第 43 卷 第 6A 期
2016 年 6 月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

基于模糊测试和遗传算法的 XSS 漏洞挖掘

程 诚 周彦晖

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 为解决 Web 应用跨站脚本(XSS)问题,在研究当前各种 XSS 漏洞挖掘方法的基础上,通过对 XSS 漏洞特征、网站过滤方式、变形优化方法进行分析,提出了一种基于模糊测试和遗传算法的 XSS 攻击样本优化生成方法,以有效挖掘漏洞。该方法在构建 XSS 漏洞库的基础上,采用模糊测试方法随机预生成大量 XSS 攻击用例,采取过滤补全原则进行 XSS 攻击特征分析、选择与提取,利用遗传算法搜索 XSS 攻击特征空间,通过多次反复迭代生成最优的 XSS 攻击特征测试用例。分析表明,该方法能有效地发现 Web 应用中的 XSS 漏洞。

关键词 XSS 漏洞,模糊测试,遗传算法,过滤补全原则,最优攻击特征

中图法分类号 TP393 文献标识码 A

Findding XSS Vulnerabilities Based on Fuzzing Test and Genetic Algorithm

CHENG Cheng ZHOU Yan-hui

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract To solve the Web application cross-site scripting problem, on the base of the current study of various XSS vulnerabilities mining methods, a new optimization generation method of XSS attack sample was proposed. It mines the vulnerability based on fuzzy testing and genetic algorithms by analyzing features of XSS vulnerabilities, ways of site filtering, and methods of distortion optimization. First, XSS vulnerability database was constructed. Second, the fuzzing testing was used to randomly pre-generate a lot of XSS attack test cases. Then, filter and complementation principle was taken to analyze, select and extract XSS's attack features. Finally, genetic algorithms was applied to search for XSS attack features space to generate optimal XSS attack features test case through many iterations. Analysis shows that the method can effectively detect XSS vulnerabilities in Web applications.

Keywords XSS vulnerability, Fuzzing test, Genetic algorithm, Filter and complement principles, Best attack signatures

1 引言

近 20 年来,互联网技术在全球范围内得到了持续快速的发展。我国的中国互联网络信息中心(CNNIC)2015 年 2 月发布的《中国互联网络发展状况统计报告》指出:截至 2014 年 12 月,我国网民规模达 6.49 亿,互联网普及率攀至 47.9%,53.1%的中国网民认为自身比较或非常依赖互联网[1]。可见,Web 应用已经在人们的生活中扮演了越来越重要的角色,并逐渐改变了人们的生活方式。在 Web 应用中存在的各种安全漏洞也随之逐渐暴露出来,这些漏洞可能导致 Web 应用遭受各种攻击,严重影响了社会稳定、经济发展和人们的正常生活。对于持续快速发展的 Web 应用而言,存在的核心安全漏洞在很大程度上都是用户可提交任意输入[2]。而据 OWSAP 的调查显示,跨站脚本(XSS)是一直名列十大 Web 安全威胁前三甲的 Web 应用安全漏洞[3],故 XSS 漏洞的防护是 Web 安全的重要内容,而预防 XSS 攻击的前提是找出 XSS 漏洞所在,因此,在 Web 应用发布前对其进行全面彻底的安全性测试,发掘 Web 应用中的 XSS 漏洞和消除潜在的安全隐患是非常有必要的。目前,挖掘 XSS 漏洞需解决以下问题:针对网站对用户输入的过滤机制,如何更好地生成符合要求的 XSS 攻击特征的测试用例。这就必须考虑优化 XSS 漏洞攻击测试用例的生成方法。

针对基于 XSS 漏洞测试用例的过滤、优化等问题,国内外诸多研究者都提出了一些建设性的理论和研究思路。如文献[4]提出基于模糊测试的 XSS 漏洞检测方法,但没有对网站过滤机制进行深入分析。文献[5]提出基于免疫原理的 Web 攻击检测方法,对 Web 攻击数据进行编码,然后对编码后的数据集合进行学习,产生免疫集来检测 Web 攻击;该方法具有高检测率等优点,但变异过程复杂,收敛速率慢。文献[6]提出了通过反过滤规则集转换 XSS 代码并用自动爬虫程序实现漏洞代码的自动注入和可用性检验的 XSS 漏洞挖掘技术,依此方法获取 XSS 漏洞代码的转换形式及漏洞的注入入口。文献[7,17]提出了一种结合污染传播模型的代码静态分析及净化单元动态检测的方法,其中包括 XSS 漏洞所对应的源规则、净化规则和接收规则的定义及净化单元动态检测算法的描述。文献[8]提出基于支持向量机的 Web 攻击检测技术,利用支持向量机从训练样本中自动分析获取规律,并通过这些规律对未知数据进行判断来检测可疑的 Web 攻击。文献[9]提出的 policy generation 和 boundary injection 方法将 XSS 攻击检测设定在服务器端进行,较明显地减少了漏报

