STI

Sécurité des Technologies Internet

Projet 2

Application de messagerie sécurisée

Aspect sécuritaire (et fonctionnel)

Professeur	Assistant	Assistante
Abraham Rubinstein	Yann Lederrey	Lucie Steiner
abraham.rubinstein@heig-vd.ch	yann.lederrey@heig-vd.ch	lucie.steiner@heig-vd.ch

septembre 2019 – février 2020

Nom, prénom : _____

Nom, prénom : _____

Table des matières

1. Objectif

L'objectif de ce projet est de s'assurer que l'étudiant ait acquis les connaissances en développement Web **sécurisé** suffisantes pour mener à terme une analyse de menaces ainsi que pour identifier et corriger les failles dans un code. Ce projet consiste donc pour l'étudiant à analyser et sécuriser par lui-même une **application Web**.

Ce travail fait suite au projet 1. Pour rappel, ce dernier demandait de réaliser l'aspect fonctionnel d'une application de messagerie Web pour entreprise. Ainsi, le but consiste à reprendre l'application précédemment développée et effectuer une analyse de menaces complète. Cela permettra ensuite d'apporter les aspects sécuritaires manquant à l'application.

Le cadre de ce projet se résume à la sécurisation de l'**applicatif uniquement**. La sécurisation du serveur web (configuration Apache, modules, HTTPS, etc) ou la sécurisation de la machine (OS, VM, etc) ne font pas parti du cadre de ce projet et ne seront donc pas évalués. Vous avez pourtant la possibilité de faire des commentaires et recommandations sur votre rapport concernant la sécurité de l'infrastructure ou de procéder à sa sécurisation si le temps le permet.

Ce travail sera réalisé par groupes de 2 étudiants et **la constitution des groupes doit être différente par rapport aux groupes du projet 1**. De plus, chaque groupe travaillera sur une application Web différente de celle développée durant le projet 1.

2. Exceptions

Si un groupe souhaite utiliser une autre technologie ou encore modifier n'importe quel élément de la donnée, il est invité à prendre contact avec le professeur responsable afin d'en discuter.

3. Technologies à utiliser

3.1. Environnement

- Docker (votre application sera livrée dans une image)
- PHP ou le language de votre choix (PHP hautement conseillé… vous pouvez choisir des vieilles versions de PHP/autres pour simuler des conditions adverses aux développeurs)
- SQLite (éventuellement MySQL, mais SQLite est hautement conseillé pour simplifier le travail)

  Si d'autres librairies ou technologies doivent être utilisées, elles doivent être validées par le professeur (Bootstrap et autres sont autorisés pour embellir l'application. Pourtant, seul l'aspect fonctionnel à un impact sur la note)

3.2. Validation

De préférence, l'ensemble de l'application devra fonctionner sur une image Docker. Toutefois, sur autorisation du professeur, d'autres supports (MAMP, WAMP, etc) peuvent également être acceptés. Dans ces cas-là, l'environnement d'exécution devra être particulièrement bien détaillé dans le manuel afin qu'il puisse être facilement reproduit.

4. Cahier des charges

L'application finale sécurisée devra comporter les mêmes fonctionnalités que sa version non-sécurisée. De ce fait, le cahier des charges **du point de vue fonctionnel** reste le même que pour le projet 1.

4.1. Définition globale

L'application doit permettre la mise en œuvre d'une messagerie électronique au sein d'une entreprise. Cette messagerie sera une application Web uniquement et se base sur l'interaction avec une base de données (pas de SMTP ou autres protocoles de communication).

4.2. Authentification

Une authentification simple sera nécessaire afin d'accéder à l'application. Seule la page de login sera accessible sans être authentifié.

4.3. Rôles et authentification

L'application devra proposer deux rôles différents :

- Collaborateur
- Administrateur

Un mécanisme d'authentification simple (utilisateur – mot de passe) devra permettre d'accéder aux fonctionnalités. Pour pouvoir se connecter, un utilisateur devra être défini comme « actif ». Les fonctionnalités détaillées pour chaque rôle sont définies plus loin dans ce document.

