```

Test :

```
john@Deb1:~$ sleep 100 &
[1] 675
john@Deb1:~$ sleep 100 &
-bash: fork: retry: Ressource temporairement non disponible
-bash: fork: retry: Ressource temporairement non disponible
-bash: fork: retry: Ressource temporairement non disponible
^C-bash: fork: Appel système interrompu

john@Deb1:~$ sleep 100 &
-bash: fork: retry: Ressource temporairement non disponible
-bash: fork: retry: Ressource temporairement non disponible
^C-bash: fork: Appel système interrompu

john@Deb1:~$ _
```

La commande sleep interrompt l'exécution d'un processus pendant au moins l'intervalle spécifié par le paramètre Secondes.
On ne peut lancer que deux processus après on ne peut plus rien faire.

Questions 15 :

```
root@Deb1:~# nano /etc/pam.d/common-session_
```

Pour configurer le umask de manière persistante à chaque démarrage via PAM, vous pouvez éditer le fichier “/etc/pam.d/common-session”. Ajoutez la ligne suivante :

```
session required pam_umask.so umask=077
```

Test :

```
root@Deb1:~/test4# mkdir test
root@Deb1:~/test4# ls -l
total 4
drwx------ 2 root root 4096 29 sept. 18:49 test
```

Questions 16 :

Pour interdire à l'utilisateur test l'accès via SSH à un serveur spécifique, vous devez modifier le fichier “/etc/security/access.conf” et ajouter une règle PAM :

“- : test : ALL”

```
 GNU nano 7.2                                    /etc/security/access.conf *
# All lines from here up to the end are building a more complex example.
##############################################################################
#
# User "root" should be allowed to get access via cron .. tty5 tty6.
#+:root:cron crond :0 tty1 tty2 tty3 tty4 tty5 tty6
#
# User "root" should be allowed to get access from hosts with ip addresses.
#+:root:192.168.200.1 192.168.200.4 192.168.200.9
#+:root:127.0.0.1
#
# User "root" should get access from network 192.168.201.
# This term will be evaluated by string matching.
# comment: It might be better to use network/netmask instead.
#          The same is 192.168.201.0/24 or 192.168.201.0/255.255.255.0
#+:root:192.168.201.
#
# User "root" should be able to have access from domain.
# Uses string matching also.
#+:root:.foo.bar.org

User "root" should be denied to get access from all other sources.
-:root:ALL
#
# User "foo" and members of netgroup "nis_group" should be
# allowed to get access from all sources.
# This will only work if netgroup service is available.
#+:@nis_group foo:ALL
#
# User "john" should get access from ipv4 net/mask
#+:john:127.0.0.0/24
#
# User "john" should get access from ipv4 as ipv6 net/mask
#+:john:::ffff:127.0.0.0/127
#
# User "john" should get access from ipv6 host address
#+:john:2001:4ca0:0:101::1
#
# User "john" should get access from ipv6 host address (same as above)
#+:john:2001:4ca0:0:101:0:0:0:1
#
# User "john" should get access from ipv6 net/mask
#+:john:2001:4ca0:0:101::/64
#
# All other users should be denied to get access from all sources.
#-:ALL:ALL
-:test:ALL
```

Ensuite, assurez-vous que le module PAM pam_access.so est bien activé dans la configuration SSH ” account required pam_access.so” en modifiant ”/etc/pam.d/sshd” :

```
root@Deb1:~# nano /etc/pam.d/sshd
  GNU nano 7.2                                                  /etc/pam.d/sshd
# account  required     pam_access.so

# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close

# Set the loginuid process attribute.
session    required     pam_loginuid.so

# Create a new session keyring.
session    optional     pam_keyinit.so force revoke

# Standard Un*x session setup and teardown.
@include common-session

# 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 noupdate

# 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 Un*x password updating.
@include common-password
account required pam_access.so
```

il faut préciser a ssh qu’il faut utiliser pam_access.so dans sa politique d’authentification.