程 诚(1990—),女,硕士生,主要研究方向为软件测试,E-mail:15213343984@163.com;周彦晖(1972—),男,副教授,硕士生导师,主要研究方向为信息安全、软件工程,E-mail:xiaohui@swu.edu.cn。

和误报率。文献[10]在文献[9]的基础上提出一种基于特征匹配的检测方法，加入了数据预处理、特征匹配、攻击处理等几个重要模块。文献[11]提出一种应用遗传算法自动生成XSS跨站脚本漏洞检测参数的方法，设计了一套检测参数集、编码解码策略和攻击参数数据库，使用了遗传算法的交叉、变异、选择操作，并按照跨站脚本漏洞的参数规则和遗传算法的原理实现其算法。文献[12]提出一种基于遗传算法的入侵检测特征选择算法，通过删除冗余和不相关特征以及偏F检验优化特征子集，再运用遗传算法得到最优特征集合。文献[13]提出了一种基于遗传算法的特征组合选择方法，依据Fisher准则计算各种特征组合的适应度函数，提出了一种快速高效的特征选择方法。

在研究当前各种XSS漏洞挖掘方法的基础上，本文提出了基于模糊测试和遗传算法的XSS漏洞挖掘方法，采取模糊测试模拟大量XSS攻击行为输入，选用遗传算法进行攻击行为优化，通过建立XSS漏洞特征库，构造XSS漏洞的过滤、补全规则及遗传优化规则对XSS漏洞特征进行选择、提取，以生成最优的XSS特征测试用例，从而优化XSS漏洞攻击行为以深入挖掘XSS漏洞，找出可能存在XSS漏洞的地方。优化过程主要包括：基于漏洞库和模糊测试的XSS攻击样本的预生成，基于过滤和补全规则的XSS有效攻击样本的生成，以及基于遗传算法的XSS最优攻击特征样本的生成。具体优化生成过程如图1所示。

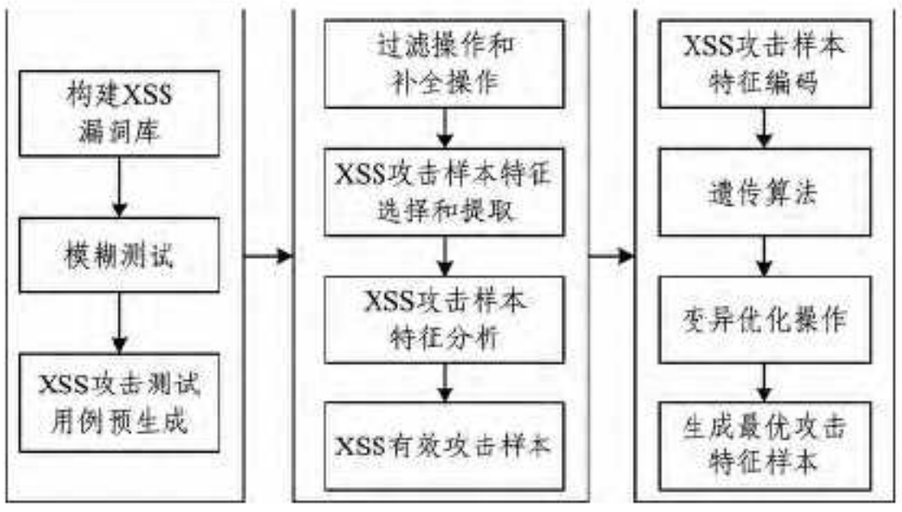

图1 XSS攻击测试用例优化生成过程

2 XSS有效攻击样本的生成

2.1 XSS有效攻击样本的预生成

2.1.1 XSS漏洞库的构建

定义1(XSS漏洞库) 将已知的任何用于XSS攻击的源代码C_{xss}的实例代码集合A定义为XSS漏洞库，库中每一个XSS代码C_{xss}代表了一种攻击类型。

$$A=\{a_1,a_2,\cdots,a_i,\cdots,a_m\},0<i<m,m=+\infty \tag{1}$$

根据式(1)，随着越来越多的XSS漏洞被发现，此漏洞库是一个可变的、增量的库。本文采用国外著名安全工程师Rsnake研究的XSS攻击脚本列表XSS Cheat Sheet[16]作为XSS漏洞库原型，利用其中XSS攻击示例进行Web应用模糊测试的XSS漏洞挖掘。XSS漏洞库示例如表1所列。

