

Systeme, Scripts et Sécurité

Un voyage à travers les commandes, Scripts et interactions avancées



Introduction du sujet

Pour ce projet, il est attendu un fichier nommé "[revision_systeme.pdf](#)" regroupant les différentes étapes appliquées dans chaque exercice, les lignes de commande utilisées avec une explication de leurs fonctionnements et les différentes recherches effectuées tout au long de cette semaine.



Création d'une VM Debian

À l'aide de l'outil de votre choix, créer une machine virtuelle Debian avec interface graphique. En fonction de votre processeur (Intel, AMD, M1 ...)
télécharger l'une des deux images disponibles :

→ [AMD64](#)

→ [ARM64](#)

Votre machine virtuelle doit avoir comme nom de session : **La_Plateforme** et
mot de passe : **LAPlateforme_**

Connecter votre machine à internet en modifiant tous les fichiers nécessaires.

Commandes de recherche avancée

Utilisez les lignes de commande pour créer cinq fichiers textes nommés
"**mon_texte.txt**" et assurez-vous qu'ils contiennent le texte suivant : « **Que la
force soit avec toi.** » .

Répartissez ces fichiers dans les répertoires suivants : "**Bureau**",
"**Documents**", "**Téléchargement**", "**Vidéos**" et "**Images**".

À partir du répertoire de votre session , utilisez le terminal et le mot "**force**"
pour localiser les cinq fichiers "**mon_texte.txt**".

Compression et décompression de fichiers

Créez un répertoire nommé "**Plateforme**" dans le dossier "Documents" de
votre session et ajoutez-y le fichier "**mon_texte.txt**" précédemment créé.

Dupliquer ce fichier **quatre fois** dans le même répertoire, formant ainsi un
total de cinq fichiers dans le répertoire "**Plateforme**".

Ensuite, archivez le répertoire "**Plateforme**" en utilisant les commandes "**tar**"
et "**gzip**". Explorez différentes options de compression lors de cette étape.



Décompressez les archives créées en utilisant les commandes appropriées. Explorez les diverses options de décompression.

Manipulation de texte

Pour cet exercice, votre mission consiste à utiliser la commande "**awk**" afin d'extraire des informations spécifiques d'un fichier CSV.

Pour commencer, créer un **script python** permettant la création d'un fichier CSV et l'ajout des données suivantes :

- Jean, 25 ans, Paris
- Marie, 30 ans, Lyon
- Pierre, 22 ans, Marseille
- Sophie, 35 ans, Toulouse

Une fois votre script python créé et fonctionnel, extraire les informations relatives aux villes de chaque personne en utilisant la commande "**awk**".

Manipulation de texte

Dans le but d'explorer les **processus**, commencez par recenser tous ceux qui sont actifs sur votre système.

Cherchez la commande permettant de fermer un processus, puis exécutez-la pour terminer un processus spécifique.

Dans certaines situations, il peut être nécessaire de terminer un processus de façon forcée. Cherchez et exécutez cette commande.

Comparez et observez les différences entre le processus de terminaison normale et celui effectué de manière forcée.



Surveillance des ressources système

Mettre en place une surveillance en temps réel de l'utilisation du CPU, de la mémoire et d'autres ressources système. Affichez les informations dans le terminal à l'aide de commandes appropriées.

Une fois que vous avez visualisé ces informations, enregistrez-les dans un fichier CSV. À vous de déterminer la méthode à utiliser pour accomplir cette opération.

Scripting avancé

Développer un **Script Shell** visant à automatiser la sauvegarde périodique du répertoire « **Plateforme** » créé précédemment.

Assurez-vous d'intégrer une fonctionnalité de gestion de **l'historique** des sauvegardes, permettant ainsi de conserver un **suivi chronologique** des opérations effectuées.

Automatisation des mises à jour logicielles

Créez un **script automatisant** la recherche de mise à jour des logiciels existants sur le système. Il doit offrir la possibilité à l'utilisateur de procéder à la mise à jour de ces logiciels.

Gestion des dépendances logicielles

Élaborez un script ayant pour objectif de simplifier l'installation et la gestion des dépendances logicielles pour un projet web, tout assurant la compatibilité entre les différentes versions.



Le script doit installer les éléments suivants :

- **Un serveur Web** (Apache ou Ngnix)
- **phpMyAdmin**
- **Un système de gestion de base de données relationnelle** (MySQL ou MariaDB)
- Un environnement JavaScript côté serveur (Node.js) avec **npm**
- Un système de contrôle de version (**git**)

Sécuriser ses scripts

Il est essentiel de garantir la **sécurité** de vos scripts. **Identifiez les risques** liés à la négligence de la sécurité de ceux-ci.

Procédez à la sécurisation des différents scripts développés précédemment.

Utilisation d'API Web dans un script

À l'aide d'un **Script Shell**, exploiter les données d'une **API Web** (celle de votre choix). Assurez-vous que la **communication** avec l'API se fasse de manière sécurée.

Inclure une **gestion des erreurs** afin d'anticiper tout comportement inattendu en cas de problème avec l'API.

Le **logging** est un concept important en cybersécurité. Renseignez-vous sur la **journalisation** (logging) et enregistrez les différentes requêtes vers l'API ainsi que les réponses envoyées par l'API.



Compétences visées

- Administrer et sécuriser les infrastructures systèmes
- Mettre en œuvre et optimiser la supervision des infrastructures
- Participer à la détection et au traitement des incidents de sécurité

Rendu

Le projet est à rendre sur :

<https://github.com/prenom-nom/SystemeScriptSecurite>

Pensez à mettre votre repository en **public**.

L'évaluation se fera sous forme de présentation sans support à l'équipe pédagogique.

Base de connaissances

- [Créer une VM avec une image disque](#)
- [Commandes Linux les plus utilisées](#)
- [Commande : AWK](#)
- [Gestion des processus](#)
- [Comment créer un CSV avec python ?](#)