DÉLÉGATION DE DROITS AVEC L'OUTIL SUDO TENTATIVE DE CONTOURNEMENT DE LA POLITIQUE DE SÉCURITÉ

DESCRIPTION DU THÈME

Propriétés	Description
Intitulé long	Délégation de droits avec l'outil SUDO et tentative de contournement de la politique de sécurité
Formation(s) concernée(s)	☐ Classes de première Sciences et technologies du management et de la gestion (STMG) ☐ Terminale STMG Système d'information de gestion (SIG) ☒ BTS Services Informatiques aux Organisations
Matière(s)	☐ Sciences de gestion ☐ SIG ☐ Bloc 1 – Support et mise à disposition de services informatiques ☐ Bloc 2 SISR – Administration des systèmes et des réseaux ☐ Bloc 2 SLAM – Conception et développement d'applications ☒ Bloc 3 SISR – Cybersécurité des services informatiques ☒ Bloc 3 SISR – Cybersécurité des services informatiques
Présentation	L'objectif de côté Labo est de : • Comprendre et manipuler l'outil sudo pour gérer les permissions. • Modifier le fichier sudoers de manière sécurisée. • Déléguer des permissions spécifiques à un utilisateur. • Expérimenter les risques liés à une mauvaise configuration.
Savoirs	Principe du moindre privilège
Compétences	**Sécuriser les équipements et les usages des utilisateurs** • Gérer les accès et les privilèges appropriés • Vérifier l'efficacité de la protection **Assurer la cybersécurité d'une infrastructure réseau, d'un système, d'un service** • Prévenir les attaques • Analyser les incidents de sécurité, proposer et mettre en œuvre des contre-mesures
Prérequis	Connaissances de base sur Linux
Outils	Machine sur Linux avec l'outil « sudo »

Mots-clés	sudo, élévation de privilèges, principe du moindre privilège, audit de l'utilisation de commandes
Durée	4 heures
Auteur·e·s	David Duron avec la relecture, les tests et les suggestions d'Apollonie Raffalli
Version	V1.0
Date de publication	Novembre 2025

DERNIÈRES RÉVISIONS

Ce tableau contient les modifications apportées au document après sa publication uniquement.

Date	Auteur·e	Description

SOMMAIRE

A. L'OUTIL SUDO

A.1. GÉNÉRALITÉS

L'outil SUDO permet de déléguer certains droits à un utilisateur ou à un groupe, sans pour autant ouvrir les droits à tout le monde et tout le temps. Par ailleurs, l'utilisation de ces droits peut être consignée dans les journaux d'événements et permettre ainsi une traçabilité des opérations effectuées avec élévation de privilèges grâce à la commande sudo. La trace laissée n'est plus anonyme ou générique, comme quand on partage à plusieurs un compte d'administration : chaque opération réalisée avec des privilèges est consignée avec le nom de l'utilisateur qui l'a effectuée.

Sudo est couramment utilisé dans les scénarios suivants :

- **Installation de Logiciels :** les utilisateurs peuvent utiliser sudo pour installer des logiciels système sans avoir besoin d'accès administratif complet.
- **Mise à Jour du Système :** les mises à jour du système, y compris la mise à jour de sécurité, sont généralement effectuées avec sudo.
- **Gestion des Services :** redémarrer des services, démarrer/arrêter des démons système, etc.
- **Configuration Réseau :** modifier les paramètres réseau, ajouter des routes, etc.

Vous utiliserez une machine (potentiellement virtuelle) LINUX (Debian/ubuntu/Kali, etc.) pour mener à bien cette découverte.

Vérifier que l'outil sudo est installé sur votre système en affichant sa version (vous pouvez bien sûr avoir une version différente) :

```
lambda@mv-snmp-dd:~$ sudo --version
Sudo version 1.9.13p3
La version du greffon de politique de sudoers est 1.9.13p3
La version de la grammaire du fichier sudoers est 50
Sudoers I/O plugin version 1.9.13p3
Sudoers audit plugin version 1.9.13p3
lambda@mv-snmp-dd:~$
```

Si besoin, en tant que « root » :

↗️ Installer le paquet sudo.

A.2. UTILISATION DE BASE DE « SUDO »

Sudo permet d'exécuter une commande en prenant l'identité d'un autre utilisateur. Si on ne précise pas quel utilisateur ou quel groupe (avec les options –u ou –g), ce sera l'identité de root qui sera utilisée.

Si le droit d'exécuter la commande via sudo nous est accordé, il faut entrer son mot de passe – bien qu'on soit déjà connecté. Ceci pour éviter l'utilisation d'une console laissée ouverte (par négligence) pour effectuer une commande nécessitant des privilèges.

Si l'authentification est réussie, elle restera valable pendant 5 minutes (temps par défaut qui peut être variable selon les distributions et redéfini dans le fichier de configuration sudoers).

A.3. PETITE EXPLICATION POUR LES EXPERTS

Comment fonctionne « sudo » ? En fait cette commande a le bit setUID activé, comme on peut le voir ci-dessous :

```
lambda@mv-tpsudo-dd:/usr/bin$ ls -al /usr/bin/sudo
-rwsr-xr-x 1 root root 281624 27 juin   2023 /usr/bin/sudo
lambda@mv-tpsudo-dd:/usr/bin$
```

- Le droit ‘s’ positionné à la place du ‘x’ de la partie « utilisateur » des droits correspond à l’activation du bit setUID.
- Quand la commande sudo est exécutée, **ce sont les droits du propriétaire de la commande qui sont utilisés et non ceux de celui qui l’exécute**. Dans ce cas, le propriétaire de la commande est root (comme on le voit dans les informations à la suite des droits).

A.4. LE FICHIER DE CONFIGURATION « SUDOERS »

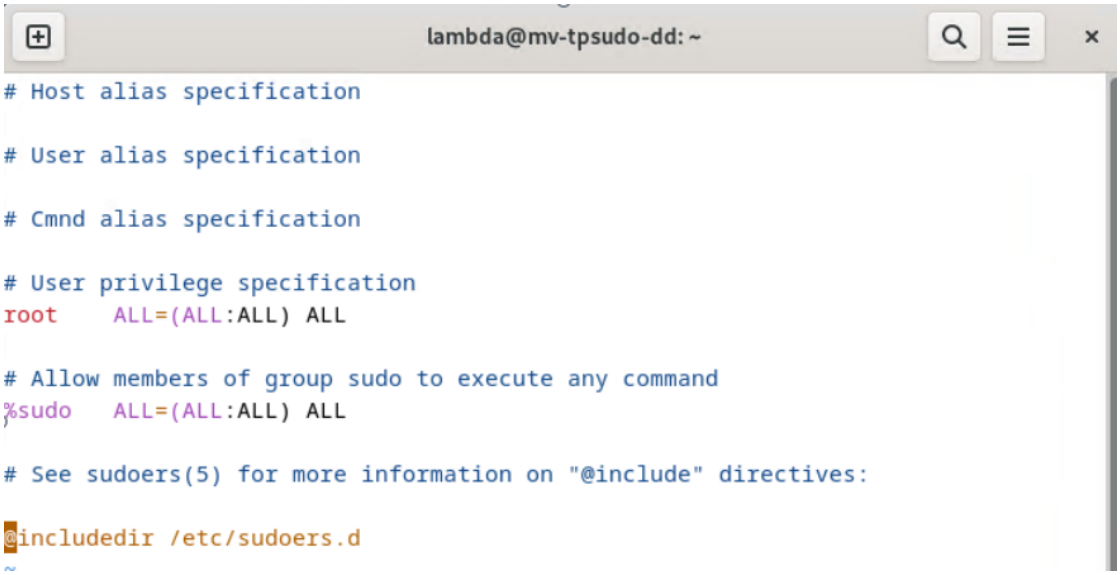
En fait, il faut plutôt parler des fichiers « sudoers » au pluriel, car il y a le fichier /etc/sudoers, mais aussi tous les éventuels fichiers ajoutés dans le répertoire /etc/sudoers.d/.

```
lambda@mv-tpsudo-dd:~$ ls -al /etc/sudoers
-r--r----- 1 root root 1714 30 juin  07:55 /etc/sudoers
lambda@mv-tpsudo-dd:~$
```