表1 XSS漏洞源代码实例表

漏洞	代码实例
a_1	〈script〉alert(ccl23);〈/script〉
a_2	〈img src="javascript:alert(/XSS attack/)"〈/img〉
a_3	〈iframesrc="javascript:alert(/XSS attack/)"〈/img〉
…	…

表1只列出了XSS漏洞代码的一部分。整个漏洞库应该涵盖JavaScript、HTML、CSS、XML等所有存在漏洞可能的全部类别和格式的代码实例。

2.1.2 基于模糊测试和漏洞库的XSS攻击样本预生成

基于模糊测试和漏洞库的XSS攻击样本预生成[20]的流程如图2所示，描述如下：

(1)输入漏洞库，即输入漏洞代码示例和成功检测到的标志，并不断更新漏洞库；

(2)生成模糊测试数据，读取漏洞库，自动化生成大量XSS攻击测试用例；

(3)执行模糊测试数据，将XSS攻击测试用例通过自动发送过程来执行检测；

(4)监测结果，对比XSS攻击成功和异常情况，记录并保存XSS攻击成功的测试用例，得到攻击成功的XSS攻击样本集合1。

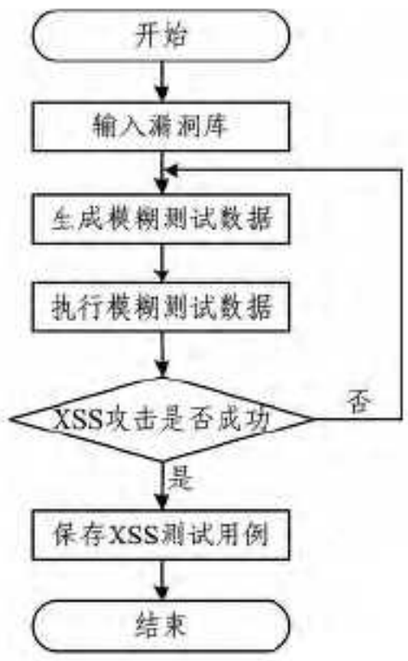

图2 基于模糊测试和漏洞库的XSS攻击样本预生成

2.2 基于过滤补全规则的XSS有效攻击样本的选择与提取

基于模糊测试和漏洞库的方法可生成一些XSS攻击测试用例，但并不能完全包括可进行XSS攻击的测试用例，且在实际应用中，大部分的网站都会对客户端输入的内容进行严格的过滤。针对上述情况，本文设计了一套反网站过滤的过滤补全操作来对XSS攻击特征再次进行选择和提取，以生成更为全面有效的XSS有效攻击样本集合2。

定义2(过滤规则)[6] 对任何XSS代码C_{xss}进行过滤的方法集合定义为F，利用集合F中方法所能过滤掉的XSS代码集合定义为Z，可得

$$F=\{f_1,f_2\}$$
$$Z=\{z_1,z_2,\cdots,z_j,\cdots,z_n\} \tag{2}$$
$$(0<j<n,n\neq+\infty)$$

即对于任何的XSS代码C_{xss}，如果有$C_{xss}\in Z$，此类代码就会被过滤掉。即

$$F(C_{XSS})\rightarrow C^*_{XSS}\notin Z \tag{3}$$

过滤行为的描述如表2所列。

表2 过滤行为描述表

过滤行为F	描述
f_1	对敏感词、关键字、特殊字符屏蔽、禁止执行。
f_2	对经过补全规则的XSS代码进行再次过滤。

定义3(补全规则) 对任何未能过滤掉的XSS代码C^*_{xss}进行补全的方法集合定义为Y，利用Y中方法所能补全的XSS代码集合定义为W，W即为可绕过过滤机制的漏洞特征库。可得

$$Y=\{y_1,y_2,y_3,y_4,y_5,y_6\}$$

$$W=\{w_1,w_2,\cdots,w_j,\cdots,w_n\} \quad (4)$$

$0<i<j,n\neq+\infty$

补全行为的描述如表 3 所列。

表 3　补全行为描述表

补全行为 Y	描述
y_1	对敏感词、关键字、特殊字符删除。
y_2	对敏感词、关键字、特殊字符进行替换。
y_3	对敏感词、关键字、特殊字符进行字符书写方式变异(全部大写或小写、大小写混合、全部全角字符或半角字符、全半角字符混合)。
y_4	对敏感词、关键字、特殊字符进行代码格式构造,插入多个 Tab 键、空格键或者回车键。
y_5	对敏感词、关键字、特殊字符添加注释,采用"*注释*"的注释方式或者多重注释的方式。
y_6	对敏感词、关键字、特殊字符添加自己的事件。

