基于用户行为周期的移动设备异常检测方法①

吴志忠[1], 周学海[2]

[1](中电海康集团有限公司 中国电科(杭州)物联网研究院, 杭州 310012)
[2](中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘 要: 本文提出了一种分布式的移动设备异常检测系统, 该系统采用客户端-服务器架构, 客户端程序在移动设备上持续提取特征并传送给服务器, 服务器使用异常检测算法分析特征. 根据人类日常活动的规律性以及用户使用移动设备的周期性, 我们还提出了一种基于用户行为周期的异常检测方法, 通过比较待检测特征向量和以往周期相近时间段的特征向量集的距离即可判定该特征向量是否异常, 向量比较时采用不受特征间关联以及特征取值范围影响的马氏距离作为距离衡量的标准. 实验证明我们采用的移动设备异常检测系统框架和检测方法能够有效提高对移动设备恶意程序的检测率.

关键词: 异常检测; 行为周期; 马氏距离; 特征提取

User BehaviorCycle-Based Statistical Approach for Anomaly Detecting on Mobile Devices

WU Zhi-Zhong[1], ZHOU Xue-Hai[2]

[1](Institute for Internet of Things, China Electronic Technology Group Corporation, Hangzhou 310012, China)
[2](Department of Computer Science, University of Science and Technology of China, Hefei 230027, China)

Abstract: In this paper, we present a distributed anomaly detection system for mobile devices. The proposed framework realizes a client-server architecture, the client continuously extracts various features of mobile device and transfers to the server, and the server's major task is to detect anomaly using state-of-art detection algorithms. According to the regularity of human daily activity and the periodic of using mobile device, we also propose a novel user behavior cycle based statistical approach, in which the abnormal is determined by the distance from the undetermined feature vector to the similar time segments' vectors of previous cycles. We use the Mahalanobis distance as distance metric since it is rarely affected by the correlate and value range of features. Evaluation results demonstrated that the proposed framework and novel anomaly detection algorithm could effectively improve the detection rate of malwares on mobile devices.

Key words: anomaly detection;behavior cycle; mahalanobis distance; feature extraction

1 引言

随着无线通信技术、网络技术以及嵌入式技术的不断发展, 移动设备的计算和网络功能不断增强, 成为结合了电信网以及因特网的统一通信设备, 并逐渐成为个人的信息中心. 随着移动设备用户数量的飞速增长, 其上的恶意软件[1,2]也越来越多, 不仅危害到移动设备本身的安全, 也危害到用户的隐私和财产安全.

在移动设备安全防护技术中, 操作系统安全加固、数字签名、杀毒软件等还不足以遏制恶意软件, 为了能够在第一时间发现恶意软件的攻击, 有效的避免和减少对移动设备以及用户的危害, 国内外许多研究都将入侵检测技术引入到移动设备的安全防护领域. 入侵检测系统有两种类型, 一是特征检测, 类似杀毒软件, 只能检测已知的恶意软件, 并需要频繁地更新特征库; 二是异常检测, 不需要定义异常行为, 能够检测到未知入侵, 缺点是容易出现漏报和误报. 由于移动设备的网络特性以及恶意软件频繁更新的特点, 移动设备入侵检测技术的研究大部分集中在异常检测.

① 基金项目:国家自然科学基金(61272131)
收稿时间:2014-08-10;收到修改稿时间:2014-09-15

本文提出一种分布式移动设备异常检测系统, 该系统采用客户端-服务器架构, 客户端负责持续监控移动设备, 提取移动设备的特征向量, 并将向量传送到服务器; 服务器端运行异常检测算法, 分析特征向量并给出检测结果, 若检测结果为异常则发送警报给移动设备客户端.

本文还提出了一种基于用户行为周期的异常检测方法(简称 CBS). 相关研究证明人类日常活动具有周期性, 因此用户使用移动设备往往也遵循周期性规律, 比如 A 用户每天早上七点半坐公车去上班, 在公车上他习惯性的使用智能手机看早新闻. 基于用户行为周期的异常检测方法的核心思想就是将用户的行为周期作为重要参数应用到特征向量的分析和比较中, 除了能够发现普通的异常行为还能发现违反时间周期的异常行为.

