一种增强的程序行为异常检测方法

谢 丰[1] 谢丽霞[2]
(中国信息安全测评中心 北京 100085)[1] (中国民航大学计算机学院 天津 300300)[2]

摘 要 程序行为异常检测是保护应用程序的重要方法。针对异常检测的数据源选择问题,提出一种细粒度的安全审计事件L-Call,用来刻画程序行为,该事件本质上是一种具有位置属性的系统调用。为了评估程序行为偏离程度,提出一种基于切比雪夫不等式的异常度量化方法,用以在序列概率分布未知情况下估算异常强度。最后实现了基于马尔科夫模型的检测原型系统LC-ADS。试验结果表明,提出的新安全事件和异常度量化方法可较好地反映程序行为变化,LC-ADS取得了更高的检测率和更低的误报率。

关键词 L-Call,切比雪夫不等式,异常度量化,LC-ADS

Enhanced Approach to Anomalous Program Behaviors Detection

XIE Feng[1] XIE Li-xia[2]
(China Information Technology Security Evaluation Center, Beijing 100085, China)[1]
(School of Computer Science, Civil Aviation University of China, Tianjin 300300, China)[2]

Abstract Anomaly detection is an important method for protecting program. Traditionally a program is protected by means of monitoring system call, but the invoked address is often ignored. This paper presented a new audit event named as L-Call to describe the program behavior, which is the system call with invoked address in nature. A Chebyshev inequality-based method was also presented to evaluate the deviation of program behavior from normal. The deviation degree that we named as anomaly degree is based on the likelihood of L-Call sequence occurred under the unknown distribution. Finally a Markov-based prototype was constructed to evaluate the experiment, which is named as LC-ADS (i.e. L-Call based Anomaly Detection System). The experimental results show that LC-ADS acquires the better true positive rate and lower false alarm rate.

Keywords L-Call, Chebyshev inequality, Anomaly degree, LC-ADS

1 引言

入侵检测技术是一种重要的网络安全技术,传统上分为误用检测和异常检测。误用检测具有较高的检测率,但难以发现新类型攻击,而异常检测作为一种新的入侵检测技术,理论上可以发现潜在的新的攻击行为,正日益受到学术界和产业界的关注。异常检测的核心是正常行为建模,包括程序行为建模、用户行为建模和网络行为建模等,相关研究也主要围绕数据源选择、行为学习与建模等几个方面展开。其中数据源选择问题至关重要,其质量好坏将决定检测系统的效果。

与用户行为和网络行为相比,程序行为具有更好的规律性和稳定性。同时由于应用程序普遍存在缺陷而容易被利用,对程序行为的监测与保护也日益重要,因此近年来程序行为异常检测技术更多受到学术界关注。现代操作系统至少被划分为内核层和应用层两个层次,内核为应用程序提供最基本的系统服务,比如I/O管理、虚拟内存、任务调度等等。这些功能以"系统调用"方式提供给用户,而应用程序可以在自己代码中使用系统调用来实现对系统的访问或对内核功能的调用。可以说系统调用是应用层使用内核层功能的唯一接口。通常情况下,网络入侵的目标是为了获取目标系统的敏感信息,或者控制目标主机,这些操作大部分都需要内核层提供的功能,因此不可避免地要使用多个系统调用,通过监视系统调用的执行在一定程度上发现入侵行为。

基于上述思想,文献[1]提出了基于系统调用序列的入侵检测模型。该模型记录正常运行状态下的系统调用序列,一旦发现序列发生偏离就判定程序被攻击。试验结果也表明了该方法能发现许多攻击行为。受此启发,后续出现了多种基于系统调用的分析模型,包括隐马尔科夫模型[2,5,6]、自动机模型[3,9]、支撑向量机模型[8]、词袋模型[4]等。这些研究重点关注行为建模中的行为描述和学习方式,但没有对数据源做深入讨论,均使用常规的系统调用数据,容易受到"模仿攻击(Mimicry Attack)"[7]。

