AYMERICK BRES
CONSULTANT IT - FORMATEUR

Applications PowerShell et

Rendu attendu
Une infrastructure fonctionnelle complète, démontrée et expliquée à votre intervenant pour notation à chaud.

Travail individuel – Durée 08h00

Objectifs de ce TP

L'objectif de ce TP est de vous familiariser avec le langage PowerShell, spécifique à Microsoft, bien qu'il soit utilisable sous Linux. Son principal objectif est l'automatisation des tâches sous Windows. Ce TP vise également à vous faire manipuler des machines virtuelles et le serveur web Apache2 couplé à PHP.

Enfin, vous découvrirez qu'il est possible de synchroniser des dépôts en ligne (par exemple, Mega.nz ou Google Drive) sur une infrastructure en ligne de commandes (CLI) pour y effectuer des sauvegardes.

Table des matières

Travail individuel – Durée 08h00

	Pour réaliser ce TP, vous devrez posséder le logiciel de virtualisation VirtualBox et l'utiliser. Assurez-vous d'avoir le logiciel sur votre poste, et d'être à l'aise dans son utilisation. Dans le cas contraire, n'hésitez pas à vous référer à la documentation du logiciel. Les fichiers .iso pour ce TP sont généralement fournis par votre intervenant.

Préambule

Une fois le logiciel VirtualBox installé et configuré (*Cf. l'encadré ci-dessus*), vous allez devoir créer une machine virtuelle sous Debian (*latest* de préférence où via l'ISO fourni par votre intervenant) et y installer les paquets Apache2 et PHP afin de transformer votre Debian en serveur Web complet.

	Une documentation concernant l'installation complète d'un Linux Debian vous est également fournie par votre intervenant. Je vous invite fortement à la consulter si vous réalisez une installation *from scratch* de votre machine virtuelle.

Ce TP s'articule autour de plusieurs étapes clés.

À la demande de votre directeur Pierre MARTINEZ, un développeur web, vous êtes chargé de déployer le site de gestion des utilisateurs qu'il vient d'achever. Le site, élaboré en HTML, CSS et PHP, nécessite une configuration soigneuse. En tirant parti de vos compétences, vous installez un serveur Apache2 associé à PHP sous Debian pour établir l'infrastructure du serveur web. Ensuite, vous positionnez le site conformément aux spécifications et effectuez des vérifications pour garantir son bon fonctionnement.

Quelques mois plus tard, vous faites face à un licenciement pour la raison apparemment farfelue de « consommation excessive de Mojito à la cafétéria le midi ». Percevant cette mesure comme abusive, vous choisissez de réagir en tentant de pirater, par force brute, le site web que vous avez mis en place quelques mois auparavant. Vous savez que votre directeur utilise une adresse « gmail.com ». Dans cet objectif, vous élaborerez un script PowerShell destiné à explorer différentes combinaisons de mots de passe.

Après plusieurs mois d'enquête, il s'avère que vos verres ne contenaient pas de Mojito, mais simplement du sirop de menthe. Vous êtes réintégré. Pour prévenir toute perte potentielle de son site web à l'avenir, votre directeur vous charge d'automatiser la sauvegarde en ligne de commandes sur un espace de stockage cloud tiers.

En dernière étape, vous élaborerez un script PowerShell dédié à la sauvegarde de vos documents sous Windows, et vous participerez à un jeu Google.

Travail individuel – Durée 08h00

Configuration de la VM sur votre outil de virtualisation

- **Nombre de VM pour ce TP : 1 (Linux Debian) ;**
- **RAM : 4 096 Go ;**
- **Disque : 20 Go dynamiquement alloué (précisez si SSD) ;**
- **CPU : 1 vCPU ;**
- **Network : Accès par pont en DHCP ;**

	Retenez bien le mot de passe que vous allez utiliser lors de la création de votre machine virtuelle. En cas d'oubli, une nouvelle VM sera à recréer… et le TP à recommencer du début… Sur VirtualBox, vous disposez d'une partie « Description ». Écrivez le mot de passe utilisé dans cette partie.

