一种基于并发命题投影时序逻辑模型检测的入侵检测方法

陈建辉[1] 王文义[2] 朱维军[3]

(郑州航空工业管理学院计算机科学与应用系 郑州 450015)[1]

(中原工学院并行处理技术研究所 郑州 450007)[2] (郑州大学信息工程学院 郑州 450052)[3]

摘 要 基于投影时序逻辑模型检测的入侵检测方法具有描述网络入侵者分段攻击的能力,然而对并发攻击仍无能为力,因为该逻辑无法直接描述并发。针对此问题,在该逻辑的基础上定义了一种新的并发算子,并给出基于并发投影时序逻辑模型检测的入侵检测方法。对复杂攻击实例的检测表明,新方法可有效提高对并发攻击的检测能力。

关键词 入侵检测,误用检测,模型检测,并发命题投影时序逻辑

中图法分类号 TP393,TP301 **文献标识码** A

Intrusion Detection Method Based on Model Checking of Concurrent Propositional Projection Temporal Logic

CHEN Jian-hui[1] WANG Wen-yi[2] ZHU Wei-jun[3]

(Department of Computer Science and Application, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450015, China)[1]

(Institute of Parallel Processing Technology, Zhongyuan Institute of Technology, Zhengzhou 450007, China)[2]

(School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China)[3]

Abstract The intrusion detection method based on model checking of projection temporal logic can describe piecewise network attacks but concurrent attacks. We defined a novel concurrent operation and obtained a new intrusion detection method based on model checking of concurrent propositional projection temporal logic. By example, we show that our method can detect concurrent attacks efficiently.

Keywords Intrusion detection, Misuse detection, Model checking, Concurrent propositional projection temporal logic

1 引言

入侵检测技术是一种保护自己免受入侵者攻击的网络安全技术,在不影响网络性能的情况下对网络进行监测,提供对内外部攻击的实时保护。它可分为两类:异常检测和误用检测。前者可以发现未知模式的攻击行为,但误报、漏报的情况有待改善;后者则只能发现被预先定义的攻击特征相匹配的事件或事件集合,但产生的误报较少,准确率较高。传统误用检测采用的模式匹配方法,很难在维持相对小规模存储攻击模式库的条件下准确刻画千变万化的攻击模式。因此,提出基于模型检测的误用检测方法[1,2]。

模型检测是一种形式化的技术,在电路设计、网络协议等领域的验证中已取得了许多重要应用。其基本原理是:自动机描述系统模型,时序逻辑描述有待验证的性质,算法自动检测模型是否满足性质。

在基于模型检测的误用检测方法中,时序逻辑描述攻击模式,自动机描述审计记录,模型检测算法检测记录中的事件集是否与攻击模式相符。由于当前的模型检测工具已可以有效检测高达10^{120}个系统状态,因此与传统的模式匹配方法相比,模型检测方法用于大规模网络误用检测特别有效[1]。然而时序逻辑描述攻击模式的能力尚不够强大,对此文献[9]提出了一种基于命题投影时序逻辑(Propositional Projection Temporal Logic, PPTL)模型检测的误用检测方法,以实现对分段式攻击的检测。然而,PPTL由于缺乏并发描述能力,因而无法描述入侵者常用的并发攻击。我们对此问题进行了研究。

2 并发命题投影时序逻辑

2.1 语法

定义1 并发命题投影时序逻辑(Concurrent Propositional Projection Temporal Logic, CPPTL)基本公式:$\varphi ::= Ap \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid (\varphi_1, \cdots, \varphi_n)\ prj\ \varphi_0 \mid \varphi_1 \amalg \varphi_2$

2.2 语义

定义2 在Kripke结构上定义状态$s: Ap \to B$,其中Ap为原子命题,$B=\{\text{true}, \text{false}\}$。

定义3 区间σ为状态序列$\langle s_i, \cdots, s_j \rangle$,其中$i \in N, i \in \omega$,$j \in N_\omega = N \cup \{\omega\}$,且$|\sigma| = j - i$,区间可为有穷也可为无穷。