2 相关工作

在移动设备异常检测技术中, 很大一部分研究使用的异常检测算法是直接使用 PC 领域主机入侵或网络入侵的检测算法[3,4], 或是对常用的数据挖掘分类算法、机器学习算法稍加改进[5,6], 也有少数将其它领域的技术引入到移动设备异常检测中.

Cheng 等人提出的智能手机恶意程序检测系统中使用的异常检测算法是基于统计的方法, 它会持续记录一个网络流量阈值 Uthreshold, 该阈值是通过一个滑动的窗口计算的平均值, 当每天的网络流量 Utoday 超过 Uthreshold 时, 就会触发预警[7]. Timothy 和 Theresa 提出的基于电池信息的智能手机入侵保护系统使用的也是一种基于统计的异常检测方法, 称为动态阈值计算方法 DTC(Dynamic Threshold Calculation), 根据不断检测设备进程、背景灯、系统状态等信息, 动态的调节功耗阈值, 当超出阈值时将发送警报. 使用 DTC 方法可以有效的降低误报率, 因为 DTC 算法考虑更多正常的设备功耗活动, 仅当阈值真正被恶意活动超出时才会发出警报[8].

Bose 等提出的基于行为的智能手机入侵检测系统中, SVMs 被用于训练数据集并给出分类器, 分类的准确率能够高达 96%[9]. Shamili 等在其提出的移动设备异常检测系统中使用一种分布式的 SVMs 算法, 通过分布式解决方案, 客户端可以并行且及时的计算以及更新各自的支持向量, 从而有效降低机器学习算法的开销[10]. Shabtai 等在其提出的异常检测系统框架中测试了常用的数据挖掘分类算法, 并给出了最适合其系统的分类算法以及特征选择算法[11].

Schmidt 等在其基于网络流量的智能手机异常检测系统中使用的机器学习算法包括自组织映射和人工免疫系统, 并提出一种线性预测算法(Linear Prediction Algorithm), 根据待检测向量之前的 4 个特征向量预测出一个预估向量和待检测向量进行比较, 从而判定是否异常[12]. Schmidt 在其基于函数调用的智能手机异常检测系统中又提出一种机器学习算法, 称为质心机(Centroid Machine), 将可执行文件的函数调用映射到函数集空间中, 并定义两个函数特征序列的距离(欧几里德距离), 然后比较待检测的可执行文件函数序列到恶意和正常两个函数集的质心的距离来判定是否为恶意软件[13].

Yang 将游戏理论应用到移动设备异常检测中, 并将一个基于纳什平衡(Nash Equilibrium)的防御机制应用到安全服务器[14].

麻省理工学院媒体实验室(MIT Media Lab)有一个叫做“现实挖掘(Reality Mining)”的项目, 主要是通过安装在智能手机上的监控客户端来收集和用户活动相关的数据, 比如蓝牙、短信以及所在扇区等, 然后再从这些现实的数据中挖掘出用户的行为规律. 他们通过手机收集的数据发现, 用户的行为具有一定规律性, 因为用户总是会按照一定的时间工作、学习, 并且会有各种有规律的生活工作习惯, 因此可以通过建立模型来预测用户的行为[15].

3 系统架构

图 1 给出了本文所提出的智能手机异常检测系统的整体框架. 系统采用客户端-服务器模式, 客户端程序主要负责监控智能手机并收集异常检测所需的特征, 再通过 GPRS 网络、3G 网络或 WIFI 将特征向量传送到远程服务器. 服务器采用的是分布式架构, 包括一台通信服务器以及若干检测服务器, 通信服务器也是主服务器, 它会将收到的待检测特征向量根据所属的客户端号以及当前服务器的负载状况分发给恰当的检测服务器, 检测服务器会使用异常检测算法分析特征向量给出检测结果并将结果反馈给主服务器, 主服务器将这些结果保存并处理, 若检测结果为异常则发送警报给对应客户端.

图 2 描述了移动设备异常检测系统的客户端和服务器组件图. 客户端是基于 Android 操作系统的, 主要包括特征提取器、通信模块和图形用户界面; 服务器负责判定特征向量是否异常, 主要包括数据存储模块、检测模块、通信模块、客户端管理模块和图形用户界面.

