

STI - Projet

Modélisation de menaces



Matthieu Chatelan & Loan Lassalle

10 janvier 2018

Table des matières

1	Introduction	2
2	Description du système	2
2.1	Objectifs du système	2
2.2	Exigences du système	2
2.2.1	Exigences technologiques	2
2.2.2	Exigences fonctionnelles	2
2.2.3	Exigences sécuritaires	3
2.3	Constitution du système	4
2.4	Biens	4
3	Sources de menaces	5
4	Scénarios d'attaques	6
4.0.1	Scénario d'attaque	6
4.0.2	Scénario d'attaque	6
4.0.3	Scénario d'attaque	7
4.0.4	Scénario d'attaque	7
4.0.5	Scénario d'attaque	7
4.0.6	Scénario d'attaque	7
5	Contremesures	7
5.1	Contremesure	7

Table des figures

1	Diagramme de flux de données de l'application	4
---	---	---

1 Introduction

Le but de ce document est de fournir une analyse de menaces auxquels notre programme WeChat est exposé. Premièrement, une description du système est donnée fournissant ainsi une meilleure compréhension des différents composants de ce dernier, les différents rôles mis à disposition ou encore les biens qui seront à protéger.

Ensuite, une analyse des différentes sources de menaces auxquels notre programme sera exposé. Les différentes sources potentielles sont énumérées ainsi que les compétences requises pour effectuer chacune de ces attaques.

Finalement, plusieurs scénarios d'attaques seront donnés ainsi que les contremesures appropriées à mettre en place afin de les contrer.

2 Description du système

2.1 Objectifs du système

Les objectifs fixés du système étaient de développer une application web permettant d'envoyer et de recevoir des messages textes entre collaborateurs au sein d'une entreprise. Le protocole de transmission SMTP ne devait pas être utilisé pour cette application.

Cette application serait par exemple utilisée entre employés d'une entreprise de développement de logiciels.

2.2 Exigences du système

Lors du développement de l'application, plusieurs contraintes ont été fixées par le client dans le cahier des charges. Dans les sous sections ci-dessous, ces dernières sont décrites et expliquées.

2.2.1 Exigences technologiques

Pour développer l'application, les différentes technologies suivantes ou contraintes ont dû être prises en compte :

- PHP
- SQLite
- Doit fonctionner sur l'environnement CentOS fournis

2.2.2 Exigences fonctionnelles

L'application devait être en mesure proposer deux rôles différents soit **collaborateur**, soit **administrateur**. De plus, un mécanisme d'authentification simple (utilisateur - mot de passe) doit permettre d'accéder aux fonctionnalités. Pour pouvoir se connecter, un utilisateur doit être défini comme étant "actif".

Une navigation aisée via des liens ou des boutons devait être mise en place.

Un **collaborateur** doit pouvoir effectuer les actions suivantes :

1. Lire les messages reçus sous forme de liste triée par ordre de date de réception avec plusieurs informations tels que :
 - la date de réception
 - l'expéditeur
 - le sujet
 - un bouton permettant de répondre au message (avec le sujet directement défini)
 - un bouton permettant de supprimer le message
 - un bouton permettant d'ouvrir un message et de voir le contenu du corps
2. Ecrire des messages à l'attention d'un autre utilisateur. Les informations suivantes doivent être fournies :
 - destinataire (unique)
 - sujet
 - corps du message
3. Changer le mot de passe de l'utilisateur connecté.

Un **administrateur** doit pouvoir effectuer les actions suivantes :

1. Avoir les mêmes actions qu'un collaborateur
2. Ajouter, modifier ou supprimer un utilisateur, représenté par les attributs suivants :
 - Un login (non modifiable)
 - Un mot de passe (modifiable)
 - Une validité actif/inactif (modifiable)
 - Un rôle (modifiable)

2.2.3 Exigences sécuritaires

Une authentification était demandée. Cette dernière ne devait pas permettre l'accès à toute autre page que celle de login lorsque l'utilisateur n'est pas authentifié.

Lors du développement initial, aucune autre mesure de sécurité n'étaient demandées. De ce fait, aucune protections contre des attaques XSS, CSRF, SQL Injection ou autre n'ont été mises en places.

Néanmoins, les mots de passes n'étaient pas stockés en clair dans la base de donnée, mais sous forme de hash salé pour empêcher une attaque par rainbow table sur ces derniers.

2.3 Constitution du système

Le système développé comprenait plusieurs acteurs et processus ainsi qu'une base de données centrale. Dans le diagramme de flux de données de l'application (Figure 1), un acteur est représenté par un rectangle alors qu'un processus est représenté par un cercle.

