# TP Linux CTF

## Procédure de création de la VM (Configuration)

Cette section détaille les étapes suivies pour mettre en place l'environnement de base du CTF avant l'injection des vulnérabilités. L'objectif est d'obtenir une installation "propre" et fonctionnelle.

**Caractéristiques de la Machine Virtuelle**

- **Système d'exploitation :** Debian 13 (Trixie) 64-bit.
- **Ressources :** 1 vCPU, 1024 Mo de RAM, 10 Go de disque (allocation dynamique).
- **Interface :** Console uniquement (pas d'interface graphique).

### Étapes de Configuration

### Étape 1 : Installation du Système

1. **Montage de l'ISO :** Utilisation de l'image debian-13.3.0-amd64-DVD-1.iso.
2. **Partitionnement :** Choix du partitionnement assisté ("Tout dans une seule partition") pour simplifier la gestion des fichiers du challenge.
3. **Logiciels :**
   - **Désactiver :** Environnement de bureau Debian, GNOME.
   - **Activer :** Serveur SSH, Utilitaires usuels du système.

### Étape 2 : Configuration Réseau et Accès

Pour que le CTF soit accessible, la machine doit être joignable par l'attaquant.

1. **Mode Réseau :** Configuration de la carte réseau en mode **Accès par pont (Bridge)** ou **Réseau privé hôte (Host-only)** dans les réglages de la VM.
2. **Activation du SSH :**
   - Vérification du statut : systemctl status ssh
   - Recherche de l'adresse IP : ip a

### Étape 3 : Installation des Outils de Base

Installation des paquets nécessaires pour l'administration et la mise en place des futurs niveaux :

