入侵检测语言的评估[1]

孙美凤[①,②]　　龚 俭[①]
①东南大学计算机系网络中心 南京 210096
②扬州大学信息工程学院　扬州

摘要　入侵检测语言是滥用入侵检测系统用于定义攻击场景的表示规范。本文提出一种比较和评估入侵检测语言的方法，该方法建立在一个可被证明是互斥和完备的分类基础上，并基于表达能力、表示简洁性、检测强度等三个测度。使用该方法可以对现有的各类检测语言表示攻击特征并进行推理的相对有效性进行评估，从而揭示出现有检测语言的缺陷和理想的入侵检测语言应具有的特性。

关键字　入侵检测系统；检测语言；检测算法；评估
中国图分类号 TP393

An Evaluation Method for Intrusion Detection Language

Sun Meifeng[①,②]　　Gong Jian[①]
（msun、jgong）@njnet.edu.cn
①Computer Science and Technology Department, SouthEast University, NanJing, 210096
②Information and Engineering College,YangZhou University, YangZhou

Abstract Intrusion detection language is an expression specification used by IDS to describe the intrusion senarios. Based on a mutually exclusive and exhaustive taxonomy of network attacks and their detection languages, this paper proposes an evaluation method for intrusion detection languages with three metrics:expressibility, representational succinctness and detection intensity. Those well-known detection languages have been evaluated using this method in terms of their ability to express attack signature and their detection efficiencies, so as to reveal their shortages and the features that an ideal detection language should have.

Keywords　intrusion detection system; detection language; detection algorithm; evaluation

1 引言

滥用检测认为攻击可描述为针对特定漏洞的特定行为模式，这些行为模式的出现代表攻击的发生。检测语言是滥用入侵检测系统（IDS）用于定义攻击场景的表示规范，它可编码攻击场景以及响应行为以形成入侵特征库；检测算法根据入侵特征库给出的知识在输入事件

[1] 本课题得到国家自然科学基金（90104031）资助. 孙美凤，女，1970 年生，博士研究生，主要研究方向为网络安全监测. E-mail: msun@njnet.edu.cn. 龚俭，男，1957 年生，博士生导师，主要研究方向为网络安全、网络管理和网络体系结构.

流中寻找恶意行为的证据，并做出结论。因此，攻击场景表示是基于规则的 IDS 的首要内容，是攻击行为规律的概括与抽象模型。与自然的表示方法比较，检测语言表示应具有抽象性、概括性和简洁性，并受“刻画攻击现象”和“计算可接受”两个可能冲突的条件的限制。

在入侵检测发展的二十年中出现了大量的检测语言，其中包括 P-BEST[1,2]、状态语言[3,4]、有色 Petri 网[5]、简单规则语言[6~8]、陈述性的语言[9~11]等类型，它们对“刻画攻击现象”和“计算可接受”这两个条件有不同的侧重，但在实践都不能完全令人满意。为了改进现有的入侵检测体系，有必要探讨将检测语言标准化的可能，因为这种标准化可大大减少特征编写的重复劳动，扩大特征共享和交流的范围，方便不同 IDS 的集成和互操作。[12]对此进行了探索，但是这方面的工作显然还不多。

本文提出一种比较和评估检测语言的方法。该方法不仅有助于为不同的应用环境寻找最好的检测语言；而且其研究可揭示现有检测语言的不足，促进新的检测语言的出现，这对于标准化工作的需求分析是十分有益的。在给出了本文所使用的一些概念的形式定义之后，文章首先介绍了检测语言的一种分类方法，它可被证明是互斥而完备的，从而构成本文的基础。文章随后对检测语言“刻画攻击现象”的能力以及检测算法计算复杂性进行了讨论，给出了表达能力、表示简洁性以及检测强度三个测度的形式定义，并以此为依据评估了各类检测语言的完备性、简洁性和检测性能。文章最后归纳了研究结果。

2 一些基本概念

图 1 是 IDWG 给出的 IDS 模型[13]。由于研究内容限于滥用 IDS 的 Analyzer 组件，其输入是 Event，输出是 Alert，因此不发生歧义的情况下，本文直接将 Analyzer 组件称为滥用 IDS。本节首先定义一些基本概念，符号的使用尽量与 IDWG 保持一致。

