文章编号：2095-2163(2020)07-0253-04 中图分类号：TP393.08 文献标志码：A

网络化软件异常行为特征分析与识别研究

胡 柳，张四平，肖瑶星，邓慈云，卢艳芝
（湖南信息职业技术学院，长沙 410200）

摘 要：针对网络化软件异常行为的复杂性、模糊性、难控性等特点，本文提出了一种网络化软件异常行为特征分析模型。该模型由行为提取模块、行为分析模块、行为识别模块构成。通过对软件消息行为语义、系统应用程序接口、系统函数等调用情况的提取，结合信任度计算方法进行行为识别，实现对软件行为的分类。

关键词：网络化软件；异常行为；信任度

Research on analysis and recognition of abnormal behavior characteristics of networked software

HU Liu, ZHANG Siping, XIAO Yaoxing, DENG Ciyun, LU Yanzhi
(Hunan College of Information, ChangSha 410200, China)

【Abstract】 Aiming at the characteristics of complexity, ambiguity, and uncontrollability of abnormal behavior of networked software, this paper proposes an analysis model for abnormal behavior of networked software. The model consists of a behavior extraction module, a behavior analysis module, and a behavior recognition module. By extracting the calling behavior of software messages, system application program interfaces, system functions, etc., the behavior recognition is combined with the trust calculation method to realize the software behavior classification.

【Key words】 networked software; abnormal behavior; trust

0 引 言

在互联网迅速发展的环境下，软件由传统的单机软件逐步演化为现在的网络化软件。随着“数据上网”，各类关键、机密数据存在于网络，各类恶意软件不断窃取用户的敏感数据的问题时有发生。因此，网络化软件的信任危机受到了软件工程领域学者的重视。对网络化软件的异常行为进行有效的甄别，提高网络化软件的信任度是急需解决的问题。文献[1]提出了基于社会学信任理论的软件可信性概念模型，并通过度量评估实验，验证了模型的可行性和有效性。文献[2]提出了一种综合信任云评价方法，进行网络化软件的信任度量评估。文献[3]提出了基于复杂网络的网构软件信任度模型，将复杂网络的小世界特征和无标度特性引入到网构软件的信任度模型中，并进行仿真实验。网络化软件异常行为分析及可信度量评估领域内，许多学者都进行了相关的研究，同时也利用相关技术进行实验，以获取较好的效果。如采用 BP 神经网络[3]、时间变化敏感点[4]、支持向量机[5]、关联规则算法[6]等。

本文对网络化软件异常行为特征进行分析与识别，通过对软件消息行为语义、系统应用程序接口、函数等调用情况的提取，构建网络化软件异常行为识别框架结构，并进行特征提取与训练，采用信任度计算方法对软件行为进行识别，实现对软件行为的分类。

1 软件行为分类模型

软件行为的可信性是指软件运行作为分析对象，观察并判定其动作行为是否较其功能越界、违规，目前多采用通过操作记录方式来进行检测。软件行为一般分为静态和动态行为，静态分析是指对软件代码进行检测，如人工方式、执行程序流程等。而动态分析是指在软件运行时设置相应的检查点，分析系统函数调用、参数传递、地址查阅等。软件行为的表现方式如图 1 所示。

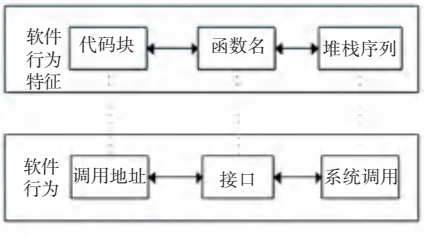

图 1 软件行为的表现方式

Fig. 1 The way software behaves

基金项目：湖南省教育厅科学研究项目(18C1600)。

作者简介：胡 柳(1988-)，男，硕士，讲师，信息系统项目管理师，主要研究方向：网络化软件开发、软件工程。

收稿日期：2020-05-09

软件行为特征的挖掘,是从软件本身大量、不完全及随机的行为操作过程中,提取有用的信息。而对采集到的软件行为需要进行识别分类,以判断其是正常的操作行为还是异常的操作行为。软件行为特征分类过程如图 2 所示。

