复杂网络环境下面向威胁监测的采集策略精化方法

李凤华[1,2,3]，李子孚[1,2]，李凌[1,2]，张铭[4]，耿魁[1]，郭云川[1]

（1. 中国科学院信息工程研究所，北京 100093；2. 中国科学院大学网络空间安全学院，北京 100049；
3. 通信网信息传输与分发技术重点实验室，河北 石家庄 050081；
4. 西安电子科技大学网络与信息安全学院，陕西 西安 710071）

摘 要：个性化采集策略是有效监测复杂网络环境面临的威胁的必要条件之一，然而安全需求和威胁类型差异等导致难以有效生成个性化的采集策略。针对上述问题，设计了面向威胁监测的采集策略自动精化方法。首先，提出了采集策略层次模型；然后，将威胁类型到采集项的精化转化为采集收益和采集成本平衡的非线性优化问题，并利用遗传算法进行求解；最后，通过模拟实验，验证可根据高层监测需求自动生成采集方案。

关键词：数据采集；威胁监测；策略精化；混合优化

中图分类号：TP302

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019096

Collection policy refining method for threat monitoring in complex network environment

LI Fenghua[1,2,3], LI Zifu[1,2], LI Ling[1,2], ZHANG Ming[4], GENG Kui[1], GUO Yunchuan[1]

1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
3. Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081, China
4. School of Cyber Engineering, Xidian University, Xi'an 710071, China

Abstract: Personalized collect policy is one of the necessary conditions for effectively monitoring threats in the complex network environment. However, differences in security requirements and threat types make it difficult to effectively generate personalized collect policy. To address the above problem, a collection policy automatic refinement method was designed. Firstly, a hierarchical model of collection policy was proposed. Then, by transforming the policy refinement into a nonlinear optimization problem, a genetic algorithm was designed to balance between collection revenue and collection cost. Finally, simulation experiments verify that according to the requirements of high-level monitoring, the acquisition scheme can be automatically generated.

Key words: data collection, threat monitoring, policy refinement, hybrid optimization

1 引言

以专用网络、天地一体化网络和物联网为代表的复杂网络[1]面临严峻的安全威胁。中国互联网络信息中心于 2018 年 7 月发布的《第 42 次中国互联网络发展状况统计报告》显示，2018 年上半年我国境内被植入后门网站数量累计 16 210 个，收集整理的信息系统安全漏洞累计 7 748 个，接到网络安全

收稿日期：2018−11−06；**修回日期**：2019−03−23

通信作者：郭云川，guoyunchuan@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.61672515）；中国科学院大学生创新实践训练计划基金资助项目

Foundation Items: The National Key Research and Development Program of China (No.2016YFB0801001), The National Natural Science Foundation of China (No.61672515), Innovative Practice Project of College Students in Chinese Academy of Sciences

事件报告累计 54 190 件。为应对上述威胁，威胁监测越来越受到人们的重视，在 Crowd Research 发布的 2017 年威胁的监测、检测和响应报告中，67%被调查的信息安全专业人员认为威胁监测是该领域的首要任务，超过威胁分析的 44%。威胁监测通过在网络设备中部署采集策略，实现对数据的有效采集，为威胁分析提供数据基础。

然而，由于复杂网络环境的如下特点，使通过人工方式制定采集策略难以为继：1)节点数量多，依据 Gartner 发布的数据，2020 年全球联网设备数量将达 260 亿台，各类设备的大规模互联，使网络中的节点数量呈爆发式增长趋势；2)节点处理能力差异，网络节点部署于不同位置，且功能需求不同，导致节点处理能力差异大，由于上述 2 个特点使人工管理网络成本高，因此需要自动化、统一管理；3)安全需求变化大，天地一体化网络、移动互联网等网络由于节点随遇接入导致其拓扑结构高变，使安全威胁蔓延，安全需求动态变化，所以需要对采集策略进行动态调整。

因此，自动化设置采集策略的方案被提出，但依然存在如下问题：1)在采集内容方面，大多使用查找表的方式确定采集内容，缺乏动态调整的特性；2)在采集频率方面，大多采用等时间间隔、周期性的采集方式，没有考虑重复采集数据的相似度，部分采集方案[2-4]考虑采集数据冗余的问题，提出了一些动态改变采集频率的方法，但是其应用场景和采集数据的类型较为单一；3)在采集资源消耗方面，未充分考虑采集设备当前的资源水平，不适用于复杂网络环境中设备的计算和存储资源受限的场景。

为了解决现有工作的不足，本文将安全策略分为高层监测需求（TDR, top detection requirement）、中层采集策略（ICP, intermediate collection policy）和低层设备指令（LDC, low device command）这 3 层结构，形成采集策略层次模型（CPHM, collection policy hierarchy model）。基于该模型，提出了基于采集成本和收益平衡的策略精化方法，将以威胁监测为目标的高层监测需求，综合考虑采集贡献度和资源消耗，细化为低层设备指令，在降低采集数据量的同时保证采集数据的精确度，做到极小量按需精准采集。本文的主要贡献如下。

1) 提出了面向威胁监测的采集策略精化系统模型，设计网络环境和策略的 3 层结构，以及策略精化规则和策略模板。

2) 针对高层监测需求的威胁类型到中层采集策略的采集项和采集频率的精化问题，通过对采集贡献度和采集资源消耗的量化，设计目标函数，通过遗传算法求解，得到与待采集设备资源水平对应的采集项和采集频率。

3) 对所提方法进行了实验验证，实验结果表明，本文方法可针对不同威胁类型生成个性化的采集策略，并在待采集设备的资源水平降低时，自适应地减少采集项，降低采集频率。

2 相关工作

现有的策略精化[5]大多集中在访问控制、网络服务质量管理、能源优化等方面。在访问控制方面，主要对访问主体、客体和操作权限进行细化；在网络服务质量管理方面，通过将服务质量细分为一系列服务级别目标（SLO, service level objective），通过实现每个 SLO 以实现整体的服务质量；在能源优化方面，采用调节链路速率、关闭或睡眠负荷低的设备的方式以减少能源消耗。

根据策略精化使用的技术，可以将其分为基于规则的策略精化、基于历史数据的策略精化和基于逻辑的策略精化等。

2.1 基于规则的策略精化

基于规则的策略精化的核心思想是使用表来关联用户、应用和服务，通过查找表的方法建立精化关系，适用于服务质量（QoS, quality of service）管理等。

文献[6]提出的策略精化引擎（POWER, policy wizard engine for refinement）是首次采用基于策略模板的精化技术，其将抽象策略中的主体、客体和权限根据上下文信息等系统模型中定义的组织实体映射为实现层的主体、客体和操作权限。文献[7-8]将系统进行分层，通过规则对各层间的策略进行精化。其中，文献[7]利用上述系统分层思想开发了 MoBaSeC（model based service configuration）工具，该工具支持交互式图形建模、自动模型分析和策略精化生成配置文件，文献[8]在文献[7]的基础上，定义了一系列公理，以获取策略规则中隐含的假设，验证了策略精化的一致性和完备性，同时形式化表示模型到现实世界的映射。

