Objectifs :

- Création, modification et suppression de comptes utilisateurs et de groupes
- Planification de tâches ponctuelles ou répétitives
- Contrôler et analyser des fichiers journaux
- Gérer les règles de sécurité de base sous Linux

I – Administration des utilisateurs :

1. Se connecter en tant que root.

2. Lister les comptes utilisateurs et les groupes existants sur le système.

3. Quels sont l'UID et le GID du compte root ?

4. Quelles sont les valeurs minimales des UID et GID utilisées par défaut lors de la création de nouveaux comptes utilisateurs ou groupes ? Faire en sorte que l'UID minimal soit 2000 et que le GID minimal soit 1500.

5. Créer les groupes grp1, grp2 et grp3. Le GID du groupe grp3 doit être 1523.

6. Créer les comptes utilisateurs util1, util2 et util3 ayant respectivement les groupes grp1, grp2 et grp3 comme groupe principal. Le compte util2 doit aussi être membre des groupes grp1 et grp3. Le pseudonyme "tux1" de l'utilisateur util1 doit être renseigné dans la configuration.

7. Noter les UID et GID des comptes utilisateurs et des groupes créés précédemment.

8. Supprimer le groupe grp3. Cela est-il possible ? Pourquoi ?

9. Supprimer le compte utilisateur util3 sans supprimer son répertoire personnel et supprimer le groupe grp3.

10. À qui appartient maintenant le répertoire /home/util3 ? Pourquoi ?

11. Comment retrouver et supprimer tous les fichiers orphelins du système qui appartenaient au compte utilisateur util3 ou au groupe grp3 ?

12. Se connecter en tant que util1 sur une console. Cela s'est-il bien passé ?

II – Cron :

1. Se connecter en tant que root.

2. Vérifier que le démon crond s'exécute actuellement, sinon le lancer.

3. Consulter les fichiers /etc/cron.allow et /etc/cron.deny.

4. Se connecter en tant que user de base sur un autre shell.

5. Afficher le contenu de la crontab de user.

6. Programmer l'exécution de la commande /bin/pwd > /tmp/pwd.out toutes les minutes dans la crontab de user.

7. Au bout d'une minute, afficher le contenu du fichier /tmp/pwd.out.

8. Retourner sur le shell en tant que root et arrêter le démon crond.

9. Supprimer le fichier /tmp/pwd.out.

10. Ajouter le nom de l'utilisateur user au fichier /etc/cron.deny.

11. Redémarrer le démon crond.

12. Retourner sur le shell sous user et éditer de nouveau la crontab de cet utilisateur. Est-ce possible ?

13. Au bout d'une minute, afficher le contenu du fichier /tmp/pwd.out.

Que peut-on en déduire ?

14. De nouveau sur le shell en tant que root, supprimer le contenu de la crontab de user et redémarrer le démon crond.

15. Programmer deux fois par jour dans une crontab système, à 9h30 et 13h30, la suppression dans /tmp des fichiers dont la date de modification est supérieure à un jour.

16. Faire en sorte que les modifications apportées à la crontab système soient prises en compte.

III – Les logs :

1. Aller dans le répertoire /var/log.
2. Lister, avec un affichage détaillé, les fichiers du répertoire du plus ancien au plus récent.
3. Consulter le fichier des messages.
4. Visualiser uniquement les dernières lignes du fichier avec un affichage mis à jour en permanence.

IV – Bases de sécurité :

1. Créer une copie de bash dans /usr/bin/hack avec le SUID positionné.

   Vérifier les droits sur /usr/bin/hack.
2. Rechercher tous les fichiers disposant des bits SUID et/ou SGID.
3. Vérifier que le système ne possède pas de rootkits connus.

V – Sécurité des services et du réseau :

1. Lister les ports à l'écoute et les processus correspondants.
2. Interdire l'accès en SSH aux machines ne faisant pas partie du réseau local.
3. Créer une règle qui interdit le ping. Vérifier depuis un autre poste.
4. Que se passe-t-il si on remplace la cible DROP par REJECT dans la règle précédente ? Pourquoi est-ce moins judicieux ?
5. Créer une règle qui interdit l'accès au serveur web monté dans le TP précédent pour les adresses sources ne faisant pas partie du réseau local.
6. Afficher toutes les règles.

7. Effacer toutes les règles et vérifier que toutes les règles ont bien été supprimées.