第43卷 第1期 计 算 机 工 程 2017年1月
Vol.43 No.1 Computer Engineering January 2017

·人工智能及识别技术· 文章编号: 1000-3428(2017)01-0241-06 文献标志码: A 中图分类号: TP309.2

基于机器学习的移动终端高级持续性威胁检测技术研究

胡 彬[1a],王春东[2],胡思琦[1b],周景春[1a]

(1.北京邮电大学 a.软件学院; b.网络空间安全学院,北京 100876;
2.天津理工大学 计算机与通信工程学院,天津 300384)

摘 要: 移动端高级持续性威胁(APT)攻击是近年来出现的一种极其危险的攻击方式,通过窃取信息对设备造成高风险且可持续性的危害。而针对移动端入侵检测的方案由于检测特征不够完善,检测模型准确率不高且存在过拟合问题,导致检测效果不理想。针对上述问题提出一种优化的检测模型,利用静态检测技术提取出终端应用的静态特征,优化模型对恶意应用的敏感程度,引用滑动窗口迭代算法提取出延迟攻击特征,以优化模型对延迟攻击的检测能力,同时使用Boost技术将决策树、逻辑回归、贝叶斯等分类算法进行融合,通过实验证明该模型提升了APT检测准确率并规避了过拟合问题。

关键词: 机器学习; 高级持续性威胁检测; 分类器; 模型融合; 静态检测; 关联分析

中文引用格式: 胡 彬,王春东,胡思琦,等.基于机器学习的移动终端高级持续性威胁检测技术研究[J].计算机工程,2017,43(1):241-246.

英文引用格式: Hu Bin,Wang Chundong,Hu Siqi,et al. Research on Advanced Persistent Threat Detection Technology for Mobile Terminal Based on Machine Learning[J]. Computer Engineering,2017,43(1):241-246.

Research on Advanced Persistent Threat Detection Technology for Mobile Terminal Based on Machine Learning

HU Bin[1a],WANG Chundong[2],HU Siqi[1b],ZHOU Jingchun[1a]

(1a.School of Software; 1b.School of CyberSpace Security,Beijing University of Posts and Telecommunications,Beijing 100876,China;
2.School of Computer and Communication Engineering,Tianjin University of Technology,Tianjin 300384,China)

【Abstract】 Advanced Persistent Threat(APT) whose main goal is to steal information becomes a dangerous attack in recent years,which can bring high risk and persistent attack to the mobile devices. The detection features of mobile terminal intrusion detection are not readily available,so the accuracy of detection model is not high enough and there is the over-fitting problem,which lead to poor detection effect. For these problems,this paper proposes an optimized detection model using static detection technology to extract terminal devices' static features which helps optimize the sensitivity of the model to the malicious application. It uses sliding window iterative algorithm to extract the delaying attack feature so as to optimize the model's detection capability of delaying attack and uses Boost technology to fuse the classification algorithms including the decision tree,logistic regression and Bayesian classifier. Experimental results show that the model can increase the detection accuracy of APT effectively and avoid over-fitting problem.

【Key words】 machine learning; Advanced Persistent Threat(APT) detection; classifier; model integration; static detection; correlation analysis

DOI: 10.3969/j.issn.1000-3428.2017.01.042

0 概述

近年来,随着移动端的病毒数量迅速增加,针对Android端的高级持续性威胁(Advanced Persistent Threat,APT)攻击变得更加多样化,研究发现,2/3的信息泄露来自APT攻击[1],由于APT攻击不受传统杀毒软件和防火墙的限制,需通过引入机器学习模型进行防护。

基于C&C服务器[2]长期潜伏在手机中的恶意应用,等待时机窃取金融和政府信息会带来极大隐患。目前针对该领域的研究较多但都存在一定缺陷,利用动态检测技术[3]监控恶意软件的行为和系

作者简介:胡 彬(1990—),男,硕士研究生,主研方向为机器学习、移动安全;王春东,教授;胡思琦、周景春,硕士研究生。
收稿日期:2016-01-15 修回日期:2016-02-24 E-mail:binhu@bupt.edu.cn

