研究与开发

基于网络爬虫与页面代码行为的 XSS 漏洞动态检测方法

柳毅，洪俊斌
（广东工业大学计算机学院，广东 广州 510006）

摘　要：XSS 漏洞是攻击 Web 应用程序、获取用户隐私数据的常见漏洞。传统的 XSS 漏洞检测工具并没有对 AJAX Web 应用程序进行针对性的检测，在检测精度方面与实际情况存在巨大差距。针对这种情况，对 AJAX 技术下 XSS 漏洞的特点进行了分析，提出了一种基于网络爬虫与页面代码行为的动态检测方法。实验结果表明，提出的方法在节省人力、时间成本与漏洞检测方面有较好的表现。

关键词：XSS 漏洞；网络爬虫；漏洞检测；AJAX Web 应用
中图分类号：TP393.08　　**文献标识码**：A
doi: 10.11959/j.issn.1000-0801.2016068

A dynamic detection method based on Web crawler and page code behavior for XSS vulnerability

LIU Yi, HONG Junbin
Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China

Abstract: XSS vulnerability is a common vulnerability of attacking the Web application and getting the user's privacy data. Traditional XSS vulnerability detection's softwares aren't specially detecting for AJAX Web application. There is a huge disparity in the inspection accuracy. According to this situation, the XSS vulnerability characteristics of AJAX Web applications were described in detail, and a dynamic detection method based on Web crawler and page code behavior was proposed. Experimental results show that the proposed method has good performance in labor-saving, time saving and vulnerability detection effect.

Key words: XSS vulnerabilitiy, Web crawler, vulnerabilitiy detecting, AJAX Web application

1　引言

随着 Web 2.0 时代的到来，Web 程序告别了沉重的请求—返回的模式，采用了更轻便的局部刷新模式，提高了用户体验。其中，AJAX（asynchronous JavaScript and XML，异步 JavaScript 和 XML）[1]技术在 Web 2.0 中占据了主导地位。然而 AJAX 技术将一部分逻辑处理从服务器端转移到客户端，暴露了更多的接口[2]，增加了许多针对 Web 应用程序的安全威胁，其中严重程度最高的是跨站脚本（cross-site script，又称 XSS）攻击[3]。从国际开源安全组织[4]

收稿日期：2015-11-09；**修回日期**：2016-01-26
基金项目：国家自然科学基金资助项目(No.61572144)；广东省自然科学基金资助项目(No.2014A030313517)；广东省科技计划基金资助项目(No.2014A010103029, No.2013B040500009)；广州市科技计划基金资助项目(No.201508010026, No.2014J4100201)
Foundation Items: The National Natural Science Foundation of China(No.61572144), The Natural Science Foundation of Guangdong(No.2014A030313517), The Science and Technology Planning Project of Guangdong Province (No.2014A010103029, No.2013B040500009), The Science and Technology Planning Project of Guangzhou(No.201508010026, No.2014J4100201)

(Open Web Application Security Project,OWASP)公布的10种最严重的Web应用程序安全漏洞排行榜可知，跨站脚本攻击一直处于前三的位置。Web 2.0时代后，跨站脚本攻击与AJAX技术相结合，出现了新的特点，可在用户毫不知情的情况下进行攻击[5]，威胁性更大。就目前而言，采取较多的行为都是被动的防范措施，比如防火墙[6]等。这样做明显忽略了应用程序级别的安全问题，使之在高层面缺乏有效的防范方式。对于Web应用程序级别的安全，虽然有很多学者进行了研究，但大多数还停留在Web 1.0时代，并没有对异步传输进行安全性检测。参考文献[7,8]虽然对AJAX有所提及，但并没有进行针对性的研究。所以针对大量使用AJAX技术的Web应用程序，设计相关的XSS漏洞检测工具是十分必要的。

本文提出了一种基于网络爬虫与页面代码行为的动态检测方法对AJAX Web应用程序进行XSS漏洞检测。针对用户提交的URL(uniform resource locator，统一资源定位符)[9]进行爬虫，提取数据输入点，完成错误数据注入，再通过模拟攻击对服务器进行请求，对返回的页面代码行为进行跟踪与分析，得出XSS漏洞威胁。实验证明该方法效率较高，漏报率较低。