定义4 解释是一个四元组$I=(\sigma, i, k, j)$,其中$i \leqslant k \leqslant j$,定义$(\sigma, i, k, j) \models \varphi$表示公式$\varphi$在区间$\langle s_i, \cdots, s_j \rangle$上被满足,当前状态为$k$。

定义5 设$\sigma = \langle s_0, s_1, \cdots, s_{|\sigma|} \rangle$为一区间,$r_1, \cdots, r_h \in N^+ \cup \{0\}$,$0 \leqslant r_1 \leqslant \cdots \leqslant r_h \leqslant |\sigma|$,区间$\sigma$到$r_1, \cdots, r_h$的投影区间定

到稿日期:2009-12-11 返修日期:2010-02-21 本文受国家(863)高技术研究发展计划(2007AA010408),河南省重大科技攻关计划(092101210104)资助。

陈建辉(1978—),男,硕士生,讲师,主要研究方向为网络安全和模型检测;王文义 教授,硕士生导师;朱维军 博士,讲师。

义为：$\sigma\downarrow(r_1,\cdots,r_h)=\langle s_{t_1},\cdots,s_{t_l}\rangle$，其中，$t_1,\cdots,t_l$ 从 $r_1,\cdots,r_h$ 中删除重复数字得到。

定义 6 满足关系$\models$定义为

I-Ap：$I\models p$ 当且仅当 $s_k[p]=\text{true}$，其中 $p\in Ap$

I-not：$I\models\neg\varphi$ 当且仅当 $I\not\models\varphi$

I-or：$I\models\varphi_1\vee\varphi_2$ 当且仅当 $I\models\varphi_1$ 或 $I\models\varphi_2$

I-and：$I\models\varphi_1\wedge\varphi_2$ 当且仅当 $I\models\varphi_1$ 且 $I\models\varphi_2$

I-next：$I\models\bigcirc\varphi$ 当且仅当 $k<j$ 且 $(\sigma,i,k+1,j)\models\varphi$

I-prj：$I\models(\varphi_1,\cdots,\varphi_n)\ prj\varphi_0$ 当且仅当存在整数 $k=r_0\leqslant r_1\leqslant\cdots\leqslant r_n\leqslant j$，使得 $(\sigma,r_0,r_0,r_1)\models\varphi_1$，$\forall 1<l\leqslant n.\ (\sigma,r_{l-1},r_{l-1},r_l)\models\varphi_l$，且 $(\sigma',k,k,|\sigma'|)\models\varphi_0$ 对以下二者其一成立：(1) $r_n<j$ 且 $\sigma'=\sigma\downarrow(r_0,\cdots,r_n)\cdot\sigma_{(r_n+1,\cdots,j)}$，或(2) $r_n=j$ 且 $\exists 0\leqslant h\leqslant n\cdot\sigma'=\sigma\downarrow(r_0,\cdots,r_h)$

I-parr：$I\models\varphi_1\amalg\varphi_2$ 当且仅当：

$I\models\varphi_1$ 且 $I\models(\varphi_2,\text{true})\ prj(\text{true}\wedge\bigcirc\text{true})$

或 $I\models\varphi_2$ 且 $I\models(\varphi_1,\text{true})\ prj(\text{true}\wedge\bigcirc\text{true})$

定理 1(两逻辑等价性定理) $L_{CPPTL}=L_{PPTL}$

证明：比较 PPTL 定义[3]（本文定义 1 到定义 6）和 CPPTL 定义，后者多了一个新算子“$\amalg$”。如果可证“$\amalg$”由其它算子表示，则可得出两逻辑等价的结论。由上面定义知，