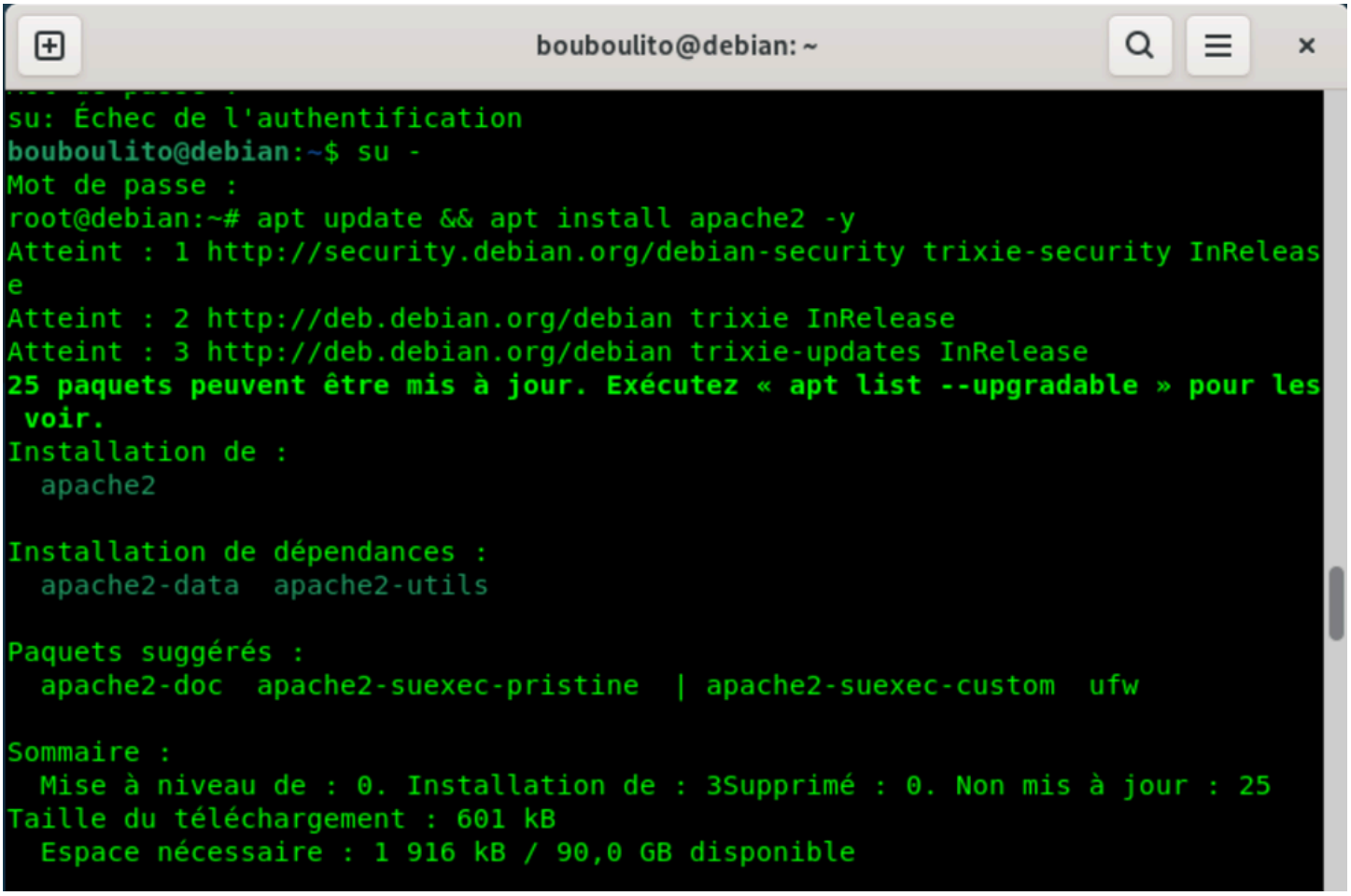
Bash

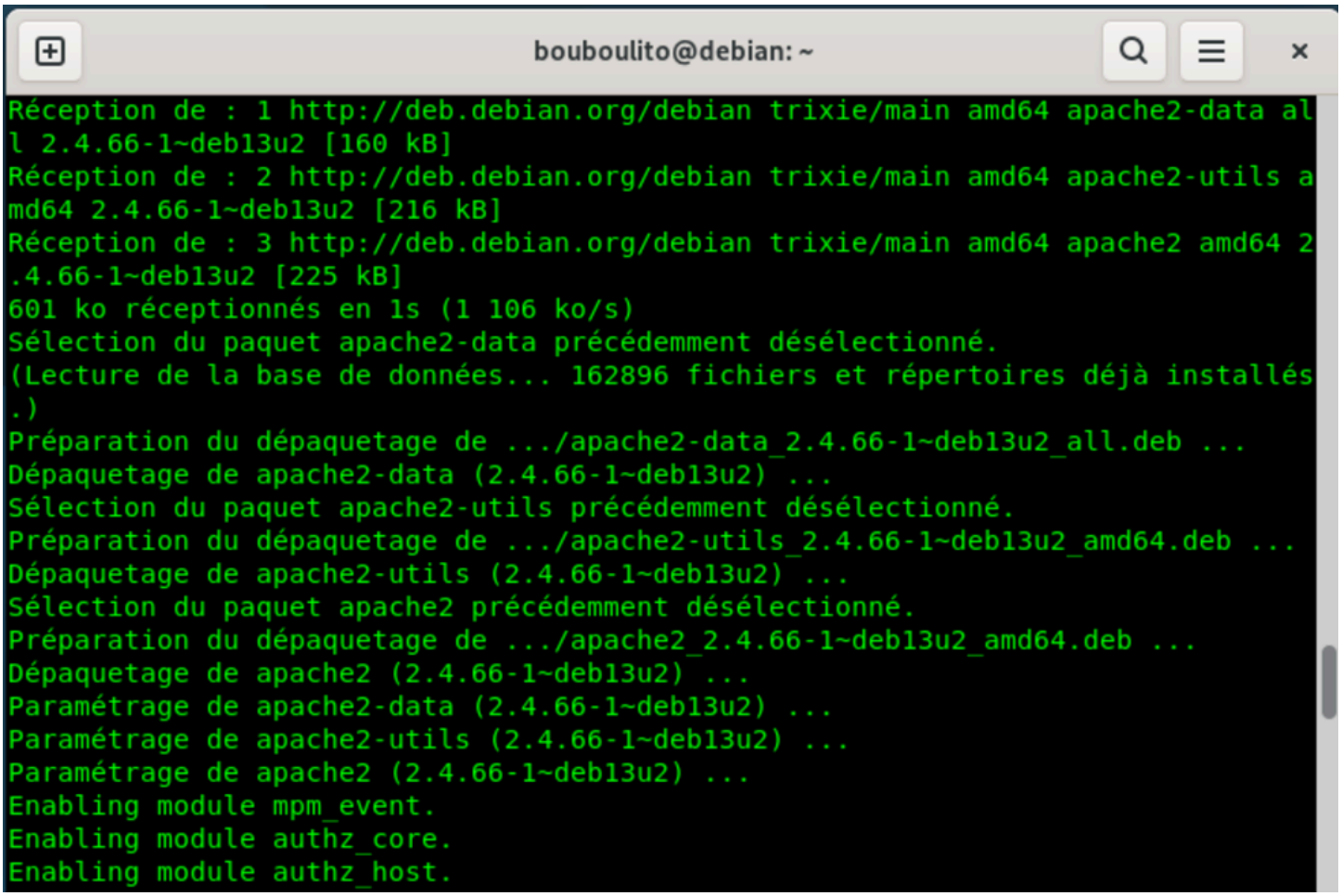
# Mise à jour des dépôts

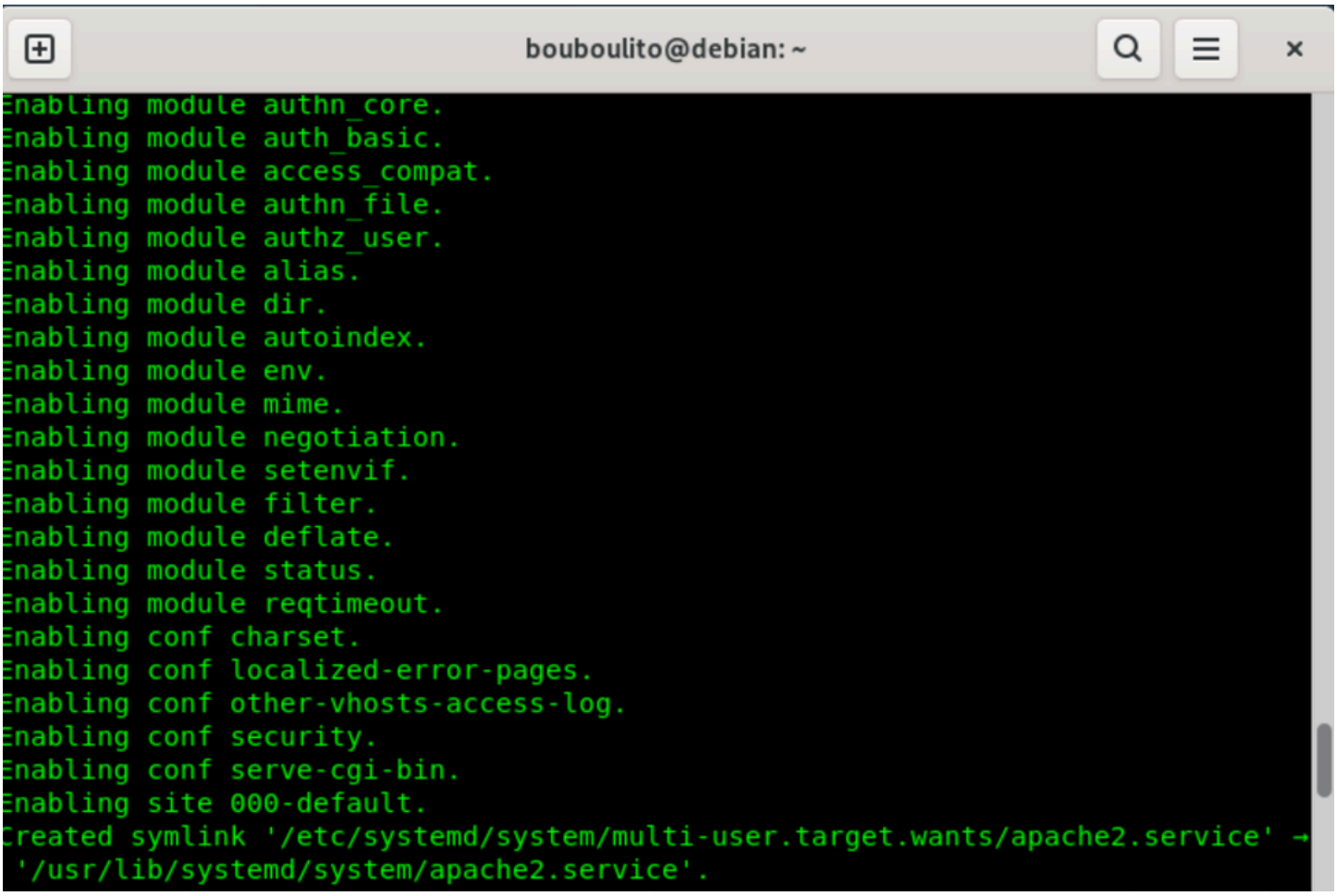
apt update && apt upgrade -y

# Installation des outils essentiels

apt install -y sudo nano vim net-tools openssh-server coreutils






**Étape 4 : Nettoyage et Snapshot (Crucial)**

Une fois la base installée, le système est nettoyé pour éviter de donner des indices sur l'installation au futur participant.

1. **Nettoyage des fichiers temporaires :** apt clean
2. **Snapshot :** Création d'un "Instantané" dans le logiciel de virtualisation (VirtualBox/VMware).
   - *Pourquoi ?* Cela permet de réinitialiser le CTF instantanément si un utilisateur casse le système ou si une erreur est commise lors de la configuration des permissions.

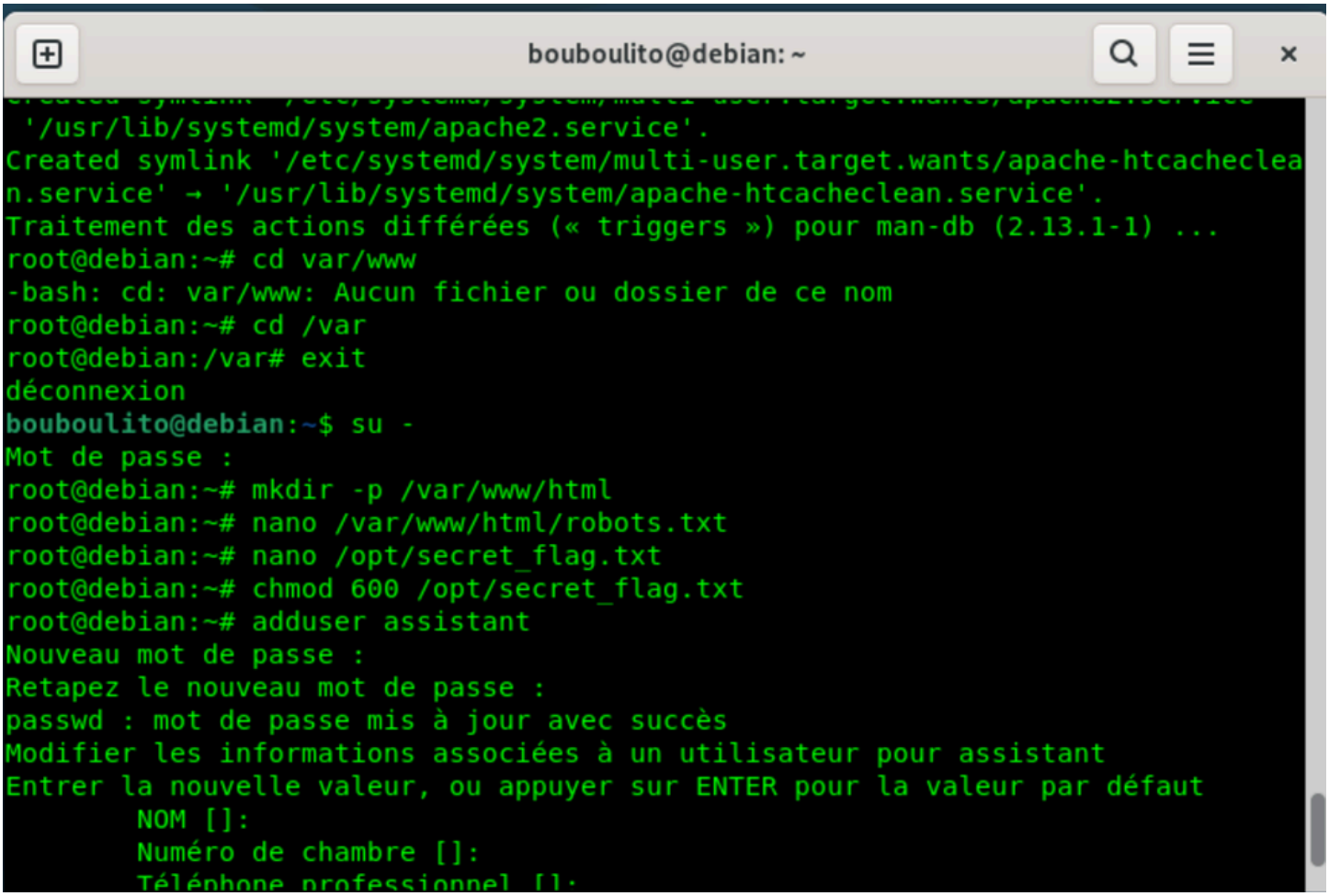
**Niveau 1 Reconnaissance et Fuite de données (Information Disclosure)**

C'est la porte d'entrée de toute intrusion système.