1.Installation web

1) Installation Apache2 et PHP

Partie cours

Apache2 est un serveur web open-source largement utilisé dans le monde entier. Réputé pour sa stabilité et sa flexibilité, il permet d'héberger des sites web de toutes tailles. Son architecture modulaire et sa prise en charge des langages de programmation dynamiques en font un choix privilégié pour de nombreux développeurs et administrateurs système.

PHP (Hypertext Preprocessor) est un langage de script côté serveur, conçu principalement pour le développement web. Il est utilisé pour créer des pages web dynamiques en générant du contenu qui s'adapte en fonction des interactions de l'utilisateur.

Durant tout le TP, en cas de difficulté, référez-vous à la documentation d'Apache2.

Toutes les commandes que vous allez utiliser, doivent être précédées de « sudo » qui permet de donner les droits superutilisateur lors de l'utilisation de la commande. Il n'est pas nécessaire d'utiliser « sudo » au début de chaque commande si vous travaillez en tant que « root ». Travailler en tant que « root » n'est pas une bonne pratique.

➢ **Mise à jour des paquets :**

Travail individuel – Durée 08h00

```
apt update -y && apt upgrade -y
```

➢ **Installation d'Apache2 :**

```
apt install apache2 -y
```

➢ **Vérification du statut d'Apache2 :**

```
systemctl status apache2
```

Si tout fonctionne, vous devriez avoir un résultat comme celui-ci :

```
root@tp-ps-svg:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-01-25 16:19:13 CET; 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 13379 (apache2)
      Tasks: 55 (limit: 4626)
     Memory: 9.3M
        CPU: 114ms
```

➢ **Vérifications post installation :**

Récupérer l'adresse IP de votre serveur Debian (`ip a`) et rentrer cette adresse IP dans la barre

d'adresse de votre navigateur : http://<IP_Debian>

Vous devriez avoir un rendu similaire :

Les fichiers de configuration principaux sont généralement dans le répertoire `/etc/apache2/`

Les fichiers concernant le site par défaut se trouvent dans `/var/www/html`

➢ **Installation de PHP :**

```
apt install php libapache2-mod-php -y
```


Travail individuel – Durée 08h00

➢ **Redémarrage d'Apache2 pour intégrer PHP :**

```
systemctl restart apache2
```

➢ **Vérification du statut d'Apache2 :**

```
systemctl status apache2
```

➢ **Testez l'installation :**

Créez un fichier de test PHP dans le répertoire racine de votre site web. Par exemple, créez un fichier nommé « **test.php** » avec le contenu suivant :

```
<?php phpinfo(); ?>
```

	Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano.

Récupérer l'adresse IP de votre serveur Debian (`ip a`) et rentrer cette adresse IP dans la barre d'adresse de votre navigateur : http://<IP_Debian>/test.php

Vous devriez avoir un rendu similaire :

Cela signifie que votre serveur PHP fonctionne, car le code PHP est interprété puis exécuté.
Si les caractères accentués ne sont pas interprétés, il faut que vous configuriez le « Charset » de votre site web sur « UTF-8 ».

2) Installation du site web

➢ **Dans `/var/www/` créez un dossier « gest_util » avec la commande `mkdir`. C'est dans ce dossier que vous allez déposer votre site web.**

Vous devriez avoir un rendu comme celui-ci :

Travail individuel – Durée 08h00

➢ **Dans ce nouveau dossier, copiez, à l'aide d'un serveur FTP/SFTP, SSH COPY ou avec nano en SSH, le contenu des 4 fichiers présents dans le dossier « site » donné par votre intervenant :**

- ☐ Index.css
- ☐ Index.php
- ☐ Login.css
- ☐ Login.php

L'utilisateur utilisé pour lire votre site web est « www-data ». **Attribuez-lui les documents.**

```
chown -R www-data:www-data /var/www/gest_util/*
```

3) Mise en place du fichier de configuration

Pour expliquer à Apache2 comment travailler avec vos ressources, vous allez devoir créer un fichier de configuration propre à chacun de vos sites web.

