Fait par Raphaël GEAY

Étape 1 : Analyse et nettoyage du serveur

🟢 Lister les tâches cron pour détecter des backdoors :

```
[root@localhost ~]# cat /var/log/cron
```

🟢 Identifier et supprimer les fichiers cachés :

```
[root@localhost ~]# find /tmp -type f -name ".*"
```

```
[root@localhost ~]# rm /tmp/.hidden_script
[root@localhost ~]# rm /tmp/.hidden_file
```

```
[root@localhost ~]# find /var/tmp -type f -name ".*"
```

```
[root@localhost ~]# rm /var/tmp/.nop
```

```
[root@localhost ~]# find /home -type f -name ".*"
```

```
[root@localhost ~]# rm /home/attacker/.bash_profile
[root@localhost ~]# rm /home/attacker/.bashrc
[root@localhost ~]# rm /home/attacker/.bash_logout
[root@localhost ~]# rm /home/attacker/.bash_history
[root@localhost ~]# rm /home/attacker/.hidden_file
[root@localhost ~]# rmdir /home/attacker
```

🟢 Analyser les connexions réseau actives :

```
[root@localhost ~]# ss
```

Étape 2 : Configuration avancée de LVM

🟢 Créer un snapshot de sécurité pour /mnt/secure_data :

```
[root@localhost ~]# lvcreate --size 1G --snapshot --name secure_snapshot
/dev/vg_secure/secure_data
```

🟢 Tester la restauration du snapshot :

```
[root@localhost ~]# rm /mnt/secure_data/sensitive1.txt
```

```
[root@localhost ~]# mount /dev/vg_secure/secure_snapshot /mnt/secure_snapshot
```

```
[root@localhost ~]# cp /mnt/secure_snapshot/sensitive1.txt /mnt/secure_data/
```

🟢 Optimiser l'espace disque :

```
[root@localhost ~]# fuser -km /mnt/secure_data
[root@localhost ~]# umount /mnt/secure_data
```

```
[root@localhost ~]# lvextend --size +100M /dev/vg_secure/secure_data
```

```
[root@localhost ~]# lvextend --extents +100%FREE /dev/vg_secure/secure_data
```

Étape 3 : Automatisation avec un script de sauvegarde

🟢 Créer un script secure_backup.sh :

```
[root@localhost ~]# dnf install tar
```

```
[root@localhost ~]# nano secure_backup.sh
```

```
CURRENT_DATE=$(date +'%Y%m%d')

SOURCE_DIR="/mnt/secure_data"
BACKUP_DIR="/backup"

BACKUP_FILE="${BACKUP_DIR}/secure_data_${CURRENT_DATE}.tar.gz"

mkdir -p $BACKUP_DIR

tar --exclude='*/.tmp' --exclude='*/.log' --exclude='.*' -czf $BACKUP_FILE -C
$SOURCE_DIR .

if [ $? -eq 0 ]; then
    echo "La sauvegarde a été effectuée avec succès : $BACKUP_FILE"
else
    echo "Erreur lors de la sauvegarde."
    exit 1
fi
```

🟢 Ajoutez une fonction de rotation des sauvegardes :

```
[root@localhost ~]# nano secure_backup.sh
```

```
cd $BACKUP_DIR
BACKUPS=($(ls -1t secure_data_*.tar.gz))

COUNT=${#BACKUPS[@]}
if [ $COUNT -gt 7 ]; then
    TO_DELETE=$((COUNT - 7))
    for i in $(seq 7 $((COUNT - 1))); do
        echo "Suppression de l'ancienne sauvegarde : ${BACKUPS[$i]}"
        rm -f "${BACKUPS[$i]}"
    done
fi
```

🟢 Testez le script :

```
[root@localhost ~]# ./secure_backup.sh
```

🟢 Automatisez avec une tâche cron :

```
[root@localhost ~]# crontab -e
```

```
0 3 * * * secure_backup.sh
```

Étape 4 : Surveillance avancée avec auditd

🟢 Configurer auditd pour surveiller /etc :

```
[root@localhost ~]# dnf install auditd
[root@localhost ~]# systemctl start auditd
[root@localhost ~]# systemctl enable auditd
```

```
[root@localhost ~]# auditctl -w /etc -p wa -k etc_changes
```

```
[root@localhost ~]# echo "-w /etc -p wa -k etc_changes" | tee -a
/etc/audit/rules.d/audit.rules
```

```
[root@localhost ~]# augenrules --load
```

🟢 Tester la surveillance :

```
[root@localhost ~]# touch /etc/test
```

```
[root@localhost ~]# ausearch -k etc_changes
```

🟢 Analyser les événements :

```
[root@localhost ~]# ausearch -k etc_changes > /var/log/audit_etc.log
```

```
[root@localhost ~]# cat /var/log/audit_etc.log
```

Étape 5 : Sécurisation avec Firewalld

🛡️ Configurer un pare-feu pour SSH et HTTP/HTTPS uniquement :

```
[root@localhost ~]# dnf install firewalld
[root@localhost ~]# systemctl start firewalld
[root@localhost ~]# systemctl enable firewalld
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=ssh --permanent
[root@localhost ~]# firewall-cmd --zone=public --add-service=http --permanent
[root@localhost ~]# firewall-cmd --zone=public --add-service=https --permanent
```

```
[root@localhost ~]# firewall-cmd --set-default-zone=drop
```

```
[root@localhost ~]# firewall-cmd --reload
```

🛡️ Bloquer des IP suspectes :

```
[root@localhost ~]# strings /var/log/secure | grep "Failed password"
```

```
[root@localhost ~]# firewall-cmd --add-rich-rule="rule family=ipv4 source
address=203.0.113.42 reject" --permanent
```

```
[root@localhost ~]# firewall-cmd --reload
```

🛡️ Restreindre SSH à un sous-réseau spécifique :

```
[root@localhost ~]# firewall-cmd --add-rich-rule="rule family=ipv4 source
address=192.168.38.0/24 service name=ssh accept" --permanent
```

```
[root@localhost ~]# firewall-cmd --remove-service=ssh --permanent
```

```
[root@localhost ~]# firewall-cmd --reload
```