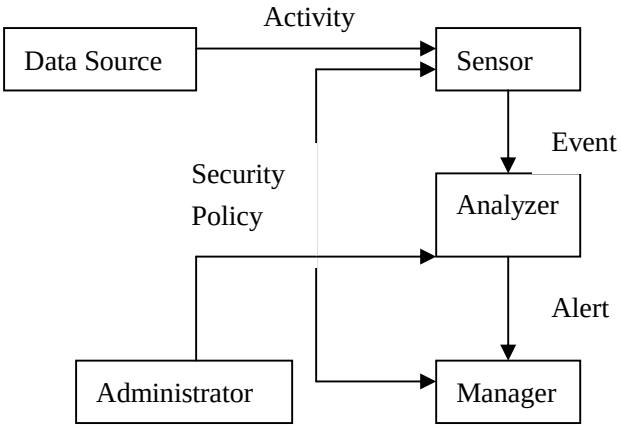

图 1　IDWG 的 IDS 模型

FieldSet：域集，$FS=\{f_1,f_2,\cdots,f_n\}$，$f_i$ 是审计记录或网络报文中特定部分的内容的集合，如源/目的地址、用户标识等。对$\forall f$，$f\in FS$，f 有名字和值域两个属性，分别记为 N(f)和 D(f)。

Event：事件 E。E 是一个数据结构，记录了系统或用户行为的信息。$E=\{(N(f_1),v_1),(N(f_2),v_2),\cdots,(N(f_n),v_n)\}$，$f_i\in FS$，$v_i$ 是事件 E 的 f_i 域的值，记为 $f_i(E)$。

Trail：事件流 T，有穷的事件序列 $E_1E_2\cdots E_n$。T 的第 I 个事件记为 T[I]，I 是事件 T[I]的位置。T 是滥用 IDS 的输入。

传统的攻击分析方法致力于揭示每个入侵的区别于背景噪声以及其它入侵的特征行为。攻击有实例、方法和类型的区别。实例是攻击的一次出现，它总是遵循一个具体的攻击方法；攻击方法是一个线性的行为模式；而攻击类型是一种攻击所有可能方法的总和，是行为序列

的集合，可通过行为序列模板来描述。为了逃避检测，入侵者可能对攻击行为以及行为的顺序尝试各种变化，因此定义的模板应当能够描述可能的变形。

Filter: 过滤器。F 是一组作用于单事件的约束 $C_1 \wedge C_2 \wedge \cdots \wedge C_n$，$C_i$ 具有（$N(f_i), r_i, t_i$）的形式，其中 $f_i \in FS$ 是一个域，r_i 是一个二元关系，$t_i \in P(D(f_i))$，$P(D(f_i))$表示 f_i 值域的幂集。例如：C_1=(端口号,大于,80)；C_2=(源地址,属于,某个子网范围)；C_3=(传输层协议类型,等于,TCP)；则 C1∧C2∧C3 定义了一种事件类型。

InterConstrain：关联约束 IC,IC 是不同 F 域值的关系 $F_i \oplus F_j$ 的集合，⊕是约束算子。⊕可以是 E_i、和 E_j 某个域值的约束，可以是 E_i、和 E_j 发生先后顺序的约束（如果把 E 在 T 中的位置也看作 E 的域，则发生顺序的约束可以统一称为域值的约束）。

Scenario：场景，记为 S。令∑是 F 的集合，$S = \{\omega \in \Sigma^+ : \omega$具有性质 P$\}$,$\Sigma^+$是∑上所有不为空的 F 的序列集合，P 是定义在∑上的一组约束 IC。例如：某攻击类型“包含 A、B、C 三个行为，要求 A 优先于 B，B 优先于 C”,则该攻击类型的 S 中的∑={A,B,C}，P={A 优先于 B,B 优先于 C}，S={ABC}。

