基于频率特征向量的系统调用入侵检测方法

张莉萍[1] 雷大江[2] 曾宪华[2]

(重庆邮电大学移通学院计算机系 重庆 401520)[1] (重庆邮电大学计算机学院 重庆 400065)[2]

摘 要 针对基于系统调用的异常入侵检测方法中较难抽取正常系统调用序列的特征库问题,提出将正常系统调用序列抽取出的子序列的频率特征转换为频率特征向量,并以此作为系统调用序列的局部和全局特征;为了保证对大规模数据集检测的准确率和速度,采用一类分类支持向量机(SVM)分类器进行学习建模,利用先前建立的特征库进行训练,建立入侵检测分类模型,最后对于待检测序列进行异常检测。在多个真实数据集上与已有的异常入侵检测方法进行比较实验,结果表明本文提出的方法的多个异常检测指标都都优于已有方法。

关键词 系统调用,入侵检测,特征向量,支持向量机

中图法分类号 TP393 **文献标识码** A

System Calls Based Intrusion Detection Method with Frequency Feature Vector

ZHANG Li-ping[1] LEI Da-jiang[2] ZENG Xian-hua[2]

(Department of Computer Science and Technology, College of Mobile Telecommunication Chongqing University of Posts and Telecom, Chongqing 401520, China)[1]

(College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[2]

Abstract In order to solve the problem of extracting feature library and detecting anomaly system calls slowly in intrusion detection methods, this paper proposed a novel two phrase intrusion detection method. In the first phrase, we extracted subsequences from normal system calls and calculated the frequency of the subsequences, and transformed the frequency feature into the frequency feature vector including continue numeric number. In order to improve the accuracy and speed of detecting anomaly system calls, the paper adopted one-class classification support vector machine(SVM) to build the detecting model, which uses the feature vector library to build the model. Finally, we conducted extensive experiment to evaluate the performance of our proposed method. The results show that our proposed method is superior to the existing methods in many evaluation metrics.

Keywords System calls, Intrusion detection, Frequency feature vector, Support vector machine

1 引言

随着网络技术的发展和应用,计算机网络安全成为一个重大问题。入侵检测作为一种网络安全防御方法,是计算机安全研究领域的一个重要分支。入侵检测技术大体上可以分为异常入侵检测和误用入侵检测[1]。误用入侵检测是基于模式匹配的检测技术,它首先对已知的入侵行为建立规则库,入侵检测时将用户的行为与规则库中的规则进行对比,将匹配成功的用户行为标记为入侵行为。尽管误用入侵检测对已知的攻击行为有很好的检测结果,但其主要缺陷是对预先位置的入侵行为无法进行检测。异常入侵检测首先根据正常系统调用构造正常模式库,入侵检测时将用户的行为与正常模式进行对比,偏离正常模式大的行为被认为是入侵行为。尽管异常入侵检测能够检测未知的入侵行为,但是对于异常的定义以及度量使这种方法遇到了最大挑战。

1996年,Sundaram认为系统调用序列用于误用入侵检测时,可以克服击键序列所存在的某些缺点[2]。同年,Forrest等指出,特权进程的系统调用局部短序列具备明显的一致性,可以区分不同的特权进程、敏感进程的异常行为,并首次提出基于系统调用的异常入侵检测技术[3]。基于系统调用的异常入侵检测技术由于鲁棒性好、误报率低以及实现简单等特点,已经被国内外学者进行了广泛研究。迄今为止,基于系统调用的入侵检测主要有基于数据挖掘的方法、基于机器学习的方法、基于统计学的方法以及基于有限状态机的方法等[4]。基于系统调用的异常入侵检测分为两个阶段:第一阶段,扫描正常行为轨迹,根据系统调用序列构建正常行为模式特征库;第二阶段,扫描可能含有异常行为的新轨迹,查找正常行为模式特征库中未出现或者与正常行为偏离度较大的模式,将包含以上模式的系统调用行为标记为入侵行为[5]。国内外学者对利用数据挖掘技术进行系统调用的异常入侵检进行了深入的研究。Liao提出了将系统调用序列中的每个调用项类比于文本分类中的词,在扫描阶段用kNN技术对系统调用行为

