文章编号:1001-2486(2003)05-0063-05

基于隐马尔可夫模型的IDS程序行为异常检测*

孙宏伟,田新广,邹 涛,张尔扬
(国防科技大学电子科学与工程学院,湖南 长沙 410073)

摘 要:提出一种新的基于隐马尔可夫模型的程序行为异常检测方法,此方法利用系统调用序列,并基于隐马尔可夫模型来描述程序行为,根据程序行为模式的出现频率对其进行分类,并将行为模式类型同隐马尔可夫模型的状态联系在一起。由于各状态对应的观测值集合互不相交,模型训练中采用了运算量较小的序列匹配方法,与传统的Baum-Welch算法相比,训练时间有较大幅度的降低。考虑到模型中状态的特殊含义以及程序行为的特点,将加窗平滑后的状态序列出现概率作为判决依据。实验表明,此方法具有很高的检测准确性,其检测效率也优于同类方法。

关键词:入侵检测系统;异常检测;隐马尔可夫模型;系统调用
中图分类号:TP18;TP393.08 **文献标识码**:A

Anomaly Detection of the Program Behaviors for IDS Based on Hidden Markov Models

SUN Hong-wei, TIAN Xin-guang, ZOU Tao, ZHANG Er-yang
(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A new method for anomaly detection of the program behaviors based on hidden Markov models is presented. The method uses system calls to represent the behavior profiles of programs based on hidden Markov models. The behavior patterns of programs are classified according to their frequency distributions, and the states of the hidden Markov models are associated with the classes of the behavior patterns. Because the collections of observations corresponding to different states are mutually disjoint, the models can be trained with a sequence matching algorithm which requires lower computational complexity and less computation time than the classical Baum-Welch algorithm. A decision rule based on the probabilities of short state sequences is adopted while the particularity of the model states is taken into account. The performance of the method is tested by computer simulation. The results show it maintains higher detection accuracy and efficiency than other alternative approaches.

Key words: IDS; anomaly detection; hidden Markov model; system call

异常检测是目前IDS(入侵检测系统)研究的一个主要方向,这种检测方法建立系统或用户的正常行为模式库,通过被监测系统或用户的实际行为模式和正常模式之间的比较和匹配来检测入侵,其特点是不需要过多有关系统缺陷的知识,具有较强的适应性,能够检测出未知入侵,但存在虚警概率高的缺点。

国内外对基于系统调用和shell命令等主机数据的异常检测方法已经进行了较长时间的研究,其中隐马尔可夫模型(HMM)方法是一个重要的研究方向。Warrender C等人利用系统调用数据,进行了针对程序行为的异常检测研究和实验[2];其方法是对每种程序(如sendmail、lpr)建立一个HMM,采用Baum-Welch算法训练模型,并利用先验知识对模型参数进行初始化。Lane T利用UNIX用户的shell命令数据,进行了针对用户行为的异常检测[3];其方法是用单个HMM描述正常用户的行为轮廓,模型的训练中同样采用了Baum-Welch算法,检测时利用近似的前向后向算法,并根据贝叶斯准则对用户行为进行判决。以上两种方法是HMM在分类问题中的典型用法,在训练数据充足的情况下能够获得比较高的

* 收稿日期:2003-01-13
基金项目:北京首信集团重大科研项目(020015)
作者简介:孙宏伟(1964—),男,高级工程师,博士生。

检测准确率,但是,模型的训练和工作中所需要的运算量很大,检测的实时性不高,这在很大程度上限制了它们的应用。

本文提出一种新的基于 HMM 的程序行为异常检测方法,此方法根据行为模式的出现频率对其进行分类,用 HMM 的状态来代表不同种类的行为模式,并引入一个附加状态;采用序列匹配方法对模型进行训练,与传统方法相比较大程度地减小了训练时间;根据模型和程序行为的特点,采用了基于状态序列出现概率的判决准则。

1 一种基于 HMM 的程序行为异常检测新方法

1.1 HMM 概述

HMM 是双重随机过程,其中一个是有限状态马尔可夫链,它描述状态的转移;另一个随机过程描述状态与观测值之间的统计对应关系。设观测值序列为 $\boldsymbol{O}=(O_1,O_2,\cdots,O_T)$,相应的状态序列为 $\boldsymbol{q}=(q_1,q_2,\cdots,q_n)$,其中 $O_t\in\boldsymbol{\Omega}_O=\{v_1,v_2,\cdots,v_M\}$,$q_i\in\boldsymbol{\Omega}_q=\{\theta_1,\theta_2,\cdots,\theta_N\}$,以上 q_t 和 O_t 分别表示按时间顺序排列的第 t 个状态及其对应的观测值,$\boldsymbol{\Omega}_q$ 和 $\boldsymbol{\Omega}_O$ 分别为状态集合和观测值集合。HMM通常用五元组 $\boldsymbol{\mu}=(\boldsymbol{\Omega}_q,\boldsymbol{\Omega}_O,\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$ 来表示,$\boldsymbol{A}$ 为状态转移概率矩阵,$\boldsymbol{A}=(a_{ij})_{N\times N}$,其中