Multiphase Scenario:多阶段场景。如果某攻击场景 S 中存在ω且|ω|>1，则 S 称做多阶段场景。|ω|是ω长度。

S 是关于一种攻击类型的知识，通常由安全专家分析获得。S 的定义包含两层含义：（1）表示 S 要求表示∑和 P；(2)S 的形式可以归一为 F 的序列集合。正是基于 S 的形式可以归一的认识，本文提出了等价特征库的思想（4.1 节），展开了评估方法的研究。|S|是 S 集合的元素个数，代表攻击的变形数，显然|S|越大，表示和检测的复杂度越大。检测语言研究 S 的表示和检测的方法。

Detectionlaguage: S 的表示规范，记为 M，将面向多阶段场景表示的语言称为多事件语言。

Signature：特征，记为ρ。ρ是按照 DL 的语法规则构造的**场景**表达式。

Signature 的定义与 IDWG 有所不同，IDWG 将 Signature 定义为从表示到行为的规则，而本文的定义只强调表示。原因在于：入侵检测中算法的行为是有限而稳定的，例如场景中一个中间事件的匹配总意味着后续事件的搜索，而最后事件的匹配总触发报警，因此行为可以隐含在表示中。ρ是 S 的存在形式，一个 S 可能对应多个ρ。

Signature Base：特征库ℜ是特征ρ的集合。$\Re=[\rho_1, \rho_2, \ldots, \rho_n]$。由于ℜ是有序集合，因此用‘[]’代替‘{}’。

KL(r)：特征ρ包含的知识，也是ρ的形式语义。$KL(\rho) = \{\omega \in \Sigma^+ : \omega$具有性质 P，∑和 P 包含在ρ中$\}$。类似可定义特征库的知识，$KL(\Re) = \cup KL(\rho_i)$

M (r): 特征ρ使用的描述语言。

ρ、ℜ、 KL(ℜ)组成了滥用 IDS 的攻击知识的定义。

Alert：报警，记为 A。A 是滥用 IDS 报告给 Manager 的关于可疑事件的信息。基于上述概念，滥用入侵检测系统(MisIDS)是作用于特征库和事件流的推理系统 f：(ℜ,E)->A。ℜ是攻击的知识，E 是数据源,f 是检测算法。

3 检测语言的分类

按照[14]的观点，分类学的理想特性包括：

（1） 互斥：要求各个类别不重叠，每个实例只能属于一个分类；

（2） 完备：分类的集合包括所有的可能性；

（3） 无二义性：分类标准具有确定性，不管由谁分类，得到相同的结果；

（4） 可重复：重复应用得到相同的分类；

（5） 可接受：符合逻辑，具有直观性，能够被大多数人接受；

（6） 有用：服务于研究目标。

其中，互斥和完备是理想分类方法的最重要特性。通过分类可以系统地描述研究对象，确定研究的边界；另外通过分类可将对所有实例的研究转为对类的研究，使结果具有普遍性。

Kumar 于 1994 年较为全面地分析了当时发生的攻击（主要基于 Unix），归纳出四个层次的攻击场景，并使用有色 Petri 网统一表达攻击特征。尽管随着互连网规模的扩大和攻击手段的进化，攻击向着大规模、分布式的方向发展，但单元攻击的手段变化不大，因此 Kumar 的攻击分类在今天仍具有指导意义。有色 Petri 网隐含子场景组合成场景的三种方式：与、或、顺序与。

首先说明两个操作。°操作具有字符串连接的所有性质，表示两个 F 序列的连接，如ω1=“AB”，ω2=“CD”，ω1°ω2 =“ABCD”， •操作表示两个 F 序列任意位置连接，ω1•ω2 = {“ABCD”，“ACBD”，“ACDB”，“CABD”，“CADB”，“CDBA”}。•操作是可交换的：ω1•ω2 = ω2•ω1；•操作可结合：ω1•ω2•ω3 = （ω1•ω2）•ω3。•操作具有非常强的表示和生成序列的能力。

