Universidade da Beira Interior
Departamento de Informática

Desenvolvimento de uma aplicação web segura

Elaborado por:

Inês Lopes, nº 37559
João Nobre, nº38464
Rita Correia, nº 38254
Rui Charrinho, nº 36412

Orientador:

Professor Dr. Pedro R. M. Inácio

Dezembro
2017

Conteúdo

Acrónimos

UC Unidade Curricular

UBI Universidade da Beira Interior

SI Segurança Informática

AES Advanced Encryption Standard

CBC Cipher Block Chaining

Capítulo 1

Introdução

1.1 Enquadramento

"Quando uma equipa pensa, reflete e aceita o novo como algo construtivo, é mais fácil assumir novos conhecimentos."[1]

Iniciamos este relatório com a citação apresentada no parágrafo anterior, pois pensamos que esta transmite uma forte ideia do enquadramento e objetivo deste trabalho, bem como a importância do trabalho em equipa que nele se verificou.

Este projeto é dirigido, tanto para os alunos do segundo ano do curso de Informática Web, como para os alunos de terceiro ano do curso de Engenharia Informática, da Universidade da Beira Interior (UBI), que frequentam a Unidade Curricular (UC) de Segurança Informática (SI).

1.2 Motivação

Sendo hoje em dia a Informática, uma das áreas de estudo mais abrangentes e com maiores prespetivas de crescimento, é cada vez mais importante, principalmente para estudantes de cursos desta natureza, o procurar saber mais. Toda a informação é pouca num mundo que evolui a um ritmo cada vez mais acelarado e desenvolvido, seja através deste tipo de projetos, como por iniciativa e pesquisa própria, para que mais tarde seja possível alcançar de forma mais rápida, exigente e eficaz certos objetivos profissionais, multiplicando ou triplicando os resultados até à data existentes.

Aquando a escolha do tema, e de acordo com o parágrafo anterior, o grupo achou por unanimidade que o tema escolhido seria algo desafiador, interessante e bastante útil, tanto para o futuro, como mesmo para o presente. A contrução de *sites* ou páginas *web* seguras, tem sido cada vez mais um paradigma e uma preocupação acrescida, tanto para os programadores, como para um utilizador informado e preocupado com a segurança dos seus dados, na navegação *online*.

[1] Citação de Helyane Dianno

1.3 Objetivos

O objetivo deste projeto é promover a solidificação de conhecimentos da UC em questão e a consequente aprendizagem de um conjunto de conceitos, metedologias e ferramentas no domínio da criptografia e, neste caso específico, da construção de aplicações *web* simples com uma pequena base de dados incorporada, e com a linguagem de programação PHP incluida.

1.4 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o enquadramento do projeto, a motivação para a sua escolha, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo – **Estado da Arte** – expõe o resumo da aplicação, do ponto de vista do utilizador e, por consequência, os métodos criptográficos utilizados no mesmo.

3. O terceiro capítulo – **Ferramentas utilizadas** – demonstra as ferramnetas utilizadas na realização do projeto tais como PHP , HTML e MySQL.

4. O quarto capítulo – **Implemmentação** – explicita o modo de utilização de algumas partes do código em PHP.

5. O quinto capítulo – **Conclusão** – contempla as conclusões finais e o trabalho futuro do nosso projeto.

Capítulo 2

Estado da Arte

2.1 Introdução

Este capítulo tem como objetivo, a explicação mais detalhada do projeto, assim como o início da apresentação dos métodos de cifra e segurança usados para a realização do mesmo.

2.2 A Aplicação

Este projeto, como já referido no capítulo 1, é uma simples aplicação *web*, que visualmente, e num primeiro contacto do utilizador com a mesma, apresenta um ecrã de *login*, onde o utilizador se pode autenticar, caso já tenha feito o seu registo. Caso contrário, existe uma opção para este se registar, que pede como parâmetros o nome, o *email*, e uma *password* que este vai inserir em dois campos consecutivos. Caso ambas as palavras-passe que utilizador introduziu sejam iguais e tenham um mínimo de quatro caracteres, este passa a ter uma conta criada na aplicação, e poderá assim fazer o seu *login* utilizando o nome e *password* anteriormente escolhidos.

