基于场景和PN机的入侵检测研究

张 巍 罗辉云 滕少华 刘冬宁 梁 路
(广东工业大学计算机学院 广州 510006)

摘 要 攻击者通过从一个攻击序列衍生出大量变种攻击序列来逃避基于规则及其它误用检测技术的检测。基于此,针对可序列化的入侵,从攻击机理入手,提取攻击的关键操作序列,构造入侵行为表达式,再对攻击序列进行拓扑排序和同构变换,以扩展形成一个入侵场景或一类入侵。进而提出了面向场景和检测一类入侵行为的方法,通过构建基于场景和检测一类入侵行为的PN(Petri Net)机来实现检测已知攻击及其未知变种攻击的目标。未知变种攻击也是一些新的攻击形态,因而从这种意义上说,该方法能检测到新的攻击行为。

关键词 入侵检测,场景,攻击序列,同构变换,拓扑排序,入侵行为表达式,PN机

中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.5.038

Intrusion Detection Based on Scenario and PN Machine

ZHANG Wei LUO Hui-yun TENG Shao-hua LIU Dong-ning LIANG Lu
(School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China)

Abstract To evade detection of rule-based or other misuse detection methods, the attacker can create a large number of variant attack sequences from one attack sequence. Therefore, aiming at the serializable intrusion, we started to study the attack mechanism, extracted key operation sequence of the attacks, constructed intrusion behavior expressions, sorted topologically attack sequence, and did isomorphic transformation for attack operations. Then one attack can be expanded to one intrusion scenario or one class of attacks. A new intrusion detection method was proposed in the paper, which is called the scenario-oriented intrusion detection. A PN machine for scenario was designed and implemented. The PN machine based on scenario can detect one class of attacks. Then, the goal of detecting the known attack and its unknown variant attacks will be achieved. So, some new derived attacks can be detected by the method in the paper.

Keywords Intrusion detection, Scenario, Attack sequence, Homogeneous transformation, Topological sorting, Intrusion behavior expression, PN machine

入侵检测可分为两类:基于行为的异常检测和基于知识/模式的误用检测[1,2]。前者的技术基础是将偏离正常用户的行为或模式认为是异常[3,4]。异常检测常采用度量集来检测,每一个度量与一个阈值相联系,它用来衡量用户特定方面的行为,阈值设置不当,会造成误报或漏报。因此,一个好的异常入侵检测系统必须能表示所有正常用户行为,并具有足够完备的度量集。但实际上往往难以构建这种完备的正常用户行为,且攻击者通过训练可以使异常行为与正常行为相似,因此,异常检测往往效果较差;误用检测技术运用已知攻击方法或入侵模式,通过判断这些攻击方法或入侵模式是否出现来检测入侵[3,4]。

由于发明一种攻击方法的难度远大于对已知攻击方法的变形改造,因此攻击者常常利用已知攻击产生变种攻击,以逃避检测[3,4]。误用检测系统的有效性依赖于训练数据的质量,根据截取到的网络数据包中是否具有某种已知攻击符号特征,或是检查某些特定端口是否非授权开放来判断是否发生了攻击。实际上,一个攻击行为可看做一个过程,通过一系列操作来实现,可转化为攻击序列,这表明检测一个攻击行为可转化为对攻击序列的检测[4]。由于基于模式匹配的检测规则与攻击序列(提炼的审计记录之间)是一一映射关系,这直接导致微小的攻击序列变化将使基于规则的入侵检测系统失效。

综上,本文的出发点是:仔细分析一个已知攻击的攻击过程,探究其攻击机理,研究攻击过程中对象的状态及其变迁的

到稿日期:2014-06-07 返修日期:2014-08-27 本文受国家自然科学基金(61402118,61272067,61104156,61370229),教育部重点实验室基金(110411),广东省科技计划项目(2012B091000173),广东省教育厅项目(粤教高函〔2013〕113号),广州市科技计划项目(2012J5100054,2013J4500028),韶关市科技计划项目(2010CXY/C05)资助。

张 巍(1964-),女,硕士,副教授,CCF会员,主要研究方向为Petri网及其应用、协同计算、数据挖掘与网络安全,E-mail:weizhang@gdut.edu.cn;罗辉云(1990-),男,硕士生,主要研究方向为Petri网及其应用、数据挖掘、网络安全,E-mail:huiyunl@126.com;滕少华(1962-),男,博士,教授,CCF会员,主要研究方向为大数据、数据挖掘、Petri网、协同计算、网络安全,E-mail:shteng@gdut.edu.cn;刘冬宁(1979-),男,博士,副教授,CCF会员,主要研究方向为人工智能逻辑、数据库与协同计算,E-mail:liudn@gdut.edu.cn;梁 路(1980-),女,博士,副教授,主要研究方向为数据库与协同计算,E-mail:lianglu@gdut.edu.cn。