统调用，如发短信、下载恶意应用等高危行为的发生，该方法再利用脚本重现病毒行为进行批量检测时极复杂，且消耗资源较大，性能难以支持，因此移植到手机上可行性不高。基于源码分析的静态检测，如分析源码调用的关键函数，反编译或提取16 进制中的特征等方式，虽然效率较快，但精准度不高，无法抵抗混淆和加密的恶意软件攻击。机器学习算法在 Android 入侵检测领域的使用相当广泛，SVM[4]、决策树[5]等也可以用来构建 Android 入侵检测模型[6]，可用来检测手机病毒但无法检测出隐藏周期较长的 APT 攻击。

本文分析 APT 攻击的特征，优化模型特征选择方法，将静态检测技术运用到特征提取过程中，提升模型对恶意应用的检测能力。针对 APT 攻击延迟攻击的特性引入滑动窗口迭代算法，加强模型对延迟攻击的检测能力。

1 攻击分析及模型设计

1.1 APT 攻击分析

APT 攻击手段多样，社会工程学在 APT 的使用中应用较为广泛[7]，攻击的第 1 步会寻找目标如金融和政府部门，利用钓鱼技术诱使用户打开一个含有病毒的载荷；第 2 步隐匿地将病毒或者后门软件安装在目标上；第 3 步，攻击者通过后门软件与目标建立连接；第 4 步，通过命令控制更多的设备提升权限，访问更接近目标的信息，这与 IKC 模型[8]的分析相近，IKC 模型优化了上述步骤，添加了 APT 攻击的潜伏期，APT 攻击完成后删除证据防止监控软件，并提出多数 C&C 控制通过加密通信将命令传输到设备。

针对 Android 平台的 APT 感染方式分为通过浏览器访问钓鱼网站、含病毒载体的电子邮件、推荐下载（如微信或短信）、连接受感染的电脑、被安装病毒载荷等，高级可持续风险攻击通常还应用了 0day 等系统漏洞，相当难以觉察，在内网中感染更多机器，在执行黑客的命令后会更新或传播病毒。

图 1 描述了一个最开始由伪基站引起的 APT 攻击场景，黑客通过伪基站假装 10086 给用户推送下载含链接的短信，用户安装虚假的中移动终端后并无察觉，虚假终端并不做破坏，唯一的目的就是在内网传播并收集有关管理员的信息。这一过程相当漫长且隐蔽，在获取高权限用户信息后，通过管理终端做跳板下载信息到个人区域，等待个人终端离开工作场所，操控其上传窃取的重要数据，整个攻击行为最难以防范的是提升权限的过程，该过程相当漫长，且都是在合法权限下进行的常规操作，难以被普通安全防护系统检测到。

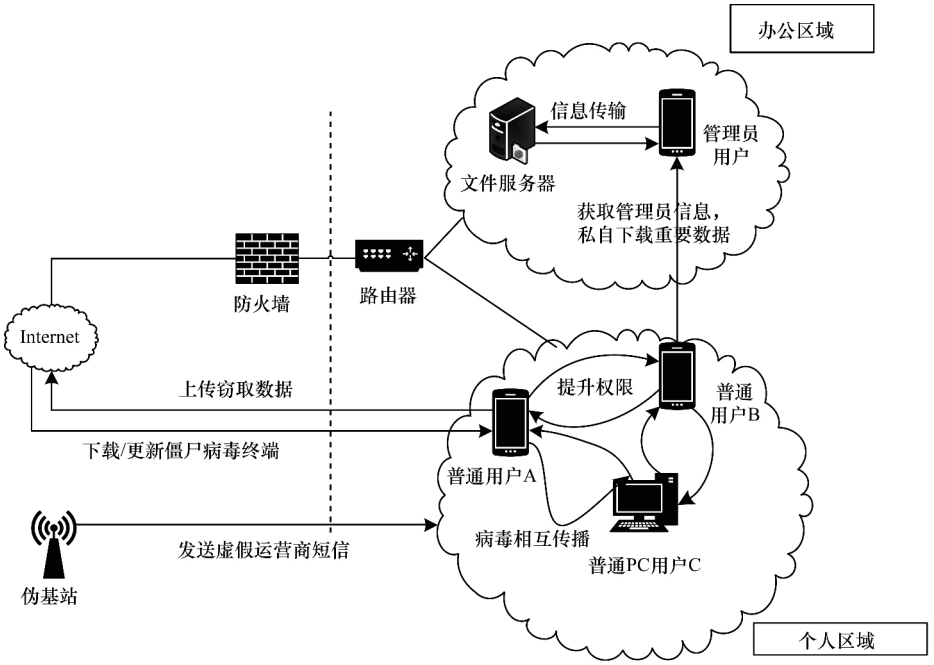

图 1 APT 攻击流程

1.2 APT 检测模型架构

基于机器学习的检测平台如图 2 所示。基于安装在手机的监控终端和处于服务器的分析预警模块及管理平台，方便系统及时地自动监控恶意行为并发出预警，平台主要功能包括：

1）终端状态监控与转发

基于 Android SDK 进行资费信息收集、硬件信息收集、进程信息收集，主要涉及 CPU、内存、Sdcard、短信、通话记录等关键部分；并将 Tcpdump 移植到移动端，定时抓取流量打包成 pcap 文件；基