本文提出了一种新的安全审计事件。该事件结合了系统调用的位置信息,检测粒度更细,能更准确描地述程序的行为,而且使用该数据源并不需要修改现有的检测模型。

同时,针对异常度量化问题,提出了一种基于切比雪夫不

到稿日期:2009-04-15 返修日期:2009-07-15 本文受国家自然科学基金(60776807)资助。

谢 丰(1977-),男,博士,助理研究员,主要研究方向为网络安全和数据挖掘等,E-mail:fengxie@126.com;谢丽霞(1974-),女,副教授,主要研究方向为网络安全等。

等式的度量方法，该方法在序列分布未知情况下可以较好地估算程序运行的异常程度。

2 一种新的安全审计事件

程序行为检测的传统数据源是系统调用，但没有考虑其发生位置。事实上即使对于同一类型系统调用，在不同位置其含义也不同，区分其位置将有助于程序行为描述模型的精确化。

在计算机程序设计中，系统堆栈由一组栈帧(Stack Frame)构成，用于保存函数参数、内部变量、上一栈帧地址和返回地址等信息。其中通过"上一栈帧地址"可以遍历堆栈中每一个栈帧内容。当系统调用发生时，将栈帧中的返回地址组织在一起，并与下一指令地址(存放在指令寄存器中)共同构成一个地址链表，以表示该系统调用的位置信息。其中每一个地址可以表示为段地址和段内偏移量方式。

本文将不重复的〈系统调用，地址信息〉映射为一个新的事件，称之为 L-Call(System Call with Location)。图 1 显示了 L-Call 的生成方法。

当数据获取模块截取到当前检测程序执行的系统调用 s 时，

Step 1 根据 EBP 指针，获取堆栈顶部栈帧 top_SF，并记为 current_SF；

Setp 2 如果 current_SF 是堆栈底部，执行 Step 5，否则执行 Step 3；

Step 3 获取 current_SF 保存的返回地址，并添加到地址链表 list 中；

Step 4 根据"上一栈帧地址"，获取上一栈帧 previous_SF 并置为 current_SF，执行 Step 2；

Step 5 查询段地址映射表，将地址链表 list 中的每一个地址转为 ID 和偏移量；

Step 6 查询事件映射表，将〈s，list〉映射为 L-Call。

图 1 安全审计事件 L-Call 的生成方法

3 增强的程序异常检测系统 LC-ADS

3.1 基于马尔科夫的检测原理

在马尔科夫模型下，每一个 L-Call 作为程序的一种执行状态，两个相邻 L-Call 看作从一种状态转移到另一状态。通过观察序列的发生概率，确定其是否异常。对于长度为 n 的短序列$(s_1,s_2,\cdots,s_n)$，其概率为

$$p(s_1,\cdots,s_n)=\pi_{s_1}\prod_{i=2}^{n}p_{s_{i-1}s_i} \tag{1}$$

其中，π_{s_1} 为事件 s_1 的初始概率，$p_{s_{i-1}s_i}$ 表示事件 s_{i-1} 到 s_i 的转移概率，可以由训练集统计得到(为防止序列概率为 0，可以假定统计概率最小值为 0.00001)。假定 $N_{s_is_j}$ 表示训练集中 s_i 和 s_j 相邻出现的次数，N_{s_i} 表示 s_i 出现的次数，N 表示训练集中 L-Call 出现的总数，则

$$p_{s_is_j}=\frac{N_{s_is_j}}{N_{s_i}},\pi_{s_i}=\frac{N_{s_i}}{N} \tag{2}$$

由于概率是在[0,1]范围，多个概率的乘积过小而不便计算，因此本文实际计算概率的对数值，即

$$\begin{aligned}\log p(s_1,\cdots,s_n)&=\log(\pi_{s_1}\prod_{i=2}^{n}p_{s_{i-1}s_i})\\&=\log(\pi_{s_1})+\sum_{i=2}^{n}\log(p_{s_{i-1}s_i})\end{aligned} \tag{3}$$

3.2 基于切比雪夫不等式的异常度量化方法

程序行为的异常点是稀疏分布还是连续发生，其含义显然不同。由于训练数据的有限性和不完备性，在检测过程中即使正常操作也会产生一些异常点。通常正常行为引发的异常点分布较为离散和稀疏，而网络入侵产生的异常点常在短时间内集中出现。通过分析异常点的分布状况，可以有效判断其产生的真实原因。