4.4. Navigation

Il devra être aisé de naviguer d'une page à l'autre, via des liens ou boutons.

4.5. Fonctionnalités

Un **collaborateur** aura accès aux fonctions suivantes :

- Lecture des messages reçus : une liste, triée par date de réception, affichera les informations suivantes :
    - Date de réception
    - Expéditeur
    - Sujet
    - Bouton permettant la réponse au message
    - Bouton permettant la suppression du message
    - Bouton permettant d'ouvrir les détails du message
        - Devra permettre l'affichage des mêmes informations/options que ci-dessus, avec le corps du message en plus
- Ecrire un nouveau message : rédaction d'un nouveau message à l'attention d'un autre utilisateur. Les informations suivantes devront être fournies :
    - Destinataire (unique)
    - Sujet
    - Corps du message
- Changement du mot de passe : afin de pouvoir modifier son propre mot de passe

Un **administrateur** aura accès aux fonctions suivantes :

- Doit avoir les mêmes fonctionnalités qu'un collaborateur, en plus des suivantes
- Ajout / Modification / Suppression d'un utilisateur : un utilisateur est représenté par :
    - Un login (non modifiable)
    - Un mot de passe (modifiable)
    - Une validité (booléen, modifiable), actif ou inactif
    - Un rôle (modifiable)

5. Evaluation

Chaque rendu sera évalué sur la base des critères suivants :

- Qualité du rendu
  - Respect des consignes (délai, etc.)
  - Présence de tous les éléments
  - Installation/utilisation aisée
- Le rapport de l'analyse de menaces
  - Qualité de l'analyse (contenu, biens, scénarios, etc… c.f. cours),
  - Qualité rédactionnelle (présentation, structure, clarté, orthographe, etc.)
- Le manuel (peut être un simple README)
  - Qualité du contenu (complet, précis)
  - Qualité rédactionnelle (présentation, structure, clarté, orthographe, etc.)
- Les aspects fonctionnels de l'application
  - Similaire au rendu 1 – la sécurité ne doit pas « casser » les fonctionnalités
  - Fonctionnalités du cahier des charges
  - Appréciation du code
- Les aspects sécuritaires de l'application
  - Appréciation sécurité
  - Méthodologie sécurité
- Présentations
  - Qualité du contenu (sujet, détails, précision, maîtrise du sujet)
  - Qualité rédactionnelle (présentation, structure, clarté, orthographe, etc.)

Ces critères sont donnés à titre indicatifs. Ils peuvent être modifiés (ajoutés, supprimés, modifiés) . Leurs pondérations peuvent être adaptées en tout moment.

6. Rendu

6.1. A rendre

- **Rapport de l'étude de menaces**
  Etude complète selon le cours, y compris les mécanismes de sécurité.
- **Code de l'application**
  Fichiers (php, html, images, etc) dans un repo GitHub. L'application doit être sécurisée au maximum.
- **Base de données**
  Si nécessaire, pour accélérer son déploiement/utilisation.
- **Manuel**
  Document permettant l'installation/lancement/utilisation de l'application sur la machine virtuelle de référence. Eventuellement les noms d'utilisateur et mots de passe nécessaires. Comme pour le projet 1, un simple README suffit, mais il doit couvrir tous les besoins.
- **Présentation (max 15 min)**
  Elle devra couvrir au minimum les éléments suivants :
  - Introduction, **répartition du travail**
  - Analyse de menaces, mécanismes de sécurité
  - Implémentation générale (fonctionnelle)
  - Implémentation des mécanismes de sécurité
  - Problèmes rencontrés, conclusion

6.2. Échéances

L'ensemble du travail doit être rendu au plus tard le **mercredi 15 janvier 2020 à 23h59**. Le rendu se fera à l'aide d'un email destiné au professeur et à l'assistant indiquant le lien GitHub et les noms des deux personnes de l'équipe de travail. Les présentations auront lieu **du 17 au 24 janvier 2020** (env 15 minutes par groupe, interventions équitables).