

Урок 1. Что такое XSS

Задание 1:

Внимательно изучите все задачи из раздела «Практика» данной методички и ответьте: как можно использовать поисковые запросы при поиске уязвимостей XSS?

Решение:

Можно использовать дорк с фильтром по урлу, который ищет, например, страницы с параметрами Search, Contact, Feedback. Вот [страница](#) с ТОП 100 дорков для поиска XSS уязвимости.

Жаль что гугл не предоставляет фильтр по исходному коду. Ну либо я о нём не знаю. Так бы было проще найти уязвимости типа XSS.

Задание 2:

Допустим, вы обнаружили, что на странице есть уязвимый к XSS параметр, в который можно выполнить инъекцию вектором `<script>alert(document.cookie)</script>`. Как проверить, к какому типу относится инъекция (Reflected, Stored, DOM, Self или Blind)? Ответ обоснуйте.

Решение:

Если результат инъекции виден сразу на этой же странице, но после обновления страницы payload необходимо загружать снова, чтобы эксплуатировать найденную XSS, то это Reflected.

Если результат инъекции виден сразу на этой же странице и после обновления данный payload сохранился на сервере и каждый раз выполняется, то это Stored.

Так же этот вектор может относиться к DOM-based XSS, т. к. имеет полное право не показываться в исходном коде страницы.


Данный вектор не может быть применим к Self типу XSS, т. к. от такого payload'а никакого импакта не будет. Злоумышленник не получит совершенно никакой информации.

Задание 3*:

Попробуйте решить задание 1 из <https://unescape-room.jobertabma.nl/> в разделе practice. В качестве решения приложите скриншот.

Решение:

The unescape() room

 **Level 1 (practice)**

Level 1 | [New](#) | [Stop](#)

Challenge: call the `tallHuman` function with argument `1` (string) by exploiting the **XSS** vulnerability.

[View HTML source](#) [View DOM](#)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello world</title>
  </head>
  <body>
    Hello, <script>tallHuman('1')</script>
  </body>
</html>
```