```
PS C:\Users\utilisateur> ssh test@192.168.1.206
test@192.168.1.206's password:
Connection closed by 192.168.1.206 port 22
```

Questions 17 :

```
root@Deb1:~# addgroup wheel
Ajout du groupe « wheel » (GID 1005).
Fait.
root@Deb1:~# usermod -aG wheel john
```

Pour restreindre l'utilisation de la commande su uniquement aux membres du groupe wheel, éditez le fichier ”/etc/pam.d/su” et ajoutez la ligne suivante ” auth required pam_wheel.so use_uid” :

```
root@Deb1:~# nano /etc/pam.d/su_
# This allows root to su without passwords (normal operation)
auth required pam_wheel.so use_uid_
auth       sufficient pam_rootok.so
```

Test :

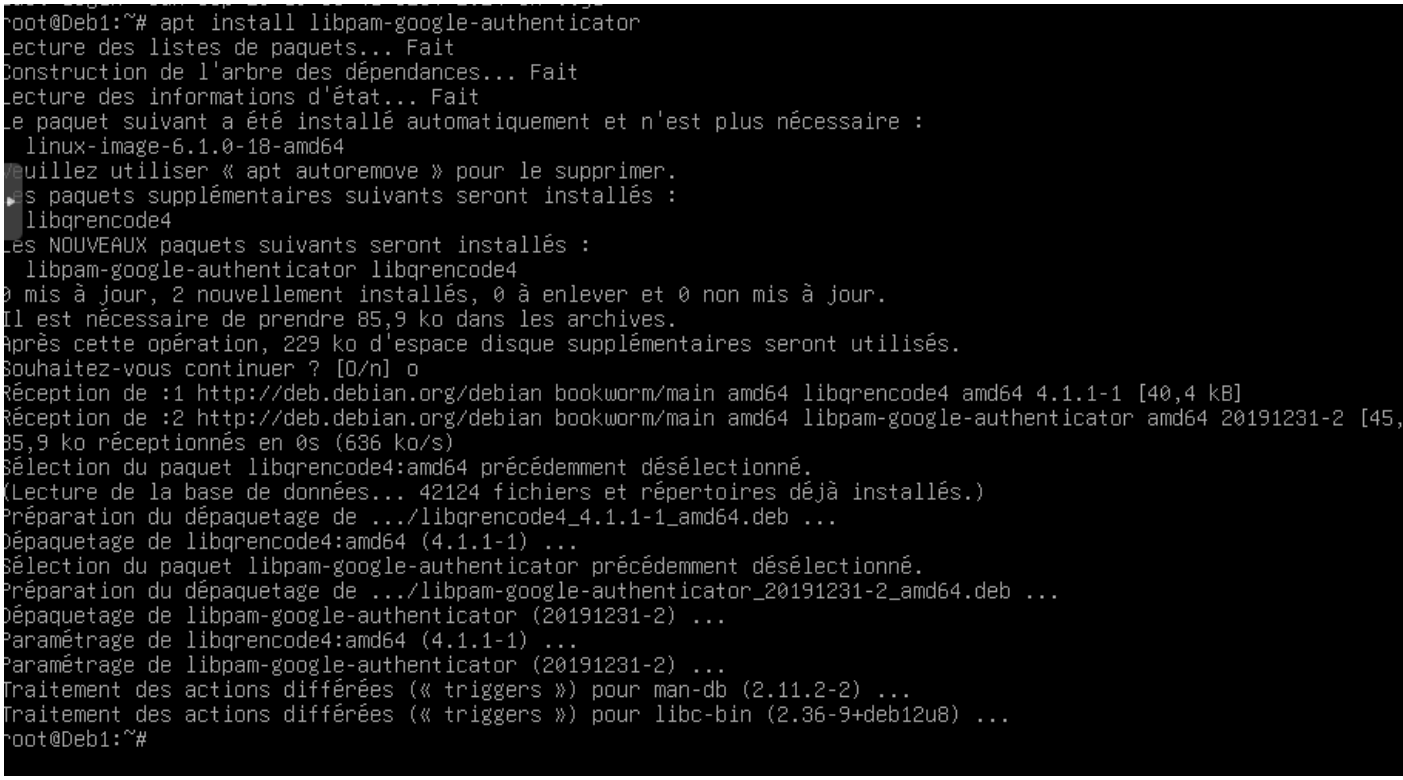
```
jim@Deb1:~$ su -
Mot de passe :
su: Autorisation refusée
john@Deb1:~$ su -
Mot de passe :
root@Deb1:~#
```

PARTIE 2 :

Pour implémenter l'authentification à deux facteurs (2FA) avec **Google Authenticator** sur Linux, voici les étapes à suivre :

Installation de Google Authenticator :

Commencez par installer le module PAM Google Authenticator sur votre machine Linux. Sur Debian, la commande est la suivante “apt install libpam-google-authenticator” :


root@Deb1:~# apt install libpam-google-authenticator
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
linux-image-6.1.0-18-amd64
Veuillez utiliser « apt autoremove » pour le supprimer.
Les paquets supplémentaires suivants seront installés :
libqrencode4
Les NOUVEAUX paquets suivants seront installés :
libpam-google-authenticator libqrencode4
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 85,9 ko dans les archives.
Après cette opération, 229 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 libqrencode4 amd64 4.1.1-1 [40,4 kB]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 libpam-google-authenticator amd64 20191231-2 [45,
85,9 ko réceptionnés en 0s (636 ko/s)
Sélection du paquet libqrencode4:amd64 précédemment désélectionné.
(Lecture de la base de données... 42124 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libqrencode4_4.1.1-1_amd64.deb ...
Dépaquetage de libqrencode4:amd64 (4.1.1-1) ...
Sélection du paquet libpam-google-authenticator précédemment désélectionné.
Préparation du dépaquetage de .../libpam-google-authenticator_20191231-2_amd64.deb ...
Dépaquetage de libpam-google-authenticator (20191231-2) ...
Paramétrage de libqrencode4:amd64 (4.1.1-1) ...
Paramétrage de libpam-google-authenticator (20191231-2) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u8) ...
root@Deb1:~#

Configuration de Google Authenticator pour un utilisateur :

Une fois installé, chaque utilisateur doit configurer son propre Google Authenticator. Connectez-vous avec l'utilisateur pour lequel vous voulez configurer la 2FA (par exemple, ops), puis exécutez “google-authenticator“ :

L'utilitaire vous posera plusieurs questions :

- **Token-based authentication** : Choisissez "y" pour permettre la génération d’un nouveau code toutes les 30 secondes.

- **Mise en place des restrictions pour éviter l’utilisation multiple d’un même code** : Choisissez "y".

- **Définir la fenêtre de tolérance de code pour éviter les désynchronisations** : Choisissez "n" pour les réglages par défaut.

- **Sauvegarder la clé secrète dans un fichier** : Choisissez "y" pour permettre la génération des codes en mode batch.

Le programme vous fournira un QR code que vous devrez scanner avec l'application Google Authenticator sur votre smartphone.


