文章编号:1671-4598(2016)02-0251-04 DOI:10.16526/j.cnki.11—4762/tp.2016.02.069 中图分类号:TP391 文献标识码:A

一种新的自学习 web 应用响应分析算法

马 昊,黄晓芳,陶 启,张亚文
(西南科技大学 计算机科学与技术学院,四川 绵阳 621010)

摘要:在 web 应用自动渗透测试技术的发展中,由于在 web 应用响应分析的自动化与智能化方面的研究不足,现有 web 应用自动渗透测试中仍然需要人为经验干预,限制了渗透测试的效率,因此,在研究了关键字响应分析技术与被动提取技术的基础上提出了自学习响应分析算法,该算法利用关键字词库对响应结果进行分析,若没有匹配成功再利用启发式分析技术进行分析,当分析结果有效则提取响应的关键字加入词库以达到自学习的目的;实验证明,该算法能够对测试响应结果自动地进行分析,突破了关键字分析技术只能分析含有关键字的响应这一局限,同时,比单纯被动响应提取技术具有更高的效率。

关键词:web 应用;响应分析;自学习

A New Algorithm of Self—learning Web Application Response Analysis

Ma Jiong, Huang Xiaofang, Tao Qi, Zhang Yawen
(College of Computer Science & Technology, Southwest University of Science & Technology, Mianyang 621010, China)

Abstract: Little research about the analysis of web applications' s response has been done . Its level of automation and intelligence is very low. And web penetration still requires human intervention which directly limit its efficiency . To solve this problem, key words analysis and negative response extraction have been studied, and with the help of the two technology a algorithm named self—learnning response analysis is proposed. At first the algorithm use key words to ananlyse the response. If it fails, heuristic analysis technology is used to work it out. The key words extracted from the response will be stored in the database which is used to keep key words. Experimental results show that the algorithm can analyse the response quickly and correctly. And it can analyse the response that key words analysis can not. At the same time, it is more efficient than negative response extraction.

Keywords: web application; response analysis; self—learning

0 引言

伴随着计算机技术与互联网技术的迅猛发展,Internet 极大地丰富了人类的生活,世界各地丰富多彩的信息得以在一瞬间进行交流。Web 应用已经遍布使用到各行各业中,各种各样的服务满足了不同人群的不同需求。同时,web 应用不再是一些共有的即能够共享的信息,大量私密的和高度敏感的信息由于网上应用的需要而保存在相关 Web 服务器中。目前,Web 应用还在朝着更加丰富和复杂、但又更易于让用户操作的方向不断地发展。在互联网给用户带来过去从未有过的丰富生活的同时,web 应用的安全问题[1]也愈演愈烈,已经成为世界所关注的焦点。因此对 web 应用的安全测试就成为了当今网络技术一个非常重要的研究课题。很多 web 应用渗透测试软件应运而生,但是,因为 web 应用响应分析技术自动化水平较低,限制了渗透测试软件在自动化智能化方向的发展,所以自动化 web 响应分析算法的研究成为解决这一问题的关键。

收稿日期:2015-08-26; **修回日期**:2015-09-17。
基金项目:国家自然科学基金项目(61303230);四川省科技支撑计划项目(11ZS2010)。
作者简介:马 昊(1990-),男,山东泰安人,硕士研究生,主要从事网络安全方向的研究。

1 相关研究

目前,对 web 安全问题的研究多是关于其运行环境的研究,例如 web 应用所依赖的服务器、服务器软件、数据库等。对 web 应用程序渗透测试技术[2-8]的研究在国内外都是刚刚起步,渗透测试程序会首先向 web 应用发送精心构造的带有攻击字符串的请求,然后根据 web 应用的响应来分析漏洞存在与否。但是,由于这些响应信息都是由程序员设计的,这就导致了由程序来理解分析这些响应信息就会有一定的困难。这也是在文献[3-7]中渗透测试程序的响应是由人工分析而没有实现分析自动化的原因。

为解决这一问题 Raghavan 等人在文[9]中提出了基于关键字的响应分析算法。该算法通过匹配响应页面中的关键字(如“ODBC Error”、“incorect”、“wrong”、“missing”、“invalid”、“erro”、“error”等)来判断渗透测试是否成功,web 应用是否存在漏洞。该算法的优点是当错误的响应信息中存在这些关键字时算法耗时短,计算资源、网络资源占用少因而具有很高的效率。缺点也非常明显,当错误的响应信息中不包含这些关键字时,该算法无法对响应做出正确的分析判断。可见单纯的关键字分析算法有着很大的局限性,不能运用于实践之中。

