



HTML5 - взгляд сквозь призму безопасности

Иващенко Т., Сидоров Д.
Chaos Constructions 2010

КТО МЫ?

Иващенко Тарас

Специалист по информационной безопасности, проповедник свободного программного обеспечения, немного параноик. Участник проекта W3AF.

Дмитрий Сидоров

Менеджер антивирусного проекта Яндекса, исследователь в области безопасности, еще больший параноик.

Повестка

- HTML5 - что он нам принесёт
- Новые теги – новые векторы атак
- <video> и <audio>
- Оффлайн хранение данных
- Междоменное взаимодействие – то, о чём так долго мечтали и боялись
- Определение местоположения – нюансы
- Web SQL Injection

История

- 1991 – HTML
- 1995 – HTML 2.0 (загрузка файлов, таблицы)
- 1996 – CSS 1 + JavaScript
- 1997 – HTML 3.2 (W3C)
- 1997 – HTML 4.0
- 1998 – CSS 2
- 1999 – HTML 4.01



История (продолжение)

- 2000 – XHTML 1
- 2002 – "Вёрстка без таблиц"
- 2005 – AJAX
- 2009 – HTML 5 (черновик)

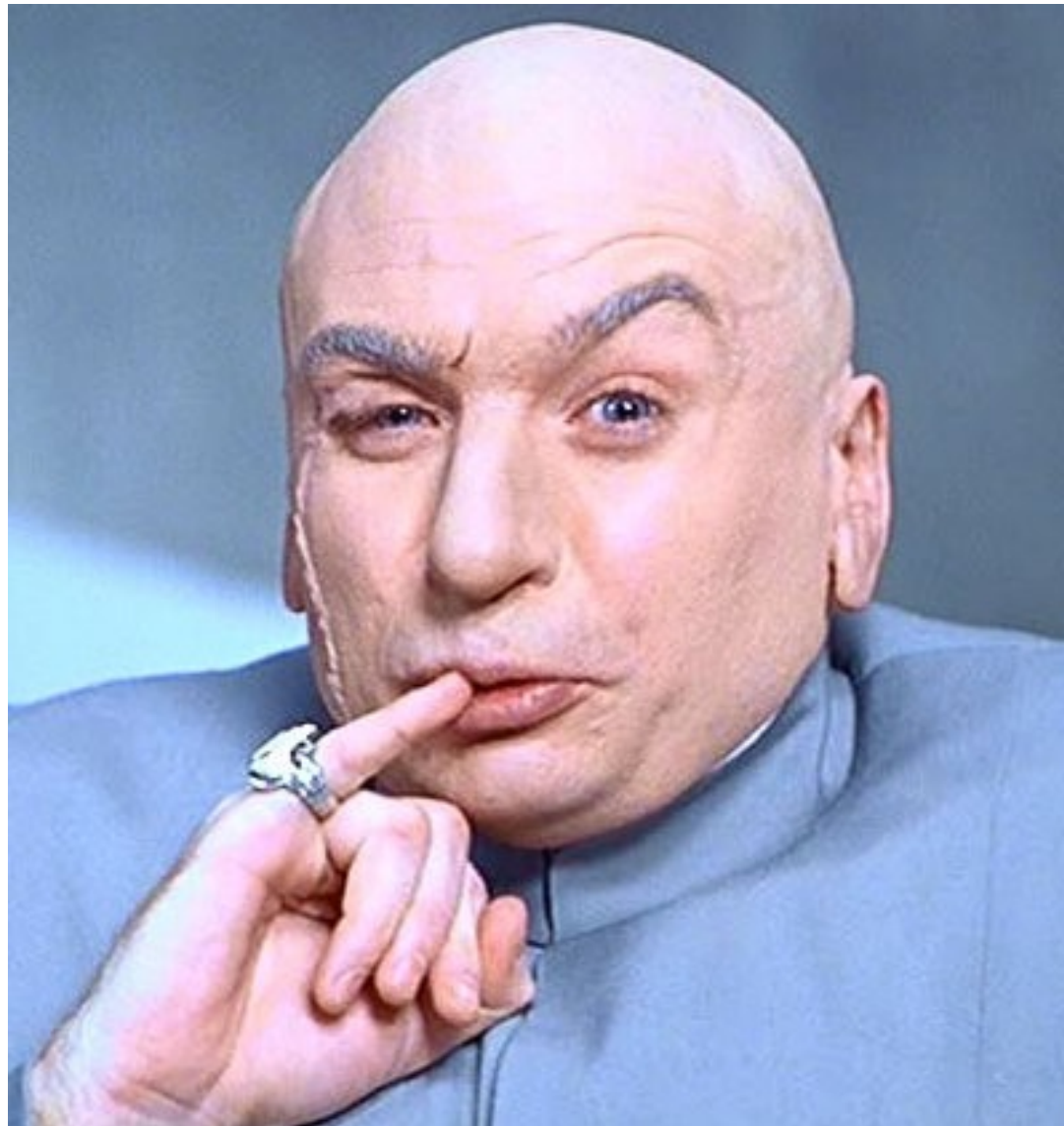
HTML5

- Будущая пятая версия одного из главных языков разметки Интернета
- Несмотря на то, что пока ещё доступен только черновик стандарта, много "вкусного" уже реализуется в популярных веб-браузерах, среди этого:
 - “Оффлайн” хранение данных в браузере – веб-хранилище, локальные БД
 - Canvas 2D API

HTML5 (продолжение)

- Междоменное взаимодействие (Cross Domain Messaging)