内在联系，把已知攻击及其所有派生的新攻击组织进一个入侵场景，进而构造 PN 机识别该入侵场景，检测这些入侵行为。

1 相关工作

使用 petri 网来描述一个攻击，可以将该攻击的所有变种攻击都用 petri 网来描述，这要求对于给定的一个新事件，按照 petri 网的变迁实施规则来确定它的输出，若在最终位置上得到一个 TOKEN，就说明发生了一次攻击[5]。Kumar 等人最早研制了一个基于 CPN 的网络入侵检测系统原型 IDIOT[6]，在 IDIOT 的实现上，使用一种描述语言来表示 CPN，并通过解释器将 petri 网转换成 C++类，但其模式匹配中必须采用试探性搜索算法。M. Slagell 等人[7,8]在 MAIDS 中提出了状态 IDS 的一种分布式 CPN 的概念和实现，完成了从软件故障树到 CPN 的转换机制，利用成熟的面向对象设计技术完成了一种分布式的 DCPN(Distributed CPN)。但此模型效率比较低的原因对于一个变迁来说，需要多次搜索来确定哪些 TOKEN 用于合一操作。因此出现所有的 k 个 TOKEN 进行合一的时间复杂度的下限为 $O^{[(k-1)/t]+1}$（t 为变迁个数）。文献[9]通过构建攻击的 CPN 树，扩展了 CPN 的构造，进而利用攻击的 CPN 树来分析入侵的动态和静态特性，体现了 CPN 在模拟网络入侵的灵活性。为了显示攻击场景中多步流程为导向的持续性威胁的攻击特性，文献[10]提出了基于扩展 petri 网的高级持续性威胁分析模型，运用层次分析法将攻击场景与 petri 网进行关联，最后遍历扩展 petri 网生成正则表达式，实现了攻击场景、攻击过程、状态空间的结合。文献[11]讨论了现有的攻击场景重建方法的局限性，并通过使用语义分析和入侵本体论提出了新的混合方法，以重建已知和未知的攻击场景。

上述将 petri 网应用于入侵检测的方法，在一定程度上能有效地检测某些入侵行为，但对于攻击者为了逃避检测，施行某些变换甚至细微的变化使 petri 网的状态数激剧增长而引发空间组合爆炸的问题，未能有效地处理。为此，本文仔细分析与研究了攻击行为的机理与特征，基于前人的入侵检测工作，提出了基于场景和 PN 机的入侵检测方法，从攻击序列出发，提取一个攻击的关键操作序列，将其扩展到关键操作类，进而形成攻击场景；然后依场景构建 PN 机，以进行检测，从而使 petri 网在模式匹配时，不必对所有变迁进行 TOKEN 复制，有效地处理了 TOKEN 的累积问题，极大地降低了时间复杂度和空间复杂度。

2 入侵场景建模

2.1 入侵实例

基于场景的入侵适用于可序列化的攻击过程，即可通过执行一系列操作来实现攻击。在文献[4,12]中列出了许多攻击实例，遍及缓冲区溢出攻击、口令攻击、拒绝服务攻击、会话劫持、权限提升攻击，以及 WSN 的病毒传播攻击[13]，并给出许多攻击的变种。下面给出具体的攻击实例。表 1 列出了 Unix 系统的一个漏洞，执行操作序列 $S_1S_2S_3S_4S_5$，将会获取 root 用户访问权[4]。将序列改为 $S_3S_1S_2S_4S_5$ 或 $S_1S_3S_2S_4S_5$ 等具有相同的攻击效果，履行了等效的权限提升攻击。入侵实例如表 1 所列。

表 1 获取 root 用户 shell 文件访问权的入侵序列 1

step	command	命令说明
S_1	cp /bin/csh/usr/spool/mail/root	假定没有邮件
S_2	chmod 4755 /usr/spool/mail/root	设置 root 文件属性
S_3	touch x	创建一个空文件 x
S_4	mail root < x	发送空文件 x 到 root
S_5	/usr/spool/mail/root	运行 root 文件

表 2 列出了 FTP Bounce to RSH 攻击的操作/命令序列，利用 FTP 协议的安全缺陷漏洞，将事先上传到 FTP 相关目录下的恶意脚本上载到目标攻击主机上，由信任关系的传递而导致攻击者权限的提升。同样，改变部分操作序列顺序，即 $T_1T_2T_3T_4T_5T_6$、$T_1T_3T_2T_4T_5T_6$ 和 $T_1T_3T_4T_2T_5T_6$ 等，也可实现等效的攻击效果。入侵序列 2 如表 2 所列。

表 2 获取 root 用户 shell 文件访问权的入侵序列 2

step	command	操作说明
T_1	PORT_SCAN	端口扫描
T_2	upload attack. sh	上传攻击脚本
T_3	type i	指定数据类型，i 为二进制类型
T_4	port IP, port	指定服务器建立数据连接时应联系的端口
T_5	RETR attack. sh	从 ftp 服务器下载 shell 攻击脚本
T_6	attack. sh	shell 攻击脚本被执行

2.2 入侵场景框架

场景是以人物为中心的环境描写，一般由人物、事件和环境组成。本文的场景指在已知一个入侵行为的基础上，通过对入侵序列同构变换及拓扑排序，扩充到包含一类入侵行为所形成的环境。入侵场景生成包括 3 部分：关键攻击序列提取、场景生成、场景库。三者的关系如图 1 所示。