YaoWen Huang 等人在文[8]中提出了一个简单的被动响应提取算法(negative response extraction, NRE),下面以 SQL 注入为例简单阐述该算法的思想:

1) 使用 SQL 注入字符串构造一个对目标 web 应用的请

求，将响应记为 R1；

2）使用一个非法字符串（如长度过大的字符串）构造会产生错误的 web 应用请求，将响应记为 R2；

3）使用一个合法字符串构造一个 web 应用请求，将响应记为 R3。

假定 R1 响应失败，对比 R1、R2、R3 三个响应的相似程度可以推断出 R1 响应的意义，其可能的组合如表 1 所示。

表 1 NRE 算法组合形式

关系	代表意义
R1＝R2＝R3	不存在注入漏洞或恶意字符串被过滤
R1＝R2≠R3	不存在注入漏洞
R2＝R3≠R1	恶意攻击字符串被过滤
R1≠R2≠R3	恶意攻击字符串被过滤

该算法可以在不需要任何附加信息（如关键字）的前提下分析出 R1 响应所代表的含义，主要功能是判定 R1 响应失败的原因是因为恶意字符串被过滤还是因为 web 应用根本不存在注入漏洞。但是该算法的缺点也是非常明显：

首先，假定 R1 响应是一个渗透测试失败的响应，那这个 R1 响应成功与否又依据什么来判定呢？

其次，对于一个 SQL 注入字符串就需要至少发送 3 次 web 请求，效率较低，网络延迟会对渗透测试软件产生很大影响。

2 自学习响应分析算法

在以上研究的基础上，我们对被动响应提取算法进行了改进，并结合关键字响应技术提出了一种自学习响应分析算法。该算法实现了自动化响应分析，算法描述如下：

1）利用关键字响应分析技术来分析带有攻击字符串请求所引发的响应，当匹配关键字成功时直接得出相应的结论，分析结束。若没有匹配成功则执行 2。

2）启发式响应分析，主要完成两项工作：渗透测试结果判定与关键字学习。下面是启发式响应分析中用到的概念与变量的定义。

定义 1 字符串 S，模拟的 get 或 post 请求中的参数（构造方法在章节 3．2．3 阐述）。

定义 2 攻击请求 Q1，参数为 S＋攻击字符串的 get 或 post 请求。

定义 3 错误请求 Q2，带有非法参数的 get 或 post 请求。

定义 4 正常请求 Q3，参数仅为 S 的 get 或 post 请求。

定义 5 响应 R1/R2/R3，为发送攻击请求/错误请求/正常请求时所产生的响应。

定义 6 关系 R，表示 R1、R2、R3 三者之间的关系。

定义 7 记录 FailCount，记录渗透测试中攻击字符串测试没有通过的个数，初始值为 0。

定义 8 屏蔽词，英文中的代词、数词、冠词、介词、连词、感叹与其他词（am、is、are、be、being 等）。

定义 9 关键字词库，响应匹配关键字的集合，每个关键字有一个频度属性与更新时间属性。

定义 10 准关键字词库，测试过程中所有字词的集合，每个字词有一个频度属性与更新时间属性。

启发式响应分析中，渗透测试结果判断的流程如图 1 所示。

