

基于隐马尔可夫模型的程序行为异常检测

张响亮, 王 伟, 管晓宏

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对入侵检测中普遍存在误报与漏报过高的问题, 提出了一种基于隐马尔可夫模型的程序行为异常检测新方法. 该方法以程序正常执行过程中产生的系统调用序列为研究对象, 建立计算机的正常程序行为模型. 在入侵检测时, 先对测试的系统调用数据用滑动窗口划分得到短序列, 再根据正常程序行为的隐马尔可夫模型求得每个测试短序列的输出概率, 如果系统调用短序列的输出概率低于给定阈值, 则将该短序列标定为“不匹配”, 如果测试数据中不匹配的短序列数占总短序列数的百分比超过另一给定阈值, 该模型就认为此程序行为异常. 实验结果表明, 与 Forrest 和 Lee 的方法相比, 所提方法的检测率的最大提高率可达 590%.

关键词: 入侵检测; 隐马尔可夫模型; 异常检测; 系统调用

中图分类号: TP393 **文献标识码:** A **文章编号:** 0253-987X(2005)10-1056-04

Detection of Anomalous Program Behaviors Based on Hidden Markov Models

Zhang Xiangliang, Wang Wei, Guan Xiaohong

(School of Electronics and Information Engineering, Xi an Jiaotong University, Xi an 710049, China)

Abstract: To improve detection accuracy, a new intrusion detection method with high efficiency was presented. The method is based on hidden Markov model (HMM) to profile normal program behaviors using traces of system calls generated during the normal execution of processes. At the stage of anomaly detection, a testing trace of system calls is divided into short system call sequences by moving along the trace with a sliding window. The output probability of a short system call sequence embedded in the testing trace is calculated based on the normal model. If the output probability of a short system call sequence exceeds a preset threshold, the short system call sequence is identified as a “mismatch”. If the ratio of the number of mismatch system call sequences to the number of all sequences embedded in the trace exceeds another preset threshold, the trace is then considered as an intrusion. Experimental results show that the proposed method improves the detection accuracy by at most 590% compared to both Forrest’s and Lee’s methods.

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

入侵检测系统 (IDS) 作为一种重要的网络安全防卫系统, 近年来已经成为计算机与网络安全研究的热点. 目前, 大部分基于主机的 IDS, 都是通过使用不同的方法, 以系统调用作为研究对象来建立正常的程序行为模型, 从而实现入侵检测的. 这些方法包括短序列匹配、数据挖掘、非负矩阵分解、主成分

分析等^[1-4]. 隐马尔可夫模型 (HMM) 可以跟踪系统调用中隐含的状态转移特性, 因此利用它会有较好的检测效果^[5]. 但是, 传统的基于 HMM 的入侵检测方法需分别计算每个系统调用的输出概率和状态转移概率, 在实际的运行环境中, 计算量比较大. 本文通过计算系统调用的短序列输出概率, 提出了一

收稿日期: 2004-11-15. 作者简介: 张响亮(1981~), 女, 硕士生; 管晓宏(联系人), 男, 教授, 博士生导师. 基金项目: 国家杰出青年科学基金资助项目(60243001); 国家自然科学基金资助项目(60243001); 国家高技术研究发展计划资助项目(2001AA140213).

种基于 HMM 的入侵检测新方法,与文献[2,3]的实验结果比较,检测效果有大幅度提高。

1 基于 HMM 的正常程序行为模型

HMM 是一个双重随机过程,即内含一个不可见的(隐藏的)从属随机过程的随机过程,此不可见的从属随机过程只能通过另一套产生观察序列的随机过程观察得到。

假定计算机在正常运行情况下的某个进程在一个时段内产生的长为 T 的系统调用序列定义为观察值 $O = \{O_1, O_2, \dots, O_t, \dots, O_T\}$, O_t 为 t 时刻产生的系统调用;系统调用序列对应的隐含状态序列为 $Q = \{q_1, q_2, \dots, q_t, \dots, q_T\}$, q_t 为 t 时刻所处状态。那么,可以使用 3 个参数描述一个基于 HMM 的正常程序行为模型 $\lambda = (A, B, \pi)$,其中 HMM 涉及到的一些参数描述如表 1 所示^[6]。

