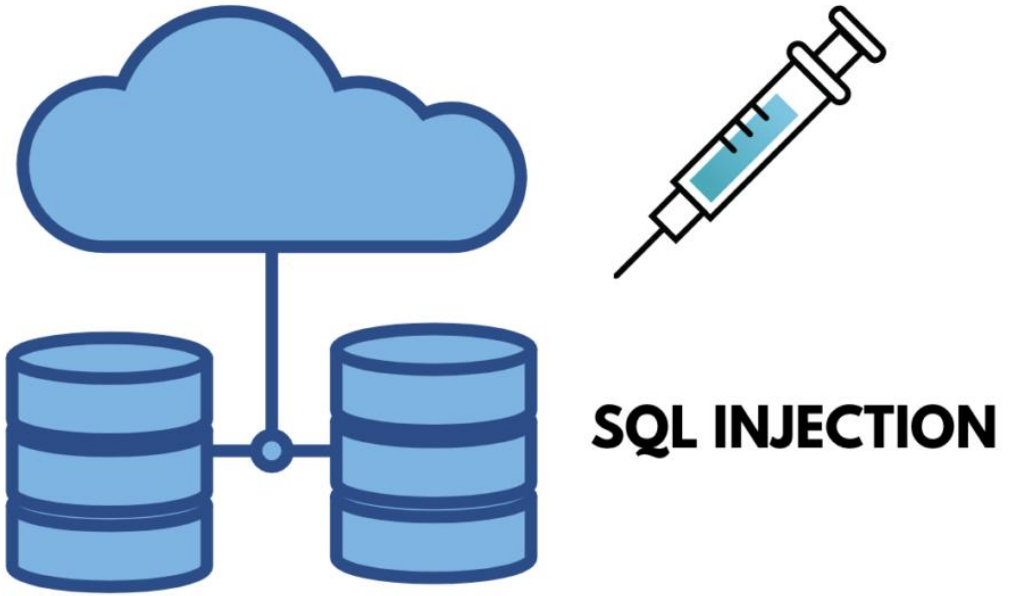
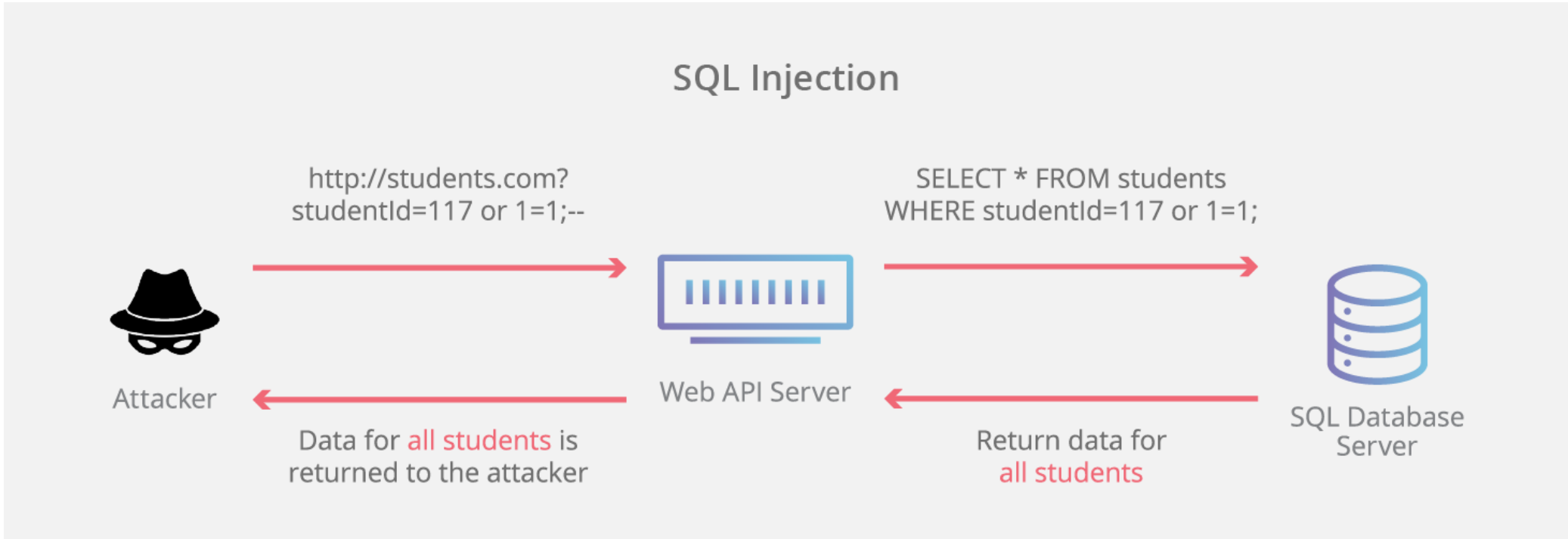
Prof. Me. Hélio Esperidião

