



Born2beRoot

Résumé: Ce document est un sujet d'Administration Système.

Table des matières

I	Préambule	2
II	Introduction	3
III	Consignes générales	4
IV	Partie obligatoire	5
V	Partie Bonus	10
VI	Rendu et peer-évaluation	12

Chapitre I

Préambule



Chapitre II

Introduction

Ce projet a pour but de vous faire découvrir le merveilleux monde de la virtualisation.

Vous allez créer votre première machine en respectant des consignes précises et en utilisant **VirtualBox**. Ainsi, à la suite de ce projet, vous serez capable d'installer votre propre système d'exploitation implémentant des règles strictes.

Chapitre III

Consignes générales

L'utilisation de VirtualBox est obligatoire.

Vous devez rendre uniquement un fichier `signature.txt` à la racine de votre dépôt. Il contiendra la signature du disque virtuel de votre machine. Pour récupérer cette signature, il faudra tout d'abord aller dans le dossier d'installation par défaut (c'est dans ce dossier que sont sauvegardées vos VMs) :

- Pour Windows : `%HOMEDRIVE%%HOMEPATH%\VirtualBox VMs\`
- Pour Linux : `~/VirtualBox VMs/`
- Pour MacOS : `~/VirtualBox VMs/`

Il suffira alors de récupérer la signature du fichier `".vdi"` de votre machine virtuelle au format `sha1`. Voici 3 exemples de commande avec un fichier `centos_serv.vdi` :

- Pour Windows : `certUtil -hashfile centos_serv.vdi sha1`
- Pour Linux : `sha1sum centos_serv.vdi`
- Pour MacOS : `shasum centos_serv.vdi`

Voici un exemple de résultat attendu :

- `6e657c4619944be17df3c31faa030c25e43e40af`



Attention, votre signature risque de changer à la suite de votre première évaluation. Pour pallier à ce problème, plusieurs solutions s'offrent à vous, comme dupliquer votre machine virtuelle ou encore utiliser les save state.



Il est bien entendu INTERDIT de rendre votre machine virtuelle dans votre dépôt Git. Durant la soutenance, la signature du fichier `signature.txt` sera comparée avec celle de votre machine virtuelle. Si les deux signatures ne sont pas identiques, vous aurez 0.

Chapitre IV

Partie obligatoire

Ce projet consiste à vous faire mettre en place votre premier serveur en suivant des règles spécifiques.

Vous devez utiliser comme système d'exploitation, au choix : Debian latest stable (pas de testing/unstable), ou CentOS latest stable. L'utilisation de Debian est fortement conseillée pour quelqu'un débutant dans ce domaine.



La mise en place de CentOS est plus complexe. Par conséquent, vous n'avez pas l'obligation de mettre en place KDUMP. Cependant, SELinux devra rester actif et sa configuration sera adaptée au sujet. AppArmor pour Debian devra également rester actif.

Vous devez créer au minimum 2 partitions chiffrées en utilisant LVM. Voici un exemple de partition attendue pour votre machine virtuelle :

```
wil@wil:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0   8G  0 disk
├─sda1                              8:1    0 487M  0 part  /boot
├─sda2                              8:2    0    1K  0 part
├─sda5                              8:5    0  7.5G  0 part
│   └─sda5_crypt                    254:0    0  7.5G  0 crypt
│       ├─wil--vg-root               254:1    0  2.8G  0 lvm    /
│       ├─wil--vg-swap_1             254:2    0  976M  0 lvm    [SWAP]
│       └─wil--vg-home               254:3    0  3.8G  0 lvm    /home
sr0                                  11:0    1 1024M  0 rom
```



Durant la soutenance, des questions seront posées en fonction du système d'exploitation choisi. Informez-vous sur ce que vous utilisez. Par exemple, connaître la différence entre aptitude et apt, mais aussi ce qu'est SELinux ou AppArmor. En bref, il faut comprendre ce que l'on utilise !

Un service **SSH** sera actif sur le port 4242 uniquement. Pour des questions de sécurité, on ne devra pas pouvoir se connecter par **SSH** avec l'utilisateur **root**.



