

Job 07:

Si j'ai bien compris la consigne, pour la 4ème ligne, on doit retourner dans la sortie standard le nom des fichiers commençant par . dans lesquels se trouve le mot-clé "alias"

Voici la commande complète que je décortique juste après:

```
echo Je suis votre fichier texte > une_commande.txt && wc -l  
/etc/apt/sources.list | cut -d " " -f 1 > nb_lignes.txt && cat  
/etc/apt/sources.list > save_sources && sudo find / -type f -name  
'.*' -exec grep -l 'alias' {} \; > liste_alias.txt
```

Tout d'abord:

```
echo Je suis votre fichier texte > une_commande.txt
```

Je fais donc ici la commande echo qui renvoie dans la sortie standard (le Terminal normalement) ce que je lui passe, mais je spécifie une sortie standard différente, ici dans un fichier texte "une_commande.txt", en mettant > parce que je veux écraser une éventuelle sortie "une_commande.txt" déjà présente

Ensuite: deux esperluettes pour indiquer que je veux exécuter une autre commande juste après la première si celle-ci retourne 0 (pas d'erreur):

```
&& wc -l /etc/apt/sources.list | cut -f1 -d' ' > nb_lignes.txt
```

Ici j'utilise la commande wc (word count) avec l'option -l (pour compter les lignes) sur le fichier /etc/apt/sources.list

Cette commande renvoie normalement, dans la sortie standard, le nombre de lignes ainsi que "/etc/apc/sources.list"

Je mets un pipe (|) pour indiquer que je veux que la sortie de ce wc soit le paramètre du cut qui suit.

Le cut est suivi de l'option -f1 pour "field one", c'est-à-dire qu'il ne gardera que le premier champ.

Ensuite, j'indique -d' ' pour "délimiter" ce que je considère comme le premier champ, par un espace.

Enfin, je redirige la sortie de ce cut vers nb_lignes.txt

Ensuite:

```
&& cat /etc/apt/sources.list > save_sources
```

Donc, à la suite des premières commandes, je fais un cat sur /etc/apt/sources.list et je redirige la sortie standard vers save_sources

Enfin, le plus dur:

```
&& sudo find / -type f -name '.*' -exec grep -l 'alias' {} \; >  
liste_alias.txt
```

Les deux esperluettes pour exécuter la commande à la suite, puis sudo car bon nombre de fichiers commençant par . sont accessibles uniquement en sudo.

Je trouve avec find dans l'ensemble du répertoire / (donc à la racine de mon ordinateur) des fichiers (avec l'option type -f) dont le nom commence par . (avec l'option -name '.*', l'étoile indique ici que seul le point au début compte, peu importe la chaîne de caractères qui suit), puis pour chacun de ces fichiers, j'exécute (avec -exec) la commande grep -l (L minuscule,

donc la liste des noms des fichiers et non les occurrences en elles-mêmes) qui va rechercher l'occurrence de "alias" dans chaque fichier trouvé par find (référéncé par {}), puis j'indique à find que la commande à exécuter est terminée avec \;

Enfin, je redirige la sortie standard de cette exceptionnelle commande vers liste_alias.txt (ça n'est pas demandé dans l'énoncé mais je préfère le faire ainsi pour ne pas noyer mon Terminal; notons que je pourrais tout à fait ne rien mettre en sortie standard et le résultat de la commande sortirait donc dans le Terminal).

Résultat de l'ensemble:

```
merkava@mantak:~/Documents/Plateforme/Job07$ echo Je suis votre fichier texte > une_commande.txt && wc -l /etc/apt/sources.list | cut -f1
sudo find / -type f -name '*' -exec grep -l 'alias' {} \; > liste_alias.txt
grep: /sys/kernel/security/apparmor/.remove: Argument invalide
grep: /sys/kernel/security/apparmor/.replace: Argument invalide
grep: /sys/kernel/security/apparmor/.load: Argument invalide
find: '/run/user/1000/doc': Permission non accordée
find: '/run/user/1000/gvfs': Permission non accordée
merkava@mantak:~/Documents/Plateforme/Job07$ ls
liste_alias.txt  nb_lignes.txt  save_sources  une_commande.txt
merkava@mantak:~/Documents/Plateforme/Job07$ cat nb_lignes.txt
18
merkava@mantak:~/Documents/Plateforme/Job07$ cat une_commande.txt
Je suis votre fichier texte
merkava@mantak:~/Documents/Plateforme/Job07$ head -n 4 liste_alias.txt
/home/User2/.bashrc
/home/merkava/rtl88x2bu/hal/efuse/rtl8822b/.HalEfuseMask8822B_USB.o.cmd
/home/merkava/rtl88x2bu/hal/.hal_intf.o.cmd
/home/merkava/rtl88x2bu/hal/phydm/halrf/.halrf_kfree.o.cmd
merkava@mantak:~/Documents/Plateforme/Job07$ head -n 4 save_sources
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETINST with firmware 20230722-10:48]/ bookworm main non-free-firmware
deb http://deb.debian.org/debian/ bookworm main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware
```

