

硕 士 研 究 生 读 书 报 告



题目 HTML5新特性与安全性

作者姓名 张浩

作者学号 21651043

指导教师 李启雷

学科专业 移动互联网与游戏制作

所在学院 软件学院

提交日期 二○一七年一月

New characteristics and security issues of HTML5

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Li Qilei

By

Zhang Hao

Zhejiang University, P.R. China

2017

摘要

随着HTML5标准规范的制定完成，再加上这些年的不断推广，HTML5在未来必将成为web应用开发的主流。因此作为开发人员，我们有必要对HTML5的新特性以及其安全性问题做一些了解。因HTML5特性众多，且本文篇幅有限，故在下文中简单介绍了HTML5其中一部分特性，包括图形及特效特性、网页多媒体特性、本地存储特性、性能与集成特性，文中对这些特性进行了简单描述并介绍了其优缺点。新的技术必然带来新的风险与安全性问题，故本文在后面介绍了一些随着新特性而引发的新安全问题与攻击手法，其中包括了XSS攻击、HTML5一些API和socket的滥用以及僵尸网络。文章对这些攻击手法进行了简单介绍，并分析是由哪些新特性引起了这些安全问题。最后在文章结尾对上述内容进行了总结，并对HTML5的未来做了展望。

**关键词： HTML5 新特性 网络安全**

Abstract

With the HTML5 standards completed, plus the further promotion of these years, HTML5 will become the mainstream of Web application development in the future. As a developer, we need know the new features of HTML5, as well as the safety problem. HTML5 features numerous, and space is limited, so are briefly introduced below part of the HTML5 features, including graphics and special effects features, Web pages, multimedia features, local storage features, performance and integration, in the introduction to these features are briefly described their advantages and disadvantages. New technologies also will bring new risks and security issues, this article introduces some of the new security questions and attacks caused by the the new features in the back, including the XSS attack, HTML5 API and socket abuse and botnets. Article simply introduced the attack , and also analysis these security problems are caused by which new features. Finally at the end of the article summarizes these and future prospect of HTML5.

**Keywords: HTML5 new features network security**

1. 新特性介绍

HTML自问世以来到现在已经有23年的历史，其1999年12月发布的HTML4.0影响最为深远并一直沿用至今。但在2004的时候，WHATWG提出了Web Applications 1.0，也就是HTML5草案的前身，并于2007年被W3C正式采纳，于是HTML的的第五次重大修改正式开始，并于2014年终于正式完成。HTML5最初的设计目的之一就是为了在移动设备上支持多媒体，因此有很多新的特性包括一系列的标签、JavaScript API、CSS3等等。下面简单介绍其中的一些特性。

HTML5新增了一个canvas元素，它可以用来在网页上绘制图形，画布是一个矩形区域，您可以控制其每一像素。canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法[3]。如使用下面一行代码就可以添加一个简单的canvas元素到网页当中去：

<canvas id="myCanvas" width="200" height="100"></canvas>

但是canvas本身并没有绘制图形的能力，其中的操作还是要开发人员通过JavaScript脚本来完成。

HTML5还提供了对内联SVG（可伸缩矢量图形）的支持，它可以用于定义用于网络的基于矢量的图形，它用抽象的视角看待图形，记录其中展示的模式而不是各个点的原始数据。它将图片看成各个“对象”的组合，用曲线记录对象的轮廓，用某种颜色的模式描述对象内部的图案（如用梯度描述渐变色）。比如一张留影，被看成各个人物和背景中各种景物的组合。这种更高级的视角，正是人类看世界时在意识里的反映。图像在放大或改变尺寸的情况下其图形质量不会有损失。矢量图中简单的几何图形，只需要几个特征数值，就可以确定。比如三角形，只需要确定三个顶点的坐标。圆只需要确定圆心的坐标和半径。描述它的函数已知的曲线也只需要几个参数就能够确定。如正弦曲线、各种螺线等等。如果用位图记录这些几何图案，则需要包含组成线条的各个像素的数据。