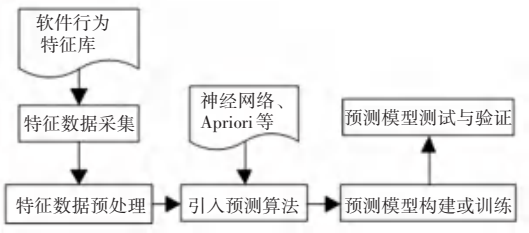

图 2 软件行为特征分类过程

Fig. 2 Software behavior feature classification process

进行软件行为特征分类时,网络化软件异常行为需要进行提取、行为分析及行为识别。其中,行为提取模块是结合软件消息行为语义、系统应用程序接口、系统函数等调用情况进行提取,主要在二进制代码下进行。行为分析是对提取到的行为记录进行分析与建模,将其逐渐进行性质明确。行为识别是在行为分析的基础上,利用相关的分类模型,依据特征库在预测算法的预测下,将指定的软件行为进行正常或异常的分类。软件行为特征分析模型如图 3 所示。

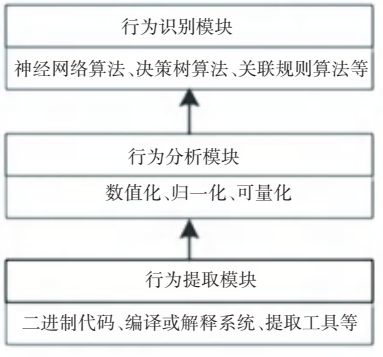

图 3 软件行为特征分析模型

Fig. 3 Software behavior characteristic analysis model

2 软件行为提取

软件行为提取是在二进制代码下进行行为分析。由于网络化软件所处环境不同而使得提取更为复杂,需要考虑不同编译环境、平台的情况。本文实验均是在 Windows 操作系统下进行软件行为提取的相关工作。

2.1 软件消息行为语义

软件消息行为语义是指软件的消息行为状态并进行分类管理。通常情况下软件消息行为语义包含:打开(open)、使用(use)、操作(oper)、关闭(close)四种。这四类语义都涉及到与软件相关或不相关的文件(file)、设备(device)、状态(status)、服务(serve)、资源(resource)等。如某个软件消息行为语义包含有“使用”,且使用计算机的网络资源,通过网络服务进行数据传输。采用数据元组表述为式(1)。

$$(M_{ouoc}, N_{fdssr}, H). \tag{1}$$

其中,M_{ouoc} 代表消息语义的打开、使用、操作、关闭;N_{fdssr} 代表四类语义涉及到的文件、设备、状态、服务、资源;H 代表四类语义行为的具体操作。

2.2 系统应用程序接口

系统应用程序接口(API)是一些预先定义好的函数,或指软件系统不同组成部分衔接的约定。其目的是用来提供应用程序与开发人员基于某软件或硬件得以访问的一组例程。

API 函数包含在 Windows 系统目录下的动态链接库文件中,用户软件的每个动作都会引发一个或几个函数的运行,并进行记录。如当单击某个窗体上的按钮时,Windows 会发送一个消息给窗体。可以理解为,操作系统的各项功能中大部分都可以采用 API 函数进行调用,即用户可能在未知的情况下,软件即可完成对操作系统的相关操作。如 GetMessage 函数调用线程的消息队列里取得一个消息,并将其放于指定的结构。此函数可取得与指定窗口联系的消息和由 PostThreadMessage 寄送的线程消息,但不接收属于其他线程或应用程序的消息。获取消息成功后,线程将从消息队列中删除该消息。如使用 API 函数 CDdoor 来控制光驱的打开与关闭,同时它也包含了对异常的处理机制。使用代码,如 Call CDdoor("set CDAudio door closed", 0, 0, 0)//用以关闭光驱门,Call CDdoor("set CDAudio door open", 0, 0, 0)//用以打开光驱门。

2.3 用户函数或自定义函数

用户函数或自定义函数是由程序员编写的功能函数。函数的调用 function call 过程可理解为调用者向被调用者传递一些参数,然后执行被调用者的代码,最后被调用者向调用者返回结果。在函数调用时,第一个进栈的是主函数中函数调用后的下一条指令的地址,然后是函数的各个参数。一条指令的执行,是根据 PC 中存放的指令地址,将指令由内存取到指令寄存器 IR 中,程序采用顺序执行的方式进行执行。但也有一些例外,如调用其它函数、调用后返回、程序的控制结构等。函数调用过程如图 4 所示。

