# Relatório

## Contents



## Image Board

O Image Board é um site similar aos *image boards* tradicionais, mas tem um sistema de contas, para que as vulnerabilidades façam sentido.

Neste site um utilizador pode criar a sua conta, fazer login, registar-se, postar ou responder a outros posts e ver o seu perfil.

Administradores do site podem ver os perfís de todos, apagar contas e elevar outros utilizadores a administrador.

## CWEs implementadas

### SQL injection

Postagens (comentários e respetivas respostas) são vulneráveis a *SQL injection*.

SQL injection ocorre quando queries não são pré processadas e geradas apartir de texto dado por utilizadores. Nestas circunstâncias, sabendo a query é fácil escrever texto que será executado como sql.

## XSS

Postagens (comentários e respetivas respostas) são vulneráveis a *Cross Site Scripting*.

Análogamente ao SQL injection, cross site scripting ocorre quando texo dos utilizadores é incorporado na página web sem tratamento.

## Path Traversal

Ainda que a vulnerabilidade em si remeta mais para acesso a localizações num sistema de ficheiros, facilemnte se pode extrapolar a interpretação a um site. Quando um utilizador consegue, inserindo valores no URL, aceder a uma página (diretório num site) à qual não deve aceder, ocorre *path traversal*.

Esta vulnerabilidade está presente na página de administração. Num produto real estariam mais ferramentas de tratamento de dados e gestão de pessoas, a ideia desta página é ser apenas um exemplo disso dado que não há nececidade destes serviços neste trabalho.

Está ainda presente na página de perfil, onde qualquer utilizador pode, apenas mudando o URL, ver o perfil privado dos outros.

## Missing Authentication for Critical Function

Esta vulnerabilidade está associada às mesmas páginas que as mencionadas em Path Traversal.

Estas páginas requerem permissões de acesso que não são verificadas. Para remover esta vulnerabilidade basta conferir se o utilizador tem login e as permissões de acesso desejadas.

## Exposure of Sensitive Information

Muitas vezes atacantes obtêm informação através dos erros demasiado explícitos. Pode ser bom para os desenvolvedores terem erros explícitos para diagnosticar erros, masdo mesmo modo que eles podem diagnosticar também os atacantes o podem.

Para corrigir isto, opções de debug (como, neste caso, do flask) devem ser desligadas.

### Reliance on Cookies without Validation and Integrity Checking and Sensitive Cookie Without 'HttpOnly' Flag

`+ Allows XSS attacks cookie acess and authentication bypass/sql injection attacks via cookie`

A incorreta gestão de cookies pode levar a ataques de XSS e SQL. Atacantes podem retirar os cookies através de XSS.

Esta vulnerabilidade está presente por todas as páginas. A sua mitigação é feita pela gestão de cookies no login.

### Weak cipher/Weak hash function

Há certas cifras e funções de hash que já foram quebradas. Isto faz com que seja possível obter palavras passe e outra informação por elas escondida de modo bastante assecível.

Estas vulnerabilidades estão mostradas na página de login.

### Use of a One-Way Hash without a Salt

Como explicado acima, uma cifra fraca pode facilmente ser quebrada. Usar *salt* faz as cifras mais fortes, dado que introduz ruido.

Assim, não utilizar *salt* deixa o criptograma mais vulnerável a ataques *rainbow-table*.

Esta vulnerabilidade está demonstrada na página de registo.

### Weak Password Requirements

Por muito seguro que seja um serviço, palavras passe fracas são meios para atacantes obterem acesso priviligiado à aplicação.

No entanto, requisitos muito fortes farão com que os utilizadores guardem as palavras passe em sítios inseguros, perdendo assim o efeito.

É muito importante forçar boas políticas de palavra passe, tendo em conta este compromisso.

Esta vulnerabilidade está mostrada na página de registo.

## Análise e atribuições

Análise detalhada das vulnerabilidades e links para recursos que nos ajudaram podem ser encontrados no diretório `analysis`, organizada por páginas.

## Divisão do trabalho de grupo

Cada elemento implementou correções às vulnerabilidades apontadas em cada página que programou e ainda elaborou as secções do relatório e análise a elas afetas.

Rúben Castelhano 97688: pesquisa de CWEs, página login/register/logout (análise das suas vulnerabilidades)

João Felisberto 98003: pesquisa de CWEs, página inicial e de administração (versão final e relatório/análise)

Vasco Santos 98391: página de perfil