```
Your new secret key is: 2JTUUYYENEDIL7OG46R2BN3GWY
Enter code from app (-1 to skip): 239073
Code confirmed
Your emergency scratch codes are:
  70171495
  38553497
  30614167
  87532004
  46498737

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

Activer Google Authenticator via PAM

Pour forcer l'utilisation de la 2FA à la connexion, vous devez modifier la configuration PAM :

Éditez-le fichier "/etc/pam.d/sshd" "auth required pam_google_authenticator.so" :

```
root@Deb1:~# nano /etc/pam.d/sshd
```

```
auth required pam_google_authenticator.so
```

Configurer SSH pour utiliser la 2FA :

Éditez-le fichier “/etc/ssh/sshd_config” et modifiez ou ajoutez les lignes suivantes pour permettre la double authentification (mot de passe + Google Authenticator) :

```
root@Deb1:~# nano /etc/ssh/sshd_config
```

```
#AuthorizedPrincipalsFile none
ChallengeResponseAuthentication yes
#AuthorizedKeysCommand none
```

```
PermitRootLogin yes
```

Après avoir effectué les modifications, redémarrez le service SSH pour appliquer les changements :

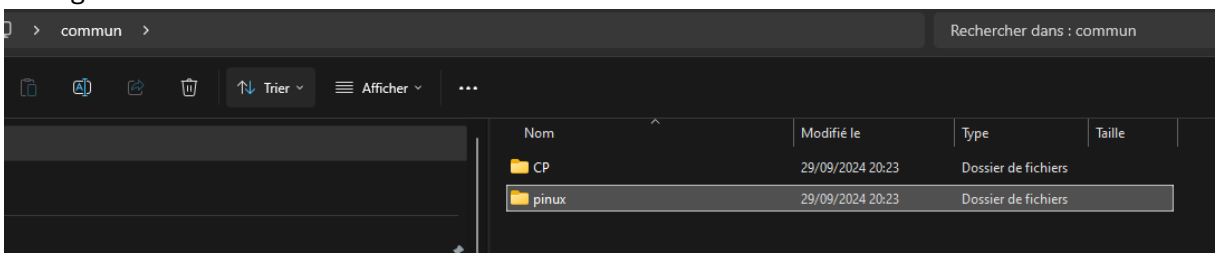
```
root@Deb1:~# systemctl restart sshd
```

Lorsque vous vous connectez via SSH ou à une session locale, vous serez d'abord invité à saisir votre mot de passe, puis à entrer le code généré par Google Authenticator.

```
PS C:\Users\utilisateur> ssh root@192.168.1.207
(root@192.168.1.207) Password:
(root@192.168.1.207) Verification code:
Linux Deb1 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) 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: Sun Sep 29 19:51:32 2024
root@Deb1:~#
```

Partie 3 :

Partage dossier windows :