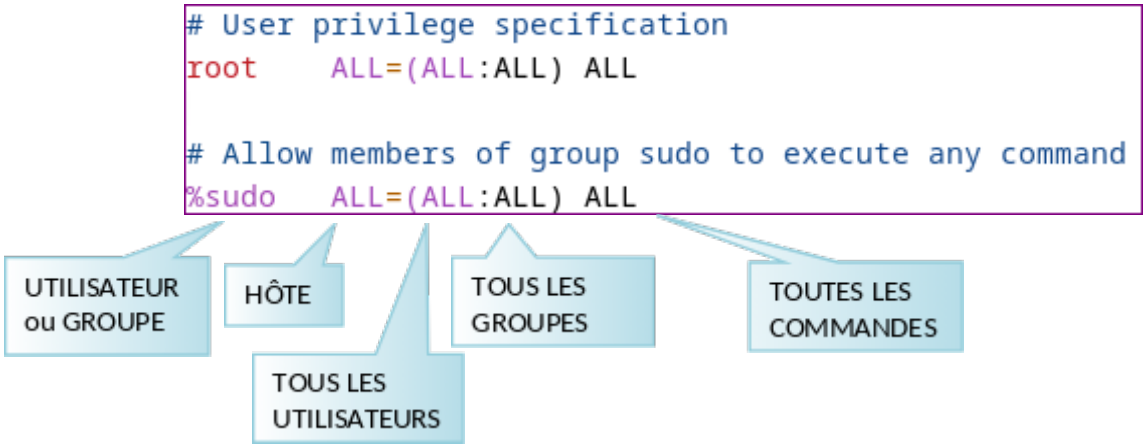
```
lambda@mv-tpsudo-dd:~$ ls -al /etc/sudoers.d/
total 20
drwxr-xr-x   2 root root  4096 23 oct.  11:40 .
drwxr-xr-x 125 root root 12288 23 oct.  12:40 ..
-r--r-----   1 root root  1068 30 juin  07:55 README
lambda@mv-tpsudo-dd:~$
```

i Le répertoire sudoers.d ne contient qu’un fichier README initialement, mais des parties de configuration peuvent être ajoutées dans ce répertoire. Le contenu de ces fichiers sera lu comme complément de la configuration du fichier /etc/sudoers

L’inclusion de ce dossier figure d’ailleurs à la fin du fichier de configuration /etc/sudoers (Il faut les privilèges root pour visualiser / éditer le contenu de ce fichier) :



Mais les lignes qui nous intéressent particulièrement sont les deux lignes précédentes :



Comme l'indique le commentaire, la dernière commande autorise les membres du groupe 'sudo' d'exécuter n'importe quelle commande en prenant les droits de n'importe quel utilisateur ou groupe.

Quelques précisions :

- UTILISATEUR ou GROUPE : On utilise simplement l'identifiant de l'utilisateur (par exemple root sur la 1ère ligne) ou bien, s'il s'agit d'un groupe, on préfixe son identifiant par %.
  Attention, un seul utilisateur ou groupe par ligne !
- HÔTE : On peut indiquer le nom de la machine à laquelle s'applique cette règle. ALL s'applique n'importe quelle machine si le fichier est partagé sur plusieurs systèmes via un service comme LDAP.
- TOUS LES UTILISATEURS : permet de préciser l'utilisateur dont on prend les droits ; en spécifiant ALL, on englobe tous les utilisateurs, y compris l'utilisateur 'root'.

  NB : Sans précision dans les arguments de la commande, c'est en tant que root qu'on exécute la commande qui suit sudo.
- TOUS LES GROUPES : même principe que pour les utilisateurs, permet de préciser éventuellement le groupe dont on peut prendre les droits.
- TOUTES LES COMMANDES : permet de préciser la ou les commandes que l'utilisateur (ou le groupe) spécifié en début de ligne peut exécuter sur la machine.

  Utiliser la virgule comme séparateur s'il y a plusieurs commandes.

A.5. FONCTIONNEMENT POUR UN UTILISATEUR 'LAMBDA'

Sur une machine Linux Debian, lors de l'installation par défaut avec une interface graphique, deux comptes sont créés :

- Un compte administrateur : root – c'est le « super-utilisateur » de la machine, avec tous les droits.
- Un compte 'lambda' qui n'a pas de droit particulier, mais qui peut se connecter à la machine, possède un répertoire 'home', etc.

Si vous ne disposez pas d'un tel compte (peut importe son nom), créer un nouveau compte qui aura 'lambda' comme identifiant, avec le nom « Laurent AMBDA ».

- Créer éventuellement le compte lambda/pass$$lambda via la commande adduser lambda.
- Se connecter avec ce compte.

La commande « id » permet d'afficher les groupes auxquels appartient un utilisateur (les numéros d'id et les groupes peuvent être différents) :

```
lambda@mv-tpsudo-dd:~$ id
uid=1000(lambda) gid=1000(lambda) groupes=1000(lambda),24(cdrom),25(floppy),29(audio),30(dip),
44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner)
```

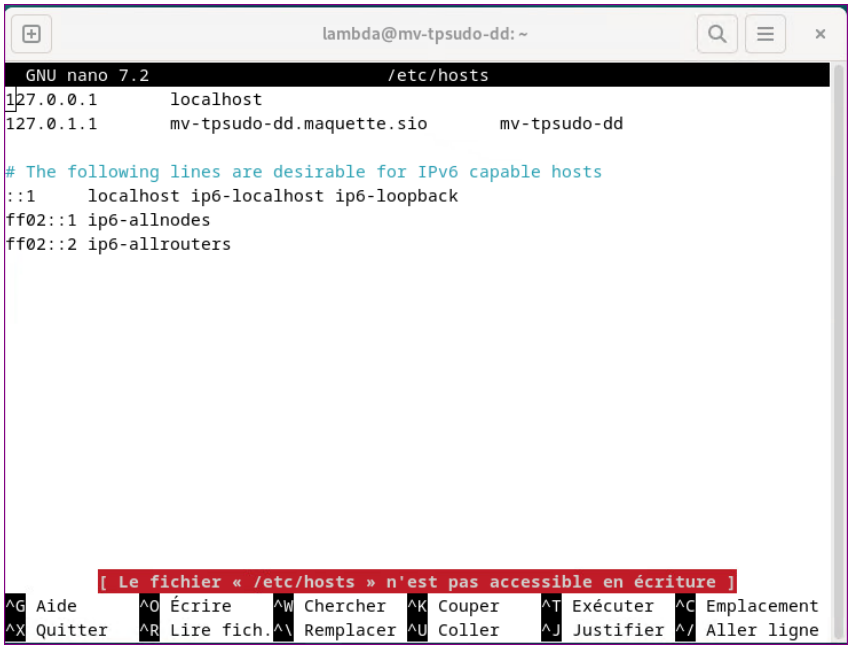
On peut aussi utiliser la commande « groups » :

```
lambda@mv-tp-dd:~$ groups lambda
lambda : lambda cdrom floppy audio dip video plugdev users netdev bluetooth lpadmin scanner
lambda@mv-tp-dd:~$
```

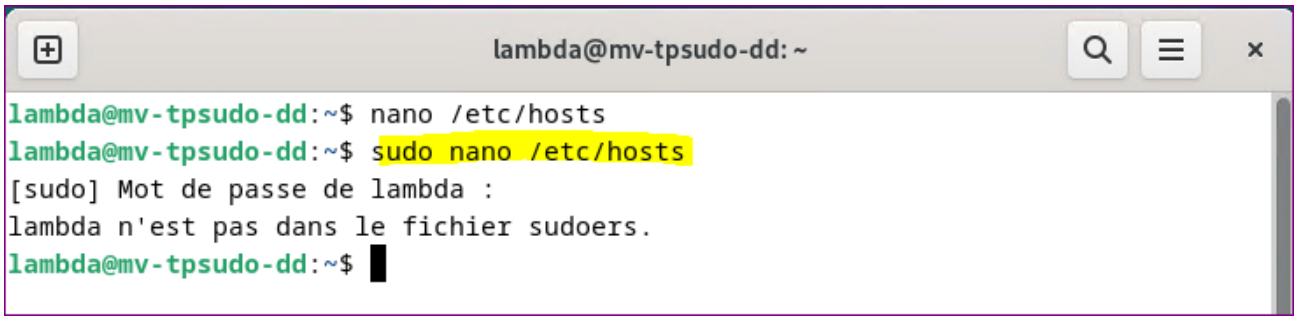
- Lambda n'appartient à aucun groupe ayant des privilèges particuliers, en tous cas pas au groupe sudo.

Vérifions le fonctionnement de ***sudo*** pour cet utilisateur, en précisant qu'aucune configuration n'a été effectuée à ce sujet pour l'instant pour cet utilisateur.

Si cet utilisateur tente par exemple d'ouvrir un fichier système, ici avec l'éditeur nano, au mieux l'accès lui est possible en lecture seule :



S'il tente d'exécuter la commande en préfixant par la commande sudo, son mot de passe lui est demandé, mais ensuite il est indiqué que l'utilisateur n'a aucune directive qui le concerne dans le fichier sudoers :



En effet il n'y a pas d'instruction le concernant, et il ne fait pas non plus partie du **groupe sudo**, ce qui lui donnerait les droits indiqués par la ligne déjà étudiée :

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Nous allons créer un autre utilisateur et l'ajouter au groupe sudo pour étudier le fonctionnement de cet outil.

