

基于渗透测试的 XSS 漏洞检测方法

霍国庆, 曹天杰

(中国矿业大学计算机科学与技术学院, 江苏省徐州市 221116)

摘要: Web 漏洞以对大多数网站产生严重威胁。其中跨站脚本 XSS (Cross-site Scripting) 漏洞是对用户及网站损害较重的漏洞之一。针对现有动态检测 XSS 漏洞方法效率上的不足, 提出一种改进的渗透测试检测方法, 通过构造网页预处理模块, 利用探针算法筛选掉不存在 XSS 漏洞的页面, 同时提取注入点以及注入点所在的最小对象, 根据注入点的最小对象分类生成相应的攻击向量并分开存储, 避免无效攻击向量的测试。另外, 在 URL 参数检测 XSS 的模块, 增加伪静态页面检测 XSS, 以减少漏报率。在此基础上, 利用爬虫程序设计实现了该检测系统, 实验结果表明, 所提的检测系统在提高检测网页 XSS 漏洞效率上很有效。

关键词: XSS 检测; 动态漏洞检测; 网络爬虫; web 安全。

中图分类号: TP393.08

XSS vulnerability detection based on penetration testing

HUO Guoqing, CAO Tianjie

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116)

Abstract: Web vulnerabilities have posed a serious threat to most websites. Cross-site Scripting (XSS) vulnerability is one of the most important vulnerabilities to users and websites. In order to overcome the shortcomings of the existing XSS vulnerability detection methods, this paper proposes an improved penetration testing method. By constructing a webpage preprocessing module, the XSS vulnerability filtering page is filtered out by using the probe algorithm. At the same time, the injection points and the injection points are extracted According to the minimum object classification of injection point, the corresponding attack vector is generated and stored separately to avoid the test of invalid attack vectors. In addition, the XSS module for detecting URL parameters adds pseudo-static page detection XSS to reduce the false negative rate. On this basis, the detection system is realized by using reptile program design. The experimental results show that the proposed detection system is effective in improving the efficiency of detecting XSS vulnerabilities.

Keywords: XSS detection; dynamic vulnerability detection; web crawler; web security

0 引言

随着 Web2.0 时代的到来, 基于 Web 的应用系统被广泛应用, 在给我们提供极大便利的同时, Web 安全也成为最常见的安全问题之一, 根据 2017OWASP[1] (Open Web Application Security Project, 开放式 Web 应用程序安全项目) 公布的十大安全漏洞列表, 对比与 2013 年的旧版, 我们可以看出, XSS 漏洞威胁依然高居前三的位置。XSS (Cross Site Scripting, 跨站脚本攻击) 原理[2]是攻击者通过向 Web 页面里插入恶意的脚本代码, 当用户浏览这些被插入恶意脚本代码的网页时, 那些对网页输入规避不足的 Web 页面就会执行嵌入其中的恶意脚本代码, 从而达到恶意攻击用户的目的, 跨站脚本攻击隐蔽性较强, 攻击成本较低, 扩散能力较强, 可以窃取用户 cookie, 劫持 session, 网络钓鱼, 窃取访问历史等, 危害较大, 故针对 XSS 漏洞防范研究是非常有必要的。

基金项目: 国家自然科学基金 (61303263)

作者简介: 霍国庆(1992-), 男, 硕士研究生, 主要研究方向: Web 漏洞挖掘与防御

通信联系人: 曹天杰(1967-), 男, 教授, 博导, 主要研究方向: 密码学, 信息与网络安全. E-mail: 8044601@qq.com

1 相关工作

1.1 XSS 分类

XSS 漏洞[3]主要分为三种类型：反射型 XSS，存储型 XSS 和基于 DOM 的 XSS。按照存储方式来分：持久型 XSS 和非持久型 XSS。

1) 反射型 XSS 漏洞

该类型又被称为非持久型 XSS，通过提交恶意代码到服务器，经过服务器解析，将恶意代码加入到响应页面中，当用户点击到该恶意代码块，触发 XSS 代码（服务器中没有这样的页面和内容），一般容易出现在搜索页面。

