

基于特征点匹配的软件安全测试算法

白凤凤,刘继华,高文莲

(吕梁学院 计算机科学与技术系,山西 吕梁 033000)

摘 要: 针对当前软件安全测试算法检测准确性较差,异常点定位精度低,且检测效率不高的问题,提出一种基于特征点匹配的软件安全测试算法,完成软件安全测试. 利用因素集与因素指标矩阵的确定,和匹配指标矩阵与因素权重向量的计算,给出相似度计算结果;对相似实例的规则进行确定,获得高精度软件异常检测结果;采用二级制法对异常点进行定位,并将其转换为二进制向量;对多重异常二进制向量进行计算获得软件异常点定位集合. 实验表明,所提算法不仅提升了软件异常点检测和定位准确性,还有效增强了软件安全检测效率.

关键词: 特征点匹配;软件安全;测试

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2018)10—0099-04

Software Security Testing Algorithm Based on Feature Point Matching

BAI Feng-feng, LIU Ji-hua, GAO Wen-lian

(Department of Computer Science and Technology, LvLiang University, Lvliang 033000, China)

Abstract: In view of the current software security test algorithm's poor accuracy, outlier location accuracy and low detection efficiency, a software safety test algorithm based on feature point matching is proposed to complete the software safety test. Determined by matrix factors and factors, and the matching index matrix and attribute weight vector calculation, calculation results are given for similarity; similar instance rules are determined to obtain high precision software anomaly detection results; locate the outliers by two method, and converts it to a binary vector of multiple anomalies; the binary vector is calculated to get the software abnormal location set. The experimental results show that the proposed algorithm not only improves the accuracy of detection and location of the software outliers, but also improves the efficiency of software security detection.

Key words: feature points match; software security; test

1 引言

软件的安全性能检测一般是对软件安全性能是否满足客户对软件安全的需求进行检测,其中可能会包含机密性、访问控制和安全管理等方面^[1].

在计算机软件应用过程的角度来看,计算机软件的安全检测是不可避免的. 在黑客肆虐,病毒猖獗的环境下,很多软件由于本身存在安全问题,更容易被不同种类的病毒攻击^[2]. 综上,计算机软件的安全性检测,是保障计算机软件正常工作,与计算机安全

研究中必不可少的步骤^[3]. 计算机软件安全性测试重要性日益凸显,使得很多学者纷纷对其进行研究^[4]. 文献[5]的测试速度较快,但其中没有对异常信号进行定位,恶意数据和故障数据的检测率较低. 以高效解决当前相关研究结果中存在的问题,提高网络安全性为目的,提出一种基于特征点匹配的软件安全测试算法.

2 软件安全测试与定位

为提升软件安全测试结果的准确性,该文将软

件的安全测试分为软件恶意信号和故障信号的检测,与异常信号定位两部分。

2.1 基于特征点匹配的软件安全测试

基于特征点匹配的软件安全检测,利用安全特征和已存在的安全问题实例之间的相似度匹配完成。以定义相似度为目的,对相似元和相似系统进行定义。

相似元定义为:假设一段源程序,和源代码集合间有相似要素,则它们间就存在相似元^[6]。相似系统定义为:存在一个或者多个相似元组成的集合,可成为相似系统,例:一段源代码能够当作利用多个特征构成,这时能够将其看成相似系统。相似度定义为:其对相似系统间的贴近程度进行了描述,是相似系统之间存在的相似程度进行定量化识别的指标。能够利用定义相似度计算式,对不同系统间相似度大小进行衡量^[7]。

综合上述内容,对相似度进行计算。其中要对因素集,和因素指标矩阵进行确定:因素集是在相似元素集合选取出的,将参与相似特征匹配的软件安全性特征集合定义为:

$$T = \{t_1, t_2, \dots, t_n\} \quad (1)$$

式中, t 代表软件安全性特征; n 代表软件安全性特征数量。现已知软件安全缺陷中软件的源代码级,或者模块级的实例矩阵表达式为 $S = \{s_1, s_2, s_3, \dots, s_m\}$,式中, s 代表对应实例的软件安全特征向量, m 代表实例数量。假设利用 s_0 代表软件安全性特征向量,那么因素指标矩阵 $(S_{ij})_{n \times m}$ 表达式为:

$$(S_{ij})_{n \times m} = \begin{bmatrix} t_1 & t_2 & \dots & t_n \\ s_1 & s_2 & \dots & s_n \\ \dots & & & \dots \\ s_{m1} & s_{m2} & \dots & s_{mn} \end{bmatrix} \quad (2)$$

