一种基于软集和多属性综合的软件漏洞发现方法

唐成华[1,2] 田吉龙[2] 王 璐[2] 王丽娜[2] 强保华[1,2]
(桂林电子科技大学广西信息科学实验中心 桂林 541004)[1]
(桂林电子科技大学计算机科学与工程学院 桂林 541004)[2]

摘 要 针对软件漏洞检测中的漏洞覆盖率和人工缺陷审查等问题,提出了一种基于软集和多属性综合的软件漏洞发现方法。首先基于多检测工具的可信集成,建立了软件漏洞影响的评估模型;其次引入软集实现漏洞影响因素的度量,接着通过多属性综合的集成工具确定漏洞对软件安全的严重性影响,并最终完成软件漏洞的发现过程。实验结果表明,该方法对不同级别的漏洞均有较好的检测能力,为改善软件漏洞检测的误报率和漏报率等问题提供了一种可行的途径。

关键词 软件漏洞,软集,属性集,漏报率,误报率
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.5.037

Method for Software Vulnerability Discovery Based on Soft Set and Multi-attribute Comprehensiveness

TANG Cheng-hua[1,2] TIAN Ji-long[2] WANG Lu[2] WANG Li-na[2] QIANG Bao-hua[1,2]
(Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, China)[1]
(School of Computer Science & Engineering, Guilin University of Electronic Technology, Guilin 541004, China)[2]

Abstract Aiming at the problem of the vulnerability coverage and artificial defect review in the software vulnerability detection, a method for software vulnerability discovery based on the soft set and multi-attribute comprehensiveness was proposed. Firstly, based on trusted integrated detection tools, an evaluation model of software vulnerability factors was established. Secondly, the soft set was introduced to measure vulnerability factors, then the serious impact on software security was determined through the method of multi-attribute comprehensive integration tools, and the discovery process of software vulnerability was finally completed. Experimental results show that the method has better detection capabilities for vulnerability in different level, which provides a feasible way for the improvement of software vulnerability detection false positive rate and false negative rate.

Keywords Software vulnerability, Soft set, Attribute set, False negative rate, False positive rate

软件漏洞是信息安全问题的根源之一,漏洞存在的普遍性和危害性,已促使人们关注漏洞的分析、检测、利用和消控等技术的研究,其中漏洞检测技术根据在检测过程中是否需要运行软件程序可分为静态检测和动态检测两种。由于测试用例对软件逻辑的覆盖率问题,限制了动态检测方法对漏洞的发掘能力[1],很难发现隐蔽的安全漏洞;而静态检测是通过对软件代码进行包括词法分析、数据流分析、符号执行、定理证明等方法找到涉及安全的软件设计错误和编码缺陷等,尽管需要较多的人工参与缺陷审查,但具有更有效的深度漏洞检测能力[2]。

Williams等人[3]采用人工检测方法从软件漏洞数据库和源代码中鉴别漏洞类型,然后建立一个漏洞检测器,能自动从源代码中挖掘出漏洞的危险等级。Nahid等人[4]提出一种冷测试方法,通过建立安全目标模型来跟踪检测程序运行中的引起漏洞的原因,实现一种模型依赖的过程,然后通过漏洞测试条件来找出漏洞,此方法需要专门的语言做支撑,并提供全面的漏洞测试条件来进行检测,很难实际操作。

为了提高软件漏洞检测的效果,Ren等人[5]提出了基于序列聚类技术和模型分析的软件漏洞检测方法(MVCMA),改进k-means算法使其能够用于漏洞序列的挖掘;并通过计算疑是漏洞序列与VPL模式的相似度来设计漏洞检测方法,以降低软件漏洞检测的误报率和漏报率,不过,所使用的改进k-means算法仅能发现球形分布的聚类,不能发现任意形状的聚类,影响了漏洞检测的准确性。Zhang等人[6]将静态分析方法与动态污点分析方法相结合,提出了在SDCF框架中用于发现软件中可能潜在的威胁,但漏洞信息与特征代码匹配的过程回避了漏洞之间的影响能力。

