Computer Science and Application 计算机科学与应用, 2016, 6(6), 340-346
Published Online June 2016 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2016.66042

Hans汉斯

A Web Application Vulnerability Detection Method Based on Web Crawler Technology

Quanmin Wang, Jiawei Lei, Cheng Zhang, Xiaotong Zhao

College of Computer Science, Beijing University of Technology, Beijing
Email: lei_jiawei@126.com

Received: Jun. 1st, 2016; accepted: Jun. 19th, 2016; published: Jun. 23rd, 2016

Copyright © 2016 by authors and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

With the continuous development of Web applications, a variety of security vulnerabilities, including XSS, also generate more and more. Today, the defects of the traditional XSS defense technology have been more and more appear, such as a single type of defense, defense strength low, defense means backward. There is an urgent need to continuously improve and perfect the methods and means of defense. Aiming at this problem, this paper proposes a Web application vulnerability detection method based on Scrapy. Through the framework to provide convenient conditions to the page for extraction and analysis, specific attack vector is generated according to the different ways of attacks. Finally, we make the combination of page injection points and attack vector to achieve the objective to test whether it is vulnerable. Experimental results show that this vulnerability detection method has a great improvement in the efficiency of crawling pages and vulnerability detection.

Keywords

XSS, Web Application, Scrapy, Attack Vectors

基于爬虫技术的Web应用程序漏洞检测方法

王全民，雷佳伟，张　程，赵小桐

北京工业大学计算机学院，北京
Email: lei_jiawei@126.com

收稿日期：2016年6月1日；录用日期：2016年6月19日；发布日期：2016年6月23日

文章引用: 王全民, 雷佳伟, 张程, 赵小桐. 基于爬虫技术的Web应用程序漏洞检测方法[J]. 计算机科学与应用, 2016, 6(6): 340-346. http://dx.doi.org/10.12677/csa.2016.66042

摘　要

随着Web应用不断的发展，随之而产生的包括XSS在内的各种安全漏洞也越来越多。今天，XSS传统防御技术的缺陷已经越来越多地显现，例如防御种类单一、防御强度低、防御手段落后等，这就迫切需要不断提高和完善防御的方法和手段。针对此问题，提出了一种基于Scrapy的爬虫框架的Web应用程序漏洞检测方法。通过框架提供的便利条件对页面进行提取分析，根据不同的攻击方式生成特有的攻击向量，最后使页面注入点与攻击向量组合达到测试是否具有漏洞的目的。实验结果表明，这种漏洞检测方法在爬取页面以及漏洞检测的效率上都有了很大的提高。

关键词

XSS，Web应用，Scrapy爬虫，攻击向量

1. 引言

在因特网发展的早起，网站大部分都是静态的文档，人们可以通过浏览器读取这些文档。但是随着Web2.0 时代的到来，网站逐渐发展成为具有各种强大功能的应用程序。互联网用户可以通过各种网站了解到时事新闻，观看各种视频资源，在网上购物以及快捷支付，微博、论坛等娱乐活动。但是，由于越来越多的个人信息暴露在网上，随之而带来的安全问题也越来越多。黑客根据请求的 URL、Cookie、表单等发动攻击，从而达到某种目的。根据 OWASP TOP 10 报告，在 2013 年跨站脚本(XSS)仍然处于 10 项最严重的 Web 应用程序安全风险的第 3 位[1]，因此，它对计算机网络安全的威胁相当大[2]。

XSS 最早诞生于 1996 年，但是在 2000 年 2 月才由 US—CERT/CC 正式公布，国内关于此漏洞最早的资料也是在 2000 年[3]。

XSS 攻击作为 Web 业务的最大威胁之一，不仅危害 Web 业务本身，对访问 Web 业务的用户也会带来直接的影响，如何防范和阻止 XSS 攻击，保障 Web 站点的业务安全，是定位于业务威胁防御的入侵防御产品的本职工作[4] [5]。

XSS 漏洞和 SQL 注入漏洞一样，都是利用了 Web 页面编写不完善的弱点，所以每一个漏洞所利用和针对的弱点都不尽相同。这就给 XSS 漏洞防御带来了困难，不可能以单一特征来概括所有 XSS 攻击。今天，XSS 传统防御技术的缺陷已经越来越多地显现，例如防御种类单一、防御强度低、防御手段落后等，这就迫切需要不断提高和完善防御的方法和手段。

