Rapport d'étude de menaces

Projet de sécurité des technologies internet

Jean-Luc Blanc
Rosy-Laure Wonjamouna

Prof. Abraham Rubinstein Scharf
Assist. Stéphane Teixeira Carvalho

heig-vd
HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

January 20, 2022

Table des matières

1 Introduction

Dans le cadre de ce projet, nous sommes amenés à récupérer le précédent "Projet 1" se trouvant sur ce répo : https://github.com/kayoumido/STI_P1

Nous avons forké ce répo afin d'effectuer nos travaux. Nous ferons en premier une description du système, puis nous analyserons les éventuelles failles de sécurité ainsi que les menaces. Nous parcourons ensuite les différents scénarios d'attaques permettant d'exploiter ces failles. Finalement nous proposerons et appliquerons des contre-mesures afin de sécuriser l'application.

2 Description du système

2.1 Objectifs du système

On va faire l'hypothèse ici que nous utilisons notre système de messagerie dans le cadre d'une communication interne entre les membres d'une entreprise. On peut ainsi admettre que les administrateurs sont des membres de l'entreprise ayant un rang plus élevé (managers, direction) dans l'entreprise qui pourrait communiquer avec les employés un peu plus bas (employé) dans la hiérarchie: ce serait les collaborateurs. Les collaborateurs pourraient également communiquer entre eux, et avec les administrateurs.

- **Objectif principal**: Gérer la communication interne de l'entreprise
- **Réputation**: Une entreprise qui communique bien avec ses employés est une entreprise qui aura une bonne réputation de ce fait. Ca apporte une meilleure culture d'entreprise et des employés plus satisfaits et productifs
- **Financier**: Des documents importants relevant d'un certain secret industriel ou commercial peuvent être échangés au sein de l'entreprise. Le risque de perte financière pourrait être considérable s'il tombait aux mains d'un concurrent, ou d'un journaliste (si on est Monsanto et qu'on fait des choses pas très nettes, on pourrait tomber dans un scandale et perdre beaucoup d'argent)

2.2 Hypothèses de sécurité

- Le système de messagerie est accessible depuis le web par tous. Il faut néanmoins avoir un compte pour y accéder
- On peut supposer que les administrateurs comme les collaborateurs bénéficieraient d'un niveau de confiance faible. En effet, il pourrait y avoir des infiltrés dans l'entreprise par exemple
- On peut estimer le réseau interne de l'entreprise comme étant de confiance
- On fait aussi l'hypothèse que le système d'exploitation et le serveur web sur lequel tourne l'application sont de confiance

2.3 Exigences de sécurité

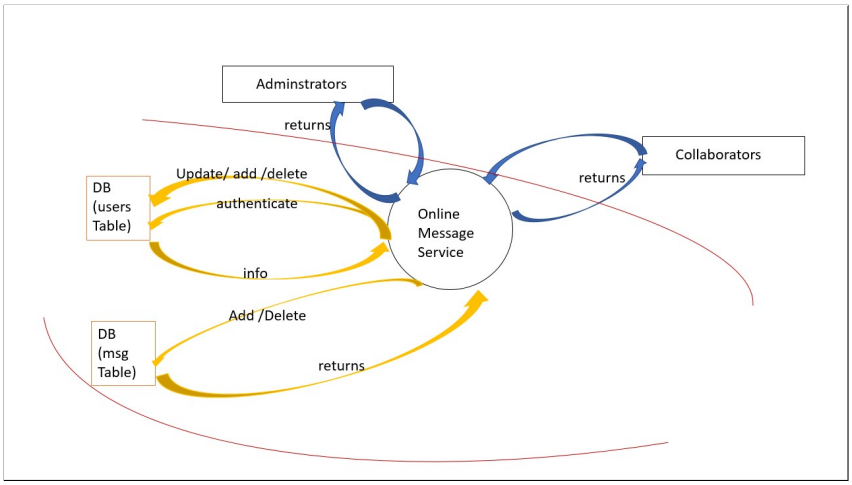
- Le contenu des messages échangés doit être protégé en intégrité, et non modifiable (fiabilité des informations échangées)
- Le contenu des messages ne doit pas être accessible à une personne extérieure à l'entreprise (confidentialité des données)
- Le site Web doit être disponible à 99% du temps (disponibilité)
- Les informations des utilisateurs doivent être protégées (protection de la vie privée)
- Les collaborateurs ne doivent pas avoir accès aux mêmes fonctionnalités que les administrateurs (contrôle d'accès)

2.4 Actifs à haute valeur

Quels sont les actifs à haute valeur du système?

