基于过程挖掘和迹聚类的网络攻击建模分析方法

魏永鹏
安徽理工大学 数学与大数据学院, 安徽 淮南 232001

摘 要:目的 针对网络入侵检测系统发出的大量警报信息分析困难的问题,提出一种基于过程挖掘和迹聚类的网络攻击警报建模分析方法,根据网络安全日志信息建立并简化网络攻击模型。**方法** 该方法使用过程挖掘技术分析警报信息中的所包含的攻击者的行为信息和所使用的攻击方法信息,并以高层次可视化模型向网络管理人员提供攻击信息。利用基于频繁序列模式的迹聚类技术对日志活动间的行为关系进行分析,提取出活动的频繁序列模式,并且根据频繁序列模式对迹进行匹配,将相似的迹聚为一类,从而将一个警报日志 L 的复杂网络攻击警报模型分解为多个子日志的简单、直观的子模型。**结果** 仿真实验表明:提出方法得到的网络攻击模型在精确度、适应度和 F_1 分数上均有较好的表现;对于复杂的攻击模型,使用迹聚类方法可以生成多个低复杂性的模型,有效地降低其复杂性。**结论** 该网络攻击建模方法引入过程挖掘和迹聚类后,相较于传统建模分析方法可以更有效地反映出网络攻击者的入侵策略,并且对于复杂的攻击模型可以有效地降低其复杂程度。

关键词:过程挖掘;迹聚类;频繁序列模式;入侵检测;安全分析
中图分类号:TP391.9 **文献标识码:**A doi:10.16055/j.issn.1672-058X.2025.0006.011

Modeling and Analysis Method for Network Attacks Based on Process Mining and Trace Clustering

WEI Yongpeng
School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan 232001, Anhui, China

Abstract: Objective Aiming at the difficulty in analyzing a large number of alert messages issued by network intrusion detection systems, a method for modeling and analyzing network attack alerts based on process mining and trace clustering was proposed to build and simplify the network attack model according to network security log information. **Methods** This method used process mining technology to analyze the behavioral information of attackers and the attack methods contained in alert messages and provided attack information to network administrators through high-level visualization models. Behavioral relationships among log activities were analyzed using frequent sequence pattern-based trace clustering to extract frequent sequence patterns of activities, and similar traces were clustered into one class by matching traces based on frequent sequence patterns, thus decomposing the complex network attack alert model of a log L into simple and intuitive sub-models of multiple sub-logs. **Results** Simulation experiments indicated that the proposed method yielded network attack models with good performance in terms of precision, fitness, and F_1score. For complex attack models, the utilization of trace clustering methods could generate multiple low-complexity models, effectively reducing their complexity. **Conclusion** This modeling method for network attacks, with the introduction of process mining and trace clustering, can effectively reflect the intrusion strategies of network attackers compared with traditional modeling and analysis methods, and can effectively reduce the complexity of complex attack models.

Keywords: process mining; trace clustering; frequent sequence pattern; intrusion detection; security analysis

收稿日期:2024-07-05 **修回日期:**2024-09-21 **文章编号:**1672-058X(2025)06-0078-08
基金项目:国家自然科学基金资助项目(61572035)资助;安徽省重点研究与开发计划项目(2022A05020005)资助.
作者简介:魏永鹏(1999—),男,河南信阳人,硕士研究生,从事 Petri 网与过程挖掘研究. Email:1730432005@qq.com.
引用格式:魏永鹏. 基于过程挖掘和迹聚类的网络攻击建模分析方法[J]. 重庆工商大学学报(自然科学版),2025,42(6):78-85.
WEI Yongpeng. Modeling and analysis method for network attacks based on process mining and trace clustering[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(6): 78-85.

1 引　言

近年来,随着网络的高速发展,网络安全问题也被越来越多人的关注。网络入侵检测系统(Intrusion Detection System,IDS)已被广泛应用于监视网络和系统活动中的安全违规行为。当它检测到安全违规时,向网络管理员发出警报,网络管理员手动分析警报以支持响应计划。然而,由于网络攻击的日益频繁,IDS 每天会发出包含大量低级信息的警报,这些信息中并不包含攻击之间的逻辑和因果关系,难以分析出简单直观的攻击路径[1]。因此,如何从大量警报中分析得出简单直观的网络攻击模型是一个重大挑战。

为了解决这个问题,现有方法提出了警报预处理和警报关联技术。警报预处理技术的主要目的是通过减少假阳性警报,以在一定程度上减少警报的数量。文献[2]提出了一种方法,该方法使用聚类技术通过根本原因分析来减少误报警报。该方法旨在自动识别根本原因,然后编写规则来过滤警报。文献[3]提出了一种启发式算法来识别关键警报(指示根本原因),以便最大限度地提高由此引发的警报数量。在最近的一项工作中,文献[4]提出了用于分析入侵警报的综合系统,该系统使用警报优先级组件,该组件可以通过为警报分配优先级来检测误报。但以上研究仅实现警报日志之间的关联,即只识别警报日志是否误报,没有对过滤后的日志进行建模以及分析简化处理。

