MÓDULO

PROGRAMAÇÃO PHP

PRÁTICA 31

SEGURANÇA EM PHP UTILIZANDO O MD5

Instituto Tecnológico

SEGURANÇA EM PHP UTILIZANDO O MD5

1.1. DESCRIÇÃO DA PRÁTICA

Tempo estimado: 1 h.

REQUISITOS
■ Ter concluído a unidade didática "Segurança em PHP".
OBJETIVOS
■ Aprofundar e testar a segurança em PHP, utilizando métodos de encriptação, nomeadamente a função password_hash.
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
■ Editor de código. ■ Servidor web.

1.2. DESENVOLVIMENTO DA PRÁTICA

Na unidade didática "Segurança em PHP" foi abordada a importância de manter os dados e os conteúdos a salvo de ataques através da encriptação. Assim, nesta prática irá ser explorada um pouco mais a fundo esta segurança com o PHP, nomeadamente utilizando o MD5.

Como visto na unidade, o MD5 é capaz de encriptar, por exemplo, as senhas dos utilizadores, mas, dada a evolução da tecnologia, o MD5 já não é totalmente eficaz por ser "muito conhecido", ou seja, se o atacante souber que se trata de uma senha encriptada com MD5, também conseguirá facilmente aceder à mesma.

A solução será utilizar uma hash de segurança para fortalecer as senhas em questão. Para isso, será utilizada a função password_hash(). Esta função utiliza uma encriptação mais complexa do que o MD5 e permite guardar a nova senha na base de dados (é importante o campo da senha estar definido como varchar na tabela da base de dados).

Antes de avançar para a prática, a sintaxe da função password_hash() é a seguinte:

```
$senhaHash =  password_hash($senha, PASSWORD_DEFAULT);
```

Sendo que a variável $senhaHash será a nova senha encriptada e a variável $senha é a senha já predefinida no código ou submetida pelo utilizador.

O que torna esta função tão difícil de decifrar é que, ao contrário da MD5, ela não cria a mesma senha duas vezes, ou seja, se repetir a linha anterior duas vezes, separadas por echos, não vai obter o mesmo resultado. Desta forma, para confirmar se a senha inserida está correta, é utilizada a função password_verify(), que não funciona com os outros tipos de encriptação. A sua sintaxe é:

```
if (password_verify('senha', $senhaHash)) {
    echo 'Senha correta!';
} else {
    echo 'Senha incorreta';
}
```

Para o foco desta prática ser apenas a encriptação e a segurança, serão utilizados os ficheiros da pasta em formato zip disponibilizados no Campus Virtual: os dois ficheiros PHP e o ficheiro de estilos.

As alterações vão ser todas realizadas no ficheiro processologin.php, começando por definir duas variáveis com os valores do login (estavam anteriormente no parâmetro da condição if):

```
$emailLogin = "pratica11@teste.com";
$senha = "senha";
```

Serão as novas chaves de autenticação para esta prática. O próximo passo será encriptar a senha, com uma linha semelhante à vista anteriormente.

```
$senhaLogin =  password_hash($senha, PASSWORD_DEFAULT);
```

Agora, para verificar se a senha está correta, vai ser utilizada uma condição if que irá devolver o valor true na variável $login, caso as senhas sejam iguais.

```
if (password_verify($pwd, $senhaLogin)) {
    $login = true;
}
```

Para terminar, falta apenas alterar os valores passados no parâmetro da condição if que lhe dará as boas-vindas se o login for feito com sucesso.

```
if ($email == $emailLogin && $login == true) {
```

Se for necessário confirmar ou comparar alguma linha, segue-se o código inteiro da página processologin.php.

```
<html>
<head>
<meta charset="UTF-8">
<title>Login - Prática 8</title>
<link href="estilos.css" rel="stylesheet">
</head>
<body>
<div class="caixa0">
<span id="logo"><img src="logo.png"></span>
</div>
<div class="caixa1">
<h2>LOGIN COM SUCESSO</h2>
<?php
$email = htmlspecialchars($_POST["email"]);
$pwd = htmlspecialchars($_POST["pwd"]);
$emailLogin = "pratica11@teste.com";
$senha = "senha";
$senhaLogin =  password_hash($senha, PASSWORD_DEFAULT);

if (password_verify($pwd, $senhaLogin)) {
    $login = true;
}
if ($email == $emailLogin && $login == true) {
```

```
echo "<h2>Olá $email <br> Bemvindo.</h2>";

} else {

Header("Location:login.php?invalid=&email=$email");

}

?>

</div>

</body>

</html>
```