TP : Sécurisation d'un Système d'Exploitation avec des Protocoles d'Authentification et des Modèles de Contrôle d'Accès

Problématique :

Vous êtes un **ingénieur en cybersécurité** travaillant pour une entreprise qui souhaite **renforcer la sécurité de son infrastructure informatique**. Actuellement, les employés et administrateurs utilisent des comptes locaux pour se connecter aux serveurs et postes de travail, ce qui **pose plusieurs problèmes** :

- **Gestion décentralisée des comptes** : Chaque machine a sa propre base d'utilisateurs, rendant difficile la gestion des accès.
- **Absence d'authentification forte** : Les utilisateurs s'authentifient uniquement avec des mots de passe, augmentant le risque de vol de credentials.
- **Mauvaise gestion des privilèges** : Les utilisateurs conservent parfois des droits excessifs ou inutiles.
- **Aucune traçabilité** : Il n'existe **aucun journal centralisé** des tentatives de connexion et des accès.

Votre mission est de **proposer et implémenter une solution** pour **centraliser l'authentification**, **renforcer la sécurité des connexions**, et **contrôler les accès** en fonction des rôles des utilisateurs.

Objectifs du TP :

À travers cette étude de cas, vous devrez :

- Configurer un serveur RADIUS et un serveur Kerberos pour centraliser l'authentification des utilisateurs.
- Intégrer ces solutions avec un système Linux pour que les connexions SSH passent par un serveur d'authentification.
- Mettre en place des modèles de contrôle d'accès (DAC, MAC, RBAC) pour limiter les privilèges et assurer la protection des données sensibles.
- Assurer la traçabilité des accès et des authentifications.

Problème 1 : Centralisation de l'authentification avec RADIUS

Contexte :

L'entreprise souhaite éviter que chaque serveur ait sa propre gestion des utilisateurs et préfère **centraliser l'authentification**. Vous devez mettre en place un serveur **RADIUS** qui sera responsable de l'authentification des connexions des employés et administrateurs.

Problème à résoudre :

1. Comment éviter que chaque machine Linux gère localement ses propres utilisateurs ?
2. Comment configurer un serveur RADIUS pour qu'il puisse gérer les demandes d'authentification des postes clients ?
3. Comment tester et vérifier que l'authentification est bien déportée sur le serveur RADIUS ?

Tâches :

- Installer et configurer un serveur **FreeRADIUS** sur une machine dédiée.
- Ajouter un utilisateur et configurer les clients (serveurs Linux) pour utiliser ce serveur RADIUS.
- Tester l'authentification depuis un poste client.
- Activer la journalisation des connexions pour analyser les tentatives d'accès.

Livrables attendus :

- Configuration du serveur RADIUS
- Captures des logs montrant le succès ou l'échec des tentatives de connexion

Problème 2 : Sécurisation des connexions avec Kerberos et authentification unique (SSO)

Contexte :

L'entreprise veut que les employés s'authentifient une seule fois et puissent ensuite accéder à différents services sans devoir **ressaisir leur mot de passe** à chaque connexion.

Problème à résoudre :

1. Comment permettre aux utilisateurs d'obtenir une authentification unique pour plusieurs services ?
2. Comment sécuriser les échanges d'authentification en évitant de transmettre des mots de passe en clair sur le réseau ?
3. Comment tester que les utilisateurs peuvent accéder aux services via des **tickets Kerberos** ?

Tâches :

- Installer et configurer un **serveur Kerberos** pour gérer les identités.
- Créer des utilisateurs et tester l'obtention de tickets via `kinit`.
- Configurer un serveur SSH pour qu'il accepte l'authentification Kerberos.
- Vérifier que les utilisateurs ne doivent pas ressaisir leurs mots de passe après leur première connexion.

Livrables attendus :

- Fichier de configuration Kerberos
- Résultats des commandes `klist`, `kinit` et `ssh` montrant l'authentification sans mot de passe

Problème 3 : Application de politiques de contrôle d'accès (DAC, MAC, RBAC)

Contexte :

Actuellement, les utilisateurs peuvent modifier leurs permissions à leur guise. L'entreprise souhaite imposer un **contrôle strict** pour éviter les accès non autorisés aux fichiers et services critiques.

Problème à résoudre :

1. Comment empêcher qu'un utilisateur malveillant modifie les permissions de ses propres fichiers pour les partager avec d'autres ?
2. Comment appliquer des **règles strictes** sur certains fichiers critiques avec SELinux ou AppArmor ?
3. Comment implémenter **un modèle basé sur les rôles (RBAC)** pour limiter les privilèges des administrateurs ?

Tâches :

- Expérimenter **DAC (Discretionary Access Control)** en manipulant les commandes `chmod`, `chown` et `umask`.
- Mettre en place **MAC (Mandatory Access Control)** avec **SELinux** ou **AppArmor** pour restreindre certains accès.
- Configurer **RBAC (Role-Based Access Control)** en définissant des groupes d'utilisateurs et des permissions spécifiques via `sudoers`.
- Tester et documenter les résultats des différentes configurations.

Livrables attendus :

- Résultats des tests montrant comment les accès sont bloqués selon les règles définies
- Captures des logs d'audit SELinux/AppArmor montrant des accès refusés

Problème 4 : Surveillance et journalisation des accès

Contexte :

L'entreprise veut **surveiller et analyser** les tentatives de connexion et d'accès aux fichiers sensibles. Elle souhaite mettre en place un **mécanisme de journalisation** efficace et être alertée en cas d'activité suspecte.

Problème à résoudre :

1. Quels fichiers et événements doivent être surveillés en priorité dans un système Linux ?
2. Comment configurer un **journal centralisé** pour suivre les authentifications et les accès non autorisés ?
3. Comment mettre en place une alerte automatique en cas de tentative de connexion suspecte ?

Tâches :

- Configurer la journalisation des connexions SSH (`/var/log/auth.log`).
- Activer l'audit des accès aux fichiers critiques via **auditd** et définir des règles (`auditctl`).
- Mettre en place une alerte automatique (ex. `fail2ban` ou `logwatch`) en cas de tentative de connexion anormale.

Livrables attendus :

- Extrait des logs montrant les accès réussis et échoués
- Configuration `auditctl` et résultats des tests d'accès
- Documentation expliquant comment activer une alerte en cas d'intrusion