表 1 HMM 参数描述

参数	描述
状态数 N	状态的有限集合 $S = \{S_1, S_2, \dots, S_N\}$
观测值数 M	观察值有限集合 $V = \{v_1, v_2, \dots, v_M\}$
初始状态分布 $\pi = \{\pi_i\}$	$\pi_i = P(q_1 = S_i)$
状态转移概率 $A = \{a_{ij}\}$	$a_{ij} = P(q_{t+1} = S_j q_t = S_i)$
输出概率 $B = \{b_j(k)\}$	$b_j(k) = P(O_t = v_k q_t = S_j)$

在 HMM 的实际应用中有以下 3 个中心问题需要解决。

(1) 评估问题:对于给定模型 $\lambda = (A, B, \pi)$ 和观察值序列 $O = \{O_1, O_2, \dots, O_T\}$, 求产生给定观察值序列的概率 $P(O | \lambda)$ 。用它可衡量给定模型与此观察值序列之间的匹配情况。

(2) 解码问题:对于给定模型和观察值序列,求可能性最大的隐含状态序列。

(3) 学习问题:对于给定的一个观察值序列,调整参数,使得观察值的输出概率 $P(O | \lambda)$ 最大。也就是,用给定的观察值序列去训练 HMM 模型,优化模型参数。

要建立正常的程序行为模型,就是由程序在正常运行过程中产生的系统调用序列确定 HMM 的参数 $\lambda = (A, B, \pi)$,也就是解决 HMM 的 3 个基本问题中的学习问题。本文使用标准的 Baum-Welch (BW) 算法进行参数估计,以解决 HMM 中的学习问题。通过不断地重估参数、调整状态转移概率和输

出概率,直到 $P(O | \lambda)$ 局部最大。

建立 HMM 需要确定状态数 N 。入侵检测的实验结果表明,如果状态数等于训练数据中系统调用类型的个数,就会获得较好的检测效果^[5]。因此,在入侵检测时,状态数的选取与实验用到的具体数据集有关。

计算输出概率 $P(O | \lambda)$ 首先要解决评估问题,这可以通过“前向-后向”算法来实现。定义前向变量 $\alpha_t(i)$ 为给定 HMM 参数 λ , 输出部分观察序列 $\{O_1, O_2, \dots, O_t\}$, 则在 t 时刻处于状态 S_i 的概率 $\alpha_t(i) = P(O_1 \dots O_t, q_t = S_i | \lambda)$; 定义后向变量 $\beta_t(i)$ 为给定 HMM 参数 λ , 观察序列在 t 时刻处于状态 S_i , 则系统输出部分观察序列 $\{O_{t+1}, O_{t+2}, \dots, O_T\}$ 的概率 $\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T, q_t = S_i | \lambda)$ 。

通过前向、后向变量,可以得到输出概率

$$P(O | \lambda) = \sum_{i=1}^N \alpha_1(i) \beta_1(i) = \sum_{i=1}^N \sum_{j=1}^N \alpha_1(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) = \sum_{i=1}^N \tau(i) \quad (1)$$

在用 BW 算法训练 HMM 时,还需要定义另外 2 个变量 $\alpha_t(i, j)$ 和 $\beta_t(i)$ 。 $\alpha_t(i, j)$ 为给定模型 λ 和观测序列 O 在 t 时刻处于状态 S_i , 而在 $t+1$ 时刻处于状态 S_j 的概率,即

$$\alpha_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \frac{P(q_t = S_i, q_{t+1} = S_j, O | \lambda)}{P(O | \lambda)} \quad (2)$$

$\beta_t(i)$ 为给定模型 λ 和观测序列 O 在 t 时刻处于 S_i 状态的概率,即

$$\beta_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{P(O | \lambda)} \quad (3)$$

从 $\alpha_t(i, j)$ 和 $\beta_t(i)$ 的定义可以得出,如果定义 E_{ij} 为从状态 S_i 转移到状态 S_j 的期望次数, E_{if} 为从状态 S_i 转移的期望次数, E_i 为处于状态 S_i 的期望次数,则有

$$E_{ij} = \sum_{t=1}^{T-1} \alpha_t(i, j) \quad (4)$$

$$E_{if} = \sum_{t=1}^{T-1} \alpha_t(i) \quad (5)$$

$$E_i = \sum_{t=1}^{T-1} \alpha_t(i) \quad (6)$$

利用式(4)~式(6)就可以得到 HMM 参数重估的方法。定义 E_{jv_k} 为处于状态 S_j 并且观测值为 v_k 的期望次数,一种新的重估模型 $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ 可以通过下面的等式得到,即

$$\bar{\pi}_i = \alpha_1(i) \quad (7)$$

$$\bar{a}_{ij} = \frac{E_{ij}}{E_j} = \frac{\sum_{t=1}^{T-1} \pi_t(i, j)}{\sum_{t=1}^T \pi_t(i)} \quad (8)$$

$$\bar{b}_j(k) = \frac{E_{jv_k}}{E_j} = \frac{\sum_{t=1, s.t. O_t=v_k}^T \pi_t(j)}{\sum_{t=1}^T \pi_t(j)} \quad (9)$$

