基于区间决策图的威胁处置策略快速匹配

张玲翠[1,2]，李凤华[1,2]，房梁[1]，郭云川[1]，李子孚[1]
（1. 中国科学院信息工程研究所，北京 100195；2. 中国科学院大学网络空间安全学院，北京 100049）

摘 要：威胁界定模糊性和策略大规模性导致难以快速准确地选取处置策略，针对该问题，提出基于区间决策图的威胁处置策略快速匹配方法。首先对威胁处置策略进行归一化描述，定义了模糊化的威胁处置策略；将威胁类型、严重程度、置信度、攻击频度、传播方式等策略匹配条件进行排序，提出了带模糊算子的区间决策图构造算法，设计了面向威胁处置的策略动态匹配算法，实验证明了其有效性。

关键词：威胁处置；区间决策图；快速匹配
中图分类号：TN929
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021074

Fastly match threat response policies based on interval decision diagram

ZHANG Lingcui[1,2], LI Fenghua[1,2], FANG Liang[1], GUO Yunchuan[1], LI Zifu[1]
1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: Due to the inaccuracy of threat detection and the scale of response policies, it is very difficult to accurately select response policies. To address the above problem, fuzzy interval decision diagram to quickly match response policy was proposed. Firstly, the response policy was formally and fuzzily defined. Considering threat type, threat level, attack frequency and propagation mode, an algorithm with fuzzy operator was designed to construct interval decision diagram. Further, a fuzzy match algorithm was proposed to quickly select response policies. Experimental results show the efficiency of the proposed approach.

Keywords: threat response, interval decision diagram, fastly match

1 引言

随着互联网、移动互联网、物联网等复杂网络的大规模应用，安全威胁的发生日趋频繁，产生的后果日益严重。国家互联网应急中心发布的《2020 年上半年我国互联网网络安全监测数据分析报告》显示，2020 年上半年，捕获计算机恶意程序样本数量约 1 815 万个，日均传播次数达 483 万余次，涉及计算机恶意程序家族约 1.1 万余个，我国境内感染计算机恶意程序的主机数量约 304 万台，境内工业控制系统的网络资产持续遭受来自境外的扫描嗅探，日均超过 2 万次。

面对如此严重的安全威胁，学术界和工业界提

收稿日期：2021−02−22；**修回日期**：2021−03−23
通信作者：李子孚，lizifu@iie.ac.cn
基金项目：国家重点研发计划基金资助项目（No.2016YFB0801001）；国家自然科学基金资助项目（No.U1836203）；中国科学院战略性先导科技专项基金资助项目（No.XDC02040400）；山东省重点研发计划（重大科技创新工程）基金资助项目（No.2019JZZY020127）

Foundation Items: The National Key Research and Development Program of China (No.2016YFB0801001), The National Natural Science Foundation of China (No.U1836203), The Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC02040400), Shandong Provincial Key Research and Development Program (Major Science and Technology Innovation Project) (No.2019JZZY020127)

出了涵盖潜在风险点识别、威胁风险评估、威胁处置策略生成与匹配、处置效果评估等在内的一整套威胁闭环响应解决方案。从静态动态的角度来看，当前威胁处置策略匹配包括 2 类：静态匹配和动态匹配[1]。静态匹配指为每类报警预先分配固定的威胁处置策略或措施，当报警发生时，查找与之相匹配的策略或措施。虽然静态匹配简单直接，但依赖于专家知识，不具备动态性和适应性。为此研究者提出了动态匹配，该方式不仅考虑报警类型，而且响应措施依赖于系统状态和当前环境等因素，从而提升了动态性和灵活程度。在这 2 种匹配中均可引入攻击损失和响应代价，以提升威胁处置效益。

现有静动态匹配方案要求当前威胁报警中的要素与响应策略库或措施库精确匹配，只有精确匹配时才能选择出响应策略或措施。虽然成本敏感匹配模式中的效果和成本属于模糊匹配，但在其他要素中属于精确匹配。然而在实践中精准快速地匹配策略库或措施库是非常困难的，其原因如下。

1) 威胁响应决策条件的模糊性。在响应策略库中，需要定义当何种威胁发生时采取何种措施，然而威胁复杂性导致很难准确地选择响应措施。如威胁响应策略中规定“在给定时间段内某网段的某类报警达到 10 次，则应该阻断该网段的 A 类流量”。若攻击者触发 9 次报警之后停止攻击，则该攻击者行为不会被阻断。在实践中，若攻击者触发 9 次报警，则应以极大的可能性阻断该类行为，这种威胁响应决策条件的模糊性导致难以有效地选取威胁处置措施。

2) 策略库或措施库规模大。复杂网络等环境中存在大量设备、海量数据、各种类型的威胁，相应地，策略库或措施库规模将非常庞大。例如，骨干网交换、大型数据中心、云存储服务、大规模社交网络后台等网络边界的防火墙、安全网关/设备策略中，响应规则可达 10 万条。庞大的策略库或措施库将消耗过多的策略匹配时间。

针对上述问题，本文提出了带模糊算子的区间决策图构造算法，设计了面向威胁处置的策略动态匹配算法，具体贡献如下。

1) 对威胁处置策略进行归一化描述，形成威胁处置策略。其中，威胁处置策略模板包括策略匹配条件、策略动作、策略有效性等；策略匹配动作包括处置命令类型、处置命令、处置区域、约束信息等，可为策略动态匹配提供基础。

2) 将威胁类型、严重程度、置信度、攻击频度、传播方式等策略匹配条件进行排序，提出了面向单策略的带模糊算子的区间决策图构造算法；针对威胁处置策略库中存在大量处置策略的现状，设计了模糊区间决策图合并算法；为了快速地选取策略，设计了基于模糊区间决策图的策略动态匹配算法。由于采用基于模糊区间决策图的策略动态匹配算法，因此能有效减少因缺乏精准威胁处置策略导致的不准确性的情况。

3) 采用邻接链表存储模糊区间决策图实现了基于区间决策图的威胁处置策略动态匹配。在实验中生成 350 组不同策略库规模的决策图。实验表明，所生成的决策图的顶点数和边数远小于原始策略数，从而大幅降低了匹配次数，提升了匹配效率。

