基于风险数据追踪的存储型XSS漏洞检测技术

李亚威 刘梓溪 丁士俊

(四川大学电子信息学院 成都 610065)

摘 要 为解决存储型XSS漏洞的自动化黑盒检测问题,提出基于风险数据追踪的存储型XSS漏洞检测方法。依此技术可以对Web应用程序上存在的存储型XSS漏洞进行快速、深入的自动化挖掘。简要介绍了为实现该技术而需编写的自动化辅助软件的设计与实现,并用实验证明了该检测技术的有效性。

关键词 存储型XSS,Web安全

中图法分类号 TP393 **文献标识码** A

Technique for Discovering Stored XSS Vulnerability Based on Tracing Risky Data

LI Ya-wei LIU Zi-xi DING Shi-jun

(College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China)

Abstract To discover stored XSS vulnerability with black-box testing, we put forward a new technique which is based on tracing risky data. This technology can discover stored XSS vulnerability automatically on Web application quickly and deeply. This paper introduced how to design the assisted software for this technique birefly as well as prove the effectiveness of this technique.

Keywords Stored XSS, Web security

1 概述

1.1 关于XSS

Web2.0技术的网站增加了很多基于脚本语言输出的动态内容,这类型的Web站点会受到XSS的攻击。恶意攻击者向Web页面插入恶意的html代码,当用户浏览包含此恶意代码的网页时,嵌入Web页面中的html代码会被执行,从而达到恶意攻击用户的特殊目的[1]。根据2013年版的OWASP Top 10,XSS在Web应用程序安全风险中排名第三[2]。

XSS漏洞可以被大致分为两类,一类是攻击者插入的恶意html代码没有进入数据库,直接被插入呈现给用户的网页中,称为反射型XSS漏洞。另一类是攻击插入的恶意html代码进入了数据库,且该数据被Web应用程序在一处或多处调用,从而在一处或多处被插入在呈现给用户的网页中,称为存储型XSS漏洞。

1.2 对存储性XSS漏洞的检测

由于存储型XSS漏洞导致恶意代码进入数据库及多次被Web应用程序调用,一般认为存储型XSS漏洞的危害要大于反射型XSS漏洞的危害,且就以黑盒测试为方法的漏洞挖掘而言,对存储型XSS漏洞的挖掘难度要大于对反射型XSS漏洞的。在商业领域,OWASP的Xenotix XSS Exploit Framework及cURL等也仅对反射型XSS漏洞和少量存储型XSS漏洞进行防御和挖掘。

在大量运用Ajax技术和大量利用JSON传输数据的Web2.0时代,用户操作所触发的任意一个JavaScript脚本都有可能导致网页内容的变化。故已经存储在数据库中的恶意代码可能在任何时刻被加载到页面上致使XSS攻击。由于Javascript脚本的灵活性和复杂性以及用户操作的不确定性,传统的基于网络爬虫的XSS漏洞挖掘方法对Web2.0时代应用程序中存在的存储型XSS漏洞的效果是欠佳的。

2 风险数据的定义及追踪

本文所提出的存储型XSS漏洞挖掘技术为黑盒测试,需要在不知道网站源代码的情况下追寻用户输入的数据。

不是所有用户能输入数据的地方都能产生XSS漏洞。产生XSS漏洞的必要条件是:用户输入的数据在某个时刻被反馈在Web页面上。

定义1 现把符合以上必要条件的用户输入的数据定义为"风险数据"。

可能导致存储型XSS漏洞的风险数据流向如图1所示。

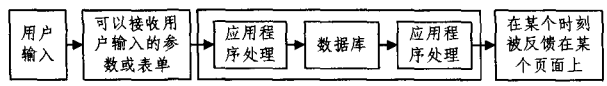

图1 可能导致存储型XSS漏洞的风险数据流向

本文提出的存储型XSS漏洞的方法,需要"追踪"风险数据。黑盒测试不需要时时刻刻跟踪数据在应用程序内部处理的具体情况和在数据库中的存储情况,只需追踪风险数据的终点,即追踪:浏览器发出什么请求后,风险数据被反馈在什么页面上,并在追踪之后建立风险数据的终点队列(见图2)。

