SÉCURISATION EXTRÊME

LABORATOIRE SEMAINE 5

Département d'informatique
Cégep de Jonquière

CONSIGNES

Dans cette activité, vous aurez à me programmer un petit programme python exploitant le Raspberry Pi et ses gadgets ! Vous aurez aussi, durant ce laboratoire, la chance de pouvoir utiliser les outils de collaboration en équipe.

MISE EN SITUATION – L'APOCALYPSE

PROLOGUE

Dans un monde alternatif au nôtre, à la suite de la Seconde Guerre mondiale et la découverte de l'énergie nucléaire, le monde est plongé dans une quête sans fin de progrès technologiques et scientifiques. Cette période voit des avancées technologiques rapides dans des domaines tels que la robotique, l'énergie nucléaire et l'informatique.

Malgré ces avancées, le monde commence à faire face à une crise énergétique majeure à cause de l'épuisement des ressources naturelles, principalement le pétrole. Ce problème pousse les puissances majeures de ce monde telles que les États-Unis, la Chine et la Russie à tenter par tous les moyens possibles de récupérer un maximum de ces ressources qui se font de plus en plus rares.

En 2066, la dernière guerre mondiale se déclenche lorsque la Chine envahit l'Alaska afin de s'emparer de ses gisements de pétrole. Cependant l'Alaska appartenant aux États-Unis, une guerre prolongée éclate entre les deux pays, mêlant ainsi beaucoup de pays dans le monde. Logiquement, l'économie mondiale perd pied aussi dans les semaines suivantes.

Le 23 octobre 2077, les tensions atteignent un point de non-retour et une guerre nucléaire totale éclate. La chine, les États-Unis, la Russie et tous les autres pays possédant l'arme nucléaire lancent toute leur puissance dans la bataille. Résultat : la terre se retrouve décimée et est totalement inhabitable à cause des radiations.

Retour dans le passé : Flashback (année 2076)

Nous sommes de retour du passé, un an avant la catastrophe nucléaire. Vous êtes encore un informaticien et le gouvernement des États-Unis fait appel à vous et SEULEMENT À VOUS afin de travailler sur un projet TOP SECRET.

Ce projet a pour but de créer une interface web sécurisé afin de faire le lancement nucléaire. Ainsi, le président pourra déclencher les bombes nucléaires en cas de besoin depuis n'importe où dans le monde.

Le projet a déjà été commencé par un autre informaticien anglais mais celui-ci a été assassiné par l'armée russe.

Vous aurez dans ce projet, à :

- Créer un serveur Web sécurisé (HTTPS)
- Protéger l'application existante contre les injections SQL
- Instaurer un nombre de tentatives maximales lors de la connexion
- Mettre en place un système de rôle dans l'application.
- Faire la page de lancement nucléaire et la page de consultation pour les autres utilisateurs non-administrateurs.

CONSIGNES

Étape A. Création du GitHub (sur votre VM)

Sur n'importe quel ordinateur, téléchargez le projet de départ fourni par l'enseignant. Ensuite, créez un environnement GIT dans lequel vous pourrez travailler en équipe.

1. Créer le répertoire GIT et liez-le à votre GitHub.

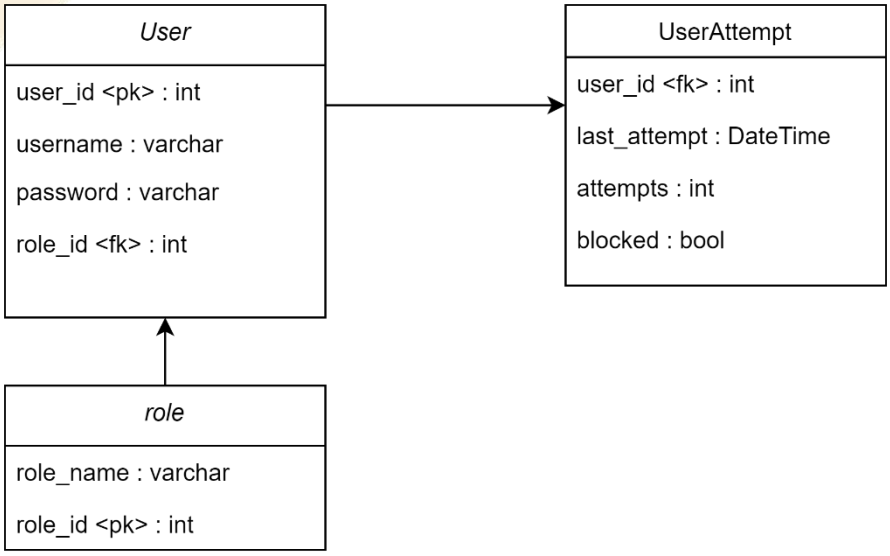
2. Créez-vous une branche de développement.