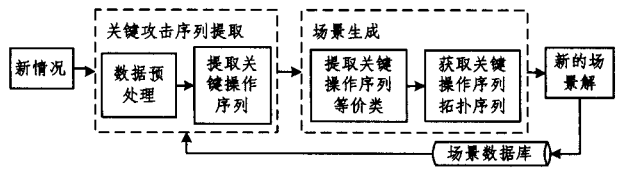

图 1 构建入侵场景

关键攻击序列提取由数据预处理和提取关键操作序列两部分组成。场景生成由构造关键操作序列的等价类和获取关键操作的拓扑序列组成。

2.3 攻击序列提取

攻击序列提取包括数据预处理和提取关键操作序列两部分。数据预处理包括：剔除无用数据、对数据进行规范化处理、统一数据格式、操作命名与分类、数据整合与变换等。然后分析攻击情况，提取攻击特征，进而形成关键操作序列。

由图 1 及上段可知，获取关键路径至关重要，它将过滤掉攻击序列中所有与入侵行为无关的操作，剩下的操作对入侵而言，是不可或缺的；否则，入侵过程将无法完成。下面给出了确定关键操作序列的策略，它由敏感操作或命令集、引发系统对象发生变化的操作或命令、问答式判断 3 部分组成。

K. Ilgun[14]给出了 STAT 的基本安全模块事件分类表，由此，我们得到了敏感操作或命令集分类表、系统文件及目录对象访问权限分类表，如表 3 和表 4 所列。

表 3　敏感操作或命令集分类

operation	事件类型	命令说明
read	open_r,open_rc,open_rtc,…	读文件
write	truncate,creat,open_wt,open_wc,…	写文件
delete	rmdir,unlink,userdel,…	删除
execute	exec,execve,…	执行文件
modify	chown,fchown,…	修改文件权限
link	link	文件链接
regedit	OpenRemoteBaseKey,SetValue, DeleteKey,…	获取/修改/删除注册表键信息

表 4　系统文件及目录对象访问权限分类

文件集合	特征描述
Fileset#1	限制读访问权限文件集
Fileset#2	限制写访问权限文件集
Fileset#3	授权读 Fileset#1 文件
Fileset#4	授权写 Fileset#2 文件
Fileset#5	不可写入的系统可执行文件
Fileset#6	有权创建系统文件或创建一个系统目录
WSD	授权写访问的系统目录
NWSD	不可写入系统目录
HARDLINK	系统硬链接文件

在表 3 与表 4 中，要特别关注引发系统对象发生变化的操作或命令：当创建一个对象(文件、目录、用户)时，应作为候选关键操作，需做进一步分析；当改变对象属主或权限的操作时，比如只读文件属性改为可写、可执行，其应归入候选关键操作，待进一步验证；若违反表 4 列出的访问规定，则被视为发生非法操作；否则作为候选关键活动，根据操作者、操作对象、施行的活动及上下文决定取舍。通过问答式判断等措施来确认攻击序列中的关键操作，如图 2 所示。

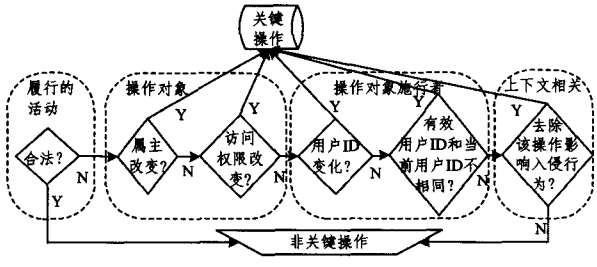

图 2　关键操作确认图

2.4　建立等价类与拓扑排序

存在一些攻击，入侵者执行相同或相似的操作，交换攻击序列的部分命令顺序，不改变攻击效果。由表 1 可知，由于 S_1 和 S_2 是顺序关系，S_1S_2 整体和 S_3 是偏序关系，S_1S_2 和 S_4、S_3 和 S_4、以及 S_4 和 S_5 之间都是顺序关系。故入侵序列 1 的所有拓扑排序有 $S_1S_2S_3S_4S_5$、$S_3S_1S_2S_4S_5$ 和 $S_1S_3S_2S_4S_5$。表 2 中，T_1 和 T_2 是顺序关系，T_2 与 T_3T_4 整体是偏序关系，剩余操作是顺序关系，因此表 2 的所有拓扑排序有 $T_1T_2T_3T_4T_5T_6$、$T_1T_3T_2T_4T_5T_6$ 和 $T_1T_3T_4T_2T_5T_6$。为获取及对付变种攻击，我们创建了操作、命令的同构类，本文的同构类(等价类)指的是属于同一个同构类中的元素，对系统具有相同执行效果，如表 5 所列。

表 5　部分操作、命令等价类

等价类	操作说明	命令
Duplicate	文件复制	cp,cat,more,less,zcat
FileCreation	创建一个新文件	touch,vi,jove,emacs,…
Print	打印一个文件	lpr,lp
Mail	发送邮件	elm,mail,pine,…
AccessControl	改变访问权限	chmod
PostScan	端口扫描	x-scan,x-way,portscan
Execution	运行一个文件	exec,sh,csh,…
……	……	……

因此攻击序列的同构变换和拓扑排序也是系统检测的关键环节。即对一个攻击序列，施行同构变换及拓扑排序后，可生成一系列具有相同效果的攻击序列。表 6 列出了表 1 的另外两种获取 root 用户 shell 文件访问权的入侵序列。