李亚威(1993—),男,主要研究方向为Web应用程序安全,E-mail:fringee@126.com;**刘梓溪**(1993—),男,主要研究方向为Web应用程序安全;**丁士俊**(1992—),男,主要研究方向为自动机算法。

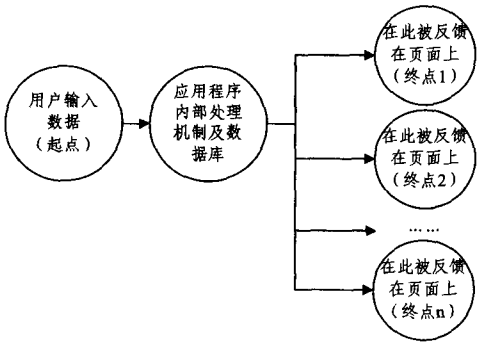

图 2　风险数据的终点队列

3　其他相关概念的定义

3.1　测试点的定义

发生存储型 XSS 攻击的必要条件是用户输入的数据进入了数据库，并在某个时刻被 Web 应用程序调用从而加载到网页上。

定义 2　如果某 URL 中的某个参数或某页面上某个表单满足以上必要条件，在进行存储型 XSS 漏洞检测的过程中就必须对该 URL 中的该参数或该页面上的该表单进行测试，以分析用户在此输入数据时是否会产生 XSS 攻击。现定义这样的参数或表单为“测试点”。

一个完整的 Web 应用程序有诸多这样的测试点，在存储型 XSS 漏洞检测时需要对 Web 应用程序中每个测试点进行逐一测试。

定义 3　在测试点中输入数据并向服务器提交时产生的 HTTP 请求报文，现定义为“测试点的请求报文”。

3.2　标记字符串定义

追踪测试点中用户输入数据的终点，需要在测试点提交特殊的字符串作为标记，该字符串所必需的性质为具有特殊性，不会与 HTTP 响应报文中的其他内容相同，从而方便自动化软件在 HTTP 响应报文中识别出来自该测试点中用户输入的数据。

定义 4　现定义这个特殊的字符串为“标记字符串”。

3.3　测试用例定义

测试用例集是指所有可以用于 XSS 攻击的实例代码及其所有通过任何方式转义变形后的结果的集合。根据 XSS Filter Evasion Cheat Sheet[] 整理出所有已知的可以用于跨站攻击的 XSS 实例代码的集合。对每个代码进行重编码、敏感词绕过、代码方式重构、字符书写方式混淆、代码格式重构[3]这些操作来将其扩展成为测试用例集。

定义 5　现定义可以用于 XSS 攻击的字符串为“测试用例”。

4　基于风险数据跟踪来检测存储型 XSS 漏洞

要实现本文所提出的技术，需要为之编写一个自动化辅助软件（以下简称“软件”）。该软件有一数据库，其中的一个数据表（设为表 C）存储有所有的测试用例。测试过程主要分为确定测试点、建立风险数据终点队列、自动化测试 3 个步骤。

（1）确定测试点

①测试人员选定测试点之后，使用事先选定好的标记字符串向测试点中输入，向服务器提交测试点的请求报文。软件监测浏览器与服务器之间的所有 HTTP 数据报文，将测试点的请求报文存入数据库的一个数据表中（设为表 A）。