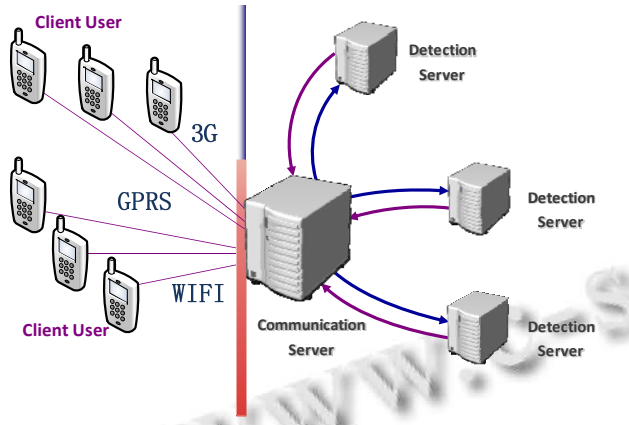

图 1　移动设备异常检测系统框架

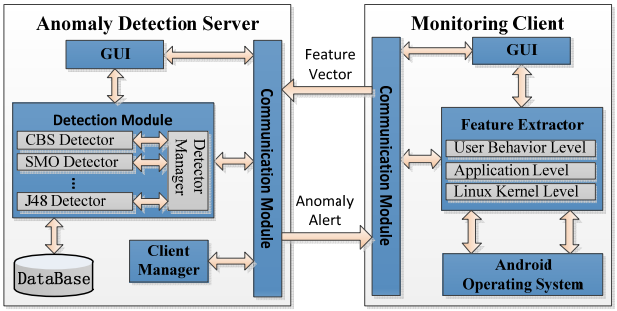

图 2　客户端及服务器组件图

4　算法描述

表 1　符号术语

τ	特征提取时间间隔, 实验中采用 30 秒
$V_i^T(k)$	设备 i 在时刻 k 收集的特征向量
$V(k)$	时刻 k 的特征向量, 设备和时间间隔均为默认值, 故略
$L(k_1,k_2)$	向量 k_1 和向量 k_2 之间的距离
$W_0(k)$	特征向量$V(k)$相关的观察窗口
$W_1(k)$	特征向量$V(k)$相关的取样观察窗口
$D_I(k)$	特征向量$V(k)$相关的内部距离
$D_E(k)$	特征向量$V(k)$相关的外部距离
α	比较系数

对于同一设备上收集的两个不同时间的特征向量$V(k_1)$和$V(k_2)$, $L(k_1,k_2)$表示两个向量间的距离或者说是相似度. 选择的距离衡量的标准将下文描述. 在我们的算法中, 异常是指待检测的特征和其相关的正常特征集在统计数值上达到了一定的偏离, 这个相关的正常特征集合又称作观察窗口, 特征$V(k)$的观察窗口为$W_0(k)$, 公式如(1)所示:

$$W_0(k)=\{k_j:k-nT-t/2\le k_j\le k-nT+t/2\},n\in[1,N] \tag{1}$$

其中 T 表示周期长度, n 表示周期数, t 表示 k 时刻附近的时间段长度, 观察窗口是指之前 n 个周期 k 时刻附近的向量. 公式中 n 不取 0, 意味着我们去除了当前周期时刻 k 附近的特征向量, 这样做是为了消除异常中慢启动现象. 因为我们收集特征的部分属性是累积量的形式, 这些属性的值的变化将会是一个慢慢增长的过程, 我们称其为慢启动, 如果将同周期时刻 k 附近的特征向量纳入观察窗口, 那么待检测向量和观察窗口间的距离会因此缩小, 异常判定的准确率也会受影响, 所以 n 只能取$[1,N]$.