- Управление проигрыванием видеороликов
- "Drag-and-drop"-функционал
- Работа с сетью – веб-сокеты
- Определение местоположения (Geolocation)

Картинка для привлечения внимания




Новые теги – новые векторы атак

В стандарте добавились новые теги и атрибуты – это значит, что пора обновлять правила вашего WAF


```
// Автофокусировка как способ автоматического исполнения
кода
<input onfocus=alert(1) autofocus>
<input onblur=write(1) autofocus><input autofocus>
// Через атрибут poster тега video
<video poster=javascript:alert(1)//
<video><source onerror="javascript:alert(1)">
// Самовыполнение JavaScript с помощью обработчика
onscroll тега <BODY> и autofocus
<body onscroll=alert(1)><br><br><br>...<br><input
autofocus>
```


НОВЫЕ КОНТРОЛЫ ФОРМ

- Добавились новые типы для `<input>` тега: email, number, color, tel, range
- “Вера вебмастеров в новые контролы”

Range `<input type="range" min="0" max="50" value="0" />` 

Input Validation

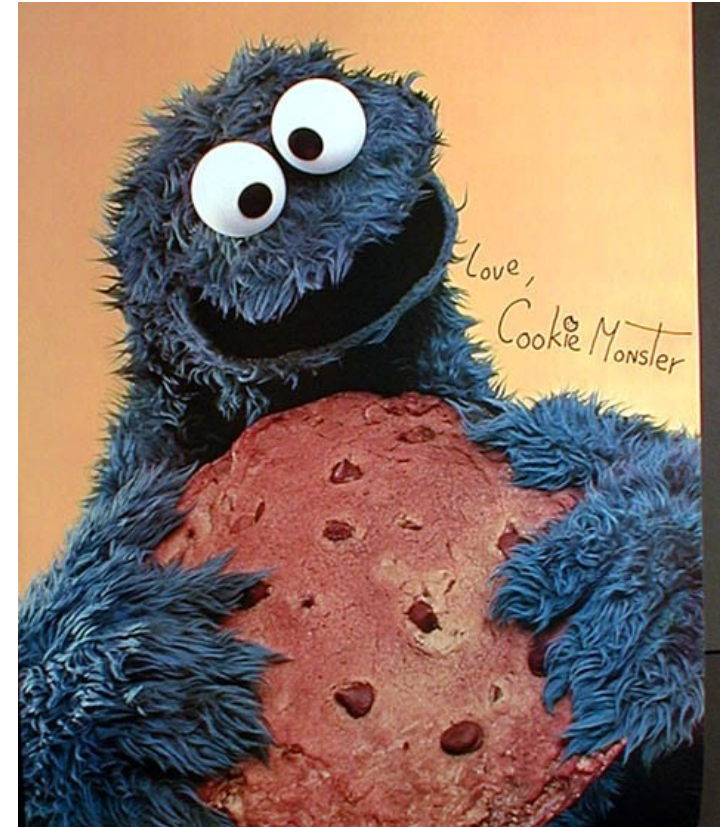
Number `<input type="number" />` 

Email `<input type="email" value="some@email.com" />` 

<video> и <audio>

- Точное определение браузера, в зависимости от используемых кодеков для проигрывания
- Еще один метод в копилку Metasploit Decloak(<http://www.decloak.net/>)
- Такие-же проблемы как и у + subtitles