文献[9-10]使用 ECA（event condition action）范式定义策略。其中，文献[9]提出了一个在领域、

语言、抽象级别等方面通用的 ECA 策略自动化精化的方法，并应用在通信系统中来管理系统的行为，采用领域建模以定义高层中的领域术语，包括领域、关系、属性等；策略建模包括策略中的事件、条件和行为；领域和策略的连接建模给出领域术语与策略之间的关系，然后定义高层策略和低层策略之间的精化规则，最后基于规则实现自动化的策略精化。文献[10]使用 ECA 范式制定规则，在系统运行时，满足策略执行条件后，将策略元素进行实例化。

文献[11-13]针对策略元素和规则进行研究。其中，文献[11]提出了提取安全策略规则的方法，使用需求工程中的技术来获取资产、威胁和对策，该方法用于在运行时实例化特定域的安全需求。文献[12]根据规则从自然语言中提取访问控制相关要素，并在欧盟数据保护条例中得到应用。文献[13]提出了安全策略词汇模型，描述高层安全策略和底层可执行策略，并制作了支持多平台、多策略语言和规则范式的工具。

基于规则的策略精化的优点是自动化程度高、精化速度快，缺点是精化的正确性取决于表内容的正确性、元素固定、不支持动态扩展和缺乏动态调整特性。

2.2　基于历史数据的策略精化

基于历史数据的策略精化的核心思想是学习系统的历史行为，并预测将来的行为来达到策略精化的目的。

文献[14-15]针对精化目标与历史数据的相似性进行策略精化。其中，文献[14]提出了基于案例推理（CBR, case-based reasoning）的策略精化方法，该方法建立高层策略和低层参数配置的映射关系，当需要为新的高层策略查找配置参数时，查找历史数据库中最相似的案例，或者在一组案例的配置参数间插入新的系统参数，以确定合适的配置参数。此外，文献[14]还提出了对策略聚类以减少查找的数据量，使用主成分分析进行降维以减少存储空间。文献[15]提出了一种基于文本挖掘的相似性度量方法，从策略库中选择类似的策略以设置本次的策略。

文献[16-17]基于历史数据，使用决策树等机器学习的方法进行策略精化。其中，文献[16]基于决策树的分类方法，利用系统状态等指标的历史信息，找到影响高层策略相应的系统状态指标，并确定阈值。其主要过程分为 4 个阶段：1)测试和开发阶段，通过设定高层次的目标，并运行系统，收集低层次系统状态指标；2)分类阶段，利用收集到的低层次系统状态指标构造决策树；3)策略细化阶段，通过决策树中叶子节点为真的路径找到关键的低层次策略元素（系统状态指标）；4)确定限制范围阶段，确定低层次策略参数的限制范围，该方法适用于服务等级协议（SLA, service-level agreement）等低层次策略的参数是数值型并且容易量化的场景。为减少数据中心的能源消耗，文献[17]通过计算能耗和性能要求之间的效用函数，使用决策树的方法对应用场景进行分类，将业务策略精化为网络级策略。

文献[18]通过建立非功能需求框架，将安全需求划分为由多个软目标实现，并将每个软目标细化为多个具体的操作化指令，构建策略精化的树状结构，对树中所有叶子节点计算评分，用于量化意图的符合情况，结合网络基本配置和服务依赖情况，实现了基于意图的策略精化。

基于历史数据的策略精化的优点是精化的准确性会随历史数据集规模的增大而增加，缺点是难于构建历史数据库，并且数据集中的错误数据会影响策略精化的准确性。

2.3　基于逻辑的策略精化

基于逻辑的策略精化的核心思想是采用事件演算、反绎推理、模型检验等方式[19-22]进行策略细化。

文献[23]使用事件演算（event calculus）形式化地定义策略，首先，利用需求工程领域的目标分解（KAOS, knowledge acquisition in automated specification）方法，将抽象目标分解为可操作的目标；其次，将可操作目标映射到相应模块，使用统一建模语言（UML, unified modeling language）对环境中各种系统实体及其关系进行建模；最后，利用反绎推理推导出使可操作性目标为真的事实，完成策略精化。文献[24]使用事件演算描述系统状态和策略应用的条件，将策略精化通过分解和操作这 2 个阶段完成，其中，分解阶段将分解规则作为输入，并将操作化策略与对象类进行匹配；操作阶段使用特定领域的描述信息和高层策略推导出可执行的动作。文献[25]提出了专家预先定义细化目标，再使用线性时序逻辑定义不同目标间的关系，通过模型检验进行策略精化，该方式更具自动性。

基于逻辑的策略精化的优点是具有一定程度的策略冲突消解能力，缺点是计算复杂度高。

2.4 其他方法

策略精化的其他方法包括基于本体的方法、不同策略描述语言之间的翻译等。基于本体的方法利用本体建模，保证语义一致性；不同策略描述语言之间的翻译包括基于角色的访问控制（RBAC, role-based access control）策略翻译为基于属性的访问控制（ABAC, attribute-based access control）策略等，为策略统一管理和精化提供了支撑。文献[26]为了解决单域内访问控制策略向多个域翻译、转换的问题，将访问控制策略转换成统一的二进制位串，然后生成较少规则数的访问控制标识语言（XACML, extensible access control markup language）描述的策略，通过动态验证和静态验证的方法，验证了转换前后策略语义的一致性，最后为关系型数据库转化为可扩展标识语言（XML, extensible markup language）格式的文档提供了安全发布函数。

通过对上述策略精化方法的分析可以看出，现有对策略精化的研究较少关注于威胁监测和信息采集领域，且策略精化方法需要在适用领域和自动化程度间进行权衡，如图 1 所示。由图 1 可知，越特定的方法，策略精化的自动化程度越高，需要人员参与的程度越低；反之，越通用的方法，策略精化的自动化程度越低，需要人员参与的程度越高。

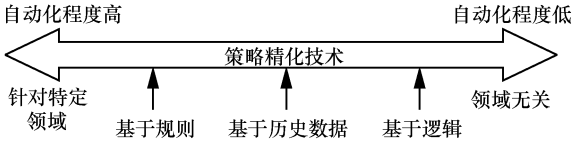


图 1 策略精化技术的自动化级别

3 采集策略精化系统模型

3.1 采集策略层次模型

网络信息系统通常包括用户层、服务层和拓扑层等层次结构，针对威胁监测，各层对应的安全策略分别是高层监测需求、中层采集策略和低层设备指令，形成采集策略层次模型，如图 2 所示。