2 XSS漏洞特征及检测工具分析

2.1 XSS漏洞特征分析

现有的XSS漏洞类型可分为3种基本类型：反射型XSS攻击(reflected/non-persistent XSS)[10]、存储型XSS攻击(stored/persistent XSS)[11]和基于DOM (document object model)的XSS攻击(DOM-based XSS)[12]。反射型XSS攻击方式是通过服务器的错误信息或检索结果等手段将注入代码“反射”回来。攻击者构造包含注入代码的恶意链接，并通过某种方式将此恶意链接发送给受害者，受害者点击访问后，注入脚本便会将服务器返回的数据传输到攻击者的服务器上[13]；存储型XSS与反射型XSS最大的不同在于，恶意脚本将被永久性地存放在目标服务器的数据库中，浏览者在访问时恶意脚本便会执行，使浏览者受到攻击[14]；在基于DOM的XSS这类攻击方式中，攻击者是通过以下过程执行JavaScript的：攻击者设计有注入JavaScript代码的恶意URL[15]，并发送给用户，用户进行点击访问，浏览器会将返回的HTML(hypertext markup language，超文本标记语言)代码解析成DOM树结构[16]，在这过程中，注入代码便会被解析出来，从而导致基于DOM的XSS攻击出现，攻击者就有可能获取受害者电脑的重要权限[17]。

2.2 XSS漏洞的检测方法及工具

高效率的爬虫算法能够在短时间内找到数据输入点，并且根据数据输入点的类型，从数据库中获取错误注入数据，组成有攻击性的get/post[18]请求注入系统中，如果产生与从数据库中获取的注入数据相对应的预期攻击行为，就可以断定Web应用程序中存在XSS漏洞。在错误数据进行注入的同时，要注意对程序中的合法过滤器进行规避[19]。目前有很多XSS漏洞扫描的工具，比如Paros[20]和X5S[21]，它们并没有对AJAX技术下产生的XSS漏洞进行深入分析。针对这种情况，提出了一种基于网络爬虫与页面代码行为的动态检测方法来实现对AJAX Web应用程序的XSS漏洞进行检测。

3 XSS漏洞定位检测方法

3.1 抓取数据输入点

抓取数据输入点的步骤如下：

(1)爬虫对Web页面的HTML代码进行扫描得到静态URL；

(2)对页面代码中的JavaScript脚本和JavaScript所添加的事件进行提取；

(3)将提取到的JavaScript代码传递给JavaScript引擎编译执行，获取动态URL；

(4)存储获取到的URL，采用散列表存储方式高效率地将重复URL去除。

散列表中使用了双重散列算法来解决冲突，其探测地址的方法如下：

$$i=URL.\text{GetHashCode}()\%\text{hashcapacity} \tag{1}$$

$$h(URL,i)=h_1(URL)+i\times h_2(URL) \tag{2}$$

其中，散列函数h_1和h_2的计算式如下：

$$h_1(URL)=URL.\text{GetHashCode}() \tag{3}$$

$$h_2(URL)=1+(((h_1(URL)\gg 5)+1)\ \%\ (\text{hashcapacity}-1)) \tag{4}$$

二度散列的$h(URL,i)$的值有可能会大于hashcapacity，所以对$h(URL,i)$进行模运算，最终计算的散列地址为$h(URL,i)\ \%$ hashcapacity。

爬虫系统分为下载与分析两个模块，使用多线程同时运行两个模块以提高爬虫效率，如果硬件条件高，可分别在各个模块中再次使用多线程进行下载与分析。图1是爬虫系统运行的流程。

