

Лабораторная работа 6

Предотвращение атак, связанных с SQL-инъекциями

Индивидуальность отчетов:

Как минимум, в имени пользователя ОС

Обязательно скриншотить каждый этап и пояснять

Обязательно выводы должны быть

1. Создадим виртуальную машину Ubuntu 20.04.1

2. Обновим установленные пакеты

```
sudo apt update
```

```
sudo apt upgrade
```

3. Установим Apache

```
sudo apt install apache2
```

4. Выполнить команды

```
sudo adduser ваше_имя_пользователя www-data
```

```
sudo chown -R www-data:www-data /var/www
```

```
sudo chmod -R g+rwX /var/www
```

5. В отчете раскрыть, что конкретно выполнялось этими командами

6. Установим MySQL

```
sudo apt install mysql-server
```

```
sudo mysql_secure_installation
```

```
sudo mysql -uroot
```

```
CREATE USER 'имя_польз'@'localhost' IDENTIFIED BY 'пароль';
```

```
GRANT ALL PRIVILEGES ON * . * TO 'имя_польз'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
alter user 'имя_польз'@'localhost' identified with mysql_native_password by  
'пароль';
```

```
exit
```

7. Установим phpmyadmin

```
sudo apt install phpmyadmin
```

```
sudo nano /etc/apache2/apache2.conf
```

в конце добавить

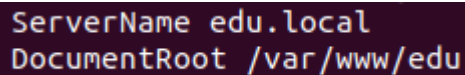
```
Include /etc/phpmyadmin/apache.conf
```

```
sudo service apache2 restart
```

8. Настроим VirtualHost

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/custom-1.conf
```

```
sudo nano /etc/apache2/sites-available/custom-1.conf
```



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

```
sudo a2ensite custom-1.conf
```

```
sudo service apache2 restart
```

8. Настроим hosts

Добавим в /etc/hosts строчку

```
127.1.1.1 edu.local
```

9. Перейти в /var/www/edu (либо создать)

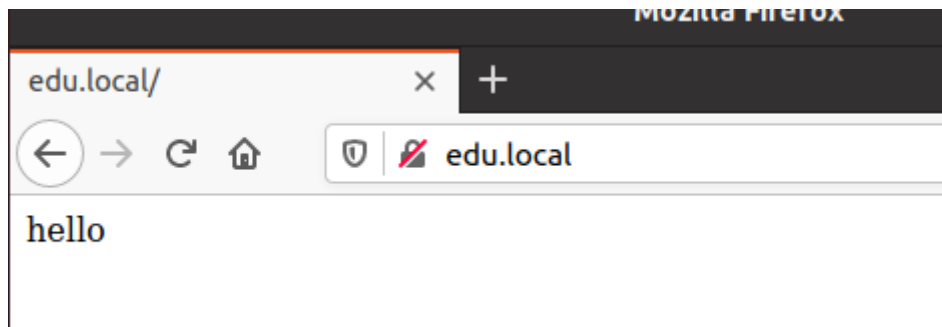
```
sudo chmod 755 -R ./
```

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

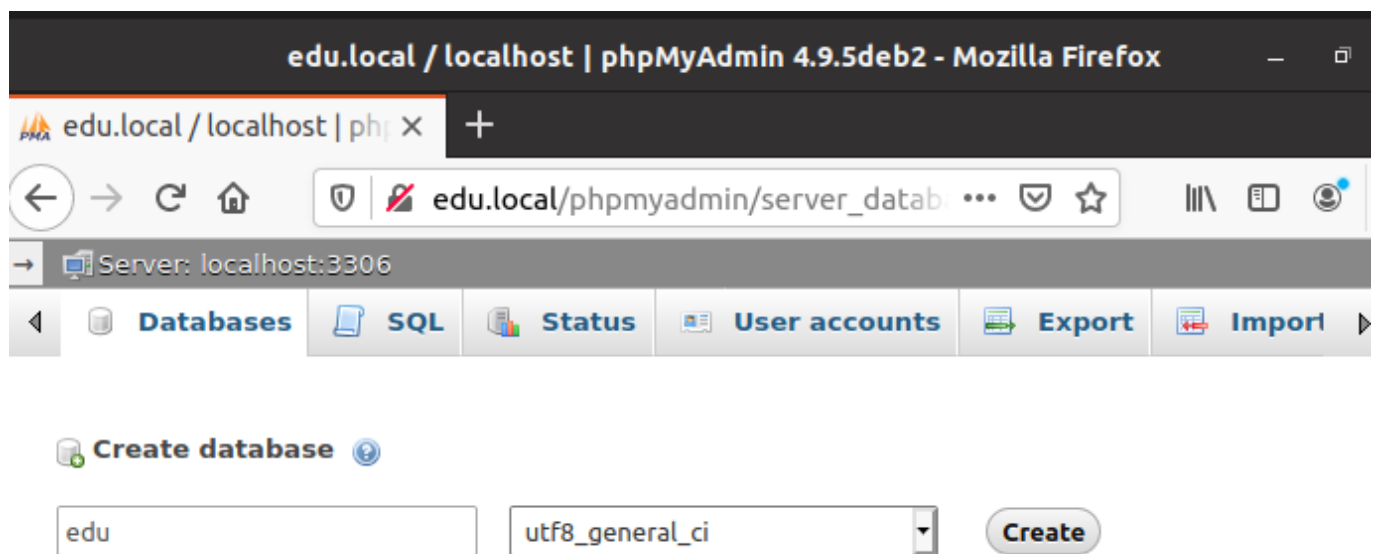
10. В /var/www/edu создать файл index.php с содержимым:

```
<?php  
    echo "hello";
```

11. Открыть в браузере edu.local

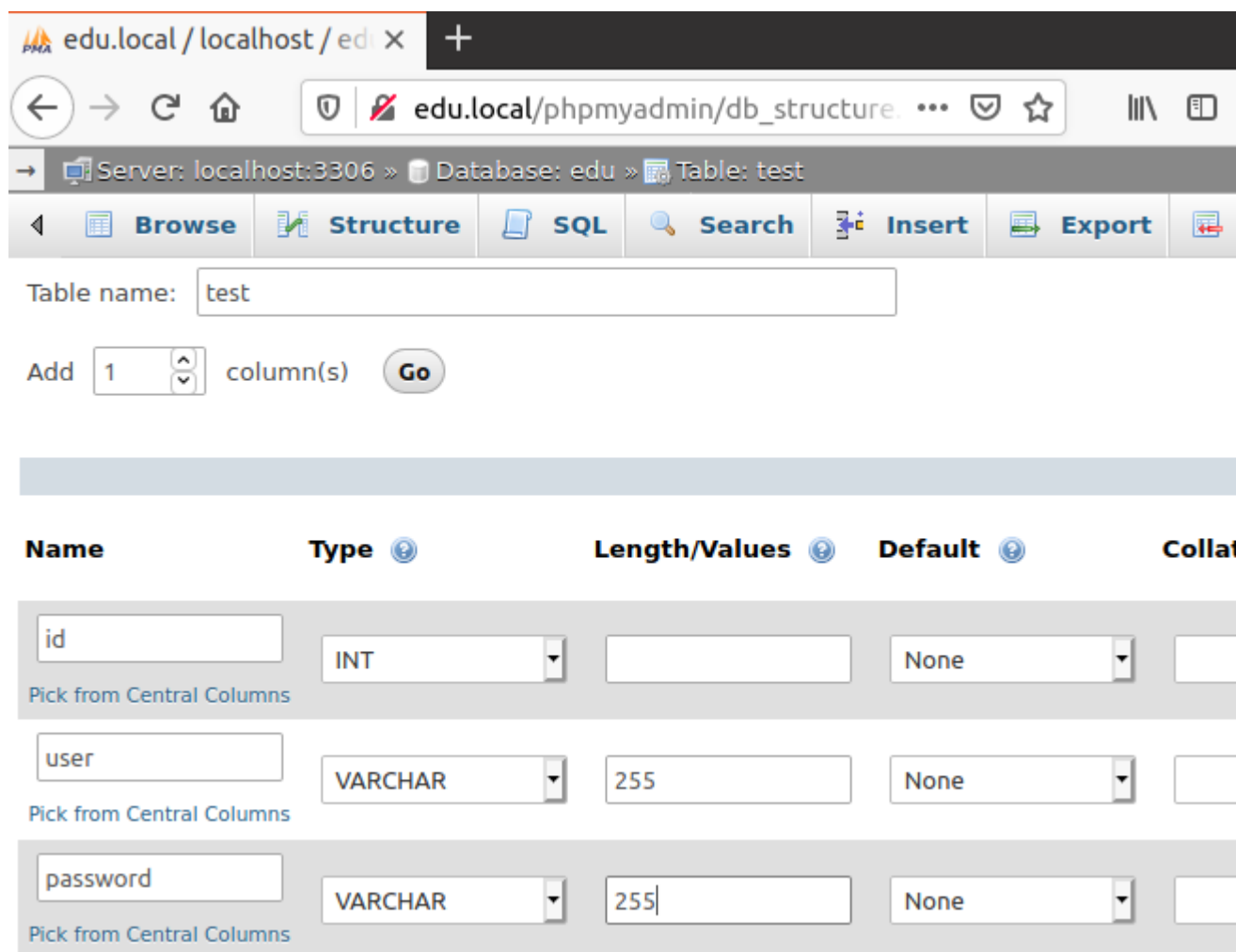


12. Откроем phpmyadmin и создадим тестовую базу данных
edu.local/phpmyadmin



13. Создадим таблицу в базе данных

Для ID выбрать auto increment (галка A_I) и согласиться с primary key



edu.local / localhost / edu X +

edu.local/phpmyadmin/db_structure. ...

Server: localhost:3306 » Database: edu » Table: test

◀ Browse Structure SQL Search Insert Export ▶

Table name: test

Add 1 column(s) Go

Name	Type	Length/Values	Default	Collat
id	INT		None	
Pick from Central Columns				
user	VARCHAR	255	None	
Pick from Central Columns				
password	VARCHAR	255	None	
Pick from Central Columns				

14. Создадим строки в базе данных