2 相关工作

根据给定的安全事件，如何选择有效的威胁响应策略来缓解事件是一个亟待解决的问题[2]。近年来，对威胁策略匹配研究集中在 2 个方面：静态匹配[3-5]和动态匹配[6-18]。其中，静态匹配旨在根据安全事件特征自动化搜索策略，自动化程度和高效性是该类方法的研究重点；动态匹配会根据安全事件上下文动态选择给定情况下的最优策略，如何找到最优策略是该类方法的研究重点。下面，本文将详细介绍威胁策略匹配的现有研究。

静态匹配方面，Somayaji 等[3]实现了对系统调用级别的安全事件特征的自动化响应策略匹配。Toth 等[4]通过评估策略可能对网络和服务产生的负面影响来自动化选择响应措施。在自动化匹配的基础上，Marouf 等[5]提出了一种基于聚类的策略重排序算法，采用该算法可实现高效的策略静态匹配。但静态匹配方法并未考虑不同运行环境下，相同的安全事件的严重程度差异，难以灵活地响应安全事件。因此，动态策略匹配方法被提出。

动态匹配方法包括基于多因素的匹配[6-12]和基于博弈的匹配[13-18]。其中，基于多因素的匹配方面，Shameli-Sendi 等[6]提出了一种考虑安全事件上下文的动态威胁响应策略匹配方法，该方法根据系统的服务依赖图和攻防树，计算并权衡了系统的安全收益、用户和服务的安全影响、安全部署成本，实现了随攻击环境改变的威胁动态响应。Guo 等[7]提出了一种细粒度的策略匹配方法，通过考虑攻击损害、部署成本、对服务质量的负面影响和安全效益，采用三维遗传算法，动态选择威胁响应策略，此外，

该方法还考虑了策略匹配的时序。Li 等[8]研究了多步攻击的策略匹配方法，该方法将该问题建模为全局优化问题，并评估安全效益、部署成本以及对服务质量的负面影响等指标，最后采用贪心算法实现了最优策略匹配。但是上述方法只考虑了效用，并未考虑攻击者与防御者的交互行为。

为量化攻防双方行为对匹配结果的影响，基于博弈的策略匹配方法被研究。其中，Luo 等[13]研究了多级攻击策略的匹配方法，该方法将攻击者与管理员之间的交互建模为非合作零和多阶段博弈，其中攻击者为领导者，管理员为追随者，通过将两者的交替操作建模为博弈树，实现考虑攻击上下文的策略匹配。Zonouz 等[14]采用了斯坦伯格随机博弈模型实现了策略的推理和匹配，该方法考虑到实际环境中统计指标可能存在的不准确性，提出了基于模糊逻辑计算威胁的各指标值，并根据预先定义的模糊规则集推理可能的行为，最后根据推理结果选择针对当前和未来的行为的最优威胁响应策略。

3　威胁处置策略归一化描述

威胁处置策略模板是对威胁处置策略的模式进行归一化描述，威胁处置策略模板包括策略模板 ID、策略匹配条件、策略匹配动作、策略有效性，如图 1 所示。威胁处置策略匹配条件是指触发/选择威胁处置策略执行的前提条件（即威胁特征）。策略匹配动作是指满足威胁处置策略匹配条件时所触发的需要执行威胁处置操作。策略有效性是针对某威胁，预先评估的策略处置效果。

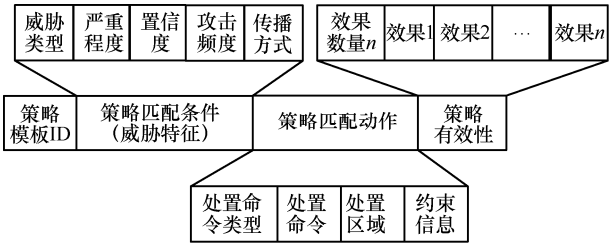

图 1　威胁处置策略模板构成示意

威胁处置策略模板库是所有威胁处置策略模板的集合。接收到报警信息后，可根据报警信息和受威胁对象的安全保障目标，从处置策略模板库中按需选取目标处置策略模板；依据报警信息等，生成威胁处置策略，为多级、多域、多类、多台对象的差异化联动响应和威胁处置的统一管理提供基础，提高了处置效率和处置效果。

3.1　策略匹配条件

在威胁处置策略匹配时，需要匹配威胁告警中的威胁特征与策略模板库中的威胁特征。威胁特征是对威胁的描述，包括威胁类型、严重程度、置信度、攻击频度、传播方式等。其中，威胁类型 THREAT_TYPE 表示预定义的威胁事件类型，如 THREAT_TYPE={DDoS, Unauthorized Access, Traffic Anomaly, FTP Trojan, … , SQL Injection}，其中，DDoS、Unauthorized Access、Traffic Anomaly、FTP Trojan、SQL Injection 分别表示分布式拒绝服务攻击、非法访问、流量异常、FTP 木马、SQL 注入攻击等。严重程度表示威胁造成危害的程度，可用 $\text{severity}=\{s\,|\,1\leqslant s\leqslant 10, s\in \mathbf{N}^{+}\}$ 表示，其数值越大表示严重程度越高。置信度表示威胁报警的可信程度，可用 $\text{confidence}=\{c\,|\,1\leqslant c\leqslant 10,\ \ c\in \mathbf{N}^{+}\}$ 表示，其数值越大，表示可信度越高。攻击频度表示每分钟上报的原始安全事件个数，$\text{attackFreq}=\{\text{af}\,|\,1\leqslant \text{af}\leqslant 10000, \text{af}\in \mathbf{N}^{+}\}$，其数值越大，表示攻击频率越高。传播方式包括攻击实体利用的恶意软件、利用的攻击工具、经由的网络类型等。

安全保障效果指该条处置策略模板实例化为处置策略并被执行后，可以满足的安全保障目标，一个处置策略模板可以实现多种安全保障效果。需注意的是，这里的安全保障效果并非百分之百满足，可以是一定程度上满足。

3.2　策略匹配动作

策略匹配动作包括处置命令类型、处置命令、处置区域、约束信息。

处置命令类型包括命令集、命令、指令和动作等。其中，命令集中包含了多种类型的命令，命令包含了多种类型的指令，指令包含了多种类型的动作。处置命令根据处置命令类型的不同而不同。