针对以上问题本文提出旨在改善爬取网站的效率以及对漏洞挖掘的误报率。实验结果表明，该方法可以提高对网站的分析度以及漏洞的挖掘。

2.相关概念和整体框架

2.1. 相关概念

跨站脚本攻击(英文全称为 Cross Site Script，为了区别于层叠样式表，简称 XSS)。

XSS 攻击，通常指黑客通过“HTML 注入”篡改了网页，插入了恶意的脚本，从而在用户浏览网页时，控制用户浏览器的一种攻击方式。在一开始，这种攻击的演示案例是跨域的，所以叫“跨站脚本”。但是发展到现在，由于 JvavScript 的强大功能以及网站前端应用的复杂化，是否跨域已经不再重要。但由于历史原因，XSS 这个名字一直保留了下来。

XSS 根据效果的不同可以分为如下几类。

(1) 反射型 XSS

反射型 XSS 指简单的把用户输入的数据“反射”给浏览器。也就是黑客需要诱使用户“点击”一个恶意的链接，才能攻击成功。反射型 XSS 也叫“非持久型 XSS”。

(2) 存储型 XSS

存储型 XSS 会把用户输入的数据“存储”在服务器端。这种 XSS 具有很强的稳定性。比较常见的场景是：黑客写下一篇包含有恶意 JavaScript 代码的博客文章，文章发表后，所有访问博客文章的用户，都会在他们的浏览器中执行这段恶意的 JavaScript 代码。黑客把恶意的脚本代码保存到服务器端，所以这种攻击方式叫做“存储型 XSS”，也叫“持久型 XSS”。

2.2. XSS 攻击手段与防御对策

XSS 攻击手段

(1) 窃取 Cookie 值

攻击者通过向网页中注入 JavaScript 代码获取到用户的 Cookie 值，从而达到某种目的。

(2) 通过 JavaScript 攻击

此类型攻击输入(1)中的进阶版，比较典型的案例是利用 XSS 制造蠕虫病毒，从而攻击网站，收集用户信息或者伪造他人发布信息，形成潜在的巨大风险。另外，随着 Ajax 技术的流行，通过 JavaScript 调用 Web 应用的各种功能程序(简称 API)在网站中的分量正在逐步增加，导致 XSS 与 JavaScript 的攻击实施起来反而变得更加容易[6]。

(3) 篡改网页

以上主要针对于需要注册会员才可登录的网站。其实，没有登录功能的网站更容易受到攻击。例如，黑客会通过一个伪造的链接诱使用户进入一个与正规网站及其相似的钓鱼网站，从而获取用户大量信息，达到各种目的。

XSS 的基础防护对策

(1) 在一般的 HTML 中，可以使用字符实体进行转义。

(2) 对于 JavaScript 或者 PHP 的脚本代码可以指定字符的编码方式。

(3) 对于用户需要输入的位置进行输入校验。

2.3. 整体框架

系统分为三个模块：网络爬虫模块、攻击向量模块、攻击检测模块。如图 1 所示。

(1) 通过 Scrapy 的爬虫框架爬取所有目标页面，并对所有爬取的页面进行分析，去重以及提取攻击注入点一遍测试。

(2) 通过对源攻击向量的挖掘策略的挖掘，对源攻击向量进行变形，产生攻击向量库。

(3) 接受(1) (2)传递的信息后，攻击与分析模块进行对目标 Web 应用进行跨站脚本攻击。每次攻击后，通过对返回信息的分析，判断出是否存在漏洞。

3. 改进的基于爬虫的 Web 应用程序漏洞检测方法

3.1. 基于 Scrapy 爬虫

在以往的针对 Web 应用程序安全检测中的爬虫模块大都是直接通过 Python 来实现开发的，最初的是单线程[7]，后来为了提高爬取速度，采用多线程的方式进行。但是随着技术的日新月异，有必要寻找