式中, i 和 j 分别代表实例数量和模块数量,根据式(2)的因素指标矩阵的计算,对匹配指标矩阵进行计算,过程中,将实例的软件安全性特征,和对应软件的源代码级,或者模块级安全特征实施差运算操作,再进行归一化处理,获得匹配指标矩阵 $(d_{ij})_{n \times m}$:

$$\Delta = (d_{ij})_{n \times m} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & & & \dots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix} \quad (3)$$

其中:

$$d_{ij} = \frac{|s_{0i} - s_{ij}|}{\sum_{j=1}^m |s_{0i} - s_{ij}|} d_{ij} = 0 \quad (4)$$

式中, d_{ij} 代表实例的软件安全性特征,和对应软件的源代码距离值; s_{0j} 代表第 j 个软件安全特征向量; s_{ij} 代表因素指标矩阵中第 i 行第 j 列软件安全特征向量值。

依据上述计算权重的方式,对因素权重向量 W 进行计算:

$$W = (w_1, w_2, w_3, w_n) \quad (5)$$

根据式(5)的计算,对相似度值进行计算,则模块和实例之间的欧式距离为:

$$d_{0i} = \sqrt{\prod_{j=1}^n w d_{ij}} \quad (6)$$

据式(6)可得到模块和实例之间相似度值 γ_{α} :

$$\gamma_{\alpha} = 1 - d_{0i} \quad 0 < \gamma_{\alpha} < 1 \quad (7)$$

基于相似学理论,如果 $\gamma_{\alpha} = 1$,那么证明软件安全特征属于完全匹配的状态;如果 $\gamma_{\alpha} = 0$,那么证明软件安全特征属于完全不匹配状态;如果 $0 < \gamma_{\alpha} < 1$,那么证明随相似度的不断变大,匹配实例相似性更强。

在基于特征点匹配的软件安全测试算法中,以历史经验为依据,应选取一个阈值当作相似匹配判定依据^[7]。假设相似程度值比该阈值低,那么说明当前检测出的源代码为正常,软件不存在安全隐患;假设相似程度值比该阈值高,那么表示当前检测出的源代码出现了异常,软件安全存在缺陷,需要做进一步地记录和分析,该阈值的提出可有效提升软件异常检测效率,文章不对该方面进行详细阐述。

满足相似度制定的可行解也许不唯一,需要设置一个实例确定规则,该规则即为比较实例和模块相似程度,把最大的相似度实例当作借用实例,则有:

$$S_Rule(j) = \max(\gamma_{\alpha}) * \lambda \quad (8)$$

式中, $S_Rule(j)$ 代表第 j 个模块实例确定规则; $\max(\gamma_{\alpha})$ 代表最大的相似度运算; λ 代表软件安全检测阈值,该值控制在 $0.3 \sim 0.5$ 间,检测效果最佳。上述计算中,如果有几个实例均与最大相似度值相对应,那么选择一个实例当作相应配置结果。

利用上述过程,软件安全性测试算法,既能够对当前软件源代码,与异常源代码实例之间的相似程度进行描述,还能够利用满足特征匹配条件的各个实例特征,对于软件安全检测的重要性,选出可能性最大且最有代表性的实例,以此大大提升检测算法的准确性。

2.2 软件安全测试结果定位

根据 2.1 对软件安全性检测的结果,利用二进制法对异常点进行定位. 主要集中于软件的易损性,异常点的易传播性,以及难以定位性上.

要对异常点进行高精度定位,首先要对异常域进行分析^[8]. 在异常域中,一个软件异常点 c 属于一类异常,这些域均为异常点定位监控设备的集.

其次,将相同域合并至 C_1 :组合一部分已经得到的相同集合,同时将这些相同集合进行等效分类为 $C_1, C_2, \dots, C_M, M < 4N$, 其中, N 代表异常点定位监控设备的数量.

将 C_1 转换为二进制向量,按照异常点定位监控设备有序排列,实现相关二进制转换,即 $Bin(C_1)$, 那么二进制向量的长度,就是软件中异常点定位监控总数. 假设第 J 个异常点定位监控告警,那么 $Bin(C_1)$ 中第 J 位是 1, 否则为 0.