2.3 Implementações criptográficas necessárias

De forma a garantir a segurança dos dados dos utilizadores, foram implementados e integrados vários mecanismos de criptografia, sendo estes: um máximo de 3 tentativas de *login*(caso as credenciais de acesso estejam incorretas), palavras passe cifradas com o algoritmo Advanced Encryption Standard (AES) em modo Cipher Block Chaining (CBC), o uso de um ficheiro que guarda a chave de cifra (gerada de forma segura e utilizada para cifrar as *passwords* dos utilizadores), implementção do algoritmo HMAC-SHA256 para a salvaguarda dos ficheiros e uso de um gerador aleatório para o valor de salt.

2.4 Conclusões

O grupo adquiriu conhecimento dos métodos de cifra e segurança durante as aulas, da UC de SI, os quais decidiu aprofundar.

Para esse aprofundamento, recorreu-se à pesquisa dos métodos necessários, no qual resultou o interesse acrescido para a realização do projeto.

O projeto resulta assim, da junção e implementação de todos estes métodos, numa aplicação *web* final.

Capítulo 3

Ferramentas Utilizadas

3.1 Introdução

Este capítulo refere-se ao modo como foram utilizadas as ferramentas na implmentação do código, ferramentas estas como HTML, PHP e MySQL.

Para além disto foi também gerado um certificado digital para garantir o protocolo de transferência de hipertexto seguro (HTTPS).

3.2 HTML

Uma das linguagens usadas neste trabalho foi o HTML. O seu uso consiste na criação da interface gráfica da aplicação. Para a sua implementação recorremos ao uso de, maioritariamente, botões e caixas de texto. Todas as páginas de HTML criadas utilizam, para a sua parte mais estética, referências a duas folhas de estilo CSS.

As folhas de estilo CSS melhoram o aspecto visual das páginas HTML. Elas, no nosso caso, são responsáveis pelo tamanho de letra e posição do texto, das cores, do tamanho e posição dos botões e de certas acções como o realçamento de uma caixa de texto ou de um botão.

Numa fase inical do trabalho, optámos pela utilização de JavaScript nas páginas HTML. A sua implementação foi, no entanto, eliminada nalgumas partes do projeto devido aos diversos erros que causava. O seu uso incidiria nalguns pormenores visuais, como o uso de um GIF que apareceria quando o sistema demorava curtos instantes a carregar uma nova página.

3.3 PHP

O uso de PHP foi fundamental para a realização da aplicação. Todas as instâncias (todas as páginas que são visíveis para o utilizador) utilizam esta linguagem. O PHP permite a ligação entre as diferentes páginas, assim como o seu normal funcionamento. Este também faz a ligação entre as variáveis (como o nome de utilizador e palavra chave) e a base de dados.

3.4 MySQL

O MySQL foi usado para a criação da base de dados e a sua gestão no código em PHP. O uso inicial desta ferramenta foi em linha de comandos de uma máquina Linux. Porém para a coerência entre os diferentes membros e melhor gestão da base de dados, procedeu-se à sua criação no PHPMyAdmin existente no servidor independente Xampp. Este permite a criação de uma base de dados, tendo a opção de não se trabalhar com código em MySQL.

3.5 Conclusões

O uso destas ferramentas é essencial para a implementação do código , sendo que tanto a linguagem PHP e a linguagem HTML são a base para tal e, consequentemente, a utilização de uma base de dados feita em MySQL para o armazenamento de todos os dados dos utilizadores.

Para a visualização do projeto é necessário proceder ao comando localhost/index.php. Este comando acede à página em questão e permite a navegação pelas outras páginas do projeto, sendo que para isto ser possível é preciso escrevê-lo na barra de navegação de um *browser*. Este é um comando utilizado quando se efetua a ligação para a *web* através do terminal ou quando se usa um servidor como o Xampp.

Capítulo 4

Implementação

4.1 Introdução

Este capítulo demonstra a utilização do PHP aquando o registo de um utilizador e o consequente *login* na página.

4.2 Registar um utilizador

O nosso projecto contém diversos ficheiros com várias linhas de código, pelo que iremos apenas explicar as funções mais importantes para o funcionamento da nossa aplicação *web*.