表 6　另两种获取 root 用户 shell 文件访问权的入侵序列

step	Command_1	Command_2
δ_1	jove x	cp /bin/csh/usr/spool/mail/root
δ_2	cat/bin/csh /usr/spool/mail/root	touch x
δ_3	chmod 4755 /usr/spool/ mail/root	chmod 4755 /usr/spool/ mail/root
δ_4	elm root < x	mail root < x
δ_5	exec /usr/spool/mail/root	/usr/spool/mail/root

下面利用正则式去描述表 1 和表 2 攻击序列形成的入侵场景，由于序列中的每个元素表示某类操作，该元素用同类中任一元素替代后有相同的运行结果，每个操作又可以重复执行多次，还有可能返回之前的操作再重复执行。因此表 1 的入侵场景 1 有如下入侵操作串：

$((S_1^+S_2^+S_3^+S_4^+)|(S_1^+S_3^+S_2^+S_4^+)|(S_3^+S_1^+S_2^+S_4^+))^+S_5$

其中，$S_i^+=\{S_i,S_iS_i,S_iS_iS_i,\cdots\}$表示任意个 S_i 串，| 表示析取。

表 2 的入侵场景 2 可以有如下入侵操作串：

$((T_1^+T_2^+T_3^+T_4^+T_5^+)|(T_1^+T_3^+T_2^+T_4^+T_5^+)|(T_1^+T_3^+T_4^+T_2^+T_5^+))^+T_6$

其中，$T_i^+=\{T_i,T_iT_i,T_iT_iT_i,\cdots\}$表示任意个 T_i 串，| 表示析取。

因此，对上述操作串施行检测，将能检测到一种攻击方法的所有变种攻击方法。

2.5　构造场景数据库

场景库存放一系列入侵场景，每个场景代表一类入侵活动，场景库的结构如表 7 所列。

表 7　场景库结构

字段	说明
ScenarioDescription	入侵场景的描述说明
ScenarioType	场景类型
Command	入侵攻击序列集合
EmergencyGrade	攻击序列危害程度
Count	攻击操作或命令总数
Target	目标对象

3　检测入侵的 PN 机建模

从上段，得到了入侵的关键攻击序列、序列的拓扑排序和等价类，从而由一个入侵形成一个攻击场景。下面将基于场景来构造入侵检测的 PN 机模型，检测入侵操作串。

3.1　操作行为表达式与 PN 机模型

定义 1　设 Σ 是有限操作集，Σ^* 是 Σ 上的操作串集合，$P(\Sigma^*)$ 是 Σ^* 的幂集，称作操作步集合，$P(\Sigma^*)^*$ 是 $P(\Sigma^*)$ 上的操作步串集合。则对于任何 $L\subseteq P(\Sigma^*)^*$，称 L 是 Σ 上的一个操作语言。

定义 2[15]　设 L_1,L_2 分别是 Σ 上的两个操作语言，定义如下语言算子：

(1)选择算子“/”，$L_1/L_2=\{\alpha_1/\alpha_2\mid\alpha_1\in L_1\vee\alpha_2\in L_2\}$；

(2)并发算子“‖”，$L_1\parallel L_2=\{\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix}\mid\alpha_1\in L_1\wedge\alpha_2\in L_2\}$；

(3)连接算子“。”，$L_1\circ L_2=\{\alpha_1\circ\alpha_2\mid\alpha_1\in L_1\wedge\alpha_2\in L_2\}$；

(4)闭包算子“$*$”，$L_1^* = \bigcup_{i=1}^{\infty} L_1^i$，其中 $L_1^0 = \{\emptyset\}$，$L_1^i = L_1 \circ L_1^{i-1}$，$i \geqslant 1$；

(5)前向算子“$\prec$”，$\Delta_a = \{x_1, x_2, \cdots, x_k\} \subseteq \Sigma$，$\Delta_a \prec a$，表示对于 $\forall x_i \in \Delta_a$，$x_i \prec a$，$a \in \Sigma$，即 Δ_a 中的元素先于 a 发生。

例：假设操作集合 $\Sigma = \{t_1, t_2, t_3, t_4, t_5\}$，$\gamma = (t_1(t_2 t_3 \parallel t_4(\Delta_{t_5} \mapsto t_5)))^*$，其中 $\Delta_{t_5} = \{t_2\}$，则 γ 是 Σ 上的一个操作行为表达式。

定义 3[16] 称 $PN = (P, T; F, M_0, C, P_C, F_C, H_C, \Sigma, h, G_f)$ 为一个着色 PN 机，它满足如下条件：

(1)$PN = (P, T; F, M_0)$ 是一个 petri 网；

(2)C 是颜色集合，C_{MS} 表示颜色集的多重集合，用于变迁 t 在不同发生色和同种发生色都可并发[17]；

(3)P_C 是库所内 TOKEN 到颜色集的映射：$P \to C_{MS}$；没有 TOKEN 时记为 $P_{C0}(p)$；

(4)F_C 是弧到颜色集的映射：$F \to C_{MS}$，用于表示变迁发生的条件；

(5)H_C 是事件集到颜色集的映射：$\Sigma \to C_{MS}$，用于 $H_C(t)$，$t \in T$ 表示变迁 t 所赋予事件的颜色集；

(6)Σ 是有限操作集；

(7)$h: T \to \Sigma$ 为标识函数；

(8)$G_f \subseteq R(M_0)$ 为终止标识集。

定义 4 设 $P0$ 和 Pf 分别表示初始状态和终止状态，t，t_1，t_2 表示变迁。识别基本算子的 PN 机定义如下。

(1)连接算子“$\circ$”