以 XSS 漏洞语句〈img src="javascript:alert(/XSS attack/)"〈/img〉为例,javascript 为某网站过滤的敏感词部分,采取过滤补全规则后得到的绕过过滤机制的 XSS 漏洞代码 W 如表 4 所列。

表 4　过滤补全操作代码示例表

过滤补全操作	代码示例
$F\cap y_1$	〈img src=":alert(/XSS attack/)"〈/img〉
$F\cap y_2$	〈img src=" jav javascript ascript:alert (/XSS attack/)"〈/img〉
$F\cap y_3$	〈img src="JavAsCripT:alert(/XSS attack/)"〈/img〉
$F\cap y_4$	〈img src="javascript:alert(/XSS attack/)"〈/img〉
$F\cap y_5$	〈img src="jav/*ascr*/ipt:alert(/XSS attack/)"〈/img〉
$F\cap y_6$	〈img src="javascript:onafterupdate= alert(/XSS attack/)"〈/img〉

2.3　XSS 有效攻击样本特征分析

使用已生成的 XSS 有效攻击样本集合 1 和集合 2 来定制 XSS 攻击测试用例的模型,并选择和提取 XSS 攻击特征。既保证完全针对跨站脚本 XSS 漏洞,同时也不乏随机性。由于提取的 XSS 漏洞源代码数据是非结构化的,需要对其进行结构化处理。将源代码数据转换为固定的特征向量作为 XSS 漏洞攻击的测试用例。

用人工挑选和数学统计相结合的方式对 XSS 漏洞源代码数据进行特征选择,可将 XSS 漏洞特征提取为 4 个参数段。

定义 4(XSS 漏洞特征)　对于任何的 XSS 攻击的源代码 C_{xss},其代码特征 $T_{C_{xss}}=(L,P,K,S)$,包含 4 个特征参数:L 为漏洞产生位置,P 为 Web 页面中的脚本注入点,K 为常见的特殊关键字,S 为常见的特殊关键字符。

漏洞产生位置有 4 个,分别存在于 l_1:Web 页面的 HTML 注释〈html〉〈/html〉,l_2:INPUT 元素〈input〉〈/from〉,l_3:〈script〉〈/script〉块结构,l_4:〈body〉〈/body〉块结构[11]。即

$L=\{l_1,l_2,l_3,l_4\}$

Web 页面的脚本注入点有 3 个,分别为 p_1:Text,p_2:TextArea 和 p_3:Password 控件[11]。即

$P=\{p_1,p_2,p_3\}$

常见的特殊关键字有 k_1:script,k_2:prompt,k_3:location.hash,k_4:@import,k_5:eval,k_6:XMLHttpRequest,k_7:ActiveXobject 等。即

$K=\{k_1,k_2,k_3,k_4,k_5,\cdots,k_n\}$

常见的特殊关键字符有"s_1",s_2 !,s_3",s_4 &,s_5:,s_6;,s_7,〈s_8〉,s_9=,s_{10} ?,s_{11} @等。即

$S=\{s_1,s_2,s_3,s_4,s_5,\cdots,s_n\}$

3　XSS 最优攻击样本的生成

3.1 基于遗传算法的 XSS 攻击特征优化流程

传统的遗传算法仅包含选择、交叉、变异的基本遗传操作,实际应用中由于遗传算法的自身缺陷,它只能找到接近全局最优解,而不能保证收敛到全局最优解,即对局部空间的问题不是很有效,个体多样性减少过快,常常使测试出现早熟收敛和局部收敛性差等问题。故本文对传统遗传算法进行改进,增加精英策略及修补操作,以更全面和高效地生成最优 XSS 攻击个体。本文提出的基于遗传算法的 XSS 攻击特征优化流程如图 3 所示。