2) 存储型 XSS 漏洞

存储型 XSS 又称持久型 XSS 漏洞，这种类型的恶意脚本是存储于服务器中的，在信息发表页面中，通过提交恶意代码到服务器中，当服务器过滤不严时，恶意代码将会被存储到服务器中，当用户访问该页面时就会触发代码执行，容易造成蠕虫，盗窃 cookie 等危害。

3) DOM 型 XSS 漏洞

基于 DOM 的 XSS 漏洞是由于浏览器解析机制导致的漏洞，区别于前两种 XSS，他的实现是没有服务器端的参与，不需要通过服务器端的响应。

1.2 研究现状

目前针对 XSS 漏洞检测方式有三种：静态检测，动态检测以及动态检测加静态检测的混合检测方式。

静态检测是通过对源代码进行审计分析来挖掘 XSS 漏洞，这种检测方法在检测 XSS 漏洞时更加高效。国内外已有很多学者进行了研究，例如，王旭[4]通过对 XSS 漏洞形成的分析，提出基于控制流分析和数据流分析的静态检测方法,对 JSP 源代码进行静态分析,得到相关语句之间的控制流信息和数据流信息,在使用这些信息来得到漏洞存在判断条件的状态组合,从而得出检测结果; Nuno Neves 等人[5]提出使用静态分析和数据挖掘检测来删除 Web 应用程序漏洞;但是静态检测效果限制于在白名单和黑名单的规则完备性,对与净化效果依赖于其净化规则是否完善,同时静态检测还依赖于源码,在现实应用中出于对技术的保密和安全限制,很难接触到应用源代码,因而普遍性不高。

动态检测是一种渗透测试的方式，通过构造一系列的攻击代码对应用程序进行模拟攻击，根据服务器的响应信息来分析判断应用中是否存在 XSS 漏洞。相比于静态检测，动态检测效率较低，需要花费较多的时间对网页进行爬取处理分析，并构造相应的攻击向量库。国内外也对提高动态检测效率进行了研究，沈寿忠等[6]提出基于爬虫的 XSS 漏洞检测工具设计与实现，对漏洞产生进行了分析，并通过爬虫针对表单和 URI 两方面的信息提交处对 XSS 漏洞进行检测，提高了自动化效率，但是对输入点检测不够全，漏报率较高；李威等[7]提出利用巴科斯范式（BNF）自动生成初始攻击向量，实现了攻击向量的自动化生成;但是没有对所生成的攻击向量进行分类处理，使得攻击向量的冗余较大，无效测试较多。李洁等[8]提出基于动态污点分析的 DOM XSS 漏洞检测算法，通过对输入数据进行污点标记，获取污点传播路径，生成污点数据集，较好的检测基于 DOM 的 XSS 漏洞。但是误报率较高，覆盖率不足。

动静混合检测是通过收集数据源，建立数据集，通过机器学习等方法对网站进行检测。

张海燕等[9]提出基于决策树分类的跨站脚本攻击检测方法，将 Web 中容易受到攻击的元素和对象作为特征属性，通过决策树分类方法对其进行分类，通过训练好的模型对其他 Web 应用进行检测，具有较好的效果；但是该方法的是建立在已知攻击行为中，对未发生的攻击行为检测效果较差，比较依赖于训练数据集的质量。

1.3 本文贡献

本文根据上面的分析，提出一种优化的动态检测方法：优化爬虫算法，过滤掉与检测站点无关 URI 和资源型 URI，同时通过 HASH 算法排除重复 URL；利用探子算法对网页预处理，过滤无输入输出点的页面，同时标记出输入、输出出点对，以便后续攻击脚本的检测，节省时间；对攻击向量集进行分类的生成并进行变异处理，减少了攻击请求量；此外增加对网站中伪静态页面的检测，减少系统的误报率。试验证明该系统在 XSS 漏洞检测上有较好的检测效果。