- **Le concept :** L'attaquant cherche des pistes sur le serveur web (le fichier robots.txt) pour identifier des fichiers sensibles. Il exploite ensuite des erreurs de configuration (mots de passe oubliés dans des scripts de maintenance) pour obtenir un premier accès utilisateur.
- **L'action :** Créer un fichier "indice", configurer un utilisateur "victime" avec un mot de passe caché, et mettre en place une énigme pour l'escalade vers root via un fichier caché.
- **Le défi :** L'attaquant doit d'abord explorer le service Web, puis fouiller le système pour trouver le mot de passe de l'utilisateur assistant avant de résoudre une énigme logique pour devenir root et lire le flag final.

**Étape 1 : Préparer les pistes et le Flag (Serveur Web)** Pour que l'attaquant trouve une piste sur le serveur web, nous configurons un indice public et le fichier secret que l'attaquant doit tenter de lire.

1. **Créer le dossier manuellement** : mkdir -p /var/www/html
2. **Créer le fichier "indice" (Hint) :**
   - Commande : nano /var/www/html/robots.txt
   - Écrire à l'intérieur : "Le flag est dans /opt/secret_flag.txt mais tu n'as pas les permissions !"
   - Sauvegarde
3. **Créer le véritable "Flag" :**
   - Commande : nano /opt/secret_flag.txt
   - Écrire le code secret : FLAG{RESEAU_MASTER_2026}
4. **Configurer les permissions (Très important !) :**
   - En tant que root, taper : chmod 600 /opt/secret_flag.txt




**Étape 2 : Créer l'utilisateur "victime" et cacher le mot de passe** L'attaquant va d'abord pirater un compte utilisateur simple avant de devenir root.

1. **Créer l'utilisateur :**
   - Commande : adduser assistant
   - Choisis un mot de passe simple à cacher : password1234.
2. **Cacher le mot de passe (La piste) :** On utilise un faux script de maintenance.

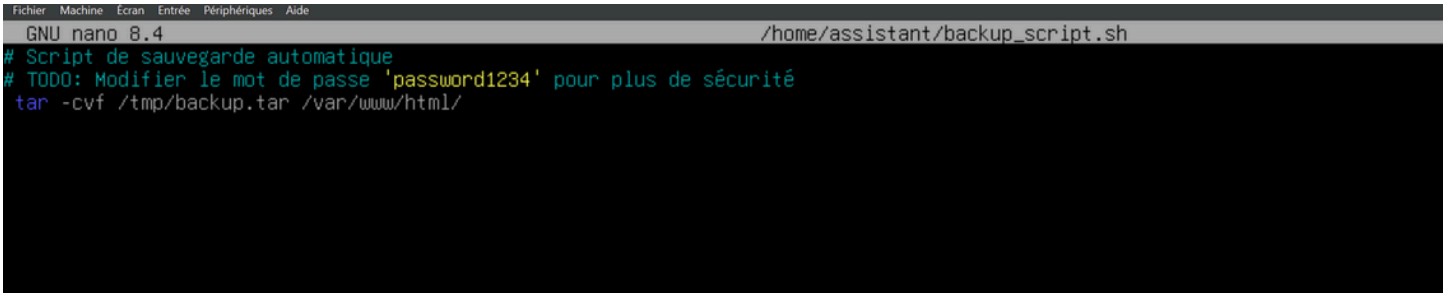
- Commande : nano /home/assistant/backup_script.sh
- Écris ceci :