令 S1,S2,S 为任意的场景，其中 S 由 S1 和 S2 组合而成：

定义 3.1

顺序与（S = S1 then S2）：S = {ω1°ω2：ω1∈S1,ω2∈S2}；

或(S = S1 or S2)：S = S1∪S2；

与(S = S1 and S2)：S = {ω1•ω2：ω1∈S1，ω2∈S2}。

与形式语言理论的研究对象不同，检测语言不存在预定义的有穷过滤器表，甚至过滤器在描述时可能不确定。例如：Snort 将端口扫描攻击的场景描述为“时间 T 内某个主机访问目的主机 TCP 端口数超过 N”，该场景可形式化描述为“$X_1X_2\cdots X_n$”，其中的 X_i 代表 IP 报文，任意两个 X_i 要求同源同宿。Kumar 将这种需要统一不确定事件的场景称为 Unified。

基于上述讨论，攻击场景存在两类 IC：组合方式和内容约束。Unified 场景的存在极大地增加了场景的复杂度，进而也增加了表示和检测算法实现的难度。但在另一方面，所有的多事件语言都使用变量支持 Unified 场景的表示。能否直接表示攻击场景的组合方式是检测语言本身的一个重要性质，也是不同检测语言的最根本的区别，这种区别导致不同形式的特征库和不同的检测能力，因此本文将检测语言包含的组合方式作为分类标准。

定义 3.2 g**关系**：令 M’是检测语言的集合，γ = {(M1,M2)：M1 ,M2∈M’且∀x：x∈{then,and,or},M1 支持 x 等价于 M2 支持 x}。

显然γ关系自反、对称和传递。

定义 3.3 **分类**m： m是检测语言集合 M’上的按照γ关系构造的划分。

m = {[{}]，[{then}] [{and}]，[{or}]，[{then , and}]，[{then , or}]，[{and , or}] ，[{then , and, or}]}，其中，对∀x： x⊆{then, and, or}，

[x] = { M | M ∈ M’且 M 支持且仅支持集合 x 中的组合方式}。

为了书写的简便，本文去掉了[x]中 x 的集合符号，即将[{then,or}]直接缩写为[then,or]。分类μ是利用检测语言的语法元素进行的分类，这一点保证了分类的无歧义和可重复。同时，分类μ是建立在等价关系上的划分，因此互斥而完备。这里的完备性是基于定义 3.1 给出的三种组合方式，而非语义的完备。

有两类不同的表示方法：过程性表示和说明性表示。过程性表示用于表示怎么做的知识，而说明性表示用于表示是或否的知识。这两种表示方法论同时存在于检测语言的研究中。

说明性表示支持的组合方式是显然的，而图形化表示（有色 Petri 网、STATL）的组合方式通常也容易直观看出。P-Best 和 Russel 是典型的事件驱动的规则语言，每个特征具有 IF…THEN…的形式，IF 给出事件类型的说明即过滤器，THEN 给出响应行为，主要行为有激

活特征和发送报警信息。激活特征行为规定了后续过滤器（then），IF 是 or 的典型语法，因此 P-Best 和 Russel 都支持 then 和 or 的直接表达。由于能够利用事实库存储匹配的中间状态，P-Best 能够直接表达 and。

表 1 给出了典型检测语言的分类及其应用

表示方法	分类	语言	应用系统
说明性	[]		Snort(1999)
	[then, and, or]	Sutekh(2001)	
	[then , and]	Musigs(1999)	CARDS(2000)
		Parsing Schema(2002)	
		ASL(2002)、	
过程性	[then]		USTAT(1992)、
	[then , or]	STATL(2000)	
			NetSTAT(1999)
	[then, and, or]	Colored Petri-Net	IDIOT(1995)
		P-BEST	NIDES(1995)、EMERALD(1997)
	[then , or]	RUSSEL	ASAX(1992)
			BRO(1999)

4 检测语言的评估测度

4.1 表达能力

从完备性的角度看，在入侵检测中使用的规则应该能够涵盖尽量大范围的检测能力。由于攻击是一个动态发展中的概念，因此根据覆盖程度定义检测语言的绝对表达能力是困难的，正如我们现在不能证明现有的程序设计语言或以我们目前的理解所设计的新型程序设计语言一定能够有效地满足未来应用的编程需要一样。

