基于状态转换图的入侵检测模型STGIDM*)

姚立红 黄 皓 谢 立
(南京大学计算机科学与技术系 南京210093)
(南京大学计算机软件新技术国家重点实验室 南京210093)

A Intrusion Detection Model STGIDM Based on State Transition Graph

YAO Li-Hong HUANG Hao XIE Li
(Department of Computer Secience and Technology, Nanjing University ,Nanjing 210093)
(State Key Laboratory for Novel Software Technology, Nanjing University,Nanjing 210093)

Abstract Pattern Match based on the serial of system calls is one of the widely used intrusion detection technologies, while the courses that cause intrusion frequently accord with chosen pattens when programs monitored are complex and the number of chosen patterns is large, which always causes some intrusions to be neglected. In this article, a new Intrusion Detection Model based on State Transition Graph is put forward. In this model, the programs' properties are accurately reflected by State Transition Graph, and transitions are introduced between states that aren't conjoint. Moreover, the number of transitions between states that aren't conjoint is one of key factors to deduce intrusions during matching.

Keywords Intrusion detection,State transition graph,Pattern match,Serial of system calls

近年来,计算机系统的安全问题日益突出,相应的安全防范技术也成为了人们研究的热点。入侵检测技术作为一种重要的安全技术,从80年代中期就已经引起人们的注意[1],与其它安全技术相比(如身份认证、访问控制等),入侵检测技术具有鲜明的特点:其它绝大多数安全技术主要强调预防或阻止外来入侵者进入系统,或者控制内部用户获取非法权限;而入侵检测主要在其它安全措施被突破、入侵者正在(或已经)进入系统的情况下发挥作用,该技术主要通过监视系统中的异常行为,及时发现入侵者,并采用相应措施(如断开网络连接、报告管理员等),以避免或尽可能减少入侵造成的损失[2]。

预先收集系统正常运行时的各种行为特征,将系统运行状况与之进行匹配比较以期发现外来入侵是常用的入侵检测技术,基于系统调用序列的模式匹配来发现入侵就是这种技术的具体应用[3]。本文首先对一般的基于系统调用序列的模式匹配技术进行分析和测试,并在此基础上提出了基于状态转换图的入侵检测模型STGIDM,最后给出了这两种检测技术的实验结果。

1 基于系统调用序列的模式匹配及性能分析

1.1 基于系统调用序列的模式匹配技术

系统调用是操作系统对上层提供的唯一接口,系统调用状况在一定程度上反映程序运行的特征,程序运行的异常将在系统调用时有所体现[4,5]。通常的基于系统调用序列的模式匹配技术主要针对系统中的对外服务(ftp服务器等)和特权程序(如UNIX中的Suid程序等),这些程序存在安全漏洞是造成入侵或内部用户获取非法特权的主要原因,监视这些程序的运行状况是否异常就可以检测出大部分外部入侵和内部越权。

受用户输入、运行环境等影响,程序执行路线复杂多变,因而不可能进行整个系统调用序列的完全精确匹配。S.Forrest等人提出了用抽样到的系统调用序列中多次重复出现的短序列描述程序行为特征的方法[6],基于系统调用的模式匹配技术是把这些多次重复出现的短序列作为模式进行匹配以发现系统异常或入侵,具体方法为:多次运行将要被作为入侵检测对象的服务程序,获得多条程序正常执行路线(即系统调用序列),从这些执行路线中抽取多次出现的短序列,把这些短序列作为刻画程序行为特征的模式收集到模式库中。在进行入侵检测时,利用日志系统等获取系统调用序列与模式库中的模式进行匹配,对不能匹配成功的情况进行统计,当达到一个既定的阈值时,即认为该程序受到入侵[7,8]。

1.2 基于系统调用序列的模式匹配的性能分析

按照常见的入侵检测系统的分类方法,基于系统调用序列的模式匹配技术是一种异常检测技术,衡量这种检测技术最主要的两个指标可以解释为:覆盖率,真正的入侵中能被检测出的比率;正确率,检测系统认定的入侵中有多少比率是真正的入侵。表1给出了模式匹配技术在3.1节指定的环境下对proftpd的性能测试结果。

