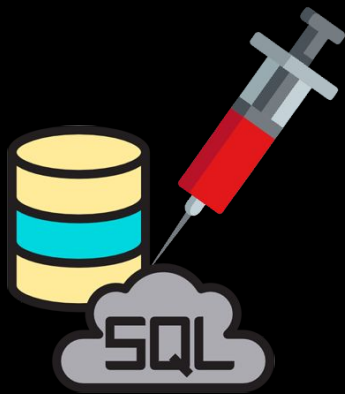


SQL Injection

Conceitos Básicos



O que é SQL Injection?



SQL Injection / Injeção SQL é uma vulnerabilidade que permite manipular de forma não autorizada as queires que a aplicação faz ao banco de dados. Este tipo de vulnerabilidade existe quando o input do usuário não é tratado corretamente pela aplicação antes de ser colocado na query SQL.

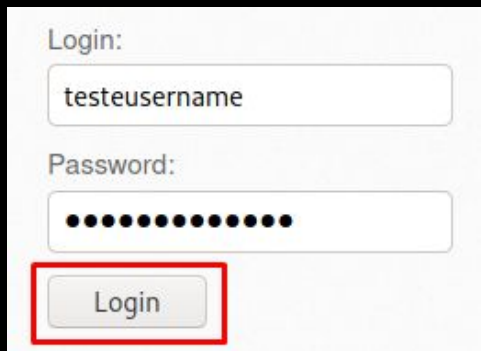
A exploração desta vulnerabilidade poderá permitir:

- Acesso a informação sensível / confidencial guardada no banco de dados;
- Criar / Modificar / Eliminar registos guardados no banco de dados;
- Ganhar acesso remoto ao servidor de banco de dados;

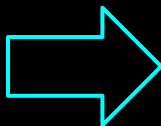


O que é SQL Injection - Exemplo

Imaginemos que queremos fazer autenticação em uma aplicação web e, para tal, temos que preencher um formulário de Login com os campos **Login** e **Password** (Input do Usuário).



The image shows a web login form. It has two input fields: one labeled "Login:" containing the text "testeusername", and another labeled "Password:" containing a series of black dots. Below these fields is a button labeled "Login". A red rectangular box is drawn around the "Login" button.



Quando clicamos no botão "Login", é feita uma requisição web, ilustrado na imagem abaixo, para a aplicação back-end. Esta requisição envia no body o input do usuário (**Login** e **Password**).



The image shows a network request in a browser's developer tools. The request is a POST to /bwAPP/sqli_3.php. The body of the request is "login=testeusername&password=testepassword&form=submit".

```
Request
Pretty Raw \n Actions
1 POST /bwAPP/sqli_3.php HTTP/1.1
2 Host: 192.168.0.105
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 54
9 Origin: http://192.168.0.105
10 Connection: close
11 Referer: http://192.168.0.105/bwAPP/sqli_3.php
12 Cookie: PHPSESSID=7fdgu1lvfe15cuipq2dv2vqhma; showhints=1; security_level=0
13 Upgrade-Insecure-Requests: 1
14
15 login=testeusername&password=testepassword&form=submit
16
```

O que é SQL Injection - Exemplo

Quando o back-end recebe a requisição web, executa o seguinte código:

```
$login = $_POST["login"];
$login = mysqli($login);

$password = $_POST["password"];
$password = mysqli($password);

$sql = "SELECT * FROM heroes WHERE login = '" . $login . "' AND password = '" . $password . "'";

// echo $sql;

$recordset = mysqli_query($link, $sql);
```

- A) Atribuir à variável `$login` o input do usuário (Login) enviado na requisição;
- B) Atribuir à variável `$password` o input do usuário (Password) enviado na requisição;
- C) Atribuir à variável `$sql` a query SQL com o input do usuário;

Query:	SELECT * FROM heroes WHERE login= ' ' . \$login . ' ' AND password= ' ' . \$password . ' ' ;
Query com Input:	SELECT * FROM heroes WHERE login='testeusername' AND Password='testepassword';

- D) Atribuir à variável `$recordset` o resultado da query SQL;