Começamos então pelo createuser.php:

```
$newuser = $_POST['newuser'];
$pw1 = $_POST['password1'];
$pw2 = $_POST['password2'];
$email=$_POST['email'];
$failedAttempts = 0;
$salt = uniqid(mt_rand(), true);
$readkey = fopen("server.key", "r");
$key = fread($readkey, filesize("server.key"));
fclose($readkey);
$enc= $pw1 . $salt;
for ($i=1; $i <=1024 ; $i++) {
    $hash = hash_hmac("sha256",$enc,$key);
}
```

Listing 4.1: Trecho de código usado que vai registar um novo utilizador à base de dados.

Esta função irá retirar os dados do utilizador (*email* e *password*) de um formulário que está presente no ficheiro do index.php . Para além disso irá retirar a *password* inserida, juntamente com o salt, e calcular o HMAC para depois inserir na base de dados. Além desses códigos o createuser.php tem várias verificações, como se as palavras-passes têm um mínimo de caracteres, se as duas *passwords*

introduzidas são iguais e se um *email* é válido. Se o utilizador inserir todos os parâmetros corretamente então aí o createuser.php irá avançar para implementação dos dados na base de dados.

Em baixo na figura 4.2 e 4.3 está o código que demostra o que foi referido anteriormente:

```
//Ver se as passes sao iguais.
if ($pw1 != $pw2) {}

//Ver se a pass tem minimo de caracteres.
elseif (strlen($pw1) < 4) {}

//Ver se as passes sao iguais.
elseif (!filter_var($email, FILTER_VALIDATE_EMAIL) == true) {}
```

Listing 4.2: Trecho de código que vai fazer várias verificações.

```
if (isset($_POST['newuser']) && !empty(str_replace(' ', '', $_POST
    ['newuser'])) && isset($_POST['password1']) && !empty(
    str_replace(' ', '', $_POST['password1']))) {

        $link = mysqli_connect('localhost', 'root', '', '
            databasesi');
        if (!$link) {
        die('Not connected : ' . mysqli_connect_error());
        }
        $sql = "INSERT INTO User (username, email, salt, password,
             failedAttempts) VALUES
                ('$newuser', '$email', '$salt','$hash','
                    $failedAttempts')";
        mysqli_close($link);
```

Listing 4.3: Trecho de código usado que vai inserir um novo utilizador à base de dados.

4.3 Login

Para além do registo do utilizador, o projeto também possui um ficheiro para fazer *login* de modo a verificar se os dados inseridos pelo utilizador estão registados na base de dados. Caso o utilizador se engane na palavra-passe terá apenas três tentativas para poder aceder à página, assim como se se enganar no *login* este será bloqueado.

```
define('DB_HOST', 'localhost');
define('DB_NAME', 'databasesi');
define('DB_USER', 'root');
define('DB_PASSWORD', '');
$con = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD) or die("
    Failed to connect to MySQL: " . mysqli_error());
//$db = mysqli_select_db(DB_NAME, $con) or die("Failed to connect
    to MySQL: " . mysqli_error());
```

```
$username = $_POST['myusername'];
$password = $_POST['mypassword'];
```

Listing 4.4: Trecho de código que vai receber os dados de utilizador o que vai permitir comparar com os dados da base de dados.

De seguida está o código que verifica primeiramente se foi inserido um *username* válido. Depois de este ser verificado irá buscar a *password* que foi inserida pelo usuário e a partir daí retirar os salt para calcular o HMAC juntamente com a chave e irá comparar a *password* que está na base de dados com a que o utilizador inseriu.

```
function SignIn($username,$password){
  if (!empty($username)){
    $link = mysqli_connect('localhost', 'root', '', 'databasesi');
    $user_get = mysqli_query($link, "SELECT * FROM User WHERE
        username='$username'");
    $row = mysqli_fetch_array($user_get);
    echo $row['username'];
    if ($row['username'] == $username){
      $failedAttempts = $row['failedAttempts'];
      if ($failedAttempts < 3){
        $salt = $row['salt'];
        $readkey = fopen("server.key", "r");
        $key = fread($readkey, filesize("server.key"));
        fclose($readkey);
        $enc= $password . $salt;
        for ($i=1; $i <=1024 ; $i++) {
            $hash = hash_hmac("sha256",$enc,$key);
        }
        if ($row['password'] == $hash){
          $failedAttempts = 0;
          $username = $row['username'];
          $newuser = $username;
          $query2 = mysqli_query($link,"UPDATE User SET
              failedAttempts = 0 WHERE username = '$newuser'") or
              die(mysql_error());
          header("Location: http://localhost/index.html");
          exit();
          }
          else{
          $failedAttempts = $failedAttempts + 1;
          $username = $row['myusername'];
          $newuser1 = $username;
          $query2 = mysqli_query($link,"UPDATE User SET
              failedAttempts = failedAttempts + 1 WHERE myusername
              ='$newuser1' ");
          exit();
          }
        }
        else{
        exit();
        }
      }
```

```
        }
    }
```