另外不同异常点具有不同的异常程度，反映了检测序列偏离正常的多少。本文使用异常度来衡量这种差异性。假定序列概率为随机变量 X，具有数学期望 $E(X)=\mu$，方差 $D(X)=\sigma^2$，则对于任意正数 ε，不等式(4)成立。

$$p(|x-\mu|\geqslant\varepsilon)\leqslant\frac{\sigma^2}{\varepsilon^2} \tag{4}$$

该不等式称为切比雪夫不等式，它给出了在随机变量 X 分布未知的情况下事件$\{|x-\mu|\geqslant\varepsilon\}$概率上限的一种估计方法。

异常度量化的基本思想如下：在训练阶段确定正常序列发生概率的波动范围，而在测试阶段，若序列概率落在该范围之内，则序列异常度为 0，超出范围，则表示该序列具有一定程度的异常，并根据相应量化公式得到定量的异常度。

本文将该范围称为"安全距离"。理论上可以将训练集中出现的最小概率确定为安全距离，但这种与训练集过度拟合的方法会导致实际检测效果不佳。可以令安全距离 $\varepsilon=d\sigma$ $(d>0)$，置信度下限值为 $c(1>c>0)$(即无论 X 属于何种分布，训练集内位于安全距离范围的样本比例不少于 c)，则安全距离系数 $d=\frac{1}{\sqrt{1-c}}$。在检测过程中，假定第 i 个序列的实际测试概率值为 l，令 $k=\frac{l-\mu}{\sigma}$，则根据式(4)可知

$$\begin{aligned}p(|x-\mu|\geqslant|l-\mu-d\sigma|)&=p(|x-\mu|\geqslant|k\sigma-d\sigma|)\\&\leqslant\frac{1}{(k-d)^2}\end{aligned} \tag{5}$$

成立。定义序列 i 的异常度(Anomaly Degree)为

$$AD_i=\begin{cases}e^{-\frac{1}{(k-d)^2}} & k>d\\ 0 & \text{others}\end{cases} \tag{6}$$

该公式表明，在安全距离内的测试样本，其异常度为 0；偏离安全距离程度越大，其异常度越大，但不会超过 1。这既能够区分不同序列的异常程度，又能防止某一个序列的概率值过小而影响最终判断。

3.3 原型系统实现

本文实现了基于 L-Call 数据源的检测原型系统 LC-ADS (L-Call-based Anomaly Detection System)，其基本框架如图 2 所示。

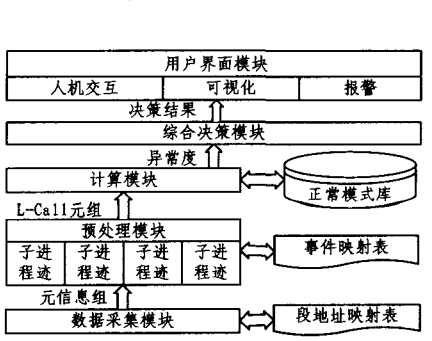

图 2 LC-ADS 基本框架

该系统主要包含以下几个模块：

数据采集模块：该模块驻留内存，负责截取指定应用程序及其子进程的系统调用和地址信息。获取的每一组数据用以

下元信息组表示：

〈ProcessID，TimeStamp，SID，Location，ReturnValue〉

ProcessID：当前跟踪进程的标识号。

TimeStamp：该系统调用发生的时间戳。

SID：该系统调用的标识号。

Location：该系统调用发生时的位置信息，表示为一组地址组成的链表。

ReturnValue：表示系统调用的返回值，但只有当系统调用为 fork，vfork 和 clone 时才有效。该值用于识别一个新进程的开始。

预处理模块：该模块对采集的原始数据进行处理，主要负责进程识别和 L-Call 映射，分为两种模式：训练模式和检测模式，分别用于系统的训练阶段和检测阶段。为了能对不同子进程的数据进行分别处理，该模块内部维护了多个队列。在训练模式下，模块首先根据元信息组的 ProcessID，将该数据发送到相应队列中，然后查询事件映射表，确定是否有相应的 L-Call。如果没有，则新添加一项。事件映射表的每一项表示为〈SID，Location，LID〉，其中 LID 表示 L-Call 的标识号。在检测模型下，通过查询事件映射表获得 LID，如果不存在，则使用一个预定义的数值代替。通过预处理模块，得到一个 L-Call 元组〈ProcessID，TimeStamp，LID〉。