为了提高测试准确度并降低测试成本,孔德光等人[7]采

到稿日期:2014-06-20 返修日期:2014-08-18 本文受国家自然科学基金(61462020,61363006,61163057),广西信息科学实验中心基金(20130329),广西自然科学基金(2014GXNSFAA118375),桂林电子科技大学研究生教育创新计划项目(2013110124)资助。
唐成华(1974-),男,博士后,副教授,硕士生导师,CCF会员,主要研究方向为网络与信息安全,E-mail:tch@guet.edu.cn;田吉龙(1989-),男,硕士生,主要研究方向为网络信息安全;王 璐(1991-),女,硕士生,主要研究方向为网络信息安全;王丽娜(1987-),女,硕士生,主要研究方向为网络信息安全;强保华(1972-),男,博士后,教授,主要研究方向为智能信息处理。

用多个静态测试工具，并对检测工具的结果进行数据融合分析，通过对漏洞的估分值大小找出漏洞。在针对检测结果的数据分析上，一般是通过基于漏洞量化评估[8]、漏洞关联的风险评估[9]等方法来完成漏洞判别。李珍等人[10]则引入区间数据点的检查点属性分级策略和属性阈值的概念，提出适合软件行为可信评估的多属性权重确定方法，同时也说明了在不考虑检查点的粒度设置下，根据决策者经验对属性进行分级的过程缺乏现实中的约束关系。以上这些漏洞数据的处理方法缺少考虑漏洞本身对软件自身的安全性影响，只是通过简单地对漏洞评分估值来判断，缺乏相关因素的综合评判。

在分析以上检测方法的基础上，本文提出一种软件漏洞发现方法，集成多个检测工具，对检测结果进行多级数据综合处理，通过在建立软件漏洞影响评估模型并解决检测工具的可信度问题的基础上，引入软集实现漏洞影响因素的度量，基于多属性综合的方法确定漏洞对软件安全的影响程度，并最终完成软件漏洞的判别过程。

1 基本概念及相关原理

集成多静态检测工具的漏洞检测，能充分利用各自的检测能力，结合相关数据分析和检测结果评估方法，以提高对软件漏洞发现的覆盖率和准确度，并降低误报率和漏报率。

定义 1 设被检测软件的真实漏洞数为 VL，若利用 n 个中的第 i 个静态检测工具检测出有 $AL(i)$ 个漏洞，其中真实漏洞数为 $T(i)$，则工具 i 的误报率 $FP(i)$ 和漏报率 $MP(i)$ 为：

$$FP(i)=\frac{AL(i)-T(i)}{AL(i)} \tag{1}$$

$$MP(i)=\frac{VL-T(i)}{VL} \tag{2}$$

定义 2 设 $FP(i)$ 和 $MP(i)$ 分别是静态检测工具 i 的误报率和漏报率，构建一个二维向量 $(FP(i),MP(i))$，利用欧式测度作为判定检测性能的主要标准，那么工具 i 在进行软件漏洞检测时的可信度可表示为：

$$k(i)=\frac{\sum_{j=1}^{m}k^j(i)}{\sum_{i=1}^{n}\sum_{j=1}^{m}k^j(i)} \tag{3}$$

式中，n 表示检测工具的个数，m 表示被检测软件的个数，$k^j(i)$ 是检测工具 i 对测试软件 j 的可信度，表示为：

$$k^j(i)=\frac{\sqrt{(FP(i)-1)^2+(MP(i)-1)^2}}{\sqrt{2}} \tag{4}$$

对不同工具的检测结果进行分析，首先将检测结果数据进行归一化处理，然后根据所定义的各漏洞对应的估分随机变量，对漏洞进行真实性评估。由于多个检测工具识别出的漏洞比单个工具识别出的更有可能是漏洞，因此在确定各漏洞对应的估分随机变量时有以下准则：①被多个工具检测出的漏洞相应权值增加；②对软件自身安全影响严重的漏洞相应权值增加；③出现次数少的漏洞相应权重降低。