本文受国家自然科学基金(61075019),重庆邮电大学移通学院青年教师基金(20110302)资助。

张莉萍(1983—),女,硕士,讲师,主要研究方向为网络信息安全、数据挖掘、人工智能等,E-mail:zhanglipingrose@163.com;雷大江(1979—),男,博士,讲师,主要研究方向为数据挖掘、并行计算等;曾宪华(1973—),男,博士,副教授,主要研究方向为模式识别、机器学习等。

进行异常识别[6]。Rawat 提出一种与 Liao 类似的方法，即用余弦相似度来计算待检测系统调用与模式特征库中系统调用的距离，用这个距离来衡量待检测系统调用与正常行为的系统调用的偏离程度，偏离程度最大的被认为是入侵行为[7]。Jecheva 提出先将每个系统调用序列划分成等长的子序列，然后利用 K-D 树存储每个子序列；在异常检测阶段也使用 kNN 技术进行异常检测，用两种距离度量 Jaro Distance(JD)和 Jaro-Winkler Distances(JWD)来计算待检测系统调用序列与正常行为模式特征库中系统调用序列的偏离度[8]。国内学者吕锋提出利用系统调用在所有进程的系统调用中出现的频率占所有系统调用出现频率和的比率作为权重的系统调用空间向量模型，然后通过互信息量作为度量抽取系统调用特征，最后利用 kNN 技术进行异常系统调用的检测[9]。通过对国内外学者研究方法的分析，采用基于 kNN 技术的系统调用序列异常入侵检测方法主要难点为：第一，正常系统调用序列的特征提取方法；第二，利用特征库建立有效的分类器对系统调用序列进行分类。目前主要研究方法直接将系统调用序列的子序列之间的相似度的和作为系统调用序列的相似度，而忽略了子序列之间的公共序列片段以及子序列在正常系统调用序列构造的特征库中出现的频率，因此会直接影响异常入侵检测的检测率和误报率。本文基于 STIDE 的系统调用异常入侵检测模式[10]，在第一阶段即特征提取阶段，将每个正常系统调用序列依据其包含的子序列在正常系统调用子序列库中出现的频率转换为固定长度的数值型频率特征向量；在第二阶段，将第一阶段提取的特征库作为正类利用一类分类支持向量机 SVDD[11]建立分类器，然后将待检测系统调用序列按照第一阶段的特征转换方式也转换为固定长度的数值型频率特征向量，利用分类器将系统调用序列分类为正常序列和异常序列。与 STIDE 以及目前效果最好的基于 kNN 的异常入侵检测方法在真实数据上进行比较实验，实验结果表明本文所提出的方法在异常检测评价指标上优于已有的方法，证明了本文提出的方法的可行性和有效性。

2 特征抽取

2.1 系统调用

Linux 内核中设置了一组用于实现各种系统功能的子程序，称为系统调用。用户或者进程可以通过系统调用实现对系统资源的访问，如果系统调用序列中出现了不期望的行为，表明黑客可能正在入侵你的系统。尽管黑客可能会将程序或者服务的运行分解，或者以非用户或者管理员意图的方式操作，但是程序对系统资源的访问在程序执行时产生的系统调用序列中可以清楚地显示出来。因此分析系统调用序列可以作为异常入侵检测的一种手段。

表 1 给出了美国新墨西哥大学计算机入侵检测项目[12]所用到的数据集中的 3 个系统调用序列。在本文中，一个进程的行为可以用一个系统调用序列来表示。

表 1 异常入侵检测用到的系统调用序列

序列	序列名称
S_1	open,read,mmap,mmap,open,read,mmap…
S_2	open,mmap,mmap,read,open,close…
S_3	open,close,open,close,open,mmap,close…

2.2 符号与术语

为了方便描述我们提出的方法，在这一节中首先给出一些定义。表 2 包含了我们用到的一些主要定义的表示符号。由于采用的是基于数据挖掘的技术，在叙述中称第一阶段的正常系统调用序列为训练数据集；称第二阶段的待检测系统调用序列为测试数据集，在上下文不产生混淆的情况下，我们在后文中不作特殊说明。