另一方面,关联技术旨在分析低级入侵警报之间的相似性和因果关系,为网络管理员提供高层次的、信息丰富的网络状态描述。文献[5]使用网络攻击警报特征之间相似程度来实现警报之间的关联。文献[6]和[7]提出了基于先决条件和后果来实现警报之间关联的方法,提出两级混合模型。该模型的第一级使用基于攻击图的方法,根据先决条件和后果关联已知攻击的警报。第二级使用基于相似性的方法,可以将错过的攻击警报关联起来,但这些警报无法映射到上一级攻击图中的漏洞。同时这种方法需要手动定义几种类型的攻击先决条件和后果,在大规模网络场景中,这个任务可能变得不切实际。因此,以上研究虽然能获得网络攻击模型,但由于模型的层次较低、复杂度较高,无法获得高层次的、简单的可视化网络攻击模型。

近年来,已有研究将过程挖掘与网络攻击建模相结合,该方法能够将攻击警报日志中的信息挖掘为工作流模型,能得到网络攻击的相应规律,但仍然存在结点和边的数量过多,网络攻击模型难以理解的问题[8-9]。为解决该问题,一些研究尝试对原始日志进行简化处理,例如采用迹聚类方法对日志进行降维[10],但此类方法尚未广泛应用于网络攻击日志的分析中。因此,可考虑采用更高效的过程挖掘方法,如启发式挖掘,以生成更简洁的过程模型[11]。因此,文献[12]提出了一种基于过程挖掘和图分割的网络攻击图分析方法,该方法首先使用启发式过程挖掘方法挖掘网络攻击图,随后将复杂的网络攻击图通过图分割的方式,分解为更简单的攻击图。文献[13]提出了一种基于过程挖掘和层次聚类的网络攻击可视化方法,通过层次聚类的方法将过程挖掘得到的模型进行简化,得到更简单的网络攻击模型。上述方法虽然能生成较为全面的网络攻击模型,但生成的模型仍然过于复杂,不利于理解。

此外,迹聚类算法可以将日志拆分为若干组同质的子日志[10]。使用现有过程挖掘算法对获得的子日志进行挖掘,可以得到结果更简单、更易于理解分析的网络攻击模型,从而将复杂的过程模型简化为多个简单的子模型,同时模型具有较低的复杂度和较高的精确性。因此,本文提出了一种基于过程挖掘和迹聚类的网络攻击警报建模及分析方法,主要工作如下:

(1) 提出了一种基于过程挖掘入侵警报建模分析方法,该方法通过融合过程挖掘方法,对日志活动间的依赖、循环关系进行分析,获取不同攻击者行为间的潜在逻辑关系,从而将一个警报日志转换为包含完备信息的可视化模型,有利于更直观的分析攻击行为。

(2) 提出一种基于迹聚类的网络攻击模型简化方法。该方法通过对日志活动间的行为序列关系进行分析,提取出活动的频繁序列模式,并且根据频繁序列模式对迹进行匹配,将相似的迹聚为一类,从而将一个警报日志 L 的复杂网络攻击警报模型分解为 n 个子日志的 n 个简单子模型。因而攻击模型的复杂程度更低,生成相对应的网络攻击警报模型更简单、直观,利于分析。

2 模型设计与构建

过程挖掘技术能够通过活动之间的行为关系挖掘工作流模式。为了进行复杂度分析和过程模型生成,本文使用启发式挖掘算法生成过程模型,因为它能更好地处理较大的事件日志。启发式挖掘通过忽略不频繁的路径来使用事件的频率和序列。它使用依赖关系和 AND/XOR 拆分连接来构造过程模型[11]。

定义 1[12] **依赖关系度量**。使用一个基于频率的度量来表示对两个事件 a 和 b 之间确实存在依赖关系的确定程度,记为 T。设 T 是一组活动,W 是 T 上的事件日志,且 $a,b\in T$,事件 b 发生在 a 之后,记为 $a>_W b$,$|a>_W b|$ 是 $a>_W b$ 在 W 中发生的次数,则对于使用启发式挖掘方法挖掘依赖关系,给出了以下公式:

$$a\Rightarrow_W b=\left(\frac{|a>_W b|-|b>_W a|}{|a>_W b|+|b>_W a|+1}\right) \tag{1}$$

其中,$a\Rightarrow_W b$ 的值始终在 -1 和 1 之间,表示 a 和 b 中的依赖关系程度。例如,在 60 条迹中,活动 b 直接跟在活动 a 之后,但相反的情况从未发生,则 a 的值 $a\Rightarrow_W b=$

60/61=0.984 表明能完全确定依赖关系。如果因为噪音导致出现一次活动 a 直接跟在活动 b 之后,则 a 的值 $a\Rightarrow_W b=59/62=0.952$,表明仍然能完全确定依赖关系,对依赖关系的影响程度很小。

然而,实际的事件日志中,会存在一些循环关系,比如活动 a 多次重复执行(例如,$abc,abbc,abb,\cdots,c$),这种称为短循环。短循环依赖关系的类型不能在依赖关系图中表示,存在活动不可观察的问题,而且它不能处理远距离依赖关系。因此,对于短循环关系使用依赖关系方程如下:

$$a\Rightarrow_W a=\left(\frac{|a>_W b|}{|a>_W a|+1}\right) \tag{2}$$

$$a\Rightarrow_{2W} b=\left(\frac{|a\gg_W b|+|b\gg_W a|}{|a\gg_W b|+|b\gg_W a|+1}\right) \tag{3}$$