2 XSS 漏洞检测系统的设计与实现

2.1 检测的框架

本文的检测系统主要又以下几个模块组成：信息收集模块，预处理模块，攻击向量集生成模块，攻击检测模块，生成报告模块。具体框架如图 1 所示：

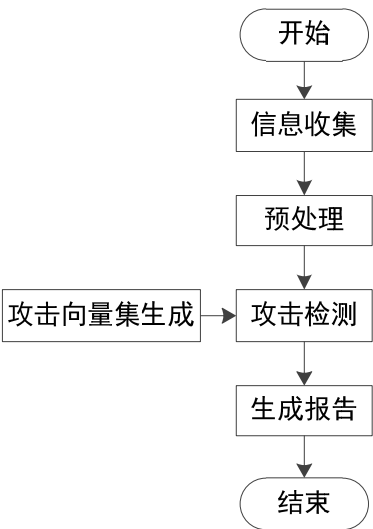


图 1 XSS 漏洞检测框架

信息收集模块是对 Web 站点进行解析，索取网页中属于主站点的 URL 以及所有输入点，其中包括伪静态页面输入点，过滤掉重复的 URL 以及资源型 URL；预处理模块使用合法向量对输入点进行注入，过滤不存在 XSS 漏洞页面和不存在注入的输入点，同时获取页面注入、输出点对；攻击向量集生成模块，自动的生成攻击向量集，并对向量集进行变异处理，以便绕过过滤机制；攻击检测模块，针对输入点，构造带有攻击向量集中的攻击向量的请求，对其进行 XSS 渗透测试；生成报告模块，对渗透测试的对反馈的结果进行匹配，判断攻击是否成功，记录结果生成报告。

2.2 信息收集

信息收集主要分为：URI 的爬取，信息分析获取。

URL 获取：根据给定的爬取 URL，通过广度优先的爬虫算法进行爬取，同时根据域名命名规则对无关 URL 进行过滤，同时利用模糊匹配算法对资源型 URL 进行过滤处理（如图片 URL 等），最后通过 HASH 算法对爬取的 URI 进行去重处理以防止环链爬取。

信息获取：对每个 URL 请求响应的网页进行解析分析，获取网页中 URL，网页散列值，以及输入点，并对无参 URL 进行伪静态检查，防止漏报。本文考虑的输入点主要包括：请求头中的参数，表单输入，URL 参数，伪静态 URL 中的参数。

具体流程如下：

(1)将初始 URL 加入待爬取 URL 队列，若队列不为空则取出 URL 队列里一条 URL 获取响应网页，否则结束该模块。

(2)解析响应网页，首先对网页进行 HASH 并存储其值至源网页 HASH 表；解析并获取网页中所有的 URL，包括动态 JS 脚本执行中所涉及的 URL；解析并获取网页输入点，对无参数的 URL 进行伪静态检测，找出其中可能的注入点。通过这些模块获取到网页 URL，以及网页输入点信息。

(3)对上面获取的 URL 进行去重处理，然后加入到待爬取队列；对输入点也进行去重处理，并将输入点信息、URI 信息保存到信息记录模块，供预处理模块调用。

伪静态检测：当前许多网站为了推广，会将 URL 设置成伪静态的方式，方便搜索，比如当我们在 url 中输入

“http://localhost/user/xiaoming/11”时,实际执行的是用户名和 id 的传参

“http://localhost/user.jsp?name=xiaoming&id=11”。本文通过唯一标识符替换的方式来检测每个 URL 是否为伪静态 URL，例如上述替换为

“http://localhost/user/OnlyString/11”,当请求响应状态码为 200 时认为其可能存在 XSS 漏洞，将该点信息加入到输入点表中进行后续检测。