处置区域为执行处置策略的对象所在范围或空间的限定，能以逻辑方式标注，也能以物理方式标注。例如，以特定 IP 段地址标识、以唯一编号标识或以经纬度标识。

约束信息为将处置策略模板具体化成某一处置策略后对该处置策略的约束条件，包括生成时间、分发时间、执行时间、有效期、持续时间、安全等级和知悉范围等。

3.3　策略有效性

有效性为处置策略应对相应特征威胁时，达到

安全保障效果的程度，可以用[0,10]的离散整数值表示，也可以用百分比、小数表示，还可以通过采用该条处置策略应对威胁时成功的比例表示。

4 威胁处置策略模糊区间决策图

4.1 威胁处置策略的模糊区间决策图表示

采用如下形式的多元函数表示威胁处置策略

$$f: D_1 \times D_2 \times \cdots \times D_n \to \text{POLICIES} \tag{1}$$

其中，D_i 可以表示威胁类型、严重程度、置信度、攻击频度、传播方式等带隶属度的策略条件，威胁类型、传播方式为离散型变量，严重程度、置信度、攻击频度为连续型变量；POLICIES 表示所有威胁处置策略的集合。为了便于描述，用 X 表示 $D_1 \times D_2 \times \cdots \times D_n$，故式(1)可描述为 $f(X) \to \text{POLICIES}$。

威胁处置策略的选择变量可用数据区间及其隶属函数表示，对于连续型变量，数据区间是定义域内由起始值和终止值构成的一段范围，可以为开区间、闭区间或半开半闭区间，例如，严重程度数据区间可以是[1, 5]或(5, 8]；对于离散型变量，数据区间是定义域内一个或多个值组成的集合，例如，威胁类型的数据区间可以是{DDoS, 非授权访问}，数据区间形式化表示为 $I \subseteq D_i$，该数据区间是隶属度值大于 0 的区域。隶属函数表示在数据区间及其边界的隶属度分布，记为 $\mu_{D_i}(I)$，隶属度表示该威胁下选择某策略的可能性，例如，严重程度在区间[1, 5]时，选择威胁处置策略 1、策略 2、策略 3 的可能性分别为 0.3、0.4、0.6。

区间划分为定义域内互不相交的区间集合，形式化表示为 $P=\{I \mid I \subset D_i,\ \forall I_i, I_j \in P, i \neq j, I_i \cap I_j = \varnothing\}$。例如，$\text{severity} \subset \{[1,5],(5,8]\}$，严重程度变量包括 2 个互不相交的区间。

给定一个区间划分 P，变量 $x_i \in P$ 代表 $\exists I \in P, \text{s.t.} x_i \in I$。定义布尔函数 $h_{x_i}(p)$ 为

$$h_{x_i}(p)=\begin{cases}0, x_i \notin P\\ 1, x_i \in P\end{cases} \tag{2}$$

当 $x_i \in P$ 时，式(1)与 x_i 独立，即对于区间划分 P 中的任何 x_i，会得到相同的威胁处置策略，如式(3)所示。

$$\forall a,b \in P, f\left(X \middle| x_{i:=a}\right) = f\left(X \middle| x_{i:=b}\right) \tag{3}$$

因此，偏函数表示为

$$f_{x_i}^{P} := f\left(x_1,\cdots,x_{i-1},b,x_{i+1},\cdots,x_n\right)\Big|_{\forall b \in P} \tag{4}$$

例如，函数 f 关于变量 severity 在区间 $P=\{[1,8]\}$ 独立。

$$f\left(\text{ttype, severity, confidence, attackFreq, transType}\right) =$$
$$\left(\text{ttype}=\{\text{DDoS}\}\right) \wedge (1 \leqslant \text{severity} \leqslant 8) \wedge$$
$$(8 \leqslant \text{confidence} \leqslant 10) \wedge (1\,000 \leqslant \text{attackFreq} \leqslant 10\,000) \wedge$$
$$\left(\text{transType}=\{\text{有线网络}\}\right)$$

因此，偏函数可表示为

$$f_{\text{severity}^{[1,8]}} = \left(\text{ttype}=\{\text{DDoS}\}\right) \wedge (8 \leqslant \text{confidence} \leqslant 10) \wedge$$
$$(1\,000 \leqslant \text{attackFreq} \leqslant 10\,000) \wedge \left(\text{transType}=\{\text{有线网络}\}\right)$$

给定一个域 D_i，如果区间集合 $P(D_i)=\{P_1,P_2,\cdots,P_{d_i}\}$ 满足 $D_i = \cup_{P \in P(D_i)} (\cup_{I \in P} I)$，则称域 D_i 被区间集合覆盖；如果区间集合 $P(D_i)$ 中没有共同区间，则说明其不相交，即 $\forall i,j \in [1,d_i], i \neq j, P_i \cap P_j = \varnothing$。

根据香农分解，函数 f 可被分解为对于任意变量 x_i 在不相交覆盖的区间集合 $P(D_i)$ 上的偏函数。

$$f(x) = \bigvee_{P \in P(D_i)} h_{x_i}(p) \wedge f_{x_i}^{P} \tag{5}$$

因此，可以将函数 f 形式化为具有如下性质的决策图，如图 2 所示。

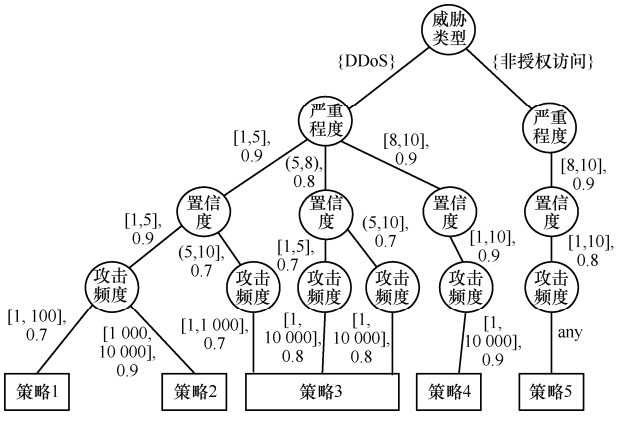

图 2 威胁处置策略的模糊区间决策图

1) 图 G 是一个有根、有向无环图，其顶点集包括 2 类节点：内部节点和叶子节点。内部节点表示威胁处置策略匹配条件，叶子节点表示策略匹配动作（用策略 ID 标识）。

2) 内部节点 $v_{x_i} \in V$ 表示函数 f 在定义域 D_i 中的变量 x_i，外出边 $e_i \in E$ 表示在区间 $P_{x_i} \in P(D_i)$ 上如式(2)所示的布尔函数，表示变量 x_i 在区间 P_{x_i} 中

