2 Gérer les mots de passe, utilisateurs et groupes avec Ansible

Fichiers sources :

- **Dossier "ansible/2"**

Forcer l'utilisation de mots de passe forts

- Installation de libpam-pwquality
  - Créer tâche Ansible pour installer et configurer le paquet
    - ansible/2/pam_pwquality.yml
      - contient 2 tâches
        - Installer libpam-pwquality
        - Configurer pam_pwquality
      - Module package
        - state: present/absent
      - Les actions Ansible sont idempotent (leur résultat sera toujours le même)
- Configuration de pam_pwquality pour forcer un politique de mots de passe forts
  - Sur Ubuntu de base la taille mini d'un pass est de 6 caractères
  - Fichier /etc/pam.d/common-password permet config pam_pwquality
  - Module Ansible lineinfile
    - regexp pour trouver la ligne à remplacer
    - line: contenu à insérer
      - Taille password minimum 12 caractères
      - Au moins une lettre minuscule
      - Au moins une lettre majuscule
      - Au moins un chiffre
      - Au moins un caractère spécial
      - Trois tentatives
      - Applique les règles y compris à root

Utilisateurs Linux

- Premiers pas avec le module User d'Ansible
  - ouvrir le fichier ansible/2/user_and_group.yml
    - Contient 5 tâches
- Génération d'un mot de passe fort
  - Utilisation de 2 outils
    - pwgen
    - mkpasswd
  - Linux stocke les passwd dans un fichier nommé shadow
  - L'algo de hachage par défaut est SHA-512
  - Commandes
    - sudo apt update
    - sudo apt install pwgen whois
    - pass=` pwgen --secure --capitalize --numerals --symbols 12 1`
    - echo $pass | mkpasswd --stdin --method=sha-512; echo $pass
  - ATTENTION ! Dans la vraie vie, ne JAMAIS stocker un pass en clair dans un fichier sur dépôt de code, utilisez Ansible Vault (ou autre) pour stocker les secrets.
    - https://docs.ansible.com/ansible/latest/user_guide/vault.html

Groupes Linux

- Premiers pas avec le module Group d'Ansible
  - Les tâches sont exécutées dans l'ordre où elles sont écrites, donc attention à l'ordre !
  - On doit créer le groupe AVANT d'y assigner un user.
- Assignation d'un utilisateur à un groupe
  - On peut spécifier plusieurs groupes en les séparants par une virgule.
  - L'option "append" conserve les précédents groupes assignés à l'utilisateur et ajoute le groupe "developers".
    Sans cette option, l'utilisateur sera supprimé de tous les groupes sauf le groupe portant son nom et celui ou ceux listés au paramètre "groups".
- Création de ressources protégées
  - man chmod
  - Modules à utiliser pour la création de fichiers :
    - Vide : touch
    - Non-vide : copy ou template

Mettre à jour la VM

- lignes commençant par "#- import_tasks: 2/"
  - Décommenter UNIQUEMENT les tâches utiles dans le playbook "site.yml" situé dans le dossier "ansible"
- Sauvegarder et fermer le fichier
- ***vagrant up***
  - Si la VM créée précédemment est éteinte
- ***vagrant provision***
  - Si la VM est en cours d'exécution
- ***vagrant provision --debug***
  - Si erreur(s)
- ATTENTION !
  - N'oubliez pas de toujours lancer les commandes vagrant à partir du dossier " vagrant"

Tester les permissions des utilisateurs et groupes

- ***vagrant ssh***
  - Connexion à la VM
- Vous êtes maintenant connecté avec l' utilisateur "vagrant" (utilisateur par défaut créé par Vagrant)
- ***getent passwd florian***
  - Vérifier que l'utilisateur "florian" a bien été créé
- Cette commande requête les fichiers /etc/passwd, /etc/shadow, /etc/group
- Si l'utilisateur n'est pas trouvé, la commande n'affiche rien.
- Vérifier les droits d'accès au dossier /opt/engineering et au fichier private.txt
  - En tant que "vagrant"
    - ***ls -la /opt/engineering***
    - La sortie de la commande indique que l' accès est refusé, car l'utilisateur "vagrant" n' est pas membre du groupe "developers" et n' a donc pas les droits de lecture sur le dossier.
  - En tant que "vagrant"
    - ***cat /opt/engineering/private.txt***
    - Même erreur
  - En tant que "florian"
    - ***sudo su - florian***
    - ***ls -la /opt/engineering***
    - La commande s'est terminée avec succès, et le fichier private.txt est bien visible.
    - ***cat /opt/engineering/private.txt***
    - Pas d'erreur. Le fichier étant vide, la commande ne renvoit rien.

Résumé

- Nous avons :
  - provisionné la VM en utilisant les modules Ansible suivants :
    - package
    - lineinfile
    - user
    - group
    - file
  - Ces modules nous ont permis de :
    - durcir la sécurité du système en imposant une politique de mots de passe forts
    - gérer des utilisateurs et groupes
    - sécuriser l'accès à un dossier et un fichier