Bash

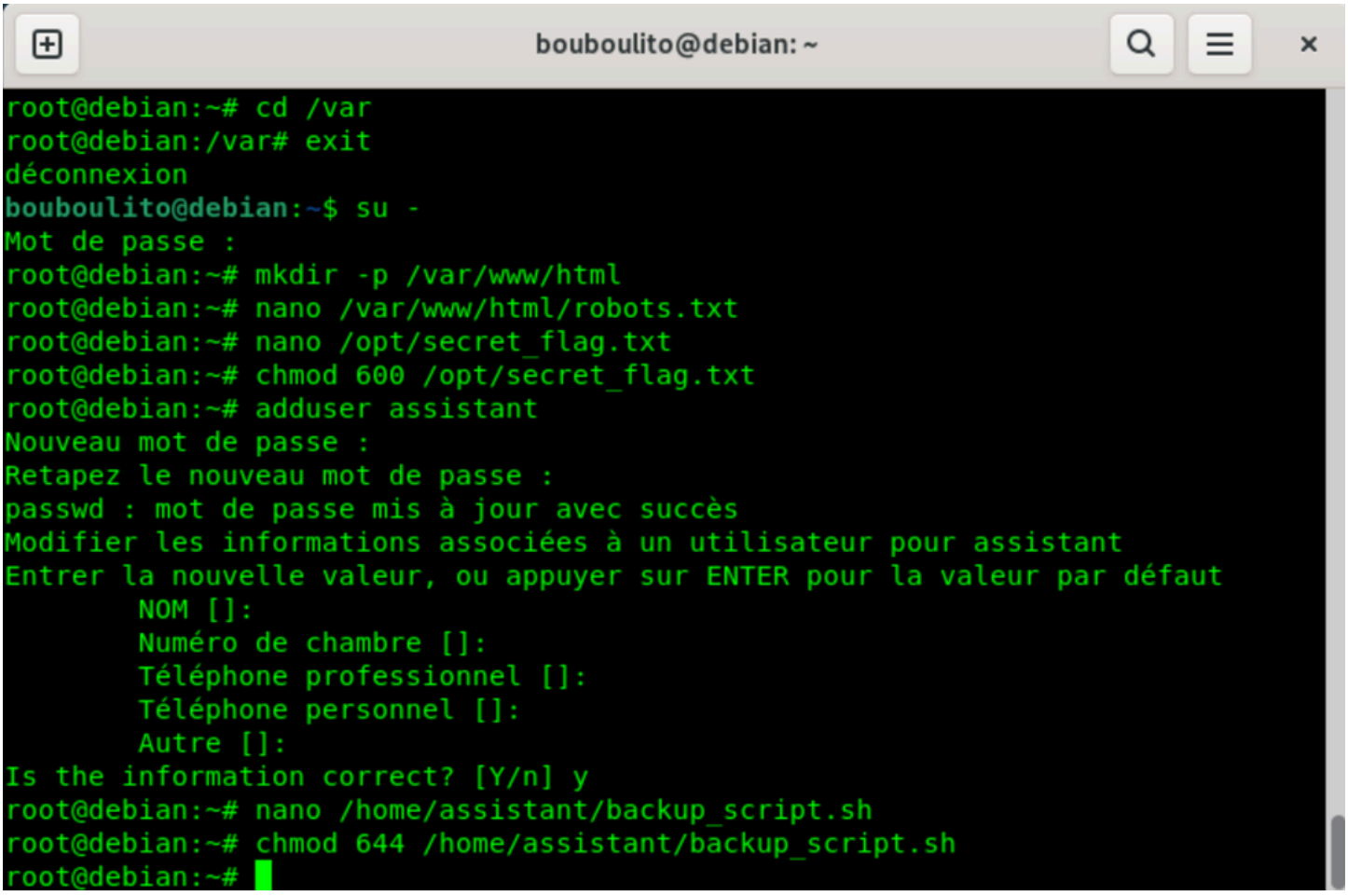
# Script de sauvegarde automatique

# TODO: Modifier le mot de passe 'password1234' pour plus de sécurité

tar -cvf /tmp/backup.tar /var/www/html/

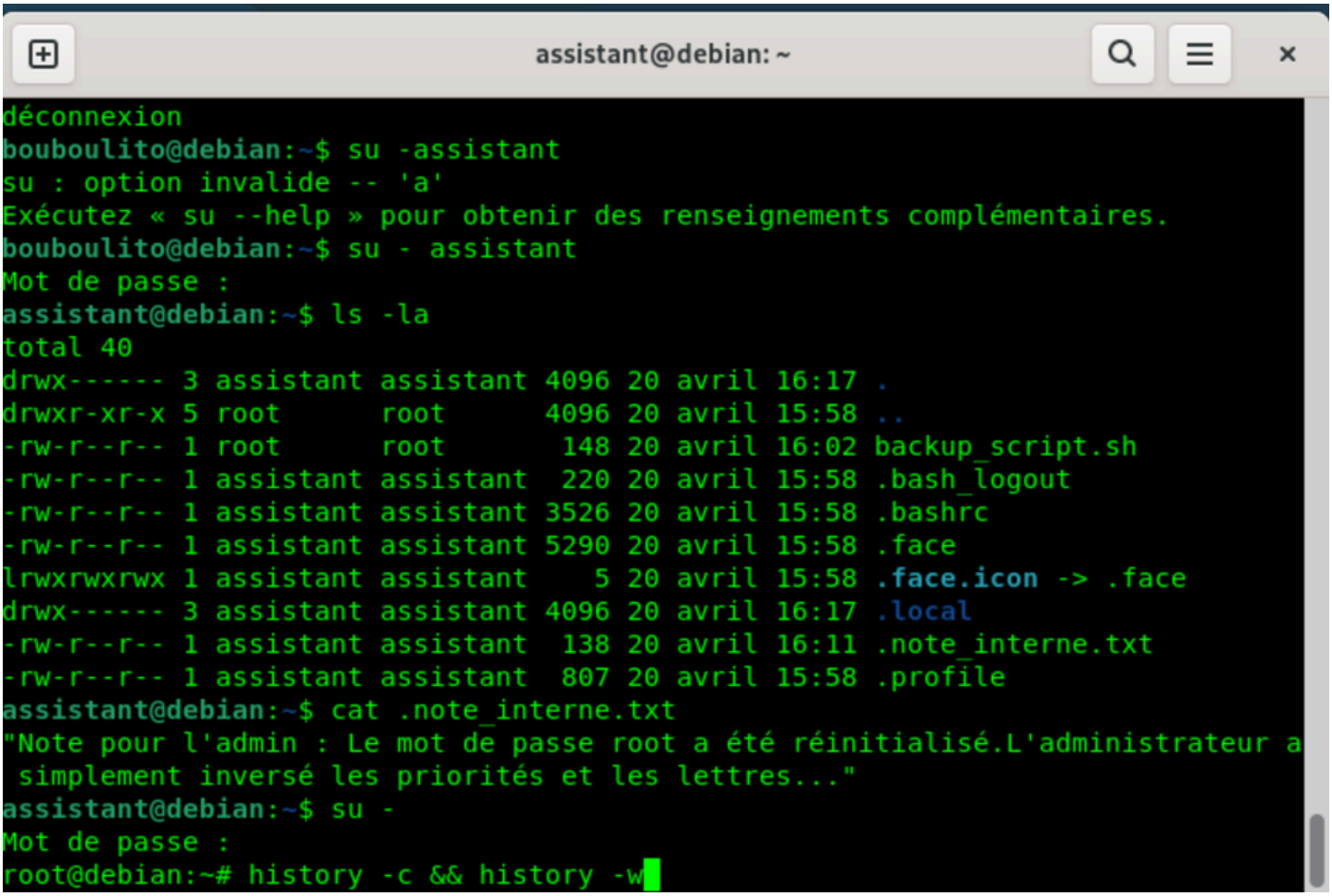



1. **Ajuster les permissions :**
   - Commande : chmod 644 /home/assistant/backup_script.sh
   - **Pourquoi ?** Pour que l'attaquant puisse trouver ce fichier et le lire sans être encore connecté comme assistant.



**Étape 3 : Configurer l'énigme de l'escalade (Accès Root)** Pour passer de assistant à root, l'attaquant doit trouver un indice caché menant au mot de passe root.

1. **Créer un fichier caché (Dotfile) :**
   - Assure-toi d'être dans le dossier de l'assistant : cd /home/assistant
   - Commande : nano .note_interne.txt (Le point au début rend le fichier caché).
   - Écris : "L'administrateur a simplement inversé les priorités... et les lettres."
2. **Configurer le mot de passe Root :**
   - Commande : passwd root
   - Action : Tape toor (l'inverse de root), puis valide deux fois.
3. **Vérifier les permissions du fichier caché :**
   - Commande : chown assistant:assistant /home/assistant/.note_interne.txt
   - Commande : chmod 644 /home/assistant/.note_interne.txt
   - **Pourquoi ?** En français : "On s'assure que l'utilisateur a les droits de lecture sur sa propre note."




**Niveau 2 L'abus de droits SUDO (privilege Escalation)**

C'est un classique indispensable.

· **Le concept :** Tu autorises l'utilisateur à lancer un programme très spécifique en tant que root, mais ce programme permet de s'échapper vers le terminal.

· **L'action :** Modifie le fichier /etc/sudoers pour permettre à l'utilisateur de lancer find ou vim sans mot de passe.

· **Le défi :** L'attaquant doit trouver comment utiliser sudo find . -exec /bin/sh \; -quit pour devenir root. C'est ce qu'on appelle exploiter les **GTFOBins**.

**Étape 1 : Installer l'outil nécessaire**

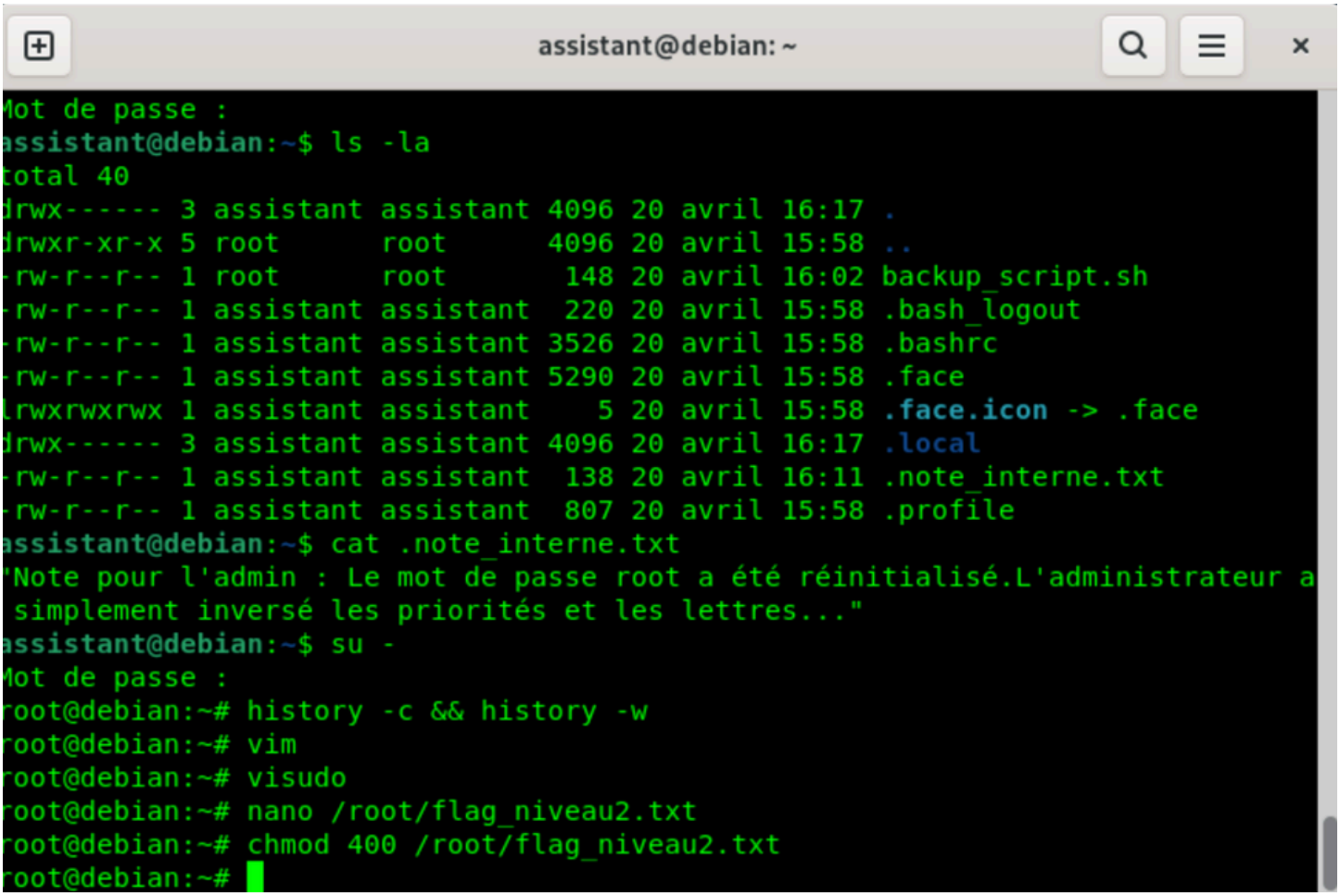
Assure-toi que l'éditeur **Vim** est présent sur la machine cible.

· **Commande :** apt update && apt install vim -y

**Étape 2 : Configurer la vulnérabilité (Le fichier Sudoers)**

Nous allons autoriser l'utilisateur assistant à lancer Vim avec les droits root, mais **sans demander de mot de passe**. C'est ici que se trouve la faille.