$$a_{ij}=P\{q_{t+1}=\theta_j\mid q_t=\theta_i\}\qquad 1\leqslant i,j\leqslant N \tag{1}$$

$\boldsymbol{B}$ 为观测值概率矩阵,$\boldsymbol{B}=(b_{jk})_{N\times M}$,其中

$$b_{jk}=P\{O_t=v_k\mid q_t=\theta_j\}\qquad 1\leqslant j\leqslant N,1\leqslant k\leqslant M \tag{2}$$

$\boldsymbol{\pi}$ 为初始状态概率矢量,$\pi=(\pi_1,\pi_2,\cdots,\pi_N)$,其中

$$\pi_i=P\{q_1=\theta_i\}\qquad 1\leqslant i\leqslant N \tag{3}$$

Baum-Welch算法、Viterbi算法和前向后向算法分别是HMM训练、解码和评估的经典算法。训练是指给定观测值序列 $\boldsymbol{O}$,确定参数 $\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$,使得 $P\{\boldsymbol{O}\mid\boldsymbol{\lambda}\}$ 最大;解码是对于给定的 $\boldsymbol{\lambda}$ 和 $\boldsymbol{O}$,求使 $P\{\boldsymbol{q}\mid\boldsymbol{O},\boldsymbol{\lambda}\}$ 最大的状态序列 $\boldsymbol{q}$;评估则是指给定模型参数 $\boldsymbol{\lambda}$,求观测值序列 $\boldsymbol{O}$ 的出现概率 $P\{\boldsymbol{O}\mid\boldsymbol{\lambda}\}$。

1.2 基于 HMM 的新方法

提出的方法主要用于以系统调用为观测数据的程序行为异常检测。此方法利用 HMM 来描述程序的正常行为,将 HMM 的状态与程序的行为模式类型联系在一起。由于不同状态对应的观测值集合互不相交,HMM 参数可根据训练数据用较为简单的序列匹配方法得到,无需运用传统的 Baum-Welch 算法,大大减少了训练时间。此外,根据 HMM 状态所代表的实际含义,采用了一种新的判决准则。下面按照建模、训练、检测的顺序对这一方法进行介绍。

(1) 建立一个 HMM 来描述一个程序的正常行为,HMM 的状态对应于该程序的行为模式类型。根据行为模式的出现频率对其进行分类。

行为模式是指程序运行过程中体现出的某种规律性。在基于系统调用数据的程序行为异常检测中,通常用迹(trace) 来代表一个程序的行为,它是该程序运行过程中所用的系统调用流。这样,行为模式在形式上表现为系统调用组成的序列。

设该程序正常运行时的迹为 $\boldsymbol{R}=(s_1,s_2,\cdots,s_r)$,它的长度为 r,其中 s_j 是按时序排列的第 j 个系统调用;$\boldsymbol{R}$ 作为训练数据用于HMM的建立和训练。$\boldsymbol{R}$ 对应的系统调用序列流可表示为 $\boldsymbol{S}=(Seq_1,Seq_2,\cdots,Seq_{r-l+1})$,其中序列 $\boldsymbol{Seq}_j=(s_j,s_{j+1},\cdots,s_{j+l-1})$,$l$ 为序列长度。确定最佳的序列长度是建模中的首要问题;根据信息熵理论和实验结论[2],程序的行为模式一般可以用长度在 6 ~ 15 之间的系统调用序列来表示。考虑到序列的长度越大,系统的存储量和工作中的运算量也会越大,因而序列长度应尽可能选得小一些。

实际中,在一个程序运行过程中的不同时间段上,系统调用序列的分布情况往往是不同的。而且,序列之间的相关性也与时间有关,特别是在序列相互相交的情况下。但是,在基于概率统计方法对程序行为进行建模时,一般都假设观测事件(系统调用序列) 是平稳的。我们按照 $\boldsymbol{S}$ 中各序列的出现频率来对