表 2 本文中用到的符号

符号	符号含义
T	正常系统调用序列训练数据集
S	待检测系统调用序列测试数据集
T_{subs}	正常子序列构成的特征库
p	频率特征向量的维数
$\tilde{T}$	训练数据集的 p 维特征向量矩阵
$\tilde{S}$	测试数据集的 p 维特征向量矩阵
T_j	训练数据集的第 j 个系统调用序列
S_i	测试数据集的第 i 个系统调用序列
$\tilde{T}_j$	与 T_j 对应的 p 维特征向量
$\tilde{S}_i$	与 S_i 对应的 p 维特征向量

2.3 相关定义与特征提取方法

为了从每个正常系统调用序列中抽取子序列，本文采用了经典的滑动窗口技术。我们称长度为 l 的沿着序列前后移动的窗格为滑动窗口。滑动窗口从序列的第一个调用滑动到最后一个调用，产生一系列的子序列，每个子序列的长度为 l。为了叙述的方便，我们给定滑动窗口的长度为一个固定大小的参数 l，由滑动窗口抽取的每个子序列定义为 w。

定义 1(滑动窗口子序列) 给定一个序列 T_j，其长度为 m_j。假如我们用长度为 $l(l \ll m_j)$ 的滑动窗口从中抽取子序列，可以得到 m_j-l+1 个长度为 l 的子序列，每个子序列我们记为 w_{ji}。定义序列的滑动窗口子序列为：

$$W_j=(w_{j1},w_{j2},\cdots,w_{j,m_j-l+1}) \tag{1}$$

定义 2(子序列频率特征) 每个长度为 l 的子序列 w_{ji}，都赋予一个频率值 f_{ji}。对于整个正常系统调用序列集合，对每个系统调用序列首先抽取其子序列，将所有的子序列形成子序列训练集 T_{subs}，称其为特征库。对于 W_j 中所包含的每个子序列 w_{ji}，在特征库 T_{subs} 中计算其出现的频率值 f_{ji}。我们定义序列 T_i 的子序列频率特征为：

$$F_j=(f_{j1},f_{j2},\cdots,f_{j,m_j-l+1}) \tag{2}$$

其中 f_{ji} 计算公式如下：

$$f_{ij}=\sum_{w_k\in T_{subs}} nLCS(w_k,w_{ji}) \tag{3}$$

其中，$nLCS$ 表示两个子序列之间的最大公共子串相似度[13]。如果两个子序列最大公共子串相似度 $nLCS$ 小于 0.5，则 $nLCS$ 取 0；在区间[0.5,1]之间原值输出，计算公式如下：

$$nLCS(w_k,w_{ji})=\frac{LCS(w_k,w_{ji})}{\sqrt{|w_k||w_{ji}|}} \tag{4}$$

定义 3(频率特征向量) 将 T_j 的子序列频率特征映射为 p 维的连续数值型特征向量，其中 p 为给定的固定参数值。定义序列 T_j 的频率特征向量为：

$$\tilde{T}_j=(c_{j1},c_{j2},\cdots,c_{jp}) \tag{5}$$

为了将子序列频率特征映射为 p 维连续数值型特征向量，我们采用如下特征映射方法：

第 1 步 首先依据定义 2，从序列 T_j 中抽取的每个子序列相对于子序列训练集的频率被计算出来，这些频率组合成 F_j。然后，将 F_j 包含的每个频率值 f_{ji} 划分到 p 个区间，并累计属于这个区间的频率值个数。对于频率值 f_{ji} 小于 1 的子序列，因为其特殊性，第一个区间[0,1]用于存储频率值小于

1 的子序列的累计个数。其他的 $p-1$ 个区间从 $f_{ji}=1$ 到 $\max(F_j)$等距划分，其中 $\max(F_j)$表示序列 T_j 所包含的在子序列训练集中出现频率最高子序列的频率值。针对 p 个区间，可以按照如下公式定义：

$$bin_{jk}=\begin{cases}[0,1), & k=1\\ (1+(k-2)*\dfrac{\max(F_j)}{p-1},(k-1)*\dfrac{\max(F_j)}{p-1}], & 2\leqslant k\leqslant p\end{cases} \tag{6}$$