2.3 预处理模块

该模块通过构造不含特殊符号的正常测试向量对注入点进行预处理测试，用来过滤掉那些不存在 XSS 漏洞的注入点，同时标记那些存在 XSS 漏洞的注入点以及其对应的输出点。通过预处理过滤模块不仅可以在后续攻击向量检测中避免无效注入攻击，同时可以获取注入-输出点键值对，方便测试结果对比，大幅提高了后续 XSS 漏洞检测效率。该模块检测过程如图 2 所示。

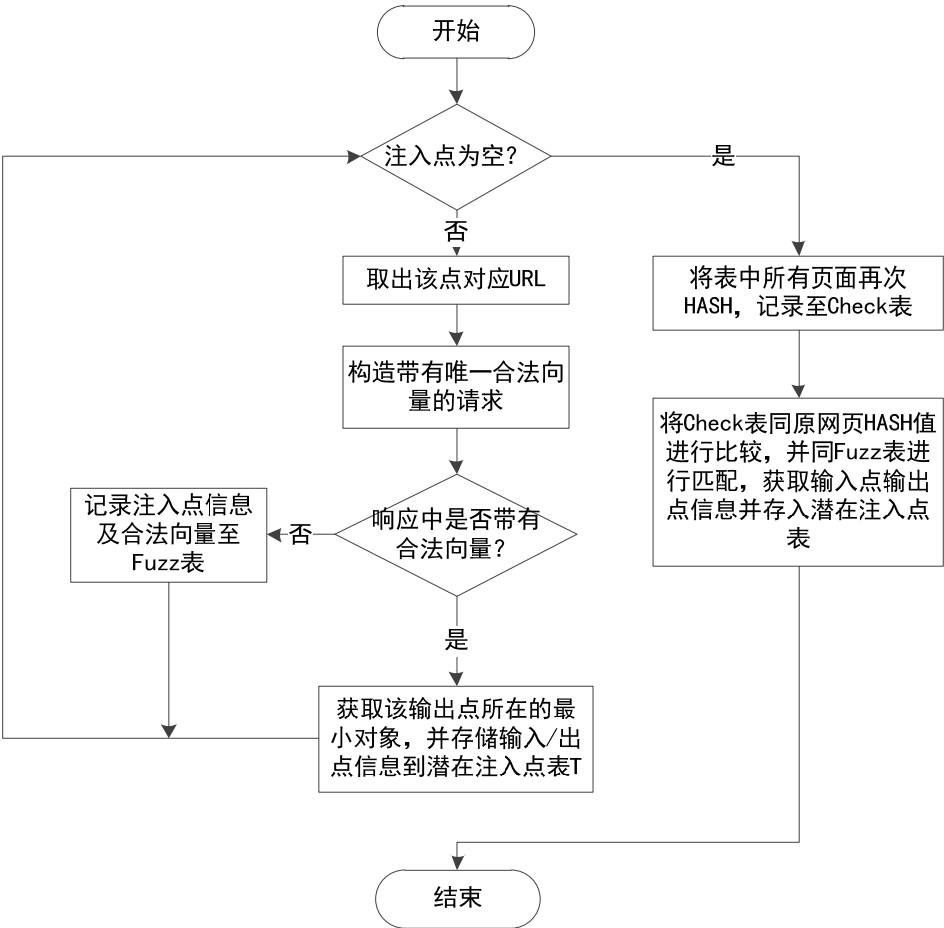


图 2 预处理模块流程图

首先若注入点表不为空，则取出一注入点，为其构造唯一合法向量进行请求。然后检测响应页是否含有唯一合法向量，若存在则将该注入点信息以及输出点信息记录到潜在注入点表中，做后续检测；若不存在则将该注入点信息及合法向量存储于 Fuzz 表中做进一步排查。最后当注入点表遍历结束时，对所有 URI 页面进行再次 HASH 并记录其值至 Check 表，将 Check 表中网页 HASH 值同原网页 HASH 值进行对比，找出 HASH 值变化的网页，同 Fuzz 表进行匹配，获取相应输入输出点信息并记录至潜在注入点表中，此步骤是为后续检测潜在存储型 XSS。