定义 3 设有 n 个漏洞，$c(j)$ 是第 j 个漏洞对软件自身安全影响程度的估分变量 $(1\leqslant j\leqslant n)$；$w(i)$ 是第 i 个工具对漏洞的估分变量 $(1\leqslant i\leqslant n)$；$k(i)$ 是第 i 个工具的可信度 $(1\leqslant i\leqslant n)$；$vul(j)$ 表示第 j 个漏洞的最后估分 $(1\leqslant j\leqslant n)$；$ck(i,j)>0$ 表示第 i 个工具检测出第 j 个漏洞 $(1\leqslant i\leqslant n,1\leqslant j\leqslant n)$，它们满足：

$$\sum_{i=1}^{n}k(i)=1 \tag{5}$$

$$vul(j)=\begin{cases}c(j)\sum_{i=1}^{n}w(i), & ck(i,j)>0\\ 0, & ck(i,j)\leqslant 0\end{cases} \tag{6}$$

$$E(X)=\sum_{i=1,j=1}^{n}vul(j)k(i) \tag{7}$$

式(7)中，$E(X)$ 表示各漏洞的估分随机变量 X 的期望，它反映了该软件漏洞的真实性。

采用基于软集和多属性综合的方法求出估分变量 $c(j)$ 等，最后得出各漏洞的期望值 $E(X)$。通过与预先设置的阈值 λ 进行比较来判别漏洞的存在性，并要求在漏洞检测过程中可以通过检测效果的反馈实时调节 λ 值。

2 软件漏洞发现方法

2.1 构造软件漏洞影响评估模型

一个软件漏洞可能会被利用而造成非授权访问、远程后门攻击、非法盗取敏感信息等非法操作，这些非法操作的可能性及其对软件安全的影响效果与软件漏洞的严重性息息相关。根据专业知识，软件安全漏洞的严重性体现在漏洞对软件的直接危险程度、由于漏洞而造成风险的概率、利用漏洞对软件功能或服务的影响程度以及漏洞被修补的难易程度等方面，如图 1 所示。

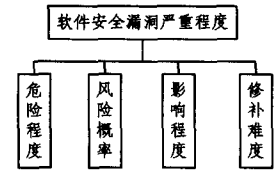

图 1 软件安全漏洞严重性评估模型

2.2 引入软集实现漏洞影响因素的度量

作为研究不确定性问题的重要方法，模糊集侧重于程度化思想，而软集强调参数化方法。尽管模糊集属于软集的一种特例，但软集的运算更加广泛。与模糊集相比，软集有如下优点：①在软集中，不需计算隶属度，因此更方便实用；②软集的代数结构更为灵活多样；③软集可克服模糊集指标化工具的不充分性缺点。本文采用软集理论计算漏洞影响因素的综合度量值。

定义 4 设 V^y 是第 y 个因素的指标集，当 G 是指标 V^y 到对象集 R 上的所有子集的映射时，称 (G,V^y) 是 R 上的软集。

定义 5 定义在对象集 R 上的软集 (G,V^y)，对象 $x_i,x_j\in R$，定义集合 $\{x_j\mid g_{V^y}(x_i)\leqslant g_{V^y}(x_j)\}$ 是对象 x_i 的偏好集合，记为 $[x_i]_{V^y}^{\geqslant}$，偏好集合中的每一个对象第 y 个因素指标的评估值之和不小于对象 x_i 第 y 个因素指标的评估值之和。

由定义 5 可知，如果对象 x_i 的偏好集合包含 x_j，那么对象 x_j 比 x_i 具备更多的特征。

定义 6 定义在对象集 R 上的软集 (G,V^y)，对指标 $v_i^y\in V^y$，指标 v_i^y 的重要度是：

$$\omega(v_i^y)=1-\frac{\sum_{x_i\in R}\frac{[x_i]_{V^y}^{\geqslant}\cap[x_i]_{V^y}^{\geqslant}}{v_i^y}}{|R|} \tag{8}$$

式中，$[x_i]_{V^y}^{\geqslant}$ 是对象 x_i 关于指标和的偏好集。

定义 7 定义在对象集 R 上的软集(G,V)，对象 $x_i(x_i\in R)$，指标 $v_j^y\in V^y$，对象 x_i 的第 y 个因素的最后综合度量值 $S(x_i^y)$可表示为：

$$S(x_i^y)=\sum_{j=1}^{n}\prod_{j=1}^{n}\omega(v_j^y)\mu(x_iv_j^y) \tag{9}$$

式中，$\mu(x_iv_i^y)$是对象 x_i 第 j 个指标的值。

构造漏洞集合 $R=\{u_1,u_2,\cdots,u_n\}$，第 y 个因素评价指标集合 $V^y=\{v_1^y,v_2^y,\cdots,v_5^y\}$，其中，评价指标 $v_1^y,v_2^y,\cdots,v_5^y$ 代表较高、高、中、低和较低 5 个级别，y 可取 1、2、3、4，代表漏洞的危险程度、漏洞的风险概率、漏洞的影响程度和漏洞的修补难度。根据专家经验和相关统计数据，通过公式计算出 4 个漏洞影响因素的综合度量值。

