**Оглавление**

[Аудит безопасности 2](#_Toc110276903)

[1. Уязвимости XSS 2](#_Toc110276904)

[2. SQL Injection 3](#_Toc110276905)

# Аудит безопасности

## Уязвимости XSS

Слабые места были обнаружены в местах получения GET-параметров при выводе детальной страницы категории, страницы товара.

Для устранения была создана функция-обработчик intHandling(), которая приводит полученное значение к типу *int*, тем самым убирает все возможные специальные символы и непредвиденные конструкции возвращая число (листинги 1-2).

Листинг 1 — Обработка числовых параметров (int).

1. function intHandling**(**$value**)** **{**
2. **return** intval**(**$value**);**
3. **}**

Пример использования в коде:

Листинг 2 — Получение данных о категории по параметру GET[‘id’] (productListController.php).

1. /\*\*
2. \* Получение данных о категории и запись в переменные, которые используются в category\_detail.php
3. \*/
4. **if(**$\_SERVER**[**'REQUEST\_METHOD'**]** **==** 'GET' **&&** isset**(**$\_GET**[**'cat\_id'**]))** **{**
5. $catID **=** intHandling**(**$\_GET**[**'cat\_id'**]);**
6. $currentPage **=** **!**empty**(**$\_GET**[**'page'**])** **?** intHandling**(**$\_GET**[**'page'**])** **:** 1**;**
7. $category **=** selectOne**(**'category'**,** **[**'category\_id' **=>** $catID**]);**
8. **if(**empty**(**$category**))** **{**
9. redirectNotFoundPage**();**
10. **}**
11. $catID **=** $category**[**'category\_id'**];**
12. $catName **=** $category**[**'name'**];**
13. $catDescription **=** $category**[**'description'**];**
14. $products **=** getAllProductsWithCatID**(**$catID**,** $currentPage**);**
15. **if(**empty**(**$products**))** **{**
16. redirectNotFoundPage**();**
17. **}**
18. $countRows **=** countFoundRowsOnLastQuery**();**
19. **}**

Так же слабые места были обнаружены в обработчике формы обратной связи. Для устранения слабых мест создана функция stringHandling(), которая обрабатывает строковые параметры убирая теги HTML, убирает лишние пробелы, и преобразует специальные символы в HTML-сущности (листинги 3-4).

Листинг 3 — Обработка строковых параметров.

1. function stringHandling**(**$value**)** **{**
2. $value **=** trim**(**$value**);**
3. $value **=** strip\_tags**(**$value**);**
4. **return** htmlspecialchars**(**$value**);**
5. **}**

Пример использования в коде:

Листинг 4 — Обработка данных полученных с формы методом POST (feedbackFormController.php)

1. **if(**$\_SERVER**[**'REQUEST\_METHOD'**]** **=** 'POST' **&&** isset**(**$\_POST**[**'btn-submit'**]))** **{**
2. $username **=** stringHandling**(**$\_POST**[**'username'**]);**
3. $email **=** stringHandling**(**$\_POST**[**'email'**]);**
4. $birthday **=** stringHandling**(**$\_POST**[**'birthday'**]);**
5. $gender **=** intHandling**(**$\_POST**[**'gender'**]);**
6. $title **=** stringHandling**(**$\_POST**[**'title'**]);**
7. $description **=** stringHandling**(**$\_POST**[**'description'**]);**
8. $params **=** **[**'username' **=>** $username**,**
9. 'email' **=>** $email**,**
10. 'birthday' **=>** $birthday**,**
11. 'gender' **=>** $gender**,**
12. 'title' **=>** $title**,**
13. 'description' **=>** $description**];**
14. **if(**checkFormParameters**(**$username**,** $email**,** $birthday**,** $title**,** $description**))** **{**
15. insertInQuestions**(**$params**);**
16. **}**
17. setcookie**(**'username'**,** $params**[**'username'**],** **time()** **+** 3600**);**
18. setcookie**(**'email'**,** $params**[**'email'**],** **time()** **+** 3600**);**
19. setcookie**(**'birthday'**,** $params**[**'birthday'**],** **time()** **+** 3600**);**
20. setcookie**(**'gender'**,** $params**[**'gender'**],** **time()** **+** 3600**);**
21. **}**

## SQL Injection

Для предотвращения sql-инъекций применяются все те же функции stringHandler(), intHandler() описанные выше.

Используется подготовленный запрос на вставку в таблицу с обращениями (с формы обратной связи).

Листинг 5 — Вставка в таблицу questions при помощи подготовленного запроса.

1. function insertInQuestions**(**$params**)** **{**
2. global $pdo**;**
3. $sql **=** "INSERT INTO questions(name, email, birthday, gender, title, description)
4. VALUE(:username, :email, :birthday, :gender, :title, :description)"**;**
5. $query **=** $pdo**->**prepare**(**$sql**);**
6. $query**->**execute**(**$params**);**
7. **return** $pdo**->**lastInsertId**();**
8. **}**