除了短循环问题之外,还存在不可观测任务,即挖掘除依赖关系以外的 *AND* 和 *XOR* 关系,对于所有不可观测任务,启发式挖掘中使用依赖关系方程如下:

$$a\Rightarrow_W b\wedge c=\left(\frac{|b>_W c|+|c>_W b|}{|a>_W b|+|a>_W c|+1}\right) \tag{4}$$

给定一个事件日志 $W=[abc^5,aedbc^4,afd^4,abcd,acde]$ 则 $a\Rightarrow_W b\wedge c=(5+5/5+4+1)=1.0$,表明 b 和 c 是 *AND* 关系;$a\Rightarrow_W b\wedge e=(0/9+2)=0.00$ 表明 b、e 是 *XOR* 关系[11]。

3 网络攻击分析方法

用于探索网络攻击过程的过程挖掘方法旨在支持发现网络攻击所使用手段的主要过程序列。由于网络攻击的方法多种多样,并且来源众多,因此分析不局限于单个 IP、单种攻击方法。为了实现这些目标,本文提出了一种基于日志相似度聚类和过程挖掘的自动过程发现方法。如图 1 所示,首先执行数据收集和集成步骤,然后将收集到的数据转换为事件日志。之后,根据分析需要对事件日志文件进行过滤、清理和转换,再将网络入侵警报按照源 IP 进行分组,同时对分组后的日志按照相似度进行聚类,将攻击过程和方法相似的聚类为一组。最后,对使用迹聚类算法获得的子日志使用过程挖掘方法,获得网络攻击模型。

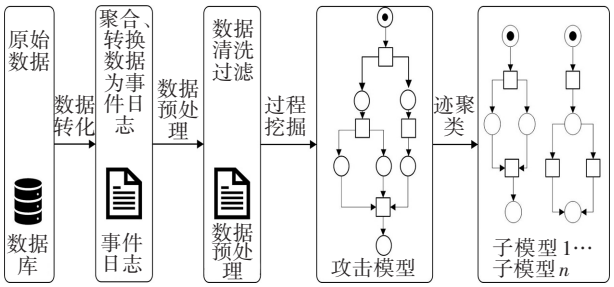

图 1 方法概述

Fig. 1 Method overview

3.1 数据聚合并创建事件日志

由于网络攻击分析方法旨在分析 IDS 设备中大量的警报日志信息,因此不需要整合来自不同数据源的数据。收集和迁移数据后,首先应该对数据进行聚合,可以用不同的方式执行警报聚合。每种聚合策略从攻击的不同角度对入侵警报进行分组。常见的聚合策略有两种[13]:(1)一对多,警报根据源 IP 地址聚合,形成组,其中单个源 IP 地址(攻击者)试图危害多个目标 IP 地址(目标);(2)多对一,根据目标 IP 地址聚合警报,形成多个源 IP 地址(攻击者)试图危害单个目标 IP 地址(目标)的组。本文使用多对一方式,即根据目的 IP 进行聚合。

下一步是创建一个事件日志文件,其中包含分析所需的所有数据。事件日志必须至少包含以下案例属性:源 IP 以识别攻击来源;目标 IP 以识别攻击来源;Activity 标识攻击方法;发动攻击时间的时间戳。日志文件还可能包含其他属性,使用这些属性可以扩大分析范围。表 1 显示了转化后的事件日志,在该表中,Timestamp、Activity 作为强制属性给出,目的 IP 以 Resource 的形式分组给出。

表 1 转化后的 IDS 警报事件日志

Table 1 IDS alert event log after conversion

Case Id	Timestamp	Activity	Resource
1	2016/4/1 9:12:00	Injection	x. x. x. x
	2016/4/1 9:15:00	Scan	x. x. x. x
	2016/4/1 9:16:00	Dos	x. x. x. x
2	2016/4/1 9:35:00	Xss	y. y. y. y
	2016/4/1 10:05:00	MitM	y. y. y. y
	2016/4/1 10:32:00	MitM	y. y. y. y
3	2016/4/1 22:36:00	Explotix	z. z. z. z
	2016/4/1 22:40:00	Worms	z. z. z. z
	2016/4/1 22:42:00	Worms	z. z. z. z

3.2 网络攻击建模

攻击模型的建立过程是方法的一个重要步骤。在这一步中,需要使用过程挖掘算法,该算法将 IDS 警报的事件日志作为输入,并生成表示在事件日志中观察到的行为的攻击模型作为输出。在本文的方法中,攻击模型由启发式挖掘算法生成。主要步骤如下:

首先将之前处理好的日志作为算法的输入,算法会根据活动的直接跟随关系构造一个依赖/频次表,根据得到的依赖/频次表,计算活动之间的依赖程度,生成一个依赖度量表,依赖程度是判断活动之间逻辑关系的重要指标;其次根据依赖/频次表和活动的依赖度量表建立活动之间依赖图,依赖图显示了活动之间的相互逻辑关系;最后将得到的依赖图转化为工作流网。根据事件日志使用启发式挖掘算法得到的网络攻击建