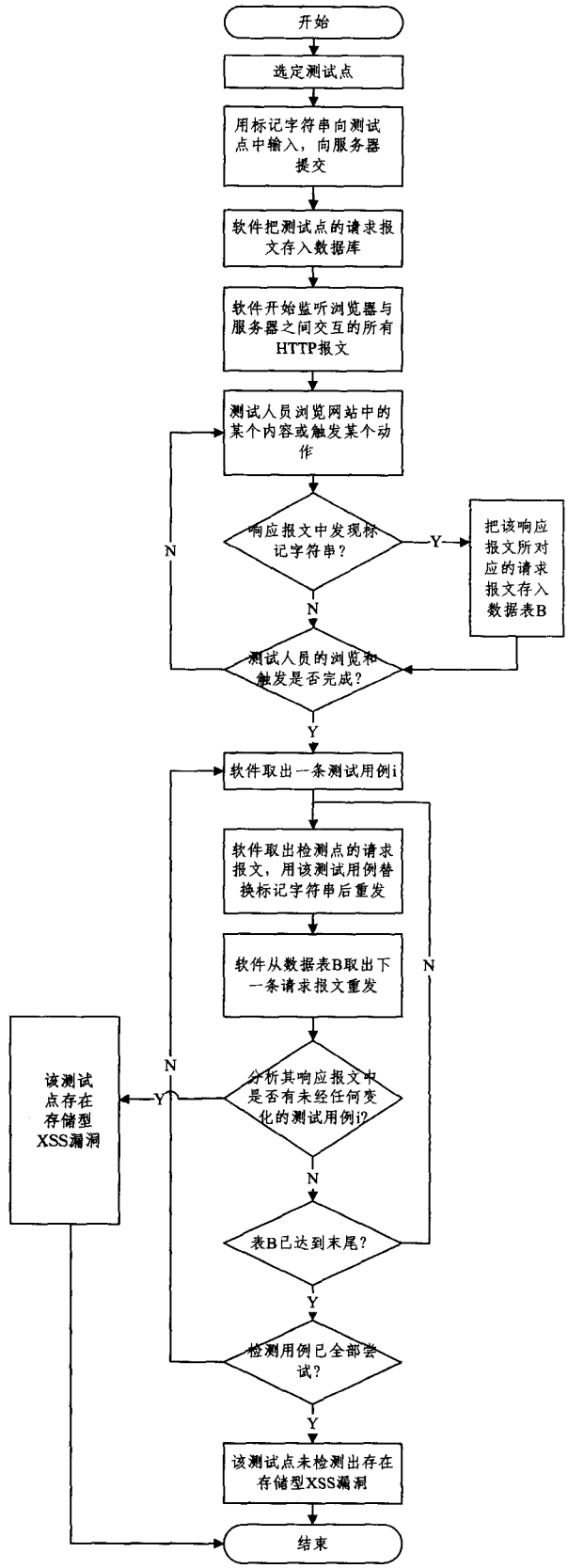

图 3　总流程

（2）建立风险数据终点队列

②测试人员尽可能地在浏览器中浏览该 Web 应用程序的各种内容，尽可能多地进行动作的交互，触发各种脚本。在

此过程中软件一旦发现某 HTTP 响应报文中出现标记字符串，则软件把与之相对应的请求报文放入数据库的一个数据表中(设为表 B)。

③测试人员浏览结束后，告知软件，软件停止监测 HTTP 数据报。

此时通过测试人员的人工触发和软件的监测，已建立起风险数据的终点队列(存储在表 B 中)。

(3)自动化测试

④软件从表 C 中取出一条测试用例，读取表 A 中的请求报文，用一条测试用例替换报文中的标记字符串，并修改 HTTP 报头中的 content length 字段为正确值，重发该报文。

⑤软件重发表 B 中所有的请求报文，分析每次的响应报文中是否包含未经修改过的测试用例。若有，则说明该测试点存在存储型 XSS 漏洞。

⑥取出表 C 中下一条测试用例，重复④、⑤两个过程。若表 C 中的所有测试用例都被尝试后，仍未发现存储型 XSS 漏洞，则可判断在该测试点是安全的，不会导致“存储型 XSS 漏洞”。