有值的一个断言，并且其隶属度为 $\mu(x_i)$。

3) 每一个顶点 v_{x_i} 的子图都是式(5)中的偏函数 $f_{x_i}^{P}$。

4.2 带模糊算子的区间决策图构造

威胁处置策略模糊区间决策图（FIDD, fuzzy interval decision diagram）定义如下。

1) FIDD 中的节点 m 用四元组 (index,val,pval,C) 表示，其中，index 表示节点的序号；val 表示该节点的威胁特征值，当节点为决策图的内部节点时，威胁特征值可以是威胁类型、严重程度、置信度、攻击频度等，具体取值为该节点连接其下级节点边的区间划分的集合 $\cup_{P\in P(D_i)}$，即 $m.\text{val}=\vee\{p\,|\,(p,c)\in m.C\}$，当节点为叶子节点时，val 为空；pval 表示该节点所能选择策略的 ID 集合；C 表示四元组 (c,p,μ,λ) 的集合，c 表示节点 m 的下级节点，μ 表示隶属度函数，λ 表示隶属度阈值，p 表示节点 m 到其下级节点 c 的隶属度函数大于 0 的区间划分，用于从模糊区间中确定清晰集。

2) FIDD 中的边用 (p,μ,λ) 表示。从根节点开始，如果节点 x_i 的隶属度大于阈值 λ，则选择 (p,μ,λ) 这条边。

威胁处置策略构造算法。从策略匹配条件和策略匹配动作等策略元素生成一条策略的决策图的算法，如算法 1 所示，算法的核心思想是将威胁类型、严重程度、置信度、攻击频度、传播方式等策略匹配条件按照节点序号由小到大排序，作为 FIDD 的内部节点，将策略匹配动作作为 FIDD 的叶子节点，整体存储为单链表结构。

算法 1　威胁处置策略模糊区间决策图 policyBuild$(e_1,e_2,\cdots,e_k,\text{ac})$

输入　策略元素 $E=\{e_1,e_2,\cdots,e_k,\text{ac}\}$

输出　用 FIDD 表示的策略：$m=e_1\wedge e_2\wedge\cdots\wedge e_k\wedge\text{ac}$

1) for each $e_i\in e$ do

2) $I\leftarrow(\text{valueList}[e_i.x]=\text{null})?(-\infty,+\infty):\text{valueList}[e_i.\text{val}]$

3) end for

4) root $\leftarrow$ null,tail $\leftarrow$ null

5) for each $x\in\text{sort}(\text{valueList.index})$ do

6) $v\leftarrow(e_i.\text{index},\text{valueList}[e_i.x],C:=(\text{valueList}[e_i.\text{val}],\varnothing))$

7) if (root==null) then

8) root $\leftarrow v$,tail $\leftarrow v$

9) else

10) tail.addchild(v)

11) tail $\leftarrow v$

12) end if

13) leaf $\leftarrow$ (0,"policy",ac.pval,$\varnothing$)tail.addchild(leaf)

14) end for

15) return root

威胁处置策略合并算法如算法 2 所示，其核心思想是采用深度优先搜索思想，对已有的和待合并的 2 个威胁处置策略决策图进行递归合并，得到新的威胁处置策略决策图，在节点两两合并过程中，分以下 3 种情况处理。

算法 2　威胁处置策略合并算法 policy Merge (m_1,m_2)

输入　2 个决策图根节点 m_1 和 m_2

输出　$m_1\vee m_2$

1) if $(<m_1,m_2,m>\in\text{computeList})$ then

2) return m

3) end if

4) if (one of m_1 and m_2 is NULL) then

5) $m_{\text{notnull}}=(m_1==\text{null})?m_2:m_1$

6) $m\leftarrow(m_{\text{notnull}}.\text{index},m_{\text{notnull}}.\text{val},m_{\text{notnull}}.\text{pval},C:=\varnothing)$

7) for each $(c,p,\mu,\lambda)\in m_{\text{notnull}}.C$ do

8) $c'\leftarrow c\vee\text{NULL}$

9) $m.C\leftarrow m.C\cup\{(c',m_{\text{notnull}}.p,m_{\text{notnull}}\mu,m_{\text{notnull}}.\lambda)\}$

10) end for

11) else if (isLeaf(m_1) & &isLeaf(m_2)) then

12) value $=m_1.\text{val}\cup m_2.\text{val}$

13) $m\leftarrow(0,\text{value},\varnothing)$

14) if (value $\notin$ leafList) then

15) leafList $\leftarrow$ leafList $\cup$ value

16) end if

17) else if $(m_1.\text{index}==m_2.\text{index})$ then

18) $\text{val}=m_1.\text{val}\vee m_2.\text{val}$

19) pval=$m_1.\text{pval}\cup m_2.\text{pval}$

20) $m\leftarrow(m_1.\text{index},\text{val},\text{pval},C:=\varnothing)$

21) for (interval $I\in$ val)do

22) for (interval $I_i\in I$)do

23) if $(m_1.\text{pval} == m_2.\text{pval})$ then

24) $c \leftarrow m_1.C[I_i] \vee m_2.C[I_i]$

25) $p \leftarrow \{I_i\}$

26) $\mu \leftarrow m_1.\mu \wedge^* m_2.\mu$

27) $\lambda \leftarrow \max\{m_1.\lambda, m_2.\lambda\}$

28) $m.C \leftarrow m.C \cup \{(c, p, \mu, \lambda)\}$

29) else

30) $c \leftarrow m_1.C[I_i] \vee m_2.C[I_i]$

31) $p \leftarrow \{I_i\}$

32) $\mu \leftarrow m_1.\mu(I_i) +^* m_2.\mu(I_i)$

33) $\lambda \leftarrow m_1.\lambda +^* m_2.\lambda$

34) $m.C \leftarrow m.C \cup \{(c, p, \mu, \lambda)\}$

35) end if

36) end for

37) end for

38) else

39) $l = \min(m_1.\text{index}, m_2.\text{index})$

40) $h = \max(m_1.\text{index}, m_2.\text{index})$

41) $m \leftarrow (m_1.\text{index}, m_1.\text{var}, C \coloneqq \varnothing)$

42) for each $(c, p, \mu, \lambda) \in m_1.C$ do

43) $c' \leftarrow c \vee m_h$

44) $m.C \leftarrow C \cup \{(m_l.p.c', m_l.\mu, m_l.\lambda)\}$

45) end for

46) $p' = \perp \ominus m_l.P$

47) $m.C = m.C \cup \{(p', m_h)\}$

48) end if

49) if $(<\text{index}, C, m> \notin \text{uniqueList})$ then

50) $\text{uniqueList} \leftarrow \text{uniqueList} \cup <\text{index}, C, m>$

51) $\text{computeList} \leftarrow \text{computeList} \cup <m_1, m_2, m>$

52) end if

53) return m

1) 均为叶子节点（步骤 11)~步骤 16)）。合并 2 个叶子节点（即表示威胁处置策略动作的节点）代表的策略集合作为最终的叶子节点。

2) 均为内部节点且节点序号相等（步骤 17)~步骤 37)）$\wedge^*$模糊算子（4.3 节具体介绍）的计算结果进行区间合并和递归计算。此种情况是 2 个节点所选择的策略集合不同，则根据$+^*$算子对 2 个节点的隶属度函数、隶属度阈值进行记录和递归运算。