随机给定初始模型 $\bar{A} = (A, B, \pi)$, 利用训练的系统调用序列 O , 通过式(7)~式(9)可以计算重估后的模型 $\bar{A} = (\bar{A}, \bar{B}, \bar{\pi})$. 从而得知, \bar{A} 将比 A 更接近实际模型, 即 $P(O|\bar{A}) > P(O|A)$. 用 \bar{A} 代替 A , 并重复进行上述重估过程, 直到满足某一限制条件为止, 例如 $P(O|\bar{A}) - P(O|A)$ 的值小于某个给定阈值. 可以看出, 重复进行重估计算, 就可以提高训练序列 O 的输出概率, 这个重估程序的最终结果被称为 HMM 的最大似然估计. 训练结束后, 得到的 $\bar{A} = (A, B, \pi)$ 即为正常程序行为的 HMM 模型.

2 异常检测

正常程序行为的 HMM 模型建立完成之后, 在检测某个给定程序产生的系统调用序列时, 应先对测试序列用长为 k 的滑动窗口进行分割, 滑动窗口每次向后移动一位. 假设测试序列长为 T , 则短序列集包含 $C = (T - k + 1)$ 个长为 k 的滑动窗口短序列, 于是可以得到系统调用短序列集 $\{X_i\} (1 \leq i \leq T - k + 1)$. 对每个短序列 X_i 按照式(1)求输出概率, 为了明显突出输出概率之间的差别, 可取输出概率为 $\lg P(X_i|\bar{A})$.

将短序列集中的每个短序列的输出概率与给定的输出概率阈值 α 进行比较, 并将小于 α 的短序列标定为“不匹配”. 然后, 统计并求出测试序列中“不匹配”的短序列数 C_{mis} 与总的短序列数 C 的比值, 该比值定义为异常度

$$= \frac{C_{mis}}{C} \quad (10)$$

再将求得的测试序列的异常度与另一给定的异常度阈值 α_a 进行比较, 如果异常度大于 α_a , 就认为产生此测试序列的程序可能为异常的; 否则, 认为测试序列的程序是正常的. 实验中, k 分别取值为 3、6、7、10 和 11.

3 入侵检测实验

3.1 实验数据

为了便于比较实验结果, 本文使用美国新墨西

哥大学 (UNM) 采集到的 sendmail 守护进程在正常运行时产生的系统调用数据和入侵进程产生的数据作为实验数据源 (<http://www.cs.unm.edu/~immsec/data-sets.htm>), 该数据描述如表 2 所示.

表 2 实验数据描述

	数据名称	T	进程数
训练数据	sendmail	24 075	117
正常测试数据	sendmail	37 596	47
	local-1	1 516	6
异常测试数据	local-2	1 574	6
	remote-1	1 861	7
	remote-2	1 553	4
	sm565a	275	3
	sm5x	1 537	8

训练时, 首先选取模型中的状态数 N , 本文所使用的训练数据共有 53 个不同的系统调用, 因此可确定模型中状态数 $N = 53$.

3.2 实验结果

采用 5 种不同的 k 进行实验, 结果如表 3 所示. 为了便于比较, 还列出了相同数据集下的 Forrest^[2]、Lee^[3] 方法的实验结果.

表 3 不同 k 下的 HMM 方法与 Forrest、Lee 方法的比较

	Forrest		Lee		本文方法			%	
	$k=11$	$k=7$	$k=11$	$k=10$	$k=7$	$k=6$	$k=3$		
正常测试数据	0.00	0.60	0.00	0.35	0.55	0.31	0.28		
异常测试数据	local-1	4.00	6.10	6.31	16.72	14.97	11.85	7.46	
	local-2	5.30	8.00	6.39	19.23	17.22	13.96	8.46	
	remote-1	5.10	11.50	14.32	25.59	23.67	21.01	14.42	
	remote-2	1.70	8.40	11.73	21.70	20.43	18.28	13.41	
	sm565a	0.60	8.10	4.91	27.74	24.91	19.63	13.19	
sm5x	2.70	8.20	2.75	28.08	24.04	19.45	10.88		

由于实验中用到的训练数据比较多, 所以训练所消耗的 CPU 时间比较长, 约为 200 min, 测试耗用的 CPU 时间如表 4 所示.