模如图 2 所示。

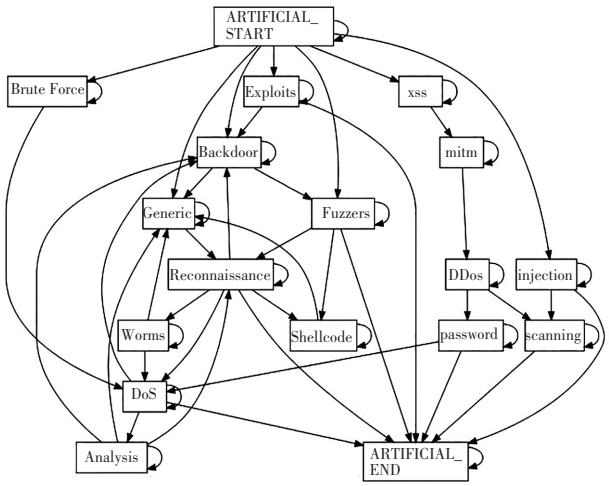

图 2　网络攻击模型示例

Fig. 2　Examples of network attack models

3.3　复杂模型的迹聚类简化

由于网络攻击的日益频繁,IDS 每天会发出包含大量低级信息的警报日志,这些信息中并不包含攻击之间的逻辑和因果关系,难以分析出简单直观的攻击路径。因此,需要对大量的低级警报日志进行处理。迹聚类可以有效地减少日志中同质信息的出现[14-15],因此,本文提出了一种基于频繁序列模式的迹聚类模型简化方法,可以将复杂的网络攻击模型拆分为多个简单、直观的子攻击模型,同时包含攻击者完备的攻击策略,便于分析人员进行分析,如图 3 所示。

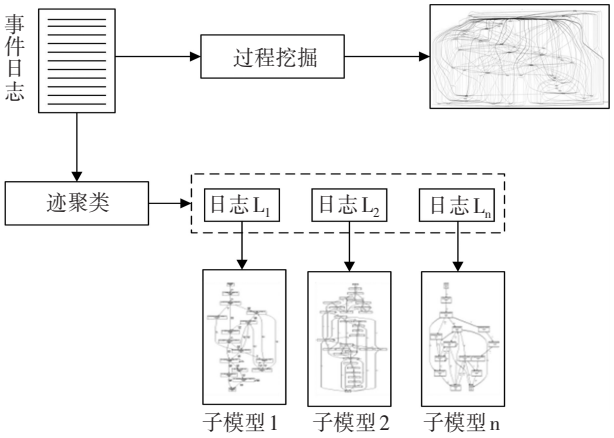

图 3　复杂模型简化流程

Fig. 3　Simpification process for complex models

3.3.1　算法概述

迹聚类是减少复杂过程模型出现概率的有效方法之一。本文所使用的迹聚类方法首先使用序列模式挖掘算法挖掘事件日志中的频繁序列模式,将事件日志拆分为多个子日志,其中包含频繁行为和非频繁行为,随后对获得的子日志进行复杂度的评估,直至最后获得符合复杂度要求的子日志,并输出获得的子过程模型,如图 4 所示。接下来算法 1 进一步阐述所提方法的主要步骤。

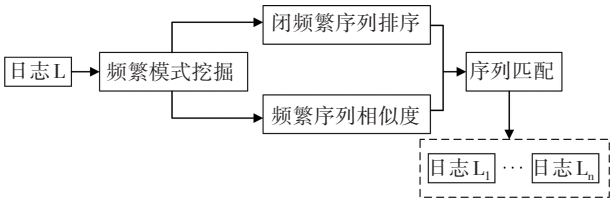

图 4　聚类算法概述

Fig. 4　Overview of clustering algorithms

步骤 1　为了防止生成包含太少迹线的簇(迹线太少意味着模型非常简单),首先设置每个潜在簇的最小大小 k。随后检查原始迹集中的迹数,如果迹的数量不足以划分为两个大于或等于 k 的子集,则算法直接结束;

步骤 2　使用频繁序列模式挖掘算法寻找事件日志中的频繁模式,根据频繁序列模式将事件日志拆分为多个子日志,包含频繁行为(frequent behavior,fb)和非频繁行为,选择频繁行为中的闭频繁行为序列对日志进行聚类划分;

步骤 3　对步骤 2 得到的多个子日志执行挖掘算法,判断模型的适应度以及复杂度,对于符合复杂度要求的日志不再进行进一步划分,不符合的日志重复步骤 1 和步骤 2,直至得到所有符合要求的子日志;

步骤 4　判断所有得到的子日志的模型适应度、复杂度是否符合要求,若符合要求,则输出相对应的日志及其过程模型。

算法 1　生成简单网络攻击模型

输入:日志 L,阈值 k。

输出:简单攻击模型 M(subLog)

```
1) function Trace Clustering(L)
2)   if sum(L)<=2 k
3)     return L
4)   else
5)     M(L)←Model Discovery(L)
6)     if M(L) is not complex then
7)       return (M(L))
8)     else
9)       fb← mine frequent sequence(L)
10)        Trace Clustering(L)
11)    end if
12)  end if
13)  return (M(subLog))
14) end function
```

