

文章编号: 1001-9081(2005)02-0383-03

基于改进否定选择匹配算法的异常检测

肖晓丽, 田悦宏, 陈 川

(长沙理工大学 计算机与通信工程学院, 湖南 长沙 410076)

(ttxxl@163.net)

摘 要: 使用了一种改进的否定选择匹配算法来检测异常行为。在这种算法中考虑了位置因素对两个序列匹配度的影响, 从而能够更加准确识别自体与非自体, 有效地减小检测集的规模。首先使用正常的序列调用生成初始检测集, 然后通过学习来扩充检测集, 使用最终得到的检测集扫描一定长度的调用序列, 通过其中异常序列的比例来显示该段序列调用是否出现了异常。最后给出了实验结果。

关键词: 异常检测; 否定选择算法; 序列匹配

中图分类号: TP393.08 **文献标识码:** A

Anomaly detection based on improved negative selection matching algorithm

XIAO Xiao-li, TIAN Yue-hong, CHEN Chuan

(College of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha Hunan 410076, China)

Abstract: A matching algorithm based on the negative selection for anomaly detection was presented in this paper. In the algorithm the effects of position between two temporal sequence to matching degree were considered. So it could distinguish accurately self and non-self, and reduced the size of detective set effectively. Using normal sequence calls, the initial detective set was created, and the detective set was extended by learning, according to the proportion of anomaly temporal sequence to judge whether this sequence was anomaly. Finally, the results of experiment was given.

Key words: anomaly detection; negative selection algorithm; sequence matching

0 引言

异常检测(anomaly detection)是目前入侵检测系统(IDS)的主要研究方向之一。异常检测^[1]的思想最早由 Denning 提出, 通过监视系统审计记录上系统使用的异常情况, 可以检测出违反安全的事件。异常检测的关键问题在于正常使用模式的建立以及如何利用该模式与当前系统/用户行为进行比较, 从而安全判断出与正常模式的偏离程度。当偏离程度较大即认为是异常, 从而触发相应的安全机制。

文献[2]基于免疫系统自体-非自体识别原理开发的否定选择算法, 通过定义自体为一个长度为 L 的字符串的多个集合 S 表示“自体”; 然后产生一个集合 R , 每个 R 中的检测器与任何 S 中的字符串都不匹配。通过不断地将 R 中的检测器与 S 比较来监控 S 的改变。符号字符串中的部分匹配有许多定义方法, 比如海明距离或者编辑距离。图 1 的形状空间示意图表示了人工免疫系统中抗体与抗原的关系。

实际中异常调用序列的获取相对于正常调用序列来说比较困难, 而且从安全的角度考虑通过识别正常序列来判断异常是相对安全的措施。因此本文中使用正常调用序列来建立, 并在此基础上使用一种新的序列匹配方法来进行这种短

序列匹配, 通过合适的选取检测器和阈值来减小检测集的规模, 从而提高系统识别速度。检测集合定期的学习更新使得能够更加准确地进行检测。

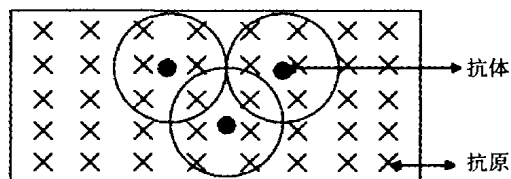


图 1 形状空间示意图

1 模型建立

1.1 短序列划分

系统中进程的行为可以用它所发出的系统调用序列来描述。对应于正常行为和异常行为系统调用的统计特征是不同的。如果某个进程所发出的系统调用序列的统计特征和正常行为调用序列的统计特征有很大的差别, 则可以确定该进程出现了异常。为了分析一段时间内的进程调用是否出现异常, 必须按照一定的方法将得到的大量调用序列划分成用于检测的短序列。

本文将正常的系统调用序列按照长度为 k 步长为 L 的滑

收稿日期: 2004-07-17; 修订日期: 2004-10-21

作者简介: 肖晓丽(1965-), 女, 湖南邵阳人, 副教授, 主要研究方向: 计算机网络; 田悦宏(1980-), 男, 硕士研究生, 陕西汉中, 主要研究方向: 计算机网络; 陈川(1960-), 女, 四川人, 副教授, 主要研究方向: 计算机网络。