采集策略层次模型规范了复杂网络环境下所有威胁监测相关的系统、服务、实体间的内部结构和相互间的关系。用户层表达了整个网络的安全监测目标和需求，即系统和管理活动应该完成何种监测任务和安全功能，例如，在指定的网络或者指定的主机上监测某类安全威胁，这一层是编写安全监测策略的基础和依据。服务层将安全需求实例化为安全服务，是从系统需求到网络配置映射的中间过程。拓扑层表达了网络设备、安全设备和终端等网络节点及其连接关系，以及在各网络节点上提供了何种监测机制以满足上层的安全监测需求，实现了安全目标的细化。

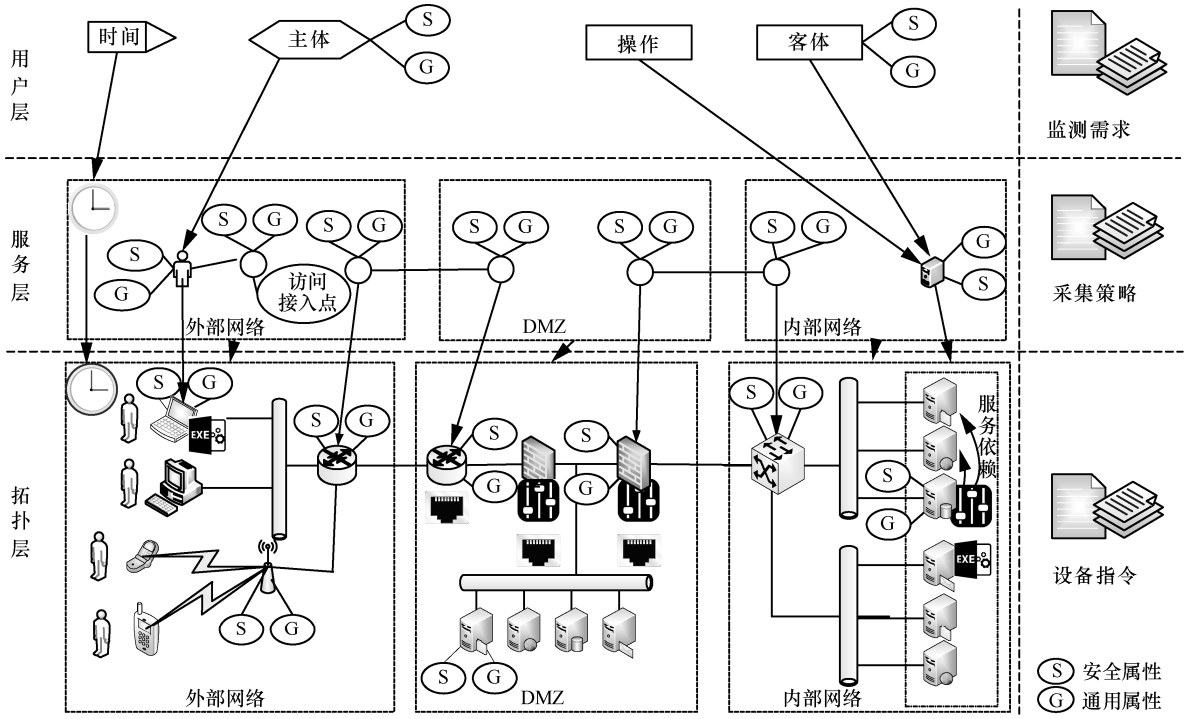


图 2 采集策略层次模型

3.2　采集策略精化定义

下面分别定义高层监测需求、中层采集策略、低层设备指令及策略精化规则和策略模板，为策略精化引擎计算威胁类型到采集项的精化提供基础。

定义 1　高层监测需求为在指定范围内监测指定威胁的需求及执行监测的约束条件。

高层监测需求可用三元组（SCOPE, THREAT_TYPE, CONDITION）表示，即 TDR=(SCOPE, THREAT_TYPE, CONDITION)。其中，SCOPE 表示采集范围的集合，可以是逻辑范围（如子网、子网类型、服务、服务类型、安全域、管理域等），也可以是物理范围（如地理区域、行政区域等）；THREAT_TYPE 表示需要监测的威胁类型，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；CONDITION 表示执行威胁监测的约束集，如执行威胁监测所需消耗资源的上限，包括计算、存储、传输和电量等资源限制，可用于限制在数据采集和威胁分析阶段的资源消耗。

定义 2　中层采集策略为在何种条件下和何种机器上，以何种频率采集何种内容。

中层采集策略可用四元组（DEVICE, COLLECT_ITEM, FREQ, CONDITION）表示，即 ICP=(DEVICE, COLLECT_ITEM, FREQ, CONDITION)。其中，DEVICE 表示待采集设备的集合，待采集设备可以是 IP 地址、管理员指定的 ID 等，待采集设备根据采集范围和威胁类型精化得出；COLLECT_ITEM 表示采集项的集合，可以是系统状态（如 CPU 利用率、内存利用率、进程信息等），也可以是流量信息（如原始流量信息、流量统计信息等），还可以是日志信息（如操作系统日志、应用程序日志等），采集项根据威胁类型精化得出，具体计算过程在第 4 节介绍；FREQ 表示采集项集合中各采集项的采集频率，对于系统状态，采集频率以实际采集频率与固有采集频率的比值表示，例如，CPU 利用率的固有采集频率为 10 次/min，实际采集频率为 20 次/min，则 CPU 利用率的采集频率为 2，采集频率在根据威胁类型计算采集项的过程中一起计算得出，具体计算过程在第 4 节介绍；CONDITION 表示执行数据采集的约束集，包括计算、存储、传输和电量等资源限制，由高层监测需求中的 CONDITION 精化得出。

定义 3　低层设备指令为在策略执行点上执行的具体采集指令及采集频率。

低层设备指令可用三元组（AGENT, CMD, FREQ）表示，即 LDC=(AGENT, CMD, FREQ)。其中，AGENT 表示在待采集设备上执行采集策略的策略执行点，可以是通用采集软件（如 SNMP 的代理、Snort、漏洞扫描工具等），也可以是自研的采集程序（如采集系统状态信息、流量采集器、威胁感知器等）；CMD 表示 AGENT 可执行的具体的采集指令；FREQ 表示实际采集频率。

定义 4　采集策略精化（CPR, collection policy refinement）为从高层监测需求到低层采集代理可执行的采集指令的翻译过程。该翻译过程分为 2 个阶段，第一阶段由高层监测需求翻译为中层采集策略，该过程是一个语义变换的过程，是威胁分析函数的逆函数，也是本文研究的重点，具体求解过程在第 4 节介绍；第二阶段由中层采集策略翻译为低层设备指令，该过程是一个语法变换的过程，可采用基于规则和查找表的方式计算，可参考文献[13,27]。