$I\models\varphi_1\amalg\varphi_2\Leftrightarrow I\models\varphi_1\wedge((\varphi_2,\text{true})\ prj(\text{true}\wedge\bigcirc\text{true}))\vee\varphi_2\wedge((\varphi_1,\text{true})\ prj(\text{true}\wedge\bigcirc\text{true}))$，因此 $\varphi_1\amalg\varphi_2=\varphi_1\wedge((\varphi_2,\text{true})\ prj\ (\text{true}\wedge\bigcirc\text{true}))\vee\varphi_2\wedge((\varphi_1,\text{true})\ prj\ (\text{true}\wedge\bigcirc\text{true}))$，注意到等式右边所有算子均在定义 1 到定义 6 中被定义，因此 $L_{CPPTL}=L_{PPTL}$ 成立。

定理 1 表明，CPPTL 与 PPTL 具有相同表达能力。如果使用它们描述网络入侵者的攻击模式，将具有相同描述能力。然而后者由于并未定义并发算子，因而缺少直接描述并发攻击的方法，而前者则可以便捷地做到这一点。

3 基于并发命题投影时序逻辑模型检测的误用检测方法

如图 1 所示，基于并发命题投影时序逻辑模型检测的误用检测方法步骤描述如下：

(1) 使用 CPPTL 公式建立攻击模式的模型：与 PPTL 公式相比，并发攻击可由 CPPTL 公式的“$\amalg$”算子描述。本文第 4 节中给出这样的描述实例。

(2) 建立审计记录的自动机模型[1]。

(3) 把 CPPTL 公式和步骤(2)中自动机作为两个输入，模型检测算法[3]验证自动机是否满足 CPPTL 公式。

(4) 如果模型检测结论是自动机满足公式，则发现审计记录与攻击模式匹配，于是报警；否则即为未发现攻击。

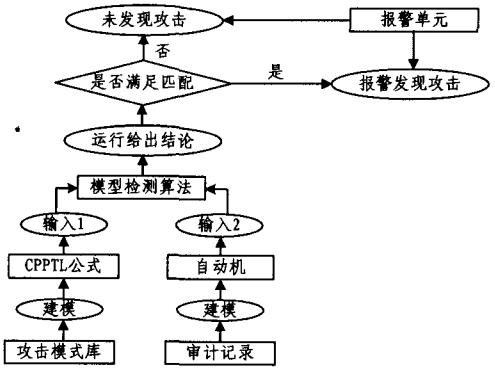

图 1 基于并发命题投影时序逻辑模型检测的入侵检测方法

4 实例研究

例 1 某次 *Telnet* 入侵在审计记录中被观测到的步骤如下：

阶段一：获取口令，我们可以标记该动作为符号 P（即由 CPPTL 原子公式 P 表示）。

阶段二：*Telnet* 服务被启用，由 CPPTL 原子公式 Q 表示。

阶段三：入侵者使用工具 *NTLM. EXE* 去除 *NTLM* 验证，该阶段有两个步骤：*admin* 下复制 *NTLM. EXE*，由 CPPTL 原子公式 R_1 表示；*NTLM. EXE* 被执行，由 CPPTL 原子公式 R_2 表示。

阶段四：入侵者关闭防火墙。该阶段有 3 个步骤：入侵者进入 *C:\windows* 目录找到 *aproman* 程序，由 CPPTL 原子公式 S_1 表示；通过 *aproman*$-a[PID]$命令查看所有进程，找到防火墙进程的 *PID*，由 CPPTL 原子公式 S_2 表示；使用 *aproman*$-t[PID]$关闭防火墙，由 CPPTL 原子公式 S_3 表示。

阶段五：入侵者为事后仍可登录，留下后门。该阶段有两个步骤：进入 *instsrv* 所在目录，由 CPPTL 原子公式 T_1 表示；执行命令 *instsrv. exeSYSHE*$-$*ALTH C:\windows\SYSTEM32\tlntsvr. exe*，以建立 *SYSHEAHTH* 服务，即为入侵后门，由 CPPTL 原子公式 T_2 表示。