虽然**Canvas 和 SVG 都允许您在浏览器中创建图形，但是它们在根本上是不同的[4]。Canvas通过 JavaScript 来绘制 2D 图形，是逐像素进行渲染的，它依赖分辨率,不支持事件处理器,其文本渲染能力也相对较弱，最适合图像密集型的游戏，其中的许多对象会被频繁重绘。而SVG是一种使用 XML 描述 2D 图形的语言，它不依赖分辨率，支持事件处理器，最适合带有大型渲染区域的应用程序，不适合游戏应用。**

在HTML4时期，网站视频的播放都是通过插件来进行的，而HTML5增加了vedio元素和audio元素，这代表着以后HTML将会逐步取代多媒体框架，例如flash。一些主流的大公司都逐步转向使用HTML5，谷歌于2015年2月26日开始自动将Flash广告转换为HTML5格式。

HTML5提供了<audio>标签，你不需要再按照第三方插件来渲染音频，大多数现代浏览器提供了对于HTML5 Audio的支持，不过目前仍旧需要提供一些兼容处理，如

<audio autoplay=”autoplay” controls=”controls”>

<source src=”file.ogg” />

<source src=”file.mp3″ />

<a href=”file.mp3″>Download this file.</a>

</audio>

浏览器将使用第一个可识别的格式。

和Audio很像，<video>标签提供了对于video的支持，由于HTML5文档并没有给video指定一个特定的编码，所以需要浏览器去决定要支持哪些编码。当前，video 元素支持三种视频格式：

Ogg = 带有 Theora 视频编码和 Vorbis 音频编码的 Ogg 文件

MPEG4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件

WebM = 带有 VP8 视频编码和 Vorbis 音频编码的 WebM 文件

在Html4的时代，浏览器存储网站个性化的数据，例如用户浏览器的痕迹，用户的相关数据等一般都只能依靠Cookie，但是Cookie存在如下限制：大多数浏览器支持最大为 4096 字节的 Cookie、大多数浏览器只允许每个站点存储 20 个 Cookie、有些浏览器还会对它们将接受的来自所有站点的 Cookie 总数作出绝对限制。因此想通过Cookie存储复杂的、关系型的用户数据基本是不可能的。

在Html5中增加了一个JavaScript对象：sessionStorage，通过此对象可以直接操作存储在浏览器中的会话级别的WebStorage。存储在sessionStorage中的数据首先是Key-Value形式的，当会话结束后，数据会自动清除。HTML5 还提供了一种在客户端没有时间限制的数据存储的新方法localStorage，这些数据不会随着Http请求发送到后台服务器，而且在HTML5的标准中要求浏览器对存储数据的大小至少要支持到4MB。

虽然Html5已经提供了功能强大的localStorage和sessionStorage，但是他们两个都只能提供存储简单数据结构的数据，对于复杂的Web应用的数据它们依然无能为力。因此Html5提供了一个浏览器端的数据库支持，允许我们直接通JS的API在浏览器端创建一个本地的数据库，而且支持标准的SQL的CRUD操作，让离线的Web应用更加方便的存储结构化的数据。

HTML5支持[XMLHttpRequest](http://baike.baidu.com/view/1105115.htm" \t "_blank)2和WebSocket等技术，并且通过这些技术解决了以前的跨域等问题，为我们的Web应用和网站在多样化的环境中提供了更快速的工作效率。XMLHttpRequest是一个浏览器接口，使得JavaScript可以进行HTTP通信。最早是由微软在IE 5中引进了这个接口。但是，这个接口一直都没有被标准化，在HTML5的概念形成后，W3C开始考虑标准化这个接口，并于2008年2月提出了XMLHttpRequest2的草案。新版本的XMLHttpRequest对象，针对老版本的缺点，做出了大幅改进。它具备如下优点：