Partie cours

Sous Apache2, vos fichiers de configuration peuvent se trouver à 2 endroits dans les dossiers **sites-available** et **sites-enabled**.

Sites-available : Contient les fichiers de configuration des sites web disponibles sur votre serveur. Chaque fichier de configuration représente un site web ou un VirtualHost. Ces fichiers peuvent être activés ou désactivés en fonction des besoins.

Sites-enabled : contient des liens symboliques (sorte de raccourci sous Linux) vers les fichiers de configuration des sites web qui sont activés. Lorsque vous activez un site web en créant un lien symbolique vers son fichier de configuration depuis *sites-available* vers *sites-enabled*, Apache2 charge ce fichier de configuration lors du démarrage et commence à servir le site web correspondant.

Travail individuel – Durée 08h00

Voici la structure de base d'un fichier de configuration pour un site sous Apache2.

➢ **Créez le fichier « gest_util.conf » dans le dossier */etc/apache2/sites-available/* en partant de ce modèle. Adaptez-le en fonction de vos besoins !**

```
# Déclaration d'un VirtualHost pour un site web sur le port 80
(HTTP) <VirtualHost *:80>
 # Adresse e-mail de l'administrateur du serveur
 ServerAdmin webmaster@example.com

 # Nom et/ou adresse IP du serveur virtuel
 ServerName 192.168.1.67
 ServerAlias www.exemple.com exemple.com

 # Emplacement des fichiers du site web
 DocumentRoot /var/www/gest_util

 # Autoriser l'accès aux fichiers dans le répertoire DocumentRoot
<Directory /var/www/gest_util>
 Require all granted
 AllowOverride FileInfo
 </Directory>

 # Journalisation
 ErrorLog ${APACHE_LOG_DIR}/example_error.log
 CustomLog ${APACHE_LOG_DIR}/example_access.log combined
</VirtualHost>
```

	Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano. La commande **`apache2ctl configtest`** permet de vérifier la syntaxe correcte de votre fichier de configuration.

➢ **Créez maintenant un lien symbolique avec la commande `ln` depuis le dossier « sites available » vers « sites-enabled » pour demander à Apache d'activer votre site, c'est-à dire de le mettre en ligne :**

```
ln -s /etc/apache2/sites-available/gest_util.conf
/etc/apache2/sites enabled/
```

Voici le résultat attendu :

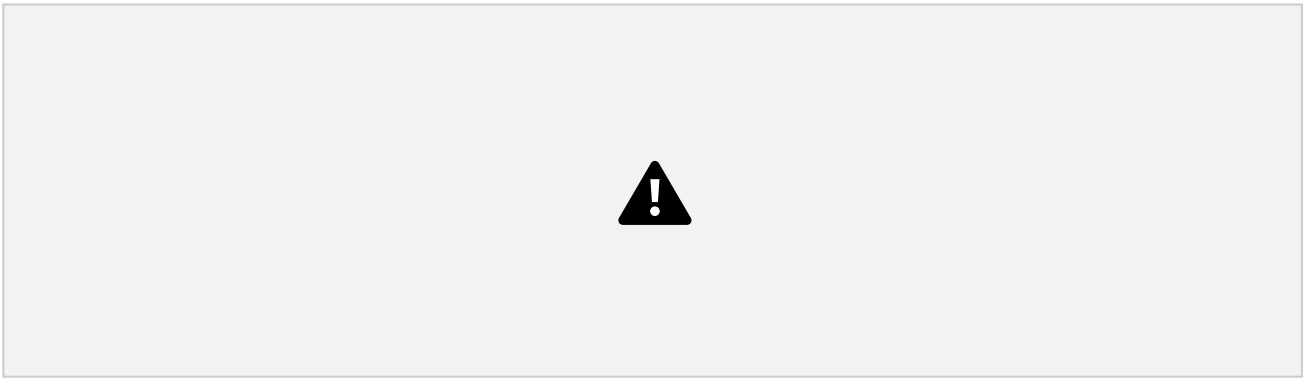
Travail individuel – Durée 08h00

➢ **Redémarrez votre serveur Apache :**

```
systemctl restart apache2
```

Récupérer l'adresse IP de votre serveur Debian (`ip a`) et rentrer cette adresse IP dans la barre d'adresse de votre navigateur : `http://<IP_Debian>`

Vous devriez avoir un rendu similaire :



Si c'est le cas, le site est fonctionnel. **Sinon Troubleshootez**.