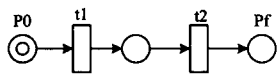

图 3 识别连接算子的 PN 机 $PN_{t_1 \circ t_2}$

(2)闭包算子“$*$”

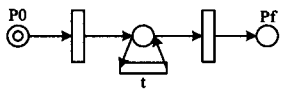

图 4 识别闭包算子的 PN 机 PN_{t*}

(3)选择算子“/”

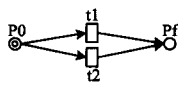

图 5 识别选择算子的 PN 机 PN_{t_1/t_2}

(4)并发算子“$\parallel$”

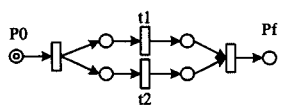

图 6 识别并发算子的 PN 机 $PN_{t_1 \parallel t_2}$

定理 1 识别一个操作行为表达式的 PN 机能够由基本算子复合生成。

证明：对于一个操作行为表达式 γ，若 γ 中算子个数为 0，则 γ 是 $\emptyset$，生成 PN 机模型如图 7 所示；若 γ 中算子个数为 1，则是由基本算子生成 PN 机模型，如图 3－图 6 所示；若 γ 中算子个数大于 1，则 γ 中的字符个数必将大于 2，则可按照定义 4 中的两个基本算子的 PN 机通过复合生成。例如本文中提到的例子：$\gamma = (t_1(t_2 t_3 \parallel t_4(\Delta_{t_5} \mapsto t_5)))^*$，其中 $\Delta_{t_5} = \{t_2\}$，生成的 PN 机如图 8 所示。

图 7 算子为 0 的 PN 机示意图

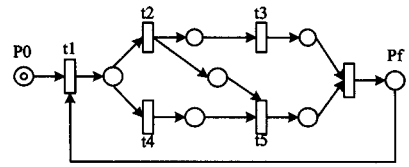

图 8 复合算子的 PN 机示意图

3.2 构建基于场景的 PN 机模型

对于任意一个可序列化的攻击，我们可以获得关键操作序列和操作等价类，经过拓扑排序，进而得到所有可能的攻击操作行为表达式。

定义 5[15] 设 $PN = (P, T; F, M_0, C, P_C, F_C, H_C, \Sigma, h)$ 为一个着色 PN 机，G_f 是一个终止标识符，$G_f \subseteq R(M_0)$，若 $L(\langle PN, G_f \rangle) = \{h(\alpha) | \alpha$ 是 T 上的操作步串，且 $M_0[\alpha > M_f, M_f \in G_f\}$，则称 $L(\langle PN, G_f \rangle)$ 为 PN 机的 PN 操作语言。

定理 2 对于任何操作行为表达式 γ，一定存在一个安全的 PN 机模型，使得 $L(\gamma) = L(PN)$。

证明：略，可参考文献[15]。

定理 1、2 可以说明任何一个操作行为表达式都可生成安全 PN 机模型，因此已知一个攻击入侵场景，即可创建识别一个入侵场景的所有关键路径的 PN 机模型。根据文献[15]的行为表达式转换成安全 PN 机算法，下面将本文提出的两个入侵场景构造出入侵检测 PN 机模型，在入侵场景 1 中，攻击操作行为表达式为：

$((\Delta_{s_2} \prec S_2^* S_3^* S_4^*)/(S_1^* S_3^* S_2^* S_4^*)/(S_3^* \Delta_{s_2} \prec S_2^* S_4^*))^* S_5$，$\Delta_{s_2} = \{S_1\}$

攻击操作逆波兰表达式为：

$\Delta_{s_2} S_2 S_3 S_4 \circ \prec \circ * \circ * \circ * \circ S_1 S_3 S_2 S_4 \circ / \circ * \circ * \circ * \circ * \circ S_3 \Delta_{s_2} S_2 S_4 \circ / \circ * \circ \prec \circ * \circ * \circ S_5 \circ *$

$\Delta_{s_2} = \{S_1\}$，其 PN 机模型如图 9 所示。

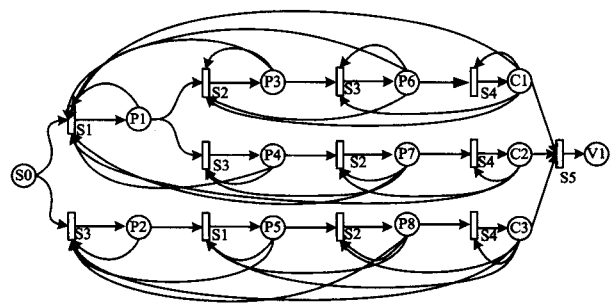

图 9 检测入侵场景 1 攻击的 PN 机模型

在入侵场景 2 中，攻击操作行为表达式为：

$((\Delta_{T_2} \prec T_2^* T_3^* T_4^* T_5^*)/(T_1^* T_3^* T_2^* T_4^* T_5^*)/(T_1^* T_3^* T_4^* T_2^* T_5^*))^* T_6$，$\Delta_{T_2} = \{T_1\}$