- Base de données utilisateurs
  - confidentialité, privacy
  - un incident où les noms des collaborateurs seraient révélés nuirait à la confiance que les employés ont en leur entreprise et sa capacité à les protéger. L'entreprise perdrait sa réputation
- Base de données messages
  - confidentialité, intégrité
  - un incident, où les messages seraient révélés, pourrait faire perdre sa réputation à l'entreprise, si des secrets industriels ou commerciaux sont révélés, l'entreprise pourrait perdre de l'argent
  - un incident, où les messages seraient modifiés, pourrait également être grave pour le fonctionnement de l'entreprise.
- Infrastructure
  - disponibilité
  - un incident nuirait à la disponibilité de la plateforme

2.5 DFD



3 Identification des sources de menaces

3.1 Hackers, script-kiddies

- Motivation : S'amuser, gloire
- Cible : N'importe quel actif
- Potentialité : moyenne

3.2 Cybercrime

- Motivation : financière
- Cible : vol de credentials des utilisateurs de l'entreprise, vol de données sensibles contenues dans les messages, spam des employés
- Potentialité : moyenne

3.3 Utilisateur au sein de l'entreprise, lanceur d'alerte

- Motivation : Nuire à l'entreprise, lancer l'alerte sur des pratiques douteuses de l'entreprise
- Cible : contenu des messages
- Potentialité : moyenne

3.4 Concurrent

- Motivation : vol de secret industriels ou commerciaux
- Cible : contenu des messages
- Potentialité : moyenne

4 Identification des scénarios d'attaque

Pour imaginer les différents scénarios d'attaque on va s'aider du modèle STRIDE et de la DFD.

Composant	S	T	R	I	D	E
Administrateur	X		X			X
Employé	X		X			X
BD messages		X		X	X	
BD utilisateurs	X	X	X	X	X	X
Application web		X	X		X	

On va essayer de couvrir chaque élément du STRIDE;

4.1 Scénario de menace 0: A partir d'un compte collaborateur, un attaquant arrive à supprimer des comptes d'utilisateurs

- Business impact : Moyen (réputation)
- Source du risque : Collaborateur mécontent, Hacker
- Motivation : challenge, nuire à un autre collaborateur
- Actif cible : Base de données utilisateur
- Scénarios d'attaque
  Pour modifier les credentials, il faut être administrateur !
  - Scénario 1 : Autorisation bypassée : Le collaborateur arrive à accéder à une page administrateur via un URL direct sans s'authentifier en tant qu'administrateur.
  - Scénario 2 : Attaque de type CSRF : un collaborateur envoie une attaque de type CSRF à un administrateur. Le script envoyé à l'administrateur déclenche la suppression d'un compte utilisateur
  - Scénario 3 : Injection de code (SQL injection) : Le collaborateur a accès à la BDD via la page de login en faisant une injection de code SQL. Il injecte du code permettant de supprimer un compte.
  - Scénario 4 : Injection de code (XSS injection, file injection) : Le collaborateur injecte un script malveillant dans un message adressé à un administrateur. Le script s'exécute dans le navigateur de l'administrateur. Le code permettait la suppression d'un compte.
- Contrôles
  - Contrôles d'accès renforcés
  - Validation des inputs à tous les points d'entrées
  - Protections anti-CSRF (Tokens anti-CSRF)
  - Chiffrement des données dans la BDD

4.2 Scénario de menace 1: Vol de messages dans la base de données

- Business impact : élevé (réputation et perte de documents industriels ou commerciaux)
- Source du risque : compétition, cybercriminel
- Motivation : financière
- Actif cible : Base de données (table des messages)
- Scénarios d'attaque
  Un attaquant vole des messages échangés entre les utilisateurs qui se trouvent dans la base de données.
  - Scénario 1 : Vol de credentials : Les credentials d'un utilisateur sont devinés/brute forcés ce qui mène à la pénétration de son compte par l'attaquant, il peut alors lire ces messages.
  - Scénario 2 : Hacking de la base de données depuis le réseau interne via Nmap et Metasploit par exemple: Le réseau interne est compromis, l'intrusion dans la base de données est locale.

  - Scénario 3 : Injection de code sur la page de login (SQL injection, file injection ...)
  - Scénario 4 : Accès direct via un URL : L'URL d'un message a été deviné par l'attaquant : Par exemple, le message est trouvable et accessible via une query string évidente : parameters?user=admin&messageid=1

- Contrôles
  - Validation des inputs à tous les points d'entrées
  - Hardening du système, mises à jour régulières
  - Politique de mots de passe renforcée
  - Protection contre le brute-force
  - Authentification forte
  - URL non devinable avec contrôle d'accès renforcé

4.3 Scénario de menace 2: Modification des messages dans la base de données

- Business impact : moyen
- Source du risque : hackers
- Motivation : challenge
- Actif cible : Base de données (table des messages)
- Scénarios d'attaque
  Un attaquant pénètre la BDD et modifie les messages dans la base de données alors que même les utilisateurs ne peuvent pas modifier ou supprimer des messages une fois qu'ils ont été envoyés
  - Scénario 1 : Injection de code (SQL injection) : Le collaborateur a accès à la BDD via la page de login en faisant une injection de code SQL. Il injecte du code permettant de supprimer un compte.
  - Scénario 2 : Hacking de la base de données depuis le réseau interne via Nmap et Metasploit par exemple: le réseau interne est compromis, l'intrusion dans la base de données est locale.
- Contrôles
  - Validation des inputs à tous les points d'entrées
  - Hardening du système, mises à jour régulières du système
  - Ajout de logs pour monitorer ce qui se passe et détecter toute anomalie
  - Contrôles d'accès renforcés à la base de données