3) 均为内部节点且节点序号不等（步骤 38)~步骤 48)）。将序号低的节点的下级节点与序号高的节点进行递归调用运算。

为降低算法复杂度，引入计算表存储结构，确保任一节点对至多一次计算（步骤 1)~步骤 3)、步骤 51)）。为避免重复生成的顶点不被多次添加到决策图中，在合并过程中同步应用化简规则进行化简（步骤 49)~步骤 52)）。

如算法 3 所示，通过依次调用威胁处置策略构造算法（算法 1）和威胁处置策略合并算法（算法 2），以渐进式方式生成全局的威胁处置策略决策图。

算法 3 威胁处置策略的模糊区间决策图生成算法 FIDDGeneration($\mathbb{E}$)

输入 策略集合 $\mathbb{E}$，$\mathbb{E} = \{E_1, E_2, \cdots, E_n\}$，$E = \{e_1, e_2, \cdots, e_k, \text{ac}\}$

输出 多类型区间决策图 $t = m_1 \vee m_2 \vee \cdots \vee m_k$

1) $t \leftarrow \varnothing$

2) for each policy element $E \in \mathbb{E}$ do

3) $m_i \leftarrow \text{policyBuild}(e_1, e_2, \cdots, e_k, \text{ac})$

4) $t \leftarrow \text{policyMerge}(t, m_i)$

5) end for

6) return t

图 3 为威胁处置策略模板库中待合并的 2 个策略，P_1 决策图表示威胁类型为 DDoS 攻击、严重程度为[1, 5]、置信度为[1, 5]、攻击频率为[1, 1 000]的情况下选择策略 1，决策图中的边还记录了对应的策略匹配条件的隶属度函数，P_2 的决策图同理。P_1 和 P_2 通过递归调用策略合并算法，逐节点自底向上进行合并，生成的模糊区间决策图如图 4 所示，其中，威胁类型和严重程度节点在 2 个策略中的区间值相同，所以各自合并为一个节点；置信度节点在进行合并时，首先对[1, 5]和[3, 10]进行无覆盖的区间划分，划分的结果为 3 个区间{[1, 3], [3, 5], [5, 10]}，然后在每个区间下，依然递归调用策略合并函数，对攻击频率节点进行区间划分，在区间划分后应用模糊算子对该区间下的隶属度函数进行合并，最终得到如图 4 所示的决策图，叶子节点的策略匹配动作包括仅执行策略 1、仅执行策略 2、执行策略 1 和策略 2 这 3 种情况。

4.3 面向威胁处置的模糊算子设计

4.2 节给出了带模糊算子的区间决策图构造，本节将讨论如何设置适用于威胁处置的模糊算子。

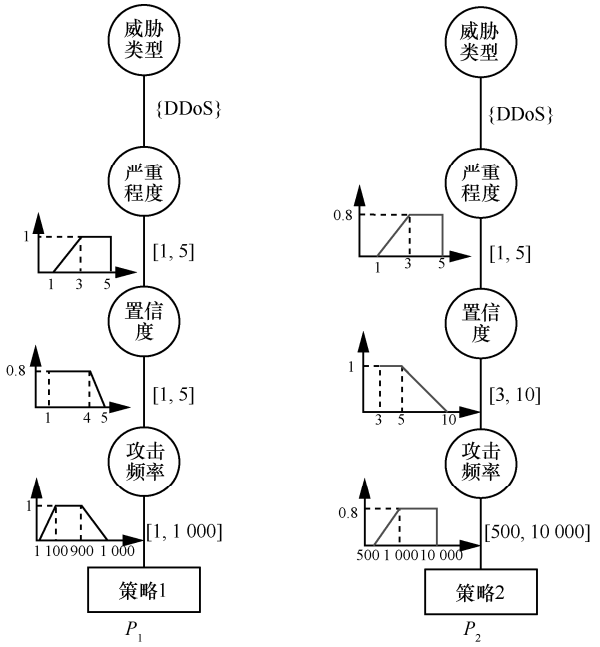

图 3　合并前的威胁处置策略 P_1 和 P_2

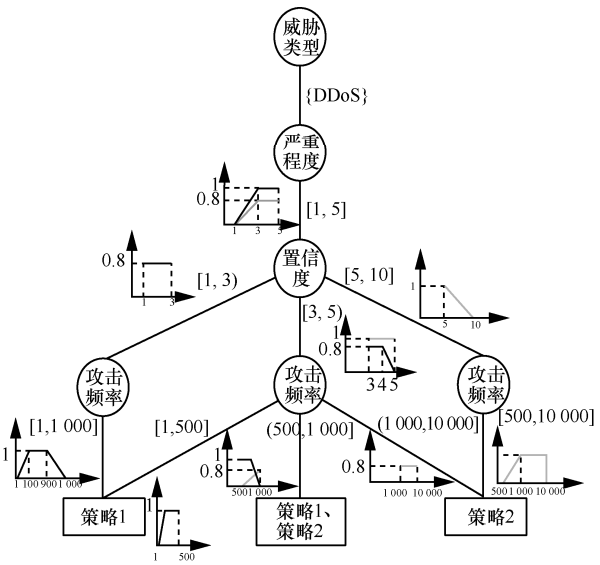

图 4　合并后的威胁处置策略

威胁处置策略决策图中的边代表模糊区间或模糊集合，如图 5 和图 6 所示。图 5 中表示在攻击频率区间[0, 100]选择策略 1 进行威胁处置的隶属度为 1，在区间(100, 105]中随着攻击频度增加，选择策略 1 的隶属度减少，呈直角梯形分布。类似地，图 6 给出了选择策略 2 的隶属度。