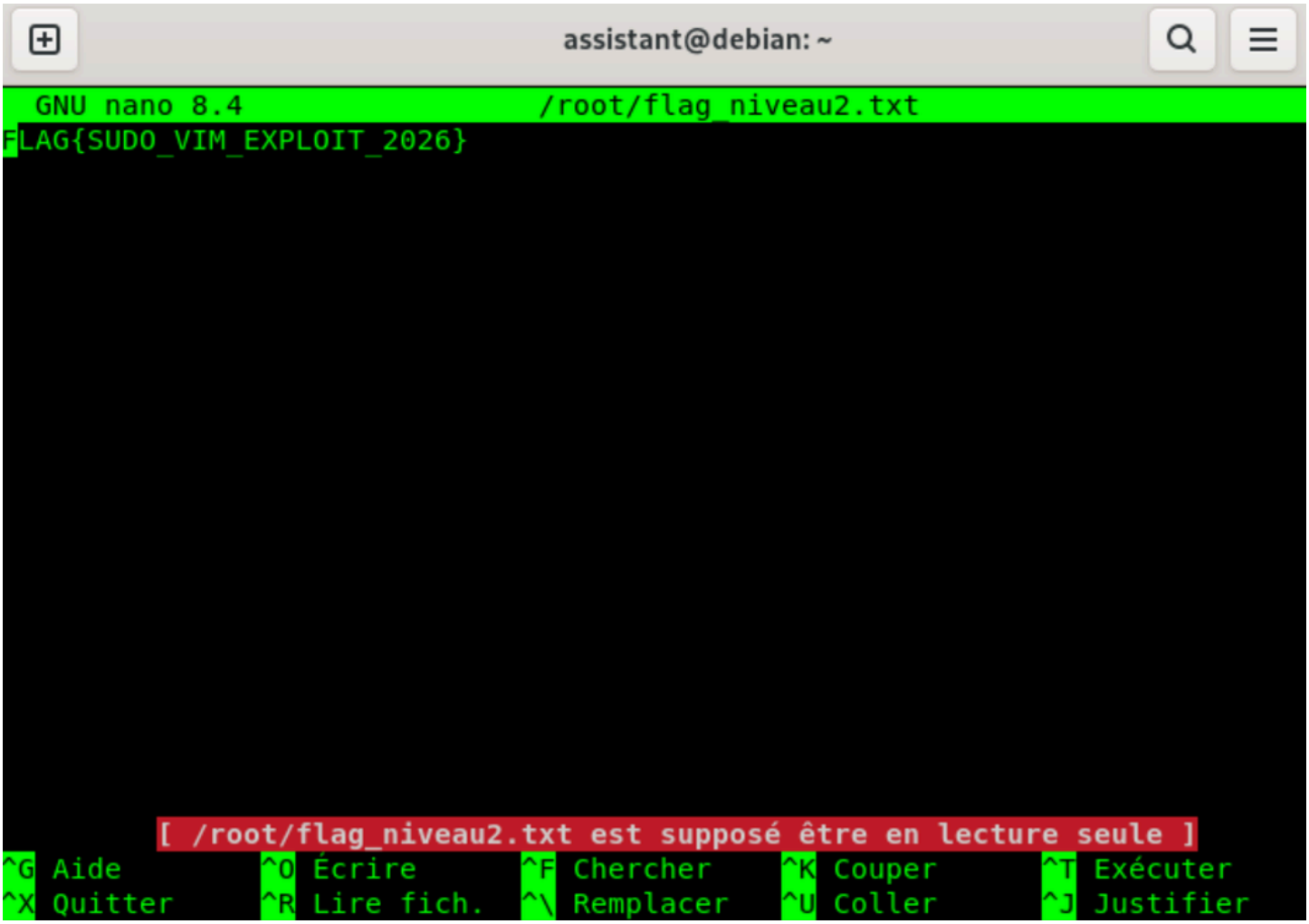
1. Ouvre le fichier de configuration des privilèges avec la commande sécurisée :
    - **Commande :** visudo
2. Ajoute cette ligne tout à la fin du fichier :
    - assistant ALL=(ALL) NOPASSWD: /usr/bin/vim
3. Sauvegarde et quitte (Ctrl+O, Entrée, puis Ctrl+X).



**Étape 3 : Créer le Flag de victoire**

Le "Flag" doit se trouver dans un endroit accessible uniquement par **root**.

1. **Commande :** nano /root/flag_niveau2.txt
2. Écris ton code secret, par exemple : FLAG{SUDO_VIM_EXPLOIT_2026}
3. Protège le fichier : chmod 400 /root/flag_niveau2.txt (Seul root pourra le lire).




## Niveau 3 : Exploitation d'une tâche automatisée (Cron Job)

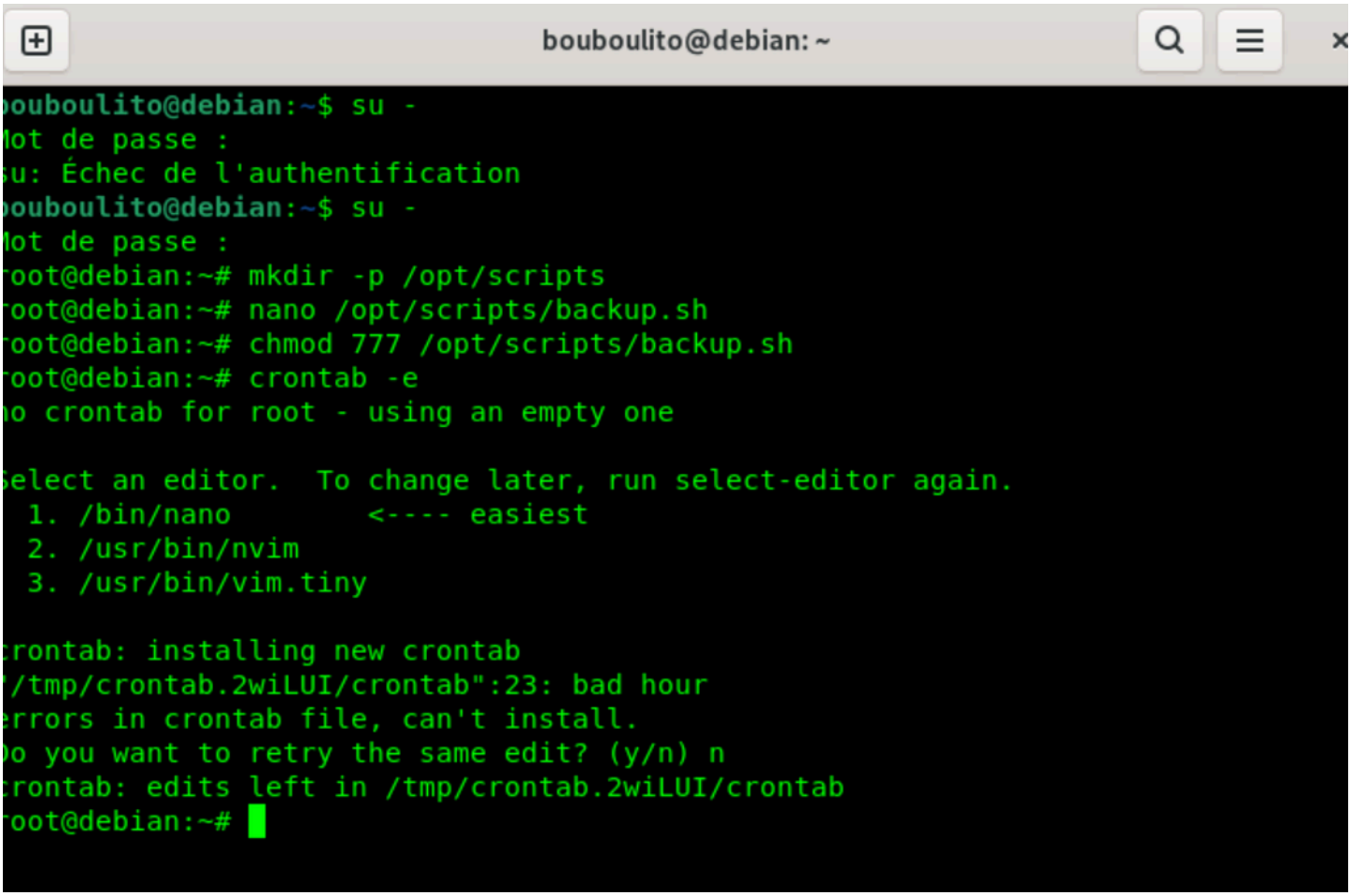
**Objectif :** Comprendre l'automatisation et les dangers des permissions globales (777).