定义 5　威胁分析函数 $\text{ThreatAnalysis}(m_1,\cdots,m_n)$。输入是特定频率的采集项内容，输出是威胁类型，本文提出的第一阶段精化是求解威胁分析函数的逆函数，即在约束条件下，由威胁类型求解采集项和采集频率，即 $\text{ThreatAnalysis}^{-1}(t)=\{m_1,\cdots,m_n\}$。

定义 6　策略精化规则集（PRRS, policy refinement rule set）为高层监测需求到中层采集策略、中层采集策略到低层设备指令之间的精化规则，包括如下类型的规则。

规则 1　采集范围、威胁类型、待采集设备精化规则，表示采集范围、威胁类型到待采集设备的映射。

$$\text{RSTDEV}:\{(s,t)|s\in\text{SCOPE},t\in\text{THREAT\_TYPE}\}\rightarrow\text{DEVICE}^n$$

规则 2　威胁类型、采集项精化规则，表示威胁类型到采集项的映射。

$$\text{RTC}:\text{THREAT\_TYPE}\rightarrow\text{COLLECT\_ITEM}^n$$

规则 3　待采集设备、策略执行点精化规则，表示待采集设备到策略执行点的映射。

$$\text{RDEVAGT}:\text{DEV}\rightarrow\text{AGENT}$$

规则 4　采集项、采集配置指令精化规则，表

示采集项到采集配置指令映射。

$$\text{RITEMCMD}:\text{COLLECT\_ITEM}\rightarrow\text{CMD}$$

定义 7 策略模板是存储结构化策略元素的通用策略，提供有关策略精化引擎输入和输出的必要信息。

采集策略模板示例如下。

collect policy= “detect threat:” <threat_type>“; condition:”[condition] “;collect content:” {<collect_item >“-”<frequency>}。

上述策略模板实例化后如下。

detect threat: DDoS; condition: CPU utility < 10%, network bandwidth <10; collect content: OS Log-1, CPU Utility-2, Memory Utility-2, ProcessInfo-1。

该模板表示监测 DDoS 攻击，约束条件为每台待采集设备的 CPU 利用率占用小于 10%，网络带宽占用小于 10 KB，采集项为操作系统日志、CPU 利用率、内存利用率、进程信息，采集频率依次为各项固有采集频率的 1、2、2、1 倍。

策略精化规则集中规则的实例化组成了一系列采集策略，其中，RSTDEV、RDEVAGT、RITEMCMD 规则采用预定义信息模型方式获得，RTC 规则采用优化计算方式获得，在本文第 4 节着重介绍。

3.3 采集策略精化过程

基于采集策略层次模型的策略精化，从高层监测需求精化为中层采集策略，最终翻译为低层设备指令，如图 3 所示，其中①~⑨表示策略精化详细流程。

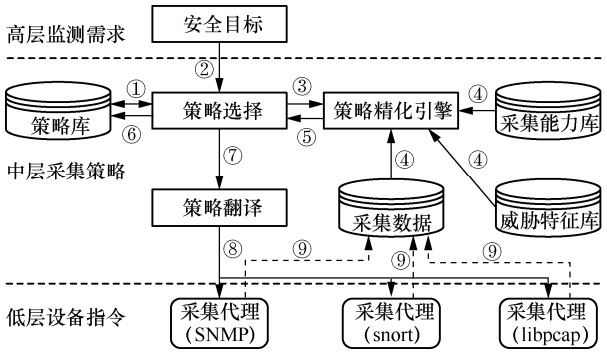


图 3 策略精化框架

由高层监测需求到低层设备指令的精化算法如算法 1 所示。

算法 1 采集策略精化算法 CPR(s, t, c)

输入 高层监测需求 TDR(s, t, c)，目标网络中的设备集合 $D(d_1, \cdots, d_n)$

输出 低层设备指令集合 LDC

1) $\text{DT}(d_1, \cdots, d_m) \leftarrow \varnothing$;
LDC $\leftarrow \varnothing$;
2) for each d_i in D do
3) if d_i.location in s then
4) r_i=FindResourcesStatus(d_i)
5) DT = DT $\cup$ $\{d_i\}$;
6) end if
7) end for
8) for each d_i in DT do
9) p_i = SearchPolicy(t, c, r_i);
10) if p_i == null
11) [p_i.item, p_i.freq] = RefineEngine(t, c, r_i);
12) p_i.type = t;
13) p_i.dr = r_i;
14) $P = P \cup \{p_i\}$;
15) end if
16) ldc_i = Translate(d_i, p_i.item, p_i.freq);
17) LDC = LDC $\cup$ ldc_i;
18) end for

采集策略精化算法具体步骤介绍如下。

步骤 1 监测需求提取与待采集设备确定（算法 1 的 2)~7)）。提取高层监测需求中的采集范围、威胁类型、执行威胁监测的约束集，使用 RSTDEV 规则，查找到待采集设备集合。

步骤 2 策略库查找（算法 1 的 9)）。根据威胁类型和执行威胁监测的约束集，查找对应的策略实例，如果查找成功，则跳转到步骤 4，如果查找失败，则跳转到步骤 3。

步骤 3 策略精化引擎生成新采集策略并存入采集库（算法 1 的 11)~14)）。将威胁类型、执行威胁监测的约束集和待采集设备的当前运行状态信息作为策略精化引擎的输入，结合采集代理的采集能力、威胁特征等信息，策略精化引擎计算采集收益和采集成本，生成适用于待采集设备当前运行状况的最优的采集项集合和各采集项对应的采集频率，其具体计算过程在第 4 节介绍。最后，将威胁类型、待采集设备的当前运行状态信息、采集项及对应的采集频率集合作为一条实例化采集策略，存储到策略库中，对策略库进行动态扩充。

步骤 4 中层策略到低层策略翻译（算法 1 的 16)~17)）。对于每个待采集设备，根据 RDEVAGT

规则，获取策略执行点信息，然后根据策略库中的采集项及对应的采集频率，使用 RITEMCMD 规则，将采集项翻译为采集配置指令，最后，在各策略执行点上，下发采集配置指令。

4　基于采集成本和收益平衡的策略精化

4.1　RTC 计算方法概述

本节提出了一种基于优化算法的威胁类型到采集项的精化方法，研究在不同资源水平下，执行采集策略的收益和因采集引起的资源消耗损失之间的平衡。通过优化求解，使策略精化引擎依据各待采集设备的资源水平生成相应的采集策略，提高采集收益，减少资源消耗。

从威胁类型到采集项的精化主要考虑采集收益和采集成本这 2 个方面的因素，其中，采集收益用于衡量采集信息对威胁监测的贡献度，采集有效数据越多，收益越大。一般情况，采集收益随采集项的采集频率单调递增，而边际收益随采集频率单调递减，即采集收益函数应满足以下限制条件。