式中，bin_{jk}表示第 k 个频率值区间。我们计算落入到每个频率值区间的频率值 f_{ji} 的累积个数，并记为 z_{jk}，其中 $1\leqslant k\leqslant p$，因此得到未归一化的频率特征向量 $Z_j=(z_{j1},z_{j2},\cdots,z_{jp})$。

第 2 步　对 Z_j 进行归一化处理，将其包含的每个元素除以 T_j 包含的子序列的个数 m_j-l+1，归一化为 0 到 1 之间的连续数值型。根据定义 3，$\tilde{T}_j$ 中的每个元素由如下归一化公式得到：

$$c_{jk}=\frac{z_k}{m_j-l+1} \tag{7}$$

定义 4(相似度度量)　因为每个系统调用序列转化为数值型的频率特征向量，故计算系统调用序列之间的相似度可以直接采用欧式距离，公式如下：

$$d(\tilde{T}_x,\tilde{T}_y)=\sqrt{\sum_{k=1}^{p}(c_{xk}-c_{yk})^2} \tag{8}$$

其中，$\tilde{T}_x=(c_{x1},c_{x2},\cdots,c_{x,p})$和 $\tilde{T}_y=(c_{y1},c_{y2},\cdots,c_{y,p})$分别为两个频率特征向量。

定义 5(高斯核函数)　本文中选用高斯核函数取代特征空间中两个对象的内积运算，表示如下：

$$K(x_i,x_j)=\exp(-\frac{1}{2\sigma^2}\parallel x_i-x_j\parallel^2) \tag{9}$$

式中，σ 表示核函数的宽度，$\parallel x_i-x_j\parallel$ 表示向量 x_i-x_j 的范数。

3　异常入侵检测

按照异常入侵检测模式，我们提出的异常入侵检测方法分为两个阶段。第一阶段主要工作为构造特征库，并且将训练集和测试集中的系统调用序列转换为频率特征向量。在第二阶段利用 SVDD 进行异常检测。SVDD(Support Vector Data Description)是基于支持向量机理论和单类分类思想的一种分类模型[11]，其思想是寻找一个超球面，使其半径尽可能小，同时包含的正类的训练样本数尽可能多。一分类支持向量机建立分类器只需要正类样本，与系统调入侵检测模式吻合，本文在特征提取后利用一分类支持向量机建立入侵检测分类器。

$\{x_i\mid x_i\in R^d,i=1,2,\cdots,n\}$为输入空间上训练数据集，寻找最优超球体在特征空间上可以被表示为：

$$\min f(R,a,\xi)=R^2+C\sum\xi_i,i=1,2,\cdots,n \tag{10}$$

约束条件为：

$$\parallel\Phi(x_i)-a\parallel^2\leqslant R^2+\xi_i,\xi_i\geqslant 0 \tag{11}$$

式中，a 为超球体中心，R 为超球体的半径，$\parallel\Phi(x_i)-a\parallel^2$ 为点到中心 a 的距离，ξ_i 为松弛因子，$\xi_i>0$ 对应位于超球体外部的异常点，参数 C 为某个指定的常数，控制对错分样本的惩罚程度，反映了尽量少的异常点数和尽可能小的超球体半径 R 的调整。

引入拉格朗日乘子 $\alpha,\beta\geqslant 0$ 将原问题式(10)和式(11)转化为无约束问题。

$$L(R,a,\xi,\alpha,\beta)=R^2+C\sum_i\xi_i-\sum_i\alpha_i*(R^2+C\sum_i\xi_i-(\parallel\Phi(x_i)-a\parallel^2))-\sum_i\beta_i\xi_i \tag{12}$$

上式分别对 R,ξ 和 a 求偏微分，并令它们等于 0 得：

$$\begin{cases}\sum_i\alpha_i=1\\ a=\sum_i\alpha_i\cdot\Phi(X_i)\\ \alpha_i=C-\beta_i\end{cases} \tag{13}$$

将式(13)表示为对偶问题，并用满足 Mercer 条件的核函数取代特征空间上的内积运算，即令 $K(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j)$，式(12)转化为 α 关于的最大化函数：

$$L=\sum_i\alpha_iK(x_i,x_j)-\sum_{i,j}\alpha_i\alpha_j(x_i\cdot x_j) \tag{14}$$

$$\text{s.t.}\begin{cases}0\leqslant\alpha_i\leqslant C\\ \sum_i\alpha_i=1\end{cases} \tag{15}$$