对多重异常相关二进制向量 $Bin(C_k)$ 进行计算,考虑到短时间范围内可能同时出现的两种异常,两者同时出现的异常域应与两者单个异常发生的异常域相等,也就是:

$$Bin(C_k) = Bin(C_1 \cup C_j) = Bin(C_1) \vee Bin(C_j)$$

(9)

式中, $\vee$ 代表或运算; $Bin(C_1 \cup C_j)$ 代表两种异常同时发生; $Bin(C_1)$ 和 $Bin(C_j)$ 代表两种单个异常发生. 假设 $Bin(C_k)$ 等于单个异常产生的二进制向量,则可将其丢弃,因发生一个异常的可能性要大于两个异常,由此,仅需保留和单个异常告警相关向量相异的二进制向量 $Bin(C_k)$ 即可.

设定, $P(C_1)$ 代表涉及单异常情况时,软件集合或双重异常时的软件集合,将该值保存. 对二进制树进行构建,该树的深度与软件异常定位监控设备数量相等,如图 1 所示:

图 1 中, V_1 代表可以感知到信号变化的设备集合; V_2 代表可以感知到信号变化,与带外干扰的设备集; V_3 代表可以感知到信号变化,与波长偏移的设备集.

从图 1 中的 a 部分可了解到,单个异常二进制向量中: $Bin(C_1) = (0100), Bin(C_2) = (0001) \dots$, 而多重的异常二进制向量中: $Bin(C_k) = Bin(C_1) \vee Bin(C_2) = (0101)$, 一旦管理员从计算机中接收了告警信息,就对告警的来源进行判断,假设告警信息来源于 $(V_3, 1)$, 那么对应的向量为 (0001) , 能够利用 $P(C_2)$ 提供的监控设备

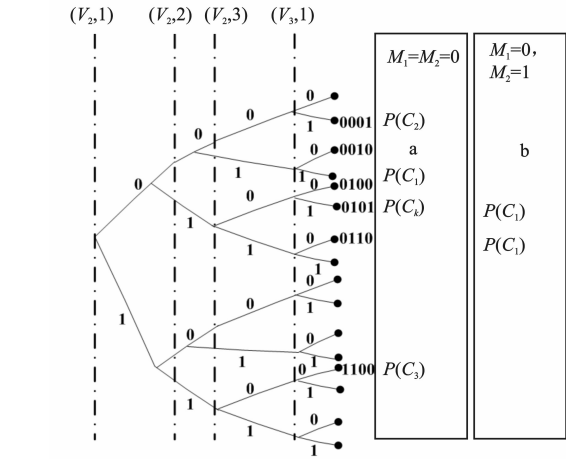


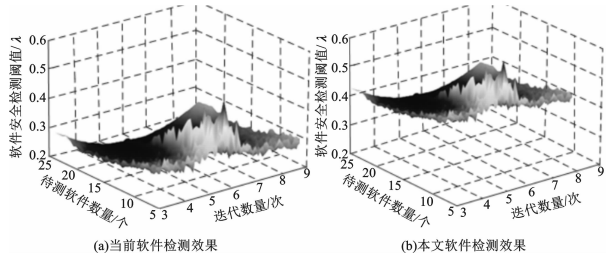
图 1 二进制故障树

集合完成异常定位. 如果收到 $(V_2, 3)$ 设备发出的告警为错误的,也就是存在误告警,即 $M_1 = 0, M_2 = 1$, 那么二进制向量 (0110) 的异常集合有可能是 $P(C_1)$, 此种状况如图 1 中的 b 部分. 综上,二进制树由根部遍历至叶子,就会给出相关异常对应的监控设备集合,即能够获得软件异常定位集合.

3 实验结果与分析

实验中,对基于特征点匹配的软件安全测试算法,异常检测效果、异常检测所需时间、异常点定位精度几方面进行验证,实验平台搭建在 MATLAB 上.

实验(一):



(a)当前软件检测效果 (b)本文软件检测效果

图 2 不同软件安全测试算法异常检测效果对比

分析图 2 实验结果可知,当前软件安全测试算法无法得到高精度异常检测结果,而基于特征点匹配的软件安全测试算法在进行软件异常检测时,利用软件安全特征,和已存在的安全问题实例之间的相似度进行匹配,以匹配结果检测软件是否存在安全缺陷,并利用满足特征匹配条件的各个实例特征,对于软件安全检测的重要性,选出可能性最大且最有代表性的实例,该环节大大提升了算法准确性,在

实例确定规则中,对软件安全检测阈值进行了考虑,进一步增强了软件安全检测正确性.