按照模型理论[15]：任何形式的知识库都编码了有关这个世界为真的命题，从而刻画了一组“可能世界”；如果两个知识库编码了同样的可能世界的集合（模型），则它们是等价的。滥用 IDS 的特征库是特殊形式的知识库，编码了所有攻击类型的知识。特征库的知识是有穷的 F 的序列的集合，因此可判定任意两个特征库的知识是否相等。

定义 4.1 表达能力相等 LG=：令 M’ 是 M 的集合，$\Re$’ 是$\Re$的集合，

$$LG= = \{(A, B): A,B\in M' \wedge$$
$$\forall x(x\in \Re' \wedge DL(x)=A \rightarrow \exists y(y\in \Re' \wedge DL(y)=B \wedge Kl(x)=Kl(y))) \wedge$$
$$\forall y(y\in \Re' \wedge DL(y)=B \rightarrow \exists x(x\in \Re' \wedge DL(x)=A \wedge Kl(x)=Kl(y)))\}。$$

类似可定义表达能力小于关系 LG< 和表达能力大于关系 LG>。

定理 4.1 then 是表示多阶段场景的必要条件，增加组合方式并不增加表达能力。

证明：

对$\forall S$：$S = \{\omega\in\sum^{+}:\omega$具有性质 P$\}$，如果|S|有穷，人们总可以通过枚举 S 中所有$\omega$的方法表示它。因为 S 是多阶段场景，$\exists\omega'$：$\omega'\in S$ 且$|\omega'|>1$,给出ω'需要且仅需要 then。

仅考虑组合方式对|S|的影响。一条使用 and、or 组合的任意场景，只要原陈述方式有穷，那么|S|一定有穷，因此可以枚举 S。

得证。

4.2 表示简洁性

为了检测攻击，IDS 的设计者或管理员需要按照语言规范将攻击场景编码成ρ，不同的语言编码同一场景可能需要不同数量的ρ。例如：一个攻击场景由 ABCD 四个过滤器组成，不同过滤器之间没有顺序要求。该攻击场景对应了有色 Petri 网的一条ρ，而仅仅支持 then 的 USTAT 需要 4!条ρ。不能全面完整地表达攻击特征给系统的分析和性能带来许多困难，包括：

a) 从实现的角度，规则数量越少，入侵检测算法越能够有效地运行；

b) 特征分散在多条规则当中，漏报率会增加，特别是在有负载均衡的运行环境中；

c) 可维护性差；

d) 特征扩展的成本高，例如在 STATL 中增加否定属性，将使得特征图变得非常复杂。

检测语言表达的性能差异不仅表现在同一场景需要不同数量的ρ，还表现在ρ的不同复杂程度上。例如一条由多个子场景经过 and、or、then 组合而成的场景 S 可被编码为有色 Petri 网的一条ρ；通过给出 and 组合的子场景的所有顺序可能，该 S 同样可被编码为 STATL 的一条ρ，但ρ中 F 的数量指数增长，因此 F 数量也可度量ρ的复杂程度。特征库$\Re$中编码的 F 的总数称为特征库规模,记为 Size($\Re$)。因为任意攻击知识一定可以通过枚举 F 的序列给出，并且这种特定形式的特征库在等价特征库中具有最大规模，所以它可以成为表示简洁性的基准，本文将这种形式的特征库称为基准特征库 Baseline($\Re$)。

定义 4.2 表示简洁性f：M’->N 是从 M 的集合 M’到自然数集合的函数。

$\forall\Re$：DL($\Re$)=M $\wedge$Size($\Re$)=k，f(M)等于将$\Re$转化为 Baseline($\Re$) 时规模的最大增长率。f(M)可表示成 k 的函数，显然f(M)越大，M 越简洁。