SQL INJECTION



Sql injection

- É uma técnica de ataque que envolve a manipulação do código SQL.
- SQL Injection é uma classe de ataque onde o invasor pode inserir ou manipular consultas criadas pela aplicação, que são enviadas diretamente para o banco de dados relacional.
- Por que o SQL Injection funciona?
  - A aplicação aceita dados fornecidos pelo usuário;
    - Você pede para digitar o nome, mas quem garante que ele não digite código sql de forma maliciosa?


SQL Injection
http://students.com?
studentId=117 or 1=1;--
SELECT * FROM students
WHERE studentId=117 or 1=1;
Attacker
Web API Server
SQL Database
Server
Data for all students is
returned to the attacker
Return data for
all students

Sql injection na prática.

- Imagine o seguinte algoritmo:

$nome = $_GET[‘nome’];
$sql = “select * from Cliente Where nome = ‘$nome’”;

Se o usuário usar digitar o valor da variável $nome for igual a:Helio temos:
$sql = select * from Cliente Where nome = ‘helio’

Sql injection na prática.

- O problema é quando o usuário não digita o nome;
- Imagine que no lugar do nome o usuário digite: `' OR 1=1; #'`

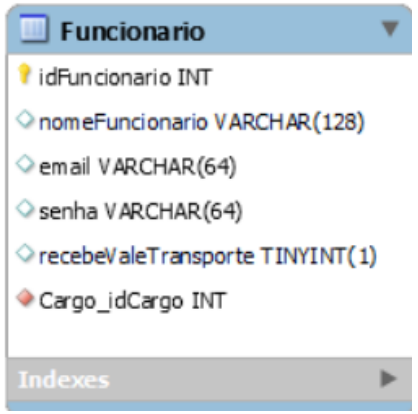
$nome = $_GET[‘nome’];
$sql = “select * from Cliente Where nome = ‘$nome’”;

- Teríamos algo assim:

```
SELECT * FROM clientes WHERE nome ='' OR 1=1; #’
```

- *Depois do # é comentário, e serão apresentados todos os clientes.*
- *Esse é um exemplo simples, mas o usuário pode criar instruções que podem potencialmente excluir o banco de dados.*

```php
private function createWithId(Funcionario $funcionario): Funcionario|false {
    $query = 'INSERT INTO Funcionario (
        idFuncionario,
        nomeFuncionario,
        email,
        senha,
        recebeValeTransporte,
        Cargo_idCargo
    ) VALUES (
        :idFuncionario
        :nomeFuncionario,
        :email,
        :senha,
        :recebeValeTransporte,
        :idCargo )';
           // Mapeia os parâmetros da query com os valores do objeto
   $params = [
            ':idFuncionario' => $funcionario->getIdFuncionario(),
            ':nomeFuncionario' => $funcionario->getNomeFuncionario(),          // Nome do funcionário
            ':email' => $funcionario->getEmail(),                              // E-mail do funcionário
            ':senha' => $funcionario->getSenha(),                              // Senha do funcionário
            ':recebeValeTransporte' => $funcionario->getRecebeValeTransporte(), // Indica se recebe vale-transporte
            ':idCargo' => $funcionario->getCargo()->getIdCargo()               // ID do cargo associado
        ];
        // Prepara e executa a instrução SQL
        $statement = Database::getConnection()->prepare($query);
        $statement->execute($params);
           // Retorna o objeto, já que o ID foi definido manualmente
        return $funcionario;
}
```



Prepara a instrução sql. Método impede um ataque comum na web chamado de ***sql injection***

*No **prepare()** os dados dos parâmetros são substituídos na instrução sql removendo qualquer possibilidade de injection*

Substitui os **:campo** Pelos valores das variáveis