于 Clamav 对应用进行检测，并存储检测日志；定期将各个终端的日志通过 https 协议传输到内网的数据处理平台。

2) 日志数据预处理

基于 python 库 scrapy 解析来自各个终端的 pcap 文件方便提取特征；将缺失值赋予初始值，异常日志数据过滤；黑名单日志直接告警，发现白名单日志默认正常。

3) 特征提取

将时间、协议名称等特征值量化；将 CPU、内存使用率、流量等参数归一化；利用滑动窗口迭代算法收集 APT 流量特征；利用关联分析算法提取关联主机的隐患特征。

4) 模型训练及预测

利用 boost 技术，将弱分类器 LR，BAYES，GBRT 等弱分类器[8] 融合，通过历史数据训练预测模型，并根据实时监控数据进行预测。

5) 告警监控及配置

该模块用于配置黑白名单，查看设备的使用状态及告警信息，可通过极光推送和 https 对地理围栏内外的设备发出上传数据、发出数据、备份数据等指令，从而在发现风险时对设备进行风险控制。

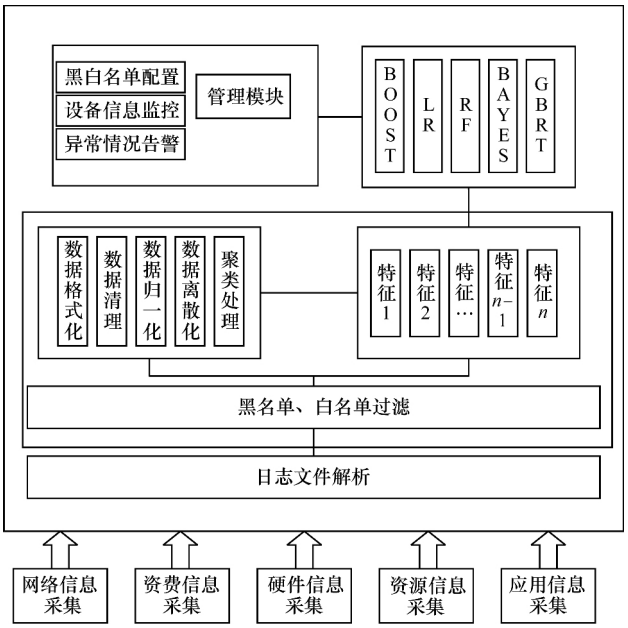

图 2　检测模型架构

2　静态特征提取

目前针对移动端的 APT 检测多局限在网络流量的研究，为提升分类器对于恶意软件的检测效果，需要引入静态检测提取应用特征。Clamav 是一款开源的杀毒引擎，经过二次开发可用于提取软件特征，主要分为软件权限特征[9] 和关键 API 特征。

2.1　软件权限特征提取

利用工具 DroidMat[10] 从 AndroidManifest. xml 提取出应用的权限特征，如表 1 所示，通过统计方法记录移动端应用的静态特征。

表 1　恶意软件常用权限部分列表

权限	权限描述
RECEIVE_SMS	接收短信
WRITE_SMS	编写短信
READ_SMS	读取短信
WRITE_EXTERNAL_STORAGE	写入外部存储
INTERNET	访问网络
RECEIVE_BOOT_COMPLETED	开机自启动

利用 Apktool 查看软件结构目录，将恶意包名如广告包名、恶意插件的包名及伪装成资源文件的文件名作为特征。

2.2　软件 API 特征提取

利用 dex2jar 将 apk 文件反编译成 java 文件，用 jd-gui 查看 APK 中 classes. dex 转化成的 jar 文件，添加高危敏感 API 的特征，如表 2 所示，如拦截短信、电话，获取手机信息、用户个人隐私等信息的敏感 API。

