**FACULDADE SENAC – JOINVILLE**

**AEO:SQL INJECTIONE DAO (DATA ACCESS OBJECT)**

**JOINVILLE**

**2022**

**NOME: Rafael Farias de Souza**

**O QUE É SQL INJECTION**

SQL Injection (Structured Query Language Injection), injeção de SQL em tradução livre, é um tipo de ataque que pode dar a usuário malicioso controle completo sobre o banco de dados de seu aplicativo em casos extremos, inserindo código SQL arbitrário em uma consulta de banco de dados. Os rankings de vulnerabilidades quase em sua maioria listam esse ataque como crítico oferecendo risco quase irremediável nos casos de exploração com sucesso. Uma vulnerabilidade de injeção de SQL, como o nome sugere, pode afetar qualquer site ou aplicativo web que use um banco de dados SQL, como MySQL, Oracle, SQL Server entre outros. Este ataque é utilizado para obter acesso não autorizado aos seus dados confidenciais, como: informações do cliente, dados pessoais, segredos comerciais, propriedade intelectual e muito mais. Os ataques de injeção de SQL são uma das vulnerabilidades de aplicativos web mais antigas, prevalentes e perigosas.

EXEMPLO: .

**O QUE É DATA ACCESS OBJECT**

O DAO é um padrão para aplicações implementadas com linguagens de programação orientada a objetos e arquitetura MVC. E que utilizam persistência de dados, onde existe a separação das regras de negócio das regras de acesso a banco de dados. E ainda, onde todas as funcionalidades de bancos de dados, tais como obter conexões, mapear objetos para tipos de dados SQL ou executar comandos SQL, devem ser feitas por classes DAO. O padrão Data Access Object (DAO) é usado para separar a camada de acesso a API ou de Operações, como banco de dados, da camada de serviço.

EXEMPLO : 

**TESTES COM SQL INJECTION JAVA**

**OBS: não estava conseguindo fazer no meu código, mais achei uma em JAVA.**

Exemplo hipotético de código Java que pode ser vulnerável a SQL injection:

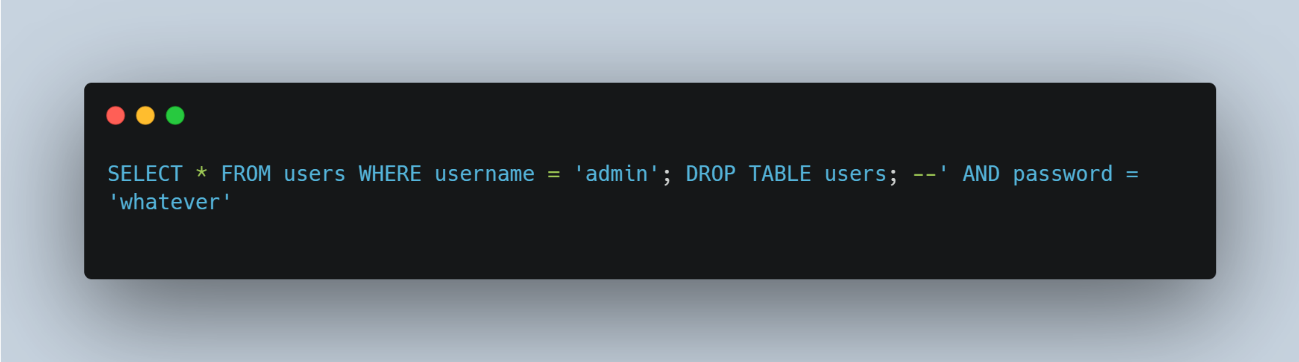
****

Neste exemplo, o código está recebendo o nome de usuário e a senha do usuário da solicitação HTTP e, em seguida, construindo uma consulta SQL para verificar se o nome de usuário e a senha correspondem a uma entrada na tabela de usuários. No entanto, se um hacker enviar um nome de usuário ou senha maliciosos que contenham comandos SQL adicionais, esses comandos podem ser executados pelo banco de dados.

