TP : Secadmin

Réalisé par : Chams TMAR (GL3-2)

Dans ce TP, on voulait écrire une application shell pour l'administration de la sécurité. Ainsi, on va afficher un menu à l'utilisateur, présentant différentes options.
En fonction du choix de l'utilisateur, le script exécute différentes actions à l'aide d'une structure de contrôle `case` :

- Si le choix est **1**, on liste les fichiers SUID.
- Si le choix est **2**, on liste les fichiers GUID.
- Si le choix est **3**, on liste les répertoires Sticky bit.
- Si le choix est **4**, on lance un sniffer `tcpdump`.
- Si le choix est **5**, on gère un firewall (`iptables`).
- Si le choix est **6**, on vérifie SUID (bit de contrôle d'intégrité pour l'utilisateur).
- Si le choix est **7**, on vérifie GUID (bit de contrôle d'intégrité pour le groupe).
- Si le choix est **8**, on vérifie Sticky bit (bit de contrôle d'intégrité pour les autres).
- Si le choix est **9**, le script quitte en utilisant la commande `exit`.

Voici le script expliqué :

```
echo "Administration de la sécurité"

while true
do

echo "1- Lister les fichiers suid"
echo "2- Lister les fichiers guid"
echo "3- Lister les répertoires sticky bit"
echo "4- Lancer un sniffer (tcpdump)"
echo "5- Gérer un firewall (iptables)"
echo "6- Check suid (bit ss integrity control for user)"
echo "7- Check guid (bit ss integrity control for group)"
echo "8- Check sticky bit (bit ss integrity control for
others)"
```

```
echo "9- Quitter"

echo ""
echo "Tapez votre choix"
read choix

case $choix in
1)    find / -perm -4000 2> /dev/null
#on recherche les fichiers réguliers (-type f) dans le répertoire racine (/) avec le bit de permission setuid défini (-perm /4000) et on redirige les messages d'erreur vers /dev/null, les supprimant.
      ;;
2)    find / -perm -2000 2> /dev/null
#similaire au premier choix, mais recherche les fichiers avec le bit de permission setgid défini (-perm /2000).
      ;;
3)    find / -perm -1000 2> /dev/null
#similaire aux deux premiers choix, mais recherche les fichiers avec le bit collant défini (-perm /1000).
      ;;
4)    echo "Insérer l'interface"
      read interface
      sudo tcpdump -i $interface
      #on exécute tcpdump pour capturer et afficher le trafic réseau sur l'interface saisie par l'utilisateur.
      ;;
5)    #vérifier les droits d'administration
      if [[ $EUID -ne 0 ]]; then
           echo "Ce script doit être exécuté en tant qu'administrateur (root)."
      exit 1
      fi
      #afficher le menu
      echo "=== Gestion du Firewall (iptables) ==="
      echo "51. Autoriser le trafic sur un port"
      echo "52. Bloquer le trafic sur un port"
      echo "53. Autoriser toutes les connexions sortantes"
```

```
    echo "54. Bloquer toutes les connexions sortantes"
    echo "55. Afficher les règles iptables actuelles"
    echo "56. Quitter"

    read -p "Entrez le numéro de l'option souhaitée : "
choice

    #utiliser la structure conditionnelle case pour traiter
le choix de l'utilisateur
    case $choice in
        51)
                #autoriser le trafic sur un port
            read -p "Entrez le numéro du port à autoriser :
" port
            iptables -A INPUT -p tcp --dport $port -j
ACCEPT
            echo "Le trafic sur le port $port est
autorisé."
            ;;
        52)
            #bloquer le trafic sur un port
            read -p "Entrez le numéro du port à bloquer : "
port
            iptables -A INPUT -p tcp --dport $port -j DROP
            echo "Le trafic sur le port $port est bloqué."
            ;;
        53)
            #autoriser toutes les connexions sortantes
            iptables -P OUTPUT ACCEPT
            echo "Toutes les connexions sortantes sont
autorisées."
            ;;
        54)
            #bloquer toutes les connexions sortantes
            iptables -P OUTPUT DROP
            echo "Toutes les connexions sortantes sont
bloquées."
            ;;
        55)
```

```
                #afficher les règles iptables actuelles
                echo "Règles iptables actuelles :"
                iptables -L
                ;;
        56)
                exit 0
                ;;
        *)
                #gérer les choix non valides
                echo "Option non valide. Veuillez choisir un
numéro entre 1 et 6."
                ;;
    esac

    #enregistrement des règles iptables
    service iptables save
    echo "Les règles iptables ont été enregistrées."

    #redémarrer le service iptables
    service iptables restart
    echo "Le service iptables a été redémarré."
    ;;
6)  echo "Saisir le nom d'utilisateur"
    read choix
    sudo find /-user $choix -perm /4000
    #on recherche les fichiers setuid (-perm /4000)
appartenant à l'utilisateur spécifié dans le répertoire
racine.
    ;;
7)  echo "saisir le nom du groupe"
    read choix
    sudo find /-group $choix -perm /2000
    #on recherche les fichiers setgid (-perm /2000)
appartenant au groupe spécifié dans le répertoire racine.
    ;;
8)  sudo find / -perm -ott 2>/dev/null
```

```
#on recherche les fichiers dans le répertoire racine avec le
bit d'exécution "autre" défini (-perm -o+t) et on redirige les
erreurs vers /dev/null, les supprimant.
      ;;
9)    break
      ;;
esac

done
```