表 2　Android 敏感 API 部分列表

敏感 API	描述
getDeviceId( )	泄露 IMEI
getSubscriberId( )	泄露 IMSI
getSimSerialNumber( )	获取 SIM 卡号
sendDataMessage( )	发送 MMS
sendTextMessage( )	发送 SMS
rawQuery( )	SQL 注入

提取出应用的静态特征后用 Sigtool 将特征转化为 16 进制，表达成 Clamav 可识别的正则表达式，即可用统计的方法进行特征收集并传入训练模型。

3　延迟攻击特征提取

该模型的流量特征提取以 Kddcup 的 41 项特征为基础，主要包含 TCP 连接基本特征、TCP 连接内容特征、基于时间的网络流量特征、基于主机的网络流量特征，在分析过 APT 的特征后，为应对延迟攻击引入滑动窗口迭代算法[11]，提取延迟攻击特征，并根据关联分析算法提取关联隐患特征。

3.1　滑动窗口迭代算法

该算法用于抵抗 APT 隐患中的延迟攻击，利用 Pcap 文件中解析出部分的流量特征，包括基本信息：发送数据包的流量，发送的字节数，连接建立时间

戳,连接结束时间戳,产生流的源目的 IP,流的源目的端口,流使用的协议。由于僵尸应用产生流量易被大量正常通信淹没,因此不考虑流方向即源目的 IP,统计预定时间窗口的所有流量,统计出 14 个特征,并用 Kmeans 将相似流量划分到同一聚类中。由于 APT 程序的通信由程序自动产生,无人为因素,因此同种 APT 软件产生的流量具有相似性,聚类后计算同一时间窗口、同一聚类中主机间通信产生流量的相似度,假设 a 和 b 是聚类中不同流量特征,包含 ip 和 14 个特征,通过计算距离确定相似度,用 H 表示相似度,计算公式如下:

$$H_{ab}=\left|\frac{\sum_{i=1}^{14}(x_{xa}-\bar{x})\times(x_{ib}-\bar{x})}{\sqrt{\sum_{i=1}^{14}(x_{xa}-\bar{x})^2}\times\sqrt{\sum_{i=1}^{14}(x_{xb}-\bar{x})^2}}\right|$$

APT 攻击多具有延迟响应的特点,因此需要滑动时间窗口迭代信息[11],TS 表示一个滑动窗口内相似度的迭代结果,$TS_n=\sum_{i=n-m}^{m}ts_i$,$TH$ 表示一个滑动窗口内可疑度 $TH=\max(th_n,th_{n-1},\cdots,th_{n-m})$ 检测迭代后的结果,TS 和 TH 在一个滑动时间窗口结束时发送到关联分析模块。

3.2 关联分析算法

$N(x)$ 表示和 x 同一聚类并且同时间段与 x 流量相似度大小满足条件的主机集合,$H(x)$ 表示主机 x 的测评结果,$S(kx)$ 表示主机 k 和 x 的相似度,$F(x)$ 是关联分析的结果,公式如下:

$$F(x)=\begin{cases}\sum_{k\in N(x)}H_x\times S_{kx}/\sum_{k\in N(x)}S_{kx}, & H_X=0\\ \left(H_X+\sum_{k\in N(x)}H_x\times S_{kx}/\sum_{k\in N(x)}S_{kx}\right)/2, & H_X\neq 0\end{cases}$$

基于相似性及风险值的关联分析算法伪代码如下:

```
输入
A: 终端风险值
B: 终端流量聚类数据
C: 同聚类终端流量相似度结果
hosts: 终端 ip 地址集合
输出
R:  网络中主机的检测结果
1. FOR X in hosts
2.     u = v = 0
3.     FOR K in B [X]
4.         u + = A [K] × C [X] [K]
5.         v + = C [X] [K]
6.     END FOR
7.     IF A [X] = 0 THEN R [X] = u/v
8.     ELSE R [X] = ( A [X] + u/v) /2
9. END FOR
10. RETURN R
```

4 Boost 模型融合技术

Boost 分类器属于集成学习模型,将较低准确率的模型组合起来,不断迭代形成一个准确度高的模型,先使用 GBRT,LR,KNN 对一组数据进行训练得到相应弱分类器,利用 xgboost 对每个弱分类器进行加权投票,算法设计流程如下:

输入

X: 样本数据

Y: 弱分类器

$E(f(x),y,i)=e^{-yf(x)}$: 误差函数

输出

$y_1y_2\cdots y_n, y\in(-1,1)$

For t in 1,2,…,T

1) 训练弱分类器,最小化权重误差函数:

$h_t(x),\varepsilon t=E(f(x),y,i)=e^{-yf(x)}$

2) 计算分类器权重:

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon t}{\varepsilon t}\right)$$

3) 权重嵌入模型:

$F_t(x)=F_{t-1}(x)+\alpha_t h_t(x)$

4) 更新分类器权重:

$w_{i,t+1}=w_{i,t}e^{-y\alpha_t h_t(x)}, i=1,2,\cdots,n$

5) 正规化:

$\sum_i w_{i,t+1}=1$

6) 得到融合后模型:

$f(x)=\sum_{t=1}^{N}f_t(x)\alpha_t$

$G(x)=\mathrm{sign}(f(x))$

xgboost 最大的特点在于,它能够自动利用 CPU 的多线程进行并行,同时在算法上加以改进,提高了精度,且 boost 模型规避了过拟合问题。

5 实验与结果分析

5.1 数据集

为检测本文提出的 APT 隐患,仿照提出的攻击方法,实现了 3 个恶意应用。第 1 个应用无界面读取本机短信内容并写入某目录,第 2 个应用无界面读取固定内容并上传至内外开有固定端口的终端,第 3 个应用模拟成正常的界面游戏程序,骗取用户点击,将文件上传至外网。为了实验更真实,设定 C&C 服务器与内网的通信间隔,设置多种不同情况进行数据收集。

第 1 种攻击软件与短信本地备份和数据本地备份软件逻辑特征相同。第 2 种攻击软件与百度云盘等云备份软件特征类似,当单纯的基于流量检测[12]和静态检测的时候这 2 种检测方法都会失效,杀毒软件处于离线状态时对于未知的软件也无法检测。第 3 种应用是 APT 攻击的最后一步,即便发现信息也被窃取。

实验环境为，10 台 Android4.4.2 手机可以上外网，且内网处于同一网段，5 台安装恶意应用，5 台只安装正常应用，设定数据收集周期为 30 s 一次。利用 TSTAT 工具对 pcap 文件预处理后，将静态特征、CPU、内存、延迟攻击特征等利用 python 对安装病毒的收集和没安装病毒的手机的特征分别进行标记，并将特征数据进行正规化后切分成 10 个部分。

5.2　实验测试

实验利用了交叉验证方法[13]，设置了 10 个子集，每个子集均做一次测试集，其余的作为训练集。交叉验证重复 10 次，每次选择一个子集作为测试集，并根据 10 次的平均交叉验证识别正确率和召回率计算出准确率作为检验结果。

1) 滑动窗口特征对比

将收集流量时间窗口[14]分别设为 5 min，10 min，20 min 3 个层次，滑动窗口的间隔时间设为 0 min，20 min，40 min，60 min，由图 3 所示，当滑动窗口为 0 min 时准确率远低于滑动窗口为 20 min 的效果，且随着滑动窗口的增大准确率增加，收集信息的间隔越短准确率越高。

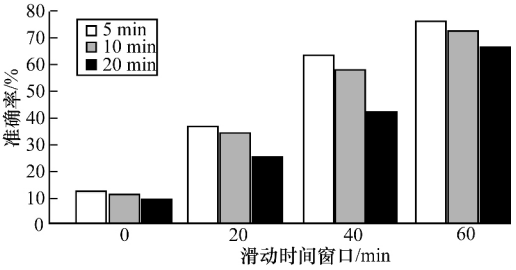

图 3　滑动窗口迭代算法结果

2) 静态特征对比

以 5 min 为收集信息时间窗口，分别设置滑动窗口为 0 min，20 min，40 min，60 min，对比是否添加静态特征(由表 1、表 2) 的检测结果。由图 4 提升并不明显，可见对于静态特征的提取存在问题，简单的静态特征不能有效地量化恶意应用带来的风险[15]，需要引入更多的静态特征，如操作流、数据流等概念，针对应用的行为特征提取出更多有效的静态特征。