On notera que tous les processus qui se situent au dessus de la base de données dans le diagramme (5 au total) sont des processus que les deux acteurs peuvent utiliser. En revanche, les 3 processus au dessous de la base de données sont des processus réservés aux administrateurs.

L'ellipse traitillée en orange représente la frontière de confiance de l'application. Cela signifie que nous considérons que les communications entre processus et la base de données sont sécurisés. Toutes entrées utilisateur doivent être contrôlés avant de pénétrer cette zone de confiance.

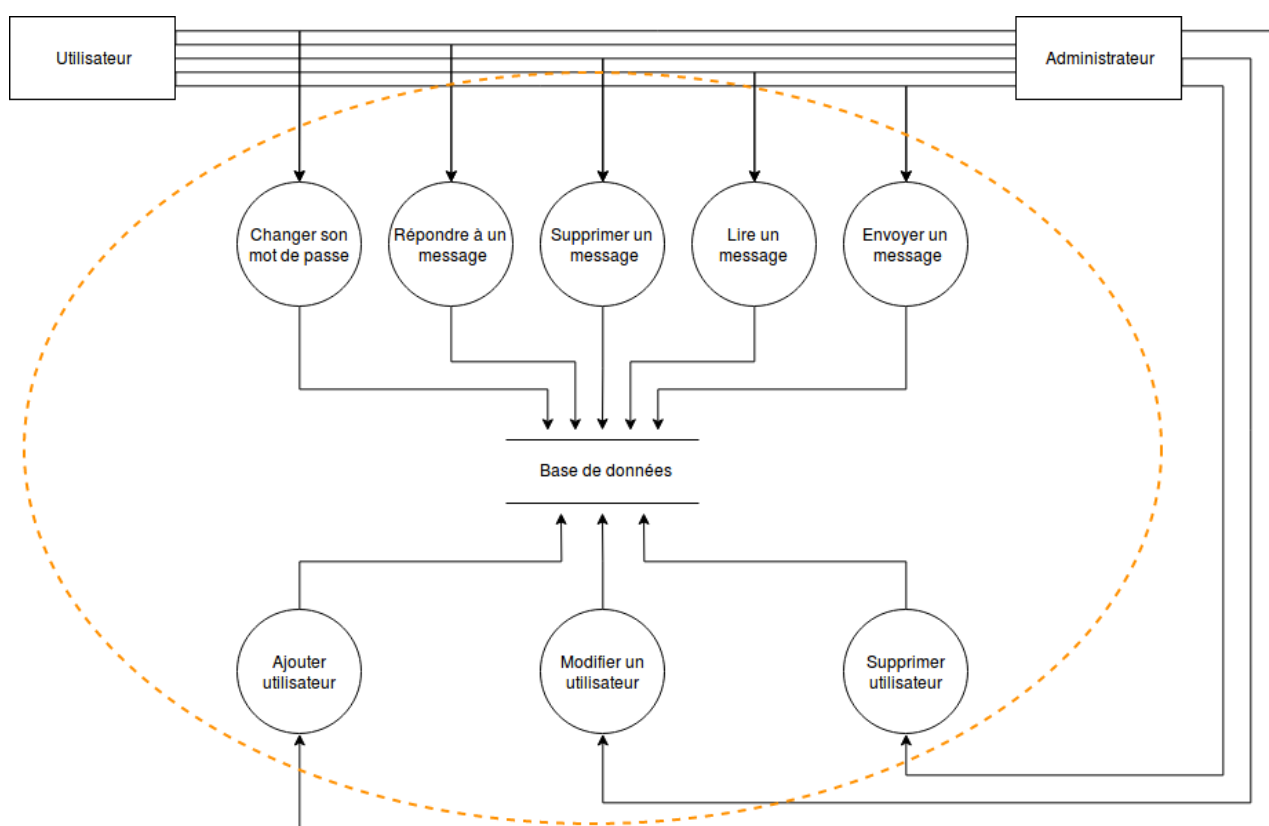


FIGURE 1 – Diagramme de flux de données de l'application

2.4 Biens

Les biens à protéger sont multiples dans une telle application. Ci-dessous, une liste non exhaustive de ces derniers

- Les messages
- Les comptes utilisateurs (leurs mots de passes)
- L'accès à l'administration des comptes

3 Sources de menaces

Initiateur	Motivations	Cible(s)	Potentialité
Utilisateur	Fun, Revanche, Curiosité	Tout éléments accessibles	Haute
Administrateur	Revanche, Curiosité	Tout éléments accessibles	Moyenne
Concurrent	Secrets d'entreprise	Base de données	Moyenne
Hackers	Gloire, Argent, Destruction	Tout éléments accessibles	Faible
Cyber-criminel	Vol d'informations, Spam, DDoS	Base de données	Faible

Parmi les différents initiateurs décrits ci-dessus, les utilisateurs représentent la plus grande menace et la menace la plus probable d'arriver. En effet, comme le programme est destiné à un usage interne à l'entreprise, a priori seuls les employés y auront accès.