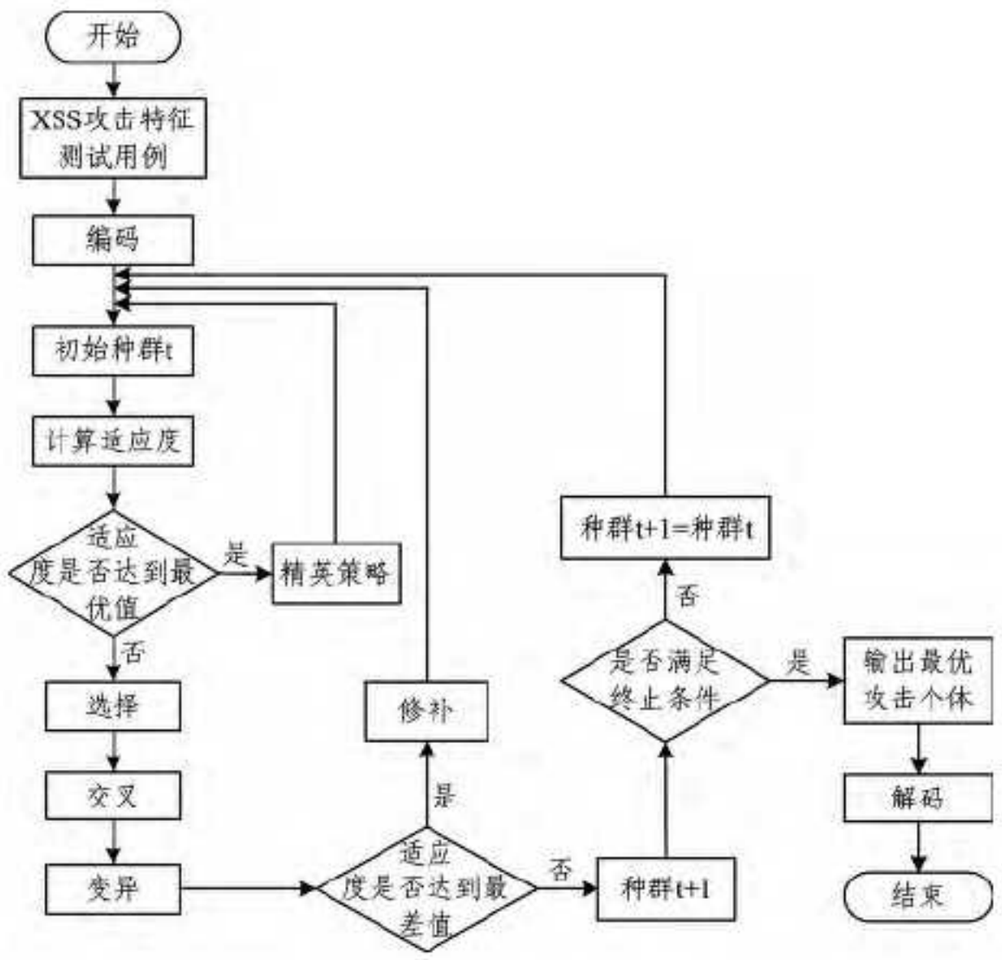

图 3　基于遗传算法的 XSS 攻击特征优化流程图

精英策略即判断在最大迭代次数内特征相关度是否达到极值,若是,则作为精英进行保留。利用交叉和变异产生新一代时,有可能丢失某个中间步骤中得到的最优解。为了防止最优解的丢失,采用精英主义,即在每次产生新一代时,首先把当前最优解原封不动地复制到新一代中,然后再进行交叉变异操作。

修补操作[18]是指判断产生的解是否达到最差适应度值,如果达到,则通过该变异算子基于样本文件中的初始值重新生成一个解,直到生成可行解为止,可行解通过适应度函数来计算,其适应度值通常优于最差适应度值。

算法流程概述:

(1)初始种群生成。读取有效攻击的 XSS 攻击样本集合 1 和集合 2,提取符合规定数量的异常、非法的 XSS 攻击特征样本,并将其进行二进制编码以产生初始种群 $P(t)$。设置进化代数计数器 $t=0$ 以及最大进化代数 T,随机生成 m 个攻击样本作为初始群体 $P(t)$。

(2)遗传操作。以适应度函数作为遗传操作的标准,对初始种群进行选择、交叉、变异、精英策略、修补的遗传操作。

(3)算法结束。若 $t=T$,则以进化过程中所得到的具有最大适应度的个体作为最优攻击测试用例输出,终止计算。当最优个体或最优群体的适应度不再上升,或者最优个体的适应度值达到某个阈值,或者演化代数达到预设值,或者满足其他的终止条件时,算法终止。

3.2 XSS 攻击特征的编码规则

采用二进制编码方式，即将提取的攻击特征信息通过一定的组合和调配构成一个染色体，并进一步转化为二进制（长度为 15 位）形式。XSS 漏洞特征值用四元组表示，将其通用形式设计为 $\{K\}\{S\}\{L\}\{P\}$alert('XSS attack')。