定理 4.2 在 then 组合的基础上，增加 or 组合将使语言的表示简洁性多项式增强，增加 and 组合将使语言的表示简洁性指数增强。

证明：

按照 定义 3.1 ：or(s1,s2) = s1$\cup$s2, s1 then s2={ω1ω2: ω1$\in$s1,ω1$\in$s2}；

or 组合就是集合的并操作，纯粹的 or 不会改变特征库的规模；仅当 or 组合与 then 结合使用，才会导致表示简洁性的增强。

$\forall$S：DL(S))$\in$[then,or]，k 是 S 中序列的最大长度，则 S 中最大序列数<$(Size(S))^k$。因此$\forall$M ,M$\in$[then,or] ，f(M)<$k(Size(S))^k$是多项式增长函数。

$\forall$S：DL(S))$\in$[then,and]，S 中入度为 0 的初始结点数（简称偏序数）最大为 I，S 可以表示成一个有向无循环图（每个节点的出度=1，存在节点的入度>1 或为 0）。S 展开成的序列数<$(I)^{(Size(S))}$，每个序列的状态数为 Size(S),则总的状态数< $(Size(S))\ (I)^{(Size(S))}$。

因此$\forall$M ,M$\in$[then, and] ，f(M)是指数增长函数。

得证。

通常人们仅从完备性的角度讨论表示方法“刻画攻击现象”的能力，这种观点可能使各种不同形式的检测语言不可比较，正如定理 4.1 中指出。表示简洁性测度评估不同检测语言表示相同攻击知识的相对性能，是表达能力测度的重要补充。

4.3 检测强度

对检测语言的第二个要求是支持高效的检测。检测算法依赖于检测语言，目前不存在通用的形式化的检测算法。对检测算法进行比较存在的主要困难是：不同检测语言支持的组合操作不同,相应检测算法面向的解空间就不同,因此必须经过某种处理建立统一的参照标准，

比较才有意义。基于等价特征库讨论检测语言的推理性能可消除了检测算法解空间的差异，从而为评估建立了统一的参照标准。

仔细观察各种检测算法，它们大部分利用了模式匹配的思想，例如：STATL、有色 Petri 网是典型的状态机模式；规则语言 RUSSEL、P-Best 直接编码了模式的查找过程，活动规则包含匹配过程的所有当前状态信息；逻辑语言和 parsing schema 算法的推导过程也是模式的搜索过程。由于入侵检测巨大的审计流量特性，匹配过程逐个状态向前推进，状态编码了匹配的过去信息，避免了回溯，因此算法的性能必然和特征库的规模具有某种关系。

定义 4.3 检测强度Á： M’ ->N 是从 M 的集合 M’到自然数集合的映射关系。

∀ℜ：DL(ℜ)=M，f(M)等于将ℜ转化为 Baseline(ℜ) 时推理时间的最大增长率。

定理 4.3[5] 包含 Unified 变量的最简单的序列特征的匹配问题是 NP 完全的。

类似于“AXBXD”的 Unified 场景因为包含统一变量 X，检测算法必须保存 X 的每个可能值，文献[5]证明该问题 NP 问题。本文仅讨论组合方式对推理复杂性的影响。

定理 4.4 在 then 组合的基础上，增加 or 组合将使语言的检测强度多项式增强，增加 and 组合将使语言的检测强度指数增强。

证明：

首先对∀S: DL(S)∈[then,or]，or 使得某个状态存在多个分支（注意与不确定性区别）。假设 E_i 使得匹配向分支 a 进行，问题是不确定的后续事件流 T 中可能正好不出现分支 a 的剩余模式而包含分支 b 的模式，因此 S 的匹配必须保存当前已到达的所有状态，用于必要时回溯。当 E_i 到达时，检测算法应该与所有已到达状态的所有出边匹配，因此 S 的时间复杂性 O(Size(S) n)，n 是输入事件数。

对∀S: DL(S)∈[then,and] ,不确定性会产生分支，当前状态同样是状态集合，记录当前可能到达的所有状态，因此 S 的时间复杂性 O(kn)，$1 \le k<O((Size(S))^{I}n)$,I 为偏序数,n 是输入事件数。