攻击操作逆波兰表达式为：

$\Delta_{T_2} T_2 T_3 T_4 T_5 \circ * \circ * \circ * \circ * \circ \prec \circ T_1 T_3 T_2 T_4 T_5 \circ / \circ * \circ * \circ * \circ * \circ T_1 T_3 T_4 T_2 T_5 \circ / \circ * \circ * \circ * \circ * \circ * \circ T_6 \circ *$

$\Delta_{T_2} = \{T_1\}$，其 PN 机模型如图 10 所示。

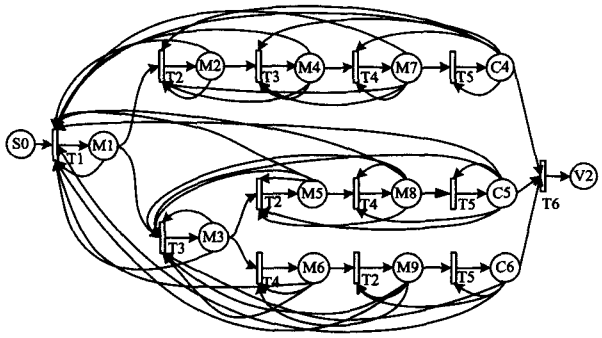

图 10　检测入侵场景 2 攻击的 PN 机模型

其中，S0(Safe state)表示初始安全状态，对象行为仍处于系统授权范围内；C1－C6(Critical state)表示系统对象处在系统脆弱状态的前一个状态；V1－V2(Vulnerable state)表示系统处于脆弱状态，即终态。P1、P2、P3、P4、P5、P6、P7、P8 及 M1、M2、M3、M4、M5、M6、M7、M8、M9 表示系统中间状态。

从图 9 和图 10 可知，当出现攻击序列：$S_1S_1S_1S_2S_2S_2S_3S_3S_3S_4S_4S_4S_5$、$S_1S_1S_1S_2S_2S_1S_2S_3S_2S_3S_4S_3S_5$、…，或者 $T_1T_1T_1T_2T_2T_3T_3T_3T_4T_4T_4T_5T_5T_6$、$T_1T_1T_1T_2T_2T_3T_2T_4T_3T_4T_5T_5T_6$、…，基于场景的 PN 机将自动识别攻击者冗余操作序列，推断出攻击者逃避入侵检测系统检测的意图。当到达系统临界状态时，系统可根据入侵危害等级，发出报警信号，提醒管理员注意。

4　复杂性分析

在基于入侵场景的 PN 机模型中，变迁在模型中负责规则逻辑的表达，体现为合一运算形式，合一运算的复杂性在于入侵检测进行时，变迁需要遍历所有 TOKEN，并将满足约束条件的参与运算。通常是以库所为参数，采用枚举逐个试探进行。下面分析试探性算法。

不失一般性，先考虑变迁出度为 1 的情况，假设 CPN 模型中某个变迁 T 的入度最大，令其为 p，则 T 的合一运算需要遍历所有可能的 p 元组，令 n 表示每种颜色的 TOKEN 至少有 n 个，如图 11 所示。

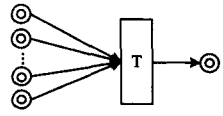

图 11　变迁 T 合一运算示意图

根据试探性算法，T 的合一运算时间复杂度为 $O(n^p)$。在一次合一运算中变迁 T 要消耗 p 个 TOKEN 并产生了 1 个新 TOKEN，则 T 的合一运算每成功执行一次整个系统的 TOKEN 数量减少了 $(p-1)$。

一般情况下，设合一运算将消耗 k 个 TOKEN 并产生 1 个新 TOKEN，某模型 N 的变迁数目为 x。依据 petri 网中变迁使能规则，有

$$x\cdot(p-1)\geqslant k-1\Rightarrow(p-1)\geqslant(k-1)/x$$

即 $p/\geqslant(k-1)/x+1$，因此变迁进行合一运算的最少时间复杂度为 $O(n^{[(k-1)/x]+1})$。

而本文中提出的是基于场景的 PN 机入侵检测模型，通过攻击操作序列的同构变换和拓扑排序，扩展到基于一类攻击的检测，每一个变迁都抽象代表着一个操作或命令的等价类。因此，假设对 k 个 TOKEN 进行分类，类别数目为 y，则有 $k\gg y$，即 y 远小于 k。因此在变迁数目 t 和每种颜色 TOKEN 数相同的情况下，本文提出的时间复杂度为 $O(n^{[(y-1)/x]+1})$。例如本文提取的部分操作或命令同构类，20 多个操作，分成 7 类。而在实际入侵检测情况下，y 的值会远远小于 k 的值，因此本文的时间复杂度远远小于原模型的时间复杂度，即 $O(n^{[(y-1)/x]+1})\ll O(n^{[(k-1)/x]+1})$。