L'utilisation de **SSH** sera testée durant la soutenance par la mise en place d'un nouveau compte. Il faut par conséquent comprendre comment fonctionne ce service.

Vous allez configurer votre système d'exploitation avec le pare-feu **UFW** et ainsi ne laisser ouvert que le port 4242.



Votre pare-feu devra être actif au lancement de votre machine virtuelle. Pour **CentOS**, vous utiliserez **UFW** à la place du pare-feu de base. Pour l'installer, vous allez devoir probablement utiliser **DNF**.

- Votre machine aura pour **hostname** votre login suivi de 42 (exemple : wil42).
- Vous allez mettre en place une politique de mot de passe fort.
- Vous allez installer et configurer **sudo** selon une pratique stricte.
- Un utilisateur sera présent avec pour nom votre login en plus de l'utilisateur **root**.
- Cet utilisateur appartiendra aux groupes **user42** et **sudo**.



Durant la soutenance, vous allez devoir créer un nouvel utilisateur et lui assigner un groupe.

Pour mettre en place une politique de mot de passe fort, il faudra remplir les conditions suivantes :

- Votre mot de passe devra expirer tous les 30 jours.
- Le nombre minimum de jours avant de pouvoir modifier un mot de passe sera configuré à 2.
- L'utilisateur devra recevoir un avertissement 7 jours avant que son mot de passe n'expire.
- Votre mot de passe sera de 10 caractères minimums dont une majuscule et un chiffre, et ne devra pas comporter plus de 3 caractères identiques consécutifs.
- Le mot de passe ne devra pas comporter le nom de l'utilisateur.
- Le mot de passe devra comporter au moins 7 caractères qui ne sont pas présents dans l'ancien mot de passe.
- Bien entendu votre mot de passe **root** devra suivre cette politique.



Après avoir mis en place vos fichiers de configuration, il faudra changer tous les mots de passe des comptes présents sur la machine virtuelle, compte root inclus.

Pour mettre en place une configuration stricte dans votre groupe **sudo**, il faudra remplir les conditions suivantes :

- L'authentification en utilisant **sudo** sera limitée à 3 essais en cas de mot de passe erroné.
- Un message de votre choix s'affichera en cas d'erreur suite à un mauvais mot de passe lors de l'utilisation de **sudo**.
- Chaque action utilisant **sudo** sera archivée, aussi bien les inputs que les outputs. Le journal se trouvera dans le dossier `/var/log/sudo/`.
- Le mode TTY sera activé pour des questions de sécurité.
- Les paths utilisables par **sudo** seront restreints, là encore pour des questions de sécurité. Exemple :
`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

Enfin, vous devrez mettre en place un petit script nommé `monitoring.sh`. Ce dernier sera à développer en `bash`.

Dès le lancement de votre serveur, le script écrira des informations toutes les 5 minutes sur tous les terminaux (jetez un oeil du côté de `wall`). La bannière est facultative. À aucun moment la moindre erreur ne doit être visible.

Votre script devra toujours pouvoir afficher les informations suivantes :

- L'architecture de votre système d'exploitation ainsi que sa version de kernel.
- Le nombre de processeurs physiques.
- Le nombre de processeurs virtuels.
- La mémoire vive disponible sur votre serveur ainsi que son taux d'utilisation sous forme de pourcentage.
- La mémoire disponible sur votre serveur ainsi que son taux d'utilisation sous forme de pourcentage.
- Le taux d'utilisation de vos processeurs sous forme de pourcentage.
- La date et l'heure du dernier redémarrage.
- Si LVM est actif ou pas.
- Le nombre de connexions actives.
- Le nombre d'utilisateurs utilisant le serveur.
- L'adresse IPv4 de votre serveur, ainsi que son adresse MAC (Media Access Control).
- Le nombre de commande exécutées avec le programme `sudo`.



Durant la soutenance, vous serez amené(e) à expliquer le fonctionnement de ce script et à interrompre son exécution sans le modifier. Regardez du côté de `cron`.