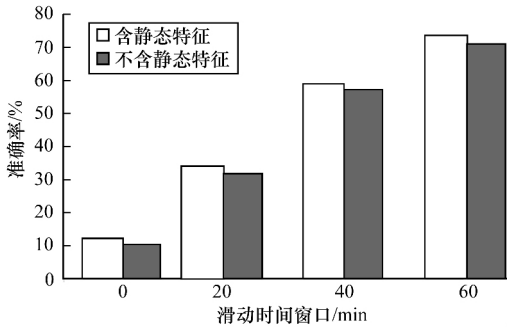

图 4　静态特征效果对比

3) 分类算法准确率对比

利用 10 交叉检验算法，分别在 10 个数据集上对比决策树、bayes、SVM、Boost、RF 算法的准确度。由图 5 利用模型融合后的分类器准确率有了较明显的提升，相比于 SVM 模型在大规模样本训练时效率下降严重、决策树易出现过拟合问题，逻辑回归模型选择不同拟合曲线对结果影响过大。贝叶斯模型无法处理基于特征组合所产生的变化结果，boost 技术在融合模型过程中，提升错误率较高的模型的权重，降低分类正确的模型的权重，最终通过加权组合弱分类器得到强分类器，且不用担心过拟合问题，在实验数据上显示出更高的准确性。

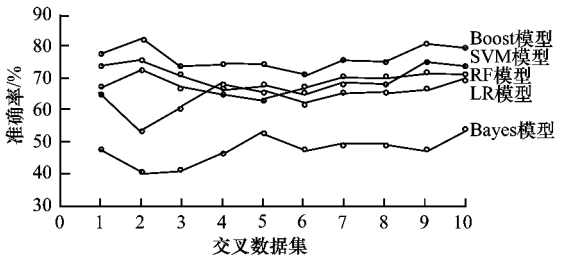

图 5　分类算法准确率对比

6　结束语

移动端 APT 攻击近年来成为一种日益流行的攻击方式，APT 攻击具有传播恶意应用、延迟攻击等特点。针对这些特点，本文提出一种新的模型，利用静态检测技术提取出终端的应用特征，优化模型对恶意应用的敏感程度。在此基础上针对 APT 的延迟攻击特性，引用滑动窗口迭代算法对模型进行优化，同时使用 Boost 技术将决策树、逻辑回归、贝叶斯等分类模型进行融合，有效提升了 APT 检测模型的准确率，并进行了实验证明模型的有效性。在今后的工作中，将基于现有静态特征进行优化，引入基于行为的数据流和控制流特征，精确量化恶意应用的行为。

参考文献

[1] Chandran S，Hrudya P，Poornachandran P. An Efficient Classification Model for Detecting Advanced Persistent Threat [C] //Proceedings of International Conference on Advances in Computing，Communications and Informatics. Washington D. C.，USA：IEEE Press，2015：333-337.

[2] Bhatt P，Yano E T，Gustavsson P. Towards a Framework to Detect Multi-stage Advanced Persistent Threats Attacks [C] //Proceedings of the 8th International Symposium on Service Oriented System Engineering. Washington D. C.，USA：IEEE Press，2014：390-395.

[3] Wu Dongjie，Mao Chinghao，Lee H M，et al. DroidMat：Android Malware Detection Through Manifest and API Calls Tracing [C] //Proceedings of the 7th Asia Joint

Conference on Information Security. Washington D. C., USA: IEEE Press, 2012: 62-69.

[4] Pehlivan U, Baltaci N, Acarturk C, et al. The Analysis of Feature Selection Methods and Classification Algorithms in Permission Based Android Malware Detection [C] // Proceedings of IEEE Symposium on Computational Intelligence in Cyber Security. Washington D. C., USA: IEEE Press, 2014: 1-8.

[5] Choudhury S, Bhowal A. Comparative Analysis of Machine Learning Algorithms Along with Classifiers for Network Intrusion Detection [C] //Proceedings of International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials. Washington D. C., USA: IEEE Press, 2015: 89-95.

[6] Alam M S, Vuong S T. Random Forest Classification for Detecting Android Malware [C] //Proceedings of International Conference on Green Computing and Communication. Washington D. C., USA: IEEE Press, 2013: 663-669.

[7] Modupe A, Olugbara O O, Ojo S O. Exploring Support Vector Machines and Random Forests to Detect Advanced Fee Fraud Activities on Internet [C] // Proceedings of the 11th International Conference on Data Mining Workshops. Washington D. C., USA: IEEE Press, 2011: 331-335.

[8] Ham H S, Choi M J. Analysis of Android Malware Detection Performance Using Machine Learning Classifiers [C] //Proceedings of International Conference on ICT Convergence. Washington D. C., USA: IEEE Press, 2013: 490-495.