动窗分割成若干短序列。根据 Forrest 等提供的系统调用数据,序列(4,2,66,66,138,66,5,23,45,4)是一个正常的系统调用序列。其中每一个数字代表一个系统调用。例如,数字5代表系统调用 open。当 $k=4, L=2$ 时上述序列将划分成4个短序列:(4,2,66,66);(66,66,138,66);(138,66,5,23);(5,23,45,4)。

当扫描整个系统调用序列后,便得到关于该系统调用的一短序列集合 $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$, 其中 $\alpha_i \in R^L$ 。

1.2 建立初始检测集

由于实际系统调用序列的随机性,使用通常的相关性算法并不能准确表达两个序列之间的相似程度。如下所示的两个序列:

- a) open, read, getrlimit, mmap, open, mmap
- b) open, mmap, read, getrlimit, mmap, open

这两个序列应该具有较好的相似度,但是常用的海明距离不能很好的表达 a)、b)之间的相似程度。在本文中采用子串相关函数并在其中加入位置因素来计算两个序列的相关性。

对于长度为 k 的两个序列 $X = (x_1, x_2, \dots, x_k)$ 和 $Y = (y_1, y_2, \dots, y_k)$ 的相关性由下面函数定义:

$$w(X, Y, i, j) = 1 + w(X, Y, i, j-1), \text{ 如果 } X_{i,j} = Y_{i,j}$$

$$w(X, Y, i, j) = 0, \text{ 如果 } j < 0 \text{ 或者 } X_{i,j} \neq Y_{i,j}$$

$w(X, Y, i, j)$ 表示两个序列中第 i 个完全匹配的两个子序列中第 j 个位置的匹配程度。

$$w(X, Y, i) = (k-1) + \sum_j w(X, Y, i, j) - |\beta_i|$$

$w(X, Y, i)$ 表示这两个子序列的匹配程度(这里 β_i 为两个子序列位置之差)。如果子串连续匹配,那么这两个子串的匹配将呈梯状递增。

通过计算所有的 $w(X, Y, i)$ 可以得到两个序列的相关程度:

$$Sim(X, Y) = \sum_i w(X, Y, i)$$

$$Sim_{\max} = Sim(X, X)$$

图2显示了使用该方法计算序列 a)、b) 的匹配度,最长的匹配子序列长度为4 位置偏移为1。使用上述公式可以得到序列 a)、b) 的相关性 $Sim(a, b) = 22, Sim_{\max} = 26 (k=6)$ 可以看出使用上述公式可以准确的表达 a)、b) 的相关程度。

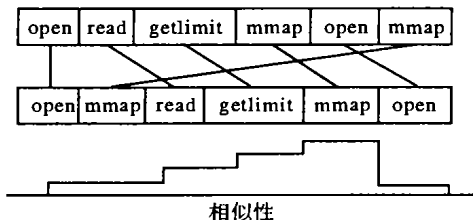


图2 计算序列 a)、b) 的匹配度

为了提高系统识别效率,还必须将得到序列集合 α 按照上述算法进行去冗余。通过给定一阈值可以将具有一定程度上匹配的短序列删除。如果 $Sim(X, Y) \geq \eta, X, Y \in \alpha (\eta$ 为阈值), 表明 X, Y 具有合适的相似度, 则从检测集合中删除其中

一个具有较低覆盖率的序列。按照这种方法计算所有检测集中的元素,最后得到的初始检测集合必定是最少的具有最大覆盖率的集合。反之如果 $Sim(X, Y) < \eta$ 则说明这两个序列对应不同的检测器,必须保留在检测集合中。通过这种方法可以有效地减小检测集的规模。

1.3 扩充检测集

初始检测集只代表了一部分正常数据调用序列的分布情况。在实际检测中正常调用序列是随机变化的,因此检测集合必须定期更新以适应这种变化。可以从系统守护程序得到数据来提取正常的序列,从而更新检测集。正常的系统调用数据点分布在一定时间段内具有一定的规律性^[3]。我们从 Forrest 提供的正常数据集中提取部分数据点(如图3所示),可以看出数据点的分布在一定程度上是有一定冗余的。为了提高更新效率对由守护程序得到数据必须进行去冗余计算。

