

## **Лабораторная работа 9**

### **Предотвращение CSRF**

#### **Индивидуальность отчетов:**

Как минимум, в имени пользователя ОС

**Обязательно скриншотить каждый этап и пояснять**

**Обязательно выводы должны быть**

## 1. Создать каталог /var/www/edu/csrf

## 2. Создать таблицу users в базе данных

Table name:  Add  column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
<input type="text" value="id"/> <small>Pick from Central Columns</small>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/> <small>PRIMARY</small>
<input type="text" value="user"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="255"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>
<input type="text" value="password"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="255"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>

## 3. Создать таблицу user\_data в базе данных

Table name:  Add  column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
<input type="text" value="user_id"/> <small>Pick from Central Columns</small>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="UNIQUE"/> <small>[user_id]</small>
<input type="text" value="secret"/> <small>Pick from Central Columns</small>	<input type="text" value="VARCHAR"/>	<input type="text" value="255"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>

Примечание. В настоящих системах надо использовать хэш с солью от пароля вместо самого пароля

## 4. Создать пользователя и зададим ему значение secret

+ Options

	id	user	password
<input type="checkbox"/> Edit Copy Delete	1	admin	12345

+ Options

	user_id	secret
<input type="checkbox"/> Edit Copy Delete	1	secret for admin

## 5. Создать страницу авторизации login.php

```
login.php x
login.php
1  <?php
2  $username = null;
3  $password = null;
4
5  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
6      if (!empty($_POST["username"]) && !empty($_POST["password"])) {
7          $dbh = new PDO('mysql:host=localhost;dbname=edu', 'newuser', '123');
8
9          $username = $_POST["username"];
10         $password = $_POST["password"];
11
12         $sql = "SELECT * FROM `users` WHERE user = :username AND password = :password";
13         $stmt = $dbh->prepare($sql);
14         $stmt->bindParam(':username', $username, PDO::PARAM_STR);
15         $stmt->bindParam(':password', $password, PDO::PARAM_STR);
16         $stmt->execute();
17         $result = $stmt->fetch(PDO::FETCH_ASSOC);
18
19         if ($result) {
20             session_start();
21             $_SESSION["authenticated"] = 'true';
22             $_SESSION["userid"] = $result["id"];
23             header('Location: index.php');
24         } else {
25             header('Location: login.php');
26         }
27     } else {
28         header('Location: login.php');
29     }
30 }
31 ?>
32 <html>
33 <head>
34     <title>Login Page</title>
35 </head>
36 <body>
37     <form id="login" method="POST">
38         <label for="username">Username:</label>
39         <input id="username" name="username" type="text" required>
40         <label for="password">Password:</label>
41         <input id="password" name="password" type="password" required>
42         <input type="submit" value="Login">
43     </form>
44 </body>
45 </html>
46
```

## 6. Создать auth.php

auth.php ×

auth.php

```
1  <?php
2  session_start();
3  if (empty($_SESSION["authenticated"]) || $_SESSION["authenticated"] != 'true') {
4      header('Location: login.php');
5  }
6
```

## 7. Создать index.php

```
index.php X
index.php
1  <?php
2  require_once('auth.php');
3
4  $dbh = new PDO('mysql:host=localhost;dbname=edu', 'newuser', '123');
5
6  // change secret
7  $newsecret = null;
8  if ($SERVER['REQUEST_METHOD'] == 'POST') {
9      if (!empty($_POST["newsecret"])) {
10         $newsecret = htmlspecialchars($_POST["newsecret"]);
11         $sql = "UPDATE `user_data` SET `secret` = :newsecret WHERE `user_id` = :user";
12         $stmt = $dbh->prepare($sql);
13         $stmt->bindParam(':newsecret', $newsecret, PDO::PARAM_STR);
14         $stmt->bindParam(':user', $_SESSION["userid"], PDO::PARAM_INT);
15         $stmt->execute();
16     }
17 }
18
19 // get user data
20 $sql = "SELECT * FROM `user_data` WHERE `user_id` = :user";
21 $stmt = $dbh->prepare($sql);
22 $stmt->bindParam(':user', $_SESSION["userid"], PDO::PARAM_INT);
23 $stmt->execute();
24 $result = $stmt->fetch();
25 ?<
26
27 <html>
28 <head>
29 | <title>User Page</title>
30 </head>
31 <body>
32 | <div>User ID: <?php echo $result["user_id"]; ?></div>
33 | <div>Secret:</div>
34 | <div><?php echo $result["secret"]; ?></div>
35
36 | <form id="change_secret" method="POST">
37 | | <label for="secret">New secret:</label>
38 | | <input id="secret" name="newsecret" type="text" required>
39 | | <input type="submit" value="Change">
40 | </form>
41 </body>
42 </html>
```