其分类,并没有考虑序列出现的时间;分类时,首先划分 W 个互不相交的频率区间,将 S 中出现频率落在同一区间的序列组成一个集合,把每个集合中的序列视为同一类型的行为模式,每类行为模式对应于HMM的一个状态;这样,每个序列集合即为一个状态的观测值集合。这里,HMM状态的观测值(或称观测事件)是长度为 l 的序列。设第 i 个区间对应的序列集合为 $L(i)$,当 $i \neq j$ 时($1 \leqslant i,j \leqslant W$),$L(i) \cap L(j) = \phi$,即不同状态对应的观测值集合是不相交的。一个序列的出现频率是指此序列在序列流中的出现次数与序列流中的序列总数之比,频率区间的划分方法对检测性能有着直接影响,在划分频率区间时,应充分考虑 S 中序列的分布特点。为了表示该程序正常运行时不会出现但是发生入侵时可能出现的行为模式(类型),我们引入一个附加状态(第 $W+1$ 个状态),它的观测值集合 $L(W+1)$ 包含除 S 中的序列之外所有长度为 l 的系统调用序列。综上所述,模型的状态集合为 $\boldsymbol{\Omega}_q = \{1,2,\cdots,W+1\}$,观测值集合为 $\boldsymbol{\Omega}_O = L(1) \cup L(2) \cup \cdots \cup L(W+1)$。

(2) 根据训练数据 $\boldsymbol{R}$,利用序列匹配方法计算 HMM 参数。

设HMM参数为 $\boldsymbol{\lambda} = (\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$,其中初始状态概率矢量 π 和状态转移概率矩阵 $\boldsymbol{A}$ 的计算方法如下:

第一步:根据 $\boldsymbol{R}$ 求出 $\boldsymbol{S}$。设定 $\boldsymbol{\pi} = (\pi_1,\pi_2,\cdots,\pi_{W+1}) := 0, A = (a_{ij})_{(W+1)\times(W+1)} := 0, m := 2, j := 1, i := g$(这里 $1 \leqslant g \leqslant W$,且满足 $Seq_1 \in L(g)$)。

第二步:如果 $m \leqslant r - l + 1$,将 Seq_m 与 $L(j)$ 进行比较;否则,跳至第四步。

第三步:若 $Seq_m \in L(j)$,则 $\pi_i := \pi_i + 1, a_{ij} := a_{ij} + 1$,并且 $m := m + 1, i := j, j := 1$,返回执行第二步;如果 $Seq_m \notin L(j)$,则 $j := j + 1$,返回执行第二步。

第四步:对于 $1 \leqslant i,j \leqslant W, a_{ij} := a_{ij}/\pi_i$。对于 $1 \leqslant i \leqslant W, \pi_i := \pi_i/(r - l)$。

上述的计算过程是按照时间顺序依次找出 $\boldsymbol{R}$ 中的行为模式及其对应的状态,同时对每个状态的出现次数和状态之间的转移次数进行统计,从而得到 $\boldsymbol{\pi}$ 和 $\boldsymbol{A}$,这是一个序列(行为模式)匹配和计数过程。序列匹配的方法是按照 j 从小到大的次序,将当前序列 Seq_m 同 $L(j)$ 进行比较,若 $Seq_m \in L(j)$,则认为匹配上一个行为模式(它所对应的状态为 j),然后再对 Seq_m 之后的下一个序列进行匹配。由于检测时不需要用到观测值概率矩阵 $\boldsymbol{B}$,其计算方法不再赘述。

(3) 检测时,利用计算出的 HMM 参数,基于状态序列出现概率对被监测程序的行为进行判决。

设被监测程序的迹为 $\overline{\boldsymbol{R}} = (\bar{s}_1,\bar{s}_2,\cdots,\bar{s}_r)$,其中 $\bar{s}_j$ 是按时序排列的第 j 个系统调用。检测时首先生成与 $\overline{\boldsymbol{R}}$ 对应的系统调用序列流(即观测值序列) $\boldsymbol{O} = (O_1,O_2,\cdots,O_{r-l+1})$,其中 $\boldsymbol{O}_j = (\bar{s}_j,\bar{s}_{j+1},\cdots,\bar{s}_{j+l-1})$;然后利用前面参数计算中的序列匹配方法得到相应的状态流 $\boldsymbol{q} = (q_1,q_2,\cdots,q_{r-l+1})$,其中 q_j 表示按时序排列的第 j 个状态,它对应的观测值(系统调用序列) 为 $\boldsymbol{O}_j$。

