Prevenindo-se contra SQL-INJECTION

O SQL Injection é uma técnica usada por usuários mal intencionados que tem como objetivo utilizar-se de falhas cometidas pelos programadores quando estão fazendo uma manipulação com um Banco da Dados qualquer. Abaixo, mostrarei como fazer de seu site mais seguro para que não haja falhas como estas que apresentei logo abaixo:

Vamos para o erro mais tradicional que muitos programadores cometem:

Digamos que na conexão com seu BD(Banco de Dados), há em alguma parte do código, uma linha parecida como esta:

```
$sql = "SELECT * FROM usuarios WHERE `usuarios`="$_POST['usuario']"
AND `senha´="$_POST['senha']";
```

Localizou ela já? Bom, então temos um grande problema. Por que? Simples, imagine que no campo de login, o usuário mal intencionado invente de colocar algo como:

```
' or 1=1--
```

E ai? Ainda não viu o problema? Eu explico, quando esse comando for escrito no campo de login e nada escrito no campo de senha e for clicado no botão para fazer login, o SQL irá entender como um comando interno e retornará este login como sendo verdadeiro e irá trazer informações do primeiro usuário deste BD. Por mais que nada tenha sido escrito no campo de senha, ele fará isso sem nenhum esforço, uma vez que o "1=1" irá informar que o usuário quer a primeira linha do BD e o "--" significa que a partir tudo vira um mero comentário, logo irá trazer informações apenas usando o Login, sem a utilização da senha. Um enorme problema, não?

Bom, vamos pegar a mesma linha 'sql' que foi citada logo acima. Agora imagine que no campo de login, o usuário decida inserir este comando:

```
douglas'--
```

Este problema é parecido com o de cima, a diferença é que ali em cima, o SQL irá retornar informações baseadas nas chaves que o usuário coloca(1=1, 2=2, ...) e neste último exemplo, o SQL retornará todas as informações que pertencerem ao login "douglas", mais uma vez, sem a necessidade de uma senha. No pior dos casos, caso o usuário tenha conhecimento de alguma forma do seu BD, ele poderia facilmente inserir um comando como esse:

```
';delete * from users--"
```

Wow, isso iria lhe dar uma baita dor de cabeça, caso não tenha feito backup do seu Banco de Dados.

Como faço então para me prevenir?

Então, há dezenas de macetes na internet para evitar essas dores de cabeça com bancos de dados, onde irei apresentar algumas delas:

1 - utilize o comando 'mysql_real_escape_string()' ao receber algum dado. Exemplo? Claro:

```
$login = mysql_real_escape_string($_POST['usuario']);
```

Este comando irá escapar todos os caracteres especiais do campo "x".

"Eu posso usar isso para a senha?"

Então, poder poder você pode, mas não é aconselhável, uma vez que você irá estar baixando a segurança do usuário, já que caracteres especiais não poderão ser postos na senha dele, algo que jamais será aconselhável fazer.

2 - utilize também o comando 'strip_tags()'. Ele irá retirar qualquer TAG HTML que o usuário tentar colocar em algum campo, seja Login ou Senha(no nosso exemplo). Exemplo de uso:

```
$login = strip_tags($_POST['usuario']);
```

*Caso necessário, você também pode adicionar as TAG's que podem ser escritas. Isso pode ser feito da seguinte forma:

```
$login = strip_tags($_POST['usuario'],'aqui você coloca as tags que podem ser aceitas');
```

3 - Comando 'trim()'. Serve para eliminar qualquer espaço que tiver no campo(qualquer campo). É como se ele pegasse tudo que ta escrito e "formasse uma
palavra só". Exemplo:

Digitei no meu campo de login algo como: " douglas martins ".

Com este comando 'trim()', o SQL irá receber a informação assim: "douglasmartins"(sem as aspas, é claro).

Exemplo de uso? Claro, bem simples:

```
$login = trim($_POST['usuario']);
```

Isso pode e deve ser feito nos campos de senha.

Criptografia

Se você já é um intermediário - veterano na área da programação, procure saber sobre 'criptografia de dados', caso já conheça sobre o assunto, nunca deixe
de utilizá-la em seus formulários, a criptografia é uma das principais camadas de segurança, com ela implantada corretamente, fica quase impossível para um invasor ter acesso a informação de qualquer usuário cadastrado no seu sistema! Por mais que você não tenha utilizado as dicas de segurança acima, quando alguém tentar acessar informações de teu BD, irá receber dados criptografados, onde o mesmo não saberá o que cada informação significa.

Por hora é só isso, nunca deixe de implementar essas dicas em suas páginas, ninguém quer ter suas informações a solta para qualquer pessoa mal intencionada usar e lhe causar muitas dores de cabeça. Abraço a todos.

Fontes:

ZoomDigital - https://goo.gl/VRjYMW

DevMedia - https://goo.gl/5GLhU6