[9] Quader F, Janeja V, Stauffer J. Persistent Threat Pattern Discovery [C] //Proceedings of IEEE International Conference on Intelligence and Security Informatics. Washington D. C., USA: IEEE Press, 2015.

[10] 何毓锟,李 强,嵇跃德,等. 一种关联网络和主机行为的延迟僵尸检测方法[J]. 计算机学报, 2014, 37(1): 50-61.

[11] Hong K F, Chen C C, Chiu Y T, et al. Ctracer: Uncover C&C in Advanced Persistent Threats Based on Scalable Framework for Enterprise Log Data [C] // Proceedings of IEEE International Congress on Big Data. Washington D. C., USA: IEEE Press, 2015: 551-558.

[12] Li F, Lai A, Ddl D. Evidence of Advanced Persistent Threat: A Case Study of Malware for Political Espionage [C] // Proceedings of the 6th International Conference on Malicious & Unwanted Software. Washington D. C., USA: IEEE Press, 2011: 102-109.

[13] Ramsey B W, Mullins B E, Temple M A, et al. Wireless Intrusion Detection and Device Fingerprinting Through Preamble Manipulation [J]. IEEE Transactions on Dependable & Secure Computing, 2015, 12(5): 585-596.

[14] Wolf T, Chandrikakutty H, Hu Kekai, et al. Securing Network Processors with High-performance Hardware Monitors [J]. IEEE Transactions on Dependable & Secure Computing, 2014, 12(6): 1.

[15] 刘文怡,薛 质,王轶骏. 基于网络流统计数据的伪装入侵检测[J]. 计算机工程, 2014, 40(7): 78-81.

编辑 顾逸斐

(上接第 240 页)

参考文献

[1] 谢娟英,谢维信. 基于特征子集区分度与支持向量机的特征选择算法[J]. 计算机学报, 2014, 37(8): 1704-1718.

[2] 谢娟英,高红超. 基于统计相关性与 K-means 的区分基因子集选择算法[J]. 软件学报, 2014, 25(9): 2050-2075.

[3] Ozay M, Yarman-Vural F T. Hierarchical Distance Learning by Stacking Nearest Neighbor Classifiers [J]. Information Fusion, 2016, 29(1): 14-31.

[4] Pal R, Kupka K, Aneja A P, et al. Business Health Characterization: A Hybrid Regression and Support Vector Machine Analysis [J]. Expert Systems with Applications, 2016, 49(1): 48-59.

[5] 唐 浩,李晓霞,钟 英. 基于稀疏表示的两级级联快速行人检测算法[J]. 计算机工程, 2015, 42(6): 221-226.

[6] 刘小龙,江 虹,吴 丹. 基于 CACC 的连续数据离散化改进算法[J]. 计算机工程, 2013, 39(4): 48-51.

[7] 杨锡运,孙宝君,张新房,等. 基于相似数据的支持向量机短期风速预测仿真研究[J]. 中国电机工程学报, 2012, 32(4): 35-41.

[8] 宋 佳. 模式识别综述及汉字识别的原理[J]. 科技广场, 2007(9): 133-135.

[9] 王宪保,陆 飞,陈 勇,等. 仿生模式识别的算法实现与应用[J]. 浙江工业大学学报, 2011, 39(5): 71-74.

[10] 查 九,李振博,徐桂琼. 基于组合相似度的优化协同过滤算法[J]. 计算机应用与软件, 2014, 31(12): 323-328.

[11] 李晓宇,张新峰,沈兰荪. 支持向量机(SVM)的研究进展[J]. 测控技术, 2006, 25(5): 7-12.

[12] 宋 晖,薛 云,张良均. 基于 SVM 分类问题的核函数选择仿真研究[J]. 计算机与现代化, 2011(8): 133-136.

[13] 蒋龙泉,鲁 帅,冯 瑞,等. 基于多特征融合和 SVM 分类器的植物病虫害检测方法[J]. 计算机应用与软件, 2014, 31(12): 186-190.

[14] 詹 毅. 朴素贝叶斯算法和 SVM 算法在 Web 文本分类中的效率分析[J]. 成都大学学报(自然科学版), 2013, 32(1): 50-53.

[15] 杨忠强. 基于属性加权和归约的朴素贝叶斯算法研究[D]. 南宁: 广西大学, 2013.

编辑 索书志