2.3 基于多属性综合的漏洞对软件安全的影响

基于多属性综合的软件漏洞的影响计算步骤如下：

步骤 1 构造软件漏洞分类标准矩阵。

设 $A=\{a_1,a_2,\cdots,a_n\}$为漏洞集合，$Y=\{Y_1,Y_2,\cdots,Y_n\}$为指标集，其中 a_i 的特性由 m 个指标的测量值反映，因此可将 a_i 表示为 m 维向量，即 $a_i=(a_{i1},a_{i2},\cdots,a_{im})$。假设 A 中的元素有 K 个评价类 $C_1,C_2,\cdots,C_k$，这些评价类构成了 A 上的属性空间 F 的有序分割类，构成的分类标准矩阵为：

$$B=(b_{jl})_{m\times k}=\begin{array}{c} \\ Y_1\\ Y_2\\ Y_3\\ Y_4\end{array}\begin{array}{c}\begin{array}{cccc}C_1 & C_2 & C_3 & C_k\end{array}\\ \begin{bmatrix}b_{11} & b_{12} & \cdots & b_{1k}\\ b_{21} & b_{22} & \cdots & b_{2k}\\ \vdots & \vdots & \vdots & \vdots\\ b_{m1} & b_{m2} & \cdots & b_{mk}\end{bmatrix}\end{array} \tag{10}$$

式中，b_{jk} 满足 $b_{j1}<b_{j2}<\cdots<b_{jk}$ 或者 $b_{j1}>b_{j2}>\cdots>b_{jk}$。

步骤 2 计算多指标综合属性测度。

首先完成单指标属性测度计算。考虑对象 a_i 的第 j 个指标值 a_{ij} 具有 C_k 的属性测度 $\mu_{ijk}=\mu(a_{ij}\in C_k)$，假定 $b_{j1}<b_{j2}<\cdots<b_{jk}$，则有：

当 $a_{ij}\leqslant b_{j1}$ 时，取 $\mu_{ij1}=1,\mu_{ij2}=\cdots=\mu_{ijk}=0$；

当 $a_{ij}\leqslant b_{ik}$ 时，取 $\mu_{ijk}=1,\mu_{ij1}=\cdots=\mu_{ijk-1}=0$；

当 $b_{ij}\leqslant a_{ij}\leqslant b_{i\,l+1}$ 时，取 $\mu_{ijk}=0,k<l$ 或 $k>l+1$，且有：

$$\mu_{ijl}=\left|\frac{a_{ij}-b_{jl+1}}{b_{jl}-b_{jl+1}}\right| \tag{11}$$

$$\mu_{ijl+1}=\frac{|a_{ij}-b_{jl}|}{|b_{jl}-b_{jl+1}|} \tag{12}$$

于是，由属性测度函数和分类标准矩阵得到评估系统的属性测度分布矩阵 W_i 为：

$$W_i(\mu_{ijk})_{m\times k}=\begin{bmatrix}\mu_{i11} & \mu_{i12} & \cdots & \mu_{i1k}\\ \mu_{i21} & \mu_{i22} & \cdots & \mu_{i2k}\\ \vdots & \vdots & \vdots & \vdots\\ \mu_{im1} & \mu_{im2} & \cdots & \mu_{imk}\end{bmatrix} \tag{13}$$

在得到第 i 个研究对象单个指标的属性测度 μ_{ijk} 之后，再经加权求和计算，得到多指标综合属性测度 μ_{ik}：

$$\mu_{ik}=\sum_{j=1}^{j=km}\omega_j\mu_{ik} \tag{14}$$

式中，$1\leqslant k\leqslant K,\omega_j$ 是第 j 个指标 Y_j 的权值，$\omega_j\geqslant 0,\sum_{j=1}^{m}\omega_j=1$。