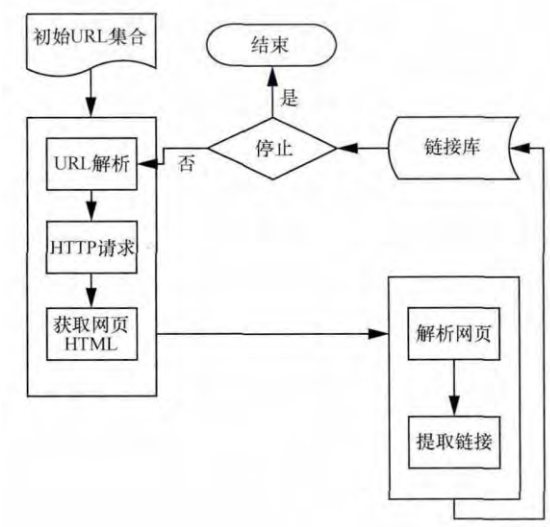

图 1　爬虫系统运行流程

3.2　错误数据注入

错误数据注入的步骤如下：

(1)从爬虫系统的散列表中，依次取出链接地址进行请求，并分析得到页面代码；

(2)提取代码中所有的表单元素；

(3)根据表单属性(get/post)、页面代码和输入区域的结构对象计算出散列值；

(4)根据散列值关联数据库中自定义注入代码进行拼接，对服务器发出请求。

图 2 是错误数据注入的流程。

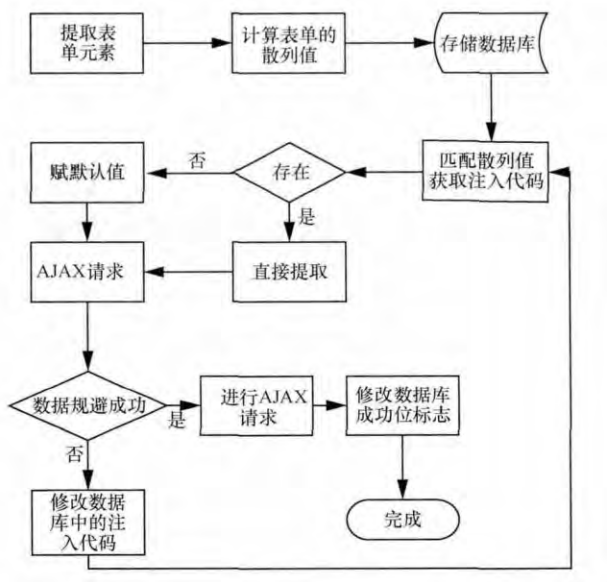

图 2　错误数据注入流程

攻击字符串是检测漏洞的关键部分，用数据库中的注入代码来构造有威胁的 URL、表单或 AJAX XMLHttpRequest 对象，向服务器发送请求。结合 JavaScript 的执行效果和语法，针对表单、URL 和 XMLHttpRequest 对象类型的攻击漏洞，表 1 列举了部分攻击字符串。

表 1　攻击字符串示例(部分)

Type	攻击字符串
常用的最基本的 XSS 攻击	<SCRIPT SRC =http://www.xxx.org/file/attack.js></SCRIPT>
16 进制编码注入 JavaScript	<Frame Source =&#x29&#x53&#x58&# x61&#x76&#x61&#x53&#x27&#x6A >
使用两个尖括号	<<SCRIPT>Document.Write();//<</SCRIPT>
引入 style 属性进行注入	<IMG STYLE="alert('XSS')">
…	…

3.3　结果分析

3.3.1　基于 AJAX 页面代码行为的 XSS 漏洞检测算法

(1)增量地构建 AJAX 返回的 DOM 树状态；

(2)记录返回数据后的页面代码行为；

(3)检测返回的代码行为是否出现相对应的恶意行为，判断是否存在 XSS 漏洞。

伪代码设计如下：

```
For each pagestate in StateQueue.Dequeue()(存储队列中的状态出列)
    BrowserControl.GetState (pagestate)(获取页面的HTML行为)
    BooleanhasXSS←DetectXSS(pagestate)
    If hasXSS then
        Write To DataBase(URL)
    End If
    BrowserControl.GetState(pagestate)
End For each
```