对∀S: DL(S)∈[then]，S 的匹配复杂性是 O(n)，n 是输入事件数。

∀S：DL(S)∈[then, and]，I 是最大偏序数；

S 的时间复杂性是 $O(kn), 1 \le k<O((Size(S))^{I}n)$, S 最多得到 $I^{(Size(S))}$个攻击序列，Baseline(S)的时间复杂性 $O(I^{(Size(S))}n)$，因此∀M:M ∈ [then, and], ℑ(M)是指数次函数。

∀S：DL(S)∈[then, or]，S 的时间复杂性是 O((Size(S))n)；假设 k 是攻击序列的最大长度，因此 S 最多得到$(Size(S))^{k}$ 个攻击序列，Baseline(S)的时间复杂性为 $O((Size(S))^{k}n)$，因此∀M:M ∈ [then, or], ℑ(M)是 K 次多项式函数。

得证。

5 检测语言的比较

（1） 表达能力

观察表 1 给出的典型的检测语言，它们分别属于[]、[then]、[then,or]、[then,and]和[then,or,and]。[]类语言不支持任何组合方式，只能表达单事件特征。其它语言在 then 的基础上不同程度的支持其它组合，统一称为多事件语言。按照定理 4.1，表 1 中检测语言的表达能力比较：

结论 1 [ ] LG< [then] LG= [then,or] LG= [then,and] LG= [then,or,and]

单事件语言的任何特征库是多事件语言特征库的特例，因此多事件语言的表达能力大于单事件语言，这和人们的直观理解一致。令人惊讶的是：尽管大量的多事件语言具有不同的

形式（复杂的 Petri 网、简单的序列），但它们的表达能力相等。

（2） 表示简洁性

表示简洁性是表达能力的补充测度,度量了检测语言表示相同知识的相对能力。按照定理 4.2 ，[then]类语言的表示简洁性为 O(1)；[then,or]类语言的表示简洁性是 $O(Q^k)$，[then,and]类语言的表示简洁性是 $O(I^Q)$；I 是偏序数，Q 为特征规模，k 是攻击序列的最大长度。表 1 中检测语言的表示简洁性比较：

结论 2 [then] LG< [then,or] LG<[then,and] LG<[then,or,and]

显然，为了表示上下文相关攻击的特征，检测语言必须支持 then 组合方式，在此基础上增加越多的组合方式,表示简洁性越强,其中 and 组合是提高表示简洁性的强有力的手段。

（3） 检测强度

检测强度度量不同检测语言在相同知识上的最大推理能力。定理 4.4 表明：[then]的特征数最大，假定特征间串行检测，[then]的检测强度 O(1)；[then,or]类语言的检测强度是 $O(Q^k)$， [then,and]类语言的检测强度是 $O(I^Q)$；I 是偏序数，Q 为特征规模，k 是攻击序列的最大长度。表 1 中检测语言的检测强度比较：

结论 3 [then] LG< [then,or] LG< [then,and] LG< [then,or,then]

比较结论 2 和结论 3 ，表示越简洁，检测强度越大。通常时间性能和空间性能是一对矛盾，但是在入侵检测环境，这两者一致。

6 结论

本文的研究表明：

a) 表示简洁性测度是一个有效而独立的测度。
b) 从表达能力的角度看，检测语言可分为两大类：单事件语言和多事件语言，表 1 中给出的那些多事件语言尽管形式不同，但其表达能力是等价的。描述是检测的前提，一个强表达的检测语言是滥用 IDS 的基础，影响 IDS 的检测准确率。因此本研究结果同时表明这些支持多事件关联检测的 IDS 在理论上检测能力上应当是等价的。
c) then 组合是检测语言描述多事件攻击特征的必要条件。And 组合能够指数地提高检测语言的表示简洁性和检测强度，是检测语言的理想特性。
d) 检测强度依赖表示简洁性。定理 4.4 不仅指出检测语言的发展方向，而且指明滥用 IDS 性能优化的方向，减少特征库规模将提高检测性能。
e) Unified 特征的匹配是 NP 完全的。该结果提醒人们：对于 Unified 特征，除穷尽搜索没有实际有效的算法。因此最好的设计可能不是寻找统一的算法解决所有攻击特征的匹配，而是将攻击特征分类，并且为每类特征寻找最优算法。同时，穷尽搜索要求保留攻击特征的所有部分匹配，随着时间的推移不可避免的产生状态爆炸问题，因此实现时简化的合理性问题变得很重要。
f) 现有语言都缺乏语义完备性的讨论。主要有两点：（1）从逻辑的角度，否定不可缺少，STATL 和有色 Petri 网一定程度的支持否定，但是前者过于复杂，后者粗糙、不准确；（2）现有的语言都假定事件瞬时发生，因此仅支持有序和无序的时序关系。这种基于单点检测的假定在分布式大规模的攻击和检测环境是否继续成立或如何保证它的成立需要进一步研究。