O que é SQL Injection - Exemplo

- 1) De seguida, o servidor verifica se a variável `$recordset` tem algum valor (resultado da query) ou não. Se não tiver nenhum valor é porque houve um erro SQL. Se tiver um valor passa para o ponto 2;
- 2) Aqui, é atribuído à variável `$row` (array) o resultado da query SQL.
- 3) De seguida é verificado se existe o valor "login" no resultado da query (`$row`). Se existir é porque a query encontrou no banco de dados um usuário com o Login e Password correspondentes ao input do usuário. Assim sendo, o back-end devolve para o front-end uma mensagem de boas vindas. Caso o valor "login" não exista no resultado da query (`$row`), é executado o passo 4.
- 4) O back-end devolve para o front-end a mensagem "Invalid credentials".

```
if(!$recordset)
{
    die("Error: " . mysqli_error($link));
}
else
{
    $row = mysqli_fetch_array($recordset);
    if($row["login"])
    {
        // $message = "<font color='green'>Welcome " . ucwords($row["login"]) . "...</font>";
        $message = "<p>Welcome <b>" . ucwords($row["login"]) . "</b>, how are you today?</p><p>Your secret: <b>" . ucwords($row["secret"]) . "</b></p>";
        // $message = $row["login"];
    }
    else
    {
        $message = "<font color='red'>Invalid credentials!</font>";
    }
}
mysqli_close($link);
```


O que é SQL Injection - Exemplo

Como neste caso o input do usuário não corresponde a um usuário registrado no banco de dados, o back-end devolve para o front-end a mensagem "**Invalid credentials!**".

```
if(!isset($recordset))
{
    die("Error: " . mysql_error($link));
}
else
{
    $row = mysql_fetch_array($recordset);
    if($row["login"])
    {
        // $message = "<font color='green'>Welcome " . ucwords($row["login"]) . "...</font>";
        $message = "<p>Welcome <b>" . ucwords($row["login"]) . "</b>, how are you today?</p><p>Your secret: <b>" . ucwords($row["secret"]) . "</b></p>";
        // $message = $row["login"];
    }
    else
    {
        $message = "<font color='red'>Invalid credentials!</font>";
    }
}
mysql_close($link);
```

Login:

Password:

Login

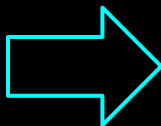
Invalid credentials!

O que é SQL Injection - Exemplo

Vamos imaginar que um hacker sabe que existe uma conta de usuário chamado "thor" e quer acessar a esta conta. Para isso, vai tentar injetar código SQL nos campos do formulario de login.

Login:

Password:



Quando clicamos no botão "Login", é feita uma requisição web, ilustrado na imagem abaixo, para o back-end da aplicação. Esta requisição envia no body o input do usuário (**Login** e **Password**).

Request

Pretty Raw In Actions

```
1 POST /bWAPP/sqli_3.php HTTP/1.1
2 Host: 192.168.0.105
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 48
9 Origin: http://192.168.0.105
10 Connection: close
11 Referer: http://192.168.0.105/bWAPP/sqli_3.php
12 Cookie: PHPSESSID=7fdgu1lvfe15cui.pq2dv2vqhna; showhints=1; security_level=0
13 Upgrade-Insecure-Requests: 1
14
15 login=thor%27%23&password=dasudasidu&form=submit
16
```

O que é SQL Injection - Exemplo

Query: **SELECT *FROM** heroes **WHERE** login= "\$login." **AND** password= "\$password.";

Input do usuário:

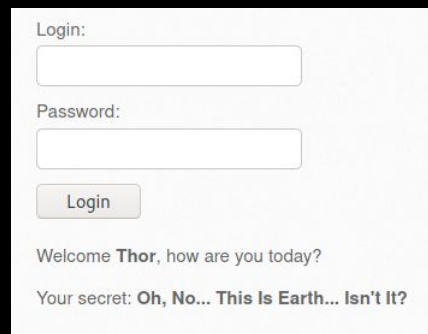
- Username: **thor'#**
- Password: **random**

Query Final: **SELECT *FROM** heroes **WHERE** login= 'thor'#' **AND** password= 'random';

Query Final: **SELECT *FROM** heroes **WHERE** login= 'thor';

O que é SQL Injection - Exemplo

Resultado: A query SQL devolve toda a informação do usuário "thor". Assim sendo, o back-end devolve para o front-end uma mensagem de boas vindas e o hacker consegue com sucesso autenticar-se na aplicação web com a conta do usuário "thor".



Login:

Password:

Login

Welcome Thor, how are you today?

Your secret: Oh, No... This Is Earth... Isn't It?

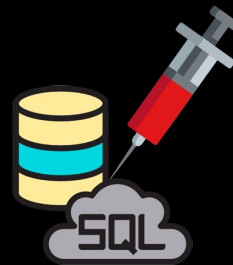
Tipos de SQL Injection



In-Band / Classic



Inferential / Blind



Out-Of-Band