由于观察窗口$W_0(k)$中的向量数量会非常大, 属性值的范围也很广, 为了减少不必要的计算, 我们从窗口$W_0(k)$取样出距离待检测向量$V(k)$最近的向量(取样比例为 25%), 组成取样观察窗口$W_1(k)$. 待检测向量$V(k)$和取样窗口中向量$\{V(k'),k'\in W(k)\}$的比较主要计算两个距离, 第一个是内部距离(internal distance), 用$D_I(k)$表示, 是指取样窗口中任意两两向量间距离的最大值, 公式如下:

$$D_I(k)=\{L(k_i,k_j),k_i,k_j\in W_1(k),k_i\neq k_j\} \tag{2}$$

另一个是外部距离(external distance), 用$D_E(k)$表示, 是指待检测向量$V(k)$和取样窗口中所有向量间的距离中的最小值, 公式如下:

$$D_E(k)=\{L(k,k_i),k_i\in W_1(k)\} \tag{3}$$

异常判定基于内部距离和外部距离的比较, 如果$D_E(k)\le\alpha D_I(k)$, 那么待检测向量$V(k)$被判定为正常, 反之则判定为异常. 在判定中我们引入了比较系数α, 它能够控制待检测向量要判定为正常的最大偏离距离, 因此比较系数可以调节算法检测的敏感度, 可以通过比较系数找到检测率和误判率的最佳折中点.

算法过程如图 3 所描述, 首先初始化相关参数α,t,N,M, 其中 M 初始化为$\emptyset$, 再通过训练集数据计算

周期 T, 在实际应用中, 周期 T 大多为一天. 对于待检测向量$V(k)$, 先确定观察窗口$W_0(k)$, 然后取样后确定取样窗口$W_1(k)$, 接下来计算内部距离$D_I(k)$和外部距离$D_E(k)$, 通过比较两个距离的大小判定向量$V(k)$是否异常.

INITIATION: Set values of α, t, N, M
Compute time period$T$

START:

1) Obtain $V(k)$
2) Define the observation window $W_0(k)$ by Eq. (1)
3) Select $W_1(k)$ from$W_0(k)\backslash M$ by distance to $V(k)$
4) Calculate the distance $D_I(k)$and $D_E(k)$
5) If $D_E(k) > \alpha D_I(k)$
Rise alarm;
Set $M = M \cup \{k\}$;
Else $V(k)$is normal
6) Increase k by one and go-back to 1)

图 3 CBS 异常检测方法伪代码

最常用的距离衡量标准是欧氏距离, 在欧氏距离中每个特征属性对距离的贡献是同等的. 但是在我们的特征集中, 属性的类别多样(数值型、类别型), 取值范围也具有很大差异, 很难归一化, 因此欧式距离并不太适用. 我们选择马氏距离(Mahalanobis distance)作为算法中的距离衡量标准, 与欧式距离不同的是它考虑到各种特性之间的联系并且是尺度无关的, 两点之间的马氏距离与原始数据的测量单位无关, 并且由标准化数据和中心化数据(即原始数据与均值之差)计算出的二点之间的马氏距离相同. 当然马氏距离也有缺点, 比如要求样本总数大于样本维数, 距离计算建立在总体样本基础之上, 这些在我们的实验中恰恰是可以忽略的.

两个服从同一分布并且其协方差矩阵为 S 的随机变量$\vec{x}$和$\vec{y}$的马氏距离计算如下:

$$L_M(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1}(\vec{x}-\vec{y})} \quad (4)$$

协方差矩阵 S 具有逆矩阵S^{-1}必须要满足以下两个条件: 1) 样本数大于向量的位数, 这个条件一般都很容易满足. 2) 样本不在同一个 n-1 维的空间中(n 为向量的位数). 如果协方差矩阵 S 没有逆矩阵S^{-1}, 就用单位矩阵代替, 此时马氏距离就退化为欧式距离.

5 实验分析

5.1 数据集的构建

由于智能手机异常检测领域目前还没有公开的得到一致认可的数据集, 因此需要自行构建试验用数据集, 数据集要尽量保证公平性.

我们在 Android 应用超市中选择了 32 个最为常用的应用程序, 使用三台三星智能手机收集数据. 32 个应用程序将分别安装到 3 台智能手机中, 并由用户分别正常使用这些应用程序各 1 小时左右, 在运行这些程序时我们的异常检测系统的客户端程序会在手机后台运行, 并每隔 30 秒提取一次特征, 这样每个应用程序在每台设备上都将收集 120 条左右特征, 总共有 1 万多条特征, 这些特征向量都标记为正常. 另外这 3 台 android 手机将分别交给三个用户日常使用, 同时我们的客户端程序会一直在后台运行提取特征, 大约收集 3 个月时间的日常特征向量. 在收集的过程中, 用户完全按照平时生活规律使用该手机, 以保证数据的公正性.