Voici un exemple d'exécution du script attendue :

```
Broadcast message from root@wil (tty1) (Sun Apr 25 15:45:00 2021):

#Architecture: Linux wil 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
#CPU physical : 1
#vCPU : 1
#Memory Usage: 74/987MB (7.50%)
#Disk Usage: 1009/2Gb (39%)
#CPU load: 6.7%
#Last boot: 2021-04-25 14:45
#LVM use: yes
#Connexions TCP : 1 ESTABLISHED
#User log: 1
#Network: IP 10.0.2.15 (08:00:27:51:9b:a5)
#Sudo : 42 cmd
```

Voici deux exemples avec des commandes simples pour vérifier une partie des demandes du sujet :

Pour CentOS :

```
[root@wil ~]# head -n 2 /etc/os-release
NAME="CentOS Linux"
VERSION="8"
[root@wil ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 enforcing
Mode from config file:       enforcing
Policy MLS status:           enabled
Policy deny_unknown status:   allowed
Memory protection checking:   actual (secure)
Max kernel policy version:    32
[root@wil ~]# ss -tunlp
Netid State  Recv-Q Send-Q   Local Address:Port   Peer Address:Port
tcp    LISTEN  0      128      0.0.0.0:4242        0.0.0.0:*    users:((("sshd",pid=822,fd=5))
tcp    LISTEN  0      128      :::4242            :::*         users:((("sshd",pid=822,fd=7))
[root@wil ~]# ufw status
Status: active

To Action From
--
4242 ALLOW Anywhere
4242 (v6) ALLOW Anywhere (v6)

[root@wil ~]# _
```

Pour Debian :

```
root@wil:~# head -n 2 /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
root@wil:/home/wil# /usr/sbin/aa-status
apparmor module is loaded.
root@wil:/home/wil# ss -tunlp
Netid State  Recv-Q Send-Q   Local Address:Port   Peer Address:Port
tcp    LISTEN  0      128      0.0.0.0:4242        0.0.0.0:*    users:((("sshd",pid=523,fd=3))
tcp    LISTEN  0      128      :::4242            :::*         users:((("sshd",pid=523,fd=4))
root@wil:/home/wil# /usr/sbin/ufw status
Status: active

To Action From
--
4242 ALLOW Anywhere
4242 (v6) ALLOW Anywhere (v6)
```

Chapitre V

Partie Bonus

Liste de bonus :

- Mettre correctement en place des partitions afin d'obtenir une structure proche de cet exemple :

```
# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                8:0    0 30.8G  0 disk
├─sda1                             8:1    0   500M  0 part  /boot
├─sda2                             8:2    0     1K  0 part
└─sda5                             8:5    0 30.3G  0 part
   └─sda5_crypt                    254:0    0 30.3G  0 crypt
      ├─LVMGroup-root              254:1    0   10G  0 lvm    /
      ├─LVMGroup-swap              254:2    0   2.3G  0 lvm    [SWAP]
      ├─LVMGroup-home              254:3    0     5G  0 lvm    /home
      ├─LVMGroup-var               254:4    0     3G  0 lvm    /var
      ├─LVMGroup-srv              254:5    0     3G  0 lvm    /srv
      ├─LVMGroup-tmp              254:6    0     3G  0 lvm    /tmp
      └─LVMGroup-var--log          254:7    0     4G  0 lvm    /var/log
sr0                                11:0    1 1024M  0 rom
```

- Mettre en place un site web WordPress fonctionnel avec, comme services, lighttpd, MariaDB et PHP.
- Mettre en place un service qui vous semble utile (NGINX/Apache2 exclus!). Durant la soutenance, vous aurez à justifier ce choix.



Dans le cadre des bonus, vous avez la possibilité de mettre en place d'autres services. Dans ce cas, il pourra y avoir plus de ports ouverts selon vos besoins. Bien entendu, les règles d'UFW seront adaptées en conséquence.



Les bonus ne seront évalués que si la partie obligatoire est PARFAITE. Par parfaite, nous entendons complète et sans aucun dysfonctionnement. Si vous n'avez pas réussi TOUS les points de la partie obligatoire, votre partie bonus ne sera pas prise en compte.

Chapitre VI

Rendu et peer-évaluation

Rendez votre travail dans votre dépôt Git comme d'habitude. Seul le travail présent dans votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.



Rappel : Il est INTERDIT de mettre votre machine virtuelle dans votre dépôt Git.