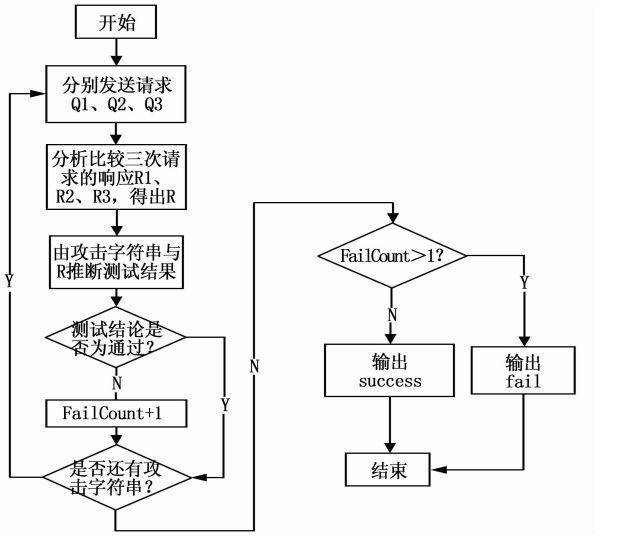

图 1 渗透测试结果判断流程图

步骤描述如下：

1）首先，分别发送请求 Q1、Q2、Q3，获得响应 R1、R2、R3。

需要注意的是，与 Y. W. Huang 等人提出的 NRE 不同，该算法需要根据攻击字符串与 R1、R2、R3 的关系 R 共同推断该测试是否成功。也就是说，需要攻击字符串与关系 R 共同确定其对应的测试结论，因为不同的攻击字符串针对同一个关系 R 会得出不同的结论。下面是 3 个示例是 3 个攻击字符串与关系 R 的组合。

示例一：攻击字符串“’”（http：//xxx.com/xx.php? id＝’）与关系 R 所对应的结论如表 2。

表 2 “’”与关系 R 对应结论

关系 R	代表意义
R1＝R2＝R3	恶意字符串被过滤
R1＝R2≠R3	可能存在注入漏洞
R2＝R3≠R1	可能存在注入漏洞
R1＝R3≠R2	不存在注入漏洞或恶意字符串被过滤
R1≠R2≠R3	可能存在注入漏洞

示例二：攻击字符串“ and 1＝1”（http：//xxx.com/xx.php? id＝X and 1＝1）与关系 R 所对应的结论如表 3 所示。

表 3 “ and 1＝1”与关系 R 对应结论

关系 R	代表意义
R1＝R2＝R3	不存在注入漏洞或恶意字符串被过滤
R1＝R2≠R3	不存在注入漏洞或恶意字符串被过滤
R2＝R3≠R1	不存在注入漏洞或恶意字符串被过滤
R1＝R3≠R2	可能存在注入漏洞
R1≠R2≠R3	不存在注入漏洞或恶意字符串被过滤

示例三：攻击字符串“ and 1＝2”（http：//xxx.com/xx.php? id＝X and 1＝2）与关系 R 对应的结论如表 4 所示。

表 4　“ and 1＝2”与关系 R 对应结论

关系 R	代表意义
R1＝R2＝R3	恶意字符串被过滤
R1＝R2≠R3	可能存在注入漏洞
R2＝R3≠R1	可能存在注入漏洞
R1＝R3≠R2	恶意字符串被过滤
R1≠R2≠R3	可能存在注入漏洞

2）然后，根据攻击字符串与关系 R 查表得出对应结论。因为，这些对应的结论里面对于成功的测试也只是给出可能成功的判断，所以只有对一个注入点构造的所有攻击字符串请求都得出可能存在漏洞时，才可以判断它可能存在漏洞，只要一个攻击字符串得出的结论是不存在漏洞，则该请求就不存在漏洞。

若最终结论是可能存在漏洞，则提取 R1 的英文文本信息，进行关键字学习，其流程如图 2 所示。

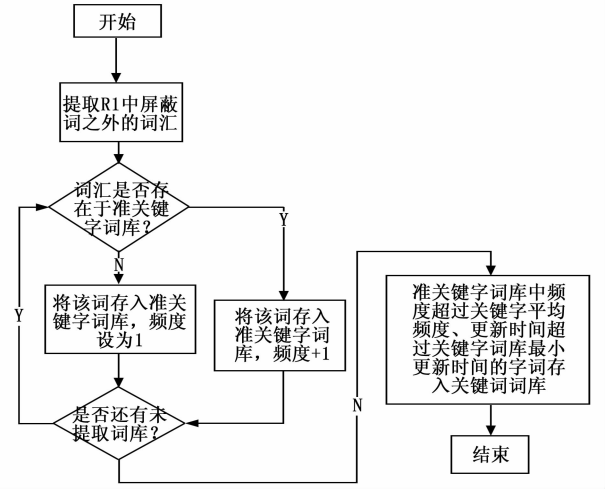

图 2　关键字学习流程图

过滤掉 R1 的屏蔽词之后存入准关键字词库，之后计算关键字词库中词条的平均匹配频度 AVG，然后把准关键字词库中频度超过 AVG 且最近更新时间大于关键字词库中最小更新时间的记录写入关键字词库。

3　算法实现

3.1　系统架构

web 响应分析算法是为了实现 web 渗透测试系统的自动化而设计，在研究基于 Java 的 Web 互动软件设计与实现[10]等文的基础上设计了该系统的总体架构如图 3 所示。