3.3.2　基于 JavaScript 的 XMLHttpRequest 检测

在 JavaScript 代码中，最关键的是 XHR(XMLHttpRequest，可扩展超文本传输请求)对象。该对象可以异步地向服务器发出请求，获取新数据，然后通过 DOM 将数据插入页面中进行局部刷新。这给攻击者留下了一个很大的漏洞，攻击者可使用 XHR 对象，注入错误代码，因此必须对 AJAX XHR 的漏洞检测进行特殊处理。

(1)收集 JavaScript 的文件信息，比如大小、文件名等，构建 JavaScript 文件特征库。

(2)检测时将页面获取的 JavaScript 文件与文件特征库

中的文件进行匹配判断是否存在漏洞。

伪代码设计如下：

```
If PageHTML.HasXHR( ) then
    JSFileNames←BrowserControl.GetFileInfo(PageHTML)
For each jsFileName in JSFileNames
    BooleanIsAjaxInDB←DataBase.Exists(jsFileName)
If IsAjaxInDB continue
Else
    JavaScriptFile ← BrowserControl.DownloadFile (jsFile Name)
    IsAjaxInDB←DataBase.HashCode.IsMatch (Java Script File.hashcode)
    If IsAjaxInDB continue
    Else
    Return HasXSS
    ...
End If
```

4 实验结果及分析

为验证本文提出方法的可行性，采用 Visual Studio 2015 对本文提出的方法进行 C# 编程实现 XSS Finder 工具，硬件方面主要采用主流的 IBM-PC 兼容机，数据库为 SQLServer 2008 R2。下面从时间复杂度和检测精度方面来评估 XSS Finder 工具。

4.1 时间复杂度

为对比 XSS Finder 在时间复杂度上的优势，选用了常用的 XSS 漏洞检测工具 Paros 和 X5S 进行对比。采取方法是对每个 Web 应用程序独立扫描 20 次，算出平均扫描时间。图 3 是 3 个软件进行扫描的时间对比。

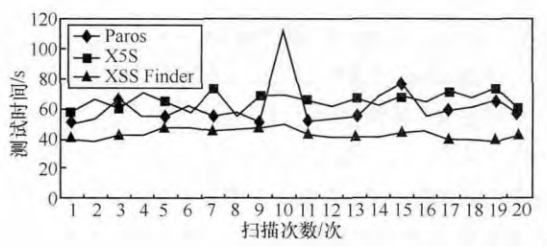

图 3 爬虫时间对比

由图 3 可知，XSS Finder 工具所消耗的时间是较低的。

4.2 检测精度

对比 XSS Finder 与两个软件的检测精度，采取的方法是选择两个网站，一个是某学校的官方网站，另一个是开源系统，名字是 PetStore，此系统是 Java 编写的基于 AJAX 的 Web 应用程序，3 个软件分别对网站进行扫描与检测，并对比最后的结果，具体见表 2。

表 2 工具查询结果对比

网站	工具	数据入口点	XSS 漏洞数	系统漏洞数
某学校官网	Paros	1 forms、8 inputs	9	15
	X5S	1 forms、5 inputs	7	15
	XSS Finder	4 forms、13 inputs	15	15
PetStore	Paros	12 forms、25 inputs	4	8
	X5S	10 forms、22 inputs	3	8
	XSS Finder	17 forms、32 inputs	8	8

从实验结果来看，XSS Finder 检测到的漏洞数和系统存在的漏洞数量是一致的，检测到的数据入口点也符合实际情况，表明 XSS Finder 检测漏洞的准确性更好。

5 结束语

通过对 AJAX 技术下 XSS 漏洞的出现原因及检测技术的研究，本文提出一种基于网络爬虫与页面代码行为的 XSS 漏洞检测方法，与两种常用的 XSS 漏洞检测工具比较，在时间复杂度方面，对 Web 应用程序独立扫描 20 次的情况下，扫描时间更短，高效地解决了 URL 重复性的问题；在检测精度方面，提出了对页面代码行为和 XHR 对象进行检测的方法，具有更高的漏洞检测准确性。

参考文献：

[1] DAHSE J. A vulnerability scanner for different kinds of vulnerabilities [DB/OL]. [2015-04-09]. http://rips-scanner.sourceforge.net, accessed.