基因包括 15 位，第 1、2 位代表可能在 Web 页面中的位置，00，01，10，11 分别代表 l_1, l_2, l_3, l_4；第 3、4 位代表脚本注入点，00，01，10 代表 p_1, p_2, p_3[11]，编码 11 则代表不使用控件；第 5—9 位代表共计所需的特殊关键字；第 10—15 位代表攻击所需的特殊关键字符。

如编码 110000000000001 代表在〈body〉块中的 Text 控件中使用了"—>"符号作为跨站脚本攻击，即插入"—>〈script〉alert('XSS attack')〈/script〉"代码。编码后所对应的部分代码如表 5 所列。

表 5 XSS 漏洞特征编码表

编码	对应的 XSS 漏洞代码
110000000000001	—>〈script〉alert('XSS attack')〈/script〉
100100000000111	〈TextArea〉〈script〉alert('XSS attack')〈/script〉

3.3 适应度函数的设计

本文采用 XSS 特征相似度计算群体 $P(t)$ 中各个个体的适应度。设 XSS 攻击样本 ε 由两类特征构成：攻击类特征 ε_1 和非攻击类特征 ε_2，且 ε_1 与 ε_2 相互独立，即 $\varepsilon=\varepsilon_1+\varepsilon_2$。而良好的特征应具有稳定性。稳定性指同一类特征应接近。在 XSS 漏洞挖掘中，对每一个疑似攻击的 XSS 代码示例编码后得到攻击特征样本，有效攻击特征样本即攻击特征的相似程度越大越好，而与非攻击特征相似程度越小越好。

本文设 P_a 表示攻击样本的攻击特征相似度，特征组合的 P_a 越大说明该攻击特征组合具有越有效的攻击行为。设存在 m 个攻击样本，而攻击样本共有 n 个攻击特征 $x_1, x_2, \cdots, x_n$，有如下定义：

定义 5（攻击特征相似度） 对攻击样本的攻击特征分类后进行归一化处理和特征划分，得到攻击样本的特征库 Z，计算攻击样本中的每个攻击特征在攻击中出现的概率 $\varphi(x)$，以及该攻击特征在攻击中攻击成功的概率。每个样本攻击特征出现概率之和的平均值即为攻击样本的类内相似度 P_a。

$$P_a=\sum_{i=1}^{n}\varphi(x_i) \tag{5}$$

定义 6（适应度函数） 攻击特征相似度与种群进化数的乘积为适应度函数 D。t 为当前进化代数，T 为进化总代数。

$$D=\sum_{i=1}^{n}\varphi(x_i)\log\frac{t}{T} \tag{6}$$

随着进化代数的增大，个体越来越接近最优解，对被选中个体的要求也越高，适应度函数值也就越大。将初始群体中的各攻击特征组合代入上面的公式中，计算出其适应度函数 D 的值。D 的值越大，说明各攻击特征攻击程度高，即说明该特征组合的攻击效果越好。

3.4 遗传算子的设计

初始群体 $P(t_0)$ 经过 n 次选择、交叉、变异运算之后得到下一代群体 $P(t_n)$。所有遗传算子操作均遵循过滤原则，不可出现需过滤的 XSS 漏洞代码。

3.4.1 选择算子

对初始种群中的测试用例逐个计算特征相关度，并输入到适应度函数中计算适应度，按照适应度大小对当代种群中的测试用例进行排序，采用轮盘赌法，选择 m 个测试用例作为父代种群，每个特征组合被选中的概率与其适应度函数的值成正比。

3.4.2 交叉算子

针对二进制染色体编码的交叉操作，将被选中的两个父代染色体编码进行交叉操作，将两个父代个体的部分结构进行替换重组以生成新的个体，通过交叉，期望将有益的基因组合在一起，具有全局搜索能力。本文采用单点交叉方式，在编码方式的 4 个参数段上，按交叉概率随机选择参数串进行单点交叉，保证交叉点均匀落在每个参数的位串上。以二进制编码为例：

parent1：11↑0000000000001

parent2：10↑0100000000111

交叉得：

child1：100000000000001

child2：110100000000111

3.4.3 变异算子

针对二进制染色体在编码选择操作之后进行变异操作，通过使用基本位变异算子，随机选择特征组合的新个体，将其染色体编码的各个元组元素按变异概率进行变异。二进制表示的基因编码是对基因编码的变异位进行值取反运算，得到变异结果。以二进制编码为例：

child1：100|0|00000000001

所示位变异得：

child3：100100000000001

及

child2：110100000000|1|11

所示位变异得：

child4：110100000000011

4 实验及结论

本文使用真实的互联网数据来验证所提出的 XSS 漏洞挖掘模型，收集常见的 XSS 攻击行为构建行为特征库，自建了一个存在 20 个 XSS 漏洞的 Web 站点，分别采用文献[7]算法和本文算法进行对比实验。表 6 所列为本文算法采用的输入参数。

