**Доклад**

**по теме**

**«Межсайтовый скриптинг (XSS - атака)»**

**Сулицкого Богдана Романовича**

**(группа НФИбд02 - 20)**

Оглавление

[Постановка задачи. 2](#_Toc149188626)

[Способы атаки 2](#_Toc149188627)

[Типы XSS уязвимостей: 3](#_Toc149188628)

[Атака с SQL инъекцией 3](#_Toc149188629)

[Результаты атаки 3](#_Toc149188630)

## Постановка задачи.

XSS (Cross-Site Scripting — межсайтовый скриптинг) — тип веб-атаки, заключающийся во внедрении на страницу сайта или приложения вредоносного кода.

И тут же встаёт множество вопросов. А что этот код может делать? А что мне с этим вредоносным кодом делать, как разработчику или пользователю? В конце концов, как можно на сайт ввести что-то, не будучи при этом администратором сайта? И так, обо всём по порядку.

## Способы атаки

А начнём мы, как и полагается, с конца. Расскажем сначала, как это нечто вообще появляется на сайте. Неужто сначала нужно как-то пробраться на сервер и переписать код сайта. Почти, только никуда пробираться не нужно. Суть XSS атаки в том, что вы переписываете сайт со стороны пользователя. А позволяет это делать язык, на котором сайт и написан. Программы, написанные на так называемом клиентском языке программирования, выполняются непосредственно браузером на вашем компьютере. А раз все скрипты запускаются вашим браузером, то к ним можно подмешать что-либо постороннее, что сервер не заметит, а вот ваш браузер выполнит с готовностью.

Но чтобы активировать скрипт он должен как-то попасть на сайт, но как?

«Одной из проблем современных браузеров является их неспособность отличить обычный текст от кода. То есть чтобы браузерная программа распознала и выполнила код HTML, он должен быть размечен тегами. Соответственно, Java-скрипты должны помещаться между тегами <script>, CSS-стили — между CSS и т. д.» Для исправления подобной уязвимости, например, достаточно при написании сайта экранировать области, в которые будущие пользователи будут вводить текст. Сделать так, чтобы системные теги не читались браузером, автоматически заменяя их.

## XSS уязвимости делятся на три типа:

Хранимые – те, что уже находятся на сайте, который вы посещаете. Самый распространённый сценарий – поля комментариев и всяческой другой подобной переписки. Вы могли даже не дойти до поля, в котором злоумышленник прописал скрипт, а сайт уже в момент загрузки всё сделал за вас. И раз, а ваши данные уже гуляют по сети.

Отражённые – менее скрытые и более распространённые XSS атаки. Если вам никто никогда не говорил, что по незнакомым ссылкам лучше не переходить – я не поверю. Собственно, через них скрипт подобного типа и активируется. Итог тот же. Нажали на кнопку – преступник получил результат.

Атака основанная на DOM-модели. Примечателен этот вид тем, что, во-первых, не привязан к языку, на котором написан сайт, но к DOM модели (это такой программный интерфейс, позволяющий скриптам получать доступ к HTML документам, а также менять и содержимое, структуру и оформление). А во-вторых, Атаку на DOM модель очень сложно отследить, ибо вся магия происходит исключительно на стороне пользователя, не отправляя вредоносную нагрузку на сервер.

Так же можно разделить атаки на активные (когда нам ничего не нужно делать, чтобы скрипт сработал, просто посетить сайт) и пассивные (когда нам нужно не полениться и тыкнуть на странно выглядящую ссылку).

## Атака с SQL инъекцией

Ещё существует атака с SQL инъекцией SiXSS. Она задействует базы данных, расположенные на сервере. То есть, в отличии от собратьев SiXSS будет исходить уже не из клиента, а с самого сервера и попадёт к пользователю при использовании заражённой базы.

## Результаты атаки

Что же делают скрипты, когда мы их активируем? Начинается разнообразие эффектов, например, считыванием введённого текста, а заканчивается там, где кончается фантазия злоумышленника и браузер. А именно, всё, до чего может браузер и всё, что в нём сохранено – потенциально в руках у устроителя атаки, к счастью, не больше, но и не меньше.

ИТОГ:  
XSS атака использует всё, что есть на сайте, чтобы запустить скрипт, так или иначе получающей доступ к браузеру или личным данным пользователя с самого сайта.

Триггером атаки может послужить скрипт, расположенный в полях для ввода текста пользователями и меняющих сайт или его отображение через DOM, или ссылка, отправленная злоумышленником и делающая то же самое.

Так же существует атака с SQL инъекцией, которая инициируется злоумышленником, отправившим на сервер заражённую SQL базу данных и могущая сработать в любой момент.

Что с этим делать программистам? Вычищать сайт на предмет явных уязвимостей по типу отсутствия экранирования и молиться.

Что делать пользователям? Молиться и не переходить по подозрительным ссылкам.

*Что аз же многогрешный на бренных сих листах   
недописах поспешно или переписах,   
то спереди и сзади читая во все дни,   
исправи правды ради, писанья ж не кляни.*

Доклад подготовлен Сулицким Богданом и котом Мефистофелем (один раз закрыл файл).