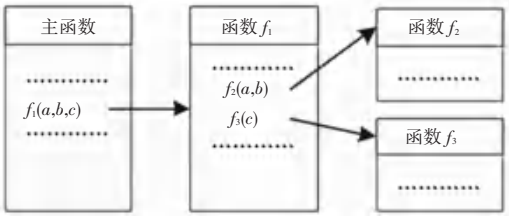

图 4　函数调用过程

Fig. 4　Function calling process

3　信任度计算方法

网络化软件异常行为信任度计算[7]，是指在软件行为特征提取的基础上，对该行为进行信任计算。本文采用加权统计方法进行，为行为识别提供数据支撑。在软件消息行为语义的基础上对打开、使用、操作、关闭四种行为进行权值分配，并定义软件相关或不相关的文件、设备、状态、服务、资源、其它等的权值。每个 (M_{ouoc}, N_{fdssr}, H) 对应到具体的行为，可表述为式(2)。

$$F_{MNH} = M_{ouoc} \times Y_m + N_{fdssr} \times Y_n + H \times Y_h. \quad (2)$$

其中，F_{MNH} 为信任度，Y_i 为权值。即统计每个消息行为的权值和。

在进行计算时需要给予初始值，并根据各软件行为进行动态的不断调整，以达到最佳的识别效果。在进行数据训练时，分别赋予的权值见表 1。

表 1　权值分配表

Tab. 1　Weight distribution table

	file(%)	device(%)	status(%)	serve(%)	resource(%)	other(%)
open(15%)	10	20	10	25	30	5
use(30%)	25	25	5	20	20	5
oper(30%)	25	30	5	25	10	5
close(25%)	10	20	10	25	30	5

提取到的软件行为进行数值化并进行加权统计后，设定符合样本特征的置信区间。与权值分配一样，置信区间的可信性也需要不断的调整，以获取最佳的效果。初始时采用 50%－100%为可信空间，如式(3)所示。

$$\begin{cases} 0 & 0\% \leqslant F_{MNH} < 50\%, \\ 1 & 50\% \leqslant F_{MNH} \leqslant 100\%. \end{cases} \quad (3)$$

其中，当 F_{MNH} 小于 50%时，则将其置于不可信区间，大于 50%则置于可信空间。50%的阈值需要不断进行优化与调整，以获得最佳的效果。

4　行为分类

针对网络化软件的行为提取与信任度计算，可获取每条行为的分类结果。本文采用 SVM 方法对数据进行分类训练。支持向量机(SVM)[8]是一种可训练的机器学习方法，它在某些方面比其它方法更具优势，如解决小样本、非线性和高维模式识别。通过非线性映射，将低维空间的特征映射到高维空间，使原本线性不可分的数据成为线性可分，但带来的是升维灾难。SVM 采用核函数解决这个问题，通常有线性核函数、多项式核函数、径向基核函数和 sigmoid 函数等，其计算方法分别如式(4)－式(7)所示。

$$K(x,y) = x \cdot y, \quad (4)$$

$$K(x,y) = [(x \cdot y) + 1] \times d, \quad (5)$$

$$K(x,y) = \exp\left(\frac{-|x-y|^2}{d^2}\right), \quad (6)$$

$$K(x,y) = \tanh(a \times (x \cdot y) + b). \quad (7)$$

SVM 模式识别与回归的软件包 LIBSVM 利用默认参数可以解决很多问题，它使用的训练数据和检验数据文件格式为：<label> <index1>：<value1> <index2>：<value2>……。图 5 是采用 LIBSVM 得到的分类图。

图 5　LIBSVM 分类图

Fig. 5　LIBSVM classification chart

本文将试验样本数据采用 LIBSVM 工具包进行训练文件的生成，并进行测试。采用准确率、精确率及召回率进行数据统计，采用 TP 表示将正类预测为正类的数量、TN 表示将负类预测为负类的数量、FP 表示将负类预测为正类的数量、FN 表示将正类预测为负类的数量，准确率 $AC = (TP + TN)/(TP + TN + FP + FN)$、精确率 $P = TP/(TP + FP)$、召回率 $R = TP/(TP + FN)$。将 80 条软件行为特征数据进行量化并生成训练文件。其中正样本 50 条、负样本 30 条。将 30 条软件行为特征数据进行测试，其中正样本 20 条、负样本 10 条，结果显示正样本中 17 条分类正确、负样本中 8 条分类正确。准确率 AC 为 83%，精确率 P 为 89%，召回率 R 为 85%。