Dans une console où nous avons les droits root :

- Créer cet utilisateur.

```
root@mv-tpsudo-dd:~# adduser superman
Ajout de l'utilisateur « superman » ...
Ajout du nouveau groupe « superman » (1001) ...
Ajout du nouvel utilisateur « superman » (1001) avec le groupe « superman » (1001) ...
Création du répertoire personnel « /home/superman » ...
Copie des fichiers depuis « /etc/skel » ...
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 superman
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
        NOM []: SUPER-HEROS
        Numéro de chambre []:
        Téléphone professionnel []:
        Téléphone personnel []:
        Autre []:
Cette information est-elle correcte ? [O/n]o
Ajout du nouvel utilisateur « superman » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « superman » au groupe « users » ...
root@mv-tpsudo-dd:~#
```

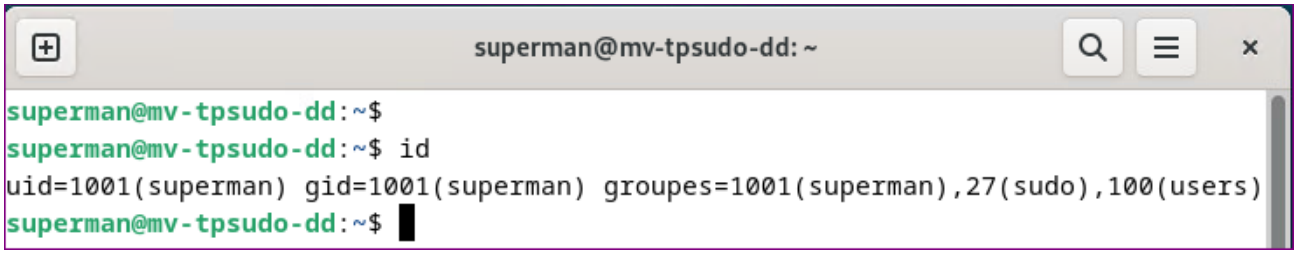
- Ajouter cet utilisateur au group sudo.

```
root@mv-tpsudo-dd:~# adduser superman sudo
Ajout de l'utilisateur « superman » au groupe « sudo » ...
Fait.
root@mv-tpsudo-dd:~#
```

- Vérifier qu'il a bien été ajouté au groupe sudo.

```
root@mv-tpsudo-dd:~# cat /etc/group | grep sudo
sudo:x:27:superman
root@mv-tpsudo-dd:~#
```

- Ouvrir un terminal en tant que ce nouvel utilisateur (superman) :

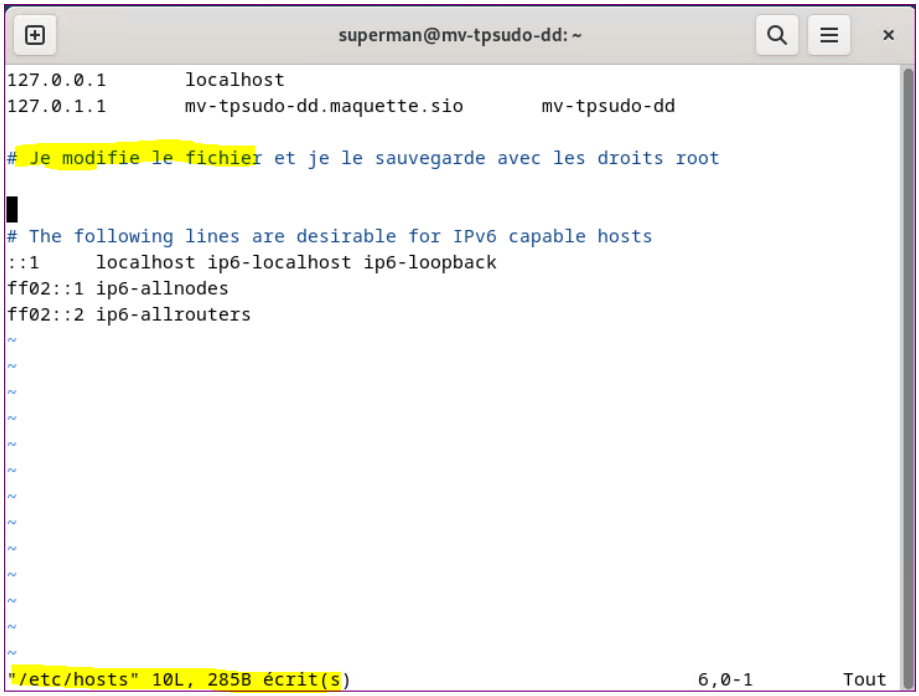


On peut contrôler la liste des groupes auxquels il appartient et constater que sudo fait partie de cette liste.

- Éditer un fichier système comme /etc/hosts en préfixant la commande par sudo.

i On obtient bien les droits 'root' pour faire cette édition, et on peut modifier le fichier.

```
superman@mv-tpsudo-dd:~$ sudo vi /etc/hosts
[sudo] Mot de passe de superman :
```



↗️ Vérifier que cette édition avec privilège a bien été enregistrée dans les journaux, à des fins de traçabilité.

```
root@mv-tp-dd:~# journalctl | grep sudo
```

Nous pouvons par exemple utiliser cette combinaison de commandes pour trouver les lignes qui contiennent le mot « sudo » dans le journal.

On trouve effectivement la ligne cherchée, en fin de journal si ‘superman‘ vient de faire son opération de modification du fichier hosts :

```
nov. 20 18:22:28 mv-tp-dd sudo[2535]: superman : TTY=pts/1 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
nov. 20 18:22:28 mv-tp-dd sudo[2535]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1001)
nov. 20 18:22:32 mv-tp-dd sudo[2535]: pam_unix(sudo:session): session closed for user root
```

ℹ️ Nous constatons bien la trace de l’exécution de la commande par ‘superman’, suivie par une ouverture d’une session ‘root’, limitée à l’exécution de la commande, puisqu’elle se referme dès que l’utilisateur aura quitté l’édition du fichier.

On peut aussi utiliser les possibilités de la commande journalctl pour une extraction plus précise du journal. Par exemple l’argument _COMM permet de connaître précisément tous les événements liés à une commande qui ont été enregistrées dans le journal.

```
root@mv-tpsudo-dd:~# journalctl _COMM=sudo | grep superman
oct. 23 17:15:10 mv-tpsudo-dd sudo[6711]:   lambda : user NOT in sudoers ; TTY=pts/1 ; PWD=/home/lambda ; USER=root ; COMMAND=superman
oct. 23 17:15:57 mv-tpsudo-dd sudo[6774]: superman : TTY=pts/1 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: superman : TTY=tty3 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: pam_unix(sudo:session): session opened for user root(uid=0) by superman(uid=1002)
oct. 28 12:04:55 mv-tpsudo-dd sudo[30441]: pam_unix(sudo:session): session opened for user root(uid=0) by superman(uid=0)
root@mv-tpsudo-dd:~#
```

Ci-dessus, on cherche les lignes où la commande sudo a été exécutée par superman (en tous cas les lignes qui concernent l’exécution de sudo et qui contiennent l’expression « superman ».

On constate que superman a utilisé ‘sudo’ à plusieurs reprises le 23 octobre et une fois le 28 octobre.

On peut utiliser aussi directement plusieurs arguments dans la commande **journalctl** plutôt que d’utiliser la commande **grep**.

```
root@mv-tpsudo-dd:~# journalctl _COMM=sudo _UID=$(id -u superman)
oct. 23 17:15:57 mv-tpsudo-dd sudo[6774]: superman : TTY=pts/1 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:15:57 mv-tpsudo-dd sudo[6774]: pam_unix(sudo:session): session opened for user root(uid=0) by lambda(uid=1002)
oct. 23 17:16:15 mv-tpsudo-dd sudo[6774]: pam_unix(sudo:session): session closed for user root
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: superman : TTY=tty3 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: pam_unix(sudo:session): session opened for user root(uid=0) by superman(uid=1002)
oct. 23 17:18:34 mv-tpsudo-dd sudo[6823]: pam_unix(sudo:session): session closed for user root
root@mv-tpsudo-dd:~#
```

Cette commande affiche les lignes concernant la commande ‘sudo’ pour l’utilisateur dont l’UID est renvoyé par la commande ‘id -u superman’, autrement dont l’UID est celui de l’utilisateur superman.

On aurait pu obtenir le même résultat en tapant successivement les 2 commandes :

```
root@mv-tpsudo-dd:~# id -u superman
1002
root@mv-tpsudo-dd:~#
root@mv-tpsudo-dd:~# journalctl _COMM=sudo _UID=1002
oct. 23 17:15:57 mv-tpsudo-dd sudo[6774]: superman : TTY=pts/1 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:15:57 mv-tpsudo-dd sudo[6774]: pam_unix(sudo:session): session opened for user root(uid=0) by lambda(uid=1002)
oct. 23 17:16:15 mv-tpsudo-dd sudo[6774]: pam_unix(sudo:session): session closed for user root
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: superman : TTY=tty3 ; PWD=/home/superman ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
oct. 23 17:18:05 mv-tpsudo-dd sudo[6823]: pam_unix(sudo:session): session opened for user root(uid=0) by superman(uid=1002)
oct. 23 17:18:34 mv-tpsudo-dd sudo[6823]: pam_unix(sudo:session): session closed for user root
root@mv-tpsudo-dd:~#
```

Sans configuration ‘superman’ a tous les droits.