为了降低主观性影响，并能真实反映各因素的重要性，确定采用以下熵权系数法计算 ω_j 值。

定义 8 设有 m 个评价指标，n 个方案，Z_{jk} 是方案 K 的评价指标 j 的样本值，Z_j^* 是评价指标 j 的理想值，那么 Z_{jk} 和 Z_j^* 接近度可用如下公式表示：

$$l_{jk}=\frac{Z_{jk}}{Z_j^*} \tag{15}$$

式中，Z_j^* 取 Z_{jk} 中的最大值。

由满足以下条件的熵来度量评价指标 j 对方案决策的相对重要性的不确定性：

$$l_{jk}=\sum_{k=1}^{n}l_{jk} \tag{16}$$

$$P=-\sum_{j=1}^{n}\frac{l_{jk}}{l_j}\ln(\ln\frac{l_{jk}}{l_j}) \tag{17}$$

对熵进行归一化，则指标 j 的决策权重的熵可表示为：

$$P(l_j)=-\frac{1}{\ln(\ln n)}\sum_{k=1}^{n}\frac{l_{jk}}{l_j}\ln\frac{l_{jk}}{l_j} \tag{18}$$

那么评价指标 j 的权值可以表示为：

$$\omega_j=\frac{1}{n-\sum_{j=1}^{m}P(l_j)}[1-P(l_j)] \tag{19}$$

步骤 3 综合计算出漏洞对软件的严重性程度。

采用最小代价准则作为识别准则。设 $t_i\in T$，假定 t_i 属于 C_1 而判别为 C_k 的代价为 h_{ik}，则对象 t_i 判别为 C_k 的全部代价可表示为：

$$\alpha_{ik}=\sum_{l=1}^{k}h_{ik}\mu_{il} \tag{20}$$

如果有：

$$\alpha_{ik_0}=\min_{1\leqslant k\leqslant K}\alpha_{ik} \tag{21}$$

则认为 t_i 属于 C_{k0} 类。

因此，第 i 个漏洞对软件的安全影响程度 $c(i)$ 为：

$$c(i)=\frac{k_0}{\alpha_{ik_0}+\varphi} \tag{22}$$

式中，φ 为一个无限小的正数，防止分母为零。

在漏洞发现过程中，为了计算漏洞的期望值，首先计算出检测工具的可信度；接着在处理工具检测结果时，采用软集计算出漏洞的风险因素的综合度量值，建立漏洞数据矩阵；然后引入属性集建立漏洞分类标准矩阵，并结合漏洞数据矩阵，求出漏洞对软件的影响程度；最后通过式(7)得出漏洞的期望值，进而可判断该漏洞是否存在。

3 实验过程与分析

3.1 软件漏洞检测工具及其可信度

选择 3 种目前流行的静态检测工具 Flawfinder、Cppcheck 和 Rats 作为测试工具，以 Squid、Zabbix、Lighttpd、Freetype 和 Thttpd 等软件作为测试用例，可以在 CVE 等漏洞库中找到测试用例的真实漏洞信息。根据式(1)和式(2)得到不同检测工具的误报率与漏报率如表 1 所列。

表 1 3 种检测工具的误报率与漏报率(%)

	误报率			漏报率		
	Flawfinder	Cppcheck	Rats	Flawfinder	Cppcheck	Rats
Squid	91.12	92.29	94.24	31.10	38.10	39.79
Zabbix	90.20	91.72	93.58	28.21	36.48	37.86
Lighttpd	89.31	91.78	93.12	28.32	36.34	37.51
Freetype	91.30	92.32	94.18	29.79	37.89	38.69
Thttpd	89.75	91.26	93.01	27.37	35.97	36.95

根据式(3)可求得 Flawfinder、Cppcheck 和 Rats 的可信度分别为 $k(1)=0.3631$、$k(2)=0.322$ 和 $k(3)=0.3149$。

3.2 实验检测过程

3.2.1 采用软集计算漏洞影响因素的综合度量值

以软件源码包 Lighttpd 中的 3 条漏洞(分别表示为 x_1、

x_2、x_3)为例。其中一条漏洞被3个不同工具解析的结果如图2和图3所示(注:cppcheck没有发现此漏洞)。

lighttpd-1.4.32/src/mod_cgi.c:944:[3](buffer)getenv:Environment variables are untrustable input if they can be set by an attacker. They can have any content and length, and the same variable can be set more than once. Check environment variables carefully before using them.

