

基于系统调用的入侵检测新方法^{*}

刘雪飞^{1,2,3}, 王申强⁴, 吴伯桥⁵, 马恒太³

(1. 北京机械工业学院 计算机与自动化系, 北京 100085; 2. 南京理工大学 计算机系, 江苏 南京 210096; 3. 中国科学院 信息安全技术工程研究中心, 北京 100080; 4. 北京四方继保自动化股份有限公司, 北京 100085; 5. 湖南信息技术职业学院 计算机系, 湖南 长沙 610200)

摘 要: 通过分析系统调用行为特征, 提出了程序的系统调用行为可以用有限状态自动机来描述的方法, 证明了算法产生的自动机的完整性, 并给出算法性能分析结果。
关键词: 入侵检测; 系统调用; 有限状态自动机

中图法分类号: TP274 文献标识码: A 文章编号: 1001-3695(2006)12-0112-03

New Intrusion Detection Method Based on System Call

LIU Xue-fei^{1,2,3}, WANG Shen-qiang⁴, WU Bo-qiao⁵, MA Heng-tai³

(1. Dept. of Computer & Automation, Beijing Institute of Mechanical Industry, Beijing 100085, China; 2. Dept. of Computer, Nanjing University of Science & Technology, Nanjing Jiangsu 210096, China; 3. Engineering Research Center for Information Security Technology, Chinese Academy of Sciences, Beijing 100080, China; 4. Beijing Sifang Automation Co., Ltd, Beijing 100085, China; 5. Dept. of Computer, Hunan Vocational Institute of Information Technology, Changsha Hunan 610200, China)

Abstract: The paper proposes a method using finite state automation describing programming system call action via analyzing system call character, proving the finite state automation's completeness and giving the result of arithmetic performance.
Key words: Intrusion Detection; System Call; Finite State Automation

入侵检测^[1]的数据源有多种, 包括: 网络数据源, 进一步可分为 Http 流、TCP/IP 流、ICMP 流、SMTP 流等; 报警信息数据源, 报警信息可以是入侵检测系统(IDS)报警信息, 也可以是其他安全系统产生的报警信息; 主机数据源, 进一步可以分为特权程序系统调用、用户击键、审计记录、系统日志等。不同的数据源可以检测的入侵类型、入侵范围不同, 实际上这三种数据源互相补充, 从不同层次、多个检测角度实现对入侵的检测。特权程序系统调用通常是攻击的重点目标, 自 1994 年 Fink 等人^[2]首次提出基于系统调用的入侵检测, 许多研究者^[3,5]以系统调用为研究对象开展对入侵的检测研究。检测特权程序具有如下优点: 特权进程可访问的系统资源多, 影响范围广, 它甚至可以绕过内核的安全审核机制而访问系统资源, 导致特权程序对系统的安全威胁大, 因此, 监视特权程序系统意义大; 特权程序的活动范围一般有较为严格的限制, 行为相对比较稳定, 适合于检测; 特权程序, 尤其是那些监听特定端口的特权程序, 形成了外部探测和入侵的自然边界。当然该方法也有局限性, 如很难检测仿冒攻击。

1 有限状态自动机简介

有限状态自动机^[4]是状态之间的迁移集合。可以用一个

五元组 $(Q, \Sigma, \delta, q_0, F)$ 表示, 其中: Q 表示有限状态集合; Σ 表示有限输入集合(字母表); δ 表示状态之间的迁移集合(等价于 $Q \times \Sigma \rightarrow Q$); q_0 表示起始状态($q_0 \in Q$); F 表示最终状态集合, 是 Q 的子集。
从概念上看, 在操作过程中, 有限状态自动机与一个当前状态 q_c 相关联。 q_c 随着时间发生变化, 即 $q_{c,0} = q_0, q_{c,t+1} = \delta(q_{c,t}, a_t)$ 。笔者猜想, 把程序产生的系统调用作为字母表, 系统调用之间的迁移作为有限状态自动机的迁移, 可以采用有限状态自动机描述程序系统调用行为。