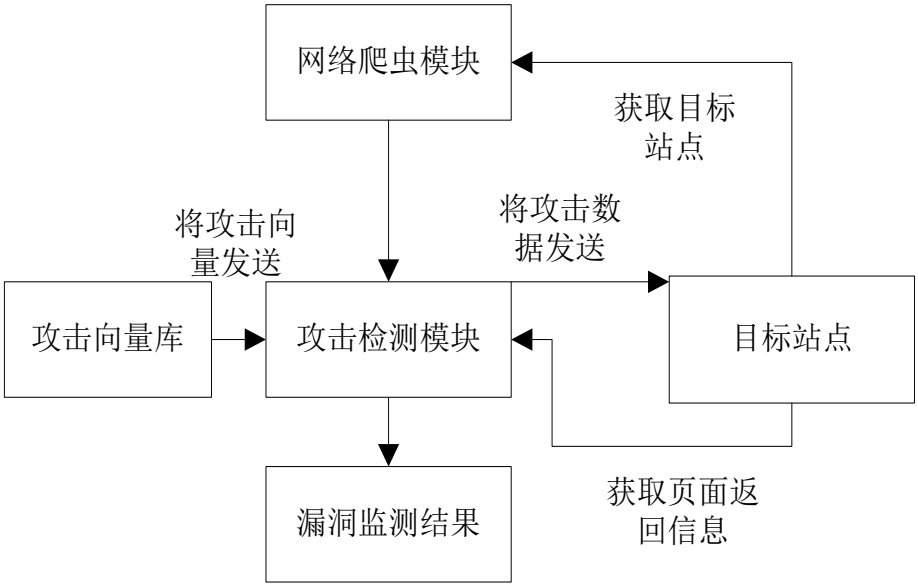

Figure 1. System block diagram
图 1. 系统模块图

出更高效更方便研究人员开发的技术[8]。

本文采用 Scrapy 的爬虫框架。Scrapy，是 Python 开发的一个快速、高层次的爬虫框架，用于抓取 web 站点并从页面中提取结构化的数据。Scrapy 用途广泛，可以用于数据挖掘、监测和自动化测试。Scrapy 吸引人的地方在于它是一个框架，任何人都可以根据需求方便进行相应的修改。它也提供了多种类型爬虫的基类，如 BaseSpider、sitemap 爬虫等[9] [10]。

传统爬虫利用的是静态下载方式，静态下载的优势是下载过程快，但是页面只是一个枯燥的 html，因此页面链接分析中获取的只是<a>标签的 href 属性或者自己分析 js、form 之类的标签捕获一些链接。在 Python 中可以利用 urllib2 模块或 requests 模块实现功能。动态爬虫在 web2.0 时代则有特殊的优势，由于网页会使用 javascript 处理，网页内容通过 Ajax 异步获取。所以，动态爬虫需要分析经过 javascript 处理和 ajax 获取内容后的页面。目前简单的解决方法是通过基于 webkit 的模块直接处理。

当通过不断爬取就会存在大量的链接，这时候就需要进行很关键的一步，URL 去重。由于运行中的爬虫主要阻塞在网络交互中，因此避免重复的网络交互至关重要。爬虫一般会将待抓取的 URL 放在一个队列中，从抓取后的网页中提取到新的 URL，在他们被放入队列之前，首先要确定这些新的 URL 没有被抓取过，如果之前已经抓取过了，就不再放入队列了。

利用 hash 表做去重操作是较容易的方法，因为 hash 表查询的时间复杂度是 O(1)，而且在 hash 表足够大的情况下，hash 冲突的概率就变得很小，因此 URL 是否重复的判断准确性就非常高。

如果 hash 表中，当每个节点储存的是一个 str 形式的具体 URL，是非常占用内存的，如果把这个 URL 进行压缩成一个 int 型变量，内存占用程度上便有了 3 倍以上的缩小[11]。因此可以利用 Python 的 hashlib 模块来进行 URL 压缩。