图2 Flawfinder解析结果

lighttpd-1.4.32/lighttpd-1.4.32/src/mod_cgi.c:944:High:getenv if(NULL!=(s=getenv("LD_LIBRARY_PATH")))
Environment variables are highly untrustable input. They may be of any length, and contain and data. Do not make any assumptions regarding content or length. If at all possible avoid using them, and if it is necessary, sanitize them and truncate them.

图3 Rats解析结果

Flawfinder给出的漏洞等级分为5类:1、2、3、4、5。Rats给出的漏洞危险等级分为:high、medium、default。按5分制评分,因此Rats解析结果的漏洞危险等级映射为:f(high)=5,f(medium)=3,f(default)=1。根据定义(4),建立漏洞危险程度的软集,如表2所列。

表2 软集(G,V^1)

R	v_1^1	v_2^1	v_3^1	v_4^1	v_5^1
X_1	3	2	4	2	1
X_2	4	3	1	2	1
X_3	5	4	1	1	1

要计算指标 v_1^1 的权重,需要先将指标 v_1^1 去掉后得到软集$(G,V^1/v_1^1)$,如表3表列。

表3 软集$(G,V^1/v_1^1)$

R	v_2^1	v_3^1	v_4^1	v_5^1
X_1	2	4	2	1
X_2	3	1	2	1
X_3	4	1	1	1

由式(8),指标 v_1^1 的重要度为:

$$\omega(v_1^1)=1-\frac{1+1+2/3}{3}=0.11 \tag{23}$$

同理可依次求得其余指标的重要度。根据式(9)求出 x_1 漏洞危险程度的综合度量值为 $S(x_1^1)=2.85$,进而可得 x_1 漏洞其余影响因素的综合度量值。表4为计算获得的3个漏洞影响因素的综合度量值。

表4 漏洞影响因素的综合度量值

Vul	V^1	V^2	V^3	V^4
X_1	2.85	2.34	2.54	2.98
X_2	2.54	2.76	2.97	2.61
X_3	2.68	2.96	2.71	2.46

3.2.2 计算漏洞对软件的影响程度

由于漏洞对软件的严重性程度分为5个等级:C_1={较低}、C_2={低}、C_3={中}、C_4={高}和C_5={较高},$C_1<C_2<C_3<C_4<C_5$,并且有Y={Y_1,Y_2,Y_3,Y_4}={危险程度,风险概率,影响程度,修补难度},为便于计算,建立文字描述与数字映射关系,如f(较高)=2.95、f(高)=2.85、f(中)=2.75、f(低)=2.65、f(较低)=2.55,这样可建立分类标准矩阵:

$$B=\begin{array}{c} \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \begin{bmatrix} 2.55 & 2.65 & 2.75 & 2.85 & 2.95 \\ 2.55 & 2.65 & 2.75 & 2.85 & 2.95 \\ 2.55 & 2.65 & 2.75 & 2.85 & 2.95 \\ 2.55 & 2.65 & 2.75 & 2.85 & 2.95 \end{bmatrix} \end{array} \tag{24}$$

由表4可得漏洞数据矩阵:

$$\begin{array}{c} \begin{array}{cccc} Y_1 & Y_2 & Y_3 & Y_4 \end{array} \\ \begin{bmatrix} 2.85 & 2.34 & 2.54 & 2.98 \\ 2.54 & 2.76 & 2.97 & 2.61 \\ 2.68 & 2.96 & 2.71 & 2.46 \end{bmatrix}\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \end{array} \tag{25}$$

由式(13)求得漏洞 x_1 的危险程度、风险概率、影响程度、修补难度的属性测度分布矩阵为:

$$W_1=\begin{bmatrix} \mu_{111} & \mu_{112} & \mu_{113} & \mu_{114} & \mu_{115} \\ \mu_{121} & \mu_{122} & \mu_{123} & \mu_{124} & \mu_{125} \\ \mu_{131} & \mu_{132} & \mu_{133} & \mu_{134} & \mu_{135} \\ \mu_{141} & \mu_{142} & \mu_{143} & \mu_{144} & \mu_{145} \end{bmatrix}=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