[2] AN H Y, SONG Y, YU T, et al. A new architecture of AJAX Web application security crawler with finite-state machine [J]. IEEE Computer Society, 2014(27): 112-117.

[3] OWASP. Cross site scripting prevention cheat sheet [EB/OL]. (2013-12-26) [2014-03-26]. https: //www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet.

[4] OWASP. Top ten project [EB/OL]. (2013-12-03) [2013-12-10]. https: //www. Cwasp.org/in dex.php/ Category:OWASP-Top-Ten-Project.

[5] LI Z, XU X, LIAO L J, et al. Using templates combination to generate testing vectors dynamically indetecting Web applications vulnerabilities[J]. Application Research of Computers, 2015, 32(10): 3004-3009.

[6] CHEN J F, WANG Y D, ZHANG Y Q, et al. Automatic generation of attack vectors for stored-XSS [J]. Journal of Graduate University of Chinese Academy of Sciences, 2012, 29(6): 815-820.

[7] WANG X L, ZHANG Y Q. A behavior-based client defense scheme against XSS [J]. Journal of Graduate University of Chinese Academy of Sciences, 2011, 25(5): 668-675.

[8] CHEN J Q, ZHANG Y Q. Design and realization of Web cross-site scripting vulnerability detection tool [J]. Computer Engineering, 2010, 36(6): 152-158.

[9] JIANG H, XU Z Y, WANG X. XSS attack defense method based on behavior [J]. Computer Engineering and Design, 2014, 35(6): 1911-1925.

[10] GUO X B, JIN S Y, ZHANG Y X. XSS vulnerability detection using optimized attack vector repertory [J]. IEEE Computer Society, 2015(50): 29-36.

[11] CUI B J, LONG B L, HOU T T. Reverse analysis method of static XSS defect detection technique based on database query language [C]//The Nineth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), November 8-10, 2014, Guangzhou, Guangdong, China. New Jersey: IEEE Press, 2014: 487-491.

[12] LIU W X, YU S Z. Research for ACK attacks in network coding[J]. Journal of Chinese Computer Systems, 2012, 32(7): 1354-1359.

[13] Rsnake. XSS(cross site scripting) cheat sheet [EB/OL]. [2013-11-15]. http: //ha. Ckers. org /xss.html.

[14] WU H Q. White hatter talks about web security [M]. Beijing: Publishing House of Electronics Industry, 2013: 152-178.

[15] GUPTA M K, GOVIL M C, SINGH G. Predicting cross-site scripting(XSS) security vulnerabilities in Web applications[C]// 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), July 22-24, 2015, Songkhla, Thailand. New Jersey: IEEE Press, 2015: 162-167.

[16] LI Y W, LIU Z X, DING S J. Technique for discovering stored XSS vulnerability based on tracing risky data [J]. Computer Science, 2014, 41(11A): 241-244.

[17] QIU Y H. The analysis and defense of XSS attack [M]. Beijing: Posts&Telecom Press, 2013.

[18] LI Z J, ZHANG J X, LIAO X K. Survey of software vulnerability detection techniques [J]. Chinese Journal of Computers, 2015, 38(4): 717-732.

[19] HALFOND W G J, ORSO A, MANOLIOS P. WASP: protecting web applications using positive tainting and syntax-aware evaluation[J]. IEEE Transactions on Software Engineering, 2008, 34(1): 65-81.

[20] SAYED B, TRAORE I. Protection against Web 2.0 client-side web attacks using information flow control [J]. The 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), May 13-16, 2014, Victoria, BC, USA. New Jersey: IEEE Press, 2014: 261-268.

[21] HELEN K, SARANDIS M, CHRISTOS D. An advanced web attack detection and prevention tool[J]. Information Management & Computer Security, 2011, 19(5): 280-299.

[作者简介]

柳毅(1976–),**男**,**博士**,**广东工业大学计算机学院副教授**,**主要研究方向为网络与信息安全**。

洪俊斌(1990–),**男**,**广东工业大学计算机学院硕士生**,**主要研究方向为网络与信息安全**。