通过获取的 URL 对页面进行数据分析，包括 HTML 事件、HTML 标签以及 HTML 属性等注入点。

整个流程如图 2 所示。

3.2. 攻击向量生成库

传统的检测方法是对于每一个注入点，都依次从数据库取出一条记录来进行提交，直到数据库中的

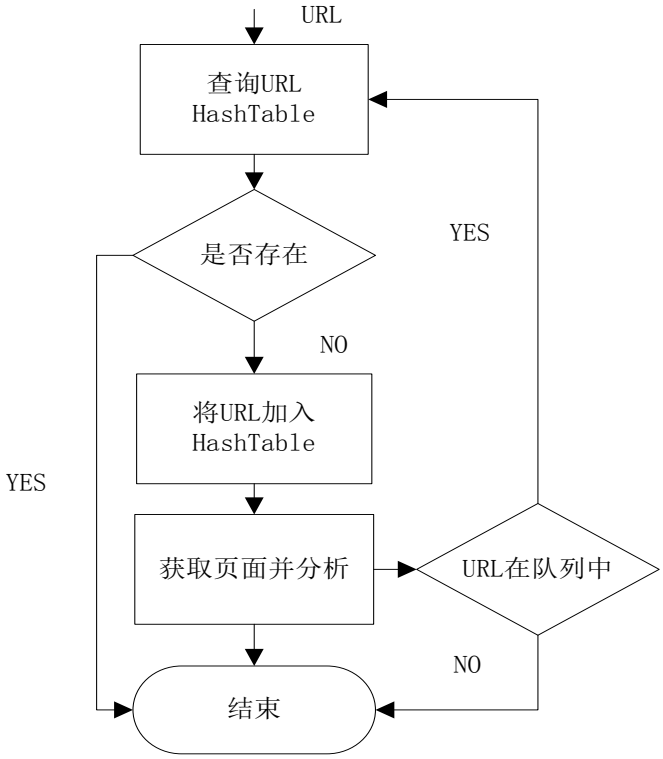

Figure 2. Reptile module flowchart
图 2. 爬虫模块流程图

记录被取完或是找到漏洞为止[12]。提交新数据时要考虑这个输入点采用的请求方式是 get 还是 post。最后在提交后的响应文本中查找是否出现了被提交的数据一模一样的字符串，若找到则说明有漏洞[13]。

传统方案的不足是使用了数据库。由于一条基本的攻击代码能转换为很多条攻击代码，因此这个数据库要记录的代码数目是巨大的，同时也是不可能记录完全的[14]。包括 CSS、HTML 及 JS 在内的编码方式有二进制、八进制、十六进制等等各种变形，这种情况对于完全依赖数据库是及其不合理的[15] [16]。

攻击代码生成库由一个函数实现，这个函数首先定义了关键词 alert(ID)，攻击代码的产生根据这个关键词被嵌入的位置可以分为三个分支：第一类是对 HTML 事件进行编写，如 onclick="alert('xss')、onload 当页面加载完成后触发以及 onmousemove 当鼠标移动就触发等；第二类是关键词被嵌入到 HTML 标签的各种变形中，大多数以提前闭合标签的形式存在，如</textarea><script>alert('xss')</script>；第三类是关键词被嵌入到 HTML 属性中的各种变形中，如 IMG 标签大小写不敏感<IMG SRC=JaVaScRiPt(‘XSS’)>、嵌入式标签，将 javascript 分开<IMG SRC=”jav ascript:alert(‘XSS’);”>以及用十六进制编码等。攻击代码生成库根据注入点的类型来调用相应的分支，并且加上适当的前缀，生成针对这个注入点的攻击代码。如图 3 所示。

3.3. 攻击检测模块

攻击检测模块采用传统的方式，通过爬虫分析出来的注入点与攻击向量模块生成的攻击向量组成数据包，然后向目标网站发送数据包，并获取相应的返回信息。 通过对返回的信息进行分析，判断出该注入点是否存在漏洞。

具体检测步骤如下：

(1) 判断哈希表中的 URL 是否为空。为真，则退出；为假，则进行(2)。

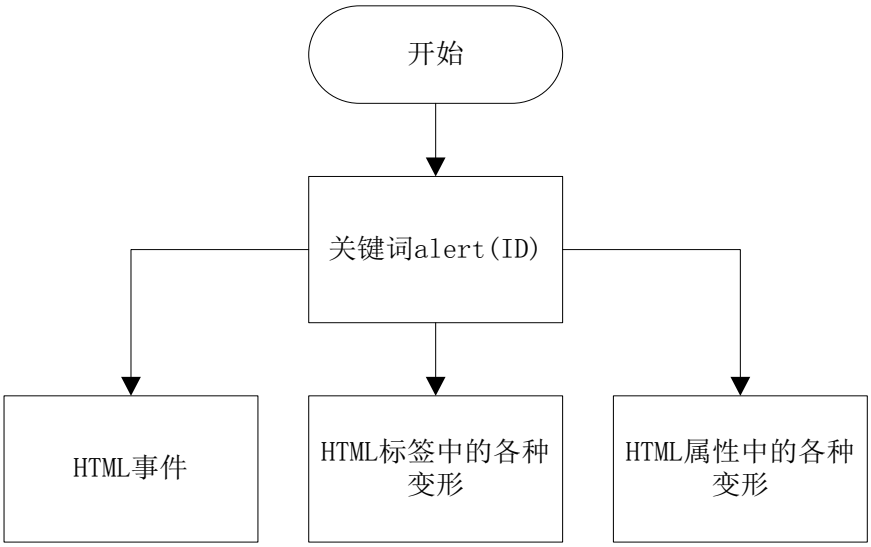

Figure 3. Attack vectors generation module
图 3. 攻击向量生成库

Table 1. XSS experimental results
表 1. XSS 实验结果

攻击方式	检测率	漏报率	误报率
HTML 事件	10/10	0/10	0/10
HTML 标签	9/10	1/10	1/10
HTML 属性	9/10	1/10	1/10