计算模块：该模块根据预处理模块输出的 L-Call 元组，计算序列的异常度，分为两种模式：训练模式和检测模式。在训练模式下，模块统计 L-Call 的出现次数和转移频率，生成程序正常行为的马尔科夫模型。在检测模式下，模块计算序列的异常度，并将计算结果发到综合决策模块。

综合决策模块：该模块记录所有异常点的强度和产生位置，并根据分布情况判断整个程序是否异常，最后将决策结果发送到用户界面模块。

用户界面模块：该模块用于人机交互和报警信息输出，以及完成系统的用户管理和日志管理等辅助功能。

4 实验分析

实验选择常见的 wu-ftpd 2.6.0 作为检测程序。该程序存在多个漏洞，利用这些漏洞，攻击者可以进入系统执行 shell 命令，并访问敏感文件。训练数据只包含正常操作样本，测试数据包含 21 个正常样本和 36 个入侵样本，其中测试数据中的正常样本不包含在训练数据中。

为便于结果比较，本文采用相同原理实现了基于系统调用的原型系统 SC-ADS(System Call based Anomaly Detection System)，它与 LC-ADS 唯一差别在于检测数据源不同。图 3 显示了两种系统的检测结果，可以看出 LC-ADS 的检测效果要明显好于 SC-ADS。

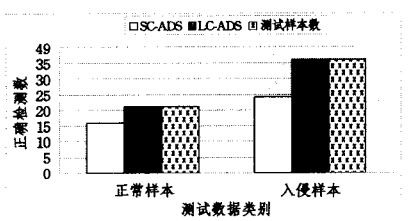

图 3 LC-ADS 与 SC-ADS 的检测效果对比图

图 4 和图 5 分别显示了在 LC-ADS 系统中正常测试样本和入侵样本的异常度分布，X 轴表示 LC-ADS 随程序运行所收集的 L-Call 序列(序列长度取 10)，Y 轴表示每一个序列的异常度。从图中可以看到，正常操作行为异常点极少，而入侵操作会导致程序行为明显异常，分布较为集中，并且在异常度最大的地方正对应着攻击行为的执行。

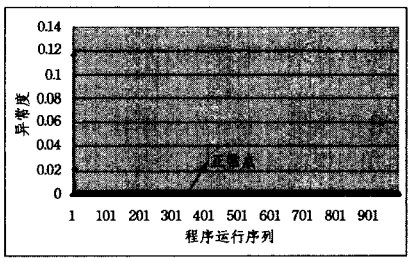

图 4 一个正常测试案例的异常度分布

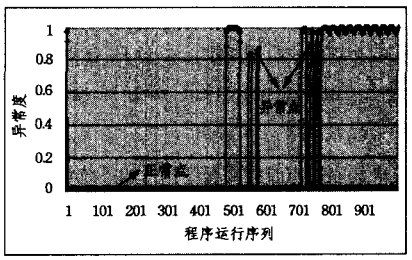

图 5 一个入侵测试案例的异常度分布

结束语 程序行为异常检测是保护应用程序的一种重要方法。本文提出的 LC-ADS 基于更细粒度的安全审计事件 L-Call。该事件使用系统调用发生时堆栈中所有返回的地址信息，以便更精确地描述程序行为。同时提出了基于切比雪夫不等式的异常度量化方法，用以将序列的异常程度归一化到[0，1]区间。本文下一步将对地址信息的提取效率展开研究，对 LC-ADS 进一步优化以提高其检测性能。

参 考 文 献

[1] Forrest S, Hofmeyr S A, Somayaji A, et al. A sense of self for unix process[C]// Proc. of the 1996 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1996: 120-128

[2] Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using system calls: alternative data models[C]// Proc. of the 1999 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1999: 133-145

[3] Sekar R, Bendre M, Bollineni P, et al. A fast automaton - based approach for detecting anomalous program behaviors [C] // Proc. of the 2001 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 2001: 144-155

[4] Kang D K, Fuller D, Honavar V. Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation[C]// Proc. of the 2005 IEEE Systems Man and Cybernetics Information Assurance Workshop. 2005: 118-125