3. Commencez votre développement en utilisant les concepts de Git vue en classe. Vous pouvez programmez le projet sur n'importe quel ordinateur nous verrons plus tard comment le migrer sur votre Raspberry PI.

Étape B. Importation de la base de données de départ (sur votre VM)

Votre enseignant vous aura fournie dans le projet, un fichier SQL contenant la base de données de base, vous aurez à compléter le reste de script.

1. Modifier le script afin de le faire correspondre à l'image suivante. Vous devrez donc créer les tables de « rôles » et de « attempts » pour sécuriser votre application :



2. Une fois votre script terminé, importez le dans phpMyAdmin.

Étape C. Sécurité #1) implémentation des rôles dans l'application (sur votre VM)

La prochaine étape pour sécuriser l'application sera d'instaurer le système de rôle. Pour ce faire, faites en sorte d'avoir 2 rôles dans votre BD :

- Admin
- Standard

Par la suite, faites aussi en sorte d'avoir 3 utilisateurs dans votre BD :

- Username: **president**, Password: **Cegep123**, role: **Admin**
- Username: **worker**, Password: **Cegep**, role: **Standard**
- Username: **population**, Password: **Cegep**, role: **Standard**
-

Une fois votre BD remplie de ces données, il vous faudra modifier votre application afin de modifier le contenu de la page selon le rôle.

En effet, une fois la connexion effectuée, vous êtes automatiquement dirigé vers la seule et unique page, la page « accueil ». Modifiez-le contenue de la page suivante :

Peut-importe le type d'utilisateur, la page doit afficher « bienvenue nom_utilisateur » « vous êtes connecté »

Cependant dépendamment du rôle vous devrez afficher :

Admin : Faites afficher un bouton de lancement nucléaire, si jamais le bouton est cliqué il devra renseigner dans la BD le fait que les bombes ont été lancées et à quel moment exactement. Le bouton doit être inutilisable dès que les bombes sont lancées. (Vous devrez donc modifier votre bd)

Standard : Faites simplement afficher si oui ou non, les bombes ont été lancées. Si jamais elles ont été lancées, afficher aussi depuis combien de temps elles l'ont été.

Étape D. Sécurité #2) implémentations de la connexion sécurisée (sur votre VM)

L'application se doit d'être sécurisé ! C'est pourquoi vous devrez instaurer un système de connexion plus robuste. Pour ce faire, utilisez vos connaissances acquises en cours afin de bloquer les utilisateurs selon les paramètres suivants :

- Si l'utilisateur essaie de se connecter plus de 5 fois incorrectement, il est bloqué pendant 15 minutes.
- Le nombre de tentatives est réinitialisé toutes les 15 minutes depuis la dernière tentative.
- Si l'utilisateur parvient à se connecter, les tentatives sont remises à zéro.

Utilisez les concepts vus en classe et surtout votre logique afin de faire un système de connexion sécurisé.

Étape E. Sécurité #3) sécurisations contre les injections SQL (sur votre VM)

Le site de base utilisait des requêtes non sécurisées, assurez-vous que toutes vos requêtes SQL depuis le PHP sont maintenant sécurisées.

Pour ce faire, utiliser les concepts vus en classe afin de sécuriser toutes vos requêtes faites à la BD depuis votre application web.

Étape F. Mise en production de la version fonctionnelle sur GIT (sur votre VM)

Votre projet est terminé d'un point de vue « programmation ». Il ne vous reste donc qu'à mettre ce site sur un serveur Web réel et de sécuriser ce serveur. Avant tout, puisque votre développement est terminé, vous pouvez faire une version officielle. Faites donc en sorte d'envoyer le contenu de votre branche de DEV dans le MAIN.

Étape G. Exportation de la base de données depuis votre VM

Pour cette étape, vous devrez faire de l'exploration technologique et faire des recherches internet afin de trouver comment exporter votre base de données en un script. Vous en aurez besoin pour avoir cette base de données sur votre serveur de production (CLIENT) sur le Raspberry PI.