当合并 2 个决策图中时，需要处理合并后所选取策略的隶属度，如果这 2 个决策图最终选择的策略相同，则采用 MAX 算子，记为 $\wedge^*$ 算子，即当模糊区间存在交叠时将该区间的最大隶属度作为最终隶属度，即 $\wedge^*(a,b)=\max(a,b)$。采用该隶属函数原因是将最可靠的策略作为最终处置策略。图 7 给出了图 5 和图 6 所代表的模糊分布在选择相同策略时，经过 $\wedge^*$ 算子运算后，得到模糊区间的模糊分布。

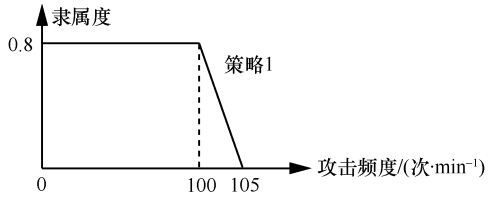

图 5　攻击频率模糊区间 P_1 的模糊分布

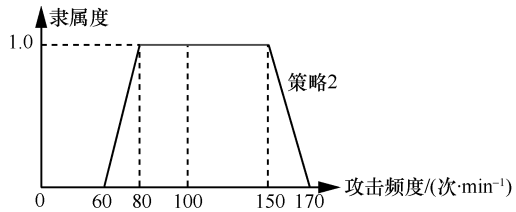

图 6　攻击频率模糊区间 P_2 的模糊分布

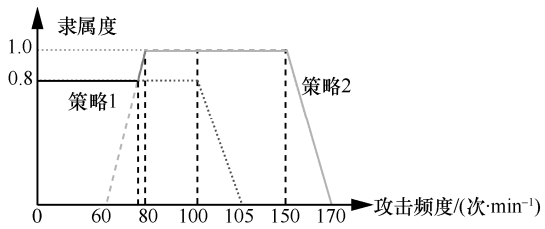

图 7　攻击频率模糊区间合并后的模糊分布 $P_1 \wedge^* P_2$

如果这 2 个决策图最终选择的策略不同，采用 MERGE 算子，记为 $+^*$ 算子，即当模糊区间存在交叠时将该区间的多个隶属度函数分段记录，即 $+^*(a,b)=\bigcup(a,b)$。采用该隶属度函数的原因是可同时选择出多条策略。图 8 给出了图 5 和图 6 所代表的模糊分布在选择不同策略时，经过 $+^*$ 算子运算后，得到的模糊区间的模糊分布，灰色粗实线表示相交区间的模糊分布。

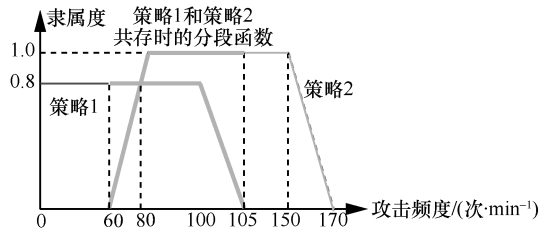

图 8　攻击频率模糊区间合并后的模糊分布 $P_1 +^* P_2$

4.4　基于模糊区间决策图的策略动态匹配

在收到新报警后，将报警中的威胁类型、严重

程度、置信度、攻击频度、传播方式等威胁特征，在威胁处置策略模糊区间决策图中自顶向下进行匹配，具体地，如算法 4 所示，将报警中的各威胁特征从模糊区间决策图的根节点开始，在具有相同序号的节点处进行匹配（步骤 4)），若报警中威胁特征的隶属度值大于模糊区间决策图中的阈值，则选择该下层节点继续匹配（步骤 6)~步骤 7)），直到匹配至叶子节点，得到策略动作（步骤 1)~步骤 3)）。需要注意的是，在上述过程中，一条报警信息在匹配到模糊区间决策图的叶子节点时，由于区间的模糊性，可得到多个策略动作，此时需要根据匹配路径中选择出的各策略的隶属度阈值进行综合排序，得到最终的策略执行顺序。例如，依照策略的隶属度大小，首先执行隶属度最大的威胁处置策略，若该威胁处置策略的效果低于预定值，再选取隶属度次之的威胁处置策略；重复执行，直到达到预定的威胁处置效果或所选的威胁处置策略执行完毕。

算法 4 基于模糊区间决策图的策略动态匹配算法 $\text{match}(m,t)$

输入 决策图根节点 t，威胁报警根节点 m

输出 威胁处置策略列表

1) if $(\text{isLeaf}(m))$ then

2) return t.pval

3) end if

4) if $(t.\text{index} == m.\text{index})$ then

5) $P = m.\text{val} \vee t.\text{val}$

6) for $((\text{interval}\, I \in P) \wedge (\mu(I) > \lambda))$ do

7) return $\text{match}(m.C[I], t.C[I])$

8) end for

9) else

10) $l = \min(m.\text{index}, t.\text{index})$

11) $h = \max(m.\text{index}, t.\text{index})$

12) for $(\text{each}(p,c) \in m_l.C) \wedge (\wedge(\mu(I) > \lambda))$ do

13) return $\text{match}(c, m_h)$

14) end for

15)end if

算法 4 将报警信息中的威胁特征在决策图中进行深度优先搜索和匹配，其时间复杂度为 $O(|V|+|E|)$，其中，$|V|$ 为决策图的顶点数，$|E|$ 为决策图的边数。对比策略匹配时常用的顺序匹配算法，即报警信息与策略库中的策略匹配条件进行逐条比较，其时间复杂度为 $O(nm)$，其中，n 为策略库中的策略数，m 为每个策略匹配条件中的威胁特征数量。

使用模糊区间决策图构建策略库，相同区间划分的威胁特征进行了合并，使生成的决策图的顶点数和边数远小于原始策略的数量，减少了策略条件匹配时的求交操作次数，降低了策略匹配的时间复杂度。

5 实验

使用C语言实现了模糊区间决策图的构造算法和基于模糊区间决策图的策略匹配算法，实验平台为超云服务器（R7410 G11），CPU 型号为 Intel(R)Xeon(R)Gold 6234 CPU@3.30 GHz，内存大小为 96 GB，操作系统为 CentOS7，内核版本为 3.10.0-1062.el7.x86_64，GCC 版本为 4.8.5，采用邻接链表存储模糊区间决策图。