求解对偶问题，得到 α_i，分析发现只有少量 $\alpha_i>0$，它们对应的数据 x_i 被称为支持向量。超球体的中心 $a=\sum_i\alpha_i\cdot\Phi(x_i)$，因此 a 由支持向量确定，而半径 R 则仅由 $\alpha_i>0$ 且 $\alpha_i\neq C$ 的数据点到 a 的距离确定，这些点就位于超球体的边界上。支持向量 x_i 及相应的拉格朗日乘子 α_i 共同组成了模型。测试样本 z 被接受为目标样本，需满足决策函数为：

$$K(z,z)-2\sum_i\alpha_iK(z,x_i)+\sum_{i,j}\alpha_i\alpha_jK(x_i,x_j)\leqslant R^2 \tag{16}$$

具体基于 SVDD 的异常入侵检测算法(SVDDID)如下：

输入：训练数据集 T，测试数据集 S，滑动窗口大小 l(l≥2)，特征空间维度 P，核宽度参数 σ，误报率容忍度 fracrej。

输出：带类别标号的测试数据集(类别标号为 0 代表正常，类别标号为 1 代表异常)。

步骤 1　针对每个 $T_j\in T$，采用滑动窗口将其划分为 $|T_j|-l+1$ 个长度为 l 子序列，根据定义 2 计算 T_j 相对应的子序列频率特征 F_j，并返回 $\max(F_j)$；

步骤 2　针对每个 $S_i\in S$，采用滑动窗口将其划分为 $|S_i|-l+1$ 个长度为子序列，根据定义 2 计算 S_i 相对应的子序列频率特征 F_i，并返回 $\max(F_i)$；

步骤 3　根据定义 3 和特征映射方法，将正常系统调用序列集合 T 所包含的每个系统调用序列所对应的子序列频率特征映射为 p 维数值型特征向量 $\tilde{T}_j$，并将所有的 p 维特征向量存储为 $\tilde{T}$；

步骤 4　根据定义 3 和特征映射方法，将待检测系统调用序列集合 S 所包含的每个系统调用序列所对应的子序列频率特征映射为 p 维数值型特征向量 $\tilde{S}_j$，并存储为 p 维特征向量矩阵 $\tilde{S}$；

步骤 5　在 $\tilde{T}$ 上用 SVDD 训练得到球心 a、半径 R 和支持向量集 V 的值。

步骤 6　在 $\tilde{S}$ 中逐个读取 $\tilde{S}_j$，并根据式(16)测试 $\tilde{S}_j$。若满足，则该特征元组类标号标记为 0，否则标记为 1。

步骤 7　输出带类标号的测试数据集。

在算法中依据特征库计算测试数据集和训练数据集的频率特征向量是基于数据驱动的数据挖掘方法的体现，能够更加准确和客观地提取数据检测特征。在步骤 5 中，采用 SVDD 对训练数据集进行训练，除了在输入中给定的核宽度参数 σ，还需要设置误报率最大容忍度参数 $fracrej$，这个参数表示在训练异常入侵检测模型时，允许发生误报率的上界，本文中设置为 0.1(10%)。

4 入侵检测试验与分析

为了准确地对本文提出的异常入侵检测方法进行评价，我们采用两个用于检验异常入侵检测算法的标准数据库作为实验数据集，在 PC 机（CPU 为 Pentium（R） Dual-Core 2.0GHz、内存为 2.0GB、操作系统为 Windows XP）上采用 C++编程进行测试；并与其他算法执行异常入侵检测的实验进行比较。