**Mise en place (Côté Administrateur)**

1. **Créer le script vulnérable :**
   - mkdir -p /opt/scripts
   - nano /opt/scripts/backup.sh
   - Contenu : #!/bin/bash \n cp /var/www/html/robots.txt /tmp/robots_backup.txt

2. **Rendre le script modifiable par l'utilisateur "assistant" :**

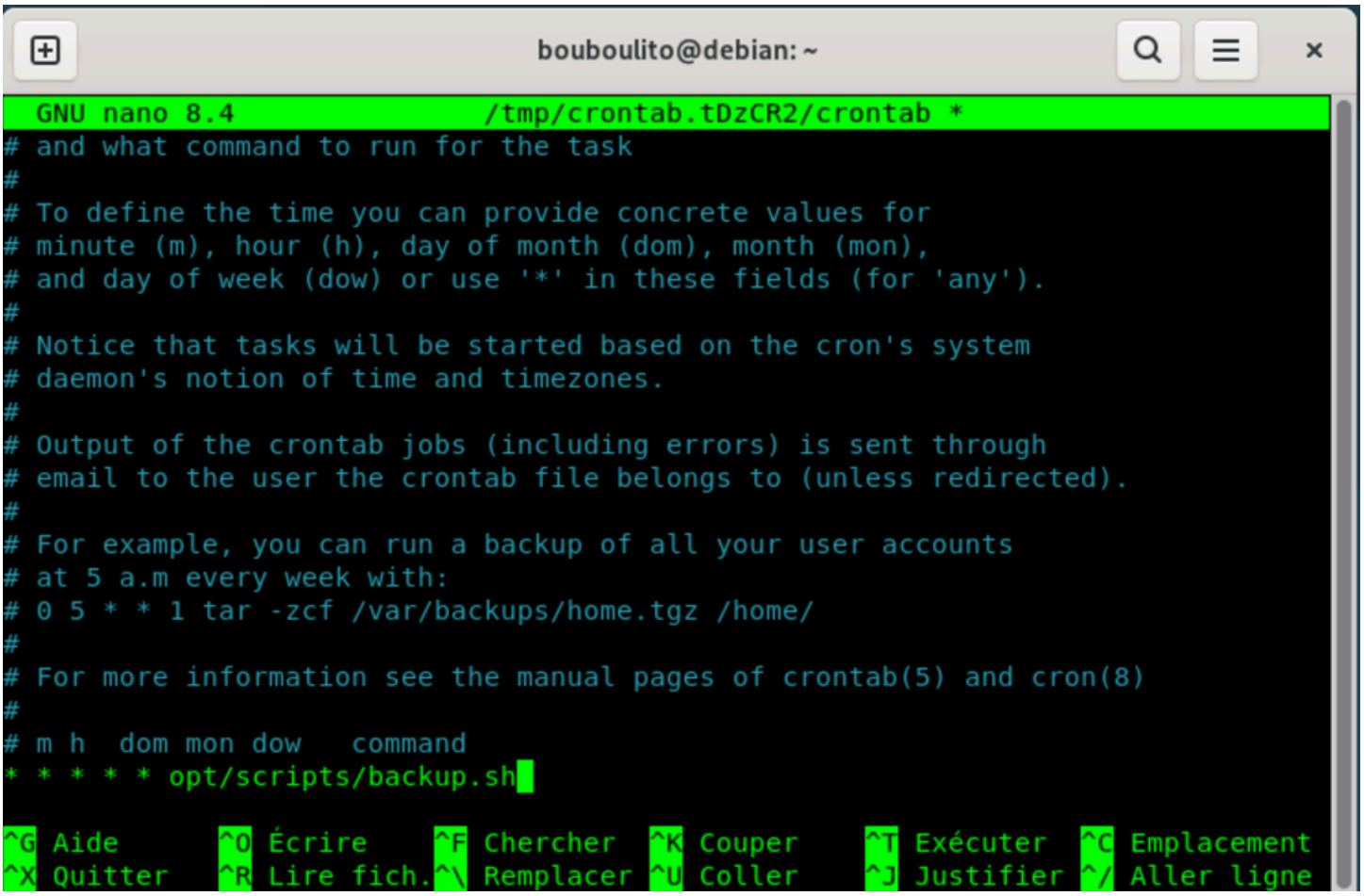
    - chmod 777 /opt/scripts/backup.sh

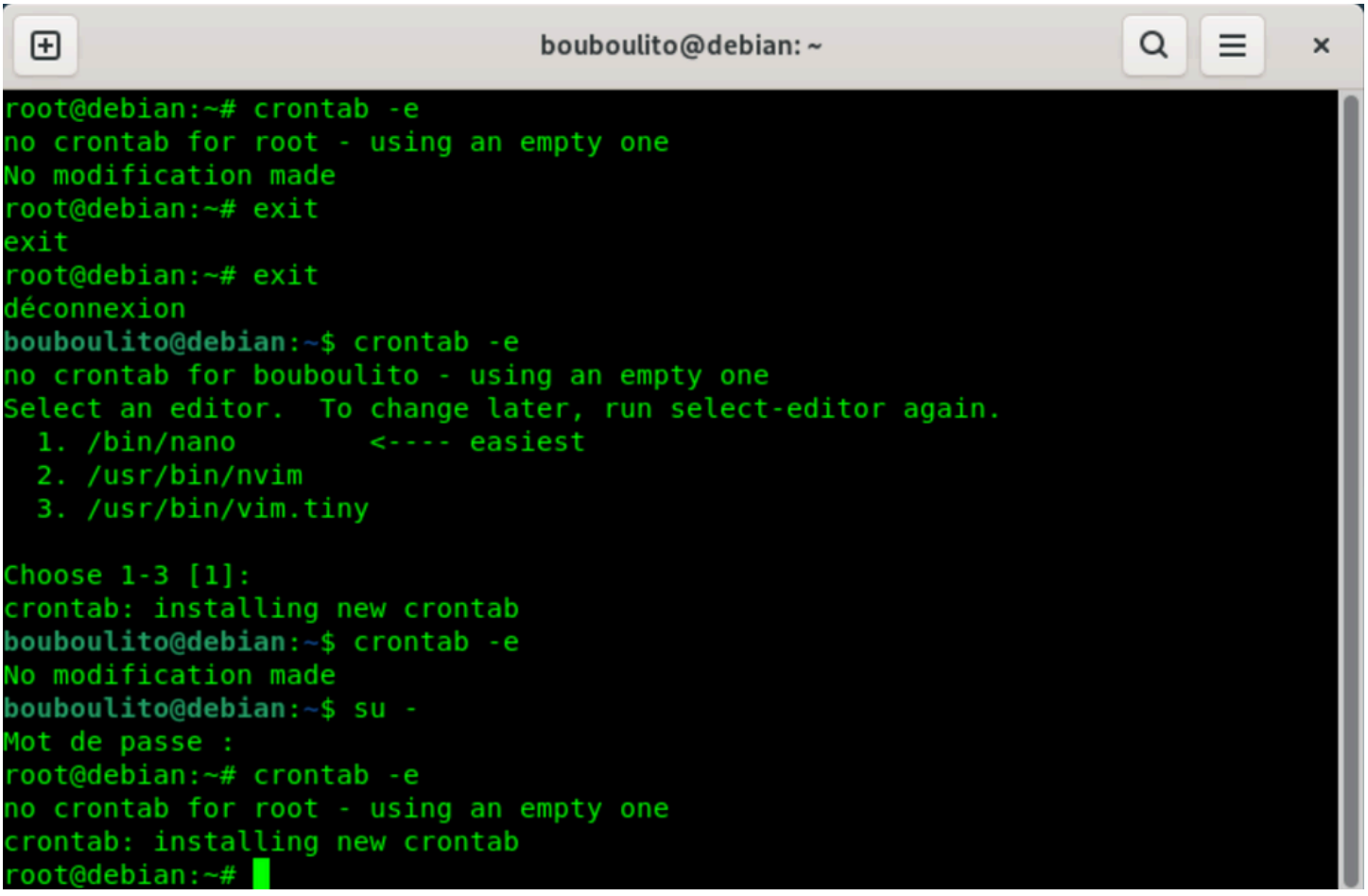



3. **Automatiser l'exécution en root :**

    - crontab -e

    - Ajoute la ligne : * * * * * /opt/scripts/backup.sh (s'exécute chaque minute).




**Description du challenge (Pour le Wiki)**

"Un script de maintenance tourne chaque minute sur le serveur avec des privilèges élevés. Si seulement vous pouviez 'aider' ce script à faire autre chose que sa sauvegarde habituelle..."

## Niveau 4 : Détournement de tâche Cron (Cron Hijacking)

C'est une technique redoutable basée sur l'automatisation.

- **Le concept :** Le système exécute automatiquement des scripts de maintenance en tant que root. Si un utilisateur peut modifier l'un de ces scripts, il peut forcer le système à exécuter ses propres commandes avec les privilèges maximum.
- **L'action :** Créer un script avec des permissions trop larges (777) et l'ajouter à la table de planification (crontab) de l'administrateur.
- **Le défi :** L'attaquant doit identifier les scripts qui tournent régulièrement et injecter un code (comme un "Reverse Shell" ou un changement de permissions) pour prendre le contrôle.

**Étape 1 : Créer le script vulnérable** Nous allons simuler un script de sauvegarde qui appartient à root mais que tout le monde peut modifier.