整个测试过程(见图 3)是连续进行的，软件会时刻保持测试人员当前使用的 Cookie 和 SessionID，保证报文的重放和响应正常。

5 实现该技术而需编写的自动化辅助软件的设计

5.1 软件主要结构

软件主要由 Record 模块、Monitor 模块、Test 模块组成。Record 模块确定测试点，Monitor 模块建立测试队列，Test 模块自动化测试。

(1)Record 模块(见图 4)：记录检测点的请求报文，并把报文中检测点的用户输入数据替换成标记字符串，将该请求报文存入数据库(设存进表 A)。

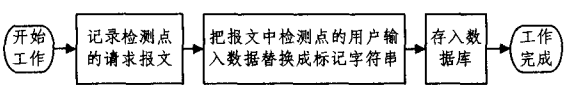

图 4 Record 模块

(2)Monitor 模块(见图 5)：进入第②步时，该模块检测浏览器与服务器交互的所有 HTTP 报文，并把带有标记字符串的响应报文所对应的请求报文存入数据库。

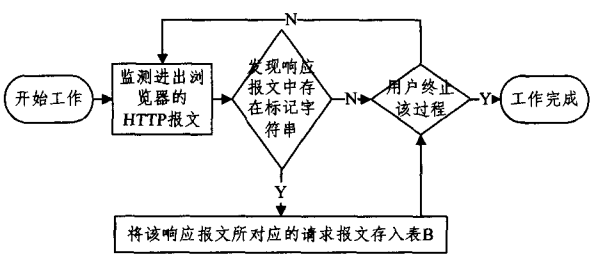

图 5 Monitor 模块

(3)Test 模块(见图 6)：该模块主要用于检测存储型 XSS 漏洞是否存在。进入第④步之后，该模块逐一读取数据库中的所有测试用例。每读取一条测试用例，用该测试用例代替检测点请求报文中的标记字符串，修改报头中相应 content length 的值，重发修改后的报文。然后重发 Monitor 模块所存入数据库的所有报文，分析每次的响应报文中是否包含未经修改过的测试用例。若有，则说明该测试点存在存储型 XSS 漏洞。

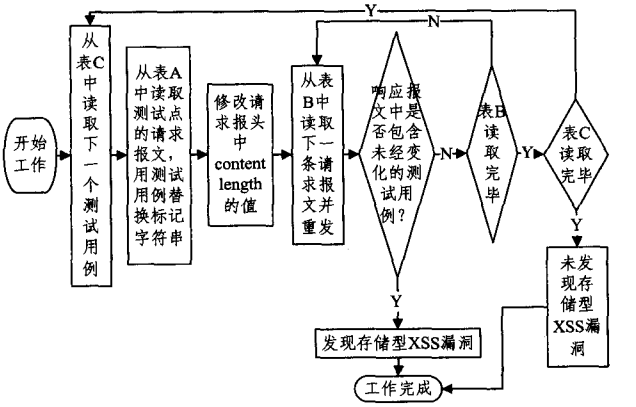

图 6 Test 模块

5.2 数据库结构

数据库中分为独立的 3 个表：表 A 中记录 Record 模块记录的检测点请求报文；表 B 记录 Monitor 模块所记录的请求报文；表 C 记录所有的测试用例。

5.3 软件测试

在本地用 wamp 架设 PHP 网站。其中 edit. php 页面允许用户编辑个人信息，并保存进数据库(见图 7)。info. php 页面允许用户查看个人基本信息，单击“More Detail”按钮时可查看更多个人信息(使用 Ajax 异步传输数据)。该测试用网站上所有的表单都可进行存储型 XSS 攻击(见图 8)。