2 程序系统调用行为特征分析和行为表示方式选择

采用程序运行产生的系统调用作为入侵检测的数据源, 首先需要了解系统调用的特征, 然后在此基础上选择合适的表达模式, 该模式能代表系统调用的正常行为, 如果运行程序产生的系统调用行为与正常模式匹配, 就认为当前的系统调用行为是正常的; 否则认为是异常的。

2.1 程序系统调用行为特征分析

通过对程序系统调用的分析, 总结了如下三个特征:

- (1) 程序系统调用行为不像用户行为那样随机波动。程序代码主要由一些声明和固定序列分支(循环、条件或者子程序调用)组成(为简单起见, 仅考虑程序的执行动作, 忽略变量值。如果考虑变量值和程序输出, 则程序行为模型将变得非常复杂)。
- (2) 程序系统调用行为非静态。不同时间、不同运行产生的程序系统调用行为不同, 即数据为非平稳数据。
- (3) 系统调用数据类型为非测度变量。系统调用为类型

收稿日期: 2005-10-19; 修返日期: 2005-12-20
基金项目: 国家“863”计划资助项目(2003AA144030); 国家“973”重点资助项目(G1999035810); 中国科学院软件所基础课题研究基金项目(CXK45634); 北京机械工业学院人才引进基金项目(59525027(J))

变量, 即非测度变量, 两变量之间不能进行相似性计算, 它们可能相同或不同。

2.2 程序系统调用行为表示方式选择

通过对系统调用特征的分析, 笔者认为, 有限状态自动机是程序行为表示的一个好选择。具体分析如下:

(1) 程序行为不具有随机噪音, 应该采用非统计方法, 同时这种非统计方法对任何非期望偏移都显著, 都被视为异常。而通常统计方法用于随机变量, 同时, 统计方法对于小偏移不敏感(实际上这个小偏移并不满足统计显著)。

(2) 程序行为是一些非平稳数据, 这点不满足许多学习算法对数据是稳定的假设。因此, 需要一个“状态”算法。

(3) 数据类型为非测度变量, 这给许多学习方法和归纳技术带来了问题。

有限状态自动机与学习模式准确匹配, 不需要假设数据平稳, 能处理类型数据, 同时, 有限状态自动机与计算机编程映射良好。而其他技术对小偏移不敏感, 需要假定数据平稳, 或者有些不能处理类型数据。

3 系统调用行为有限状态自动机学习算法

3.1 算法思路分析

根据系统调用的有限状态自动机模型来判断是否入侵很直观, 问题的关键是程序系统调用的有限状态自动机模型的建立。笔者采用训练算法, 采集程序多次正常运行的系统调用序列, 从中选择一个系统调用序列建立初始有限状态自动机, 剩下的序列对该自动机进行修正, 以求所建立的有限状态自动机满足所有的训练序列。在修正时, 采用一定的评价算法, 以求建立的有限状态自动机规模最小同时满足所有的训练序列。首先给出算法的总思路: 对当前输入系统调用, 当前状态如果存在一条该系统调用标记的迁移, 则选择该迁移。否则, 根据当前输入系统调用, “评估”所有现存状态, 看该系统调用是否存在一条到某个状态的最合适的迁移, 如果该迁移足够合适, 则创建一条到该状态的迁移。如果没有合适的状态可供选择, 则创建一个新状态, 并建立当前状态到该状态的迁移。

3.2 算法实现

具体算法实现分为两部分, 即有限状态自动机更新算法和在更新时对现有状态进行评价的评价算法。

(1) 自动机更新算法。算法开始于起始状态 q_0 。在每个程序运行之初, 设置当前状态为起始状态, 即 $q_c \leftarrow q_0$ 。下面给出当前系统调用为 a 时, 对自动机 M_c 的更新算法:

如果自动机存在触发当前状态 q_c 的系统调用 a 的迁移(即 $\delta(q_c, a) = q_d, q_d \neq q_N$): $q_c \leftarrow q_d$; 否则, 对当前自动机 M_c 的所有状态 q_i , 创建一个从 q_c 迁移到 q_i 的临时迁移。该迁移的标记为 a (即创建自动机 M_i , 也就是对自动机 M_c 增加 $\delta(q_c, a) = q_i$ 。以此前的系统调用(限定个数) 为根据对 M_i 进行评价(评价算法见后面“当前状态评价算法”), 产生评价结果 R_i , 记住最好的评价 R_b 及所对应的状态 q_b , 删除临时建立的自动机 M_i 。

如果存在最好的状态 q_b 满足 $R_b \geq G$ (G 为阈值), 则对自动机 M_c 添加迁移, 满足 $\delta(q_c, a) = q_d$, 当前状态为 $q_c \leftarrow q_b$; 否则, 创建新状态 q_n , 增加 $\delta(q_c, a) = q_n$, 当前状态 $q_c \leftarrow q_n$ 。

(2) 当前状态评价算法。该算法主要就是尽量从现有状态机中选择一个满足条件的迁移, 尽量减少状态的规模数。具体如下:

初始化匹配计数器 $n = 0$;
对限定个数的此前每个系统调用 a_i , 如果 $\delta(q_c, a_i) = q_d (q_d \neq q_N)$: 增加计数器 $n = n + 1$; 选择匹配的迁移($q_c = q_d$)。
否则, 停止循环(记录 n) ;
返回 n 值。

该算法虽然简单, 但它可以很好地描述系统调用行为, 具有完整性, 同时算法检测性能也不错。

4 算法完整性证明和算法检测性能分析

4.1 算法完整性证明

本文算法产生的自动机满足完整性要求。所谓自动机完整性, 是指自动机接收训练集中所有程序的运行实例。其证明包含两部分: 证明程序的某次运行训练被自动机接收; 证明训练自动机的所有程序运行被自动机接收。证明依赖于学习算法的两个重要性质: 除非训练字符串不被接收, 否则训练不会改变已经创建的自动机。这点很明显, 因为算法总是尽可能选择一条已存在的迁移, 这同样保证了自动机是决定性的。在任何时候创建的自动机包含之前所产生的自动机。这个性质也很容易理解, 因为算法不删除或修改已经存在的状态或迁移。有了这两条性质, 就能保证自动机接收给定字符串的训练, 则它一直接收该字符串。下面给出具体证明。

命题 1: 由字符串创建的自动机接收该字符串。

证明: 由长度为 0 的字符串训练的自动机接收该字符串。

对空字符串 ε 进行测试, 因为 $\delta(q_0, \varepsilon) = q_0$ 自动机开始和结束于起始态, 又因为训练产生的每个状态是最终态, 这样产生的自动机能接收空字符串。

归纳: 假定对给定长度为 n 的字符串进行训练产生的自动机接收该字符串。长度为 $n + 1$ 的字符串 w_{n+1} 训练产生的自动机接收字符串 w_{n+1} 。

给定字符串 $w_{n+1} = a_1 \dots a_{n+1}$, 经过 $w_n = a_1 \dots a_n$ 训练产生自动机 M_n , 现在给定 a_{n+1} 训练 M_n 。由假设可知, M_n 接收 $w_n = a_1 \dots a_n$, 产生某个状态 $q_c = \delta(q_n, w_n)$, 这时有两种情况:

(1) $M_n(q_c)$ 接收 a_{n+1} (即对每个接收状态 $q_d, M_n \delta(q_c, a_{n+1}) = q_d$), 则 $M_{n+1} = M_n$ 接收 $w_{n+1} (\delta(q_0, w_{n+1}) = \delta(q_c, a_{n+1}) = q_d)$ 。

(2) $M_n(q_c)$ 不接收 a_{n+1} , 则学习算法添加一个从 q_c 到某个可接收状态 q_a 标记为 a_{n+1} 的迁移。它能保证 $M_{n+1} \delta(q_c, a_{n+1}) = q_a$, 这样 $M_{n+1} \delta(q_0, w_{n+1}) = M_{n+1} \delta(\delta(q_0, w_n), a_{n+1}) = q_n$, 因为 q_a 是一个可接收状态, 因此 M_{n+1} 接收 w_{n+1} 。

