7 задание BackEnd

Проведен аудит безопасности веб-приложения, включающего функционал подачи заявок, авторизации и администрирования. Выявлены и устранены следующие уязвимости: XSS, Information Disclosure, SQL Injection, CSRF, Include и Upload.

1. Защита от XSS (Cross-Site Scripting)

Уязвимость: В нескольких местах выводится пользовательский ввод без экранирования.

Методы защиты:

- Экранирование выводимых данных с помощью `htmlspecialchars()`.
- Использование Content Security Policy (CSP).

```
<?php
$user_input = "<script>alert('XSS');</script>";
$safe_output = htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8');
echo $safe_output; // Выведет: &lt;script&gt;alert('XSS');&lt;/script&gt;
?>
```

2. Защита от Information Disclosure (1 балл)

Уязвимость: Утечка чувствительной информации (например, версии PHP, путей к файлам).

Методы защиты:

- Отключение вывода ошибок в production-среде.
- Настройка веб-сервера для скрытия версий ПО.

Пример кода (PHP):

```
<?php
ini_set('display_errors', 0);
ini_set('log_errors', 1);
?>
```

3. Защита от SQL Injection (2 балла)

Уязвимость: Возможность внедрения SQL-кода через пользовательский ввод.

Методы защиты:

- Использование подготовленных выражений (prepared statements).
- **Пример кода (PHP + PDO):**

```php
<?php
$pdo = new PDO('mysql:host=localhost;dbname=test', 'user', 'password');
$stmt = $pdo->prepare('SELECT * FROM users WHERE username = :username');
$stmt->execute(['username' => $_POST['username']]);
$user = $stmt->fetch();
?>
```

4. Защита от CSRF (Cross-Site Request Forgery) (2 балла)

Уязвимость: Возможность выполнения действий от имени пользователя без его ведома.

Методы защиты:

- Использование CSRF-токенов.

Пример кода:

```php
<?php
session_start();
if (empty($_SESSION['csrf_token'])) {
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}
?>
<form method="POST">
    <input type="hidden" name="csrf_token" value="<?php echo
$_SESSION['csrf_token']; ?>">
   // <!-- Остальные поля формы -->
</form>
```

Проверка токена:

```php
<?php
if ($_POST['csrf_token'] !== $_SESSION['csrf_token']) {
    die('Неверный CSRF-токен');
```

```
}
?>
```

5. Защита от Include и Upload уязвимостей (1 балл)

Include уязвимости

Уязвимость: Возможность включения произвольных файлов.

Методы защиты:

- Ограничение путей включения файлов.
- Валидация входных данных.

Пример кода:

```php
<?php
$allowed_pages = ['home.php', 'about.php'];
$page = $_GET['page'];
if (in_array($page, $allowed_pages)) {
    include($page);
} else {
    include('404.php');
}
?>
```

Upload уязвимости

Уязвимость: Загрузка вредоносных файлов.

Методы защиты:

- Проверка расширения и MIME-типа файла.
- Сохранение файлов вне корневой директории.

Пример кода:

```php
<?php
$allowed_types = ['image/jpeg', 'image/png'];
$uploaded_type = $_FILES['file']['type'];
if (in_array($uploaded_type, $allowed_types)) {
    move_uploaded_file($_FILES['file']['tmp_name'], '/safe/directory/' .
basename($_FILES['file']['name']));
```

```
} else {
    die('Недопустимый тип файла');
}
?>
```

Закючение

В ходе аудита были выявлены и устранены ключевые уязвимости веб-приложения. Примененные методы защиты соответствуют современным стандартам безопасности. Для дальнейшего улучшения рекомендуется регулярно обновлять зависимости и проводить повторные аудиты.