其中后两阶段有可能并发，*Telnet* 入侵公式为：

$$(P,Q,(R_1;R_2),((S_1;S_2;S_3)\amalg(T_1;T_2)))$$
$$prj(\text{true}\wedge\bigcirc\text{true}\wedge\bigcirc\bigcirc\text{true}\wedge\bigcirc\bigcirc\bigcirc\text{true}) \quad (1)$$

针对本例，分别使用 CPPTL 方法与现有方法检测入侵者行为，结果如表 1 所列。

表 1 基于不同时序逻辑模型检测的入侵检测方法比较

入侵行为\检测结果（能否发现攻击）	CPPTL 方法	PPTL 方法[9]	LTL 方法[1]
先关防火墙后留后门	Yes	Yes	No
后关防火墙先留后门	Yes	Yes	No
边关防火墙边留后门	Yes	No	No
两随机动作存在重合时间段	Yes	No	No
两随机动作不存在重合时间段	Yes	Yes	No

结束语 我们提出了一种新的时序逻辑，在此基础上给出基于本逻辑模型检测的入侵检测方法。现有的基于模型检测的入侵检测方法均有不足：进程代数方法[4]由于抽象程度过高，对网络安全开发人员来讲，不能直观地建立攻击库模型；LTL 方法[1]没有描述入侵者并发攻击的能力；PPTL 方法[9]的逻辑虽具该能力，但缺乏描述攻击的可操作方法。CPPTL 方法则克服了以上问题——新方法由于仍为基于自动机模型检测的方法，因此具有建模直观、便捷的特点；新逻辑中的新算子“$\amalg$”可直接描述并发攻击，而这样的攻击正是入侵者惯用的手段。

参 考 文 献

[1] Roger M, Goubault-Larrecq J. Log Auditing through Model-Checking[C] // Proceedings of the 14th IEEE Workshop on Computer Security Foundations. Washington, DC, USA: IEEE Computer Society, 2001: 220-234

(下转第 137 页)

按下 Generate Code 按钮后，GridsimHelper 将拼接文本 Model. txt，Resource. txt，User. txt 和 customizeAlg. txt，形成完整的 Gridsim 仿真代码 Experiment. java。

4 实验与分析

为了证明系统的正确性，我们首先利用 GridsimHelper 创建了一个含 5 个资源的异构网格，令其使用 min-min 调度算法运行 45 个独立的任务。图 4 是被仿真的网格、等待调度的任务以及仿真实验的结果片断。实验表明，GridsimHelper 自动生成仿真代码完全可以在 GridSim 平台上运行。

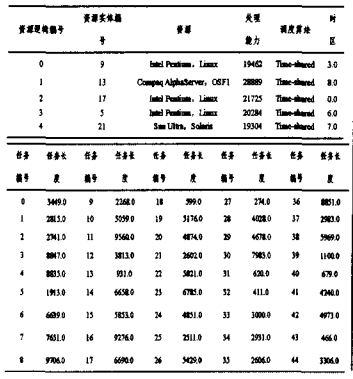

图 4 GridsimHelper 产生的代码在 Gridsim 上的运行结果

下一组实验针对 GridsimHelper 为 max-min，min-min 算法自动生成仿真代码。算法 max-min 和 min-min 为静态调度算法，故每次实验只有 1 个用户。3 次实验分别用来调度 50，100 和 150 个任务。考察指标有二：调度长度(makespan)和负载平衡度。表 1 和表 2 为仿真实验的结果，从中得出结论：min-min 算法具有较小的 makespan，但资源负载平衡度较差(不同的资源，完成系统指派给它的所有任务所消耗的时间相差很大)。与此相反，虽然 max-min 在指标 makespan 上略逊于前者，但负载平衡度明显较优。综上所述，算法 min-min 和 max-min 各有千秋。此结论与文献[7]所述相符，系统提供的调度算法正确无误。