为模拟不同策略库规模对于策略匹配的时间影响，本文设置了 50 组威胁类型、严重程度、置信度、攻击频度的可选值，以及根据策略库规模与策略动作的数量关系划分了 7 组（非常高（VH）、高（H）、偏高（MH）、中等（M）、偏低（ML）、低（L）、很低（VL））策略相似度，其中，非常高表示多个策略匹配条件选择相同的策略动作，很低表示每个策略匹配条件唯一地选择了一个策略动作，共计生成了 350 组不同策略库规模的决策图。

为验证基于模糊区间决策图的策略模糊匹配的高效性，本文选择传统的顺序匹配方案，在匹配时间和内存消耗这 2 个方面进行比较，共设置 8 组实验，其中，7 组是不同策略相似度的基于区间决策图的匹配方案，一组是顺序匹配方案。仅设置一组顺序匹配方案的原因在于原始的策略中，每个策略匹配条件仅会选择一个策略动作，策略是否相似对于顺序匹配的时间和内存消耗是没有影响的。

基于区间决策图的匹配方案与顺序匹配方案的运算时间的部分结果如表 1 所示，随策略数量的变化趋势如图 9 所示。从图 9 可以看出，7 组基于区间决策图的匹配方案明显低于顺序匹配方案，这验证了基于模糊区间决策图进行策略匹配的有效性和效率。基于决策图的匹配方案中策略相似度越高，匹配时间越短。

基于区间决策图的匹配方案与顺序匹配方案的内存消耗的部分结果如表 2 所示，随策略数量的变化趋势如图 10 所示。从图 10 可以看出，当策略相似度很低时，基于区间决策图的匹配方案的内存

消耗略低于顺序匹配方案，基于区间决策图的其他 6 组实验的内存消耗远低于顺序匹配方案，其原因是决策图高效的合并策略匹配条件和策略动作，减少了相同策略元素的重复存储，使占用的内存空间远低于扁平存储结构的顺序匹配方案。

表 1　基于区间决策图的匹配方案与顺序匹配方案的运算时间的部分结果

策略数量/个	FIDD-H/μs	FIDD-M/μs	FIDD-L/μs	顺序匹配/μs
90	4.24	4.66	6.38	14.28
160	1.08	1.30	1.74	13.96
640	0.94	0.96	0.92	49.62
2 700	3.60	10.52	38.32	215.64
4 860	6.02	18.66	71.82	382.58
12 800	14.70	47.66	179.70	1 025.64
40 000	41.58	147.98	571.68	3 393.10
47 500	46.40	177.56	699.02	4 281.26
95 040	92.60	356.18	1 500.16	10 815.56
108 000	102.24	400.28	1 704.38	12 626.86
148 470	139.62	539.76	2 427.04	18 364.28
230 720	215.54	907.20	4 003.90	30 073.94
336 960	312.54	1 303.56	5 960.52	45 423.00
425 250	396.82	1 742.00	7 378.42	57 006.88
530 000	510.78	2 136.80	9 159.92	71 390.38

表 2　基于区间决策图的匹配方案与顺序匹配方案的内存消耗的部分结果

策略数量/个	FIDD-H/B	FIDD-M/B	FIDD-L/B	顺序匹配/B
90	1 090 519.04	1 089 781.76	1 089 699.84	1 093 795.84
160	1 090 519.04	1 089 781.76	1 089 699.84	1 360 035.84
640	1 356 759.04	1 356 021.76	1 355 939.84	1 630 371.84
2 700	1 627 095.04	1 626 357.76	1 626 275.84	2 711 715.84
4 860	2 167 767.04	2 167 029.76	1 896 611.84	3 823 124.48
12 800	2 438 103.04	2 437 365.76	2 437 283.84	8 059 699.20
40 000	4 330 455.04	3 789 045.76	3 788 963.84	22 554 542.08
47 500	5 411 799.04	4 600 053.76	4 329 635.84	26 688 634.88
95 040	8 115 159.04	6 222 069.76	5 681 315.84	52 059 340.80
108 000	9 737 175.04	7 033 077.76	6 221 987.84	59 100 200.96
148 470	11 088 855.04	7 844 085.76	6 762 659.84	80 811 950.08
230 720	14 062 551.04	9 195 765.76	7 844 003.84	124 720 250.88
336 960	17 576 919.04	10 547 445.76	8 925 347.84	181 701 836.80
425 250	20 820 951.04	11 899 125.76	9 736 355.84	229 109 841.92
530 000	24 605 655.04	13 250 805.76	10 547 363.84	285 234 626.56

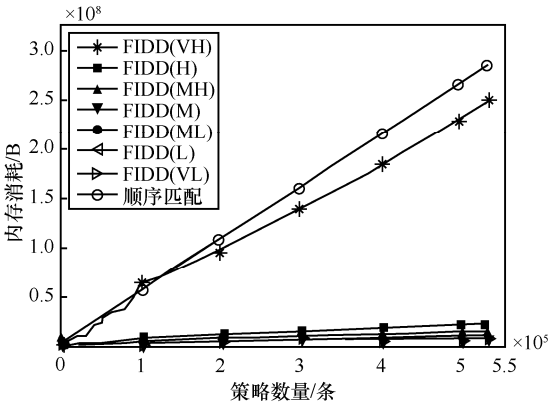

图 9　策略匹配时间随策略数量的变化趋势

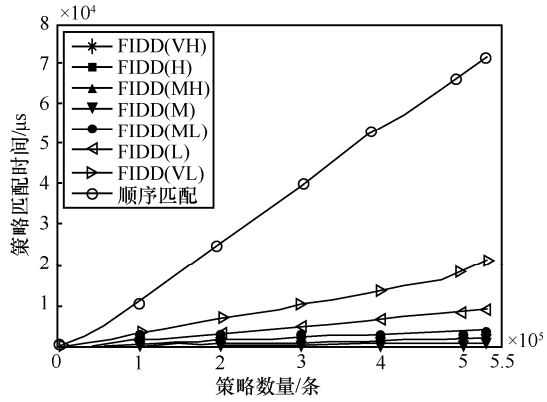

图 10　策略匹配的内存消耗随策略数量的变化趋势

策略匹配时间随决策图顶点数和边数的变化趋势如图 11 所示。图 11 验证了基于模糊区间决策的策略匹配算法的时间复杂度为 $O(|V|+|E|)$。对于稀疏图，最小边数为 $|E|=|V|-1$；对于稠密图，最大边数为 $|E|=(|V|\times(|V|-1))/2$，结合表 1 可看出，本文的模糊区间决策图为稀疏图。