Nous constaterons qu’il faut être très vigilant dans la configuration afin que ‘superman’ ne détourne l’usage du sudo pour effectuer des commandes avec les privilèges ‘root’ sans habilitations.

B. MISE EN PLACE DE DROITS SPÉCIFIQUES

B.1. UTILISATION DE VISUDO POUR L'ÉDITION DU FICHIER /ETC/SUDOERS

Pour modifier le fichier /etc/sudoers (ou tout autre fichier complémentaire à ce fichier comme nous le verrons plus tard), il est fortement recommandé d'utiliser la commande visudo.

Plusieurs bonnes raisons à cette recommandation, qui est plus qu'une bonne pratique :

- Cet outil vérifie la syntaxe et signalera les erreurs de syntaxe lors de la tentative d'enregistrement, et même certaines erreurs de programmation (cf. explication dans la dernière partie). En effet, une mauvaise syntaxe risque de générer un fonctionnement défaillant dans lequel il sera impossible d'obtenir des privilèges élevés.
- Cet outil permet une édition sécurisée en exclusivité : il est toujours préférable d'éviter une intervention simultanée de plusieurs utilisateurs (administrateurs) sur ce fichier.

Par défaut, visudo (sans argument, sans paramètre) ouvre le fichier /etc/sudoers avec un éditeur de texte, soit vi, soit nano, selon les distributions et les versions.

On peut changer l'éditeur par défaut, par exemple de la manière suivante :

```
superman@mv-tp-dd:~$ sudo update-alternatives --config editor
[sudo] Mot de passe de superman :
Il existe 3 choix pour l'alternative editor (qui fournit /usr/bin/editor).

  Sélection   Chemin              Priorité  État
------------------------------------------------------------
* 0            /bin/nano            40        mode automatique
  1            /bin/nano            40        mode manuel
  2            /usr/bin/vim.basic   30        mode manuel
  3            /usr/bin/vim.tiny    15        mode manuel

Appuyez sur <enter> pour conserver le choix actuel [*], ou tapez le numéro de sélection :
```

- On choisit le n° qui correspond au choix que l'on souhaite appliquer.

L'utilisation de visudo donne donc accès au fichier /etc/sudoers :

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"
"/etc/sudoers.tmp" 56L, 1818B                                 1,1           Top
```

- Nous constatons que l'on travaille sur un fichier temporaire, jusqu'à ce que l'on effectue la sauvegarde qui vérifiera la syntaxe.

```
superman@mv-tp-dd:~$ sudo visudo
[sudo] Mot de passe de superman :
visudo: /etc/sudoers n'est pas disponible, réessayez plus tard
superman@mv-tp-dd:~$
```

- Si le fichier est déjà en cours d'édition par un utilisateur, un message d'indisponibilité le signalera.

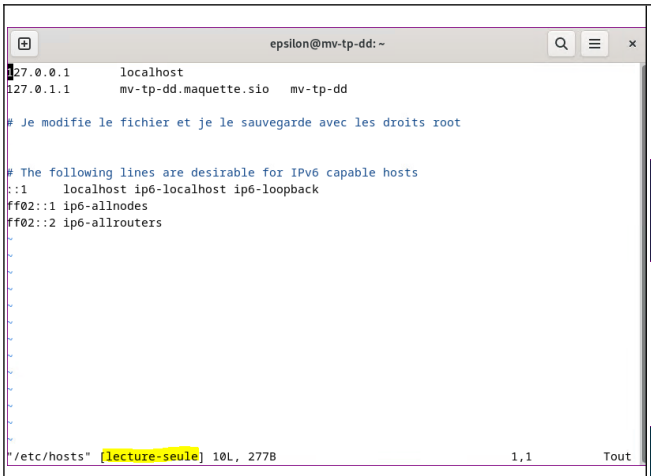
```
superman@mv-tp-dd:~$ sudo visudo
/etc/sudoers:51:47: il est fait mention de Host_Alias « ALLI » alors qu'il n'a pas été défini
Attention : /etc/sudoers:43:12: User_Alias « TOTO » n'est pas utilisé
superman@mv-tp-dd:~$ sudo visudo
/etc/sudoers:45:11: erreur de syntaxe
User_Cmnd = vi
          ^
Et maintenant ?
Les options sont :
  (e)dition du fichier sudoers (de nouveau)
  e(x)it sans sauvegarde des modifications apportées au fichier sudoers
  (Q)uitter et sauvegarder les modifications apportées au fichier sudoers (DANGER!)

Et maintenant ?
```

- Si une erreur de programmation ou de syntaxe est détectée à l'enregistrement, elle sera signalée.

B.2. CRÉATION D'UN UTILISATEUR PUIS ATTRIBUTION DES DROITS SPÉCIFIQUES

↗️ Créer un nouvel utilisateur 'epsilon', que nous ne rattacherons pas au groupe sudo, mais à qui il sera accordé des autorisations spécifiques. *Écrire ci-dessous la commande et la sortie d'écran obtenue.*

Dans l'état actuel, c'est-à-dire avant de lui donner des droits particuliers via le fichier de configuration sudoers, l'utilisateur « epsilon » peut consulter le fichier /etc/hosts, mais pas le modifier.



- Nous constatons la notification [lecture-seule]

- S'il tente un « sudo », l'accès au fichier ne lui est pas accordé car il n'y a pas de directive le concernant dans le fichier sudoers.

```
epsilon@mv-tp-dd:~$ sudo vi /etc/hosts
[sudo] Mot de passe de epsilon :
epsilon n'est pas dans le fichier sudoers.
epsilon@mv-tp-dd:~$
```

- On notera que l'échec est inscrit dans le journal des événements, puisqu'il s'agit d'une tentative d'exécution en mode privilégié.

```
nov. 20 19:14:47 mv-tp-dd sudo[2722]:  epsilon : user NOT in sudoers ; TTY=pts/0 ;
PWD=/home/epsilon ; USER=root ; COMMAND=/usr/bin/vi /etc/hosts
root@mv-tp-dd:~#
```

NB: Nous avons utilisé la commande : journalctl | grep sudo

Ajouter la possibilité pour epsilon de modifier le fichier /etc/hosts dans le fichier de configuration sudoers.

Il faut préciser le chemin complet de la commande. Si on ne le connaît pas, on peut l'obtenir en utilisant la commande which :

```
root@mv-tp-dd:~# which vi nano
/usr/bin/vi
/usr/bin/nano
root@mv-tp-dd:~#
```

Nous avons récupéré le chemin de deux commandes différentes, vi et nano.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
epsilon ALL=(ALL) /usr/bin/vi /etc/hosts
```

Epsilon pourra-t-il exécuter nano au lieu de vi ?

Vérifier qu'epsilon peut désormais modifier le fichier /etc/hosts s'il utilise la commande sudo.

C. CONTOURNEMENT DE LA POLITIQUE DE SÉCURITÉ

Nous allons mettre en évidence qu'il est possible de contourner la politique de sécurité mise en place pour l'utilisateur 'epsilon'.

Normalement, ce dernier ne peut qu'éditer et modifier le fichier /etc/hosts, et aura pour toute autre commande [réservée à un administrateur] un message d'erreur refusant l'opération.