Vous pouvez utiliser la commande suivante pour afficher les partages SMB disponibles sur une machine Windows :

```
root@Deb1:~# smbclient -L 192.168.1.155 -U utilisateur
Password for [WORKGROUP\utilisateur]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Administration à distance
        C$              Disk      Partage par défaut
        commun          Disk
        D$              Disk      Partage par défaut
        E$              Disk      Partage par défaut
        IPC$            IPC       IPC distant
        Users           Disk
SMB1 disabled -- no workgroup available
root@Deb1:~#
```

smbclient permet de voir (avec l'option -L) les partages SMB exposés par une machine windows :

```
root@Deb1:~# mount.cifs //192.168.1.155/commun /tmp/share -o username=utilisateur
Password for utilisateur@//192.168.1.155/commun:
root@Deb1:/tmp/share# df -h
//192.168.1.155/commun   931G    312G  619G  34% /tmp/share
root@Deb1:/tmp/share# ls
CP  pinux
```

netstat -a sur windows

```
TCP    192.168.1.155:445      Deb1:41164             ESTABLISHED
```

```
root@Deb1:/tmp/share# nano /etc/cle.txt
```

```
root@Deb1:/tmp/share# ccrypt -e -r -k /etc/cle.txt ./
```

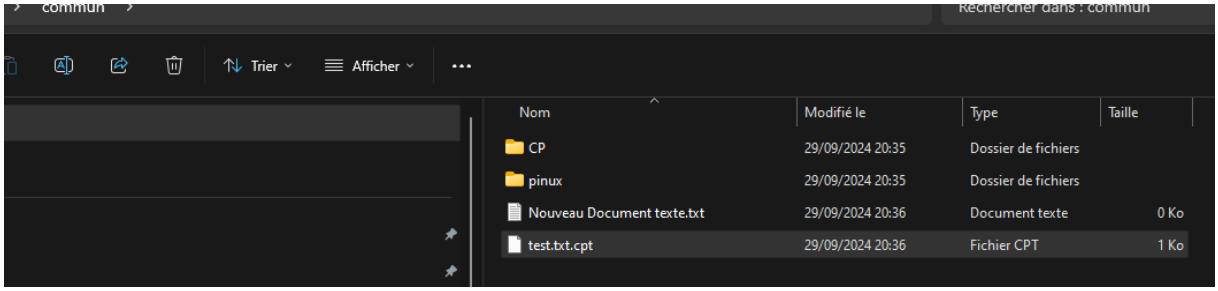
Explications des options :

- -e : Chiffrement des fichiers.
- -r : Récursivité (chiffre également les fichiers dans les sous-dossiers).
- -k cle.txt : Utilisation d'une clé stockée dans le fichier cle.txt.

```
mount.cifs //192.168.1.155/commun /tmp/share -o username=bob,rw
```

```
root@Deb1:/tmp/share# ccrypt -e -k /etc/cle.txt /tmp/share/test.txt
```

```
test.txt.cpt
```



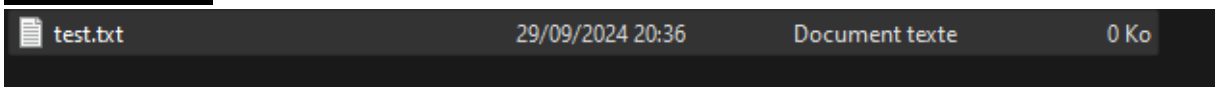

```
root@Deb1:/tmp/share# ccrypt -d -r -k /etc/cle.txt /tmp/share/test.txt.cpt
```

-d : Pour déchiffrer les fichiers.

-r : Récursivité pour parcourir les sous-répertoires.

-k keyfile.txt : Utilisation de la clé depuis le fichier de clé.

```
test.txt
```



Script Bash permettant de monter un partage SMB et de crypter ou décrypter toutes les données contenues dans le dossier partagé :