表1 模式匹配技术对proftpd的检测性能

抽样次数	形成模式最低次数	模式数目	入侵次数	检测出的次数	覆盖率%	误检人数	正确率
20	2	30	50	48	96%	198	20%
100	2	131	50	25	50%	15	62%
	3	62	50	46	92%	150	30%
200	2	269	50	22	44%	11	67%
	4	68	50	45	86%	145	23%

表1的测试结果表明,若形成的模式数目少,模式库不能全面体现程序各方面的行为特征,容易产生入侵误报;一般情况下,与任意模式的任意排列相近似的程序执行路线将不会被认定为入侵,随着模式数目的增加,不被认定为入侵的程序执行路线集合的基呈指数级上升,造成入侵的系统调用序列落到这个集合中的几率大大增加。

*)本课题得到国家"863"高技术(NO:2001AA142010)经费资助。姚立红 博士生,研究方向:信息安全、入侵检测。黄 皓 教授,研究方向:信息安全、网络安全;谢 立 教授,博士生导师,研究方向:安全操作系统、分布式智能操作系统。

可见，在模式数目较多时，模式库不能准确刻画程序的行为特征，不能单纯靠模式的匹配情况判断程序受到入侵，而对大型程序没有一定数量的模式很难全面体现其各方面的功能特征，无法降低入侵误报。针对模式匹配技术对大型程序检测入侵的不足，本文将模式之间的时序关系引入到模式库中形成状态转换图来表征程序的行为特征，同时提出了支持状态跳变的新型匹配方案。

2 基于程序状态图的入侵检测模型 STGIDM

鉴于单纯的模式匹配技术用于入侵检测的局限性，STGIDM 模型强调抽样阶段所获得的信息应得到最大限度的保留，并力求在匹配阶段尽可能利用这些信息。具体来讲主要包含两个方面：保留模式是否来源于同一次抽样及是否相邻的信息，并在匹配时通过危险指数的变化来体现；有条件地增加模式的数目，根据出现次数的差别区分不同模式类型，进行不同的处理。为此，STGIDM 模型引入了状态转换图来表示正常状态下的程序执行路线。

为了保留出现次数不多的系统调用序列，STGIDM 将模式分为两类：主模式和次模式。主模式是指反复出现达到一定次数的短序列，次模式指出现次数不太多的短序列，需要的最低次数根据需要抽取多少次模式而定。利用 STGIDM 模型进行入侵检测主要有以下步骤：

(1)对抽样获得的系统调用序列模式化

多次正常运行将要被检测的对象程序，通过一定的方法(系统日志等)[8]获得多条系统调用序列(去掉与系统安全无关的系统调用)表示的程序执行路线，图1给出了在 Linux 系统环境下获得的 FTP 服务的系统调用序列，数字表示相应的系统调用号。

1 102 33 5 3 6 117 102 27 102 102 102 5 3 6 106 12 102 162 33 5 48 19 3 102 102 6 117 102 1

2 122 102 102 48 117 102 102 102 5 3 6 106 12 102 3 102 102 3 102 102 3 102 102 5 19 33 117 33 102 102 162 1

3 122 102 102 162 117 102 12 8 102 4 102 102 4 102 102 4 102 102 6 102 12 5 3 6 102 102 89 89 102 102 162 1

图1 三次启动 FTP 服务获得执行路线片段

抽取多次(本例中为2次)出现的短序列形成主模式(参见文[5,9])，图1中用下划线标出了能形成主模式的系统调用短序列。对抽样到的程序执行路线中不能构成主模式的短序列试图形成次模式。表2列出了图1对应的所有模式。

表2 图1对应的所有模式

主模式名	代表序列
A	122 102 102
B	5 3 6
C	102 102 102
D	106 12 102
E	3 102 102
F	102 4 102
G	102 102 162

次模式名	代表序列
a	117 102 27
b	162 33 5
c	6 117 102
d	5 19 33
e	162 117 102
f	102 6 102
g	102 102 89

(2)建立并简化状态转移图

通过形成模式，用系统调用序列表示的程序执行路线也就演化成长度短得多的用模式序列表示的程序执行路线。在模式序列的基础上，建立表示程序执行路线的状态图，每一条有向边代表一模式，并在保证状态图等价的前提下进行必要的简化，如图2所示。