证毕。

命题 2: 自动机接收所有训练自动机的字符串。

证明: 已经证明由一个字符串训练的自动机接收该字符串。

归纳: 假定经过 n 个字符串 $w_1, \dots, w_n$ 训练的自动机接收这些字符串, 则经过字符串 w_{n+1} 训练之后, 它将接收所有字符串 $w_1, \dots, w_{n+1}$ 。

在 w_{n+1} 上训练时, 由于并不对状态和迁移进行删除或修改, 因此, 所产生的自动机将继续接收字符串 $w_1, \dots, w_n$ 。在 w_{n+1} 上进行训练将修改自动机以保证其接收 w_{n+1} 。证明与前面相同。

证毕。

自动机不但能接收训练集中的所有运行实例, 同时, 它还

具有一定的概括性。因此,也就没有必要在所有可能的程序运行上进行训练。例如,如果代码包含一个循环,仅需要训练几个实例甚至一个就可以产生一个好的循环,表示接收所有的运行实例。

4.2 算法检测性能分析

算法检测性能分析首先需要考虑的是对监视系统主机的影响,包括 CPU 使用和系统所需要的内存。因为要对程序行为模型化,需要足够的状态表示程序。因此,下面分析程序模型所需要维护的最大状态数、测试时间、训练时间、状态维护数,同时分析能检测到的入侵类型。

(1) 自动机最大状态数。有多种方法描述程序大小,常用的测度是自动机状态数和迁移数。对 IDS 来说,就是表示自动机的内存需要。

给定阈值 G 的系统调用串,假设在状态 s_i , 第一个系统调用至多创建一个迁移和状态(s_j), 其他系统调用则为了保证也满足当前自动机,需要创建足够的迁移和状态。每次碰到需要判断时(状态 s_k), 选择一条合适迁移,或者创建一个新迁移到状态 s_j 。如果程序运行所产生的系统调用为 s 种, G 为判断是否创建新迁移的阈值,则长为 G 的不同系统调用串(用 U 表示)数上限为 s^G , 即自动机状态数不会超过 s^G 即 U 。

对每个状态来说,因为迁移标记不可重复,因此每个状态不会超过 s 条迁移,这样,迁移总数就不会超过 Us , 上限为 s^{G+1} 。

自动机每个迁移的存储需要是每个迁移两个值(系统调用和目标状态)加表示每个状态的一个值(迁移数组的指针)。

(2) 测试时间。在测试时,每个系统调用从当前状态开始与每个迁移进行比较,直到找到一个与当前系统调用匹配的迁移,或者判断不存在匹配的迁移。这样,每个系统调用的测试时间与迁移数 l (取 $0 \sim s$ 的值)成比例,因此,最坏情况下性能为 $O(s)$ 。实际中每个状态的迁移远小于 s , 实验中一般每个状态平均 $2 \sim 4$ 次迁移。如果采用非线性搜索,则测试性能会得到改善。如果所选择的第一个迁移就满足当前系统调用,则最好的性能是 $O(1)$ 。

(3) 训练时间。在训练时,如果忽略前向缓冲区评价时间,则训练时间与测试时间相同,为 $O(l)$ (l 是每个状态的迁移数),这种情况就是当前状态存在一个标记为下一个系统调用的迁移。

如果当前系统调用与当前状态不匹配,需要评价现存状态,以决定是从中选择一个还是创建一个新状态。这涉及一个前向缓冲区评价深度(L , L 大于等于 G)。一般来说,评价每个系统调用,需要时间为 $O(l)$ 。这样最坏的情况是,整个前向缓冲区评价时间为 $O(LSl)$ (S 是该时刻自动机状态数),不会超过 $O(LU^2s)$, 也就是说最坏上限是 $O(Ls^{2G+1})$ 。实验中一般 L 等于 G 加一个常数(如 5)。系统调用个数为 N 的训练集,最坏训练总时间是 $O(NGs^{2G+1})$ 。