同理,可以求出其余漏洞的属性测度分布矩阵。

根据式(15)—式(17)计算各指标的接近度 l_{ik}、$p(l_i)$,如表5所列。

表5 评价指标熵值的计算结果

	X_1	X_2	X_3	l_{ik}	$p(l_i)$
Y_1	1	0.891	0.940	2.831	0.999
Y_2	0.790	0.932	1	2.722	0.995
Y_3	0.855	1	0.912	2.767	0.998
Y_4	1	0.876	0.826	2.702	0.996

对于指标 Y_1,根据式(18)得到:

$$P(l_1)=-\frac{1}{\ln(\ln 3)}\left[\frac{1}{2.831}\ln\left(\ln\frac{1}{2.831}\right)+\frac{0.891}{2.831}\ln\left(\ln\frac{0.891}{2.831}\right)+\frac{0.940}{2.831}\ln\left(\ln\frac{0.940}{2.831}\right)\right]=0.999$$

同理,可以求出其余各指标的熵值。根据式(19),求得 $\omega_1=0.083$。同理,可以求出其余各指标的权值:$\omega_2=0.417$,$\omega_3=0.167$,$\omega_4=0.333$。最后,根据式(14),得到多指标属性测度值:

$$(\mu_{11},\mu_{12},\mu_{13},\mu_{14},\mu_{15})=(0.584,0,0,0.083,0.333)$$

由于危害性高的漏洞被误判为危害性小的漏洞所造成的后果往往比相反误判的危害性要大,因此通过定义代价矩阵,得到漏洞 x_1 的全部代价为:

$$(\alpha_{11},\alpha_{12},\alpha_{13},\alpha_{14},\alpha_{15})=(1.997,2.165,2.332,2.338,2.419)$$

根据式(22),求出漏洞 x_1 对软件的严重性 $c(1)=0.501$。同理,可求出其余漏洞对软件的严重性评估值。

3.2.3 漏洞判别

根据式(7)计算出漏洞 x_1 的期望 E(X)=0.1819,并且能得出其他所有漏洞的期望。若按照给定的可反馈调节的阈值 λ 为0.22,则认为该漏洞 x_1 是存在的,并被发现。

3.3 实验结果分析

本文设计并开发了一个基于软集和属性综合的软件漏洞原型检测系统(VDPS),其界面如图 4 所示,能实现 3 种检测工具的单独检测结果与综合检测效果在误报率和漏报率上的对比。

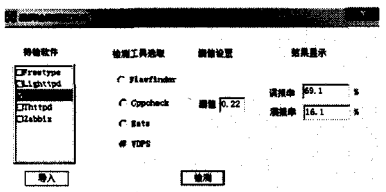

图 4 漏洞检测原型系统

由于实验所对比的检测工具软件都是已知真实漏洞个数,为统计方便,其检测结果直接计算为误报率和漏报率。VDPS 的误报率和漏报率实验数据如表 6 所列。

表 6 VDKS 检测软件的误报率和漏报率(%)

	Squid	Zabbix	Lighttpd	Freetype	Thttpd
FP	69.1	72.3	68.4	70.5	69.1
MP	16.1	18.4	17.9	16.4	17.1

图 5 和图 6 分别为 VDPS 与 3 种单独检测工具的误报率和漏报率的比较。

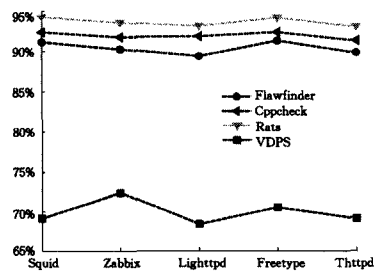

图 5 VDPS 与其他漏洞检测工具的误报率

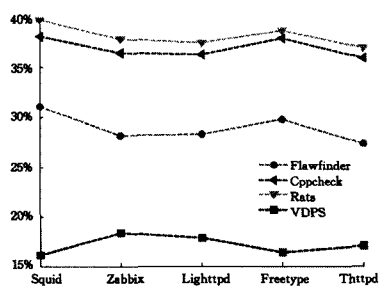

图 6 VDPS 与其他漏洞检测工具的漏报率

从实验结果可以看出,VDPS 的误报率和漏报率均有明显降低,对多种工具的检测结果进行基于软集和属性综合的数据分析,并在进行交互和反馈中调节阈值,提高了发现软件漏洞的范围和准确度,从而减少人工参与漏洞审查的工作量,并降低了漏洞误报率和漏报率。