Notons ici que:

Les .remove, .replace et .load dans le dossier apparmor ne sont pas grep-ables, ce qui est normal.

La permission pour find dans deux dossiers n'est pas accordée, mais elle le serait si j'exécutais la commande en tant que root, ce que je ne veux pas faire sur ma machine.

Enfin, notons aussi que sur la capture d'écran, je n'affiche que les 4 premières lignes des fichiers liste_alias.txt et save_sources pour ne pas avoir à mettre ici 100 captures d'écran, mais ces deux fichiers sont bien complets.

Pour aller plus loin:

Voici la commande, que je décortique juste en-dessous:

```
sudo apt install -y tree && tree -o
/home/merkava/Documents/Plateforme/PAPL/tree.save && ls -l && tail
-n 1 /home/merkava/Documents/Plateforme/PAPL/tree.save && sudo apt
update && sudo apt upgrade || echo Failed, no upgrade
```

Tout d'abord:

```
sudo apt install -y tree
```

Rien de particulier ici, l'option -y sert à automatiquement accepter la confirmation d'installation, et nous évite d'avoir à envoyer nous-mêmes un y après l'exécution de la commande (nous sommes fainéants, et ça fera de nous de bons informaticiens).

Ensuite:

```
&& tree -o /home/merkava/Documents/Plateforme/PAPL/tree.save
```

Ici (passons sur les deux esperluettes que j'ai déjà expliquées auparavant et notons que j'aurais pu utiliser une esperluette simple si j'avais voulu lancer la commande `tree -o` en arrière-plan, mais ici je préfère attendre qu'elle se termine avant d'exécuter la prochaine), `tree` génère l'arborescence puis l'option `-o` lui indique de la rediriger vers le fichier `tree.save` (que j'ai pris soin de ranger correctement dans un dossier créé pour l'exercice).

Ensuite:

```
&& ls -l
```

Simple affichage des éléments du dossier courant, sous forme de liste pour faciliter la lecture, rien de particulier ici.

Puis:

```
&& tail -n 1 /home/merkava/Documents/Plateforme/PAPL/tree.save
```

Comme la commande `tree` indique automatiquement en fin de page, lorsqu'elle est exécutée, le nombre d'éléments (fichiers et dossiers) dans l'arborescence, nous reprenons simplement la dernière ligne de `tree.save` avec `tail` (option `-n 1`, pour indiquer que l'on ne veut qu'une seule ligne de la fin, donc la dernière ligne).

Enfin:

```
&& sudo apt update && sudo apt upgrade || echo Failed, no upgrade
```

Ici, j'utilise `||` pour indiquer une condition. On interprète donc cette partie de la commande comme suit:

Si `sudo apt upgrade` retourne autre chose que 0 (et donc n'est pas réussie), alors on exécute `echo Failed, no upgrade` (notons que je me permets de renvoyer un message, bien que je pourrais simplement faire un `echo` sans rien en paramètre)

Et comme `sudo apt upgrade` ne s'exécute que si `sudo apt update` s'exécute correctement (grâce aux deux `&&`), alors si `sudo apt update` ne s'exécute pas correctement, `sudo apt upgrade` ne s'exécute pas et ne retourne pas 0, et `echo Failed, no upgrade` s'exécute.

Précision: ces retours de la part des commandes sont ce que l'on appelle des exit codes, qui vont de 0 à 255 (parce que codés sur 8 bits, donc 1 octet). Le 0 par exemple correspond à une réussite, le 1 à une erreur générale.