根据 Symbian、Windows Mobile 等智能手机恶意程序的意图, 可以将恶意程序分为如下三类: DoS 攻击, 窃取用户隐私以及消耗用户资费. 因此, 我们开发了 4 个和恶意程序意图类似的概念性恶意程序用于实验. 分别是 1)浮点计算器: 一种 DoS 攻击型的恶意程序, 程序启动后便会在后台循环进行大量的浮点运算, 大量占用 CPU 资源, 可以导致 Android 设备卡死甚至瘫痪; 2)短信发送器: 后台短信发送程序, 开机自启动, 通过 API 不断向指定号码发送短信; 3)后台下载器: 程序安装后会在后台自动从指定地址下载文件; 4)后台上传器: 启动后会自动将文件上传到指定的地址. 4 种恶意程序均分别在三台智能手机上运行 1 小时, 收集异常数据集.

5.2 实验结果

首先给出的是用户活动周期性的依据, 我们统计了同一设备连续两周的网络下载量 *NET_RECV*, 以 3 个小时为一段, 并且进行归一化(最大值为 1), 结果如图 4 所示, 横坐标为时间, 纵坐标为归一化后的值. 图中每天的 *NET_RECV* 曲线都类似, 夜间的流量最低,

中午休息以及晚上休息时间段的值为最高, 其它相同时间段也几乎相同; 第一周的周一至周五和第二周的周一至周五在峰值较为接近, 两周的周末峰值也接近. 说明该用户工作日使用手机遵循一定规律, 周末休息使用设备也有规律.

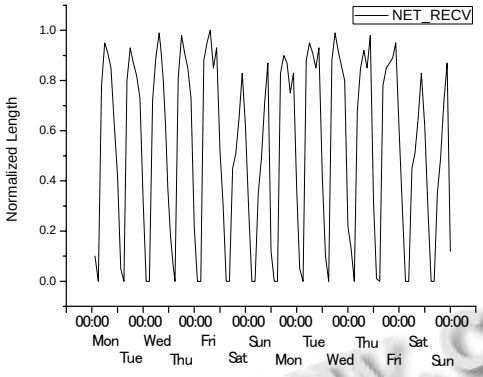

图 4 同一设备连续两周的 NET_RECV 值

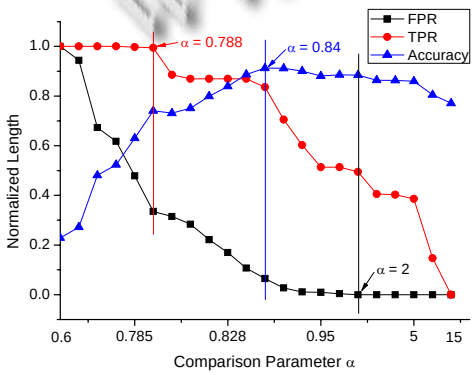

图 5 CBS 关于比较系数 α 的性能

图 5 给出 CBS 异常检测方法的性能, 横坐标为比较系数 α, 纵坐标为百分比. 我们将准确率、检测率以及误判率放在同一个图中, 方便比较 α 对检测性能的影响并找出最佳的 α 取值. 当 $\alpha \le 0.60$ 时, 所有特征向量都被判定为异常, 检测率和误判率都为 100%, 准确率最低, 仅 20%左右. 当 $\alpha \ge 15$ 时, 所有特征向量都被判定为正常, 此时检测率为 0. 当 $0.6 < \alpha < 15$ 时, 检测率从 100%下降到 0%, 误判率也从 100%降到 0%, 而准确率是则是先上升后下降. 若检测性能要侧重准确率, 则取 α=0.84, 此时准确率高达 91.23%, 检测率能够达到 83.61%, 误判率为 6.5%, 还可以接受. 若要更侧重检测率, 则可以取 α=0.788, 此时可以保证高达 99.44%的检测率, 但是相应的误判率也上升到了 33.52%, 整体的检测率为 74.02%, 偏低. 若需要较低的误判率, 可以取 α=2, 误判率近乎于 0, 检测率有 50%左右.