```
Run SQL query/queries on table edu.test: ⓘ  
1 INSERT INTO `test`(`user`, `password`) VALUES ("admin", "12345")
```

15. Убедимся, что всё правильно создано

Navigation: [Browse](#) | [Structure](#) | [SQL](#) | [Search](#) | [Insert](#)

✓ Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM `test`
```

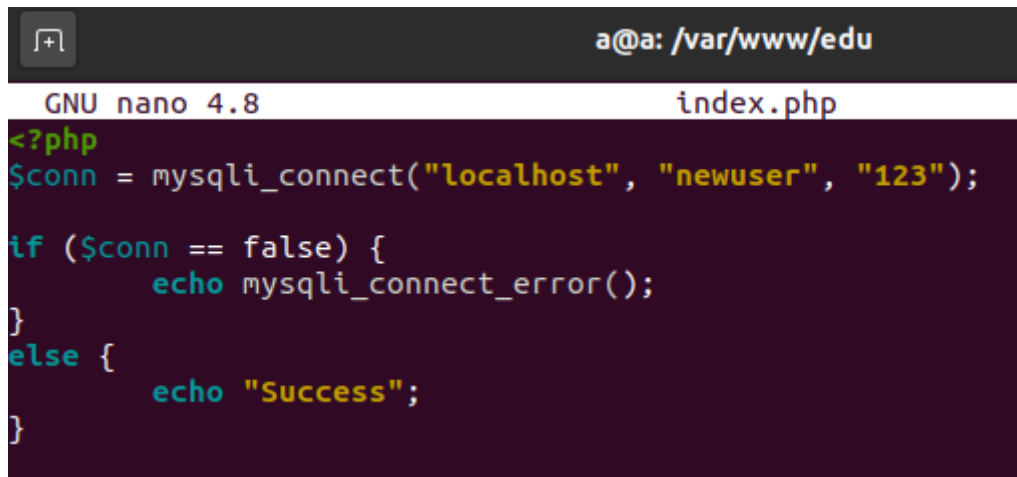
☐ Profiling [\[Edit inline\]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create \]](#)