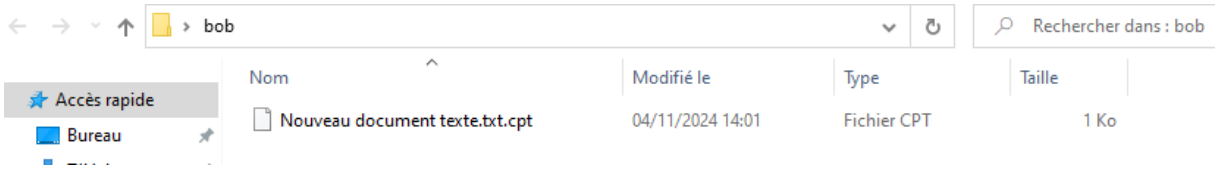
```bash
#!/bin/bash

# Vérification du nombre de paramètres
if [ "$#" -ne 4 ]; then
    echo "Usage: $0 <chemin_partage> <utilisateur/mot_de_passe> <-e|-d> <chemin_clé>"
    exit 1
fi

# Paramètres
PARTAGE="$1"
USER_PASS="$2"
ACTION="$3"
CLE="$4"

# Extraction de l'utilisateur et du mot de passe
UTILISATEUR=$(echo "$USER_PASS" | cut -d'/' -f1)
MDP=$(echo "$USER_PASS" | cut -d'/' -f2)

# Point de montage temporaire
MOUNT_POINT="/tmp/share"

# Vérification de l'action (chiffrer ou déchiffrer)
if [ "$ACTION" != "-e" ] && [ "$ACTION" != "-d" ]; then
    echo "Action non valide. Utilisez '-e' pour chiffrer ou '-d' pour déchiffrer."
    exit 1
fi

# Création du répertoire de montage s'il n'existe pas
if [ ! -d "$MOUNT_POINT" ]; then
    mkdir -p "$MOUNT_POINT"
fi

# Montage du partage SMB avec mount.cifs
mount.cifs //$PARTAGE "$MOUNT_POINT" -o user="$UTILISATEUR",password="$MDP"

# Vérification de l'effectivité du montage
if ! mountpoint -q "$MOUNT_POINT"; then
    echo "Le montage a échoué."
    exit 1
fi

echo "Partage monté avec succès sur $MOUNT_POINT."

# Chiffrement ou déchiffrement avec ccrypt
if [ "$ACTION" == "-e" ]; then
    echo "Chiffrement des fichiers..."
    find "$MOUNT_POINT" -type f -exec ccrypt -e -k "$CLE" {} \;
elif [ "$ACTION" == "-d" ]; then
    echo "Déchiffrement des fichiers..."
    find "$MOUNT_POINT" -type f -exec ccrypt -d -k "$CLE" {} \;
fi
```

```bash
# Chiffrement ou déchiffrement avec ccrypt
if [ "$ACTION" == "-e" ]; then
    echo "Chiffrement des fichiers..."
    find "$MOUNT_POINT" -type f -exec ccrypt -e -k "$CLE" {} \;
elif [ "$ACTION" == "-d" ]; then
    echo "Déchiffrement des fichiers..."
    find "$MOUNT_POINT" -type f -exec ccrypt -d -k "$CLE" {} \;
fi

# Démontage du partage SMB
umount "$MOUNT_POINT"
if [ $? -ne 0 ]; then
    echo "Erreur lors du démontage du partage SMB."
    exit 1
fi

echo "Partage démonté avec succès."

exit 0
```

Test Script :

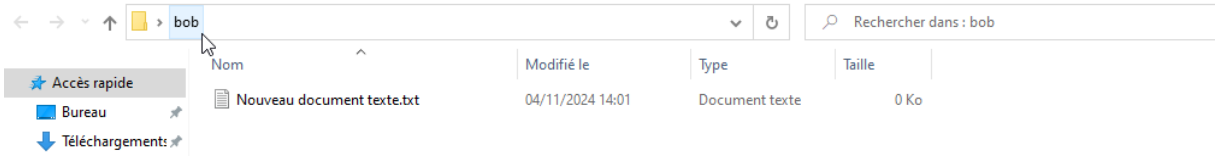
```
root@ClientDebian2:/home/theo# ./bash.sh 192.168.1.98/bob utilisateur/Azerty@123 -e /home/theo/cle.txt
Partage monté avec succès sur /tmp/share.
Chiffrement des fichiers...
Partage démonté avec succès.
root@ClientDebian2:/home/theo#
```

Si on regarde le partage sur Windows on voit que le contenu a bien été crypté :



Test décryptage :

```
root@ClientDebian2:/home/theo# ./bash.sh 192.168.1.98/bob utilisateur/Azerty@123 -d /home/theo/cle.txt
Partage monté avec succès sur /tmp/share.
Déchiffrement des fichiers...
Partage démonté avec succès.
root@ClientDebian2:/home/theo#
```




```
root@Deb1:/# apt install lynis
```

```
Lynis security scan details:

Hardening index : 60 [############        ]
Tests performed : 253
Plugins enabled : 1

Components:
- Firewall               [V]
- Malware scanner        [X]

Scan mode:
Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]

Lynis modules:
- Compliance status      [?]
- Security audit         [V]
- Vulnerability scan     [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                     : /var/log/lynis-report.dat

================================================================================
```