Kumar 在文献[6]把攻击模式分为了 5 类："存在型"、"顺序型"、"偏序型"、"持续型"、"间隔型"。在 Kumar 层次性划分的基础上，本文将攻击分为简单攻击和复杂攻击。简单攻击大体上可对应于 Kumar 攻击模式的存在型，即可用一种攻击手段实施完毕，通过简单检测技术即可检测出的攻击行为。复杂攻击则指那些利用系统状态信息、上下文分析及网络拓扑信息等，体现为多个攻击行为围绕着一个攻击目标的一种攻击[18,19]，而不是仅仅依靠简单计数机制或匹配机制来进行判断。因此，在基于入侵场景的检测机制中，由简单攻击构成的攻击序列表现为复杂攻击，其候选攻击序列是简单攻击集合的幂集。所以把违反安全策略的事件作为元素得到 IDS 考察的数据全集 D_a；从 D_a 中筛选出由简单检测机制可以检测出的简单攻击事件集合 S_a，其中 $S_a\subseteq D_a$。从 S_a 中通过场景生成机制生成一类攻击序列，称为 C。则每类攻击序列 $C_i\in 2^{D_a}(i=1,2,\cdots)$，至此，所有的攻击序列 $\bigcup_{i=1}^{\infty}C_i\subseteq 2^{D_a}$。

假设 $\alpha=a_1a_2\cdots a_n$ 为抽象出的攻击者的攻击操作行为序列，并具有如下过程：

$$(1)\alpha=a_1a_2\cdots a_n\xrightarrow{\text{提取关键操作}(m\leqslant n)}b_1b_2\cdots b_m$$

$$(2)b_1b_2\cdots b_m\xrightarrow{\text{操作同构变换}(C\text{为操作等价类名称})}C_{b_1}C_{b_2}\cdots C_{b_m}$$

$$(3)C_{b_1}C_{b_2}\cdots C_{b_m}\xrightarrow{\text{操作拓扑排序(至少为 }m!\text{种)}}(C_{b_1}C_{b_2}\cdots C_{b_m})\mid(C_{b_2}C_{b_1}\cdots C_{b_m})\mid(C_{b_m}C_{b_2}\cdots C_{b_1})\mid(\cdots)\xrightarrow{\text{提取影响入侵的攻击序列}}(C_{b_1}C_{b_2}\cdots C_{b_m})\mid(C_{b_2}C_{b_1}\cdots C_{b_m})\mid(\cdots)\xrightarrow{\text{识别入侵者冗余操作}}((C_{b_1}^{+}C_{b_2}^{+}\cdots C_{b_m}^{+})\mid(C_{b_2}^{+}C_{b_1}^{+}\cdots C_{b_m}^{+})\mid(C_{b_m}^{+}C_{b_2}^{+}\cdots C_{b_1}^{+})\mid(\cdots))^{+}\xrightarrow{\text{构建自动检测攻击行为 PN 机}}\text{入侵场景 PN 机模型。}$$

因此，本文提出的基于场景的入侵检测 PN 机模型，降低了用于入侵检测的 petri 网的复杂度，减缓了状态组合复杂性[20,21]，使之对攻击表达更准确，刻画更严谨。

5　构建入侵检测体系

基于场景和 PN 机的入侵检测系统体系结构如图 12 所示。系统的输入是网络数据包，经过前端检测代理和后端检测代理后，系统的输出是针对不同行为所采取的相应措施。

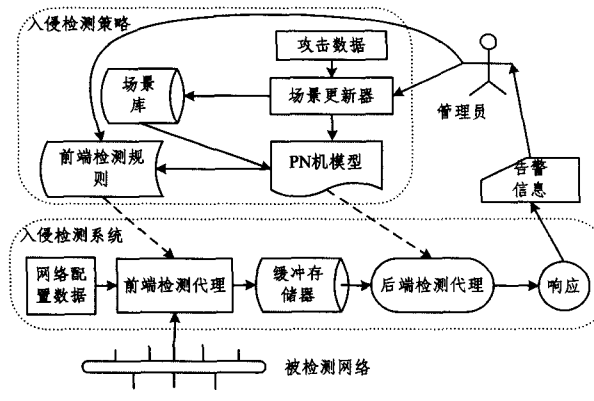

图 12　基于场景和 PN 机的入侵检测体系架构

该体系结构由两大部分组成：入侵检测策略和入侵检测系统[22]。策略制订模型包括4个部分：场景更新器、场景库、PN机模型、前端检测规则。场景更新器将不断出现的、新的攻击数据组织成场景，通过场景更新器存入场景库中，进而构造检测该入侵场景的PN机模型。场景库存放入侵场景。PN机模型根据入侵场景构造检测入侵的PN机，PN机的变迁用来封装规则以及规则之间的关系，TOKEN表示匹配的历史状态信息，库所视为TOKEN的容器，通过基本算子操作将检测入侵的PN机按照自顶向下、逐步细化的方式组织成攻击场景模型。前端检测规则将入侵场景攻击模型划分为原子事件，供给前端检测代理进行数据采集和过滤。入侵模型策略制订流程如下：