☐ Show all | Number of rows:

+ Options

				id	user	password
<input type="checkbox"/>		Edit		Copy		Delete
1	admin	12345				
<input type="checkbox"/>		Edit		Copy		Delete
2	user	54321				

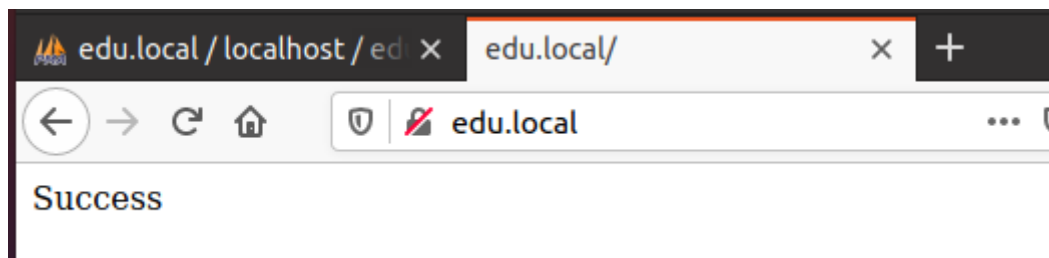
16. Изменим наш index.php



```
a@a: /var/www/edu
GNU nano 4.8 index.php
<?php
$conn = mysqli_connect("localhost", "newuser", "123");

if ($conn == false) {
    echo mysqli_connect_error();
}
else {
    echo "Success";
}
```

17. Откроем в браузере и проверим



18. В отчете подробно описать, что произошло

19. Добавим пользовательские данные

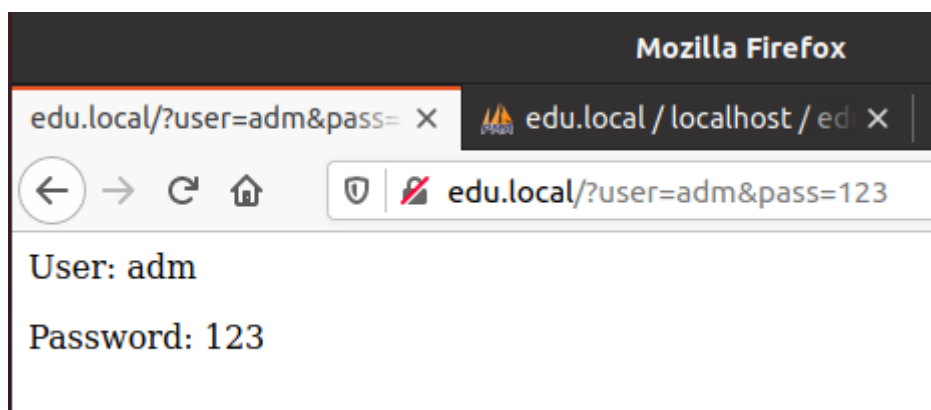
Для упрощения будем использовать GET-параметры. Так не стоит делать в реальных системах

```
a@a: /var/www/edu
GNU nano 4.8 index.php
?php
$conn = mysqli_connect("localhost", "newuser", "123");

if ($conn == false) {
    echo mysqli_connect_error();
}
else {
    echo "";
}

$username = $_GET["user"];
$password = $_GET["pass"];

echo "<p>User: " . $username . "</p>";
echo "<p>Password: " . $password . "</p>";
```