4.1 实验数据的选择及其处理

第一个数据库由美国新墨西哥大学提供，其给出的正常系统调用序列由基于某个操作系统平台下的计算机程序进行正常操作产生，如 sendmail、ftp、lpr 等计算机程序。异常系统调用序列由程序运行在某个不正常的模式下产生，如有黑客已经入侵程序运行的操作系统。本文基于可用的数据集进行了广泛实验，鉴于篇幅，本文只给出数据库中的两个数据集上的实验结果，这两个数据集为 UNM_SM（UNM Synthethic Sendmail Data）和 CERT_SM（CERT Synthethic Sendmail Data）。对于以上两个原始的数据集，每个数据集包含的正常系统调用序列和异常系统调用序列数量都较少。为了能够构建有意义的训练数据集和测试数据集，本文采用常用的做法，即针对每个较长的系统调用序列，采用固定长度滑动窗口，以一定的步长跳跃取子串，将从正常系统调用序列中取出的子序列当作正常系统调用序列；从异常系统调用序列中取出子序列，并在正常系统调用序列中检测其是否出现，如果出现则丢弃，否则当作异常系统调用序列，新增加的序列和原来的序列混合后增大了数据集的规模。对于另外一个异常入侵检测标准数据库，本文采用一个基础信息安全模块审计数据[15]。这些数据来源于 1998 年林肯实验室的一个网络安全项目“DARPA”，从一台 Solaris 主机采集得到。正常系统调用序列和异常系统调用序列分别在正常和异常模式下经过几周的时间收集。本文采用其中 3 周收集到的数据，分别命名为 DARPA_W1、DARPA_W2、DARPA_W3。表 3 中给出各个数据集的属性及其实验设置。

表 3 实验所用到数据集属性

数据集名称	序列种类数目	序列平均长度	正常序列数目	异常序列数目	训练集大小	测试集大小
CERT_SM	56	803	1811	172	811	1050
UNM_SM	53	839	2030	130	1030	1050
DARPA_W1	67	149	1000	800	10	210
DARPA_W2	73	141	2000	1000	113	1050
DARPA_W3	78	143	2000	1000	67	1050

4.2 实验对比与分析

为了验证本文提出的方法进行系统调用入侵检测的效果，我们采用了两种异常检测评价方法对实验结果进行比较。第一种评价方法基于检测率（DR）和误报率（FR）[16]，公式如下：

$$DR=|AO|/|CO| \tag{17}$$

$$FR=(|BO|-|AO|)/(|DN|-|CO|) \tag{18}$$

其中，$|AO|$表示被检测出来的真实异常数目；$|BO|$表示被检测出的异常数目（其中可能包含正常数据）；$|CO|$表示整个数据集中真实异常的数目；$|DN|$表示整个数据集所含数据点的数目。对于异常检测问题一般同时采用第二种方法，即利用 ROC 曲线下面积 AUC（Area Under Curve）进行度量[17]。ROC 曲线由不同误报率下的检测率所构成，曲线下的面积越大（接近 1），表示异常检测方法的效果越好；如果 AUC 值为 0.5，表示异常检测器的性能处于随机划分异常数据和正常数据的水平，无法进行有效检测。

在对比试验中选取经典的异常入侵检测方法 STIDE[5] 和最新的基于 kNN 的异常入侵检测方法[8]（本文使用 JD 距离的方法，简称为 JDKNN）。对各种方法，我们进行大量的实验，为公平起见，给出每种方法的最优结果进行比较。对于 STIDE，我们发现其对滑动窗口的长度取值较为敏感，对于滑动窗口长度取值小于 5 和大于 10，结果表现较差。通过我们的实验，将最优滑动窗口长度取值为 5。对于 JDKNN 方法，也需要用滑动窗口去抽取子序列，其对滑动窗口的长度取值与 STIDE 有相同的性质，其全局最优滑动窗口长度为 7；我们对[2,32]范围的 k 值进行遍历，发现 k 值越大，实验结果表现越差，最优的 k 值为 4。对于我们提出的方法 SVDDID，其涉及到滑动窗口长度 l、特征向量维度 p、核函数宽度 σ 和误报率容忍度 $fracrej$ 3 个参数。经过大量实验，我们发现滑动窗口长度 l 取值全局最优为 6；特征向量维度 p 全局最优为 5。检测准确性结果在表 4 和表 5 中给出（最好的结果用黑体标示），从中可以看出，本文提出的方法 SVDDID 在检测结果的准确性上优于已经存在的两种方法，尤其是对于较为复杂的数据集。

