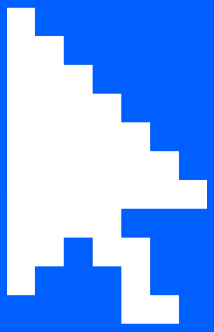
La Plateforme
La grande école du numérique pour tous

Rapport du Commandeur Data :

Sous-système PSMM (Python, Shell, Mariadb, Mail)
Objet : Centralisation des protocoles de gestion d'erreurs



Introduction du sujet

Le but est de récupérer les logs de serveurs FTP, SQL, Web et d'archiver les tentatives d'accès avec un compte / mot de passe non valide dans des bases SQL avec envois de mail journalier à l'administrateur système.
Pour cela, vous allez avoir besoin de trois serveurs : FTP, MariaDB, Web.

Job 01

Il va falloir créer 3 VM Debian (elles peuvent être reparties parmi les différents étudiants du groupe):
- Serveur FTP : 1 Go - 1 Vcpu – 8 Go HDD.
- Serveur Web : 1 Go - 1 Vcpu – 8 Go HDD (apache/nginx et site avec auth/basic).
- Serveur SQL : 2 Go - 2vcpu – 8 Go HDD

Pour les VM : les nommer du nom-de-votre-goupe_la-fonction
par exemple pour le groupe « nextgeneration» les VM seront nommées : nextgeneration_ftp, nextgeneration_mariadb, nextgeneration_web

Une fois tous les outils installés, placer ces VM dans le réseau de la plateforme

Pour des raisons de sécurités, sur ces trois serveurs l'accès SSH au compte root n'est pas autorisé, seul un compte « monitor » peut se connecter en SSH.

L'accès ne peut se faire qu'avec des clés SSH, et Le compte monitor doit faire partie du groupe sudo.

Job 02

Créer une VM Debian sans interface graphique, avec python, les outils MySQL/MariaDB (pas le serveur), client FTP, de quoi envoyer des mails en python.
Une fois tous les outils installés, placer cette VM dans le réseau de la plateforme

Job 03

Vous devez faire un script "**ssh_login.py**", qui depuis la VM , va se connecter sur un des serveurs et lancer une commande shell (par exemple «df» ou «ls» ou autre).

Job 04

En utilisant le script précédent, vous devez faire un script "**ssh_login_sudo.py**", qui depuis la VM, va se connecter sur un des serveurs et lancer une commande shell, cette fois en mode «sudo».

Job 05

Maintenant que l'on peut lancer des commandes shell en «sudo» , on va se connecter sur le serveur MariaDB/MySQL.
Vous devez faire un script "**ssh_mysql.py**", pour vérifier que vous avez bien accès au serveur MariaDB/MySQL.

Job 06

On va attaquer les choses sérieuses. À présent, il va falloir récupérer dans les fichiers log de MariaDB/MySQL, les tentatives d'accès depuis un compte / mot de passe incorrect, pour cela faites plusieurs accès avec des utilisateurs/MdP qui ne sont pas bons.
On va stocker ces « accès » dans votre base de données, avec le nom du compte utilisé pour l'accès, la date/heure, et l'IP du poste ayant tenté l'accès.
Vous devez écrire le script "**ssh_mysql_error.py**"

Job 07

On va désormais s'attaquer au serveur FTP, sur votre poste installer FileZilla et accéder au serveur FTP, et depuis la VM depuis un client FTP en CLI

Faites plusieurs accès avec des utilisateurs/MdP qui ne sont pas bons (pour que le serveur FTP génère des logs avec des erreurs d'accès).

Écrire le script “**ssh_ftp_error.py**”, pour qu’il récupère les logs d’erreur et les écrire dans votre base de données.

Job 08

On va maintenant s’attaquer au serveur Web. Le serveur Web est protégé par un compte utilisateur et mot de passe.

Faites plusieurs accès avec des utilisateurs / MdP qui ne sont pas bons (pour que le serveur Web génère des logs avec des erreurs d’accès).

Écrire le script “**ssh_web_error.py**”, pour récupérer les logs d’erreurs et les écrire dans votre base de données.

Job 09

Maintenant que l’on a créé nos bases SQL avec les erreurs d’accès, il va falloir avertir l’administrateur. Il faut écrire un script “**ssh_serveur_mail.py**”, qui va envoyer un mail à l'administrateur avec les historiques des tentatives de connexion de la veille.

Job 10

Faire une sauvegarde locale de votre base de donnée, horodaté, avec conservation de sept dernières sauvegardes, avec une planification toutes les 3h.
Il faut écrire un script “**ssh_cron_backup.py**”

Job 11

Maintenant, il va falloir récupérer l'état ressources système RAM / CPU / DISK des différents serveurs, les stocker dans une base de données et ne garder que les dernières 72h.
Il faut écrire un script "**ssh_system_status.py**"

Job 12

Reprendre le script précédent, en ajoutant l'envoie de mail à l'administrateur système si une des conditions suivante apparaît :

- le taux d'utilisation CPU dépasse 70 %
- le taux d'utilisation Disque dépasse 90 %
- le taux d'utilisation RAM dépasse 80 %

Ces valeurs doivent pouvoir être modifié facilement dans le script.

Il faut écrire un script "**ssh_system_mail.py**", prévoir en tache planifiée toutes les 5 min.

Job 13

Reprendre le script précédent (toujours en tache planifiée toutes les 5 min), et modifier pour l'envoi du mail : ne pas envoyer plus d'un mail par heure à l'administrateur.

Job 14

Maintenant que tous les scripts fonctionnent. Il faut penser à faire la mise à jour des différents serveurs.
Attention, les serveurs étant sur le réseau de la plateforme, il faut être connecté pour avoir accès à Internet.

Il faut écrire un script “**ssh_update.py**” , qui se connecte à alcasar , vérifie les mises à jour , s'il y en a les faits et vérifier qu’il n’y a pas besoin de redémarrer les serveurs après mise à jour , dans ce cas, envoyez un mail à l'administrateur système et ensuite se déconnecter d’alcasar.

Job 15

Maintenant que l’on sait envoyer des mails , en fonction des événements , on va « s'amuser » avec le chat de Google.
Vous devez créer un «Space» Google en ajoutant tous les membres de votre groupe , et en ajoutant votre accompagnateur Pédagogique .
Depuis un script en python ou en bash envoyer un message dans le chat qui détaille les événements ou l’état des serveurs périodiquement .

Compétences visées

- ➔ Administration système
- ➔ Virtualisation
- ➔ Scripting
- ➔ archivage
- ➔ logging

Rendu

Le projet est à rendre sur **https://github.com/prenom-nom/holodeck**

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

Base de connaissances

Python Mariadb

Proftp logging

FileZilla

Shell

Mail avec gmail

Nginx Autentification

Debian monitoring

Crontab

Gdrive