Лабораторная №3

XSS и CSRF

Цель работы:

Получить представление о том, какие бывают уязвимости в безопасности веб-сайтов и как они возникают.

Задача 1.

Разработать веб-приложение, состоящее из двух страниц. Первая предлагает пользователю ввести некоторый текст и сохранить его на сервере. Вторая выводит пользователю сохранённый текст в двух вариациях: с экранированием символов и без. Текст не обязательно хранить в базе или на диске, достаточно хранить его в памяти.

Необходимо продемонстрировать эксплуатацию хранимой XSS на данном сайте. Для этого необходимо ввести в поле ввода и сохранить строку вида

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

После этого при открытии страницы должен произойти вывод сообщения «1». При этом одно поле будет пустым, второе будет содержать приведённый выше ввод.

Полезные ссылки:

https://ru.wikipedia.org/wiki/Межсайтовый_скриптинг

http://kmb.ufoctf.ru/web/xss/main.html

Задача 2.

Разработать 2 веб-приложения. Первое содержит 2 формы ввода с несколькими полями (одна форма должна иметь защиту от CSRF, другая нет). Содержимое форм сохраняется на сервере и отображается на странице (возможно объединить страницы ввода форм и вывода их текущего сохранённого состояния). Также существует кнопка генерации cookie, при нажатии на которую

сервер генерирует случайную последовательность в качестве cookie, сохраняет её и передаёт клиенту. Сохранение формы должно происходить только если cookie клиента совпадает с серверной. Второе приложение должно содержать единственную страницу, которая содержит скрытую копию форм первого приложения и код их отправки, например:

```
<form action="https://bank.example.com/send.php" style="display: none;">
    <input type="hidden" name="sum" value="9000">
    <input type="hidden" name="target" value="счет злоумышленника">
</form>
<script>
document.forms[0].submit();
</script>
```

Таким образом, при заходе на страницу приложения 2 должно меняться состояние одной из сохранённых форм приложения 1 (в случае, если в приложении 1 данным браузером была сгенерирована cookie).

Полезные ссылки:

https://github.com/codedokode/pasta/blob/master/security/xsrf.md

https://habr.com/ru/company/oleg-bunin/blog/412855/