可以设置HTTP请求的时限。

可以使用formdata对象管理表单数据。

可以上传文件。

可以请求不同域名下的数据（跨域请求）。

可以获取服务器端的二进制数据。

可以获得数据传输的进度信息。

2.安全性问题

但在革新的同时，HTML5也无可避免的带来了一些新的安全问题，早在2010年的黑帽大会上，Kuppan就公布了一系列通过HTML5技术进行的攻击方法。下面，就针对HTML5其中一些安全性问题和攻击方法展开简单讨论。

（1）XSS攻击

XSS即跨站脚本攻击，是一种被动的攻击手法，它主要通过在web页面里插入恶意代码，当用户浏览该页时，插入的代码会被执行从而达到攻击的目的。例如某个网页的输入框代码为<input type=”text” value=””>,而当我们在输入框中输入”onclick=“alert(‘xss’)时，上行代码就会变成<input type=”text” value=””onclick=”alert(‘xss’)”>,此时当用户点击该输入框时就会触发弹窗效果。那么XSS有什么危害呢？比如可以获取当前会话的cookie，控制用户的一些行为等等。

上面列举了一个最简单的XSS的例子，但在通常的一些网站中开发者都会加入自己或他人写的XSS过滤器以防止XSS攻击，但是HTML5添加了一系列的新特性，例如用于绘画的canvas元素，用于媒体回放的video和audio元素，对本地离线存储你更好的支持等等。这些新标签在提供了方便的新功能的同时也为XSS攻击绕过现有的过滤器提供了新的载体。这些标签提供了许多新的属性，包括onerror,formation等等，这些属性都可以被利用来执行JavaScript的代码，从而为XSS攻击提供了便利。

HTML5提供的两种客户端存储数据的新方法，localStorage和sessionStorage,用以弥补使用cookie进行大量数据存储时速度慢且效率也不高的缺点。但是本地存储在方便开发者的同时也带来了安全风险的问题，例如它也会遭受XSS的侵扰，从而引起信息泄露等问题。而且，由于HTML5的本地存储不支持HTTPonly，使得这一用来阻止通过JavaScript窃取cookie的方法失效，进一步降低了本地存储的防侵入能力。由于XSS不仅能够获取当前cookie，也有可能获取到本地存储中的其他信息，所以当本地存储中有隐私信息或敏感信息时，这种情况所造成的危害会被进一步扩大。

因此就目前来说，由于localStorage存储对于XSS并没有什么有效的抵御机制，尽量还是使用cookie来存储会话信息，并用HTTPonly来对信息进行保护。而且在使用localStorage存储信息的时候尽量不要将敏感信息或私密信息存储在其中，并且在对存储的信息进行操作时不论是输入还是输出都应该进行严格过滤以防XSS的产生。

（2）跨域请求伪造攻击

CSRF（跨站请求伪造攻击），也被称为“One Click Attack”或者Session Riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用[3]。尽管听起来像跨站脚本（[XSS](http://baike.baidu.com/view/50325.htm" \t "_blank)），但它与XSS非常不同，并且攻击方式几乎相左。XSS利用站点内的信任用户，而CSRF则通过伪装来自受信任用户的请求来利用受信任的网站。与[XSS](http://baike.baidu.com/view/50325.htm" \t "_blank)攻击相比，CSRF攻击往往不大流行（因此对其进行防范的资源也相当稀少）和难以防范，所以被认为比[XSS](http://baike.baidu.com/view/50325.htm" \t "_blank)更具危险性。

攻击通过在授权用户访问的页面中包含链接或者脚本的方式工作。例如：一个网站用户A可能正在浏览聊天论坛，而同时另一个用户B也在此论坛中，并且后者刚刚发布了一个具有A银行链接的图片消息。B编写了一个在A的银行站点上进行取款的表提交的链接，并将此链接作为图片地址。如果A的银行在cookie中保存他的授权信息，并且此cookie没有过期，那么当A的浏览器尝试装载图片时将提交这个取款表和他的cookie，这样在没经A同意的情况下便授权了这次事务。