(4) 状态维护数。要反映在不同时间以及程序的不同运行时程序的不同行为,需要维护一定数量的状态。程序模型所需要维护的状态等于自动机状态数,即不超过 s^G 。

(5) 入侵检测类型。该方法主要用于检测程序滥用,主要检测引起程序行为与通常情况不同的入侵,独立于滥用目的。

很显然,该方法能检测到缓冲区溢出入侵和木马入侵。在

有些条件下能检测到竞争条件入侵。该方法也能检测到提升权限入侵引起的认证错误。

该方法不能检测那些没有在审计迹中留下蛛丝马迹的入侵,如外部误用、硬件误用、拦截数据,也不能检测到冒充合法用户的入侵,但如果冒充合法用户时或内部用户试图提升其特权,则会被检测到。

5 总结

本文提出的算法具有如下优点: 类似于其他异常检测方法,该方法能发现未知攻击。即使攻击利用未知漏洞,该方法也能检测到。方法构造上和采取的表示方式上使得验证比较快,对监视系统性能影响小。一旦某个系统调用偏离其期望行为,能够实时检测到,从而可以对入侵进行自动实时响应。

检测率高。训练速度快,需要的内存少。无需对程序的所有运行进行训练。无需手工指定程序期望行为。算法监视程序行为不是用户行为,保护了用户隐私。因为对正常的用户隐私是需要保护的,只有确定用户可疑时,才应该了解可疑用户的行为。算法本质上自增,可以在线学习程序行为。

误报率低,尤其在自增学习之后。该学习算法对训练数据的要求没有其他算法严格,训练数据中可以出现反例。

尽管算法具有种种优点,但它也存在如下局限性: 该方法仅针对特定的入侵类型,即涉及权限程序的误用,它不能保证检测所有攻击。因此,需要与其他检测方法结合使用。程序执行之初就需要跟踪系统调用: 一旦进程开始运行时就偏离其期望行为,则在以后不能发现这些异常行为。软件升级之后,需要对其重新训练。该方法只能检测到入侵的起点。一旦入侵者逃避了检测,则其以后的入侵行为将很难检测到,除非他再次提升特权。当然,这种情况可以通过与用户异常检测程序结合起来检测入侵。因为采用贪婪学习算法,不可避免有其不足。如果有限状态机的状态或连接添加不对,不可能删除该状态或连接,或把它改动到正确的位置。该方法不能检测到系统调用的其他参数,而这些参数可能对检测入侵起着重要作用^[5]。因此,考虑其他参数对入侵检测的影响将是下一步的工作。

参考文献:

- [1] 卿斯汉,蒋建春,马恒太,等.入侵检测技术研究综述[J].通信学报(信息安全专辑),2004,25(7):19-29.
- [2] Fink G, Levitt K. Property-based Testing of Privileged Programs[C]. Proceedings of the 10th Annual Computer Security Applications Conference, 1994.154-163.
- [3] 刘雪飞,马恒太,张秉权,等.基于系统调用的异常入侵检测研究[J].计算机工程与应用,2004,40(17):40-43.
- [4] Hopcroft J E, J D Ullman. Introduction to Automata Theory, Languages, and Computation[M]. Addison-Wesley, 1979.30-78.
- [5] 苏璞睿.基于特权进程行为的入侵检测方法研究[D].北京:中国科学院软件所,2005.15-30.

作者简介:

刘雪飞(1975-),女,湖南人,博士,主要研究方向为信息安全、数据挖掘;王申强(1975-),男,河北人,中级工程师,硕士,主要研究方向为电力系统网络安全;吴伯桥(1979-),男,湖南人,助教,学士,主要研究方向为信息安全;马恒太(1970-),男,山东人,副研究员,博士,主要研究方向为大型网络安全、并行计算。