Лабораторная работа 8. Часть 1.

Предотвращение атак, связанных с XSS

Индивидуальность отчетов:

Как минимум, в имени пользователя ОС

Обязательно скриншотить каждый этап и пояснять

Обязательно выводы должны быть

1. Откроем ранее созданную виртуальную машину Ubuntu

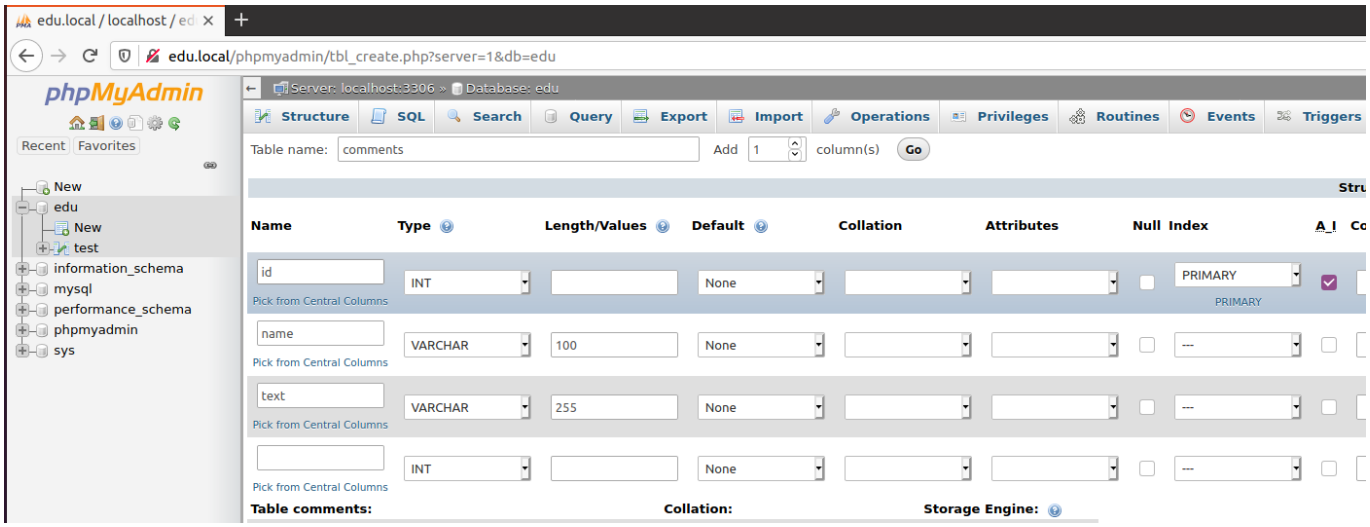
2. Создадим таблицу comments в базе edu

С полями:

id int primary A_I

name varchar 100

text varchar 255



3. Создадим папку /var/www/edu/comments

4. Создадим файл style.css

```css
.form {
  width: 300px;
  margin-top: 50px;
  margin-left: auto;
  margin-right: auto;
}

.form div {
  margin-bottom: 10px;
}

.form input {
  width: 100%;
}

.form textarea {
  width: 100%;
  height: 150px;
}

.comments {
  width: 300px;
  margin-top: 100px;
  margin-left: auto;
  margin-right: auto;
}

.comments__item {
  margin-top: 30px;
}

.comments__name {
  color: #969696;
}

```

5. Создадим файл index.php

```php
<?php
$dbh = new PDO('mysql:host=localhost;dbname=edu', 'newuser', '123');
$name = $_POST["name"];
$comment = $_POST["comment"];
if (!empty($name) && !empty($comment)) {
    $sql = "INSERT INTO `comments` (`name`, `text`) VALUES (:name, :text)";
    $stmt = $dbh->prepare($sql);
    $stmt->bindParam(":name", $name, PDO::PARAM_STR);
    $stmt->bindParam(":text", $comment, PDO::PARAM_STR);

    if (!$stmt->execute()) {
        echo "ERROR";
    }
}

$sql = "SELECT `name`, `text` FROM `comments`";
$comments = $dbh->prepare($sql);
$comments->execute();

?>
<html>

<head>
  <title>Комментарии</title>
  <link rel="stylesheet" href="./style.css" />
</head>

<body>
  <section class="form">
    <form action="" method="POST">
      <div>Имя:</div>
      <div><input name="name" type="text"></div>
      <div>Комментарии:</div>
      <div><textarea name="comment"></textarea></div>
      <div><input type="submit"></div>
    </form>
  </section>

  <section class="comments">
    <h1>Комментарии</h1>
    <div>

<?php
foreach ($comments->fetchAll() as $row) {
    echo '<div class="comments__item">';
    echo '<div class="comments__name">' . $row['name'] . '</div>';
    echo '<div>' . $row['text'] . '</div>';
    echo '</div>';
}
?>

    </div>
  </section>

</body>

</html>
```

6. Создадим файлик reflected.php

```php
<?php
$category_name = $_GET["category"];

echo "<p>Category: " . $category_name . "</p>";

```

7. Создадим файлик dom.html

```html
<script>
  let hash = window.location.hash.substring(1);
  eval(hash);
</script>
```

8. /var/www/edu/comments должна иметь такой вид

```
a@a:/var/www/edu/comments$ ls -l -a
total 24
drwxr-xr-x 2 www-data www-data 4096 окт  6 03:39 .
drwxrwxr-x 3 www-data www-data 4096 окт  6 02:39 ..
-rwxr-xr-x 1 www-data www-data   81 окт  6 03:39 dom.html
-rwxr-xr-x 1 www-data www-data 1325 окт  6 03:09 index.php
-rwxr-xr-x 1 www-data www-data   91 окт  6 03:13 reflected.php
-rwxr-xr-x 1 www-data www-data  386 окт  6 02:41 style.css
```

9. Использовать XSS

Использовать в качестве вредоносного кода:

<script>alert(1);</script>

Либо:

alert(1);

На страницах поэксплуатировать уязвимость в учебных целях:

http://edu.local/comments

http://edu.local/comments/reflected.php

http://edu.local/comments/dom.html

В отчет добавить скриншоты и пояснения для каждого случая