3.3.2　网络攻击模型复杂度判断

为了对网络攻击复杂模型进行简化,就需要对模

型的复杂度进行判断,本文所使用的复杂度判断方法基于文献[12-13]而来,根据模型的节点数量和边数量进行衡量,判断规则如下:

$$\text{Complex jugement}(G)=\begin{cases}0, |V|<N_1\\ 1, N_1<|V|<N_2 \ \&\ \text{Simlicity}(G)<\gamma\\ 0, N_1<|V|<N_2 \ \&\ \text{Simlicity}(G)\geqslant\gamma\\ 1, |V|>N_2\end{cases} \tag{5}$$

其中,$|V|$为攻击模型 G 结点的个数。若$|V|<N_1$,则网络攻击模型 G 为非复杂网络攻击模型;若$|V|>N_2$,则网络攻击模型 G 为复杂模型,需要进行简化;网络攻击模型满足 $N_1<|V|<N_2$,则通过稀疏度 Simplicity(G)进行判断,公式如下所示:

$$\text{Simlicity}(G)=\frac{|V|}{|E|} \tag{6}$$

其中,由文献[13]可知:$N_1=15$,$N_2=30$。

3.3.3 网络攻击事件日志迹聚类

该方法的第一步通过给出的序列模式挖掘算法 Γ 和最小支持度 $min-sup$,从事件日志 L 中挖掘出闭频繁序列模式,定义如下:

定义 2[14] **序列模式**。迹 $s=\langle s_1,s_2,s_3,\cdots,s_m\rangle$ 是不同活动的有序排列,其中当 $1<=i<=m$ 时,s_{i+1} 跟随 s_i。

当一条迹匹配序列模式时,这说明该条迹包含一个子序列,其中活动以相同的顺序出现。

定义 3[14] **序列模式支持度**,令 L 为事件日志,π_l 为标签函数,其中 $\pi_l(\sigma)=<a_1,a_2,\cdots,a_n>$是序列模式。$\sigma$ 匹配 S 当且仅当整数 $i_1,i_2,\cdots,i_m$ 使得 $1\leqslant i_1<i_2<\cdots<i_m\leqslant n$ 且 $s_1=a_{i_1},s_2=a_{i_2},\cdots,s_m=a_{i_m}$。记 $S\subseteq\pi_l(\sigma)$。

日志 L 中序列 S 的支持度是 L 中匹配 S 的迹的数量,即

$$\text{Support}(S,L)=\frac{|\{S\subseteq\pi_l(\sigma)\mid\sigma\in L\}|}{|L|} \tag{7}$$

定义 4[16] $CFBS=\{fb\mid fb\in\Gamma(T,min-sup)\}$,其中 T 是来自 L 的迹集合,Γ 是闭序列模式挖掘算法,$min-sup$ 是最小支持度。

根据上述定义,闭频繁行为序列(Close Frequent Behavior Sequence,CFBS)是从 T 中挖掘出的频繁出现的行为序列。本文认为,事件日志中某些频繁出现的子序列能够揭示业务流程的一些特别重要的标准,有助于区分事件日志中隐藏的具有不同功能的子过程模型。使用顺序模式的另一个好处是顺序模式不仅可以表示轨迹的连续结构行为,还可以表示不连续的轨迹行为。例如,给定包含一组迹 $T=\{<A,C,D,E>,<A,C>,<A,E>\}$和最小支持度 $min-sup=0.4$ 的事件日志,可以得到频繁行为集 $T'=\{<A,C>,<A,E>\}$,顺序模式$<A,C>$是一个连续的迹行为,因为活动 C 总是紧挨着 A 出现在一个迹中,而$<A,E>$是一个不连续的迹行为,因为活动 A 和 E 可能出现在一个离散的迹中。

下一步,将使用字符串匹配技术对上一步得到的闭频繁行为序列模式集对已有的事件日志进行划分。首先对闭频繁行为序列进行长度排序,得到闭频繁行为序列排序表;其次使用余弦相似度计算闭频繁行为序列之间的相似度,获得闭频繁行为序列的相似度度量值,将具有较高相似度的长度相近的闭频繁行为序列分为一组,若所有闭频繁行为序列的长度都一致,即只有一种长度,则根据闭频繁行为序列相似度进行划分;最后根据划分所得到的闭频繁行为序列集对事件日志进行划分。此外,不包含频繁行为的迹分为一组。对于得到的迹分组,对每组迹分别使用过程挖掘算法,然后判断生成的网络攻击模型是否为复杂模型,若为复杂模型,重复上述操作,算法如下所示:

算法 2 根据闭频繁序列进行日志聚类

输入:日志 L,闭频繁序列 $CFBS$,阈值 k

输出:简单攻击模型 M(subLog)

```
1) function Trace Clustering(L)
2)   M(L)←Model Discovery(L)
3)   if M(L) is not complex then。
4)     return (M(L))
5)   else
6)     for each close frequent sequence fb ∈ CFBS do
7)       for each trace t ∈ T do
8)         if fb⊆t then
9)           T1 = T1 ∪ {t}
10)        else
11)          T2 = T2 ∪ {t}
12)        end if
13)      end for
14)      if M(T1) is complex then
15)        repeat Trace Clustering(T1)
16)      end if
17)    end for
18)  end if
19) end function
```