2.Script de *Brute Force*

	Disclaimer : Art. 323-1 du CP : « *Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de trois ans d'emprisonnement et de 100 000 € d'amende.* » Le contenu de ce Travail Pratique est strictement éducatif afin de vous faire réfléchir sur les menaces existantes pour mieux vous en protéger. Votre intervenant n'est pas responsable des actions que vous mènerez en dehors du cours. Respectez les lois, obtenez des autorisations et utilisez vos compétences de manière éthique et légale.

➢ **Une fois l'installation du site web terminée, prenez quelques minutes pour <u>comprendre son fonctionnement et déterminer les angles d'attaques</u> possibles. Analysez le fonctionnement interne du code, les actions réalisées côté client et côté serveur, le fonctionnement et les dépendances des pages, les forces et les faiblesses du site…**

Travail individuel – Durée 08h00

	Jalonnement : Une fois ce travail effectué, appelez votre intervenant et discutez ensemble des angles d'attaques trouvés avant de passer à la suite du TP.

1) Réalisation du script de *Brute force* en PowerShell

	Dans ce contexte, l'utilisation des outils et intelligences artificielles doivent être utilisés au strict minimum. Je préfère être sollicité moi. Ce qui m'intéresse principalement est votre manière de faire, plus que le rendu final. Cela vous permettra également d'avoir des scripts différents et de les comparer.

Vous avez trouvé sur internet un dictionnaire de mots de passe couramment utilisés (fourni par votre intervenant : « dictionnaire.txt »). Vous avez comme très forte intuition que le mot de passe de votre directeur se trouve dans cette liste.

➢ **Réalisez un script PowerShell pour vous permettre de brut-forcer ce site web. Trouvez également l'adresse électronique de votre directeur.** Les formats les plus couramment utilisés sont « prénom.nom ».

2) Utilisation du script de *Brute force* en PowerShell

Cahier des charges du script PowerShell

- ☐ Doit prendre en paramètre (variable) l'adresse du site ;
- ☐ Doit prendre en paramètre le fichier « dictionnaire.txt » (Attention : syntaxe UNC) ;
- ☐ Doit charger la liste des mots de passe dans le dictionnaire en une liste PowerShell ;
- ☐ Doit tester chaque mot de passe, avec la bonne adresse électronique ;
- ☐ Doit indiquer le mot de passe testé en sortie standard, en gris si le mot de passe est erroné et en vert si le mot de passe est trouvé ! ;
- ☐ Doit attendre une seconde entre chaque test ;
- ☐ Attendre une action utilisateur avant de fermer la fenêtre.

Quelques commandes PowerShell pour vous aider à réaliser ce script dans le fichier « tuto.ps1 ».
Clic-droit → Modifier.

Optionnel :

Travail individuel – Durée 08h00

Pour jouer avec des mots de passe différents de vos collègues, modifiez la valeur de « **`sha1($_POST['password']) ==`** » dans le fichier « login.php » avec l'une des valeurs suivantes :

- ☐ f6c65efdbb0a998450b814f468ab64d1f0643bc5
- ☐ 98970771d37a3de5cf72779ed1405760ff815d8e
- ☐ 02a62a4b1edb94d6e4502aa3b035061082034dcc

- [ ] 25d54c0389c268b37fa5898d09f1f64375224e93
- [ ] 5dcfad9016fe0747e38871d047d9268da3735f41
- [ ] 01e31c8323f6659d63ef7c0c070b474817c0c30b

	Jalonnement : Une fois ce travail effectué et ce script fonctionnel, appelez votre intervenant et discutez ensemble des résultats obtenus avant de passer à la suite du TP.

