# Introduction

# Description du système

## DFD

## 

## Biens

Les seuls biens que contient cette application sont les informations client. Elles se composent : du nom de l’utilisateur, du hash de son mot de passe et de la liste de ses emails.

## Périmètre de sécurisation

Comme on peut le voir sur le diagramme ci-dessus, l’application possède une zone sécurisée qu’il n’est pas possible d’atteindre sans se connecter. Cette zone est séparée en deux parties : la partie utilisateur qui est accessible par les utilisateurs normaux et les administrateurs et la partie administrateur qui est accessible seulement par ces derniers.

La seule page qui est accessible sans s’authentifier est celle de login. Lors d’une tentative de connexion sur une des pages protégées, une vérification est effectuée. Si jamais l’utilisateur ne s’est pas authentifié, il est automatiquement redirigé sur la page de login.

Après avoir utilisé l’application, l’utilisateur peut se déconnecter ce qui a pour effet de supprimer les informations de session en cours et de le rediriger sur la page de login. Il ne peut plus accéder à la zone protégée tant qu’il ne s’authentifie pas à nouveau.

# Sources de menace

Les sources de menace pour cette application sont les personnes voulant avoir accès à des informations potentiellement confidentiel sur l’entreprise en accédant aux emails de ses utilisateurs. Cette catégorie comprend majoritairement des hackers recherchant, soit à exposer des informations sur l’entreprise au public, soit à monnayer les informations récupérées avec l’entreprise. Mais on peut aussi très bien trouver des script kiddies voulant essayer leurs outils sur une application réelle sans avoir véritablement de but autre que de s’amuser et de développer ses connaissances.

# Scénarios d’attaques

## Login

Le login de l’application est un lieu d’attaque très probable. La motivation principale étant que, si l’attaque réussi, on peut avoir accès à un compte utilisateur. De plus, en effectuant quelques recherches, l’attaquant se rend vite compte qu’il n’y a pas de politique de sécurité sur les mots de passe et que aucune protection contre le brute-force n’est mise en place sur le site. Il est donc (plutôt aisé) d’effectuer une attaque pas brute force sur un compte admin (ou utilisateur) afin d’obtenir un accès privilégié à l’application. Un accès administrateur permettrait de modifier la base de données facilement et de changer les mots de passe des autres utilisateurs. L’attaquant aurait alors un contrôle total de l’application.

Cette attaque est de type : "Elevation of Privilege" et, dans un deuxième temps, "Information disclosure".

## Base de données

Les requêtes effectuées par l’application à la base de données étaient vulnérables. En effet, il était possible d’effectuer des injections SQL lors de celles-ci. Le but de cette attaque est d’accéder aux informations contenues dans la base de données et de potentiellement les modifier afin de la corrompre en changeant son contenu.

Cette attaque est de type : "Information disclosure" et "Tampering".

# Contre-mesures

## Login

Afin d’empêcher ce type d’attaque, nous avons rajouté un captcha à remplir lors de chaque tentative de login. Cette mesure empêche l’utilisation de scripts permettant de brute-forcer le mot de passe et rend donc ce type d’attaque beaucoup plus compliqué et beaucoup plus gourmand en temps.

De plus, nous avons rajouté un nombre minimal de caractère pour le mot de passe de l’utilisateur. Il faut maintenant que le mot de passe soit composé de 8 caractères ou plus. Cette modification permet de rallonger encore le temps nécessaire pour effectuer une attaque par brute-force

## Base de données

Afin de se prévenir contre ce type d’attaque, nous avons changé les requêtes effectuées à la base de données par des "prepared statement" afin de prévenir tout risque d’injection SQL.

# Conclusion