目前很多互联网站点都在使用令牌（token）来防御CSRF攻击，但是随着HTML5技术的普及，攻击者很有可能窃取到CSRF的令牌，如通过CSRF token的请求URL(GET请求)，利用CORS协议，攻击者可以注入一个CSRF payload跨域请求到目标站点上。不过要进行利用的话需要服务端添加一个HTTP 头字段“origin”，HTTP头中的“origin”控制着跨域资源共享是否使用，并且需要设置该属性withCredentials为true，攻击者可以使用POST方法，通过XHR建立一个隐蔽的连接，就能够实现一次CSRF攻击。

（3）HTML5一些API和socket的滥用

WebSocket是HTML5的又一大新特性，在如今这个web应用越来越普遍的时代，消息推送也变的越来越重要。在以前，这一过程基本上都是基于http的单向通信，使用轮询，反向Ajax等技术来向服务器获取数据，既不方便逻辑也较为复杂。而为了解决这个问题，HTML5推出了WebSocket，这是一种新型的全双工通信机制，实现了服务器与客户端之间高效的双向通信。这一机制使得真正意义上的消息推送得以实现。

但是虽然HTML5限制了WebSocket可以使用的端口，但它仍然会成为实现攻击的载体。就在2010年，Kuppan发布了一个概念验证工具JS-Recon，它可以通过WebSocket对内网进行扫描。你可以想象一下，你打开了一个网页，而这个网页通过WebSocket对你的端口进行扫描，而后发现了你内部网络上一个开启的端口，它就可以通过这个端口以你的浏览器为媒介来建立连接，从而绕过防火墙达到入侵的目的。

HTML5还新增了一个地理定位的功能，其Geolocation API 用于获得用户的地理位置[4]。该特性可能侵犯用户的隐私，这些信息的泄露可能造成许多社会问题。虽然HTML5中规定使用该特性是要经过用户同意，但依旧可以通过一些手段来获取受害者的地理位置，比如诈骗等，因而用户应当谨慎操作，不能轻易授权给网站以查看自己地理位置的权限。

（4）僵尸网络

僵尸网络 Botnet 是指采用一种或多种传播手段，将大量主机感染bot程序（[僵尸程序](http://baike.baidu.com/view/2702065.htm" \t "http://baike.baidu.com/_blank)）病毒，从而在控制者和被感染[主机](http://baike.baidu.com/view/23880.htm" \t "http://baike.baidu.com/_blank)之间所形成的一个可一对多控制的网络[5]。僵尸网络在互联网上已存在多年，对网络安全也产生了不可忽视的影响。而随着HTML5标准的完成和推广，基于支持HTML5的web前端浏览器的僵尸网络也随之而产生。

对于其技术基础的支撑首先就是HTML5中的Web Workers特性，这是一种多线程机制，通过使用它可以允许后台运行多个JavaScript线程，具有三大特征：能够长时间运行并持续响应、理想的启动性能，以及理想的内存消耗[6]，其最初的目的是为了使浏览器可以在不影响用户操作的同时处理其他事务，但这也恰恰为恶意的JavaScript脚本提供了异步环境。而后是HTML5另一新增特性——CORS，即跨域资源共享机制，HTML5针对同源策略放宽了一些限制，允许来自同一站点的页面访问其页面的方法和属性等，

因此浏览器是可以发起跨域请求的，这样你就可以外链一个外域的图片或者脚本。COR是页面层次的控制模式，在这一模式中，访问控制的职责放到了页面开发者的手中，而不是服务器管理员。而前端僵尸网络立足的关键就在于HTML5支持CORS机制之后并未禁止跨域发送请求。而且HTML5支持XMLHttpRequest，这一特性又为攻击者提供了便利。

由于XMLHttpRequest请求可以发送到跨域服务器，跨域服务器也会对之前的XMLHttpRequest做出响应，并且跨域服务器对XMLHttpRequest的响应和正常访问这个服务器产生的响应没有任何区别。因此，对于服务器来说，假设有10000个XMLHttpRequest同时对某资源发出请求，以上的请求与10000个用户同时用浏览器正常访问这个资源是等价的，因为都是做同样的处理，都是返回同样的内容。这也就是说，使用Web Worker和XMLHttpRequest进行DDOS攻击是可行的。

