**Projeto eHealth Corp**

Análise

Analysis

Segurança Informática e nas Organizações

Prof.João Paulo Barraca

Departamento de Eletrónica,Telecomunicações e Informática

Ano Letivo 2022-2023

Equipa 38

Bruno Lins- 101077

Filipe Barbosa-103064

Miguel Gomes-103826

Pedro Durval-103173

Índice

[Exploração das Vulnerabilidades 3](#_Toc119524132)

[CWE-20 3](#_Toc119524133)

[CWE-79 3](#_Toc119524134)

[CWE-89 4](#_Toc119524135)

[Login 4](#_Toc119524136)

[Barra de pesquisa 4](#_Toc119524137)

[CWE-256 5](#_Toc119524138)

[CWE-620 6](#_Toc119524139)

[CWE-756 6](#_Toc119524140)

# Exploração das Vulnerabilidades

## CWE-20

Esta vulnerabilidade pode ter alguma interferência no site, como no nosso caso.

Na nossa marcação de consultas, como a hora, está como ***“input type=”text””*** qualquer texto inserido aparecerá como a hora marcada( não há qualquer tipo de verificação se a hora inserida está, de certa forma, com o formato desejado), como o exemplo que aparece na imagem:

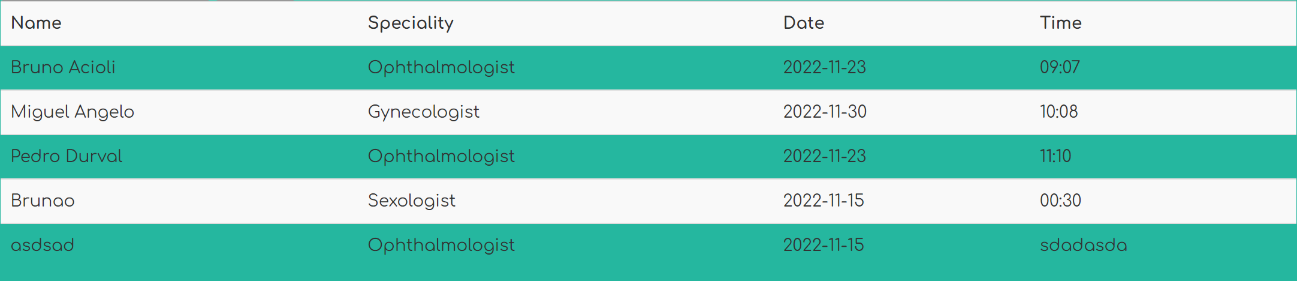


Figura - Vulnerabilidade CWE-20

## CWE-79

Através de JavaScript, assim que aplicamos um código na zona de reviews como:

<**script**>alert(“VIRUS”)<**\script**>

Cada vez que um utilizador da plataforma abrir a página irá aparecer um ***alert***.

Outro exemplo de ***Cross-Site-Scripting*** : é possível enviar um script para o site de forma, a que cada vez que o utilizador clique em determinado comentário seja direcionado para um site potencialmente perigoso:

<**script**>  
 Function redirecttoPage(){  
 window.open(“http://www.potentialdangerourssite.com)  
 document.getElementbyId(compiler).style.display = ‘block’;  
 }  
</**script**>  
<**td**><**a** style color = ‘black’; onClick = ‘redirecttoPage ()’> RECEIVE INFINTY MONEY</**a**></**td**>

## 

## CWE-89

### Login

Assim que é inserido ***username” -- //*** no campo de username do login do site, o atacante consegue fazer login sem necessitar de password, porque assim o código malicioso inserido comenta a parte da palavra-passe.

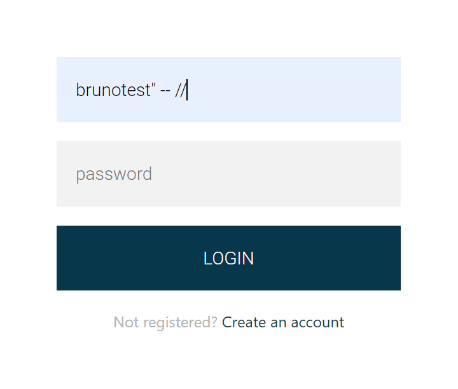


Figura -Vulnerabilidade Login CWE-89

### Barra de pesquisa

Na barra de pesquisas, é possível descobrir vários registos da base de dados através de ***SQLInjection*** como:

**UNION** **SELECT** 1, 2,**password** **FROM** **login** **WHERE** username **LIKE** "brunotest" -- //

Com isto, é possível descobrir a palavra passe do utilizador “brunotest”, como é descritível na imagem:

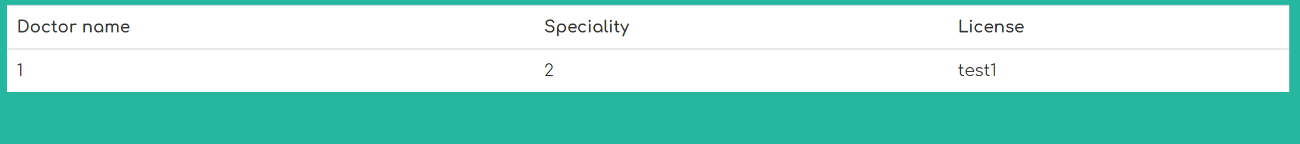


Figura - Vulnerabilidade Barra de pesquisa CWE-89

" UNION **SELECT** 1, username, **password** **FROM** login -- //  
**Or**  
“ **UNION** **SELECT** 1, “<br>”,username,**password** **FROM** login -- //

Com isto, é possível descobrir todas os nomes de utilizadores e respetivas palavra-passe, como é possível ver na imagem em baixo:

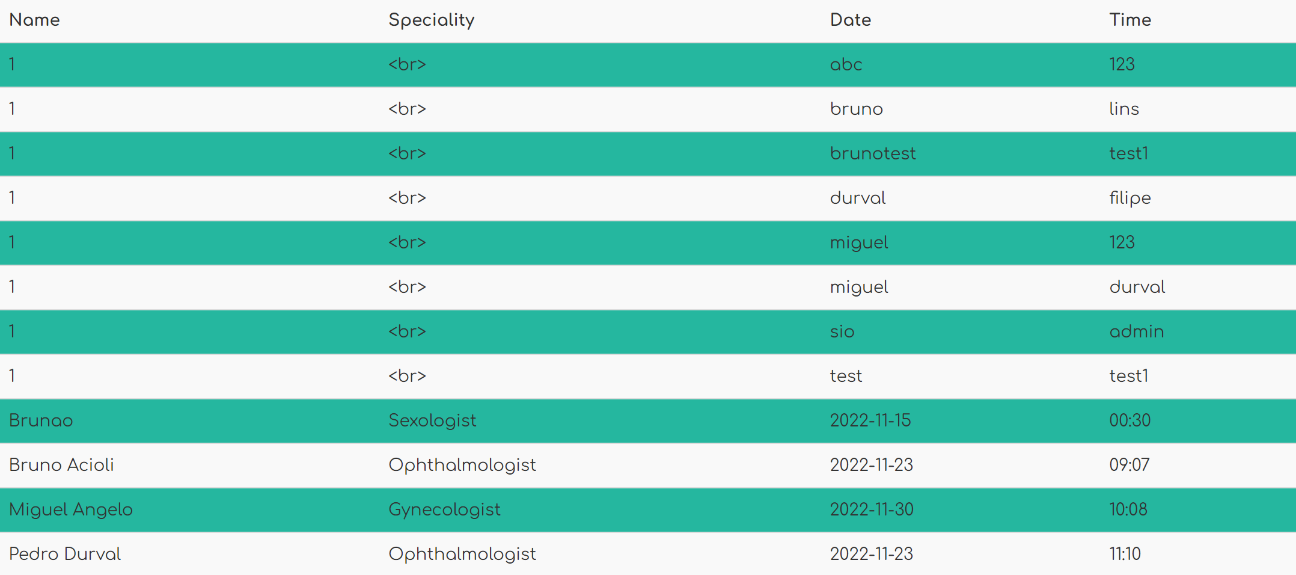


Figura - Vulnerabilidade Barra de pesquisa CWE-89

## CWE-256

A exploração desta vulnerabilidade pode ser feita em conjunto com ***SQLInjection*** , pois, assim como em cima, quando é descoberta a base de dados aparecerá tal igual como é visualizado na imagem acima. Assim torna-se bastante fácil para o atacante descobrir todas as palavras-passes dos utilizadores.

## CWE-620

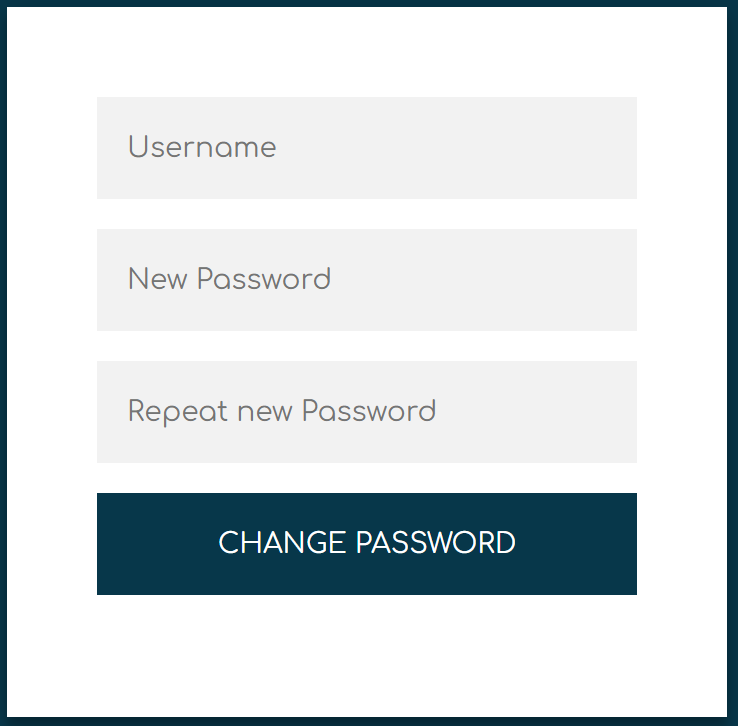
 Sem a verificação de palavra-passe, é possível alterar a palavra-passe de um utilizador sem saber a palavra-passe atual. Portanto, podemos dizer que este ataque poderá ser combinado com ***SQLInjection .*** Se o atacante entrar na conta de um utilizador como é referido em cima, basta mudar a palavra-passe do utilizador e fica com total acesso da conta.