表 4 各个方法在真实数据集上的异常检测检测率与误报率

对比方法	UNM_SM		CERT_SM		DARPA_W1		DARPA_W2		DARPA_W3	
	DR	FR	DR	FR	DR	FR	DR	FR	DR	FR
STIDE	0.78	0.02	0.64	0.02	0.20	0.15	0.36	0.18	0.60	0.16
JDKNN	0.84	**0.02**	**0.94**	**0.005**	0.20	0.20	0.52	0.15	0.48	0.10
SVDDID	**0.88**	0.04	0.90	0.03	**0.65**	**0.10**	**0.60**	**0.09**	**0.72**	**0.08**

表 5 各个方法在真实数据集上的 AUC

对比方法	UNM_SM	CERT_SM	DARPA_W1	DARPA_W2	DARPA_W3
STIDE	0.93	0.90	0.62	0.73	0.80
JDKNN	0.95	**0.99**	0.75	0.92	0.91
SVDDID	**0.98**	0.97	**0.92**	**0.96**	**0.92**

通过分析 STIDE、JDKNN 和 SVDDID 的实验结果可以得到如下结论：对于 DARPA 数据集，其异常检测正确率都较低。通过比较 SVDDID 和其它两种方法在 DARPA 数据集上的检测结果，可以得出这样结论，即较之将序列的局部特征直接求和作为异常度量的方式（即 STIDE 和 JDKNN 所采用方式），将抽取出来的子序列的局部特征作为异常调用序的度量方式（SVDDID 采用）获得效果更好，同时也证明本文提出的方法在提高检测的正确性和降低误报率上起到了重要作用。

结束语 通过对基于系统调用的异常入侵检测技术进行分析，指出目前技术共同存在的技术难点。本文结合经典的 STIDE 两阶段异常入侵检测模式，在第一阶段提出采用正常系统调用序列的子序列构造正常库；第二阶段采用 SVDD 建立异常入侵检测模型。在实验阶段采用两种异常检测评价方法对检测结果进行评价，实验结果表明本文提出的方法在两种评价标准上都优于已有的异常入侵检测方法。

参考文献

[1] Axelsson S. The base-rate fallacy and the difficulty of intrusion detection [J]. ACM Transactions on Information and System Security, 2000, 3(3): 186-205

（下转第 339 页）

经过大量样本实验，从正确识别率、错误识别率、平均识别时间3个方面证明了该识别方法的可用性。由表1中统计数据可以看出，该检测方法达到了比较高的正确识别率，将错误识别率降低在5%以下，同时耗费了很少的CPU运算时间，证明了该识别方法的可用性，对木马数据流特征的分析具有重要的意义。

结束语 本文在对木马会话数据流进行时序分析的基础上，提出基于时序分析的无指纹木马控制行为识别方法。该方法首先对木马网络数据流进行时序分簇处理，然后计算分簇数据的加权欧氏距离，利用木马数据流分簇距离的平稳分布特性进行判别，从而识别出木马控制行为。实验表明，该识别方法能够有效地识别出一般木马数据流中的控制行为，具有较高的正确识别率；同时该方法高效地利用了处理器资源，在高速的网络环境中也能做到实时处理。此外，该方法已经被实际应用于网络入侵检测和木马行为监控等实时网络数据流分析当中，并取得了较好的效果，表明该方法具有很强的实用性。但是方法仍存在一些不足，即在时延较大的拥塞网络状况下，准确识别率略有降低，如何在拥塞网络环境下保证较高的准确识别率还有待进一步研究。

参 考 文 献

[1] Zhang Li-ke, White G B. An Approach to Detect Executable Content for Anomaly Based Network Intrusion Detection[C]// Proc. of Parallel and Distributed Processing Sysmposium. Long Beach, USA: [s. n], 2007: 1-8

[2] 井小沛，汪厚详，聂凯，等. 面向入侵检测的基于IMGA和MKS-VM的特征选择算法[J]. 计算机科学，2012，39(7)：262-264

[3] Nie Fei-ping, Xiang Shi-ming, Jia Yang-qing, et al. Trace Ratio Criterion for Feature Selection[C]// Proceedings of National Conference on Artificial Intelligence. Chicago, USA: [s. n], 2008: 672-675