1) 采集收益的一阶导数大于或等于 0。

2) 采集收益的二阶导数小于或等于 0。

本文的采集收益定义为

$$f_1(x_1,\cdots,x_m)=\lambda\ln\left(1+\sum_{i=1}^{m}\ln(1+c_ix_i)\right) \tag{1}$$

其中，λ 表示模型的系数，$\lambda>0$；c_i 表示采集项 i 的采集贡献度，$c_i>0$；x_i 表示采集项 i 的采集频率，$x_i\geqslant 0$，采集频率分为连续和离散 2 种类型，当采集项为系统状态类或日志类时，采集频率为连续值；当采集项为流量时，采集频率取离散值 0 或 1，$x_i=0$ 时，表示不对采集项 i 进行采集。根据式(2)和式(3)可知，式(1)满足限制条件 1)和 2)。

$$\frac{\partial f}{\partial x_i}=\frac{\lambda c_i}{(1+c_ix_i)\left(1+\sum_{i=1}^{m}\ln(1+c_ix_i)\right)}>0 \tag{2}$$

$$\frac{\partial^2 f}{\partial x_i^2}=\frac{-\lambda c_i}{(1+c_ix_i)^2\left(1+\sum_{i=1}^{m}\ln(1+c_ix_i)\right)^2}<0 \tag{3}$$

采集成本衡量因采集引起的待采集设备的资源消耗，包括待采集设备在计算资源、存储资源和传输资源等方面的消耗，一般认为采集消耗的资源越少越好，尤其是在资源受限环境，例如，天地一体化网络环境中卫星的计算资源和存储资源十分有限，用于管理的信令大小仅有数百字节，不能因采集影响管理网络的正常运行。采集成本定义为

$$f_2(x_1,\cdots,x_m)=w_{\mathrm{c}}\sum_{i=0}^{M}\mathrm{CRI}(x_i)+w_{\mathrm{s}}\sum_{i=0}^{M}\mathrm{SRI}(x_i)+w_{\mathrm{n}}\sum_{i=0}^{M}\mathrm{NRI}(x_i) \tag{4}$$

其中，$\mathrm{CRI}(x_i)$、$\mathrm{SRI}(x_i)$、$\mathrm{NRI}(x_i)$ 分别为采集项 i 的计算资源、存储资源和传输资源的消耗情况。

因此，RTC 的精化目的是确定目标函数 $f_1(x_1,\cdots,x_m)-f_2(x_1,\cdots,x_m)$ 最大时的 $X(x_1,\cdots,x_m)$，使采集项及其采集频率在降低资源开销的情况下尽可能全面、准确地反映网络实际运行状况。

下面，分别从采集收益和采集成本这 2 个方面进行介绍。

4.2　采集收益

由于同一采集项对监测多种威胁均有相关性，因此需要对采集项对威胁监测的贡献程度进行整体考虑，参考软件工程中的非功能性需求（NFR, non-functional requirement）框架[28]，将威胁监测作为最高层的安全目标，将待监测的威胁类型作为软目标（softgoal），将采集项作为操作（operationalization），参考软件目标依赖图（SIG, softgoal interdependency graph）设计威胁依赖图（TIG, threat interdependency graph），如图 4 所示。

TIG 定义为

$$\mathrm{TIG}=(V, E)$$

其中，$V\in\{\mathrm{TMG}, \mathrm{TTG}, \mathrm{CT}\}$ 是顶点集合，TMG 代表威胁监测目标的集合，是图 4 中的根节点，TTG 代表威胁类型的集合，是图 4 中的中间层节点，CT 代表采集项集合，是图 4 中的叶子节点；E 是边 e 的集合，$e=(v_1,v_2)$ 表示 v_1 和 v_2 间具有连接关系，其中，$v_1,v_2\in V$。

本文参考文献[29]中的威胁分类方式和检测方法，添加了威胁类型对于安全目标的影响程度。为计算采集项对威胁监测的影响，参考 Affleck 等[30]对 NFR 框架中操作化得分的定量计算，利用递推法求底层采集项对监测威胁的贡献程度。

首先，威胁类型定义为

$$\forall \mathrm{TTG}\in V, 0.0\leqslant \mathrm{TTG}_{\mathrm{weight}}\leqslant 1.0 \tag{5}$$

其中，$\mathrm{TTG}_{\mathrm{weight}}$ 为该威胁类型对于安全目标的影响权重，如图 4 中根节点到中间层节点的数值，其数值越大，代表对网络安全的影响程度越高。

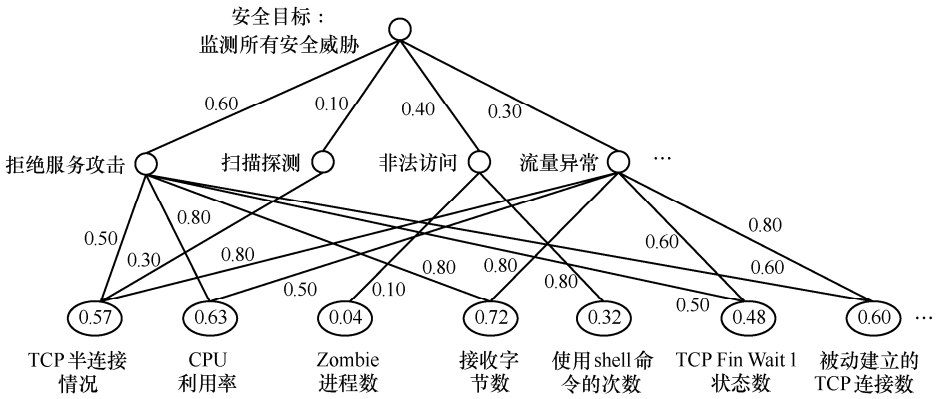


图 4 威胁依赖图示例

采集项的贡献度定义为

$$\mathrm{CT}_{\mathrm{store}}=\sum_{\mathrm{LTG}}\mathrm{TTG}_{\mathrm{weight}}\times\mathrm{impact}_{\mathrm{TTG}\times\mathrm{CT}} \tag{6}$$

其中，$\mathrm{CT}_{\mathrm{store}}$ 采用自顶向下方式进行计算，$\mathrm{impact}_{\mathrm{TTG}\times\mathrm{CT}}$ 表示威胁类型与采集项的关联度，如图 4 中的中间层节点到叶子节点的数值，可从以下 3 个方面进行考虑：1) 发生该威胁引起的攻击事件时，该采集项较系统正常运行时采集内容的变化程度；2) 发生威胁时采集项的内容发生改变的可能性；3) 分析该类威胁时，使用该采集项的必要性。

利用图 4 所示的 TIG，计算采集项 CPU 利用率的贡献度，CPU 利用率与拒绝服务攻击和流量异常均有相关性，$\mathrm{CT}_{\mathrm{store}}$=(0.80×0.60)+(0.50×0.30)=0.63，代表 CPU 利用率对于威胁监测的贡献度是 0.63。