基于Web Worker的僵尸网络用途很多，比如用以生成Bitcoin,一种目前非常火热且价值较高的虚拟货币，再比如可以利用其来进行暴力密码破解，利用其多线程的特性，使用基于浏览器的僵尸网络来进行破解，当僵尸程序达到一定数量级的时候其破解效率就可以和专业级的破解程序相当。其攻击方式也非常丰富，比如分布式拒绝服务攻击，即使目标服务器不支持跨域共享协议，但它仍然要处理该请求，从而达到了分布式拒绝服务攻击的效果。

（5）点击劫持

点击劫持，也被称为UI-覆盖攻击。这个词首次出现在2008年，是由[互联网](http://baike.baidu.com/view/6825.htm" \t "_blank)安全专家罗伯特·汉森和耶利米·格劳斯曼提出的[7]。它是通过覆盖不可见的框架误导受害者点击。虽然受害者点击的是他所看到的网页，但其实他所点击的是被黑客精心构建的另一个置于原网页上面的透明页面。这种攻击利用了HTML中<iframe>标签的透明属性。

在以前，一般网站都是通过frame busting这种方法来防止点击劫持，即通过写一段JavaScript代码，以禁止iframe的嵌套。在HTML5中，由于引入了新标签和新层叠样式表（CSS3），使得点击劫持变的更加容易[8]。而且由于HTML 5将针对iframe元素增加sandbox属性，其本来目的是为了防止不信任的[Web页面](https://www.baidu.com/s?wd=Web%E9%A1%B5%E9%9D%A2&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYmvP-rjnLm1fknAwWrH6s0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHRLPWbkPjR1PWTvP1R1rHT3n0" \t "_blank)执行某些操作。但由于其可以限制iframe页面中的JavaScript脚本执行，攻击者就可以利用沙箱的这一特性使得frame busting失效。

3.总结

虽然HTML5无可避免的存在一定的安全风险，但随着HTML5规范的完成，其在未来必将成为发展的主流。截至2015，有80%的App将全部或部分基于HTML5， HTML5将会带来一个统一的网络，无论是笔记本，台式机，还是智能手机都能够很方便的浏览基于HTML5的网站。

而作为一个开发者，我们也应当与时俱进，在学习使用HTML5新规范的同时也要考虑到程序的安全性问题。因此服务提供商应该有针对性的对使用HTML5的程序加强程序安全防护，并在程序发布之前尽可能进行完善的安全性测试。根据上文的一些攻击方式，我们可以就以下几个方面对HTML5进行安全性的加强。

首先，我们应当加强对本地存储的访问的控制。慎用Web Storage来进行信息存储，对于重要信息依旧使用cookie存储来替代本地存储。而且，为了防止明文存储方式存储的字符串被后台读取，应当配合用户证书来对这些数据进行加密。其次，加强对输入输出内容的验证。输入输出这一过程是危险频发的时段，如不注意就可能引发SQL注入、XSS攻击等危险。因此，作为一个开发者要时刻注意严格控制用户的输入输出过程。另外，在使用HTML5的CORS技术来实现自由的跨域访问时，应当注意目标服务器是否是值得信任的。而且，因为HTML5中引入了很多新的标签属性，开发者应该更新自己的XSS黑名单过滤器，以阻止XSS攻击。

参考文献

[1]www.w3school.com.cn/html5 canvas

[2]www.w3school.com.cn/html5 画布vs SVG

[3]http://baike.baidu.com CSRF

[4]www.w3school.com.cn/html5中HTML5地理位置

[5]http://baike.baidu.com 僵尸网络概念

[6]张剑，陈剑锋，王强，《HTML5新特性及其安全性研究》

[7]http://baike.baidu.com 点击劫持

[8]孙松柏，Ali Abbasi,诸葛建伟等《HTML5安全研究》