结束语 本文重点在于利用多工具对软件漏洞检测的数据进行分析从而发现漏洞的过程。通过在建立软件漏洞影响评估模型并解决检测工具的可信度问题的基础上,引入软集理论获得漏洞影响因素的度量,结合多检测工具的方法优点以确定基于多属性综合的漏洞对软件安全的严重影响,为解决软件漏洞检测与发现提供了一种可行的思路,实验表明该方法是有效的,但不足之处在于是基于其他检测工具的实现,漏洞特征较有限,因此,提高多检测工具的智能综合能力,并改进相关漏洞知识和识别规则是下一步的主要研究方向。

参 考 文 献

[1] 陈平,韩浩,沈晓斌,等. 基于动静态程序分析的整形漏洞检测工具[J]. 电子学报,2010,38(8):1741-1747

[2] 张大林,金大海,宫云战,等. 基于缺陷关联的静态分析优化[J]. 软件学报,2014,25(2):386-399

[3] Williams C C, Hollingsworth J K. Automatic mining of source code repositories to improve bug finding techniques[J]. Journal of IEEE Transactions on Software Engineering, 2005, 31(6): 466-480

[4] Nahid S, Amel M, Edgardo M de O, et al. An advanced approach for modeling and detecting software vulnerabilities[J]. Information and Software Technology, 2012, 54(9): 997-1013

[5] Ren J D, Cai B L, He H T, et al. A method for detecting software vulnerabilities based on clustering and model analyzing[J]. Journal of Computational Information Systems, 2011, 7(4): 1065-1073

[6] Zhang Ruo-yu, Huang Shi-qiu, Qi Zheng-wei, et al. Static program analysis assisted dynamic taint tracking for software vulnerability discovery[J]. Computers & Mathematics with Applications, 2012, 63(2): 469-480

[7] 孔德光,郑烃,陈超,等. 基于数据融合的源代码静态分析漏洞检测技术[J]. 小型微型计算机系统,2008,29(6):1109-1112

[8] 李鑫,李京春,郑雪峰,等. 一种基于层次分析法的信息系统漏洞量化评估方法[J]. 计算机科学,2012,39(7):58-63

[9] 周亮,李俊娥,陆天波,等. 信息系统漏洞风险定量评估模型研究[J]. 通信学报,2009,30(2):71-76

[10] 李珍,田俊峰,杨晓晖. 基于检查点分级属性的软件动态可信评测模型[J]. 计算机研究与发展,2013,50(11):2397-2405

(上接第 164 页)

[5] Shio K, Singh M P, Singh D K. Routing Protocols in Wireless Sensor Networks- A Survey[C]// International Journal of Computer Science and Engineering Survey, 2010(1): 66-67

[6] Zhou Ji-liang, Cao Qi-ying, Li Cai-xia, et al. A genetic algorithm based on extended sequence and topology encoding for the multicast protocol in two-tiered WSN[J]. Expert Systems with Applications: An International Journal, 2010, 37(2): 1684-1695

[7] Babaie S, Shokraneh S, Ghaffari A, et al. CCGA: Clustering Based on Cluster Head with Genetic Algorithm in Wireless Sensor Network[C]// International Conference on Computational Intelligence and Communication Networks. 2010: 367-371

[8] Nematy F, Rahmani N, Yagouti R. An Evolutionary Approach for Relocating Cluster Heads in Wireless Sensor Networks [C]// Proceedings of the 2010 International Conference on Computational Intelligence and Communication Networks. 2010: 339-343

[9] Awwad S A, Ng C K, Noordin N K, et al. Cluster Based Routing Protocol for Mobile Nodes in Wireless Sensor Network[J]. Wireless Personal Communications: An International Journal, 2011, 61(2): 251-281