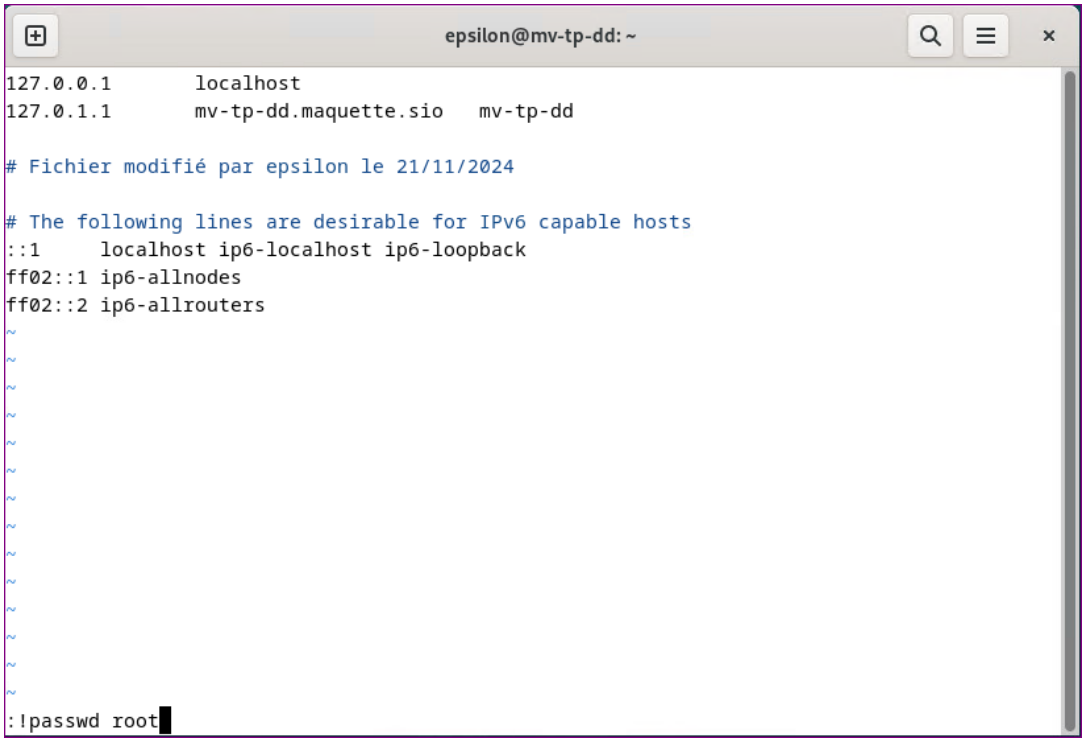
Voici 3 exemples de commandes qu'il ne peut pas exécuter :

```
epsilon@mv-tp-dd:~$ sudo vi /etc/hostname
Sorry, user epsilon is not allowed to execute '/usr/bin/vi /etc/hostname' as root on mv-t
p-dd.maquette.sio.
epsilon@mv-tp-dd:~$
epsilon@mv-tp-dd:~$ sudo vi /etc/network/interfaces
Sorry, user epsilon is not allowed to execute '/usr/bin/vi /etc/network/interfaces' as ro
ot on mv-tp-dd.maquette.sio.
epsilon@mv-tp-dd:~$
epsilon@mv-tp-dd:~$ passwd lambda
passwd : Vous ne devriez pas voir ou modifier l'information du mot de passe pour lambda.
epsilon@mv-tp-dd:~$
epsilon@mv-tp-dd:~$ sudo passwd lambda
Sorry, user epsilon is not allowed to execute '/usr/bin/passwd lambda' as root on mv-tp-d
d.maquette.sio.
epsilon@mv-tp-dd:~$
```

- Il ne peut pas éditer le fichier /etc/hostname, ni le fichier /etc/network/interfaces.
- Il ne peut pas changer le mot de passe d'un autre utilisateur, même s'il préfixe la commande par sudo !

Mais nous avons vu précédemment que pendant l'édition du fichier /etc/hosts l'utilisateur dispose des droits « root » pour pouvoir modifier ce fichier. Une session s'ouvre donc lorsque la commande « sudo vi /etc/hosts » est exécutée, et ne se ferme que lorsque l'édition de ce fichier est terminée, enregistrée ou abandonnée.

Mais, dans l'éditeur vi, il est possible de taper une commande dans l'interpréteur shell en utilisant le « :! » Comme indiqué dans la fenêtre ci-dessous :



Nous retournons transitoirement à l'invite de commande où nous avons tapé la commande « sudo vi /etc/hosts ».

↗️ Modifier le mot de passe de 'root' en utilisant **l'élévation de privilèges** momentanée, permise par l'utilisation de sudo.

```
epsilon@mv-tp-dd:~$ sudo vi /etc/hosts

Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès

Appuyez sur ENTRÉE ou tapez une commande pour continuer
```

La modification de mot de passe s'enchaîne sans souci !

On voit bien la dangerosité liée à l'exploitation d'une élévation de privilèges, bien connue dans le domaine de la cybersécurité : un utilisateur qui ne devrait pas avoir certains droits (ici celui de modifier le mot de passe root) les obtient grâce à une élévation de privilèges ponctuelle, nécessaire pour pouvoir modifier le contenu d'un fichier.

On note également que cette modification du mot de passe aurait des implications pour la suite :

- l'administrateur légitime ne pourrait plus se connecter
- en revanche l'utilisateur malveillant serait en mesure de se connecter en tant que root et perpétrer d'autres forfaits ...

On revient ensuite dans l'éditeur de fichier. Si on ferme le fichier et qu'on consulte le journal en lien avec sudo, on ne voit que la commande d'édition, et pas le changement de mot de passe.

```
nov. 21 11:20:56 mv-tp-dd sudo[4251]:  epsilon : command not allowed ; TTY=pts/0
 ; PWD=/home/epsilon ; USER=root ; COMMAND=/usr/bin/vi /etc/hostname
nov. 21 11:21:18 mv-tp-dd sudo[4263]:  epsilon : command not allowed ; TTY=pts/0
 ; PWD=/home/epsilon ; USER=root ; COMMAND=/usr/bin/vi /etc/network/interfaces
nov. 21 11:22:15 mv-tp-dd sudo[4271]:  epsilon : command not allowed ; TTY=pts/0
 ; PWD=/home/epsilon ; USER=root ; COMMAND=/usr/bin/passwd lambda
nov. 21 11:26:23 mv-tp-dd sudo[4274]:  epsilon : TTY=pts/0 ; PWD=/home/epsilon ;
 USER=root ; COMMAND=/usr/bin/vi /etc/hosts
nov. 21 11:26:23 mv-tp-dd sudo[4274]: pam_unix(sudo:session): session opened for
 user root(uid=0) by (uid=1002)
nov. 21 11:31:25 mv-tp-dd sudo[4274]: pam_unix(sudo:session): session closed for
 user root
root@mv-tp-dd:~#
```

- Mais on peut avoir une idée du temps écoulé entre le début de l'édition et la fermeture puisque qu'on a le début et la fin de connexion en tant que root : il s'est écoulé 5 minutes
- On note dans cette même copie d'écran que pour les tentatives d'exécution de commandes autres que celles permises depuis l'invite de commande, la consignation de ces tentatives est bien réalisée.
- En revanche on ne sait pas qu'entre l'ouverture et la fermeture du fichier autorisé pour epsilon, il y a eu un changement de mot de passe en profitant de l'utilisation de sudo.

Si on consulte les journaux, on retrouvera le changement de mot de passe de root comme événement consigné à 11h27, donc dans l'intervalle de temps où epsilon a bénéficié des droits root, mais on ne sait pas qui l'a changé finalement :

journalctl | grep passwd

```
nov. 21 11:21:35 mv-tp-dd passwd[4269]: passwd: can't view or modify password in
formation for lambda
nov. 21 11:22:15 mv-tp-dd sudo[4271]:  epsilon : command not allowed ; TTY=pts/0
 ; PWD=/home/epsilon ; USER=root ; COMMAND=/usr/bin/passwd lambda
nov. 21 11:27:50 mv-tp-dd passwd[4281]: pam_unix(passwd:chauthtok): password cha
nged for root
nov. 21 11:27:50 mv-tp-dd passwd[4281]: gkr-pam: couldn't update the login keyri
ng password: no old password was entered
```

On peut aussi utiliser la syntaxe de journalctl vue précédemment, avec l'argument _COMM :

```
root@mv-tpsudo-dd:~# journalctl _COMM=passwd
oct. 23 12:40:33 mv-tpsudo-dd passwd[5431]: pam_unix(passwd:chauthtok): password changed for delta
oct. 23 12:40:33 mv-tpsudo-dd passwd[5431]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 23 17:11:19 mv-tpsudo-dd passwd[6552]: pam_unix(passwd:chauthtok): password changed for superman
oct. 23 17:11:19 mv-tpsudo-dd passwd[6552]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 23 18:13:21 mv-tpsudo-dd passwd[6979]: pam_unix(passwd:chauthtok): password changed for espsilon
oct. 23 18:13:21 mv-tpsudo-dd passwd[6979]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 23 18:15:34 mv-tpsudo-dd passwd[7148]: pam_unix(passwd:chauthtok): password changed for epsilon
oct. 23 18:15:34 mv-tpsudo-dd passwd[7148]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 23 18:52:33 mv-tpsudo-dd passwd[13814]: can't view or modify password information for lambda
oct. 23 18:56:34 mv-tpsudo-dd passwd[13842]: pam_unix(passwd:chauthtok): password changed for root
oct. 23 18:56:34 mv-tpsudo-dd passwd[13842]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 23 18:57:36 mv-tpsudo-dd passwd[13952]: pam_unix(passwd:chauthtok): password changed for root
oct. 23 18:57:36 mv-tpsudo-dd passwd[13952]: gkr-pam: couldn't update the login keyring password: no old password was entered
oct. 28 12:49:36 mv-tpsudo-dd passwd[31434]: pam_unix(passwd:chauthtok): password changed for root
oct. 28 12:49:36 mv-tpsudo-dd passwd[31434]: gkr-pam: couldn't update the login keyring password: no old password was entered
root@mv-tpsudo-dd:~#
```

On perçoit au passage que l'ancien mot de passe n'a pas été demandé, donc le mot de passe a bien été réinitialisé, avec les droits 'root', mais ce n'est pas 'root' lui-même qui l'a changé.