Figura - Vulnerabilidade CWE-620

## CWE-756

Como foi referido esta vulnerabilidade é bastante comum e, caso não seja “contrariada”, pode revelar informação bastante suscetível acerca da base de dados.

Para criar um erro na ***query*** basta colocar umas aspas e, caso não seja implementada uma página de erro, a página de erro implementada pela biblioteca usada no servidor irá revelar o erro na ***query***, mostrando a linha de erro, onde aparecerá nomes das tabelas, nomes das colunas, etc. Assim o atacante fica com informação demasiado sensível em relação à base de dados, tornando-se assim mais fácil outro tipo de ataques.

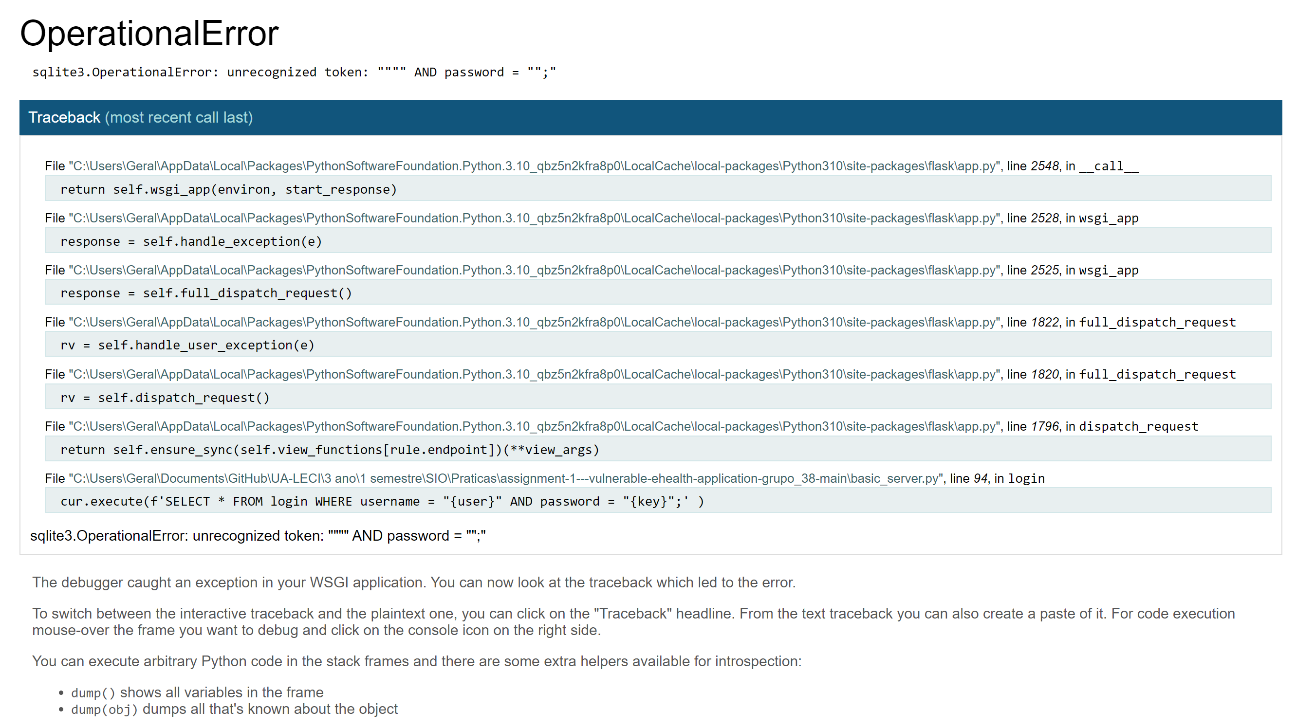
Um exemplo de uma página de erro, caso não seja implementada:

Figura - Vulnerabilidade CWE-756

Como já referido, nesta página de erro é visível informação bastante suscetível acerca da base de dados do servidor, o que potencia o atacante a procurar por mais suscetibilidades.