STI - Projet 2 -Rapport d’étude de menace

Nikolaos Garanis et Nemanja Pantic

# 

# Table des matières

[**Table des matières**](#_us9vo6y29fwp) **2**

[**Introduction**](#_kloxzwa49nn2) **3**

[**Source menace**](#_88ehvbfe9u7t) **3**

[DFD](#_r8w3289wyqzi) 3

[Identifier ses biens](#_nyi1uqd84nhf) 3

[Périmètre de sécurisation](#_lnqni5qghdex) 3

[**Scénario d’attaques 1**](#_o90cwjs0ys7t) **4**

[Business impact](#_17enlvvw6lw) 4

[Threat source](#_mjz4e7mbyy6s) 4

[Motivation](#_n1w9j2bkzuvc) 4

[Targeted asset](#_n6xdf2o9y8gs) 4

[Contre mesure](#_jncg301zctua) 4

[**Scénario d’attaques 2**](#_yibwxfuvbaj9) **5**

[Business impact](#_id91shu8zt25) 5

[Threat source](#_tmunp4qh4o32) 5

[Motivation](#_f95a0lh6tud1) 5

[Targeted asset](#_fphs44uqccr0) 5

[Contre mesure](#_c95bt3p95wmo) 5

[**Scénario d’attaques 3**](#_3uqnciyzobtz) **6**

[Business impact](#_vtnvn5gszdgw) 6

[Threat source](#_kv0p2xdmae4j) 6

[Motivation](#_zksu8xnlmah) 6

[Targeted asset](#_cvy256x0ladp) 6

[Contre mesure](#_r3wuptbhemes) 6

[**Conclusion**](#_y8gl4frzmtqf) **6**

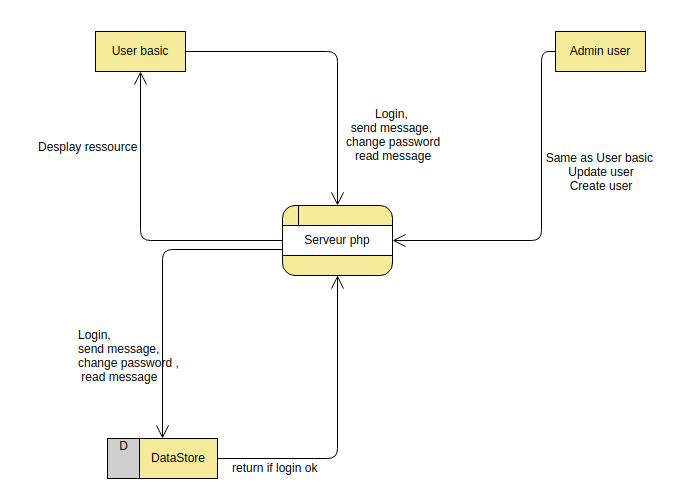
# 

# Introduction

Ce laboratoire est la suite du laboratoire numéro 1, il faut cette fois-ci, avec un autre binôme, reprendre une des applications développées lors du projet 1 et y ajouter de la sécurité

# Source menace

## DFD



Nous faisons confiance au DataStore, au serveur PHP et à l’admin user

## Identifier ses biens

Les données utilisateurs

Le contenu des emails échangés

## Périmètre de sécurisation

Nous allons fixer des problèmes présents sur l’application, rien d’autre.

# Scénario d’attaques 1

Vol d’identifiants d’utilisateurs

## Business impact

Haute : des utilisateurs non souhaités ont accès à des données de clients.

## Threat source

Hackers, espionnage, script kiddies

## Motivation

Financière, challenge, avantage industriel

## Targeted asset

Base de données, saisies du client  
  
Attack scenario

Brute force, XSS, sniffing

## Contre mesure

Utiliser des moyens de communications chiffrés, renforcer la sécurité des mots de passe (test de force), utiliser un bon algorithme de chiffrage, bloquer une IP s’il y a un grand nombres de requête.

# Scénario d’attaques 2

Exécution de code malveillant

## Business impact

Haute : modification des données (corrompue)

## Threat source

Hacker, utilisateurs malveillants, espionnage

## Motivation

Financière, challenge

## Targeted asset

Base de données

Attack scenario

XSS, CSRF

## Contre mesure

Interdire certains caractères, implémenter un token anti-csrf

# Scénario d’attaques 3

DoS

## Business impact

Moyen : interruption du service de messagerie, l’impact va dépendre de l’importance de nos clients

## Threat source

Concurrence, hacker, script kiddie

## Motivation

Financière, s’amuser

## Targeted asset

Serveur

Attack scenario

Grands nombres de requêtes sur le serveur

## Contre mesure

Bloquer une adresse IP après x requêtes dans un délai défini

# Conclusion

Il a été intéressant de trouver des failles, de les exploiter et de les corriger. En particulier l’implémentation des tokens CSRF ainsi que le filtrage des entrées utilisateurs. Dans ce qui a été analysé nous n’avons pas implémenté d’HTTPS, ce qui fait qu’il est toujours possible pour un attaquant de sniffer le réseau et de tout voir transiter en clair et finalement nous n’avons pas implémenté de moyen d’empêcher le brute force, ce qui fait qu’un attaquant peut faire tomber nos services et tomber sur le mot de passe d’un client après x essais.