Por exemplo, se o hacker enviar o seguinte como o nome de usuário:



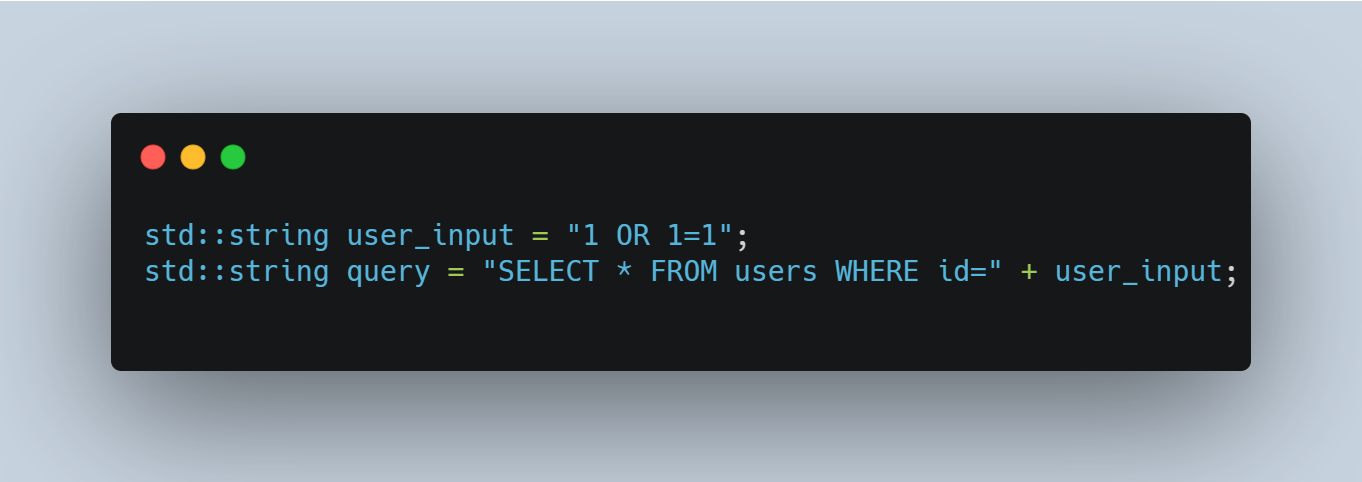
A consulta SQL gerada seria:



Isso faria com que a tabela de usuários fosse excluída, o que provavelmente causaria problemas para o aplicativo. Para evitar esse tipo de ataque, é importante validar corretamente as entradas do usuário e usar parâmetros preparados em vez de concatenar strings para construir consultas SQL.

**EXTRA (SQL INJECTION EM C++)**

A SQL injection é uma forma de ataque que visa explorar vulnerabilidades em aplicações que utilizam a linguagem SQL. Isso pode ser feito inserindo comandos maliciosos em entradas de dados, como formulários de login ou consultas de banco de dados. Em um exemplo de SQL injection em C++, um atacante poderia inserir código malicioso em um campo de entrada de um formulário, como o seguinte:



Neste caso, o atacante está inserindo o comando "1 OR 1=1" na variável user\_input, que é concatenada com a consulta SQL na variável query. Quando essa consulta é enviada para o banco de dados, ela retornará todos os registros na tabela users, independentemente do valor especificado no campo id, pois o comando OR 1=1 sempre será avaliado como verdadeiro. Isso permite que o atacante visualize informações confidenciais ou altere ou exclua dados do banco de dados. Para evitar esse tipo de ataque, é importante validar cuidadosamente todas as entradas de dados e utilizar funções de escape adequadas para garantir que apenas os dados válidos sejam inseridos no banco de dados. Além disso, é recomendável utilizar APIs de acesso a banco de dados que forneçam camadas de segurança adicionais.

**REFERENCIAS:**

<https://www.sitedefender.com.br/o-que-e-slq-injection/>

<https://rogeriofontespsi.wordpress.com/2016/04/25/o-padrao-dao/>

<https://www.jdevtreinamento.com.br/dao-estrutura-de-acesso-ao-banco/>

<https://www.devmedia.com.br/sql-injection-em-multiplas-plataformas/31389>

<https://www.devmedia.com.br/sql-injection/6102>