Веб-хранилище - оффлайн хранение данных



Веб-хранилище - оффлайн хранение данных

- <http://www.w3.org/TR/webstorage/>
- Более вместимое, чем куки, хранилище вида "ключ-значение" на стороне веб-браузера с доступом из JavaScript
- localStorage – для долговременного хранения, sessionStorage – для сессионного применения
- Firefox 3.5, Safari 4.0, IE8, Google Chrome, Opera 10.50

localStorage – пример

```
<p>Вы просматривали эту страницу <span  
id="count">сколько-то </span> раз.</p>  
<script>  
if (!localStorage.pageLoadCount)  
    localStorage.pageLoadCount = 0;  
localStorage.pageLoadCount += 1;  
document.getElementById('count').textContent =  
localStorage.pageLoadCount;  
</script>
```

Веб-хранилище и безопасность

- Подчиняется механизму **HTML5 Origin**, то есть данные доступны для всех страниц на одном домене (+ протокол и порт)
- Ограничения по размеру данных (рекомендуется 5 МБ на домен): Firefox, Safari, Opera, Google Chrome – 5МБ, IE – 10МБ

localStorage и Firefox

В Firefox действует лимит на .example.com, таким образом один поддомен, может занять всё место, отведённое для домена

```
// Firefox 3.6.8
for (var i = 0; i < 100; i++) {
  try {
    localStorage.setItem(rand(1, 10000).toString() +
      'foo'+i.toString(), 'AA...AA'+i.toString());
  } catch (e) {alert(i.toString()+'|'+e);break;}}
```

Вставка null byte в ключ localStorage приводят к “забычивости” Firefox

LocalStorage и Google Chrome

А в Google Chrome можно занять **всё дисковое пространство**, создав кучу iframe на wildcard домен.

```
<script>
    for(var i=0; i<10; i++) {
        var iframe =
document.createElement('iframe');
        iframe.src = 'http://'+randomString()
+'.example.com/ddos.html';
        document.body.appendChild(iframe);
    }
</script>
```

Веб-хранилище и безопасность?

От кук к новому виду хранилища перекочевали и старые проблемы:

- Отслеживание пользователей
- DNS-спуфинг атаки

Помимо этого из-за особенностей ограничения доступа (протокол+домен+порт) имеем проблемы на `example.com/~user/`

Cross-Document Messaging

- <http://dev.w3.org/html5/postmsg/>
- Веб-браузеры по причинам безопасности запрещают взаимодействие (доступ и обмен данными) документов, размещённых на разных доменах.
- Система междокументных сообщений позволяет (в идеале) безопасным способом обмениваться данными документам, размещённым на разных доменах
- Firefox, Google Chrome

Cross Domain Messaging

Сайт a.example.com взаимодействует с foo.com

example.com/index.html

```
PostMessage('Hello', 'http://foo.com/iframe.html');
```



```
window.addEventListener('message',  
receiver, false);
```

foo.com/iframe.html

Пример

```
<div id="msg">...</div><script>
window.addEventListener('message', receiver, false);
function receiver(e) {
    if (e.origin !== 'http://example.com') {return;}
    document.getElementById('msg').innerHTML =
    'Origin: ' + e.origin + ' From: ' + e.source + '
Data: ' + e.data;}
</script>
```


```
<script>
function postMsg() {
var o = document.getElementById('ifra');
o.contentWindow.postMessage(document.getElementById('msg')
.value, 'http://foo.com/messaging.html');
return false;
}</script>
```

CDM - безопасность

- Разработчики должны явно проверять атрибут origin перед использованием данных
- Так же не нужно забывать валидировать сами принятые данные
- Использование (*) в качестве targetOrigin
- Перерождение DOM-based XSS

Определение местоположения

- <http://www.w3.org/TR/geolocation-API/>
- Вообще говоря, не является частью HTML5, но при этом часто упоминается в одном контексте
- Безопасность ограничивается доменом
- Пересылка не только GPS-координат
- XSS на разрешённом для сбора координат сайте ведёт к печальным последствиям

 slides.html5rocks.com wants to know your location. [Learn More...](#)