表 4 HMM 方法的测试耗用的 CPU 时间 s

	CPU 时间				
	$k=11$	$k=10$	$k=7$	$k=6$	$k=3$
正常测试数据	92.6	86.6	68.5	62.3	45.2
local-1	3.57	3.25	2.48	2.08	1.28
local-2	3.62	3.34	2.49	2.16	1.31
异常测试数据					
remote-1	4.25	3.96	2.89	2.58	1.56
remote-2	3.20	3.01	2.26	1.98	1.22
sm565a	0.56	0.49	0.37	0.34	0.22
sm5x	2.99	2.78	2.13	1.87	1.17

4 实验结果的分析与讨论

从表 3 可以得出以下结果.

(1) 异常测试数据的异常度比正常测试数据的异常度明显大得多,因此使用本文方法很容易准确地将程序的正常行为与异常行为区分出来.

(2) 在 k 相同、正常测试数据的大体相当的情况下,使用本文方法得到的异常测试数据的比 Forrest 和 Lee 这 2 种方法大得多,这说明利用本文方法可以更有效、更准确地将异常行为检测出来,检测效果要好于这 2 种经典方法.

分析表 3 和表 4 的结果,可以得出以下结论.

(1) k 的大小会影响, k 越大,正常测试数据与异常测试数据的差距就越大,运算量也随着 k 的增加而增大,消耗的 CPU 时间亦随之增加.

(2) HMM 在正常程序行为的建模过程中用于训练的时间较长,2 万多条系统调用需耗时约 200 min 这也是考虑数据的转移特性以提高入侵检测的精度所必须付出的代价.但是,一旦模型建好,测试过程则比较简单,近 4 万条系统调用在测试过程中消耗的时间仅在 1 min 左右,因此基于 HMM 的程序异常行为检测可以通过离线训练、在线检测来实现.

5 结 论

本文提出了一种基于 HMM 的程序异常行为检测新方法.利用 HMM 可以跟踪状态转移的特点,用正常的系统调用序列建立系统正常程序行为

模型.在检测时,用系统调用短序列作为研究对象,根据短序列与正常行为模型的匹配程度进行标定,并统计“不匹配”短序列所占的百分比,进而判断系统调用所在的程序行为是否异常.实验结果表明,使用本文方法的检测效果要优于其他几种经典方法.在接下来的研究中,我们将寻找比 BW 更快的训练算法以建立程序正常行为的 HMM 模型,从而减少训练所消耗的时间.另外,本文使用 HMM 考虑了数据的转移特性,在提高入侵检测精度的同时,训练和测试时间也较长,这不利于实时入侵检测.因此,在今后的研究工作中,还应考虑将 HMM 方法与其他考虑了数据频率特性的方法融合起来,借此减少建模时间,提高入侵检测的实时性.

参考文献:

- [1] Forrest S, Hofmeyr S A, Somayaji A, et al. A sense of self for Unix processes [A]. 1996 IEEE Symposium on Security and Privacy, Oakland, USA, 1996.
- [2] Lee W, Stolfo S. Data mining approaches for intrusion detection [A]. 7th USENIX Security Symposium, Berkeley, USA, 1998.
- [3] Wang Wei, Guan Xiaohong, Zhang Xiangliang. Profiling program and user behaviors based on non-negative factorization for anomaly intrusion detection [A]. 43rd IEEE Conference on Control and Decision, Nassau, Bahamas, 2004.
- [4] Wang Wei, Guan Xiaohong, Zhang Xiangliang. A Novel intrusion detection method based on principal component analysis in computer security [A]. International IEEE Symposium on Neural Networks, Dalian, China, 2004.
- [5] Warrender C, Forrest S, Pearlmuter B. Detecting intrusions using system calls: alternative data models [A]. 1999 IEEE Symposium on Security and Privacy, Oakland, USA, 1999.
- [6] Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition [J]. Proceedings of the IEEE, 1989, 77(2): 257-289.

(编辑 苗 凌)