(1)将不断出现的攻击数据由场景更新器组织成入侵场景；

(2)构造检测某入侵场景的PN机；

(3)以入侵场景划分原子事件制订前端检测规则；

(4)以检测入侵的PN机实现后端检测代理机制。

入侵检测系统由网络配置数据、前端检测代理、缓冲存储器、后端检测代理和响应模块组成。

网络配置文件主要是对被检测网络的拓扑结构信息、重要的主机和服务器信息进行描述，描述信息包括主机和服务器上运行的操作系统、所在网段、开启的服务、服务的协议类型。

前端检测代理由数据采集和数据分析组成，数据采集负责对原始数据进行过滤解包和数据的规格化；数据分析则根据前端检测代理规则对采集来的数据进行网络层、运输层、应用层原语的解析，将不符合协议的数据简单地丢弃，并从符合的数据中检测出有用的信息，组织成时间序列的事件格式送到缓冲存储器中，供后端检测代理进行检测。

后端检测代理负责对前端检测代理提供的数据进行处理，从中找到检测该入侵场景的PN机模型。

该体系结构包括了网络安全的检测策略、实施检测、响应入侵。当新的攻击形式出现时将组织成入侵场景并转化成PN机模型，进而实施新的入侵检测策略。

结束语 本文深入地研究了入侵行为，探讨了攻击序列中关键操作之间的依赖关系，研究了将一个已知攻击通过派生及变种组织成一类攻击，以形成入侵场景的技术，进而提出了基于场景和PN机的入侵检测模型，该模型由入侵检测策略制订和入侵检测系统两大部分组成。本文的工作探讨并解决了一些已知攻击及其变种攻击的检测问题。因此从某种程度上说，我们提出的方法能够检测新的攻击。

参 考 文 献

[1] Modi C, Patel D, Borisaniya B, et al. A survey of intrusion detection techniques in cloud [J]. Journal of Network and Computer Applications, 2013, 36(1): 42-57

[2] Pradhan M, Pradhan S K, Sahu S K. A Survey on Detection Methods in Intrusion Detection System [J]. International Journal of Computer Application, 2012, 3(2): 81-90

[3] Teng Shao-hua, Du Hong-le, Wu Nai-qi, et al. A cooperative network intrusion detection based on fuzzy SVMs [J]. Journal of Networks, 2010, 5(4): 475-483

[4] Teng Shao-hua, Zhang Wei, Fu Xiu-fen, et al. Cooperative intrusion detection model based on state transition analysis[J]. Lecture Notes in Computer Science, 2008, 5236: 419-431

[5] Dolgikh A, Nykodym T, Skormin V, et al. Colored Petri nets as the enabling technology in intrusion detection systems [C]// Proc. of the 2011 Military Communications Conference. IEEE, 2011: 1297-1301

[6] Kumar S. Classification and detection of computer intrusions [D]. The degree of Doctor of Philosophy, Purdue University, 1995

[7] Slagell M. The Design and Implementation of MAIDS (Mobile Agents for Intrusion Detection System) [D]. The degree of Doctor of Philosophy, Iowa State University, 2001

[8] Helmer G, Wong J, Slagell M, et al. Software fault tree and coloured petri net-based specification, design and implementation of agent-based intrusion detection systems [J]. International Journal of Information and Computer Security, 2007, 1(1): 109-142

[9] El Bouchti A, Haqiq A. Malicious Insider Attacks Based Colored Petri Nets Approach [J]. International Journal of Engineering & Technology, 2013, 1(4): 177-191

[10] Zhao W, Wang P, Zhang F. Extended Petri Net-Based Advanced Persistent Threat Analysis Model [C]// Proc. of the 2013 3rd International Conference on Computer Engineering and Network. 2013: 429-434

[11] Saad S, Traore I. Extracting attack scenarios using intrusion semantics [C] // Proc. of the 5th International Conference on Foundations and Practice of Security. 2013: 278-292

[12] Bishop M, Peisert S. Your security Policy is what? [R]. The University of California, Davis, 2006

[13] 庄克深，张宏，张棍，等. 无线传感器网络中的病毒传播动力学研究 [J]. 计算机科学，2013，40(3)：187-191

[14] Ilgun K. USTAT: A Real-time Intrusion Detection System for UNIX [C]// Proc. of the IEEE Symposium on Research in Security and Privacy. 1993: 16-28

[15] 蒋昌俊. 离散事件动态系统的PN机理论 [M]. 北京：科学出版社，2000

[16] 刘培顺. 判决PN机理论及其在入侵检测中的应用 [D]. 成都：西南交通大学，2005

[17] 袁崇义. Petri网的应用[M]. 北京：科学出版社，2011

[18] Ben-Porat U, Bremler-Barr A, Levy H. Vulnerability of network mechanisms to sophisticated DDoS attacks [J]. IEEE Transactions on Computers, 2013, 62(5): 1031-1043

[19] Guitton C, Korzak E. The Sophistication Criterion for Attribution: Identifying the Perpetrators of Cyber-Attacks [J]. Journal of Royal United Services Institute, 2013, 158(4): 62-68

[20] 沙静，杜玉越. 基于标号随机Petri网的GSM性能分析[J]. 计算机科学，2012，39(7)：29-31

[21] 李凤英，古天龙，常亮，等. 一种基于赋时Petri网和ZBDD的装配序列规划方法[J]. 计算机科学，2012，39(2)：175-178

[22] 吴希. 基于Petri网的层次型入侵检测系统[D]. 南京：东南大学，2005