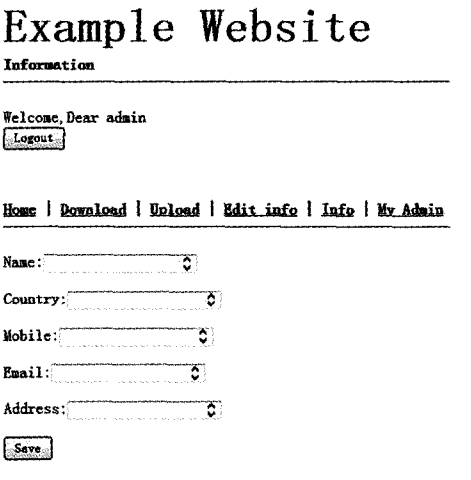

图 7 edit. php 页面

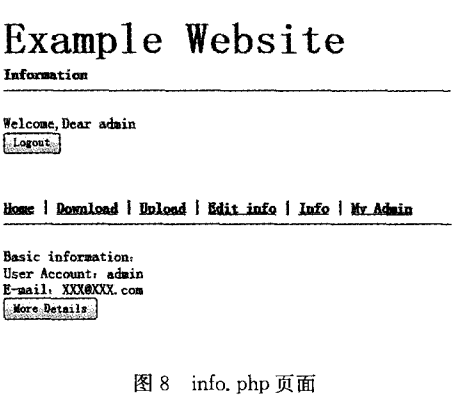

图 8 info. php 页面

测试结果：实例网站中 edit. php 文件接收用户输入时未经严格过滤而进入数据库，在 info. php 中通过同步和异步的

方式将先前用户的输入显示在页面前，未经严格编码转换。通过本文提出的基于风险数据追踪的存储型 XSS 漏洞检测技术可以快速成功地检测出上述存储型 XSS 漏洞。

结束语 本文所提出的基于风险数据追踪的存储型 XSS 漏洞检测技术能自动化检测出 Web2.0 时代中应用大量 Ajax 技术和 JSON 传递数据的 Web 应用程序中存在的存储型 XSS 漏洞，帮助渗透测试人员节省精力与时间。本文所提出的方法在一定程度上依赖于报文的重放，对于多数 Web 应用程序，本测试技术是有效的，但在一些结构功能较为特殊的 Web 应用程序中，较短时间内同一报文的多次重放未必能收到结构相同的响应结果，从而也使本技术具有一些局限性。

参 考 文 献

[1] 黄玮，崔宝江，胡正名. Web 应用程序客户端恶意代码技术研究与进展[J]. 电信科学，2009(2)：72-79

[2] OWASP. OWASP Top Ten Project[R]

[3] 吴子敬，张宪忠，管磊，等. 基于反过滤规则集和自动爬虫 XSS 漏洞深度挖掘技术[J]. 北京理工大学学报，2012(4)：396-400

(上接第 223 页)

4.2.4 隔离就医的影响

由图 9 可以看出，隔离就医对控制疫情的重要性。因此，国家应该及时披露信息，以及采取有效措施督促有发病状况的人及时就医。当然除了督促和宣传之外，还应该结合当地医疗条件，尽量增加就医的方便性。

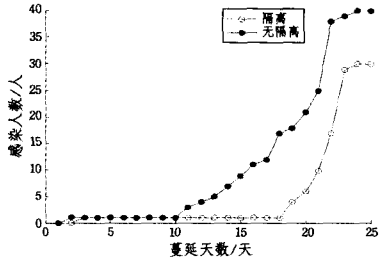

图 9 隔离就医对疾病蔓延的影响

4.2.5 接种疫苗的影响

根据 3.1.4 节考虑接种疫苗的影响，并假设第 13 天研究出疫苗并可投入使用。图 10 表明了疫苗对控制疫情的重要性。一方面，国家应该加大对医疗研究的投入，提升我国的研究水平。另一方面，由于我国人口众多，制定合理的接种疫苗方案(数量，接种人群等)也会对控制疾病的蔓延产生很大的影响。