表 2 给出 CBS 与常用分类算法的检测性能比较, CBS 异常检测算法整体性能高于常用的分类算法, 并且可以根据需要调整比较系数 α 的取值, 找到满意的准确率和误判率的折中点.

表 2 CBS 与常用分类算法的检测性能比较

算法	准确率	检测率	误判率
J48	0.6093	0.3972	0.3278
BN	0.5146	0.9972	0.6285
RF	0.7700	0.1500	0.0016
KNN	0.8685	0.4278	0.0008
CBS α=0.788	0.7402	0.9944	0.3352
CBS α=0.84	0.9123	0.8361	0.0651
CBS α=2	0.8843	0.4944	0

6 总结

本文首先提出了一个分布式移动设备异常检测系统框架, 智能手机客户端负责收集特征, 服务器负责对特征向量进行异常检测分析. 其次, 我们发现人类日常活动具有周期性, 用户使用智能手机也具有周期性的规律, 因此, 我们提出了一种基于用户行为周期的异常检测方法, 通过比较待检测特征向量和以往周期相近时间段的特征向量集的距离, 判定该特征向量是否异常, 判定时采用不受特征间关联以及特征取值范围影响的马氏距离作为距离衡量的标准. 实验证明该异常检测方法能够有效检测到各种情况下的后台恶意程序.

在未来的研究中, 我们将持续改进该异常检测方法, 特别是当异常检测系统对准确率和检测率要求较高时, 有效降低误判率.

参考文献

1 Schmidt AD, Albayrak S. Malicious Software for Smartphones[Technical Report]. TUB-DAI 02/08-01, 2008.

2 Schmidt AD, Schmidt HG, Batyuk L. Smartphone malware evolution revisited: Android next target? Malicious and Unwanted Software (MALWARE) 4th International Conference. 2009.

3 Patcha A, Park JM. An overview of anomaly detection

techniques: Existing solutions and latest technological trends. Computer Networks, 2007, 51(12): 3448–3470.

4 Chandola V, Banerjee A, Kumar A. Anomaly Detection: A Survey. ACM Computing Surveys, 2009, 41(3): 151–158.

5 Shon T, Kim Y, Lee C, Moon J. A machine learning framework for network anomalydetection using SVM and GA. IEEE Workshop on Information Assurance and Security. US Military Academy, West Point, NY. 2005.

6 Li Y, Guo L. An efficient network anomaly detection schemebased on TCM-KNN algorithm and data reduction mechanism. IEEE Workshop on Information Assurance and Security. US Military Academy. West Point, NY. 2007. 20–22.

7 Cheng J, Wong SHY, Yang H, Lu SW. SmartSiren: Virus detection and alert for smartphones. Proc. of MobiSys. 2007. 258–271.

8 Timothy KB, Theresa M. Mobile Device Profiling and Intrusion Detection using Smart Batteries. HICSS, 2008.

9 Bose A, Hu X, Shin KG, Park T. Behavioral detection 9 of malware on mobile handsets. Proc. of the 6th International Conference on Mobile Systems, Applications and Services. New York. 2008. 225–238.

10 Shamili AS, Bauckhage C, Alpcan T. Malware detection on mobile devices using distributed machine learning. 2010 20th International Conference on Pattern Recognition (ICPR). 2010.

11 Shabtai A, Kanonov U, Elovici Y, Glezer C, Weiss Y. "Andromaly": A behavioral malware detectionframework for android devices. Journal of Intelligent Information System, 2012, (38): 161–190.

12 Schmidt AD, Peters F, Lamour F, Scheel C, Camtepe SA, Albayrak S. Monitoring smartphones for anomaly detection. Mobile Networks & Applications, 2009, 14(1): 92–106.

13 Schmidt AD, Clausen JH, Camtepe A, Albayrak S. Detecting Symbian OS malware through static function call analysis. 2009 4th International Conference on Malicious and Unwanted Software (MALWARE). 2009.

14 Yang F, Zhou XH, Jia GY, Zhang QY. A non-cooperative game approach for intrusion detection in smartphone systems. Communication Networks and Services Research Conference (CNSR). Eighth Annual. 2010.

15 Eagle N, Pentland A, Lazer D. Inferring social network structure using mobile phone data. Proc. of the National Academy of Sciences, 2009, 106(36): 15274–15278.