第 32 卷 第 16 期 计 算 机 工 程 2006 年 8 月
Vol.32 № 16 Computer Engineering August 2006
·安全技术· 文章编号：1000—3428(2006)16—0125—02 文献标识码：A 中图分类号：TP309

入侵检测中的归纳学习方法

刘培顺[1]，王学芳[2]
(1. 中国海洋大学计算机科学系，青岛 266071； 2. 中国海洋大学数学系，青岛 266071)

摘 要：结合使用着色 Petri 网和 EDL 语言描述攻击模型，该文给出了使用归纳学习对攻击模型进行泛化和特化操作，泛化后的模型可以检测出与已知攻击实例类似的未知攻击行为，实现了攻击知识库进行自动更新和扩展的方法。攻击实例首先使用 EDL 语言表述为一个攻击实例模型，对实例模型进行泛化得到攻击实例的 3 层概念空间，进而转化为着色 Petri 网模型，利用着色 Petri 网的运行机制对攻击行为进行检测。实验结果表明该方法对于具有相似攻击行为的未知攻击的检测非常有效。
关键词：入侵检测；归纳学习；着色 Petri 网；泛化；特化

Inductive Learning in Intrusion Detection

LIU Peishun[1], WANG Xuefang[2]
(1. Department of Computer Science, Ocean University of China, Qingdao 266071;
2. Department of Mathematics, Ocean University of China, Qingdao 266071)

【Abstract】This paper proposes the method for generalization and specialization of attack pattern using inductive learning, which can be used updating and expanding knowledge database. The attack pattern is established from an example by using the colored Petri net and EDL, after generalization it can be used to detect unknown attacks whose behavior are similar to the example. In practice the attack pattern first described by EDL from an example, then the pattern is generalized thus the concept spaces of attack are given and they can be transformed to Colored Petri net, when detection searches the intrusion from the top down by virtue of the concept space of the attack pattern. In fact the concept space of pattern indicates a depth-first search way.
【Keywords】Intrusion detection; Inductive learning; Colored Petri net; Generalization; Specialization

入侵检测中的漏报和误报问题是急需解决的两个问题，通常漏报和误报问题是一个相互矛盾的问题，对未知攻击模式的识别是漏报问题产生的主要原因[1~5]。本文首先使用着色 Petri网对攻击行为建模，然后使用归纳学习的方法对模型实例进行泛化，生成一个 3 层的概念空间，从而可以检测出与模型实例相似的未知攻击，提供了特化方法解决引泛化产生的误报问题，为解决这一问题提供了新的解决方法。

1 入侵检测中的归纳学习方法

1.1 用 EDL 构建模型

为便于建模，本文利用了定序并发表达式作为转换手段，因为任何一个并发表达式都可以转化为一个安全标号 PN 机[6]，且并发表达式的描述与自然语言的转换是很容易的。

为了把攻击行为表示为定序并发表达式，需要定义一种描述攻击行为的语言。本文使用了一种用于描述攻击的行为模式的语言(Event Descriptive Language，EDL)，且使用 EDL 可以构建定序并发表达式。EDL 的定义见文献[5]。使用 EDL 描述的攻击模式分为：(1)事件的定义；(2)事件之间的关系。事件通过关系构成了一个定序并发表达式。事件的描述由“事件类型(event-type)＋事件角色(event-Role)+事件操作(event-operation)＋操作目标(object)＋操作结果(result)＋时戳(time)+条件(constraint)”等不同的域构成，各个域的值可以是变量，也可以是常量。事件通过运算算子连接构成一个并发表达式，用文献[6]给出的转化算法，可以把 EDL 描述的攻击行为转化为一个着色 Petri 网。

例 W32.Blaster@mm 运行时所执行的部分操作如文献[7]所示。通过由 EDL 描述的 W32.Blaster@mm 行为模式，进而转化为 CPN 模型，如图 1 所示。

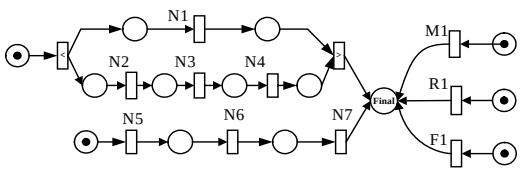

图 1 W32.Netsky.B@mm 的 CPN 模型

W32.Blaster@mm 的 EDL 描述σ如下：

```
Attack “Blaster” [MemMon, Regmon, Filemon, Netmon]
MemMon{
Add object=process   result= “msblast.exe” #define M1 }
Regmon {
Add object=“HKLM\Software\Microsoft\Windows\CurrentVersion
  \Run”   Result= “windows auto update”= “msblast.exe”
  #define  R1 }
  Filemon {
  add object=
  “c:\Winnt\system32” Result=“msblast.exe” #define F1 }
  Netmon {
  Attacker:
  Server: Protocol=Tftp, listen= udp/69    #define N1
```

作者简介：刘培顺(1975—)，男，博士，主研方向：密码学，信息安全，Petri 网应用；王学芳，博士、讲师
收稿日期：2005-10-23 **E-mail：**liups@263.net

Send: ip=$IP port=tcp/135 #define N2
Send: ip=$IP1 [IP1=N2.$ IP] port=tcp/4444 #define N3
Recv: sourceip=
$IP1 [IP1=N2.$IP locloPort=udp/69 #define N4
Victim:
Server: listen=tcp/4444 #define N5
Recv: souceip
=$ IP1 localport=tcp/4444, message= “tftp –i * get”
#define N6
Client: Protocol=Tftp, ip= $IP2, [$IP2=N6.$IP1]
port=udp/69 #defien N7 }
Relation＝(M1, F1, R1, AND (N1, FOLLOW(N2, N3, N4)), FOLLOW(N5, N6, N7))