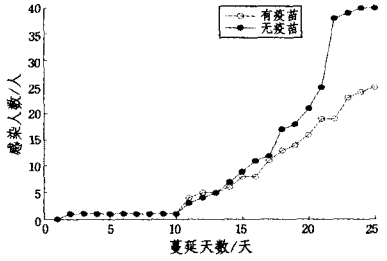

图 10 疫苗对疾病蔓延的影响

结束语 (1)本文尝试使用复杂适应系统的相关方法，初步实现了传染病传播过程的模拟。通过合理的引入积累量这一属性，设计了各主体间的相关规则，从微观个体行为出发了解传染病传播这一宏观规律，并且给出了合理的防控措施。

(2)本文考虑了人际交往、公众卫生意识等对传染病传播的影响，这符合复杂性科学的观点，即传染病学问题不仅仅是自然科学的医学问题[16]。复杂性科学视角下的传染病防治，应该是提倡多学科、多样性和多元化的防治。

(3)通过对模型的不断完善，模型能够更为有效地帮助了解更为一般的传染病传播蔓延过程。这种模型区别于传统的 SIR 模型，它不仅可以省去复杂的微分方程运算，而且比传统的模型更易于直观分析及拓展研究。

参 考 文 献

[1] 姜启源，谢金星，等. 数学模型[M]. 北京：高等教育出版社，2011

[2] 王小莉，曹志冬，曾大军，等. 应用 SEIR 模型预测 2009 年甲型 H1N1 流感流行趋势[J]. 国际病毒学杂志 ISTIC，2011，18(6)

[3] 段玮，杨鹏，张奕，等. 北京市中小学生甲型 H1N1 流感感染影响因素的病例对照研究[J]. 国际病毒学杂志，2011，18(1)：11-14

[4] 彭志行，鲍昌俊，赵杨，等. ARIMA 乘积季节模型及其在传染病发病预测中的应用[J]. 数理统计与管理，2008，27(2)：362-368

[5] 瞿毅臻，李琦，甘杰夫. 基于 Repast 平台的 SARS 传播仿真建模研究[J]. 计算机科学，2008，35(2)：286-288

[6] 余雷，薛惠锋，高晓燕，等. 基于元胞自动机的传染病传播模型研究[J]. 计算机工程与应用，2007，43(2)：196-198

[7] Carpenter，Connie V. Agent-based modeling of seasonal population movement and the spread of the 1918-1919 flu：the effect on a small community. Diss. University of Missouri--Columbia，2004

[8] 谭跃进，邓宏钟. 复杂适应系统理论及其应用研究[J]. 系统工程，2001，19(5)：1-6

[9] 梁志妹. 基于多 Agent，复杂网络与 GIS 的甲型 H1N1 流感传播仿真平台研究[D]. 昆明：云南师范大学，2011

[10] 倪顺江. 基于复杂网络理论的传染病动力学建模与研究[D]. 北京：清华大学航天航空学院，2009

[11] 林国基，贾珣，欧阳颀. 用小世界网络模型研究 SARS 病毒的传播[J]. 北京大学学报：医学版，2003，35(z1)：66

[12] 杨忠. 几种重要传染病疫苗的研究现状[J]. 解放军预防医学杂志，2004，21(5)：384-387

[13] 李亮，嵇红，张伟伟，等. 甲型 H1N1 流感流行病学及应对策略研究进展[J]. 现代医学，2010，38(1)：77-81

[14] 吴慧，宋淼，申辛欣. 1996-2009 年中国狂犬病流行病学分析[J]. 疾病监测，2011，26(6)：427-430

[15] 曹志冬，曾大军，王全意，等. 北京市甲型 H1N1 早期流行的特征与时空演变模式[J]. 地理学报，2010，65(3)：361-368

[16] 吴彤，从复杂性科学视野看 SARS 的防治[J]. 科学技术与辩证法，2003，10(20)

[17] Downey A B. 复杂性思考[M]. 张龙，译. 北京：机械工业出版社，2012

[18] MATLAB 宝典[M]. 北京：电子工业出版社，2007