5　结束语

通过构建网络化软件行为分类模型，将行为提取模块、行为分析模块、行为识别模块三个模块进行整合，并将软件的软件消息行为语义、系统应用程序

接口、系统函数等调用情况进行提取,采用综合加权方法计算行为特征的信任度,并结合 SVM 方法进行分类。模型为软件行为特征分析提供了一定的思路。但行为特征提取上还需要进一步丰富,采用更为全面的特征项进行描述,同时在信任度计算上需要结合不断演化调整的阈值进行优化,以获得更为有效的分类结果,为软件行为特征的识别提供强有力的支撑。

参考文献

[1] 杨曦,罗平,贾古丽. 基于社会学信任理论的软件可信性概念模型[J]. 电子学报,2019,47(11):2344-2353.

[2] 张洁. 网络化软件的信任度量模型优化仿真分析[J]. 计算机仿真,2016,33(10):278-281,299.

[3] 徐婵,刘新,吴建,等. 基于 BP 神经网络的软件行为评估系统[J]. 计算机工程,2014,40(9):149-154.

[4] 刘红英. 大数据下网络化软件时间变化敏感点检测仿真[J]. 计算机仿真,2019,36(12):353-356,395.

[5] 杨柳. 网络化软件交互式行为可信性监测分析仿真[J]. 计算机仿真,2018,35(6):304-307,312.

[6] 胡柳,邓杰,赵正伟,等. 一种基于关联规则的网络软件缺陷预测方法[J]. 信息技术与网络安全,2018,37(4):41-44.

[7] 林杰,刘波. 融合信任计算与语义分析的博客推荐算法[J]. 计算机工程,2018,44(7):271-278.

[8] 刘方园,王水花,张煜东. 支持向量机模型与应用综述[J]. 计算机系统应用,2018,27(4):1-9.

(上接第 252 页)

图 7 观看课件

图 8 课程管理

图 9 课件上传

Fig. 7 Watch courseware Fig. 8 Course management Fig. 9 Courseware upload

平台的数据存储主要包括客户端状态信息和教学课件。服务器采用了 MySQL 作为后台数据库,存储教师上传的各类教学课件以及课件被使用的状态信息。利用 HTML5 的 manifest 离线存储技术,当客户端遇到突然断网的情况,平台依然能记录当时的学习状态,待联网后与服务器自动同步。此外,平台的离线存储功能还可以对教学课件进行缓存,利于学生离线学习,而在网络状态下,平台自动加载、更新课件状态。这样的离线存储技术一方面提高了客户端的加载速度,另一方面也提高了学生的学习效率[8]。

4 结束语

跨平台、高交互的移动学习平台为移动学习提供了更加有力的支持。本文主要采用自动适配系统的 Ionic 前端设计框架和以数据交互为核心的 AngularJS 框架,设计并实现了一款低部署成本、高交互性的移动学习平台。平台界面友好,导航清晰,性能稳定,知识内容按模块分类呈现,操作便捷快速。但平台也有一些不足之处,如在低版 Android 系统存在性能缺陷、视频缓存速度较慢等,后期还需采用性能优化、视频编码等技术继续完善。

参考文献

[1] 吕竹筠,郭路路,李德贵,等. 大学生运用智能手机进行移动学习方式的探究[J]. 教育教学论坛,2019(8):230-232.

[2] 赵学铭,叶颖,王茜. 基于 HTML5 交互式移动学习平台的设计与实现[J]. 黑龙江科技信息,2017(3):197-199.

[3] 商锦,林亮,王雨,等. Ionic 在混合模式 APP 中的应用[J]. 软件导刊,2017,16(5):132-134.

[4] 高兴建,花晓慧,邢溧萍. 基于 Ionic 的混合移动应用的研究与实现[J]. 计算机时代,2018(3):31-34.

[5] 童茂林. 基于 Ionic 框架的混合应用开发技术探究与实现[J]. 无线互联科技,2017(19):133-135,138.

[6] 邓璐娟,陈欣欣,雷科伟,等. 基于 Ionic 的自适应前端技术研究与应用[J]. 计算机系统应用,2018,27(11):84-89.

[7] 刘青丹,王舒憬,强杰. Ionic+AngularJS 框架在跨平台旅游 APP 客户端系统中的应用[J]. 工业控制计算机,2018,31(1):142-143.

[8] 赵学铭,王刚. 基于 HTML5 的交互式移动学习平台研究[J]. 现代教育技术,2016,26(9):106-112.