对于系统状态类和流量类采集项与威胁的关联度，可以根据 KDD99 数据集[31]中连接的 41 个特征划分，并用其异常程度定量描述采集项与威胁的相关度。

对于日志类采集项与威胁的关联度，可定义为

$$\mathrm{impact}_{\mathrm{TTG}\times\mathrm{CT}_i}=\frac{\mathrm{validFieldNum}_i}{\mathrm{fieldNum}_i}\times\mathrm{validRate}_i \tag{7}$$

其中，$\mathrm{validFieldNum}_i$ 为采集项 i 的威胁分析有效字段数，$\mathrm{fieldNum}_i$ 为采集项 i 记录的总日志字段数，$\mathrm{validRate}_i$ 为日志有效率，即可被威胁分析使用的有效记录数与总采集记录数的比值。

4.3 采集成本

因采集引起的待采集对象的成本主要从计算资源、存储资源和传输资源这 3 个方面进行考虑。

1) 计算资源

利用 Perf 等性能分析工具，对不同采集项单独进行采集时捕获其 CPU 利用率（task-clock-msecs），预估各采集项的计算资源消耗。各采集项的计算成本定义为

$$\mathrm{CRI}(x_i)=a_i\ln(1+x_i) \tag{8}$$

其中，a_i 为采集项 i 的计算成本，x_i 为采集项 i 的采集频率。

2) 存储资源

在进行本地缓存时，各采集项的存储成本定义为

$$\mathrm{SRI}(x_i)=n_i x_i \tag{9}$$

其中，n_i 为采集项 i 的字节数。

3) 传输资源

与采集信息传输相关的时延有处理时延、排队时延、传输时延、传播时延等，当前网络带宽为 BW，则各采集项的传输成本定义为

$$\mathrm{NRI}(x_i)=\frac{n(x_i)}{\mathrm{BW}} \tag{10}$$

总采集成本如式(4)所示，其中，x_i 是决策属性，代表采集频率。当采集项 i 为系统状态和日志时，$x_i\in[0,+\infty)$；当采集项 i 为流量时，$x_i\in\{0,1\}$，“0”代表不采集，“1”代表采集。采集频率乘以预设固有采集频率等于实际采集频率。

w_{c}、w_{s} 和 w_{n} 是 3 个权重，分别代表计算资源、存储资源和传输资源对于成本计算的重要程度，由于计算资源、存储资源和传输资源的变化是可逆的，因此可以采用自适应加权。当上述三者中的某一项剩余资源减少得快时，会导致其对采集成本的影响较大。w_{c}、w_{s}、w_{n} 分别定义为

$$w_{\mathrm{c}}=\frac{S+N}{2(C+S+N)} \tag{11}$$

$$w_{\mathrm{s}}=\frac{C+N}{2(C+S+N)} \tag{12}$$

$$w_{\mathrm{n}}=\frac{C+S}{2(C+S+N)} \tag{13}$$

其中，C、S 和 N 分别是采集代理的当前剩余的计算资源百分比、剩余的存储资源百分比和剩余的传输资源百分比。当某一项资源剩余情况较好时，会导致采集方案对该项采集成本的影响不明显，并且可以限制在不同资源水平下的采集成本，例如，在其他条件相同的情况下，电量充沛时的采集成本较低，电量剩余量少时的采集成本增加。C、S 分别定义为

$$C=1-\mathrm{utilization}_{\mathrm{CPU}} \tag{14}$$

$$S=1-\mathrm{utilization}_{\mathrm{disk}} \tag{15}$$

其中，$\mathrm{utilization}_{\mathrm{CPU}}$ 为采集代理当前的 CPU 利用率，$\mathrm{utilization}_{\mathrm{disk}}$ 为采集代理当前的磁盘利用率。

4.4　目标函数

目标函数定义为

$$\begin{aligned}\max f(X;Y)=&\lambda\ln\left(1+\sum_{i=0}^{M}\ln(1+c_i x_i)\right)-\\&w_{\mathrm{c}}\sum_{i=0}^{M}\mathrm{CRI}(x_i)-w_{\mathrm{s}}\sum_{i=0}^{M}\mathrm{SRI}(x_i)-\\&w_{\mathrm{n}}\sum_{i=0}^{M}\mathrm{NRI}(x_i)+\\&\lambda\ln\left(1+\sum_{i=0}^{N}\ln(1+c_i y_i)\right)-\\&w_{\mathrm{c}}\sum_{i=0}^{M}\mathrm{CRI}(y_i)-w_{\mathrm{s}}\sum_{i=0}^{N}\mathrm{SRI}(y_i)-\\&w_{\mathrm{n}}\sum_{i=0}^{N}\mathrm{NRI}(y_i)\end{aligned}$$

$$\text{s.t.}\begin{cases}\forall x_i\geqslant 0\\ \forall y_i=\{0,1\}\\ \sum_{i=0}^{M}\mathrm{CRI}(x_i)+\sum_{i=0}^{N}\mathrm{CRI}(y_i)\leqslant \max_{\mathrm{CR}}\\ \sum_{i=0}^{M}\mathrm{SRI}(x_i)+\sum_{i=0}^{N}\mathrm{SRI}(y_i)\leqslant \max_{\mathrm{SR}}\\ \sum_{i=0}^{M}\mathrm{NRI}(x_i)+\sum_{i=0}^{N}\mathrm{NRI}(y_i)\leqslant \max_{\mathrm{NR}}\end{cases} \tag{16}$$

其中，X 表示系统状态类和日志类的采集频率集合，是连续变量，取“0”代表不采集，其他值代表采集频率；Y 表示流量类的采集频率，取“0”代表不采集，取“1”代表采集。约束条件表示计算资源消耗小于或等于 $\max_{\mathrm{CR}}$，存储资源消耗小于或等于 $\max_{\mathrm{SR}}$，传输资源消耗小于或等于 $\max_{\mathrm{NR}}$。

4.5　基于遗传算法的采集项精化方法

RTC 的精化是一个离散型和连续型混合的非线性优化问题，求解该问题的方法有数学方法、演化计算方法等，其中，数学方法包括罚函数法、可行方向法、逐步二次规划法等，演化计算方法包括遗传算法、粒子群算法、文化算法等。演化计算用概率的变迁规则来控制搜索的方向，在概率意义上朝最优解方向靠近，在有限时间内可得到近似最优解，本文采用遗传算法对威胁类型到采集项进行精化，主要分为如下 3 个部分。

1) 编码

编码是根据问题的解空间确定染色体中个体表现型的长度，本文采用二进制编码作为基因型编码，若采集频率取值范围为 $[a,b]$，精度为小数点后 l 位，则二进制编码长度 k 需满足

$$2^{k}<10^{l}(b-a)<2^{k+1} \tag{17}$$

对式(17)进行解析，得到 k 的取值范围为

$$\mathrm{lb}(10^{l}(b-a))-1<k<\mathrm{lb}(10^{l}(b-a)) \tag{18}$$

本文方案中，对于系统状态类和日志类采集项的采集频率为连续型，其取值范围为[0,10]，设定精度为小数点后 4 位，按照式(18)，二进制编码串长度为 17；对于流量类采集项的采集频率为离散型，取值为 0 和 1，二进制编码串长度为 1。