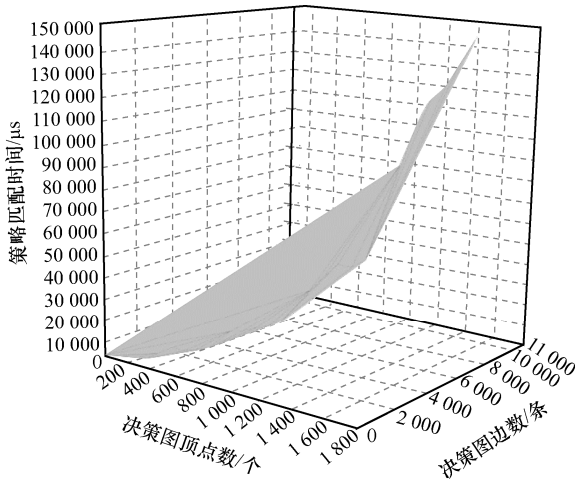

图 11　决策图顶点数、边数与策略匹配时间的变化趋势

通过不同策略库规模下生成决策图，并进行策略匹配的实验可知，采用模糊区间决策图组织策略库，形成了对威胁特征状态空间的隐式表示，使生成的决策图的顶点数和边数远小于原始策略数，减少了策略条件匹配次数，较顺序比较各策略条件的威胁特征，具有数据级优势。

6 结束语

如何快速准确地选取威胁处置策略是确保网络安全中的一项重要挑战，针对该问题，本文提出了基于区间决策图的威胁处置策略快速匹配方法。首先设计了威胁处置策略归一化描述方法，定义了模糊化的威胁处置策略；依据威胁类型、严重程度、置信度、攻击频度、传播方式，设计了单威胁策略的模糊构造算法；针对多条模糊处置策略，设计了模糊区间决策图合并算法和策略快速匹配算法，实验表明了所提方法的有效性。

虽然所提方法能有效地选取威胁处置策略，但其准确度依赖于隶属度函数；在实际应用中不同威胁特征下采用威胁处置策略的隶属度不同，未来需要依据历史处理策略的有效性，采用机器学习等方法准确地获取不同威胁特征下威胁处置策略隶属度。

参考文献：

[1] SHAMELI-SENDI A, CHERIET M, HAMOU-LHADJ A. Taxonomy of intrusion risk assessment and response system[J]. Computers & Security, 2014, 45: 1-16.

[2] NESPOLI P, PAPAMARTZIVANOS D, GÓMEZ MÁRMOL F, et al. Optimal countermeasures selection against cyber attacks: a comprehensive survey on reaction frameworks[J]. IEEE Communications Surveys & Tutorials, 2018, 20(2): 1361-1396.

[3] SOMAYAJI A, FORREST S. Automated response using system-call delay[C]//Usenix Security Symposium. Berkeley: USENIX Association, 2000: 185-197.

[4] TOTH T, KRUEGEL C. Evaluating the impact of automated intrusion response mechanisms[C]//18th Annual Computer Security Applications Conference. Piscataway: IEEE Press, 2002: 301-310.

[5] MAROUF S, SHEHAB M, SQUICCIARINI A, et al. Adaptive reordering and clustering-based framework for efficient XACML policy evaluation[J]. IEEE Transactions on Services Computing, 2011, 4(4): 300-313.

[6] SHAMELI-SENDI A, LOUAFI H, HE W B, et al. Dynamic optimal countermeasure selection for intrusion response system[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(5): 755-770.

[7] GUO Y C, ZHANG H, LI Z F, et al. Decision-making for intrusion response: which, where, in what order, and how long?[C]//2020 IEEE International Conference on Communications. Piscataway: IEEE Press, 2020: 1-6.

[8] LI F H, LI Y J, LENG S Y, et al. Dynamic countermeasures selection for multi-path attacks[J]. Computers & Security, 2020, 97: 101927.

[9] ROY A, KIM D S, TRIVEDI K S. Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees[C]//IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE Press, 2012: 1-12.

[10] HUGHES K, MCLAUGHLIN K, SEZER S. Dynamic countermeasure knowledge for intrusion response systems[C]//2020 31st Irish Signals and Systems Conference. Piscataway: IEEE Press, 2020: 1-6.

[11] LI X, ZHOU C J, TIAN Y C, et al. A dynamic decision-making approach for intrusion response in industrial control systems[J]. IEEE Transactions on Industrial Informatics, 2019, 15(5): 2544-2554.

[12] KOTENKO I, DOYNIKOVA E. Selection of countermeasures against network attacks based on dynamical calculation of security metrics[J]. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 2018, 15(2): 181-204.

[13] LUO Y, SZIDAROVSZKY F, AL-NASHIF Y, et al. A game theory based risk and impact analysis method for intrusion defense systems[C]//2009 IEEE/ACS International Conference on Computer Systems and Applications. Piscataway: IEEE Press, 2009: 975-982.

[14] ZONOUZ S A, KHURANA H, SANDERS W H, et al. RRE: a game-theoretic intrusion response and recovery engine[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 395-406.

[15] LIANG X N, XIAO Y. Game theory for network security[J]. IEEE Communications Surveys & Tutorials, 2013, 15(1): 472-486.

[16] MANSHAEI M H, ZHU Q Y, ALPCAN T, et al. Game theory meets network security and privacy[J]. ACM Computing Surveys, 2013, 45(3): 1-39.

[17] MAHARJAN S, ZHU Q Y, ZHANG Y, et al. Dependable demand response management in the smart grid: a stackelberg game approach[J]. IEEE Transactions on Smart Grid, 2013, 4(1): 120-132.

[18] KIENNERT C, ISMAIL Z, DEBAR H, et al. A survey on game-theoretic approaches for intrusion detection and response optimization[J]. ACM Computing Surveys, 2019, 51(5): 1-31.

[作者简介]

张玲翠（1986− ），女，河北故城人，中国科学院信息工程研究所博士生、高级工程师，主要研究方向为网络与系统安全。

李凤华（1966− ），男，湖北浠水人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为网络与系统安全、信息保护、隐私计算。

房梁（1989− ），男，山西太原人，博士，中国科学院信息工程研究所副研究员，主要研究方向为网络安全、访问控制。

郭云川（1977− ），男，四川营山人，博士，中国科学院信息工程研究所教授级高级工程师、博士生导师，主要研究方向为访问控制、网络安全。

李子孚（1992− ），女，内蒙古赤峰人，博士，中国科学院信息工程研究所工程师，主要研究方向为网络与系统安全、访问控制。