其中爬虫模块用于爬取 web 应用的内容，内容分析模块将网页内容构造成 DOM 树，将可以与 web 应用进行交互的 get 请求或 post 表单提交给渗透测试模块，渗透测试模块使用攻击字符串构造请求发送给 web 应用，并将 web 响应提交给响应分析模块，最后把分析结果生成报告输出。web 响应分析算法的实现是在响应分析模块中。

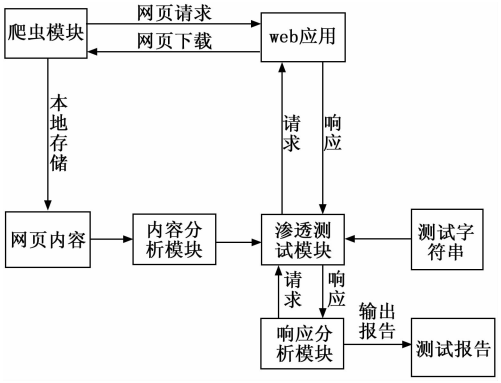

图 3　系统总体架构图

3.2　算法实现

3.2.1　建立词库

为了存储关键字需要建立关键字词库与准关键字词库，在数据库中建立两个数据库表，分别取名为 keywords 与 pre_keywords。这两个表除表名不同以外，其他都相同，都具有 3 个属性列，分别为 name（关键字拼写）、freq（关键字被匹配的频度）、update（最后一次被匹配的时间）。其中，关键字词库是根据渗透测试人员的实践经验添加的关键字，并为其 freq 与 update 赋予初值；准关键字词库是所有测试收集到的准关键字集合。

为了过滤代词、数词、冠词、介词、连词、感叹与其他词，创建一个屏蔽词库表 filter_words。

3.2.2　关键字匹配

将测试请求的响应解析为 DOM 树并抽取常见标签（如＜title＞、＜h＞、＜tr＞、＜b＞、＜p＞等）中的英文文本信息。

将文本信息拆分为单词，经过屏蔽词库的过滤后存储在一个字符串数组中。然后先使用关键字词库进行匹配（字符串完全相同或字符串是关键字的一部分，不区分大小写），若匹配成功将该关键字的频度 freq＋1，若没有匹配成功，则使用准关键字词库中频度超过关键字词库平均频度 80％的关键字进行匹配，同样，匹配成功将该关键字的 freq＋1，同时更新 update。以此法匹配字符串数组中的所有单词，只要有一个匹配成功，就可以得出“渗透测试可能成功”的结论，并将总共匹配成功的关键字数记为 N，这个 N 可以在一定程度上说明结论的可信性，N 值越大，结论越可信，因此可以作为报告中的一个参数。如果都没有匹配成功则进行启发式分析。

3.2.3　启发式分析

在启发式响应分析中字符串 S 的构造方法如下：

对于 get 请求如 http：//xxx.com/xxx.asp? Id＝10、http：//xxx.com/xxx.php? Loginname＝xxx 等，将“＝”后面的字符串赋值给 S；对于 post 请求，需要根据表单的 name 属性来构造 S，如对于＜input type＝" text" name＝" email" class＝" input－text" id＝" email" tabindex＝" 1" value＝"" /＞，如果匹配到 name 属性中含有“email”这一特征字，则需要构造一个邮箱地址形式的 S，同理，匹配到 name 属性“name”、“login”等则需要构造一个简单字符串；如果我们设

置的所有特征字都没有匹配到，就使用一个简单默认字符串。

字符串 S 构造完成之后，按照启发式分析算法，依次发送请求 Q1、Q2、Q3。然后调用 Winmerge 比较响应 R1、R2、R3，得出关系 R，用 R 与攻击字符串查表得出测试结论。

对同一个请求，所有攻击字符串结果判定都为“可能存在漏洞”时，才判断该请求的注入点可能存在漏洞。若最终测试结果为“可能存在漏洞”，就将 R1 解析为 DOM 树，抽取其主要标签的文本信息，经屏蔽词库过滤后，存入准关键字词库，如果该词已存在则 freq+1 并更新 update。

然后计算关键字词库的平均频度 AVG，把准关键字词库中频度超过 AVG 且更新时间大于关键字词库中的最小更新时间的字词加入到关键字词库中，并删除其在准关键字词库中的记录。