2) 适应度函数

适应度函数是对算法所产生的染色体进行评价，并基于适应度值选择染色体的函数，本文采用式(16)中的目标函数作为适应度函数，采用式(16)中的约束函数判断染色体中的个体是否是可行解，若不满足约束函数，则标记为非可行解，在对当代种群的最优个体做记录等操作时不考虑非可行解。

3) 遗传算子

遗传算子分为交叉算子、变异算子和选择算子。交叉操作在种群中随机选取 2 个个体作为父个体，并把 2 个父个体的部分码值进行交换操作。目前主流的交叉算子包括单点交叉、双点交叉、均匀交叉、算术交叉等，本文采用经典的单点交叉，以最大限度地保存父个体的优势。对于变异算子，本文采用以变异概率对染色体中的个体进行补运算，以完成二进制编码中新搜索领域的开辟。除上述交

叉算子和变异算子外，还需确定合适的交叉概率 P_c 和变异概率 P_m，综合考虑收敛速度、产生新个体的能力，达到防止算法陷入局部最优的目的。目前 2 个概率值主要依靠经验的方法得到，一般情况下 P_c 取值范围为[0.40, 0.99]，P_m 取值范围为[0.000 1, 0.100 0][32]。选择算子依据个体适应度值选择在下一代中被保留还是淘汰，目前主流的选择算子有轮盘赌选择、排序选择、期待值选择等，本文选择轮盘赌方法，使适应度值大的染色体被选中的概率大。

5 模拟实验

5.1 实验方案

本文针对高层监测需求中的威胁类型到中层采集策略的采集项和采集频率的精化进行模拟实验。实验平台为：Intel(R) Core(TM) i7-7500U 2.70 GHz，8 GB 内存，1 TB 存储，Windows 10 操作系统，Python 3.6.5 软件。通过设置不同的高层监测需求，验证监测不同威胁类型和待采集设备处于不同资源水平情况下，精化后的采集项和频率的合理性。

表 1 为实验中各采集项的采集贡献度。对于 DDoS 攻击，在受到该类攻击后，系统的计算和存储资源水平受到影响，所以 CPU 占用率和内存利用率这 2 项采集项的贡献度较大，而端到端的 IP 流量也有利于分析该类攻击，所以其贡献度也较大；对于非法访问，在受到该类攻击时，总体流量不会有明显变化，所以端到端的 IP 流量贡献度较小，但是针对非法访问的服务流量需要着重采集，所以特定服务的业务流量采集项的贡献度较大；对于 FTP 木马，采集 FTP 服务日志及特定服务的业务流量有利于分析该类威胁，所以这 2 项的贡献度较高。

表 1 采集项采集贡献度验证实验参数

采集项名称	DDoS	非法访问	流量异常	FTP 木马
系统日志	0.80	0.50	0.60	0.20
FTP 服务日志	0.10	1 .00	0.10	1.00
端到端的 IP 流量	1.00	0.10	1.00	0.20
特定服务的业务流量	0.40	1.00	0.80	1.00
CPU 占用率	0.80	0.40	0.80	0.40
内存利用率	0.60	0.30	0.60	0.30

表 2 为各采集项每次采集的计算、存储、传输的成本。日志类采集项以读取文件方式进行采集，占用非常少的计算资源，在存储和传输成本方面，每次增量采集的日志量一般不大，所以分别估算为 0.192 和 0.128；流量类的端到端的 IP 流量采集方式占用计算资源较日志类高，同时过滤某一特定服务的业务流量也较端到端的 IP 流量采集方式计算成本大，所以这 2 项的计算成本分别估算为 0.200 和 0.600，在存储和传输成本方面，端到端的 IP 流量采集方式最高，估算为 0.300，特定服务的业务流量采集较全采集的存储和传输成本低，估算为 0.128；系统状态类采集项以系统调用或读文件方式进行采集，计算成本适中，估算为 0.300，在存储和传输成本方面，由于存储状态类信息占用的空间少，因此较其他类低很多，估算为 0.040。

表 2 采集项采集成本验证实验参数

采集项名称	计算成本常量	存储成本常量	传输成本常量
系统日志	0.100	0.192	0.192
FTP 服务日志	0.100	0.128	0.128
端到端的 IP 流量	0.200	0.300	0.300
特定服务的业务流量	0.600	0.128	0.128
CPU 占用率	0.300	0.040	0.040
内存利用率	0.300	0.040	0.040

实验分为如下 3 个部分。

1) 面对不同威胁类型时，生成适用于监测该类威胁的个性化的采集项及对应的采集频率。具体而言，在待采集设备处于相同资源水平时，例如，当前剩余资源非常丰富，即剩余的计算资源百分比、存储资源百分比和传输资源百分比均为 1，分别监测 DDoS、非法访问、流量异常、FTP 木马攻击，或者一些组合攻击，观察和分析策略精化引擎生成的采集项及对应的采集频率情况。

2) 在监测同一种威胁时，待采集设备处于不同资源水平，采集策略精化引擎生成的采集项及对应采集频率情况。

3) 在监测同一种威胁时，采用遗传算法、粒子群算法、穷举法和随机法在不同规模的采集项情况下进行精化，比较精化生成采集项及采集频率的时间和目标函数的值。

5.2 实验及结果分析

在待采集设备上，使用表 1 和表 2 所示的实验参数针对 4 类攻击进行策略精化，系统参数 λ=2，得到每个采集项的采集频率如图 5 所示。对于 4 类

攻击均以不同频率采集 CPU 占用率和内存利用率；对于非法访问和 FTP 木马等攻击，分别以不同频率采集 FTP 服务日志和 FTP 的业务流量；对于流量异常攻击，采集端到端的 IP 流量；对于 DDoS、非法访问和流量异常这 3 类攻击，均以不同频率采集系统日志。

图 6 为待采集设备处于不同计算资源水平时针对 DDoS 的监测，即在不同约束条件下进行采集策略精化，得到每个采集项的采集频率。在计算资源水平较低时，计算资源在成本计算的权重较高，为达到最大的采集收益，选择较少占用计算资源的系统日志和端到端的 IP 流量等采集项，且其采集频率较高；当计算资源水平升高后，计算资源在成本计算的权重降低，选择较多占用计算资源的系统状态类采集项，且其采集频率随计算资源水平升高而逐渐升高。