En effet, si un utilisateur change son propre mot de passe, l'ancien lui est demandé pour sécuriser la modification :

```
epsilon@mv-tp-dd:~$ passwd
Changement du mot de passe pour epsilon.
Mot de passe actuel :
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Vous devez choisir un mot de passe plus long.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
epsilon@mv-tp-dd:~$
```

On trouve alors une trace dans le journal similaire à celle-ci :

```
oct. 28 12:53:58 mv-tpsudo-dd passwd[31800]: pam_unix(passwd:chauthtok): password changed for epsilon
oct. 28 12:53:58 mv-tpsudo-dd passwd[31800]: gkr-pam: changed password for login keyring
```

D. SÉCURISATION POUR ÉVITER LE CONTOURNEMENT

Il est possible d'utiliser une option quand on autorise une commande pour un utilisateur (ou un groupe) avec élévation de privilèges. **Il s'agit de l'option NOEXEC.**

L'option NOEXEC empêchera le contournement de la politique de sécurité, en empêchant le lancement d'un environnement shell d'échappement (invite de commande) comme nous venons de le faire.

La ligne ajoutée précédemment dans le fichier /etc/sudoers devient :

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
epsilon ALL=(ALL) NOEXEC:/usr/bin/vi /etc/hosts
```

↗ Vérifier l'efficacité de cette option : quel est le message d'erreur ?

ℹ À noter que la même problématique existe avec 'nano' : il est possible d'ouvrir une session shell root via nano (^R^X pour exécuter une commande).

D'autres solutions existent comme celle de ne permettre que la commande « sudoedit » qui est spécialement conçue pour éditer un fichier dont la modification nécessite des privilèges..

Il s'agit de la meilleure pratique :

Sudoedit permet d'éditer un fichier qui nécessite des privilèges élevés sans lancer l'éditeur lui-même en tant que root. Par conséquent seules les commandes autorisées habituellement pour l'utilisateur seront acceptées.

ℹ La commande « sudoedit » est en fait une contraction de sudo et edit (qui fait appel à la commande /usr/bin/editor) ; c'est la raison pour laquelle cette commande ne nécessite pas le préfixage par sudo.

ℹ La commande sudoedit n'est pas un binaire distinct mais un mode de fonctionnement intégré à sudo : il ne faut pas dans ce cas écrire le chemin absolu de la commande.

E. UTILISATION DES EXCLUSIONS POUR LIMITER L'USAGE D'UNE COMMANDE

Dans une règle, il est possible de limiter le champ d'action de cette règle en interdisant l'utilisation de certains arguments.

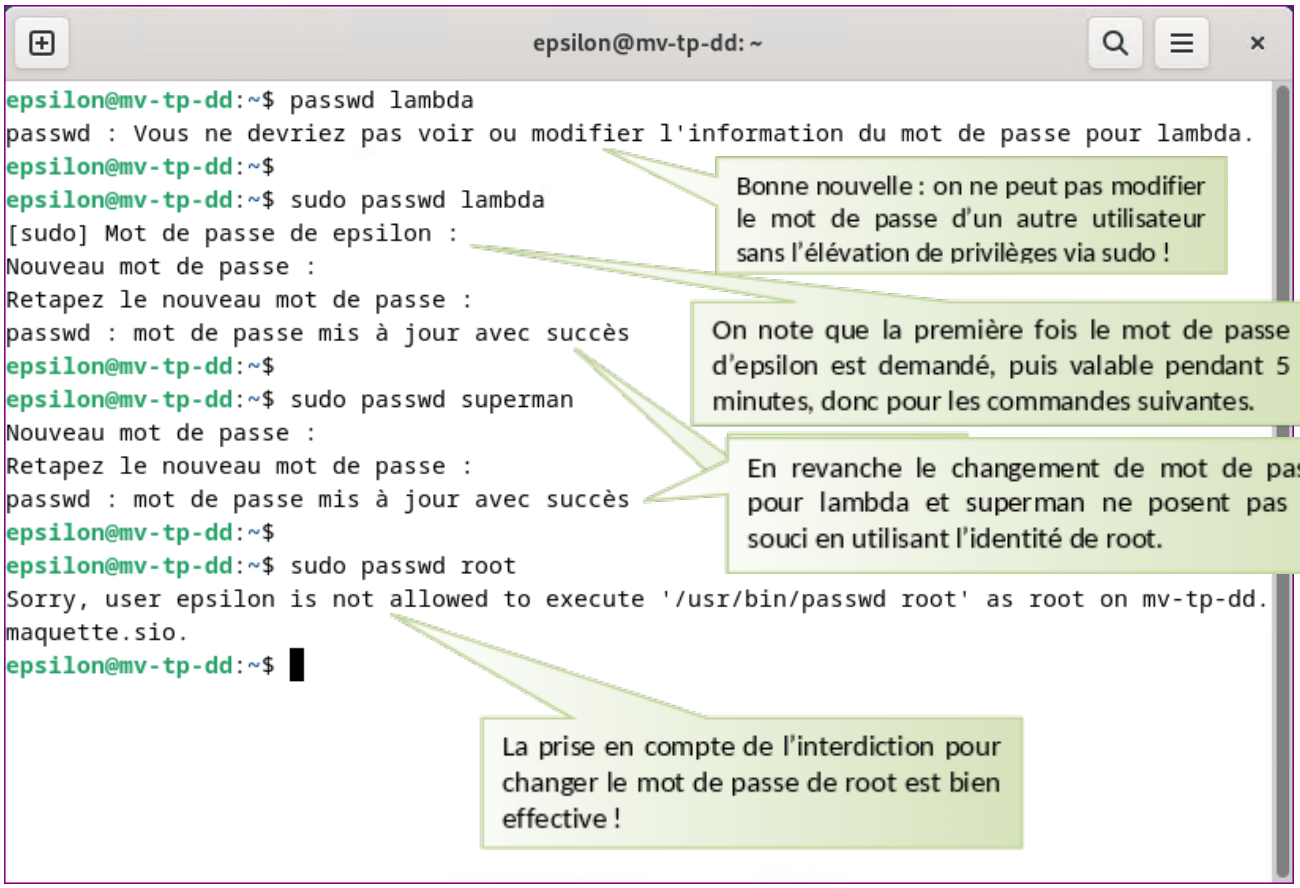
Imaginons par exemple que notre utilisateur « epsilon » soit autorisé à changer le mot de passe d'un autre utilisateur, par exemple celui de lambda, mais en aucun cas celui de root.

La règle pourrait s'écrire ainsi : epsilon ALL=(ALL) /usr/bin/passwd,!/usr/bin/passwd root

Notre liste de spécifications, concernant les privilèges utilisateurs, deviendrait :

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
epsilon ALL=(ALL) NOEXEC:/usr/bin/vi /etc/hosts
epsilon ALL=(ALL) /usr/bin/passwd,!/usr/bin/passwd root
```

↗ Vérifier la bonne application de cette règle.



À noter qu'il est possible d'affiner la règle avec des expressions régulières.

```
epsilon ALL=(ALL) /usr/bin/passwd, !/usr/bin/passwd root
```

Ici, epsilon peut exécuter passwd avec n'importe quel argument, sauf root.

Problème : epsilon peut toujours exécuter :
sudo passwd --stdin nom_user
sudo passwd --expire nom_user
sudo passwd -l nom_user

Ces options permettent de manipuler des comptes système (mysql, www-data...) ce qui pourrait être une faille.

```
epsilon ALL=(ALL) /usr/bin/passwd [A-Za-z0-9]*, !/usr/bin/passwd root
```

Ici, epsilon ne peut utiliser passwd qu'avec un argument constitué uniquement de lettres et/ou de chiffres.

Vérifier le bon fonctionnement de cette nouvelle règle modifiée. *Écrire les tests effectués et les résultats des tests.*

F. UTILISATION DES ALIAS ET CRÉATION D'UN FICHIER COMPLÉMENTAIRE DANS /ETC/SUDOERS.D

Comme on l'a vu dans le fichier de configuration, il est possible de définir des alias, pour les hôtes, pour les utilisateurs/groupes, pour les commandes :

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
epsilon ALL=(ALL) NOEXEC:/usr/bin/vi /etc/hosts

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Cette possibilité peut permettre de simplifier les règles définissant les privilèges accordés à certains utilisateurs pour l'exécution de certaines commandes.

On peut définir ces alias dans le fichier /etc/sudoers, mais nous allons « faire d'une pierre deux coups » en montrant que l'on peut aussi les écrire dans un fichier séparé, dans le répertoire « /etc/sudoers.d ».