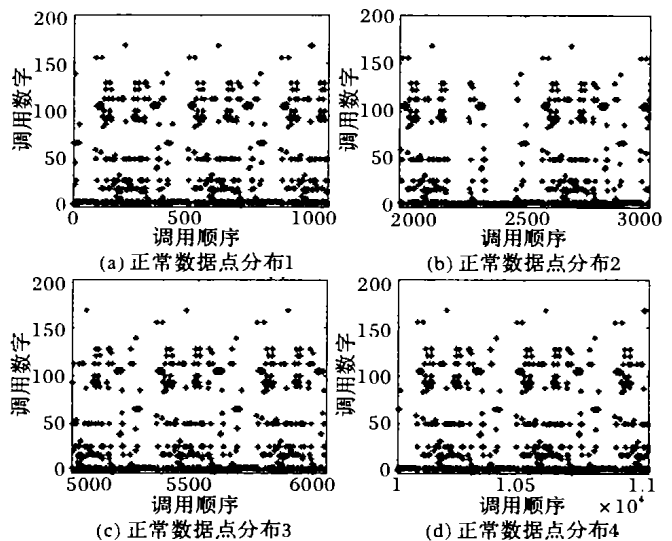


图3 正常数据集中提取部分数据点

假设正常系统调用序列为 C , 按照步长 k 划分该序列得到集合 G 。 $\forall Y \in G$ 计算:

$$Sim_D(Y) = \text{Max}\{Sim(X, Y)\}, \text{ 其中 } X \in D$$

D 为得到的初始检测集合。如果 $Sim_D(Y) \geq \eta$ (η 为阈值) 表明初始检测集中有与 Y 相似的序列, 反之说明对于 D 来说序列 Y 是一个新的正常序列, 所以将 Y 加入检测集 D 。通过这种方法可以建立比较完善的检测集合 \tilde{D} 。

1.4 异常检测

假设监测到一段系统调用序列 C , 按照步长 k 划分该调用序列得到待检测集合 G 。 $\forall Y \in G$ 计算:

$$Sim_D(Y) = \text{Max}\{Sim(X, Y)\}, \text{ 其中 } X \in \tilde{D}$$

如果 $Sim_D(Y) \leq \eta$ 则标记序列 Y 为异常, 统计序列 C 中的所有标记为异常序列。如果异常序列比例超过一定的门限值则认为此段系统调用序列异常。反之则标记该段序列为正常。

2 实验结果

文献[4]中说明了窗口大小与训练数据的条件熵以及分

类错误率的关系,认为滑动窗口长度的理想值为 6。本文中也是用这个结果,并取步长为 3,使用 Matlab6.5 来建立模型并进行数据分析。

实验的原始数据包括正常的系统调用序列:sendmail 守护程序的调用序列加上几个 sendmail 程序调用。异常的系统调用序列:3 个 sscp 攻击序列,2 个 sys-local 攻击序列,2 个 sys-remote 攻击序列,2 个 decode 攻击序列,1 个 sm5x 攻击序列和 1 个 sm565a 攻击序列。

我们使用一部分 sendmail 调用序列来构建初始检测集。在构建过程中要求检测集中的序列可以通过相关度(1,2)和一定的阈值覆盖整个正常序列集,可以通过这种方法确定试验中阈值的选择。

根据正常调用序列数据我们得到了阈值和检测集规模之间的关系,如图 4,从中可看出考虑位置因素后当阈值大于 17 时,检测集的规模将趋于稳定。而一般的按序列位进行匹配算法则过早的趋于稳定。因此,在实验中阈值取为 17。并得到阈值为 17 所对应的初始检测集。

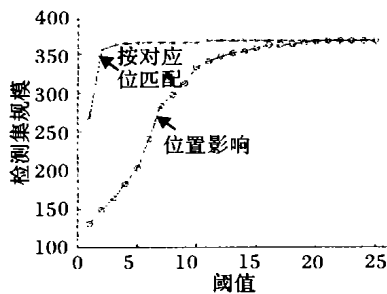


图 4 阈值与抗体规模

利用已有的初始检测集对大量的正常数据进行计算并将满足条件 $Sim_D(Y) < \eta$ 的序列添加到检测集中。实验中从 sendmail 守护程序中提取部分数据进行冗余处理后学习,在学习过程中,初始检测集规模不断增大但是增大幅度不断减小并趋于稳定(图 5)。最后得到的检测器的数量在 500 左右。而一般的按序列的对应位匹配的算法收敛速度比较慢从而导致检测集规模的迅速扩大。可以看出通过这种改进的匹配算法可以有效的减少检测集的规模从而提高系统检测效率。