- **Commande :** mkdir -p /opt/scripts
- **Commande :** nano /opt/scripts/backup.sh
- **Écris à l'intérieur :**

  Bash

  #!/bin/bash

  # Sauvegarde automatique des logs

  cp /var/log/auth.log /tmp/auth_backup.log

- **Permissions (La faille) :** chmod 777 /opt/scripts/backup.sh

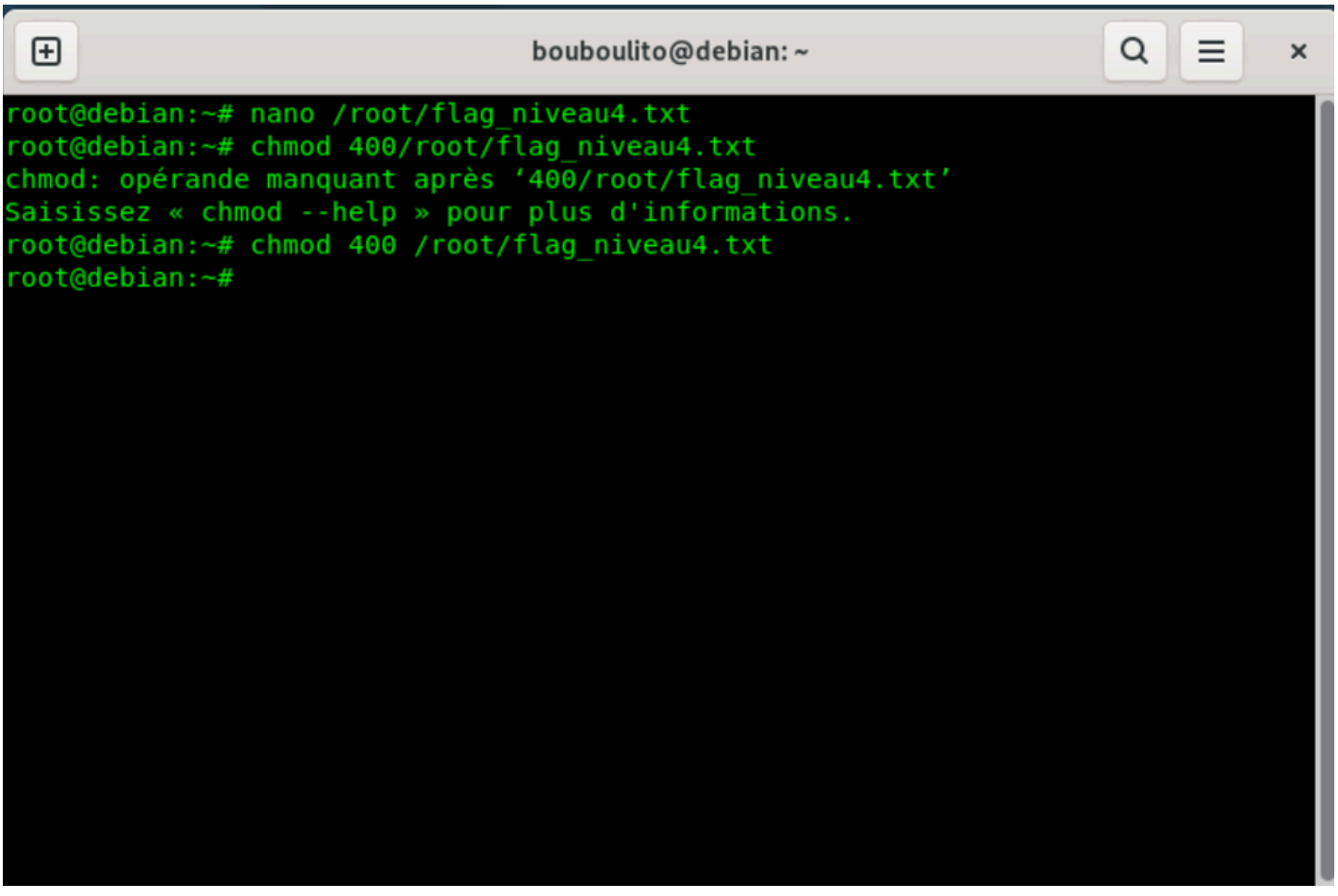
**Étape 2 : Configurer l'automatisation (Cron)** Il faut maintenant dire au système de lancer ce script chaque minute.

1. **Ouvre le planificateur :** crontab -e
2. **Ajoute cette ligne tout à la fin :** * * * * * /opt/scripts/backup.sh
3. **Sauvegarde et quitte.**

**Étape 3 : Créer le Flag** Le flag sera placé dans le répertoire personnel de root.

1. **Commande :** nano /root/flag_niveau4.txt
2. **Écris le code secret :** FLAG{CRON_HIJACK_SUCCESS_2026}


3. **Protège le fichier :** chmod 400 /root/flag_niveau4.txt



## Niveau 5 L'exploitation du Bit SUID (Set User ID)

C'est le défi final qui utilise les permissions avancées d'Unix.

· **Le concept :** Le bit SUID permet à un utilisateur de lancer un programme avec les permissions du propriétaire (root). Si on active ce bit sur un outil de lecture, l'utilisateur peut lire n'importe quel secret sur le disque.

· **L'action :** Créer une copie d'un utilitaire système (cat) et lui donner des droits spéciaux pour qu'il devienne une "porte dérobée".

· **Le défi :** L'attaquant doit scanner le système pour trouver les fichiers SUID et découvrir qu'il peut lire le flag final, qui est en plus encodé pour corser l'épreuve.

**Étape 1 : Créer le binaire à privilèges** Nous allons créer une version de cat nommée super_reader qui a des pouvoirs de root.

· **Commande :** cp /bin/cat /usr/bin/super_reader

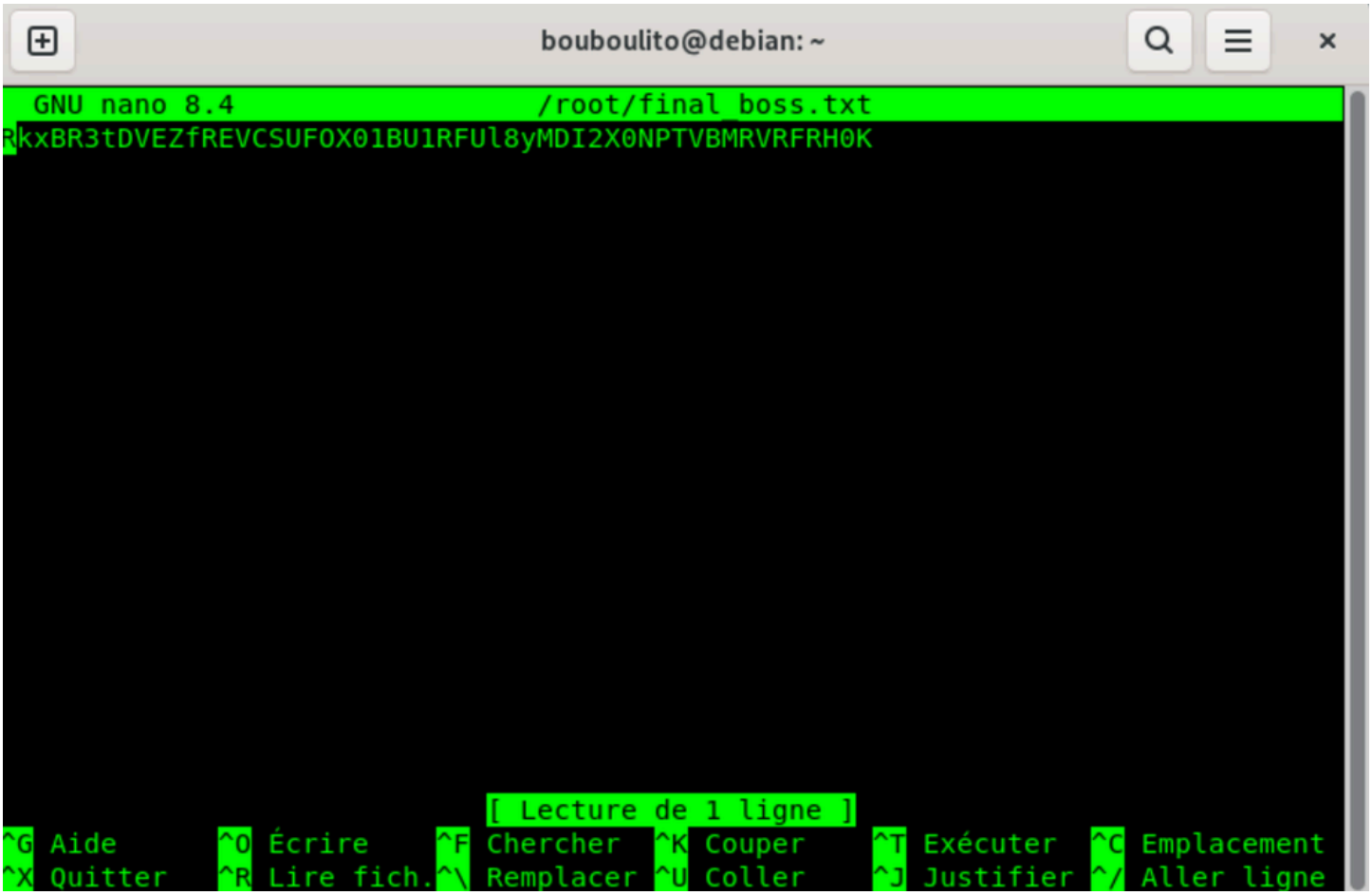
· **Commande :** chown root:root /usr/bin/super_reader