2.4 攻击向量集生成

渗透测试中影响漏洞检出率除了输入点的爬取外就是攻击向量集的覆盖率了，攻击向量集覆盖率越高检出率越好，当时过多的攻击向量则会造成过多的请求交互，造成效率上的降低。本文通过对攻击向量集进行归类生成，在攻击检测时通过输入点类型来选取对应的攻击向量集，这样就节省大量的检测时间，从而提高检测效率。本文根据以往的攻击类型，将向量集分为五类：HTML 文本，注释，脚本代码，HTML 属性，CSS 属性。并对每个向量集中的攻击向量进行变异处理，以便绕过 Web 应用中的过滤检测，变异包括对其进行编码，大小写转换，插入特殊符号，攻击向量二次生成等方法，使攻击向量库尽可能覆盖所有漏洞类型，以减少漏报率，提高检测可信度。表 1 是每个类型的举例以及其一些变异后的样式。

表 1 攻击向量集

类型	举例	变异
HTML 文本	javascript:alert('xss')	javascript:alert(String.fromCharCode(120,115,115))
脚本代码	setTimeout('alert('xss')',0)	setTimeout('\u0061\u0063\u0065\u0072\u0074('xss'),0)
HTML 属性	onerror=eval;throw'alert(XSS)'	onerror=eval;throw'alert\x28xss\x29'
注释	*</script>alert("xss")</script>#	*</script>/**/**/alert("xss")</script>#
CSS 属性	body{xss:expression((window.x==1)?':eval('x=1;alert(xss);')':);}	body{\078\073\073:\065\078\070\072\065\073\073\069\06f\06e((window.x==1)?':eval('x=1;alert(xss);')':);}