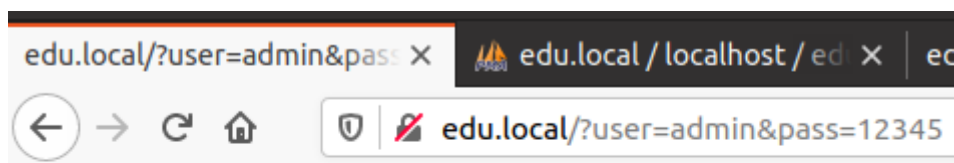


20. Добавим проверку существования записи пользователя с введенными логином и паролем

В конец файла допишем

```
$sql = "SELECT * FROM `edu`.`test` WHERE user = '$username' AND password = '$password'";  
$result = mysqli_query($conn, $sql);  
  
if (mysqli_num_rows($result)){  
    echo "<p>Login successful</p>";  
}  
else {  
    echo "<p>Wrong username/password</p>";  
}
```

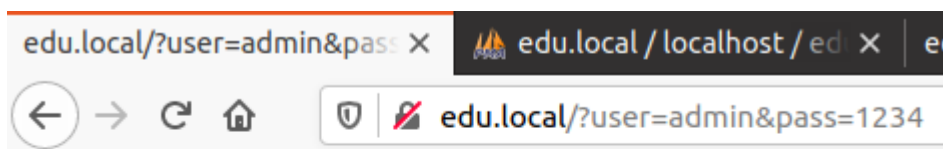
Проверим:



User: admin

Password: 12345

Login successful

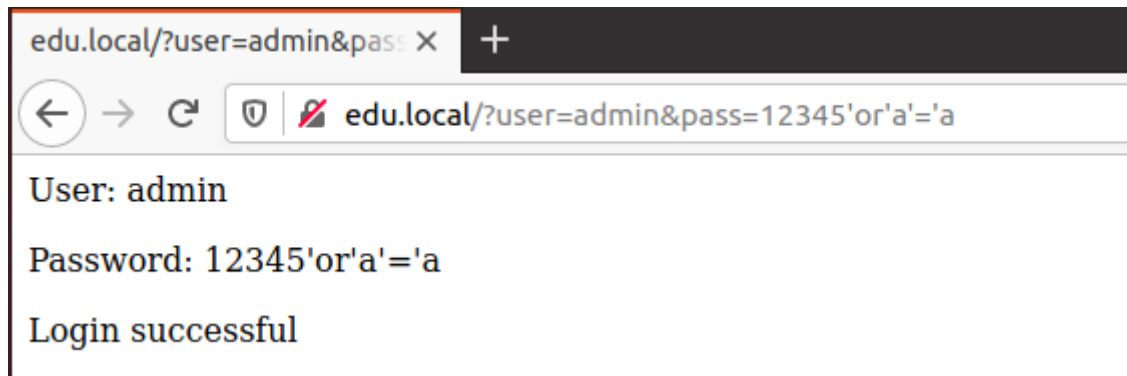


User: admin

Password: 1234

Wrong username/password

21. Используем SQL-инъекцию



22. В отчете описать, что произошло, почему проверка логина и пароля прошла успешно