迹聚类算法:使用上述概念,算法 1 描述了本文的迹聚类方法。算法采用贪心策略,检测最长闭频繁行为序列和行为序列相似度以迭代拆分事件日志。根据算法 1,对于输入事件日志 L,首先获取 L 的频繁行为序列和行为相似度,并初始化日志集。之后,迭代地将日志 L 分成几个子日志,直到生成的子日志对应的模型复杂度都较低或 L 中没有日志可以进一步划分(第 2~

15 行)。此外,如果生成的子日志(即 T_1 和 T_2)中的轨迹数小于阈值 k,则将生成的子日志合并到下一长度的闭频繁行为序列的子日志中。这里,阈值 k 用于防止算法生成具有太少迹的子日志。最后,算法返回一个子日志集合。

下面通过一个实例来对算法 2 进行具体解释,表 2 给出了 4 条迹的事件日志。

表 2　事件日志案例

Table 2　Event log cases

id	act	id	act	id	act
T_1	A	T_2	E	T_3	D
T_1	B	T_2	C	T_3	F
T_1	C	T_2	D	T_4	A
T_1	E	T_2	F	T_4	E
T_1	D	T_3	A	T_4	C
T_1	F	T_3	B	T_4	F
T_2	A	T_3	C		

首先根据事件日志计算迹的频繁序列模式,该事件日志一共包含四条迹变体,为 T_1=<A,B,C,E,D,F>,T_2=<A,E,C,D,F>,T_3=<A,B,E,D,F,G>,T_4=<A,E,C,F>。通过使用频繁序列模式算法可以计算得出闭频繁序列模式为<A,E,F>,<A,C,D,F>,之后根据算法 2,依次从最长序列进行聚类,可以得到聚类结果为 C_1={T_1,T_2},C_2={T_3,T_4}。

4　仿真实验与结果分析

本节介绍了一个案例研究,旨在评估本文中提出的方法。首先,介绍案例研究中使用的数据集,其次,对所使用的网络攻击模型生成算法进行质量分析,保证所生成的模型能较好地反映出攻击者的攻击策略,随后对所使用的迹聚类方法进行可行性验证,说明所提出的方法的合理性,最后运用所提出的方法对得到的复杂模型进行简化分析,说明本文方法的有效性。

4.1　数据集

为了评估所提出的方法,本文采用的实验数据集为某 IDS 设备在 2016 年所生成的网络攻击警报。实验数据集中主要记录了攻击者的行为信息,其中主要字段如表 3 所示。选择 4 月份的警报进行分析。为了执行所提出的方法的第一步,聚合具有相同源 IP 地址的警报。然后,定义案例,将时间跨度 t 设置为 1 d。因此,在同一天触发的具有相同源 IP 地址的警报与同一案例相关联。此外,对于启发式过程挖掘,本文使用 python 的 pm4py 库实现。将按天处理好的事件日志导入 python 程序中,即可得到每个 IP 每天受到攻击的网络攻击模型。

表 3　字段说明

Table 3　Field description

字　段	含　义	说　明
Timestamp	时间戳	攻击警报发生的时间
SourceIP	源 IP	攻击者的 IP 地址
SourcePort	源端口	攻击者使用的端口号
DestinationtIP	目的 IP	攻击目标的 IP 地址
DestinationPort	目的端口	攻击目标的端口号
Lable	标签	攻击的类型
EventLogID	日志 ID	日志记录的顺序

4.2　结果分析

4.2.1　网络攻击模型质量分析

为了评估模型的质量,使用已有的度量指标,即适应度(fitness)和精确度(precision)进行评估[17]。具体而言,适应度量化了日志中案例可以被过程模型完全重放(replay)的比例。精度量化了模型能中案例与日志中案例不一致的比例。适应度与精度之间往往存在一个平衡,为了寻找这个平衡,引入了 F_1 分数来评估模型的整体性能[18]。本文使用的对比方法为归纳式算法[19]和 alpha[20]算法。本文选取了五天的日志分别计算精确度、适应度、F_1 分数,并进行平均化处理,结果如图 5 所示。

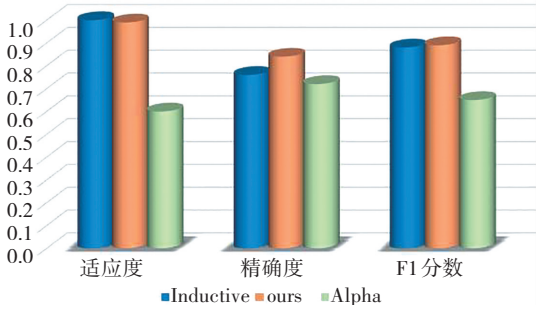

图 5　不同挖掘方法模型质量比较

Fig. 5　Comparison of model quality for different mining methods

从图 5 可知,基于启发式挖掘得到的网络攻击模型在精确度、适应度和 F_1 分数上的值优于归纳式以及 alpha 算法,因此,可以表明启发式过程挖掘算法能够有效地从警报日志中挖掘出攻击者所对应的攻击模型,以便于分析攻击者采取的攻击策略。