本模型只能检测出当前的攻击实例，要使模型能够检测与该攻击类似的其他攻击，必须对模型进行泛化。攻击模型泛化的目的是提高模型的检测能力，使之能够检测出与攻击实例相似的攻击行为。

定义 孤立信息子：在 Petri 网中，如果一个子网从输入集仅仅经过一个变迁就到达输出集，则称这样的子网为孤立信息子。

1.2 泛化过程

归纳学习采用自底向上的方式构建概念空间，泛化过程分以下 3 步完成，设原始的使用 EDL 描述的攻击行为的并发表达式为α。

(1)对事件操作结果域泛化算法，分为事件泛化、关系泛化和处理孤立信息子 3 个步骤。

各个步骤的算法如下，输出为α_1。

事件结果域泛化算法如下：

输入为模型α
输出为模型α1
1)事件泛化：
begin 遍历α的所有事件
begin 读取α内的一个事件
if 事件的 result 域是常量
then 使用变量替换常量规则，增加选择项规则和构造型规则进行泛化
if 事件的 result 域的值在前面的事件描述中出现过
then 建立事件的关联
更新背景知识库
end
end
2)关系泛化：
begin 遍历α的所有事件
begin 读取α内的一个事件 a
begin 遍历α内的剩余事件 b
if b.操作＝a.操作 and b.object 域＝a.object 域
then 使用扩展值域规则把 b 合并到 a，a 增加 POWER 算子，并从剩余事件中去掉 b
end
end
end
3)处理孤立信息子：
begin 遍历α的所有事件
begin 获取α的孤立信息子
if α存在多个孤立信息子
then 修改多个孤立信息子的关系为 AND
if α存在一个孤立信息子
then 根据泛化选项，判断是否删除这个孤立信息子
end
end

更新背景知识库：
输入为事件 a
begin 遍历背景知识库的所有事件
begin 读取背景知识库内的一个事件 b
if b.操作包含 a .操作 and b.object 域 包含 a.object 域
then 增加背景知识 b 的权重
end
if 在背景知识库中没有找到 a 的类似事件
then 添加事件 a 到背景知识库中
end

变量替换常量规则就是把常量用变量替换，把原来的常量作为变量的值域里的一个值。增加选择项规则和构造型规则是在事件相关域的值域里加入背景知识，背景知识来自对其他攻击模型的学习，用户也可以自行添加。

为了降低误警率，需要对孤立信息子进行处理，如果有多个孤立信息子可以进行复合，把它们的关系改为 AND，检测结果通过 Fuse 算子融合后输入到终态，如果只有一个可以去掉或保持不泛化。

(2)对事件操作对象域泛化算法：对α_1进行第 2 步泛化(事件泛化、关系泛化)，算法流程与事件结果域泛化算法类似，输出为α_2。

(3)对α2 内所有事件的“ object ”,“ result ”,“ constraint ”使用舍弃条件规则进行泛化，并对泛化后在同一个子式内的相同操作项进行合并。设由此得到的并发表达式为α3。

在泛化时合并相同项的操作对应于 Petri 网模型中的替代操作。在泛化过程中通过引入变量和适当的扩展变量的值域把攻击实例检测的具体事例进行泛化，使得检测模型从一个点扩展到一个空间。通过引入背景知识增强了对泛化的目的性，背景知识在归纳学习过程中不断得到加强。

α1、α2、α3 构成了一个 3 层的概念空间，是一个逐步泛化的层次。α1、α2 维护了一个比α泛化的概念空间可以检测出不同程度泛化的未知攻击。泛化后α3 维护了一个模型检测事件类型和操作的一个列表，这层概念空间在检测时可以尽快地淘汰不需要的事件，提高检测效率。

检测时根据概念空间进行模式匹配，检测到的事件与模型中的事件匹配不再是严格的匹配，而是一种近似匹配，检测对应事件的域的包含关系，如果模型事件的包含了检测事件，则表示匹配成功。攻击模型泛化后检测能力大大提高，但是相应的误警率也提高了，出现超泛化现象。得到的概念空间不仅要足以覆盖所有的正例，还要排除所有的反例，这需要对泛化后的攻击模型使用特化操作来排除反例。

在 CPN 描述的攻击模型中，当出现反例时，需要根据具体的情况进行处理，主要分为以下几种情况：

(1)反例与孤立信息子有关

在 CPN 描述的攻击行为模型中，孤立信息子泛化后容易出现超泛化现象。在泛化时，如果有多个孤立信息子，可以把多个孤立信息子组成一个 AND 子式，防止超泛化现象。当出现反例时，如果反例主要是因为孤立信息子泛化导致的，则一般采取以下两种方式进行特化：

1)去掉孤立信息子：如果该攻击行为的网模型的其他信息比较丰富，且孤立信息子所代表的信息不是描述攻击行为的关键信息(驻留、传播、感染)，可以考虑去掉孤立信息子。

2)添加条件：如果孤立信息子的信息对于攻击行为的描述比较重要，在出现和孤立信息子有关的反例之后，可以考虑把反例的有关信息作为条件添加到孤立信息子的事件描述中。

(2)反例与某一个事件有关 (下转第 162 页)