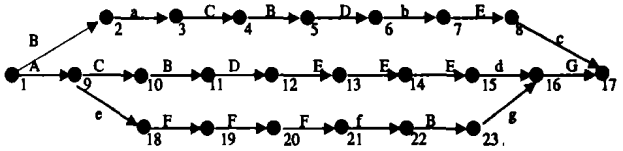

图2 与图1对应的等价简化后的状态图

若状态图比较复杂，还可以对状态图进行一些合理的非等价简化。部分相同执行路线的合并就能大幅度简化状态图，也很少改变状态图的表现能力。假定 f(x,y)表示状态 x 经过模式 f 到达状态 y，若存在下面两条程序执行片断：

A_1: $f_1(a_1,a_2)$、$f_2(a_2,a_3)$、……、$f_i(a_i,a_{i+1})$、……、$f_n(a_n,a_{n+1})$

A_2: $f_1(b_1,b_2)$、$f_2(b_2,b_3)$、……、$f_i(b_i,b_{i+1})$、……、$f_n(b_n,b_{n+1})$ $(n>=2)$

那么，可去掉状态节点 $b_i(1\leqslant i\leqslant n+1)$，以 b_i 结尾的边改为以 a_i 结尾，以 b_i 开始的边改为以 a_i 开始，即 a_i 将承担 b_i 的输入边和输出边，如图3。

(3)静态扩充状态图的表现能力

从状态图形成过程看，状态图的表现能力非常有限，只能表达抽样到的程序执行路线，有必要从抽样到的程序执行路线推测出其它程序执行路线加入到状态图中。比如循环在大多数程序中非常普遍，而循环的次数往往会随具体情况做变动，需要把它反映到状态图中，如图3。

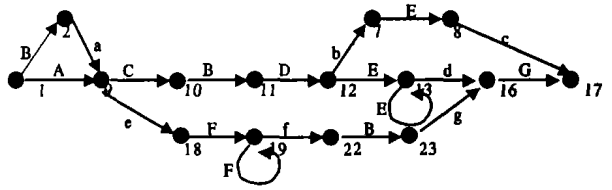

图3 简化和静态扩充后的状态图

(4)动态扩充状态图的表现能力：定义状态转换规则

静态扩充能力非常有限，若在实际的入侵检测过程中严格按照状态图进行匹配，那么没有抽样到的正常程序执行路线很大一部分将被认定为入侵，有必要对状态转换规则进行改进，即在无法转换到下一个状态时，可以转移到不相邻状态。这种不相邻状态间转移比常规的状态转移包含有更大的入侵可能性，需要在危险指数的变化上反映这种可能性。由于形成模式后才能进行状态转移，需要设置与模式等长接收缓冲区队列暂存接收到的系统调用，待队列满时，再进行状态转移。同模式匹配技术一样，STGIDM 模型也设定危险指数以体现所检测程序路线构成入侵的可能性，当危险指数超过一定上限时，即认为监视对象受到入侵。具体的处理规则如下：(假定模式长度为 m，当前状态为 x，接收队列中的内容为$(s_1,\cdots,s_m)$)

① 若序列$(s_1,\cdots,s_m)$与模式 f 匹配成功，且存在转换路径 f(x,y)，那么下一状态跳转至 y，同时按表3处理缓冲区队列和危险指数，其中 O(m)表示与模式长度相关的常数。

表3 相邻状态转移

变化 \ 模式 f 的类型	主模式	次模式
危险指数	减 O(m)	减 O(m)/2

表4 不相邻状态转移

模式f的类型 / 变化	主模式	次模式
危险指数	减O(m)/2	不变

② 若序列$(s_1,\cdots,s_m)$与模式f匹配成功，且存在转换路径$f(x',y),x'<>x$，那么下一状态跳转至y，同时按表4处理缓冲区队列和危险指数。