Installation du serveur Web (apache) et de la base de données sur le PI

- **AVANT TOUT FAIRE VOTRE UPDATE ET UPGRADE !!!!!!!!**
- Installez apache qui sera le serveur web qui hébergera votre site PHP

```
sudo apt install apache2 -y
```

- Installez PHP :

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

- Installez MySQL et mettez un mot de passe a « root » lors de la config :

```
sudo apt install mariadb-server mariadb-client -y

sudo mysql_secure_installation
```

- Installer PHP-MY-ADMIN:

```
sudo apt install phpmyadmin -y
```

- Vous pouvez avoir accès à phpMyAdmin (127.0.0.1/phpmyadmin) avec l'user root et le mot de passe que vous avez choisi

Étape H. Importation due de la base de données

- Vous pouvez avoir accès à phpMyAdmin (127.0.0.1/phpmyadmin) avec l'user root et le mot de passe que vous avez choisi.
- Importez simplement votre script de base de données et assurez-vous d'importer aussi vos données.

Étape I. Importation du projet sur le Raspberry PI

Sur votre raspberry Pi, rendez-vous dans le fichier **var/www/HTML.** Tout comme dans votre cours de PHP c'est dans ce fichier que vous hébergerez vos sites.

Cependant vous devez donner les droits à ce dossier :

```
cd /var/www

sudo chmod -R 777 .
```

- Relancez apache afin d'être sûr que la configuration a été prise en compte.

```
sudo systemctl restart apache2
```

- Clonez ensuite votre projet GIT dans le dossier /var/www/HTML. Vous aurez donc par la suite votre projet en version MAIN dans un dossier dans « www /HTML »
- Vous devriez donc avoir accès à votre site avec « 127.0.0.1/dossier_git/acceuil.php »

Étape J. Testez votre serveur depuis un autre ordinateur

Pour connaitre le ip de votre serveur, entrez la commande :

```
Hostname -I
```

Vous pouvez ensuite utiliser cet ip depuis un autre ordinateur pour avoir accès au site
(Ex : 192.168.2.52/dossier_git/accueil.php)

Étape K. Sécurité #4) Implémentation du serveur web sécurisé avec HTTPS

- Installez le générateur de certificat SSL auto-signée OPENSSL (gratuit) :

```
sudo mkdir /etc/ssl/private

sudo chmod 777 /etc/ssl/private .
```

- Générez votre paire de clé et mettez le mot de passe par la suite dans le passphrase :

```
sudo openssl genpkey -algorithm RSA -out
/etc/ssl/private/server.key -aes256
```

- Générez ensuite le certificat SSL autosigné avec la clé privée crée précédemment :

```
sudo openssl req -new -x509 -key /etc/ssl/private/server.key -out
/etc/ssl/certs/server.crt -days 365
```

- Activez les modules HTTPS sur les Raspberry Pi :

```
sudo a2enmod ssl

sudo a2enmod headers
```

- Configurer le fichier de configuration pour accepter les connexion HTTPS. Allez donc modifier le fichier suivant :

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

- Ajoutez les lignes suivantes au fichier :

```
<VirtualHost *:443>

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/HTML



    SSLEngine on

    SSLCertificateFile /etc/ssl/certs/server.crt

    SSLCertificateKeyFile /etc/ssl/private/server.key
```

```
    <Directory /var/www/HTML>
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # Optionnel : Forcer HTTPS
    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteCond %{SERVER_PORT} !^443$
        RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
    </IfModule>
</VirtualHost>
```

- Activez le site HTTPS et redémarrez apache :

```
sudo a2ensite 000-default.conf

sudo systemctl restart apache2
```

- Vous pourrez donc avoir accès au site web sur un serveur web sécurisé HTTPS en ajoutant https devant l'ip du Raspberry Pi (ex : https://192.168.52/dossier_git/acceuil.php

ÉVALUATION :

L'évaluation de ce laboratoire se fait par la complétion des étapes suivantes :

- **Étape E = 20%**
- **Étape F = 40%**
- **Étape G = 70%**
- **Étape H = 90%**
- **Étape I = 100%**

Faites vérifier le projet à la fin par le professeur il pourra regarder si chaque étape à bien été réalisé.

BONNE CHANCE !!!