4.2.2　聚类简化结果验证

聚类方法可行性验证。本组实验主要为了验证本文所用聚类方法在复杂网络攻击模型简化上的可行性。本文首先使用 Petri 网建模工具实现了恶意邮件和 Nmap 扫描两种网络攻击的流程,其次使用 PLG 工具对生成了 10 000 条迹的人工日志。然后应用本文的迹聚类方法对日志进行划分,使用启发式过程挖掘方法从

两个子日志中发现过程模型。图 6 展示了网络攻击的具体流程,包含了多个子流程,看起来较为复杂,不利于网络管理员进行分析,表 4 描述了各个变迁的含义。图 7 和图 8 描述了经过迹聚类后产生的两个子过程模型。其中,图 7 描述了恶意邮件攻击的流程,该流程通过获取管理员的 E-mail 地址,向其发送具有恶意链接或信息的邮件来达到目的。图 8 描述了 Nmap 扫描攻击的流程,该流程首先通过主机发现,随后确定端口状况,在已经确定的端口上查找运行的具体应用程序的名字与版本信息,针对获取的信息进行网络攻击。通过该仿真实验说明本文的迹聚类方法能够把日志划分为多个同质的子日志,证明了方法的可行性。

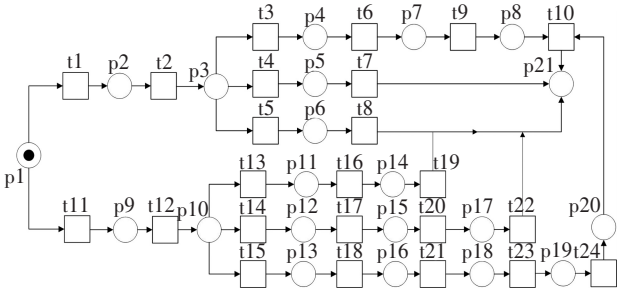

图 6 恶意邮件和 Nmap 扫描网络攻击 petri 网流程

Fig. 6 Petri net process of malicious email and Nmap network scanning attacks

表 4 变迁含义

Table 4 Meanings of changes

变 迁	描 述	变 迁	描 述
t1	获取管理员 Email	t13	获取端口
t2	发送恶意邮件	t14	网络资源信息
t3	安装后门	t15	发现漏洞
t4	安装跳板	t16	Dos 攻击
t5	安装 Sniffer	t17	获取网络资源
t6	嗅探	t18	利用漏洞
t7	远程控制	t19	系统崩溃
t8	攻击其他目标	t20	IPS 攻击
t9	获得账号口令	t21	获取权限
t10	获得控制权	t22	植入木马
t11	获取域名、IP 信息	t23	提升权限
t12	Nmap 扫描	t24	取得控制权限

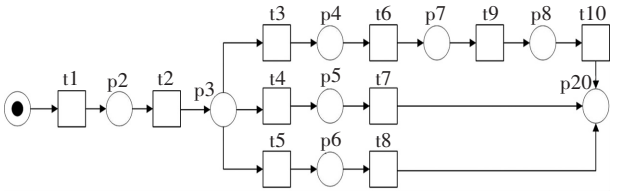

图 7 恶意邮件网络攻击 petri 网流程

Fig. 7 Petri net process of malicious email network attacks

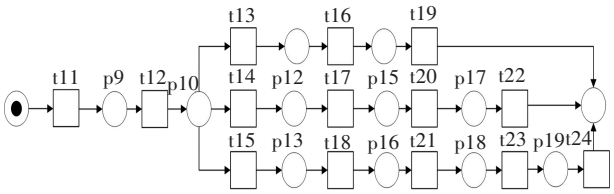

图 8 Nmap 扫描网络攻击 petri 网流程

Fig. 8 Petri net process of Nmap network scanning attacks

4.2.3 网络攻击模型简化

由于 IDS 通常会发出大量警报,使得人工分析极为困难,因此旨在评估与 IDS 生成的大量警报相关的简化方法。选择数据集中发出警报数量最多的那一天。所选日期为 2016 年 4 月 11 日,这一天发出的警报数量有 294 833 条。由此可知,对如此大量的警报进行手动分析是不可行的。这需要自动化方法来协助网络管理员分析警报。

第一步是基于警报源 IP 地址的一对多聚合对警报进行分组。如前所述,目标是形成与同一攻击者的行为相关联的警报组。然后,由单个警报形成的组和仅由相同攻击方法形成的组(即与多阶段攻击无关)被过滤。因此得到了由 3 286 个案例,23 个活动组成的数据集。下一步包括使用事件日志来发现攻击模型。这一步的结果生成了一个具有 19 个顶点和 86 条边的攻击模型,该模型被归类为复杂模型。因此,为了使模型分析可行,必须使用迹聚类技术将模型聚类成更小、更简单的模型。在应用本文的聚类算法后,复杂模型被聚类为 10 个子模型。图 9 显示了聚类过程后其中一个模型的示例。从图 9 可知,攻击者以 4 种攻击方式开始,随后按照一定的攻击顺序采取不同类型的攻击方法对网络进行入侵。因此,图 9 的攻击子模型可以表明本文的方法能较好地反映出攻击者的攻击策略。