3.Automatisation d'une sauvegarde informatique

1) Automatisation de la sauvegarde avec PowerShell

➢ **Vous allez devoir réaliser, dans cette partie, <u>un script PowerShell fonctionnant sous Windows</u>, dont le but va être d'effectuer une sauvegarde informatique des dossiers et des fichiers les plus importants de votre poste.**

	Dans ce contexte, l'utilisation des outils et intelligences artificielles doivent être utilisés au strict minimum. Je préfère être sollicité moi. Ce qui m'intéresse principalement est votre manière de faire, plus que le rendu final. Cela vous permettra également d'avoir des scripts différents et de les comparer.

<u>Cahier des charges du script PowerShell</u>

- [ ] Renseignez dans un tableau PowerShell le chemin des dossiers et fichiers que vous souhaitez sauvegarder ;
- [ ] Pour chaque élément de la liste, la copie doit s'effectuer et un message, de couleur bleue, doit apparaître en sortie standard ;
- [ ] La destination sera un dossier automatiquement créé au format « svg_*JJ-MM-AAAA* » lui même présent dans un dossier « Sauvegardes » créé sur votre poste par vos soins ;

Travail individuel – Durée 08h00

- [ ] Une fois votre script terminé, un message de fin doit apparaître en sortie standard et la fenêtre rester ouverte jusqu'à action de l'utilisateur ;
- [ ] Les erreurs doivent être gérées.

<u>Quelques commandes PowerShell pour vous aider à réaliser ce script dans le fichier « tuto.ps1 ».</u>
Clic-droit → Modifier.

	Jalonnement : Une fois ce travail effectué et ce script fonctionnel, appelez votre intervenant et discutez ensemble des résultats obtenus avant de passer à la suite du TP.

2) Envoi d'un fichier sur le Cloud en CLI

Le site https://mega.io (ou https://mega.nz) est un hébergeur cloud qui vous permet de stocker gratuitement **jusqu'à 20 Go de données**. Il s'agit d'un des sites sur le marché vous permettant de stocker le plus de données avec une offre gratuite.

➢ **Si vous n'en possédez pas déjà un, créez-vous un compte gratuit chez MEGA.**

Plusieurs outils ont été développés, comme le logiciel MEGA qui permet de faire de la synchronisation en miroir depuis votre PC comme le fait OneDrive de Microsoft par exemple.

Il possède également un autre atout fondamental, la possibilité de communiquer avec votre compte MEGA directement en ligne de commande, en installant MEGA CMD.

1. **Sélectionnez la distribution « Linux » puis la version « Debian » dans la liste déroulante. Copiez-collez les commandes données en fonction de la version de votre Debian.** 2. **Avec l'aide de la documentation officielle, faites une sauvegarde ponctuelle de votre site internet de la mission 1, normalement présent dans `/var/www/html`.**
3. **À l'aide d'une tâche CRONTAB, automatisez votre sauvegarde pour qu'elle ait lieu quotidiennement, sauf le week-end, à 02h30 du matin.**

La commande `mega-help -f` permet de vous aider dans l'utilisation de l'outil.

	Jalonnement : Avant de passer à la partie suivante, appelez votre intervenant pour un contrôle intermédiaire de votre travail réalisé jusqu'à présent. Assurez-vous que tout soit fonctionnel avant son arrivée.

Travail individuel – Durée 08h00

	Une fois ce TP terminé, gardez le bien en poche. Gardez également cette machine virtuelle avec Apache2 et le site web fonctionnel : nous nous en resservirons plus tard dans l'année. Si la VM de ce TP est supprimée ou perdue, ce TP sera à recommencer !

4.Jeu Google : INTERLAND

Le jeu Google INTERLAND est un jeu ludique destiné à une population néophyte en matière de cybersécurité.

Ce jeu vous fera découvrir 4 thèmes et vous permettra de générer des certificats de réussite le cas échéant. N'hésitez pas à utiliser vos écouteurs pour plus d'immersion dans le jeu.

Pour clôturer ce TP et ce chapitre, prenez un temps pour effectuer ce jeu, terminer les 4 niveaux, et générer les certificats de réussite à montrer à votre intervenant.