[5] Jia C F, Yang F. An intrusion detection method based on hierarchical hidden markov models[J]. Wuhan University Journal of Natural Sciences, 2007, 12(1): 135-138

[6] Qian Q, Xin M J. Research on hidden markov model for system call anomaly detection[J]. Lecture Notes in Computer Science, 2007, 4430: 152-159

(下转第 101 页)

$$\Pr[C^{4I}_{MRHF\text{-}CTS}<C_0]=\Pr[|h_{SD}|^2+|h_{R_2D}|^2+\frac{1}{\bar{\gamma}}f(\bar{\gamma}|h_{SR_1}|^2,\bar{\gamma}|h_{R_1D}|^2)<\frac{2^{3C_0}-1}{\bar{\gamma}}]\to(\frac{1}{6\sigma^2_{SD}\sigma^2_{R_2D}}\frac{\sigma^2_{SR_1}+\sigma^2_{R_1D}}{\sigma^2_{SR_1}\sigma^2_{R_1D}})(\frac{2^{3C_0}-1}{\bar{\gamma}})^3 \quad (8)$$

4.2) R_1 转发时隙、R_2 接收并错误译码，即 $P_{42}=\Pr[|h_{SR_2}|^2+(1\sqrt{\gamma})f(\bar{\gamma}|h_{SR_1}|^2,\bar{\gamma}|h_{R_1R_2}|^2)<(2^{3C_0}-1)\sqrt{\gamma}]$，此时

$$\Pr[C^{4II}_{MRHF\text{-}CTS}<C_0]=\Pr[|h_{SD}|^2+\frac{1}{\bar{\gamma}}f(\bar{\gamma}|h_{SR_1}|^2,\bar{\gamma}|h_{R_1D}|^2)+\frac{1}{\bar{\gamma}}f(\bar{\gamma}|h_{SR_2}|^2,\bar{\gamma}|h_{R_2D}|^2)<\frac{2^{3C_0}-1}{\bar{\gamma}}]\to(\frac{\sigma^2_{R_1D}\sigma^2_{R_2D}+\sigma^2_{SR_1}\sigma^2_{R_2D}+\sigma^2_{SR_2}\sigma^2_{R_1D}+\sigma^2_{SR_1}\sigma^2_{SR_2}}{6\sigma^2_{SD}\sigma^2_{SR_1}\sigma^2_{SR_2}\sigma^2_{R_1D}\sigma^2_{R_2D}})(\frac{2^{3C_0}-1}{\bar{\gamma}})^3 \quad (9)$$

综合式(4)至式(9)，即可得到双中继混合转发协作传输策略的中断概率为

$$\begin{aligned}&\Pr[C_{MRHF\text{-}CTS}<C_0]\\&=P_1\times\Pr[C^1_{MRHF\text{-}CTS}<C_0]+P_2\times\Pr[C^2_{MRHF\text{-}CTS}<C_0]+P_3\times(P_{31}\times\Pr[C^{3I}_{MRHF\text{-}CTS}<C_0]+P_{32}\times\Pr[C^{3II}_{MRHF\text{-}CTS}<C_0])+\\&P_4\times(P_{41}\times\Pr[C^{4I}_{MRHF\text{-}CTS}<C_0]+P_{42}\times\Pr[C^{4II}_{MRHF\text{-}CTS}<C_0])\end{aligned} \quad (10)$$

2.2 性能评估与数值结果

仿真环境为瑞利衰落、加性高斯白噪声信道。为简化分析过程，假设各链路平均信噪比相等，期望传输速率 $C_0=1$bits/(s·Hz)。图2给出了多中继混合转发协作传输策略协作中继节点数分别为1和2时的大信噪比情况下近似的中断概率性能。为了体现MRHF-CTS的性能增益，同时给出了直接传输、AF协作传输策略、DF协作传输策略以及多中继完全译码DF协作传输策略相同条件下的大信噪比近似中断概率性能曲线。可以看到，多中继混合转发协作传输策略在有效抑制因错误检测DF协作传输而严重降低系统性能的前提下，获得了与AF协作传输相同的分集增益(大信噪比时中断概率曲线的斜率[12])，并趋近多中继DF协作通信系统的最好性能，即协作中继节点均正确接收，DF协作传输(多点协作传输)时的系统性能。