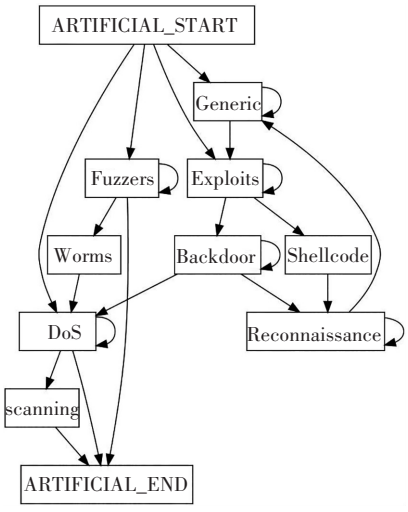

图 9 聚类后攻击子模型

Fig. 9 Attacking sub-models after clustering

5 结　论

本文提出了一种基于过程挖掘和迹聚类的网络攻击模型生成及简化方法,该方法可以通过分析攻击行为之间的依赖关系,构建网络攻击模型;对于生成的模型还可以检查其复杂程度,并对复杂模型通过迹聚类方法进行简化,该迹聚类方法主要通过检测警报事件日志的闭频繁序列来实现对同质的日志进行聚类,将复杂的事件日志分为多个简单的子日志,并得到相应的子攻击模型,增强了模型的可读性。为了验证本文方法的有效性,通过使用一个网络攻击数据集,证明了本文的方法可以有效地反映出网络攻击者的入侵策略,并且对于复杂的攻击模型,可以有效地简化其复杂程度。此外本文只考虑了对已有警报进行建模分析,没有考虑警报实时在线分析,下一步可以考虑对入侵进行实时检测及分析,以便能更好地对网络系统进行保护。

参考文献(References):

[1] NING P, CUI Y, REEVES D S. Constructing attack scenarios through correlation of intrusion alerts[C]//Proceedings of the 9th ACM Conference on Computer and Communications Security. New York: ACM, 2002: 245-254.

[2] JULISCH K, DACIER M. Mining intrusion detection alarms for actionable knowledge[C]//Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2002: 366-375.

[3] ZONG B, WU Y, SONG J, et al. Towards scalable critical alert mining[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2014: 1057-1066.

[4] SHITTU R, HEALING A, GHANEA-HERCOCK R, et al. Intrusion alert prioritisation and attack detection using post-correlation analysis[J]. Computers & Security, 2015, 50: 1-15.

[5] LEE S, CHUNG B, KIM H, et al. Real-time analysis of intrusion detection alerts *via* correlation[J]. Computers & Security, 2006, 25(3): 169-183.

[6] NING P, XU D. Learning attack strategies from intrusion alerts[C]//Proceedings of the 10th ACM Conference on Computer and Communications Security. New York: ACM, 2003: 200-209.

[7] WEIJTERS A J M M, RIBEIRO J T S. Flexible heuristics miner (FHM)[C]//Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining. Piscataway: IEEE Press, 2011: 310-317.

[8] PHILLIPS C, SWILER L P. A graph-based system for network-vulnerability analysis[C]//Proceedings of the 1998 Workshop on New Security Paradigms. New York: ACM, 1998: 71-79.

[9] JHA S, SHEYNER O, WING J. Two formal analyses of attack graphs[C]//Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15. Piscataway: IEEE Press, 2002: 49-63.

[10] REIJERS H A, MENDLING J, DIJKMAN R M. Human and automatic modularizations of process models to enhance their comprehension[J]. Information Systems, 2011, 36(5): 881-897.

[11] WEIJTERS A, VAN DER AALST W M P, DE MEDEIROS A K A. Process mining with the heuristics miner-algorithm[J]. Technische Universiteit Eindhoven, 2006, 166(6): 1-34.

[12] CHEN Y, LIU Z, LIU Y, et al. Distributed attack modeling approach based on process mining and graph segmentation[J]. Entropy, 2020, 22(9): 1026.

[13] DE ALVARENGA S C, BARBON S, MIANI R S, et al. Process mining and hierarchical clustering to help intrusion alert visualization[J]. Computers & Security, 2018, 73: 474-491.

[14] LU X, TABATABAEI S A, HOOGENDOORN M, et al. Trace clustering on very large event data in healthcare using frequent sequence patterns[M]. Cham: Springer International Publishing, 2019: 198-215.

[15] SUN Y, BAUER B, WEIDLICH M. Compound trace clustering to generate accurate and simple sub-process models[M]. Cham: Springer International Publishing, 2017: 175-190.

[16] SUN Y, BAUER B. A novel top-down approach for clustering traces[M]. Cham: Springer International Publishing, 2015: 331-345.

[17] SONG M, GÜNTHER C W, VAN DER AALST W M P. Trace clustering in process mining[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 109-120.

[18] DE WEERDT J, VANDEN BROUCKE S, VANTHIENEN J, et al. Active trace clustering for improved process discovery[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(12): 2708-2720.

[19] VAN DER AALST W, WEIJTERS T, MARUSTER L. Workflow mining: discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142.

[20] LEEMANS S J J, FAHLAND D, VAN DER AALST W M P. Discovering block-structured process models from event logs-a constructive approach[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 311-329.

责任编辑:陈　芳