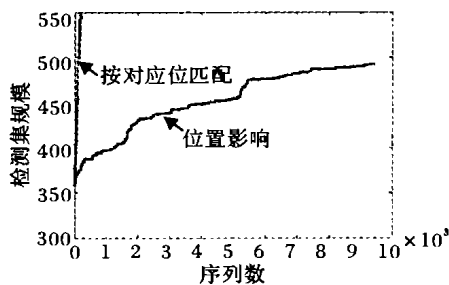


图 5 检测集规模变化图

表 1 给出了使用上述方法对得到的检测集进行检测的实验结果,其中正常 sendmail 序列的异常区域表明该方法所产生的误报率。从试验结果可以看出正常的调用序列的异常区域和异常调用序列之间的区别非常大,因而证明这种方法是有效的。

表 1 检测的实验结果

序列类型	异常区域 (%)	序列类型	异常区域 (%)
正常 sendmail 序列	0.14	Syslog-local-2 攻击	8.40
Syslog-remote-1 攻击	9.35	Sm5x 攻击	7.42
Syslog-remote-1 攻击	3.88	Sm565a 攻击	6.67
Syslog-local-1 攻击	6.75	Sscp 攻击	8.06

图 6 给出了试验中四种攻击序列阈值的波动,可以看出攻击序列的阈值只是在一定时间段内的波动非常大,所以在实际检测过程中应一次提取一段系统调用作为待检测集。这样可以减小检测攻击时的正常序列的干扰。

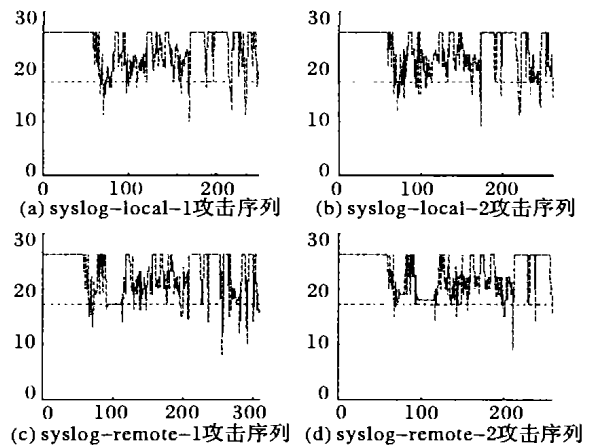


图 6 四种攻击序列阈值的波动

3 结语

为了提高系统检测异常行为的效率,本文使用一种与位置相关的匹配算法,并且为了反映最新的正常行为,检测集可定期更新。为了提高检测集更新效率对得到的正常调用序列首先进行去冗余计算。这种检测方法在实验中取得了比较理想的结果,有效的减小了检测集的规模。当然,也不能排除用于试验的数据比较简单和理想等因素。另外本文中使用的匹配算法能够很好的表达两个序列间的相似程度,这种匹配算法不仅可以用于异常检测,也可以用于其他需要进行匹配计算的算法中。

参考文献:

- [1] DENNING DE. An Intrusion Detection Model [J]. IEEE Trans on Soft Engineering, 1987, 13(2): 222-232.
- [2] FORREST S, HOFMEYR SA. Immunology as Processing Design Principles for Immune Systems & Other Distrubuted Autonomous Systems[M]. SEGAL A, COHEN IR eds Oxford Univ Press, 2000.
- [3] FORREST S, HOFMEYR SA, SOMAYAJI A. A Sense of Self for Unix Process[A]. Proceedings of 1996 IEEE Symposium on Computer Security and Privacy [C]. Oakland, California: IEEE Computer Society Press, 1996. 120-128.
- [4] LEE W. A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems [J]. PhD thesis, Columbia University, 1999.
- [5] LANE T, BRODLEY CE. Temporal Sequence Learning and Data Reduction for Anomaly Detection[J]. ACM Transactions on Information and System Security, 1999, 2(3): 295-331.