# SQL INJECTION EM SISTEMAS WEB: MALEFÍCIOS E COMO PREVENIR

Luis Antônio Magagnien[1], Vitor Matheus Ulrich[2], Matheus Pelozatto Sampaio[3], Fabiano de Oliveira Wonzoski[4]

1. Graduando em Ciência da Computação, Unoesc, Videira, SC
2. Graduando em Ciência da Computação, Unoesc, Videira, SC
3. Graduando em Ciência da Computação, Unoesc, Videira, SC
4. Coordenador do curso de Ciência da Computação, Unoesc, Videira, SC

**Autor correspondente:** Matheus Pelozatto Sampaio, matheusp.sampaio@gmail.com

**Área:** Ciências Exatas e Tecnológicas

**Introdução:** No cenário digital atual, a segurança da informação é crucial para organizações e desenvolvedores. Dentre as diversas ameaças, destaca-se a SQL Injection, uma técnica que consiste em inserir instruções SQL maliciosas nas entradas de dados de um sistema. Segundo a OWASP, esta vulnerabilidade figura entre as dez principais ameaças à segurança de aplicações web. A técnica explora a maneira insegura como as aplicações constroem consultas SQL a partir das entradas dos usuários, o que possibilita a manipulação não autorizada do banco de dados. Um caso notório foi o da operadora britânica TalkTalk em 2015, que sofreu um ataque de SQL Injection resultando na exposição de dados de aproximadamente 157.000 clientes, com um prejuízo estimado em 60 milhões de euros e a perda de 95.000 clientes. **Objetivo:** Este artigo visa analisar na prática o impacto de um ataque de injeção de SQL por meio de uma aplicação web e banco de dados simulados, além de apresentar práticas eficazes de prevenção, com o intuito de contribuir para o desenvolvimento de sistemas mais seguros. **Método:** Criou-se um ambiente de teste com PHP e MySQL, contendo uma tabela de "usuarios". Desenvolveu-se uma página de login vulnerável que concatenava os dados do formulário diretamente na query SQL. Para testar a falha, utilizou-se o payload ' OR '1'='1 no campo de senha. Comparativamente, foi implementada uma versão segura da mesma consulta utilizando prepared statements, que separam a consulta dos parâmetros e impedem a manipulação da lógica original. **Resultados:** Na aplicação vulnerável, o ataque de SQL Injection burlou a autenticação com sucesso, exibindo "Login bem-sucedido!" mesmo com credenciais inválidas, comprovando a falha de segurança. Em contrapartida, o mesmo ataque na versão segura falhou, retornando "Usuário ou senha incorretos" e demonstrando a eficácia da prevenção. **Conclusão:** O experimento demonstrou de forma prática os riscos associados a consultas SQL inseguras e ao tratamento inadequado dos dados de entrada do usuário. As principais causas para essa falha são a concatenação de strings para criar consultas e a falta de validação dos dados. Para prevenir tais ataques, recomenda-se o uso de métodos seguros, como os disponíveis em ORMs, que separam a lógica da consulta de seus parâmetros. Este estudo evidencia a importância de incorporar práticas de segurança desde o início do desenvolvimento de software.

**Palavras-chave:** SQL Injection; Segurança da Informação; Vulnerabilidade; Aplicações Web.