1602.5 XSS 检测模块

这个模块通过对待检测站点进行攻击向量模拟攻击，测试出 XSS 漏洞。检测流程如图 3 所示。

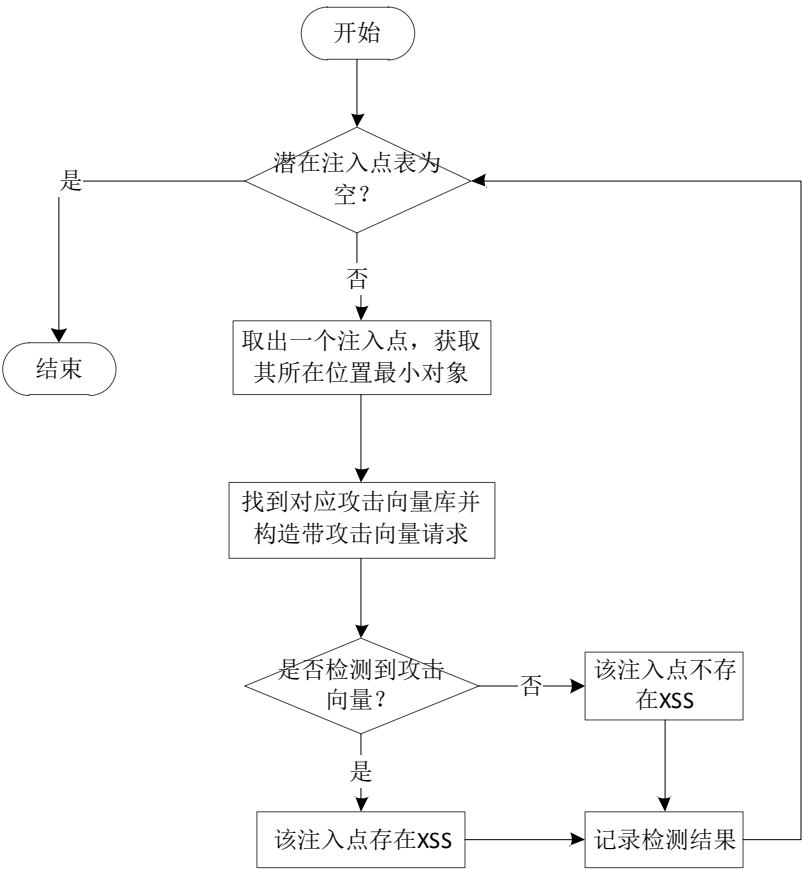


图 3 XSS 漏洞检测流程图

165XSS 检测模块是基于前几个模块的基础上实现的，首先从潜在注入点表中取出注入点，根据输出点所在位置的最小对象选择相应的攻击向量库，结合攻击向量库中获取的攻击向量

构造带攻击向量的请求，然后模仿用户访问该输入点对应的输出点，检测是否有攻击向量存在，若存在则标记其 XSS 漏洞类型并记录在检测结果中，若不存在则此注入点不存在注入，遍历下一个注入点重复操作，直到所有注入点遍历完毕，总结并给出检测结果。

3 实验结果与分析

为了验证本文提出的方法在检测 XSS 漏洞方面的有效性和检测效率，本文实现了一个 XSS 漏洞检测的原型系统，并通过 WebXsser 应用（约含有 50 个漏洞其中伪静态 3 个）对该系统进行了性能测试，同时选择 AWVS 漏洞扫描器进行扫描对比。本文采用 3 个性能分析指标，包括：检测到的漏洞个数、误报数、检测时间。表 2 为检测结果。

表 2 WebXsser 应用检测结果

检测系统	检测的漏洞数	误报数	检测时间（s）
AWVS	47	4	260
本系统	45	2	280

根据分析实验结果对比可以发现，本文检测系统相比于 AWVS，其检测漏报率较小，测试时间较短，但是误报率较高，总的来说针对前面提到的不足之处，本文的检测方法起到较好的优化效果。

4 结束语

本文通过利用网页预处理模块对待检测应用进行预处理，过滤掉无 XSS 漏洞的注入点，同时增加伪静态检测模块，以及对攻击向量集进行分类的生成、并进行变异处理，减少无效攻击请求，使得本文的检测方法在检测 XSS 漏洞上有较好的检测效果。但是本文检测系统误报率较高，在后续的研究中会针对攻击判定进行分析研究，同时对攻击向量集的优化进行研究，以达到用较小的攻击向量集实现 XSS 漏洞检测，进一步提高检测效率。

[参考文献] (References)

[1] OWASP Top 10 Application Security Risks 2017[OL]. [2017].
https://www.owasp.org/index.php/Top_10-2017_Top_10

[2] Gupta S, Gupta B B. Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art[J]. International Journal of System Assurance Engineering & Management, 2017:1-19.

[3] 吴翰清.白帽子讲 Web 安全[M].北京:电子工业出版社, 2012

[4] 王旭. 基于控制流分析和数据流分析的 Java 程序静态检测方法的研究[D].西安电子科技大学,2015.

[5] Nuno Neves; Miguel Correia. Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining[J]. IEEE Transactions on Reliability.IEEE.2016,65(1):54-69.

[6] 沈寿忠,张玉清.基于爬虫的 XSS 漏洞检测工具设计与实现[J].计算机工程,2009,35(21):151-154.

[7] 李威,李晓红.Web 应用存储型 XSS 漏洞检测方法及其实现[J].计算机应用与软件,2016,33(01):24-27+37

[8] 李洁,俞研,吴家顺.基于动态污点分析的 DOM XSS 漏洞检测算法[J]. 计算机应用.2016.36(5):1246-1249.

[9] 张海燕,莫勇.基于决策树分类的跨站脚本攻击检测方法[J].微型机与应用,2015,34(16):55-57+61.