(2) 通过爬虫对页面分析出注入点。

(3) 注入点与攻击向量模块生成攻击所用的数据包，然后向目标网站进行发送，并获取相应的返回信息。

(4) 通过对返回信息的判断该注入点是否存在漏洞。为真，则转到步骤(5)；为假，则转到步骤(1)。

(5) 记录漏洞的相关信息，然后转到步骤(1)。

4. 实验结果分析

对一个网站分别采取静态和动态抓取页面进行对比。在静态抓取中，页面的长度是 63,838，页面内抓取的链接数量只有 166 个。而在动态抓取中，页面的长度增长到了 195,991，而链接数达到了 1422，有了近 10 倍的提升。并且实验发现，采用 Scrapy 进行爬取页面，在爬取时间以及内存占用上都得到了很大的提升。

本文使用真实的互联网数据来进行实验，根据 OWASP 提供的以及网上常见的 XSS 攻击方式进行实验。并且自建了一个存在 XSS 漏洞的 Web 站点，在实际中分别采用了针对 HMTL 事件、HTML 标签以及属性等几种方式进行攻击。实验结果如表 1 所示。

根据实验结果表明，本文的基于爬虫技术的 Web 应用程序漏洞检测方法基本可以检测出站点的各种类型的 XSS 漏洞，漏报率以及误报率较低。

5. 结束语

本文提出的基于爬虫技术的 Web 应用程序漏洞检测方法，旨在通过使用框架提高页面的爬取效率，

以及对页面分析能力的提升；通过对攻击向量制定相应的策略，从而提高了针对千变外化的脚本漏洞的灵活性，从而大大提高对漏洞检测的概率。

由于不断的发展，攻击者会不断的发现行的攻击方式与手段，所有在后续的工作中需要对攻击方式进行更细致的分析与完善，为以后能更有效的提早发现网站漏洞打下基础。

基金项目

国家自然科学基金项目(61272500)。

参考文献 (References)

[1] Wichers, D. (2013) The Top 10 Most Critical Web Application Secutity Rishk. OWASP.

[2] 吴耀斌, 王科, 龙岳红. 基于跨站脚本的网络漏洞攻击与防范[J]. 计算机系统应用, 2008(1): 38.

[3] 维基百科. 跨站点脚本[EB-OL]. http://zh.wikipedia.org.wiki/XSS

[4] 酷壳-CoolShell.cn. 新浪的 XSS 攻击[EB/OL]. http://coolshell.cn/articles/4914.html

[5] 陈嘉讯. 论跨站脚本攻击(XSS)的危害、成因及防范[J]. 网络与信息, 2008, 22(9):80-80.

[6] Schafer, J.B. Frankowski, D. Herlocker, J. and Sen. S. (2007) Collaborative Filtering Recommender Systems. In: *The Adaptive Web*, Volume 4321 of the Series Lecture Notes in Computer Science, 291-324. http://dx.doi.org/10.1007/978-3-540-72079-9_9

[7] 张宗之. 基于爬虫技术的 web 应用漏洞挖掘的研究[D]: [硕士学位论文]. 北京邮电大学, 2013.

[8] 肖征. 基于网络爬虫的网络漏洞扫描检测系统的设计与实现[D]: [硕士学位论文]. 长春: 吉林大学, 2014.

[9] Maedche, A. (2006) Ontology Learning for the Semantic Web. Kluwer Academic Publishers.

[10] Bradshaw. S. (2004) Reference Directed Indexing: Redeeming Relevance for Subject Search in Citation Indexes. In: *ECDL*, 499-510.

[11] 凌妍妍, 孟小峰, 刘伟. 基于属性相关的 Web 数据库大小估算方法[J]. 软件学报, 2008, 19(2): 224-236.

[12] Friedl, J.E.F. (2006) Mastering Regular Expressions. 3rd Edition, O'reilly Media Inc., 12(18): 4140-4143.

[13] Bates, D. (2010) Regular Expressions Considered Harmful in Client-Side XSS Filters. ACM.WWW 2010. USE:ACM, 91-100.

[14] Klein, A. Dom Based Cross Site Scripting or XSS of the Third Kind. http://www.Webappsec.org/projects/articles/071105.html

[15] 王津涛. HTML, CSS, Javascript 整合详解. 北京: 机械工业出版社出版, 2008.

[16] 风信子, 施威铭研究室. Javascript 最新网页制作. 北京: 人民邮电出版社, 2001.

再次投稿您将享受以下服务：

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击：http://www.hanspub.org/Submission.aspx