## 8. Проверить работоспособность

```
a@a: /var/www/edu/csrf
a@a:/var/www/edu/csrf$ ls -l -a
total 20
drwxr-xr-x 2 a www-data 4096 ноя  3 03:19 .
drwxr-xr-x 4 a www-data 4096 ноя  3 03:13 ..
-rwxr-xr-x 1 a www-data  139 ноя  3 02:37 auth.php
-rwxr-xr-x 1 a www-data 1191 ноя  3 03:13 index.php
-rwxr-xr-x 1 a www-data 1349 ноя  3 02:52 login.php
a@a:/var/www/edu/csrf$
```

edu.local/csrf/login.php

Username:  Password:

edu.local/csrf/index.php

User ID: 1

Secret:

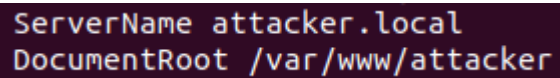
new admin secret

New secret:

## 9. Создать сайт злоумышленника

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/custom-2.conf
```

```
sudo nano /etc/apache2/sites-available/custom-2.conf
```



```
ServerName attacker.local
DocumentRoot /var/www/attacker
```

```
sudo a2ensite custom-2.conf
```

```
sudo service apache2 restart
```

Добавить в /etc/hosts строчку

```
127.1.1.1 attacker.local
```

Создать /var/www/attacker

Зададим права доступа

```
sudo chmod 755 -R ./
```

```
sudo chown имя_пользователя:www-data -R ./
```

## 10. Создать файл index.php в /var/www/attacker

```
index.php x
index.php
1  <html>
2  <body>
3      <form id="change_secret" style="display:none;" action="http://edu.local/csrf/index.php" method="POST">
4          <label for="secret">New secret:</label>
5          <input id="secret" name="newsecret" type="text" value="attacker" required>
6          <input type="submit" value="Change">
7      </form>
8
9      <script>
10         window.onload = () => document.getElementById("change_secret").submit();
11     </script>
12 </body>
13 </html>
```

## 11. Авторизоваться на edu.local

## 12. Открыть attacker.local

## 13. Подробно описать в отчете, что произошло и почему



## 14. Изменим файл /var/www/edu/csrf/index.php

```
index.php x
index.php
1  <?php
2  require_once('auth.php');
3
4  $dbh = new PDO('mysql:host=localhost;dbname=edu', 'newuser', '123');
5
6  // adding CSRF token if not exists
7  if (!isset($_SESSION['CSRFToken'])) {
8      $token = base64_encode(openssl_random_pseudo_bytes(32));
9      $_SESSION['CSRFToken'] = $token;
10 }
11
12 // change secret
13 $newsecret = null;
14 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
15     // check CSRF token from form and from session storage
16     if (!empty($_POST['CSRFToken']) && $_POST['CSRFToken'] == $_SESSION['CSRFToken']) {
17         if (!empty($_POST['newsecret'])) {
18             $newsecret = htmlspecialchars($_POST["newsecret"]);
19             $sql = "UPDATE `user_data` SET `secret` = :newsecret WHERE `user_id` = :user";
20             $stmt = $dbh->prepare($sql);
21             $stmt->bindParam(':newsecret', $newsecret, PDO::PARAM_STR);
22             $stmt->bindParam(':user', $_SESSION["userid"], PDO::PARAM_INT);
23             $stmt->execute();
24         }
25     }
26 }
27
28 // get user data
29 $sql = "SELECT * FROM `user_data` WHERE `user_id` = :user";
30 $stmt = $dbh->prepare($sql);
31 $stmt->bindParam(':user', $_SESSION["userid"], PDO::PARAM_INT);
32 $stmt->execute();
33 $result = $stmt->fetch();
34 ?<
35
36 <html>
37 <head>
38     <title>User Page</title>
39 </head>
40 <body>
41     <div>User ID: <?php echo $result["user_id"]; ?></div>
42     <div>Secret:</div>
43     <div><?php echo $result["secret"]; ?></div>
44
45     <form id="change_secret" method="POST">
46         <label for="secret">New secret:</label>
47         <input id="secret" name="newsecret" type="text" required>
48         <!-- adding hidden field with CSRF token -->
49         <input type="hidden" name="CSRFToken" value="<?php echo($_SESSION['CSRFToken']) ?>">
50         <input type="submit" value="Change">
51     </form>
52 </body>
53 </html>
```

Строчки 6-9: создаётся случайным образом CSRF токен, если он ещё не был создан, и сохраняется в хранилище сессий

Строчки 15, 16, 25: проверка получения токена из формы и сравнения его с сохранённым токеном в хранилище сессии

Строчки 48, 49: создание скрытого поля формы, содержащего ранее созданный токен

**15. Убедиться в работоспособности смены секрета на edu.local**

**16. Убедиться, что атака больше невозможна, открыв attacker.local**

**17. Сделать финальные выводы по всей работе**