Nous imaginons le scénario suivant :

- Les techniciens informatiques doivent avoir un certain nombre d'autorisations sur des fichiers système et des commandes système :
  - Ils doivent pouvoir faire un sudoedit du fichier interfaces, ainsi que du fichier resolv.conf.
  - Ils doivent pouvoir lancer les commandes ifdown et ifup.
- Les utilisateurs concernés par ces droits auront les logins suivants : omicron et omega. Ils seront associés à l'alias utilisateur TECHINFO.
- On utilisera l'alias de commandes NETCDES pour la liste des commandes autorisées.
- Le nom de la machine que vous utilisez fera également l'objet d'un alias, ainsi que la liste des machines (srv1 et srv2) sur lesquelles ces techniciens pourraient intervenir. Cet alias sera nommé SRVINFO.
- Le fichier qui contiendra toutes ces directives se nommera SUDO_TECHINFO.

Il faut donc dérouler les étapes suivantes…

Pour chaque manipulation, vous noterez soigneusement toutes les commandes mises en œuvre, les éléments de configuration ajoutés et/ou modifiés ainsi que les procédures de tests réalisés.

↗️ Créer les utilisateurs.

↗️ Vérifier que ces utilisateurs n'ont pas les autorisations énoncées dans le scénario présenté ci-avant.

Création d'un fichier complémentaire dans /etc/sudoers.d

Le scénario prévoit de mettre les nouvelles directives dans un fichier à part, dans le répertoire prévu à cet effet, à savoir /etc/sudoers.d. Ce fichier doit être nommé SUDO_TECHINFO.

On peut commencer par le créer et vérifier qu'il se trouve bien dans le répertoire /etc/sudoers.d :

```
root@mv-tp-dd:~# touch /etc/sudoers.d/SUDO_TECHINFO
root@mv-tp-dd:~# ls -al /etc/sudoers.d
total 20
drwxr-xr-x   2 root root  4096 21 nov.  20:25 .
drwxr-xr-x 123 root root 12288 21 nov.  20:11 ..
-r--r-----   1 root root  1096 27 juin   2023 README
-rw-r--r--   1 root root     0 21 nov.  20:25 SUDO_TECHINFO
root@mv-tp-dd:~#
```

Puis on peut l'éditer avec notre éditeur préféré, vi ou nano, ou mieux utiliser la commande visudo qui permet de le modifier en mode exclusif, comme il faudrait le faire pour éviter un travail concurrentiel de deux utilisateurs en même temps sur un fichier de configuration de ce type.

Un autre avantage, c'est que cette commande visudo vérifie la syntaxe du contenu et signalera les erreurs éventuelles. C'est très très très utile !

```
root@mv-tp-dd:~# visudo /etc/sudoers.d/SUDO_TECHINFO
```

Voici un exemple de fichier de commandes correct, correspondant au scénario :

```
 GNU nano 7.2                                          /etc/sudoers.d/SUDO_TECHINFO.tmp
# Mise en place des autorisations spécifiques aux techniciens informatiques

# Host alias specification - Attention les noms de machines ne doivent pas avoir de majuscules !

Host_Alias SRVINFO = mv-snmp-dd, srv1, srv2

# User alias specification

User_Alias INFOTECH = omega, omicron

# Cmnd alias specification

Cmnd_Alias NETCDES = sudoedit /etc/network/interfaces, sudoedit /etc/resolv.conf, /sbin/ifdown, /sbin/ifup


# Specification des privileges pour les techniciens informatiques

INFOTECH        SRVINFO = (root) NETCDES
```

Remplacer **mv-snmp-dd** par le nom de votre propre machine.
Attention il s'agit du hostname, défini dans le fichier /etc/hostname.

Si lorsque vous quittez l'édition du fichier, vous avez un message similaire à celui-ci :

```
root@mv-tp-dd:~# visudo /etc/sudoers.d/SUDO_TECHINFO
/etc/sudoers.d/SUDO_TECHINFO:13:22: erreur de syntaxe
Cmnd_Alias NETCDES = NOEXEC: sudoedit /etc/network/interfaces, sudoedit /etc/res
olv.conf, /sbin/ifup,/sbin/ifdown
                    ^~~~~~~
Et maintenant ?
Les options sont :
  (e)dition du fichier sudoers (de nouveau)
  e(x)it sans sauvegarde des modifications apportées au fichier sudoers
  (Q)uitter et sauvegarder les modifications apportées au fichier sudoers (DANGE
R!)

Et maintenant ?
```

C'est qu'il y a des erreurs de syntaxe !

Ci-dessus, ce n'était pas très évident, mais il pointait NOEXEC qui est une option et ne se met pas dans la liste des commandes !

Très très utile !

Dans ce cas de figure, on retourne en édition pour corriger l'erreur.

> ⚠ Si on ne trouve pas l'erreur et qu'on ne veut pas perdre toutes les lignes tapées, on peut aussi mettre en commentaire la ligne qui pose problème.

S'il s'agit d'une simple erreur de programmation supposée, visudo autorise l'enregistrement, mais signale sa suspicion par un message adapté, par exemple un nom de variable déclaré, mais non utilisé.

Exemple

```
/etc/sudoers.d/SUDO_TECHINFO:18:33: il est fait mention de Cmnd_Alias « NETCDE » alors qu'il n'a pas été défini
Attention : /etc/sudoers.d/SUDO_TECHINFO:13:12: Cmnd_Alias « NETCDES » n'est pas utilisé
root@mv-snmp-dd:~# _
```

- Cmd_Alias « NETCDE » non défini.
- Cmd_Alias « NETCDES » défini mais non utilisé.

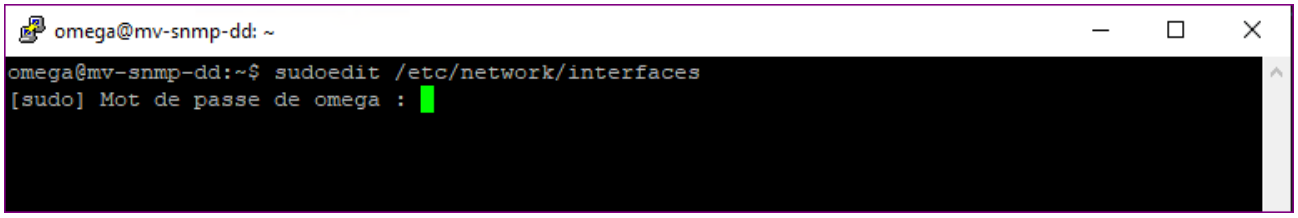
Dans ce cas précis, on avait oublié un S = NETCDES dans la dernière ligne du fichier de configuration :

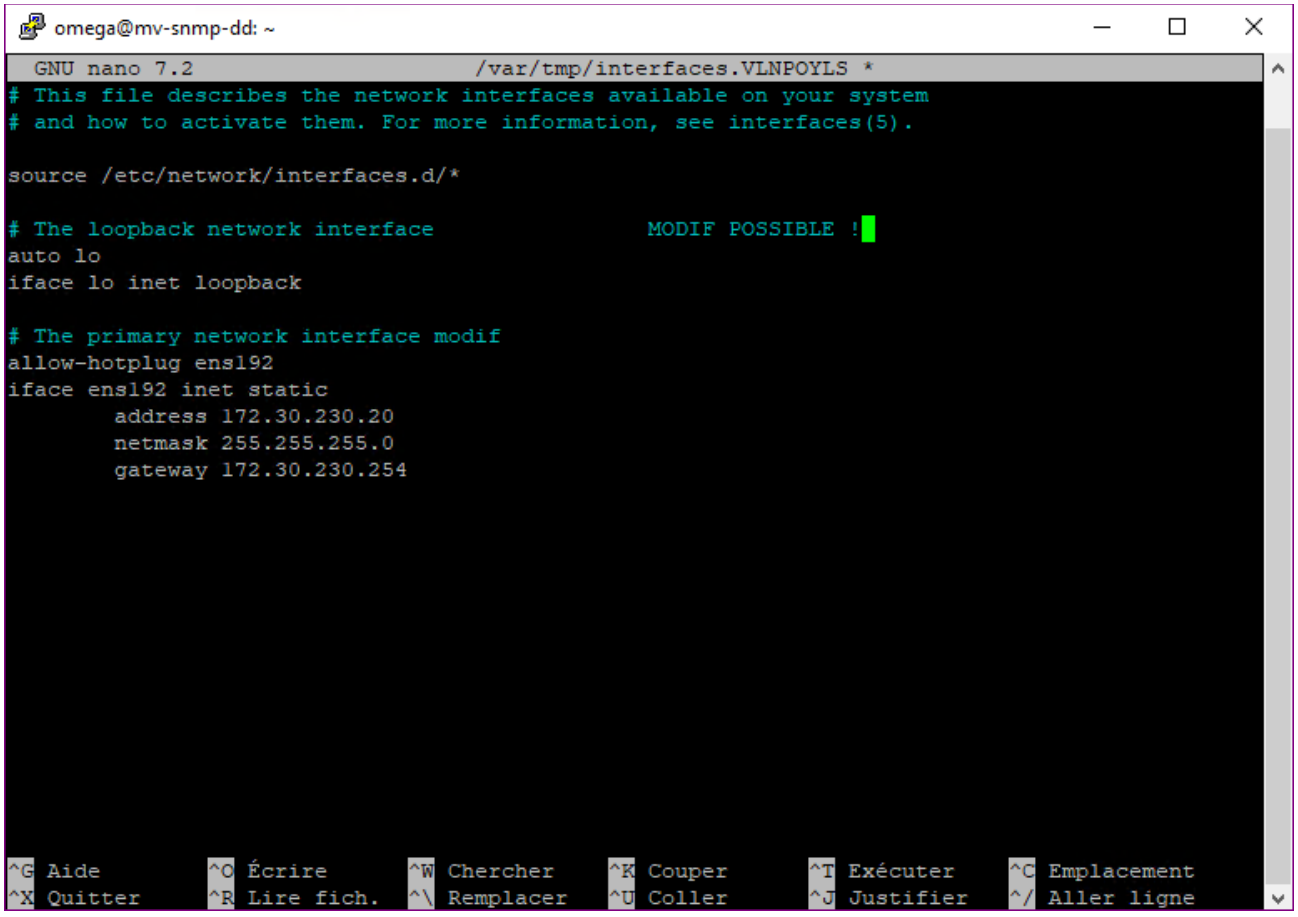
```
  GNU nano 7.2                                          /etc/sudoers.d/SUDO_TECHINFO.tmp *
# Mise en place des autorisations spécifiques aux techniciens informatiques

# Host alias specification - Attention les noms de machines ne doivent pas avoir de majuscules !

Host_Alias SRVINFO = mv-snmp-dd, srv1, srv2

# User alias specification

User_Alias INFOTECH = omega, omicron

# Cmnd alias specification

Cmnd_Alias NETCDES = sudoedit /etc/network/interfaces, sudoedit /etc/resolv.conf, /sbin/ifdown, /sbin/ifup


# Specification des privileges pour les techniciens informatiques

INFOTECH SRVINFO = (root) NETCDE
_
```