23. Перепишем index.php в части работы с базой данных на PDO

```
GNU nano 4.8
<?php
$dbh = new PDO('mysql:host=localhost;dbname=edu', 'newuser', '123');

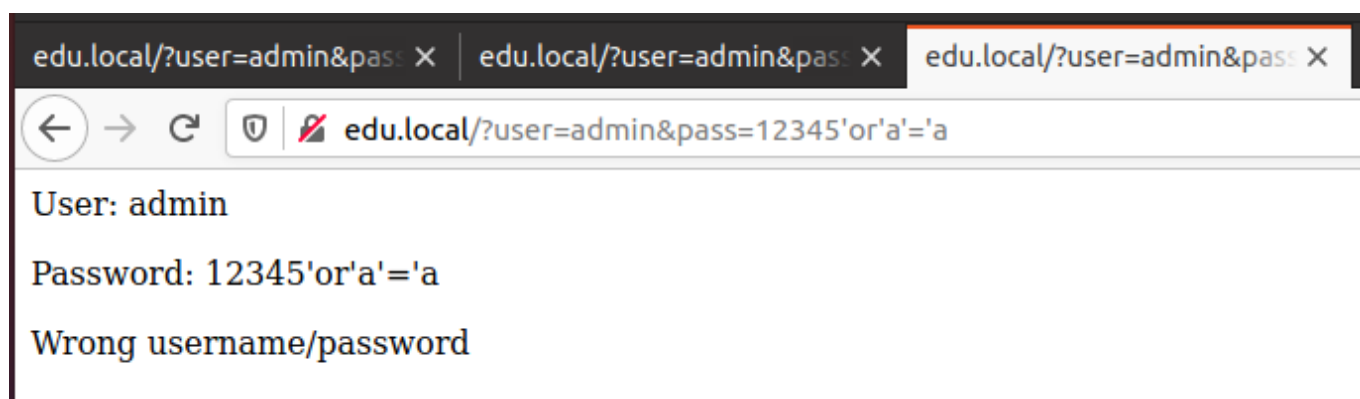
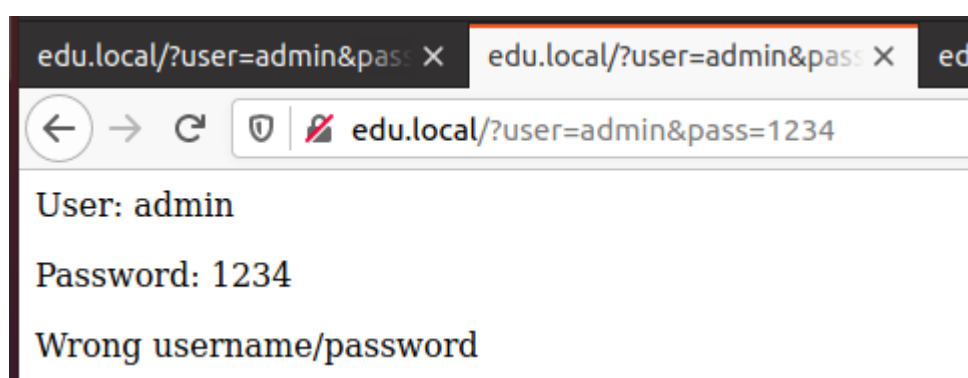
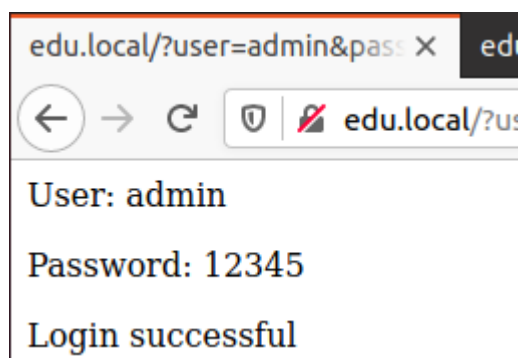
$username = $_GET["user"];
$password = $_GET["pass"];

echo "<p>User: " . $username . "</p>";
echo "<p>Password: " . $password . "</p>";

$sql = "SELECT * FROM `test` WHERE user = :username AND password = :password";
$stmt = $dbh->prepare($sql);
$stmt->bindParam(':username', $username, PDO::PARAM_STR);
$stmt->bindParam(':password', $password, PDO::PARAM_STR);
$stmt->execute();
$result = $stmt->fetch(PDO::FETCH_ASSOC);

if ($result){
    echo "<p>Login successful</p>";
}
else {
    echo "<p>Wrong username/password</p>";
}
```

24. Проверим и убедимся, что SQL-инъекция теперь невозможна



25. В отчете описать, что произошло. Описать, почему SQL-инъекция теперь не работает

26. Описать в отчете, что такое и как работает PDO