结束语 为了减少 GridSim 仿真代码的编写难度，使不熟悉 Gridsim 环境和 Java 编程的调度算法设计者也能够轻松地编写出可以在 GridSim 平台上运行的仿真实验代码，本文具体地提出并实现了一个具有图形用户接口的 GridSim 代码生成器 GridsimHelper。它可以基于实验者输入的网格资源特性和需要运行的任务特性构造实验环境，并允许用户随意选择和编写调度算法。实验结果表明，GridsimHelper 的代码生成过程以及系统所提供的两种经典调度算法正确，确实达到了减轻调度算法设计者工作强度的目的。

表 1 GridsimHelper 为算法 min-min 生成的仿真代码的性能

任务数	任务长度(MI)		单个资源完成所有任务的时刻		资源的负载不平衡程度
	最小	最大	最早	最晚	(s)
50	274.0	9706.0	1586.54	1795.67	209.13
100	274.0	9706.0	3149.48	3366.18	216.70
150	274.0	9706.0	4810.63	4989.62	178.99

表 2 GridsimHelper 为算法 max-min 生成的仿真代码的性能

任务数	任务长度(MI)		单个资源完成所有任务的时刻		资源的负载不平衡程度
	最小	最大	最早	最晚	(s)
50	274.0	9706.0	1780.40	1826.25	45.85
100	274.0	9706.0	3345.73	3407.35	61.62
150	274.0	9706.0	4954.96	4996.57	41.61

参考文献

[1] Buyya R, Murshed M. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing [J]. The Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002, 14 (13-15): 1175-1220

[2] 卢鹏，金海，谢夏，等. 关于网格模拟器的研究[J]. 高性能计算技术，2005，173(2)：5-9

[3] Foster I, Kesselman C. The Grid: Blueprint for a new computing infrastructure[M]. 1th ed. San Francisco: Morgan Kaufmann Publishers, 1998: 1-20

[4] 刘宴兵，杨茜慧，王文斌. 基于 GridSim ToolKits 的网格仿真环境设计与实现[J]. 计算机科学，2008，35(6)：83-85

[5] 李炯，卢显良，董仕. 基于 GridSim 模拟器的网格资源调度算法研究[J]. 计算机科学，2008，35(8)：95-97

[6] Sulistio A, Chee Shin Yeo, Buyya R. Visual Modeler for Grid Modelling and Simulation (GridSim) Toolkit[C]// Proc. of the 3rd International Conference on Computational Science (ICCS 2003, LNCS Series). Melbourne: Springer Verlag Publications, 2003: 1123-1132

[7] Braun T D, Siegel H J, Beck N. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(6): 810-837

(上接第 117 页)

[2] 张燕，傅建明，孙晓梅. 一种基于模型检查的入侵检测方法[J]. 武汉大学学报：理学版，2005，51(3)：319-322

[3] Duan Z, Tian C, Zhang L. A decision procedure for propositional projection temporal logic with infinite models[J]. Acta Informatica, 2008, 45(1): 43-78

[4] Rohrmair G T, Lowe G. Using data-independence in the analysis of intrusion detection systems[J]. Theoretical Computer Science, 2005, 340(1): 82-101

[5] Sheyner O, Haines J, Jha S, et al. Automated generation and analysis of attack graphs[C]// Proceedings of the 2002 IEEE Symposium on Security and Privacy. 2004: 273-284

[6] Nan Z, Mark R, Dimitar P G. Evaluating Access Control Policies Through Model Checking[C]// Lecture Notes in Computer Science, Information Security. Springer Berlin, 2005, 3650: 446-460

[7] Sebastian S, Michael V, Hartmut K. Identifying Modeling Errors in Signatures by Model Checking[C]// Lecture Notes in Computer Science, Model Checking Software. Springer Berlin, 2009, 5578: 205-222

[8] Olivain J, Goubault-Larrecq J. The Orchids Intrusion Detection Tool[C]// Proceedings of the 17th International Conference on Computer Aided Verification. Lecture Notes in Computer Science. Springer, Edinburgh, Scotland, UK, 2005, 3576: 286-290

[9] 朱维军，王逸冉，周清雷. 一种基于投影时序逻辑模型检测的入侵检测方法[J]. 网络安全技术与应用，2010，3：25-27