实验(二):

针对软件安全测试算法的异常点定位精度实验,以异常点检测结果为依据,利用该文提出的二进制制法对异常点进行定位.图3为软件运行信号波形,其中存在干扰信号和异常信号,以此对该文所提算法进行验证,实验结果如下:

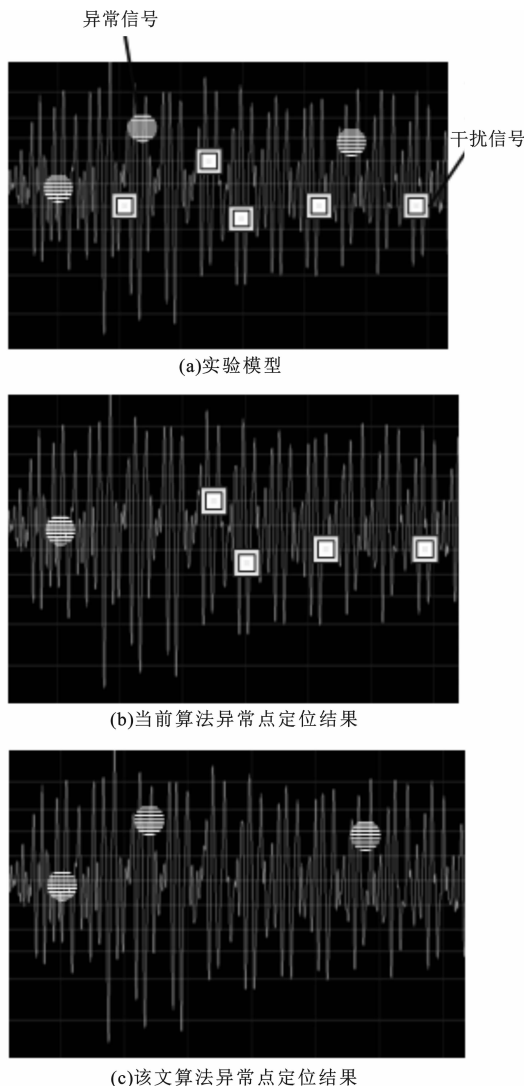


图3 不同算法异常点定位结果对比

由图3可知,当前算法软件异常点定位精度要低于该文算法,当前算法异常点定位结果中,干扰信号对定位结果的干扰,使真正的异常点没有被准确定位,相反,干扰信号被误定位情况严重,表明当前算法可靠性较低.该文所提算法为提升异常点定位精度,对异常域进行了定义,同时二进制树由根部遍历至叶子,给出了相关异常所对应的监控设备集合,这样获得的软件异常定位集合遍

历了整个故障树,此举可大大提升软件异常点定位精度.

4 结束语

计算机运行中,软件作为功能模块中不可缺少的部分,对大众工作和学习有着十分重要的影响.而软件存在的安全问题,不利于计算机的正常运行,当前软件安全测试算法无法实现异常点高精度地检测与定位,可行性较差.该文提出一种基于特征点匹配的软件安全测试算法,将软件源代码安全特征,和已知软件安全性的缺陷特征实施相似度匹配操作,完成软件异常检测,通过二进制制法对异常检测结果进行定位,可高效解决当前算法存在的问题,具有现实意义.

参考文献:

- [1] 孙科,覃杨森.基于GSPN的机载构件化软件系统可靠性测试方法[J].电子设计工程,2017,25(17):16-20.
- [2] 刘广,黄百乔,刘畅.基于减函数的多层贝叶斯离散型软件可靠性验证测试方案[J].计算机应用研究,2017,34(3):761-764.
- [3] 吕晗阳,徐浩军,朱和铨.一种虚拟试飞验证飞控软件安全性评估方法[J].计算机仿真,2017,34(10):45-48.
- [4] 黎晖,石小华,林柯军,等.工程装备嵌入式软件测试环境平台技术研究[J].计算机测量与控制,2016,24(4):10-12.
- [5] 高超,郑小妹,贾晓启.面向云平台的多样化恶意软件检测架构[J].计算机应用,2016,36(7):1811-1815.
- [6] 李楠楠.全对偶算法在移动应用软件测试中的应用研究[J].科技通报,2017,33(8):175-179.
- [7] 杨阳,吴际,苑春春,等.面向DO-178C软件测试过程的目标符合性论证模式[J].计算机研究与发展,2017,54(3):654-668.
- [8] 崔芮华,王绍敏.基于多维特征量的航空串联故障电弧检测[J].科学技术与工程,2017,17(13):38-43.

作者简介:

白凤凤 女,(1976-),硕士,副教授.研究方向为软件测试、数据挖掘等. E-mail:bff123@163.com.

刘继华 女,(1975-),硕士,副教授.研究方向为软件测试、云计算等.

高文莲 女,(1968-),硕士,副教授.研究方向为计算机网络、数据挖掘、计算机应用.