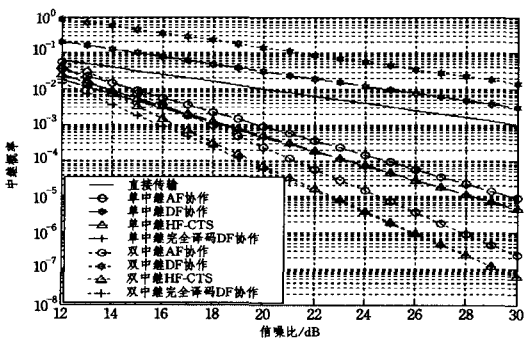

图2 不同策略的中断概率性能对比(大信噪比)

结束语 本文从有效简化全局网络规划及信令开销入手，考虑优化多中继协作通信系统性能，提出了一种多中继混合转发协作传输策略。在不同协作中继节点选择性地采用DF或AF传输方式的基础上，充分利用无线网络的广播特性，在数据中继转发的过程中，考虑协作中继节点间的交互信息。研究结果表明，本策略在以较低的系统设计复杂度获取分集增益的同时，显著地提高了多中继协作通信系统的传输性能。

参考文献

[1] Laneman J N, Tse D N C, Wornell G W. Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior [J]. IEEE Transactions on Information Theory, 2004, 50(12): 3062-3080

[2] Su Wei-Feng, Sadek A K, Liu K J R. Cooperative Communication Protocols in Wireless Networks: Performance Analysis and Optimum Power Allocation[J]. Wireless Personal Communication, 2008, 44: 181-217

[3] Zhao Yi, Adve R, Lim T J. Symbol Error Rate of Selection Amplify-and-Forward Relay Systems [J]. IEEE Communications Letters, 2006, 10(11): 757-759

[4] Ibrahim A S, Sadek A K, Su Wei-Feng, et al. Cooperative Communications with Relay Selection: When to Cooperate and Whom to Cooperate with? [J]. IEEE Transactions on Wireless Communications, 2008, 7(7): 2814-2827

[5] 刘丹谱，郝建军，乐光新. 用于机会中继的一种最佳中继选择算法[J]. 中国电子科学研究院学报，2008，3(5)：483-487

[6] 高伟东，王文博，袁广翔，等. 协作通信中的中继节点选取和功率分配联合优化[J]. 北京邮电大学学报，2008，31(2)：68-71

[7] 惠鏸，朱世华，李国兵. 一种基于放大转发的中继选择策略[J]. 西安交通大学学报，2008，42(4)：450-453

[8] Laneman J N, Wornell G W. Distributed Space Time Coded Protocols for Exploiting Cooperative Diversity in Wireless Networks[J]. IEEE Transactions on Information Theory, 2003, 49(10): 2415-2425

[9] Zhang Jun, Lok T M. Performance Analysis of Multiple-relay Decode-and-Forward Cooperation System [C]// IEEE Tencon 2005. Melbourne, 2005

[10] Zhang Jun, Lok T M. Multiple Source Multiple Relay Cooperation System[C]// IEEE International Conference on Communications (ICC). Istanbul, Turkey, 2006: 3735-3740

[11] Stanojev I, Simeone O, Bar-Ness Y. Performance Analysis of Collaborative Hybrid-ARQ Protocols over Fading Channels[C]// Sarnoff Symposium. 2006: 1-4

[12] Laneman J N. Limiting Analysis of Outage Probabilities for Diversity Schemes in Fading Channels [C]// Proc. IEEE Global Communications Conference(GLOBECOM). San Francisco, CA, 2003

(上接第66页)

[7] Wagner D, Sotl P. Mimicry attacks on host-based intrusion detection systems[C]// Proc. of the 2002 ACM Conference on Computer and Communications Security. New York: ACM Press, 2002: 255-264

[8] Yan Ye. Text Image Compression Based on Pattern Matching [D]. University of California, 2002

[9] 饶鲜，董春曦，杨绍全. 基于支持向量机的入侵检测系统[J]. 软件学报，2003，14(4)：798-803

[10] 李闻，戴英侠，连一峰，等. 基于混杂模型的上下文相关主机入侵检测系统[J]. 软件学报，2009，20(1)：138-151