表 6 本文算法采用的输入参数

参数编号	参数名称	参数大小
1	初始种群规模	100
2	交叉概率	0.5～0.9
3	变异概率	0.005～0.01
4	最大遗传代数	300

XSS 漏洞的挖掘结果如表 7 所列。经过分析，本文提出的方法能成功地检测出 19 个 XSS 漏洞；此外，本文检测出了更多由于遗传算法的 XSS 代码变形优化而造成的 XSS 漏洞。

表 7 测试结果对比

检测方法	XSS 脚本漏洞数量	漏报个数
文献[7]算法	16	4
本文算法	19	1

由此可见，所提出的方法能显著减少漏报和误报，从而更有效地挖掘 Web 应用中的跨站脚本漏洞。但本文结论是建

（下转第 364 页）

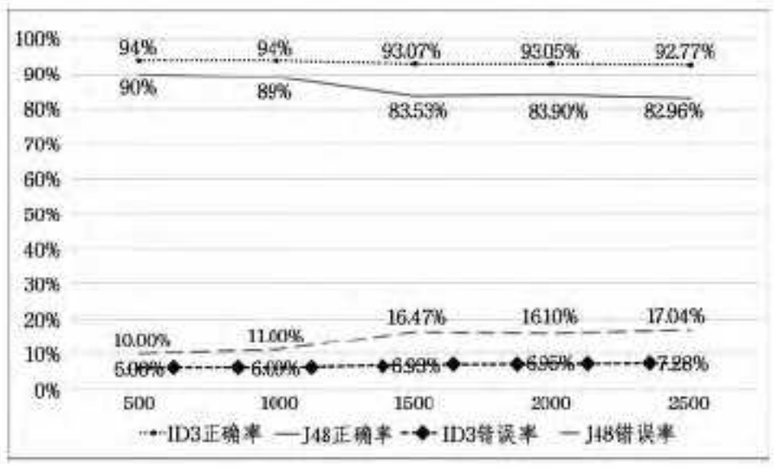

图 4　分类结果正确率与错误率折线图

如图 4 所示，以上算法均能在监测规模扩大时将监测的正确率保持在一个较高的范围内，而 ID3 算法更适用于本系统，准确率更高，同时稳定性也更好。

结束语　针对手机中的应用软件产生的异常流量进行监测十分不便的问题，本文在手机所连接的无线网络出口处使用非侵入式进行流量提取并建立决策树模型进行分类。通过实验结果可以看出，本文所研究的系统对手机异常流量具有较高的监测率，这样为推断手机中的不良软件提供了一种新的方法，可以提高连接在本系统的数台手机的安全性，尤其是在多部设备产生巨大规模流量和数据记录时，本文的研究内容在处理效率和准确率上就更显优势。

参 考 文 献

[1] 文伟平，梅瑞，宁戈，等. Android 恶意软件检测技术分析和应用研究[J]. 通信学报，2014，35(8)：78-85

[2] 杨欢，张玉清，胡予濮，等. 基于多类特征的 Android 应用恶意行为检测系统[J]. 计算机学报，2014，37(01)：15-27

[3] 周裕娟，张红梅，张向利，等. 基于 Android 权限信息的恶意软件检测[J]. 计算机应用研究，2015，32(10)

[4] Carela-Español V，Barlet-Ros P，Bifet A，et al. A streaming flow-based technique for traffic classification applied to 12＋1 years of Internet traffic[J]. Telecommunication Systems，2015：1-14

[5] Divakaran D M，Su L，Liau Y S，et al. SLIC：Self-Learning Intelligent Classifier for Network Traffic[J]. Computer Networks，2015，91：283-297

[6] Chen Z，Liu Z，Peng L，et al. A novel semi-supervised learning method for Internet application identification[J]. Soft Computing，2015：1-13

[7] Groleat T，Arzel M，Vaton S. Stretching the Edges of SVM Traffic Classification With FPGA Acceleration[J]. IEEE Transactions on Network & Service Management，2014，11(3)：278-291

[8] Tegeler F，Fu X，Vigna G，et al. BotFinder：finding bots in network traffic without deep packet inspection[C]// Proc Co-next. 2012：349-360