根据文献[2] 和我们的实验结论,实际中发生入侵(异常) 时程序的行为通常仅在其运行过程中某些较短的时间段内与其正常运行时的行为有较大差别,这些时间段内异常序列(集合 $L(W+1)$ 中的序列) 集中出现,而在其它时间段上,其行为同正常运行时的行为差别较小。基于这一情况,我们用滑动窗在 $\boldsymbol{O}$ 中截取短序列,以短序列为数据单元进行分析。设短序列为 $\boldsymbol{S}_{o_j} = (\boldsymbol{O}_j,\boldsymbol{O}_{j+1},\cdots,\boldsymbol{O}_{j+u-1})$,其中 u 表示短序列的长度,$1 \leqslant j \leqslant r - l - u + 2$;相应的状态短序列为 $\boldsymbol{S}_{q_j} = (q_j,q_{j+1},\cdots,q_{j+u-1})$。两种短序列流可分别表示为 $\boldsymbol{S}_O = (S_{o_1},S_{o_2},\cdots,S_{o_{r-l-u+2}})$ 和 $\boldsymbol{S}_q = (S_{q_1},S_{q_2},\cdots,S_{q_{r-l-u+2}})$。

按照传统准则,应根据 $P(\boldsymbol{So}_j/\boldsymbol{\lambda})$ 对被监测程序的行为进行判决,其计算公式为:

$$P\{\boldsymbol{S}_{o_j} \mid \boldsymbol{\lambda}\} = P\{q_j \mid \boldsymbol{\lambda}\} P\{\boldsymbol{O}_j \mid q_j,\boldsymbol{\lambda}\} \prod_{k=j}^{j+u-2} P\{q_{k+1} \mid q_k,\boldsymbol{\lambda}\} P\{\boldsymbol{O}_{k+1} \mid q_{k+1},\boldsymbol{\lambda}\} \tag{4}$$

这里,没有采用传统准则,而是将 $P\{\boldsymbol{S}_{q_j}/\boldsymbol{\lambda}\}$ 作为判决依据:

$$P\{\boldsymbol{Sq}_j \mid \boldsymbol{\lambda}\} = P\{q_j \mid \boldsymbol{\lambda}\} \prod_{k=j}^{j+u-2} P\{q_{k+1} \mid q_k,\boldsymbol{\lambda}\} \tag{5}$$

我们之所以用 $P\{\boldsymbol{S}_{q_j}/\boldsymbol{\lambda}\}$ 而不用 $P\{\boldsymbol{So}_j \mid \boldsymbol{\lambda}\}$ 作为判决依据,主要基于以下考虑:(1) 观测值集合与状态集合之间有明确的映射关系,每个状态所对应的观测值集合是根据程序正常运行时的迹(训练数据)

定义的,因而状态本身以及状态之间的转移情况能够反映正常行为与异常行为之间的差别。(2) $P\{S_{q_j} \mid \lambda\}$ 的计算量比 $P\{S_{o_j} \mid \lambda\}$ 小,它只用到了模型参数 π 和 A。(3) 在(4)式中,对 $P\{S_{o_j} \mid \lambda\}$ 的计算假设了观测值之间是相互独立的,即观测值只与当前状态有关,根据我们的实验和分析,这一假设并不很符合实际情况,因而根据(4)式得到的 $P\{So_j \mid \lambda\}$ 不宜作为判决依据。

检测中,通过对 $P\{S_{q_j} \mid \lambda\}$ 加窗平滑处理得到如下判决式:

$$D(j) = \frac{1}{w}\sum_{k=j-w+1}^{j} P\{S_{q_k} \mid \lambda\} \tag{6}$$

式中 $D(j)$ 表示状态序列 S_{q_j} 对应时刻的判决值,w 为窗长度(状态短序列流 S_q 中第 w 个状态短序列之后每个短序列所对应的时间点上都有一个判决值输出)。对 $D(j)$ 设定一个门限 η,若它大于 η,将被监测程序在窗长度时间内的行为判为正常行为。否则,将其判为异常行为。w 是一个重要参数,它决定了从被监测程序开始运行到检测系统对其行为做出判断的最短时间(即检测时间)。在不考虑计算时间的情况下,检测时间为 $w + u - 1$ 个状态持续时间。

2 特点分析

新方法主要有以下几个特点:

(1) 它是一种异常检测方法,主要体现在 HMM 的状态以及各状态对应的观测值集合是根据程序正常运行时的迹(训练数据)确定的,HMM 参数也是根程序正常运行时的迹(训练数据)计算得到的,并不需要有关程序异常行为的先验知识。