4.4 Scénario de menace 3: La liste des credentials des utilisateurs est révélée

- Business impact :élevé (réputation et financier)
- Source du risque : Cybercriminel
- Motivation : challenge, nuire, obtenir de l'argent
- Actif cible : Base de données (Table des utilisateurs)
- Scénarios d'attaque :

  - Les mêmes que pour la modification des messages dans la base de données

- Contrôles
  - Contrôles d'accès renforcés à la base de données
  - Chiffrement des données (mot de passe et nom d'utilisateur) contenues dans la base de données

4.5 Autres scénarios de menaces

D'autres scénarios sont imaginables cependant les contrôles qu'on pourrait imaginer pour limiter la réalisation de ces scénarios, auront déjà été implémentés, grâce aux scénarios précédents. On peut tout de même les lister ici :

- S'authentifier dans l'application avec des credentials volés
- Un utilisateur s'authentifie de nombreuses fois pour faire du DoS
- Un malware est introduit dans le réseau de l'entreprise

5 Contremesures

Dans cette section, on va rentrer un peu plus en détail sur les contremesures à implémenter pour protéger la web application contre les attaques.

5.1 Contremesures spécifiques contre le vol de credentials

Contre le vol de credentials, on peut mettre en place plusieurs défenses:

- Mettre un message d'erreur général qui ne précise pas si c'est le mot de passe qui est faux ou si c'est le nom d'utilisateur qui est faux. Cela permet de ne pas donner d'indices à un attaquant potentiel
- Mettre en place une politique de mot de passe forte comme :
  - Un mot de passe doit avoir au moins 8 caractères
- Protections contre le brute-force :
  - Mise en place d'un système de CAPTCHA
  - Limitation du nombre de tentatives d'ac
  - Un mot de passe doit contenir au moins un chiffre, une lettre majuscule, une lettre minuscule et un caractère spécial cès à un même compte en un temps donné. Au bout d'un certain nombre de tenatatives, on pourrait désactiver le compte par exemple.
  - Système d'authentification forte (2FA-Two Factor Authentication) mis en place après un certain nombre de tentatives d'accès à un même compte

5.2 Contremesures spécifiques aux injections de code

Contre les injections de code, on peut mettre en place une défense à plusieurs niveaux :

- Validation des inputs à tous les points d'entrées
  1. La validation des inputs peut se faire en vérifiant le format attendu dans le champ (date, age, zip code). On peut le faire en vérifiant l'input type de la donnée ou en utilisant des expressions régulières.

2. Si seulement un ensemble de valeur est attendu, on peut comparer la donnée entrée à chaque valeur de l'ensemble

- Utilisation de SQL queries parametrisées
- Faire de l'escaping
- Utilisation de procédures stockées

A noter qu'en PHP, l'utilisation de requêtes dites "requêtes préparées" couvrent la plupart de ses défenses.

5.3 Contremesures spécifiques aux attaques de type CSRF

- La réauthenticfication de l'utilisateur à chaque manipulation sensible qu'il se prépare à effectuer.
- La génération de tokens anti-CSRF aléatoires présents dans le "body" et le "cookie header" de la requête.

6 Nos choix d'implémentation

Dans cette section, nous allons présenter brièvement nos divers choix d'implémentation pour effectuer des contremesures aux failles de sécurité trouvées.

6.1 Algorithme de hashage

Dans l'application de messagerie de base, l'algorithme MD5 était utilisé afin de hashé les mots de passe, cet algorithme est complètement déprécié et n'est plus dans les normes de sécurité. A la place, nous avons implémenté donc la solution de la méthode "password_hash", intégrée nativement à PHP. Afin de vérifier le hash du mot de passe lors d'une connexion, on emploiera donc la méthode native "password_verify".

6.2 CSRF

Pour protéger notre application web d'une attaque CSRF, nous avons intégré à notre projet l'outil "CSRF Guard", une solution proposée par l'OWASP qui prend la forme d'une librairie, elle permet d'activer une protection anti-CSRF à l'aide d'un token.

7 Conclusion

Ce rapport atteste du travail d'identification des menaces contenant un certain risque envers notre application web de messagerie. Nous avons élaboré différents scénarios d'attaques possibles ainsi que des contre-mesures. Ces dernières ont été implémentées afin d'augmenter le niveau de sécurité de la messagerie. Cette liste de scénarios n'est pas exhaustive, on pourrait imaginer d'autres scénarios, comme un scan du réseau du serveur ou bien une attaque "ClickJacking".