N'y avoir accès que en interne ne signifie pas que ce n'est pas accessible de l'extérieur. Un attaquant externe (hacker, concurrent, cybercriminel) pourra tout à fait utiliser de l'ingénierie sociale pour inciter un utilisateur à lancer un programme malveillant afin de rendre l'accès à distance possible par le biais d'un reverse shell par exemple.

4 Scénarios d'attaques

Dans cette section, plusieurs scénarios d'attaque sont décrits. Grâce à ces scénarios, il est possible d'imaginer les différents points vulnérables de l'application et ainsi permet de mettre en place des contremesures appropriées.

#	Impact	Source de menace	Bien ciblé
4.0.1	Perte de confidentialité	Utilisateur, Concurrent, Hacker, Cyber-criminel	BDD
4.0.2	Perte de disponibilité	Tous	Tout élément actif
4.0.3	Perte d'intégrité	Hacker, Concurrent	BDD

4.0.1 Scénario d'attaque

Dans ce scénario d'attaque, les différents acteurs listés en tant que source de menace tenteront d'accéder à des informations qui ne leurs sont pas destinées. Pour parvenir à ses fins, un attaquant pourra essayer de multiples vecteurs d'attaque tels que :

- Manipulation de l'URL
- Injection SQL dans les divers champs de saisie disponibles
- Utilisation d'une faille de XSS stocké afin de récupérer des cookie de session

Avec peu de recherches sur internet et quelques tentatives, un attaquant avec de faibles connaissances sera en mesure de récupérer des données confidentielles tels que des mails ou encore des mots de passes en utilisant une injection SQL.

Avec une attaque plus élaborée, cet employé pourrait récupérer des cookie de session en se servant d'une attaque par XSS. Un message avec du code php sera stocké sur le serveur et quand l'utilisateur cible se connectera à sa boîte de réception, le code malicieux sera exécuté.

4.0.2 Scénario d'attaque

Ce scénario d'attaque est basé sur le scénario 4.0.1. En effet, des hackers, concurrents ou des cybercriminels pourraient être intéressés par les informations potentiellement confidentielles échangées entre les différents collaborateurs de l'entreprise.

Leurs motivations seraient en grande partie financières. En effet, un Hacker pourra faire par exemple du chantage à l'entreprise en menaçant de divulguer les informations récoltées. Un concurrent pourra se servir des informations obtenues afin de proposer des produits ou services similaires à ceux proposés par l'entreprise attaquée sans avoir à les développer.

Si le programme de messagerie développé n'est utilisé que sur un réseau interne, la première phase d'une attaque pour un attaquant externe serait de pénétrer le réseau interne de l'entreprise. Pour ce faire, de l'ingénierie sociale peut être tentée envers les collaborateurs par le biais de mails forgés et de pièces jointes malicieuses. Ce procédé sera à répéter pour chaque scénario où les attaquants sont externes au réseau de l'entreprise.

4.0.3 Scénario d'attaque

4.0.4 Scénario d'attaque

Ce scénario concerne la destruction d'informations par des tiers tels que des hackers ou encore des concurrents. Plusieurs approches sont possibles :

- Attaques XSS réfléchi ou stocké
- Attaques CSRF
- Injections SQL

Grâce à une attaque XSS, un attaquant pourra dérober les cookie de session d'un administrateur pour ensuite se connecter en son nom et supprimer des utilisateurs ou les basculer en inactif.

Avec l'aide d'un peu d'ingénierie sociale, un attaquant pourrait forger un lien pour supprimer des utilisateurs ou des messages et ensuite inciter un administrateur à cliquer sur son lien. Afin de dissimuler le contenu du lien, des services tels que "google short url" peuvent être utilisés.

A l'aide d'une injection SQL, un attaquant pourrait complètement supprimer toutes les entrées dans la base de données ou encore faire des entrées arbitraires tels que ajouter des administrateurs, modifier des entrées, etc...

4.0.5 Scénario d'attaque

4.0.6 Scénario d'attaque

5 Contremesures

tableau récap et montrer new implémentation

#	Scénario	Vulnérabilité	Contremesure
5.1	4.0.1	xxx	yyy

5.1 Contremesure