4 实验与结论

我们把渗透测试模块与响应分析模块组合成一个 Demo 程序。为渗透测试模块提供一系列 web 应用的请求，然后由分析测试模块进行分析。

Demo 演示程序界面如图 4 所示。

图 4 Demo 演示程序界面

为了验证算法的可行性与正确性我们做了 4 个实验，分别为：

实验一，连续测试 50 次本地搭建的带有 SQL 注入漏洞的 web 应用，即对同一请求测试 50 次；

实验二，测试 200 个不存在 SQL 注入漏洞的 get 请求；

实验三，测试 200 个存在 SQL 注入漏洞的 get 请求；

实验四，随机测试 500 个 get 请求。

其中 get 请求都是通过 google 语法“inurl”搜索得到，实验三中的请求也都经过人工测试确实是含有 SQL 注入漏洞的。由于只是测试该算法是可行性与正确性，测试数据不是很多，关键字只设置了“odbc”、“incorect”、“wrong”、“missing”、“invalid”、“mistake”、“error”并将关键字词库的频度统一设定为 50，时间设定为测试时的系统时间。实验结果如表 5 所示。

表 5 实验结果

实验名称	存在漏洞请求数	新增关键字
实验一	50	mysql、root
实验二	0	无
实验三	183	warning、webroot、usr
实验四	44	无

四次实验之后关键字频度如图 5 所示。

实验一是对同一测试请求的多次训练，成功将我们设置的

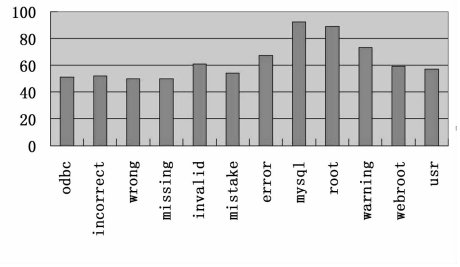

图 5 关键字频度

mysql、root 加入关键字词库；实验二的对象因没有漏洞故没有产生新的关键字；实验三的对象是已经证明存在 SQL 注入的请求，找出 183 个请求，新增 warning、webroot、usr 三个关键字；实验四是随机寻找的 get 请求，因为存在漏洞的请求较少，没有新增关键字。

经过分析可以看出，该算法可以以一个较高的正确率自动化分析 web 应用请求，并自学习添加关键字。克服了单纯关键字分析与 NRE 的不足，同时又兼具了二者的优点。其不足之处在于，需要大量的测试数据来进行训练，初始关键字频度的设定对于算法运行时的正确性与效率有着较高的影响，但这一参数的设定需要在长时间实践中确定，而且随着程序员 web 开发习惯的变化应做出相应的调整，这也是我们下一步研究改进的方向。

5 结束语

本文提出的自学习响应分析算法实现了对响应的自动化分析与判定，对于 web 应用渗透测试程序自动化的发展有着积极的意义。进一步的研究工作是改进将关键字从准关键字词库加入到关键字词库的算法，使得自学习响应分析算法能够更快更准确的将关键字找到并加入关键字词库。

参考文献：

[1] 张炳帅. web 安全深度剖析 [M]. 北京：电子工业出版社，2015.

[2] 于莉莉，杜蒙杉，张 平. web 安全性测试技术综述 [J]. 计算机应用研究，2012 (11)：4001－4005.

[3] 邢 斌，高 岭，孙 骞. 一种自动化的渗透测试系统的设计与实现 [J]. 计算机应用研究，2010，27 (4)：1384－1387.

[4] 潘古兵，周彦晖. 基于静态分析和动态检测的 XSS 漏洞发现 [J]. 计算机科学，2012 (39)：51－53.

[5] 张宗之. 基于爬虫技术的 web 应用漏洞挖掘的研究 [D]. 北京：北京邮电大学，2012.

[6] 浦 石. Web 安全渗透测试研究 [D]. 西安：西安电子科技大学，2010.

[7] 赵丽娟. Web 应用程序渗透测试方法研究 [D]. 长沙：中南大学，2014.

[8] Huang Y W，Huang S K，Lin T P. Web application security assessment by fault injection and behavior monitoring [A]. Proceedings of the 12th International Conference on World Wide Web，2003 [C]. Budapest，Hungary，2003.

[9] Raghavan S，Garcia－Molina H. Crawling the HiddenWeb [A]. Proceedings of the 27th VLDB Conference 2001 [C]. Roma，Italy，2001.

[10] 蔡小刚，何 丽，周利华. 基于 Java 的 Web 互动软件设计与实现 [J]. 计算机测量与控制，2003：44－46.