[9] Wang B，Chua K C，Srinivasan V，et al. Information Coverage in Randomly Deployed Wireless Sensor Networks[J]. IEEE Transactions on Wireless Communications，2007，6(8)：2994-3004

[10] Narayanan V A，Sureshkumar V，Rajeswari A. Automatic Traffic Classification Using Machine Learning Algorithm for Policy-Based Routing in UMTS-WLAN Interworking[M]// Artificial Intelligence and Evolutionary Algorithms in Engineering Systems. Springer India，2015：305-312

(上接第 331 页)

立在实验的基础上的，在实际应用中需要进行改进和完善。

结束语　通过对 Web 应用跨站脚本漏洞进行分析，提出了结合模糊测试和遗传算法的 Web 应用跨站脚本漏洞挖掘方法，以优化 XSS 漏洞特征测试用例的生成方法及动态挖掘流程。在该方法中，通过定义漏洞库模糊测试方法、过滤规则和补全规则进行 XSS 漏洞特征选择与提取，并采用遗传变异规则来搜索特征空间，对 XSS 特征测试用例进行过滤、补全、选择、交叉、变异操作，通过多次反复迭代选出最优的 XSS 攻击特征测试用例，由此实现 Web 应用跨站脚本漏洞的挖掘。分析表明，该方法能有效挖掘 Web 应用中跨站脚本漏洞。

但是，本文目前针对 XSS 漏洞库的构建主要是人工输入，且 XSS 漏洞遗传变异只考虑了一维变异，故如何实现 XSS 漏洞库的自动输入及如何构建 XSS 漏洞特征测试用例的多维变异[18]是下一步的工作重点。

参 考 文 献

[1] 中国互联网络信息中心(CNNIC). 第 27 次中国互联网络发展状况统计报告(2015-2)[R/OL]. http://www.cnnic.cn/hlwfzyj/hlwxzbg/201502/P020150203551802054676.pdf

[2] OWASP. OWASP Top Ten Project[R]. 2013

[3] Stuttard D，Pinto M，石华耀. 黑客攻防技术宝典：Web 实战篇[M]. 2009

[4] 刘为. 基于模糊测试的 XSS 漏洞检测系统研究与实现[D]. 长沙：湖南大学，2010

[5] 温凯，郭帆，余敏. 自适应的 Web 攻击异常检测方法[J]. 计算机应用，2012，32(7)：2003-2006

[6] 吴子敬，张宪忠，管磊，等. 基于反过滤规则集和自动爬虫的 XSS 漏洞深度挖掘技术[J]. 北京理工大学学报，2012，32(4)：396-396

[7] 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(B6)：51-53

[8] 吴少华，程书宝，胡勇. 基于 SVM 的 Web 攻击检测技术[J]. 计算机科学，2015，42(S1)：362-364

[9] Shahriar H，Zulkernine M. S2XS2：a server side approach to automatically detect XSS attacks[C]// 2011 Ninth IEEE International Conference on Dependable，Autonomic and Secure Computing. Sydney：IEEE Press，2011：7-14

[10] 王春东，邱晓华. 基于特征策略的 XSS 漏洞检测技术研究[J]. 天津理工大学学报，2013，29(5)：25-29

[11] 许静，练坤梅，田伟，等. 应用遗传算法自动生成 XSS 跨站点脚本漏洞检测参数的方法[P]. 中国，2015-05-30

[12] 朱红萍，巩青歌，雷战波. 基于遗传算法的入侵检测特征选择[J]. 计算机应用研究，2012，29(4)：1417-1419

[13] 郭慧，王晓菊，刘明艳，等. 基于遗传算法的入侵检测系统特征选择方法研究[J]. 华北科技学院学报，2014，11(9)：68-72

[14] 韦存堂. SQL 注入与 XSS 攻击自动化检测关键技术研究[D]. 北京：北京邮电大学，2015

[15] 牛皓. 基于网络爬虫的 XSS 漏洞检测系统的研究与设计[D]. 北京：北京邮电大学，2015

[16] https://www.owasp.org/index.php/XSS—Filter—Evasion—Cheat—Sheet

[17] 潘古兵. Web 应用程序渗透测试方法研究[D]. 重庆：西南大学，2012

[18] 吴志勇，王红川，孙乐昌，等. 遗传算法在多维 Fuzzing 技术中的应用[J]. 小型微型计算机系统，2011，32(5)：998-1004

[19] 王云. 基于爬虫和模糊测试的 XSS 漏洞检测工具设计与实现[D]. 广州：华南理工大学，2015