· **Commande (Activer le SUID) :** chmod 4755 /usr/bin/super_reader

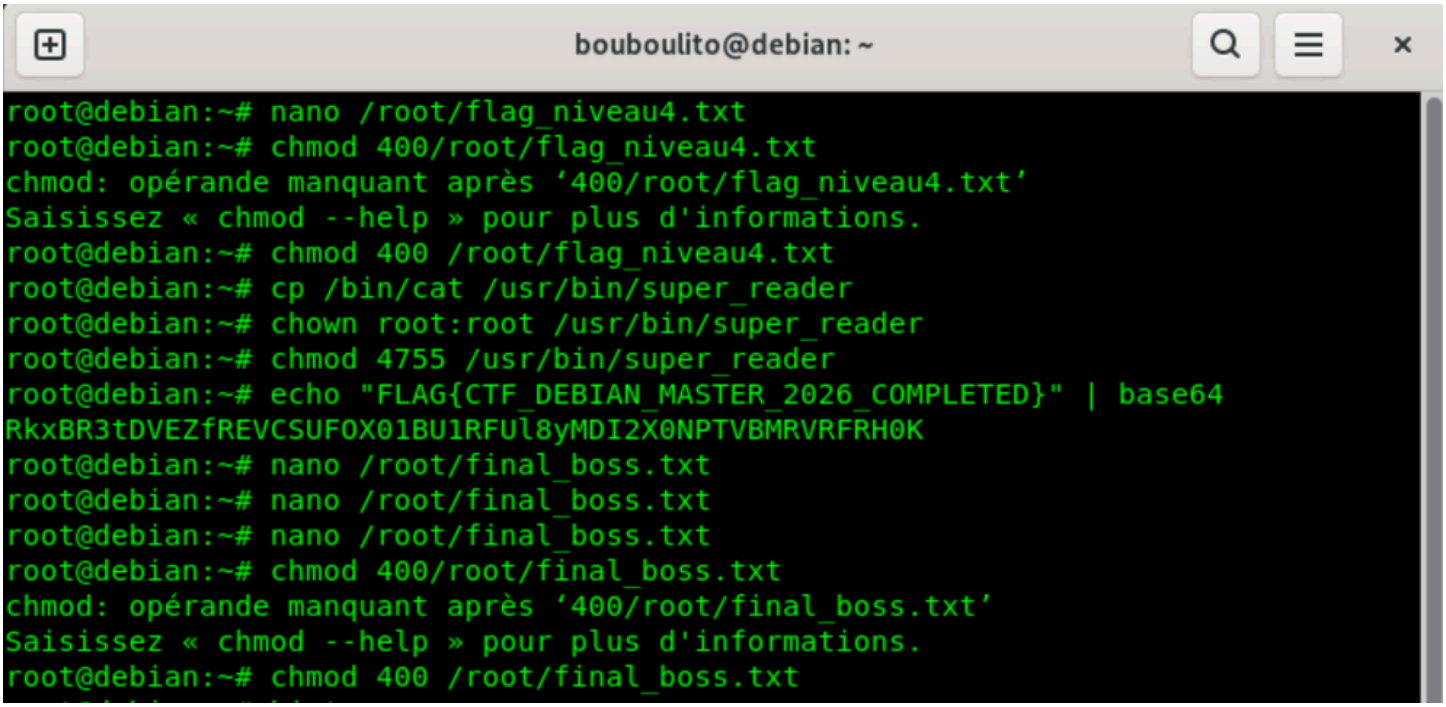
**Étape 2 : Créer et encoder le Flag Final** Pour ce dernier niveau, nous allons utiliser l'obfuscation.

1. **Génère ton flag en Base64 :** echo "FLAG{CTF_DEBIAN_MASTER_2026_COMPLETED}" | base64

2. **Copie le résultat obtenu** (exemple : RkxBR3tDVEZfREVCSUFOX01BU1RFUl8yMDI2X0NPTVBMRVRFRH0K).

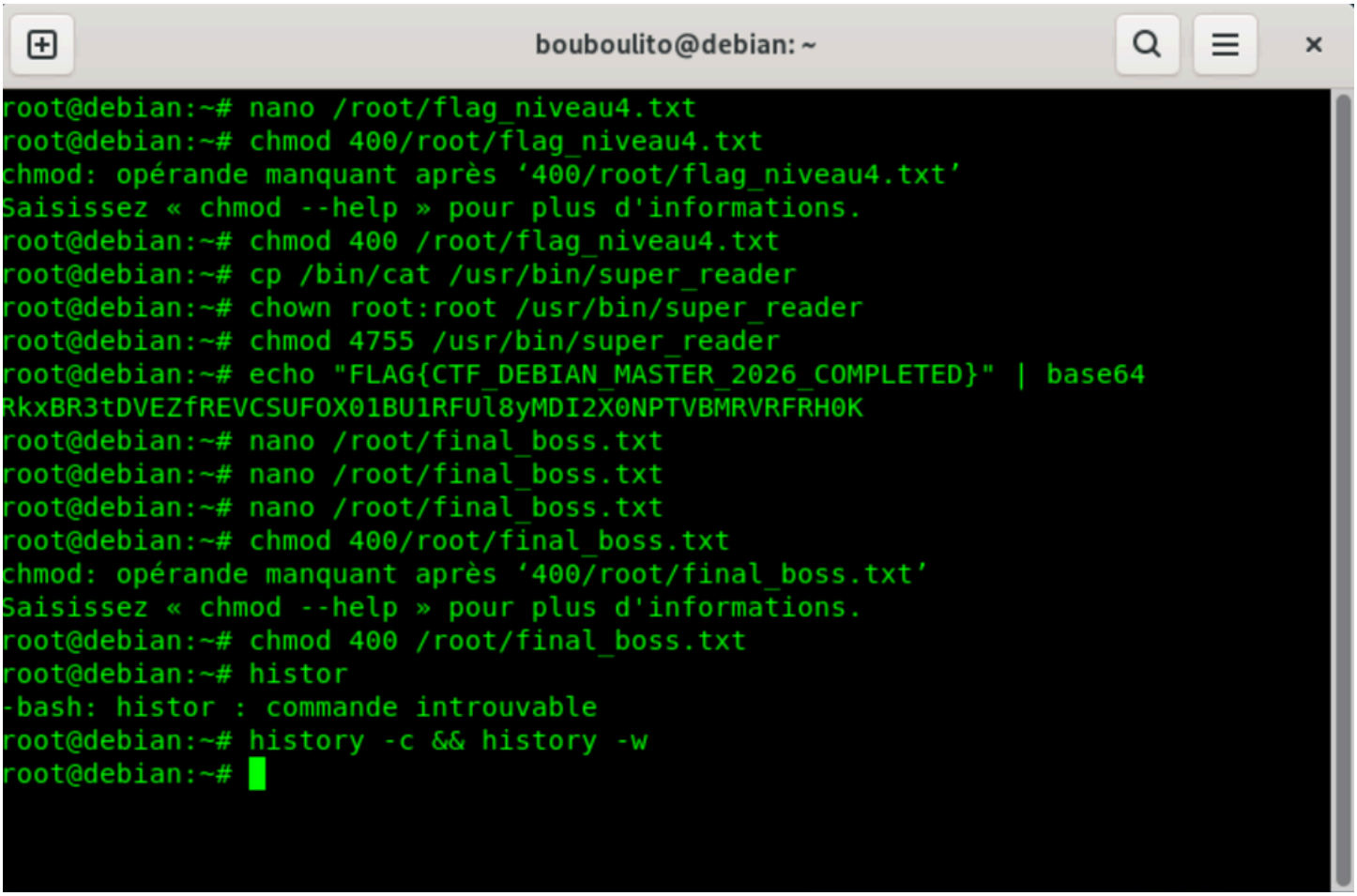



3. **Crée le fichier secret :** nano /root/final_boss.txt
4. **Colle la chaîne Base64 à l'intérieur**, sauvegarde et quitte.
5. **Protège le fichier :** chmod 400 /root/final_boss.txt



**Étape 3 : Nettoyer les traces (Optionnel mais recommandé)** Pour que le participant ne puisse pas simplement tricher en regardant ce que tu as fait :

- **Commande :** history -c && history -w




## Conseils de mise en œuvre (Best Practices)

- **Audit des fichiers :** Vérifiez toujours avec ls -l que vos fichiers flag.txt ne sont pas lisibles par un utilisateur simple sans l'exploit prévu.
- **Nettoyage final :** Avant de distribuer la VM, effacez vos traces :
- Bash
- history -c && history -w
- **Stabilité :** Testez la chaîne complète de l'utilisateur level0 jusqu'à root avant de valider la Snapshot finale.