[4] Wang Sui-yu, Baird H S. Feature Selection Focused Within Error Clusters[C]// Proceedings of the 19th IEEE ICPR'08. [s. l]: IEEE Press, 2008: 1-4

[5] 易军凯，陈利，孙建伟. 网络心跳包序列的数据流分簇检测方法[J]. 计算机工程，2011，37(24)：201-524

[6] Nehinbe J O. Automated technique for debugging network intrusion detection systems[A]// IEEE 2010 International Conference on Intelligent Systems, Modelling and Simulation (ISMS)[C]. Liverpool, 2010: 363-367

[7] Wuu L C, Hung C H, CHEN S F. Building intrusion pattern miner for Snort network intrusion detection system[J]. Journal of Systems and Software, 2007, 80(10): 1701-1714

[8] 郭文忠，陈国龙，陈庆良，等. 基于粒子群优化算法和相关性分析的特征子集选择[J]. 计算机科学，2008，35(2)：113-147

[9] 陈友，沈华伟，李洋. 一种高效的面向入侵检测系统的特征选择算法[J]. 计算机学报，2007，30(8)：1395-1407

[10] 陈友，程学旗，李洋，等. 基于特征选择的轻量级入侵检测系统[J]. 软件学报，2007，18(7)：1639-1650

(上接第333页)

[2] Sundaram A. An Introduction to Intrusion Detection [J]. Crossroads, 1996, 2(4): 3-7

[3] Forrest S, Hofmeyr S A, Somayaji A, et al. Sense of self for Unix processes [C]// Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. Oakland, CA, USA: IEEE Computer Society Press, 1996: 120-128

[4] 吴瀛，江建慧，张蕊. 基于系统调用的入侵检测研究进展[J]. 计算机科学，2011，38(1)：20-25

[5] Forrest S, Hofmeyr S A, Somayaji A. Intrusion Detection Using Sequences of System Calls [J]. Journal of Computer Security, 1998, 6(3): 151-180

[6] Liao Yi-hua, Vemuri V R. Use of K-nearest Neighbor Classifier for Intrusion Detection [J]. Networks and Security, 2002, 21(5): 438-448

[7] Rawat S, Gulati V P, Arun K P, et al. Intrusion Detection Using Text Processing Techniques with a Binary-Weighted Cosine Metric [J]. Journal of Information Assurance and Security, 2006, 1(1): 43-50

[8] Jecheva V, Nikolova E. An adaptive KNN algorithm for anomaly intrusion detection [C]// Interaction of theory and practice: key problems and solutions. Burgas Bulgaria: Burgas Free University, 2011: 198-204

[9] 吕锋，刘泉永. 利用KNN算法实现基于系统调用的入侵检测技术[J]. 微计算机信息，2006，22(93)：76-78

[10] Forrest S, Warrender C, Pearlmutter B. Detecting Intrusions Using System Calls: Alternate Data Models[C]// Proceedings of the 1999 IEEE ISRSP. IEEE Computer Society, Washington, DC, USA, 1999: 133-145

[11] Tax D M J, Duin R P W. Support Vector Data Description[J]. Machine Learning, 2004, 54(1): 45-66

[12] University of New Mexico. Computer Immune Systems Project [OL]. http://www.cs.unm.edu/~immsec/systemcalls.htm

[13] Budalakoti S, Srivastava A, Otey M. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety [J]. IEEE Transactions on Systems, Man and Cybernetics (Part C: Applications and Reviews), 2009, 39(1): 101-113

[14] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data set[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Dallas, TX, United states: IEEE Computer Society Press, 2000: 427-438

[15] Thomas C, Sharma V, Balakrishnan N. Usefulness of DARPA dataset for intrusion detection system evaluation[C]// Proceedings of SPIE-The International Society for Optical Engineering. Orlando, FL, United States: IEEE Computer Society Press, 2000: 220-237

[16] Cerioli A, Farcomeni A. Error rates for multivariate outlier detection [J]. Computational Statistics and Data Analysis, 2011, 55(1): 544-553

[17] Fawcett T. An introduction to ROC analysis [J]. Pattern Recognition Letters, 2006, 27(8): 861-874