Определение местоположения

“Если вы согласитесь, Firefox соберёт **информацию о ближайших точках беспроводного доступа и IP-адресе вашего компьютера**. Затем Firefox **отправит** эту информацию провайдеру сервиса определения местоположения по умолчанию, а именно Google Location Services, чтобы определить ваше приблизительное местоположение. “,
<http://www.mozilla.com/ru/firefox/geolocation/>

Пример кода

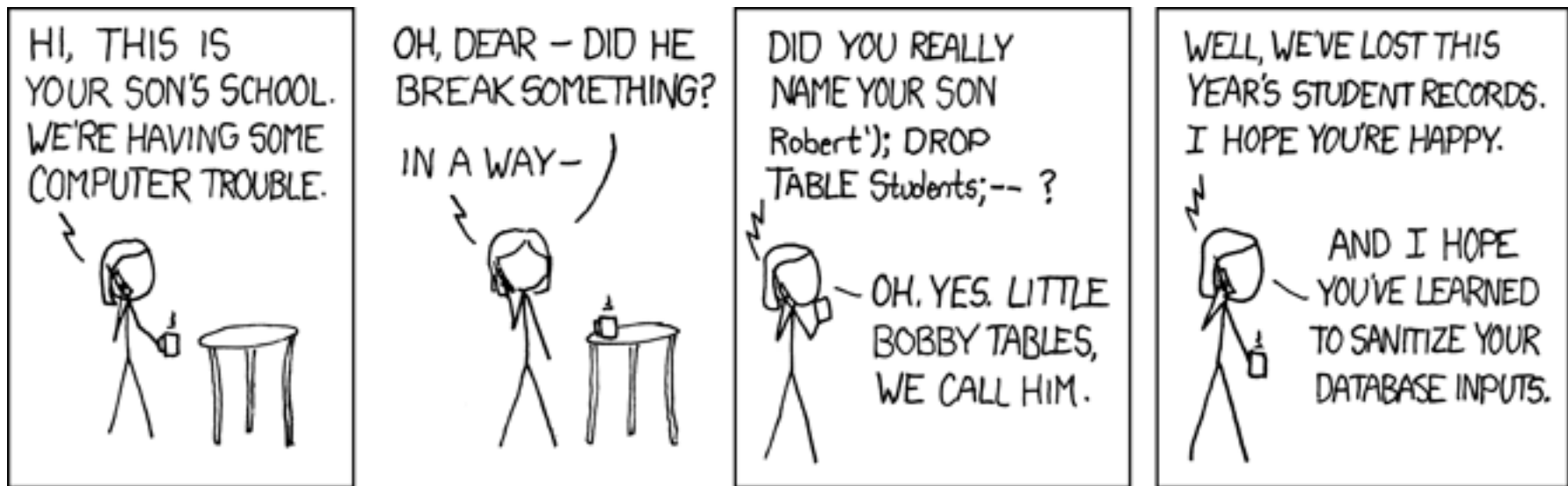
```
if (navigator.geolocation) {  
  
navigator.geolocation.getCurrentPosition(function(position  
) {  
    var lat = position.coords.latitude;  
    var lng = position.coords.longitude;  
    var options = { position: new google.maps.LatLng(lat,  
lng) }  
    var marker = new google.maps.Marker(options);  
    marker.setMap(map);  
});  
}
```

Web SQL Database

- <http://dev.w3.org/html5/webdatabase/>
- Такие же проблемы как у localStorage и sessionStorage
- Новый вектор атаки – Web SQL injection
- Спецификация предусматривает защиту от SQL Injection
- Спецификацией рекомендуется запретить выполнение SQL в IFRAME

Web SQL Injection

```
var param = get_url_param( 'id' );
db.readTransaction(function (t) {
    t.executeSql('SELECT title, author FROM docs WHERE
id=' + id, [], function (t, data) {
        report(data.rows[0].title, data.rows[0].author);
    });});
```



W3AF



- Фреймворк для проведения аудита безопасности веб-приложений
- GPLv2
- Уже добавлены модули для поиска мест использования Web Storage. Добавим и другое.

Информационные источники

- <http://dev.w3.org/html5/spec/>
- <http://en.wikipedia.org/wiki/HTML5>
- “HTML5 Security Cheatsheet”,
<http://heideri.ch/jso/>
- <http://www.html5rocks.com/>

Вопросы?