③ 没有模式匹配成功，不进行状态跳转，队列缓冲区$(s_1,\cdots,s_m)$的第一个系统调用s_1出队列，系统危险指数加1个单位。

3 测试结果比较与分析

3.1 测试结果比较

该测试基于的操作系统环境是Linux(核心版本2.2.10)系统，在该系统支持的190种系统调用中，筛选出40种与系统安全关系密切的系统调用。同时修改0x80号自陷入口以获得所需进程的系统调用序列。入侵监视对象是一个服务程序proftpd (Verison 1.2.0pre4)，该程序被发现有缓冲区溢出等漏洞。此外，为了便于入侵成功和性能比较，特意新添一些漏洞。

程序执行路线的样本数目及其代表性在很大程度上决定了所获得模式库或状态图刻画程序运行特征的能力。表5给出了在相同的样本上的测试结果。

从表5可以看出，在样本数目较多的情况下，STGIDM模型两个方面的性能指标都达到可以接受的程度，它比单纯的模式匹配技术在入侵漏报方面得到很大的改善，覆盖率能够达到90%左右，尽管误报次数有所增加，由于正确检测的次数有明显增加，误报率没有明显上升，基本维持在40%以下。

表5 测试结果比较

样本数目	模式长度	形成(主/次)模式最低次数	抽取的(主/次)模式数	检测技术	入侵次数	检测出的次数	误检次数
100	3	2	131	模式匹配	50	25	15
		3/2	62/69	STGIDM	50	46	26
250	3	2	269	模式匹配	50	22	11
		4/2	68/201	STGIDM	50	45	24

3.2 结果分析

通过对被模式匹配认为正常的程序执行路线的具体分析发现：真正正常的程序执行路线常常对应程序各种功能的特定组合，一般来讲通过有限的几个或几十个抽样执行路线片段就能大致组合出来，而入侵的程序执行路线由于不对应正常功能，匹配成功的模式将会散乱地分布到许多条抽样执行路线上。所以，在STGIDM模型中，入侵的程序执行路线在通过状态图匹配时要频繁进行不相邻状态的转移；而对正常的程序执行路线来讲，只在组合该路线的有限片段的接头处才需要状态跳变，在每个片段的内部可以实现连续的匹配。

比较STGIDM模型的规则(2)和(3)可发现，通过状态跳变匹配成功与不通过状态跳变匹配成功对危险指数的影响不同，前者匹配成功一次相当于后者两次匹配成功，在其它条件相当的情况下，前者对应的程序执行路线更容易被STGIDM模型认定为入侵。因而，STGIDM模型通过考察执行路线匹配成功模式间的相邻关系，能够检测出许多不能被单纯的模式匹配技术发现的入侵，大幅度减少入侵漏报现象。

结束语 模式匹配技术是近年来开始出现的重要入侵检测技术，日益受到人们的重视。由于单纯利用模式库很难描述程序的特征，使得对大型服务程序的入侵检测效果不很理想，本文提出的STGIDM模型较好地解决了这个问题。本文提出的模型已经在Linux平台得到实现，收到了预期的效果。

参考文献

1 Denning D E. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, 1987, SE-13(2): 222～232

2 Escamilla T. Intrusion Detection: Network Security beyond the Firewall John Wiley & Sons, Inc, ISBN: 0471290009, 1998. 1～15

3 Wespi A, Debar H. Building an Intrusion-Detection System to Detect Suspicious Process Behavior, Second International Workshop on the Recent Advances in Intrusion Detection, 1999

4 Somayaji A, Forrest S. Automated Response Using System-Call Delays, 9th USENIX Security Symposium, Aug. 2000

5 Kosoresow A P, Hofmeyr S A. Intrusion Detection via System Call Traces. IEEE Software, 1997, 14(5): 35～42

6 Forrest S, et al. A sense of self for Unix processes. In: Proc. of 1996 IEEE Symp. Security & Privacy. 120～128

7 Debar H, Dacier M, Wespi A. Reference Audit Information Generation for Intrusion- Detection Systems. In: 14th Int'l Information Security Conf., Vienna, Austria & Budapest, Hungary, Aug. 1998

8 Wespi A, et al. Audit Trail Pattern Analysis for Detecting Suspicious Process. In: First Intl. Workshop on the Recent Advances in Intrusion Detection, 1998

9 Dehar H, Dacier M, Nassehi M, Wespi A. Fixed vs Variable-length Patterns for Detecting Suspicious Process Behavior. In: 5th European Symposium on Research in Computer Security, 1998. 1～15