En fait c'est une seule erreur, l'oubli du 'S' qui a provoqué les deux messages d'avertissement

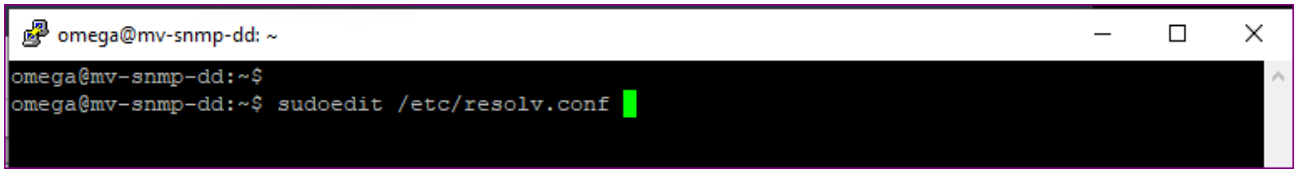
Recette de la mise en place des autorisations

Accès au fichier interfaces (en modification)





Accès au fichier de configuration du résolveur DNS (en modification)



Désactivation et réactivation de l'interface

> ⚠ Il est préférable de faire cela dans la console de votre hyperviseur, car dans une console SSH à distance vous serez déconnecté sans pouvoir réactiver l'interface.

```
omega@mv-snmp-dd:~$ sudo ifdown ens192
[sudo] Mot de passe de omega :
omega@mv-snmp-dd:~$
omega@mv-snmp-dd:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:50:56:bf:b3:a4 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
omega@mv-snmp-dd:~$
```

✅ On vérifie que l'interface est bien désactivée : l'adresse IP n'est plus montrée.

```
omega@mv-snmp-dd:~$ sudo ifup ens192
omega@mv-snmp-dd:~$
omega@mv-snmp-dd:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:bf:b3:a4 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 172.30.230.20/24 brd 172.30.230.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:febf:b3a4/64 scope link
       valid_lft forever preferred_lft forever
omega@mv-snmp-dd:~$
```

✅ On vérifie que l'interface est bien montée et que l'adresse IP est bien attribuée.

Vérification de l'enregistrement des opérations sudo dans le journal

On utilise à nouveau la commande : journalctl | grep sudo

```
nov. 21 22:04:12 mv-snmp-dd sudo[753]:    omega : TTY=tty1 ; PWD=/home/omega ; USER=root ; COMMAND=/sbin/ifdown ens192
nov. 21 22:04:12 mv-snmp-dd sudo[753]: pam_unix(sudo:session): session opened for user root(uid=0) by omega(uid=1002)
nov. 21 22:04:12 mv-snmp-dd sudo[753]: pam_unix(sudo:session): session closed for user root
nov. 21 22:04:29 mv-snmp-dd sudo[772]:    omega : TTY=tty1 ; PWD=/home/omega ; USER=root ; COMMAND=/sbin/ifup ens192
nov. 21 22:04:29 mv-snmp-dd sudo[772]: pam_unix(sudo:session): session opened for user root(uid=0) by omega(uid=1002)
nov. 21 22:04:29 mv-snmp-dd sudo[772]: pam_unix(sudo:session): session closed for user root
nov. 21 22:11:12 mv-snmp-dd sudo[828]:    omega : TTY=tty1 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/interfaces
nov. 21 22:11:32 mv-snmp-dd sudo[832]:    omega : TTY=tty1 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/interfaces
nov. 21 22:12:31 mv-snmp-dd sudo[850]:    omega : TTY=pts/0 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/interfaces
nov. 21 22:18:44 mv-snmp-dd sudo[925]:    omega : TTY=pts/0 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/resolv.conf
nov. 21 22:19:55 mv-snmp-dd sudo[930]:    omega : TTY=pts/0 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/resolv.conf
omega@mv-snmp-dd:~$ _
```

Ou bien : journalctl _COMM=sudo _COMM=sudoedit

```
omega@mv-tpsudo-dd:~$ journalctl _COMM=sudo _COMM=sudoedit
Hint: You are currently not seeing messages from other users and the system.
      Users in groups 'adm', 'systemd-journal' can see all messages.
      Pass -q to turn off this notice.
oct. 28 16:55:23 mv-tpsudo-dd sudo[40295]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/int>
oct. 28 16:55:52 mv-tpsudo-dd sudo[40303]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/int>
oct. 28 16:56:39 mv-tpsudo-dd sudo[40323]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/int>
oct. 28 16:57:08 mv-tpsudo-dd sudo[40328]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/network/int>
oct. 28 17:00:05 mv-tpsudo-dd sudo[40339]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/sbin/ifdown ens18
oct. 28 17:00:05 mv-tpsudo-dd sudo[40339]: pam_unix(sudo:session): session opened for user root(uid=0) by epsilon(uid=1006)
oct. 28 17:00:05 mv-tpsudo-dd sudo[40339]: pam_unix(sudo:session): session closed for user root
oct. 28 17:00:27 mv-tpsudo-dd sudo[40343]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/sbin/ifup ens18
oct. 28 17:00:27 mv-tpsudo-dd sudo[40343]: pam_unix(sudo:session): session opened for user root(uid=0) by epsilon(uid=1006)
oct. 28 17:00:27 mv-tpsudo-dd sudo[40343]: pam_unix(sudo:session): session closed for user root
oct. 28 17:08:21 mv-tpsudo-dd sudo[40389]:    omega : TTY=pts/3 ; PWD=/home/omega ; USER=root ; COMMAND=/usr/bin/editor -- /etc/resolv.conf
omega@mv-tpsudo-dd:~$
```

✅ On retrouve bien l'exécution par l'utilisateur omega, mais en tant que root, des commandes ifdown et ifup, avec ouverture et fermeture tout de suite derrière d'une session root.

✅ On retrouve bien ensuite l'édition des fichiers /etc/network/interfaces et /etc/resolv.conf.

G. PROLONGATION DU SCÉNARIO EN AUTONOMIE

Les administrateurs veulent attribuer des permissions spécifiques à 'epsilon' sans lui accorder un accès complet. Ce dernier doit avoir un certain nombre d'autorisations sur des fichiers système et des commandes système sur toutes les machines. Il doit pouvoir :

- Sur toutes les machines
  - Modifier la configuration réseau (/etc/network/interfaces) ainsi que le fichier resolv.conf.
  - Redémarrer le service réseau.
- Sur la machine que vous utilisez, les machines srv1 et srv2
  - Ajouter et supprimer des utilisateurs (sauf root, son propre compte et les techniciens 'omega' et 'omicron'). *Aucun argument autre que le nom de l'utilisateur ne devra être utilisé lors de l'ajout d'un utilisateur.*

↗ Modifier la configuration pour accorder des permissions ciblées.

↗ Tester les permissions.

↗ Vérifier l'enregistrement des opérations sudo dans le journal.