(2)HMM 的状态具有明确的含义,它是程序的行为模式类型。状态对应的观测值不是系统调用,而是系统调用序列。不同状态对应的观测值集合是互不相交的。状态的"隐含"是指在观测数据中状态并非直接可见,而是需要通过序列匹配得到。

(3) 根据 HMM 中状态的特殊含义以及程序行为的特点,采用了基于状态序列出现概率的判决准则。

3 实验设计及结果

采用新墨西哥大学 CIS 系统调用实验数据库中 lpr 和 ps 两种程序的数据进行了实验。其中 lpr 程序的数据包含一个 UNIX 平台上 15 个月的活动记录,共有 4298 个正常的迹和 1001 个含有入侵的迹,数据量为 260 多万个系统调用。ps 数据相对较少,包括 24 个正常的迹和 26 个含有入侵的迹。两种系统调用数据的详细说明可参见文献[2]。

我们做了三次交叉实验,每次实验将正常迹的数据量的三分之二作为训练数据用于 HMM 的建立和训练,其余的三分之一和入侵迹的全部数据用于此方法的性能测试。在对程序的行为模式进行分类时,采用了均分法来确定每类行为模式对应的频率区间。含有入侵的程序的行为仅在某些小的时间段内与相应的正常程序的行为有较大差别,实验中,如果在一个含有入侵的程序(迹)中某个时间点上将其行为判为异常,我们即认为成功检测到该程序(迹)中的入侵行为。因而,检测概率是以迹为单位进行计算的。但对于正常的迹,我们更关注错误判决的次数在总判决次数中的比例(特别是在程序运行时间很长的情况下),所以将这一比例定义为虚警概率。表 1 给出了在参数设置为 $u = 6, w = 15$,虚警概率为 0.1% 的情况下 l 和 W 变化时三次实验的平均检测概率。

表 1 l 和 W 变化时的检测概率(%)

Tab.1 Detection rate with varied l and W(%)

W \ l	6	9	12	15
6	92.12	92.79	92.91	92.57
8	93.96	94.36	94.23	93.82
10	94.53	95.18	94.80	94.65

以上结果表明,l 对检测性能的影响较小,而 W 对检测性能有较大的影响。在 $6 \leqslant W \leqslant 10$ 的区间内,

W 越大,检测概率越高。

进行了检测性能和效率的对比,采用 Christina 等人的传统 HMM 方法做了三次交叉实验,实验中参数设置为 $l = 6, W = 10$。根据实验结果,传统 HMM 方法的训练时间约为 4.5h,而新 HMM 方法的训练时间仅为 0.3h。但是,新 HMM 方法要占用较大的空间来存储各状态对应的观测值(系统调用序列)集合,而传统 HMM 方法则不需要这样。图 1 给出了两种方法的 ROC 曲线。

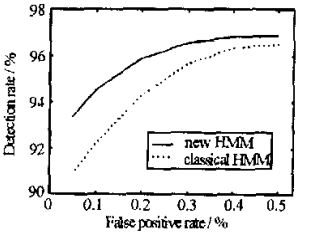

图 1 两种方法的 ROC 曲线

Fig.1 ROC curves for the two methods

由图 1 可见,在虚警概率较低的区间,新 HMM 方法的检测性能明显高于传统 HMM 方法;而随着虚警概率的升高,两种方法在检测性能上的差距逐渐减小。

4 结 论

提出一种基于 HMM 的 IDS 异常检测新方法。实验表明,在以系统调用为观测数据的程序行为异常检测中,此方法具有很高的检测准确率,其检测效率也优于传统的 HMM 方法。根据实验结果,在对程序的行为模式进行分类时,不同的频率区间划分方法对应的检测性能有一定的差异,因而,根据具体程序的行为特点选择合适的频率区间划分方法是进一步提高检测性能的重要途径。此外,对于以 shell 命令为观测数据的用户行为异常检测,本文的方法也具有一定的参考价值,但具体的检测性能还有待验证。

参 考 文 献:

[1] Rabiner L R, Juang B H. An Introduction to Hidden Markov Models[J]. IEEE ASSP Magazine, 1986(1):4 - 16.

[2] Warrender C, Forrest S, Pearlmutter B. Detecting Intrusions Using System Calls: Alternative Data Models[C]. Proc the 1999 IEEE Symposium on Security and Privacy, Berkely, California, USA: IEEE Computer Society, 1999:133 - 145.

[3] Lane T. Machine Learning Techniques for the Computer Security Domain of Anomaly Detection[D]. Purdue University, 2000.

[4] Kosoresow A P, Hofmeyr S A. A Shape of Self for UNIX Processes[J]. IEEE Software, 1997, 14(5):35 - 42.