按照事件的组合方式进行检测语言分类是本文提出的另一个概念。这种分类建立在语法的基础上，因此是无歧义和可重复的。由于组合方式与检测语言的描述能力密切相关，因此本文提出的分类学是按照检测语言的相对表达能力进行分类，直接面向本文的研究目标。

参考文献

1. ebra Anderson, Thane Frivold, Alfonso Valdes . Next-generation Intrusion Detection Expert System (NIDES) A Summary. SRI-CSL-95-07, 1995. http://www.sdl.sri.com/nides/reports/4sri.pdf

2. A. Porras and P. G. Neumann. EMERALD: event monitoring enabling responses to anomalous live disturbances. In : Proceedings of the 20th National Information Systems Security Conference. Baltimore, Maryland,USA, 1997. 353—365. http://www.sdl.sri.com/emerald/emerald-niss97.html.

3. K. Ilgun. USTAT: A real-time intrusion detection system for UNIX[Master's thesis]. Computer Science Dept., University of California, Santa Barbara, USA, 1992.

4. Vigna, G. and R.A. Kemmerer. NetSTAT: a network-based intrusion detection system. Journal of Computer Security, 1999,7(1): 37-71.

5. S. Kumar. Classification and Detection of Computer Intrusions[PhD thesis], Dept. of Computer Science, Purdue University, USA, 1995.

6. .Habra, B.Le Charlier, A. Mounji, and I. Mathieu. ASAX: Software Architecture and Rule-based Language for Universal Audit Trail Analysis. In : Proc Of (ESORRICS)' 92. Springer-Verlag, 1992:435-450.

7. V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks, 1999,31(23--24):2435—2463.

8. M.Roesch. Snort-Lightweight Intrusion Detection for networks. In: Proceedings of USENIX LISA' 99 conference,1999: 229-238.

9. J.-L.Lin and X.Sean Wang amd S.Jajodia. Abstraction-Based Misuse Detection: High-level Specifications and Adaptable Strategies. In: Proc. Of the 11th Computer Security Foundations Workshop, Rockport, MA, 1998:190--201.

10. C.Michel and L.Me. Adele: an Attack Description Language for knowledge-based Intrusion Detection. In: Proc. Of the 16th International Conference on Information Security,2001. http://citeseer.nj.nec.com/michel01adele.html.

11. J-P.Pouzol and M.Ducasse. From Declarative Signatures to Misuse IDS. In : Proceedings of the RAID International Symposium, Davis, CA, 2001,2212:1-21.

12. G.Vigna, S.T. Echmann, and R.A. Kemmerer. STATL: An Attack Language for State-based Intrusion Detection. Dept of Computer Science University of California Santa Barbara, 2000. http://citeseer.nj.nec.com/eckmann00statl.html.

13. Mark Wood,Mike Erlinger.Intrusion Detection Message Exchange Requirements[EB/OL]. http://www.ietf.org/internet - drafts/draft - ietf - idwg - requirements – or.txt. October 20, 2001.

14. John Howard. An Analysis of Security Incidents on the Internet. PhD thesis, Carnegie Mellon Univerisity, August 1998.

15. Nils J.Nilsson 著 郑扣根 庄越挺 译. 人工智能.北京：机械工业出版社，2000.

16. Harry R.Lewis, Christos H.Papadimitriou 著 张立昂 刘田 译.计算理论基础. 第二版.北京：清华大学出版社，2000.