

Методы генерирования пользовательских паролей с использованием языка программирования PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будет подробно рассмотрен метод для создания уникальных комбинаций символов для паролей на сайтах с использованием языка программирования PHP. Созданная функция поможет существенно повысить безопасность личных данных пользователей.

Ключевые слова: Безопасность, PHP, сайт, пароль, защита

Methods of generation of the user passwords with use of the PHP programming language

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article the method for creation of unique combinations of symbols for passwords on the websites with use of the PHP programming language will be in detail considered. The created function will help to increase safety of personal data of users significantly.

Keywords: Safety, PHP, website, password, protection

При разработке любого сайта важно учитывать вопросы безопасности в плане хранения и получения информации. Зачастую при регистрации пользователи выбирают пароли низкого уровня защиты, что может существенно повысить риск кражи важных данных не предназначенных для общего просмотра. Предотвратить такие ситуации помогает разносторонний многофункциональный язык программирования PHP с его функционалом для генерирования комбинаций паролей.

Целью данной работы является рассмотрение стадий процесса генерирования уникальных комбинаций для паролей на различные сайты с целью повышения безопасности данных.

В сфере научно-исследовательской деятельности тема данной статьи имеет множество работ, представленных в свободный доступ. Автор С.О.Иванов выделил различные критерии оценивания создаваемого пароля с точки зрения безопасности, а также предложил новую классификацию относительно запоминания и печати пользователем [1]. В работе

исследователей О.Г. Яскевича и С.Г. Корчагина затронута тема процесса хеширования информации, подробно разобраны особенности и методики использования на примере языка программирования PHP [2]. Тему нашей статьи автор Е.С. Трубачев рассмотрел со стороны безопасности информационных ресурсов. Исследователь акцентировал внимание на международных стандартах конфиденциальности и типичных проблемах в этой сфере [3]. Интернет-источники показывают методы использования языка программирования PHP для безопасности пользовательской информации на сайте, выделяют алгоритмы для защиты паролей на любых сайтах[4-5].

В самом начале работы создадим форму для логина и пароля. Именно с ней мы будем работать далее.

```
1 <?
2 header("Content-Type:text/html;charset=utf-8");
3 ?>
4
5 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
6 <html>
7 <head>
8 <link rel="stylesheet" href="style.css">
9 <title></title>
10 </head>
11 <body>
12 <form method="post">
13   Логин: <br />
14   <input type="text" name="login">
15   <br />
16   Ваш пароль: <br />
17   <input type="text" name="password" value="<?=$res;?>">
18 </form>
19 </body>
20 </html>
```

Рис. 1. Код формы

Логин:

Ваш пароль:

Рис. 2. Вид в браузере

Теперь стоит определить критерии безопасности пароля. основополагающим из которых является способность противостоять подбору злоумышленника. И поэтому наш генератор паролей будет выдавать комбинацию из букв разного регистра и цифр, что сделает пароль максимально сложным.

Для хранения параметров функции генератора необходимо создать файл со следующим исходным кодом:

```
1 <?php
2 define('LENGTH',10);
3 ?>
```

Рис. 3. Код файла параметров

Настроим количество символом в комбинации и присоединим файл к нашей форме:

```
1 include 'config.php';
```

Рис. 4. Добавление в форму файла параметров

Далее создадим функциональный файл, содержащий все манипуляции с генератором и тоже подключим его к форме:

```
1 include 'functions.php';
```

Рис. 5. Добавление в форму функционального файла

Открываем функциональный файл и дополняем его часть get pass:

```
1 <?php
2 function get_pass() {
3 //код генератора
4 }
```

Рис. 6. Get pass

Первым шагом будет выбор символов для комбинации. Пишем строку с необходимым набором:

```
1 $str = "0123456789qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";
```

Рис. 7. Символы для комбинаций

Следующий шаг – размер строки и конечный символ.

```
1 $str_lenght = strlen($str) - 1;
```

Рис. 8. Конечный символ

Обозначаем место хранения уникального созданного пароля:

```
1 $str_gen = '';
```

Рис. 9. Расположение

В цикле функции выполняем создание комбинации:

```
1 for($i = 0; $i < LENGTH; $i++) {  
2  
3 $x = mt_rand(0,$str_lenght);  
4  
5 $str_gen .= $str[$x];  
6 }  
7 return $str_gen;
```

Рис. 10. Создание комбинации

В итоге получаем результат и выводим его на экран, используя input:

```
1 Ваш пароль: <br />  
2 <input type="text" name="password" value="<?=$res;?>">
```

Рис. 11. Результат

Логин:

Ваш пароль:

Рис. 12. Результат в браузере

Стоит уделить немного внимания ситуации, когда комбинация содержит одинаковые рядом стоящие символы. Редактируем код:

```
1 if($i != 0 ) {  
2 if($str_gen[strlen($str_gen) - 1] == $str[$x]) {  
3 $i--;  
4 continue;  
5 }  
6 }
```

Рис. 13. Редактирование кода

Теперь проверяем и видим обновленный результат:

Логин:

Ваш пароль:

Рис.14. Результат после изменений

Ниже представлен полный код для функции.

```
1 function get_pass() {
2
3     $str = "0123456789qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";
4
5     $str_lenght = strlen($str) - 1;
6
7     $str_gen = '';
8
9     for($i = 0; $i < LENGTH; $i++) {
10
11         $x = mt_rand(0,$str_lenght);
12
13         if($i != 0 ) {
14             if($str_gen[strlen($str_gen) - 1] == $str[$x]) {
15                 $i--;
16                 continue;
17             }
18         }
19         $str_gen .= $str[$x];
20     }
21
22     return $str_gen;
23 }
```

Рис. 15. Полный код функции

Таким образом, в работе были рассмотрены стадии процесса генерирования уникальных комбинаций для паролей на различные сайты с целью повышения безопасности данных.

Библиографический список

1. Иванов С.О. Критерии оценки пароля // Сборник научных трудов молодых ученых и специалистов. 2015. С. 187-192. URL: <https://elibrary.ru/item.asp?id=26041873> (Дата обращения: 06.01.2019)
2. Яскевич О.Г., Корчагин С.Г. Хеширование паролей (на примере языка PHP) // Интеллектуальные информационные системы. 2017. С. 132-135. URL: <https://elibrary.ru/item.asp?id=32787258> (Дата обращения: 06.01.2019)
3. Трубачёв Е.С. Проблемы информационной безопасности. Методы и средства защиты информационных ресурсов // Вестник Волжского Университета им. В.Н. Татищева. 2009. №14. С. 47-54. URL: <https://elibrary.ru/item.asp?id=17660332> (Дата обращения: 06.01.2019)
4. Функция генерации пароля на PHP URL: <https://www.it-rem.ru/funktsiya-generatsii-parolya-na-php.html> (Дата обращения: 06.01.2019)
5. Защита паролей в современном PHP URL: https://www.ibm.com/developerworks/ru/library/wa-php-renewed_2/ (Дата обращения: 06.01.2019)