Listing 4.5: Trecho de código que vai receber os dados de utilizador e vai compará-los com os da base de dados.

4.4 Conclusões

Depois da implementação do código referido anteriormente é possivel observar o funcionamento adequando das funções demonstradas na secção 4.2 e 4.3.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

O desenvolvimento deste projeto foi bastante importante para o grupo ganhar e solidificar conhecimentos na área de SI, em criptografia e no desenvolvimento da aplicação *web*.

A construção deste projeto permitiu-nos também ter uma leve ideia de todo o esforço, trabalho e pormenor existente por detrás da criação de uma simples aplicação, como esta.

Um dos fatores notórios também a destacar neste projeto, foi todo o trabalho e esforço em equipa, que aconteceu.

Futuramente, numa empresa, o trabalho em equipa é uma componente crucial para o sucesso, pois assim é possível arranjar mais e melhores soluções para um determinado problema , demorando menos tempo a realizar uma determinada tarefa, e tornando o trabalho gerado mais efciente. Trabalhar em equipa cria sinergia, ou seja, existe uma associação concomitante de várias pessoas executoras de uma determinada função, que contribuem para uma ação coordenada, criando-se um somatório de esforços em prol de um mesmo fim. O efeito resultante desta sinergia tem, normalmente, um efeito bastante superior ao efeito de cada pessoa, se trabalhasse individualmente, sem um objetivo final comum previamente estabelecido. Como se costuma dizer: "O todo supera a soma das partes".

No entanto, e mesmo com todo este somatório de esforços, o grupo sentiu bastante dificuldade na realização deste projeto, não só devido ao tempo escasso que houve para a realização do mesmo, como também devido a toda a complexidade envolvente. Para além da implementação das ferramentas de criptografias requeridas, o grupo teve que manter uma postura auto-didata, no que toca à criação de uma base de dados e de tudo o que isso envolve e, por outro lado, no que toca à programação em PHP, nunca utilizada por qualquer elemento do grupo até então.

De maneira geral, e tendo em conta os poucos conhecimentos que temos na área, ficámos satisfeitos com o trabalho final resultante, pois tivemos a oportunidade de adquirir conhecimentos bastante bons, e também de aplicar aqueles que ganhamos em contexto de aula.

5.2 Trabalho Futuro

Na parte das três tentativas, estas não ficaram bem implementadas pois não aparece nenhuma tentativa. Quando o utilizador se engana na *password* é reencaminhado para uma página em branco onde aparece apenas o nome de resgisto que este escolheu. Neste ponto gostaríamos de ter feito uma implementação em que seria apresentada uma mensagem de erro enquanto o utilizador não acertasse na palavra-passe. Após 3 tentativas erradas o utilizador seria reencaminhado para uma página em branco que apresentaria uma contagem decrescente e quando atingisse o valor 0 apareceria um botão para voltar para a página inicial. Outra hipótese para esta opção seria o bloqueio do botão iniciar sessão, enquanto o tempo não chegasse ao fim.

Também gostaríamos de abordar mais pormenorizadamente o tema e pesquisar diversas formas, sendo elas possívelmente mais adequadas, de implementar o código neste projeto.

1. The PHP Group, PHP, http://php.net,2017/12/14
2. INÁCIO, Pedro, Guia para a Aula Laboratorial 10, https://moodle.ubi.pt/course/view.php?id=2875, 2017/12/14.