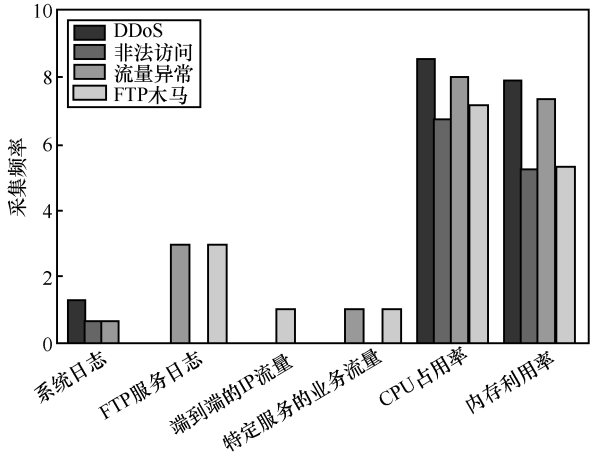


图 5　监测不同威胁类型精化生成的采集项及频率

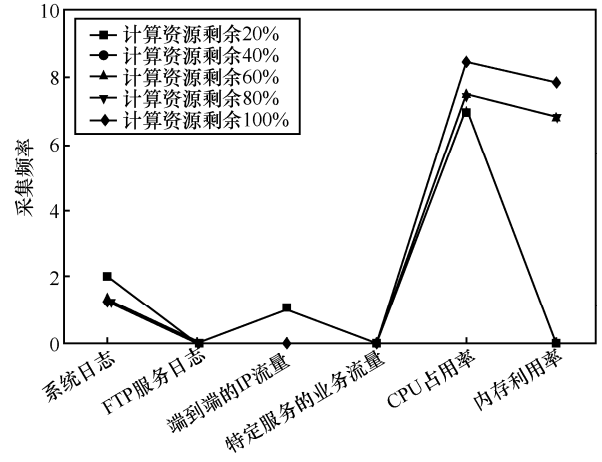


图 6　不同计算资源水平精化生成的采集项及频率

图 7 为待采集设备处于不同传输资源水平时针对流量异常的监测，即在不同约束条件下进行采集策略精化，得到系统状态日志采集频率变化情况，随着传输资源水平的升高，系统日志采集项的采集频率逐渐升高。

在对比实验中，采集项频率范围设置为[0, 10]。使用随机生成采集项和采集频率模拟未使用本文方案进行精化的方法。穷举法是对各采集项取 1~10 间的整数组合进行穷举。粒子群算法采用本文提出的目标函数作为其适应度函数，进行 100 个粒子的 200 轮迭代，个体经验系数和群体经验的加速系数均设置为 1，惯性参数也设置为 1，采用 Sigmoid 函数，根据速度分量决定离散型决策变量在迭代中取 1 或 0 的概率[32]。遗传算法采用第 4 节所述的 100 个个体的 200 轮迭代，交叉概率 P_c=0.8，变异概率 P_m=0.000 1。

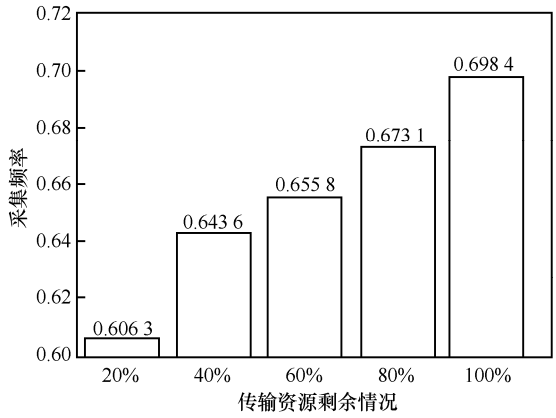


图 7　不同传输资源水平精化生成的系统日志采集项的采集频率

图 8 和图 9 为不同算法的采集策略精化的计算时间和效果比较。图 8 和图 9 的横坐标均为候选采集项数量，既有连续型决策变量，也有离散型决策变量；图 8 纵坐标为计算时间，以秒为单位，图 9 纵坐标为目标函数值，度量采集效果。

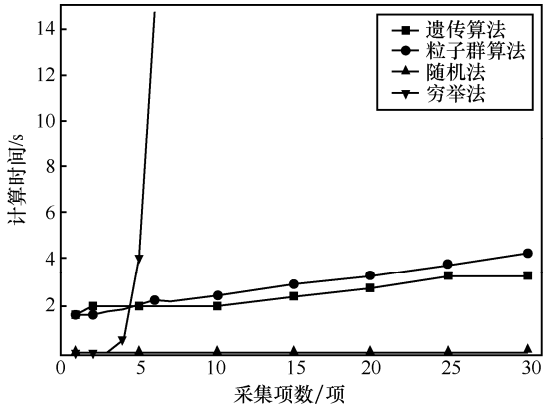


图 8　不同算法的采集策略精化计算时间比较

由图 8 和图 9 可知，随机法生成采集项的计算时间接近为 0 s，但是会使目标函数值小于 0，即采集收益小于采集成本，且随着采集项数的增加，目标函数值不断减少，即采集收益增加甚微，但是采集成本增大快。穷举法的计算时间将呈指数级增长趋势，在采集项超过 6 项时远高于其他算法，不符合精化的实时性需求。粒子群算法在计算时间方面与遗传算法相仿，但是采集效果劣于遗传算法。

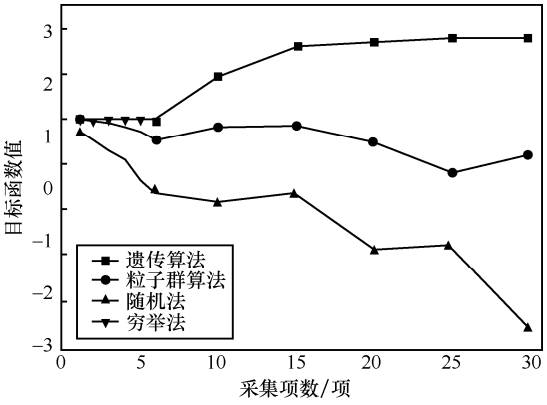


图 9 不同算法的采集策略精化效果比较

综上，遗传算法在计算时间上与粒子群算法持平，高于随机法，远低于穷举法。但是在目标函数求解效果方面，遗传算法在 4 种算法中最优，并且随着采集项数的增加，其采集效果呈上升趋势，即可供选择的采集项数增加，采集收益较采集成本增加快，不会使采集造成的资源消耗过量增加。

6 结束语

本文针对复杂网络环境下设备数量繁多且各设备资源不均衡的问题，提出了一种面向威胁监测的采集策略精化方法，该方法引入了策略模板，考虑了待采集设备的当前计算资源水平、存储资源水平和传输资源水平等因素，通过设置采集贡献度和采集成本，以混合优化计算指定威胁类型的采集项及各采集项对应的采集频率。模拟实验结果表明，该采集策略精化方法可以有效地生成采集方案，并且在资源水平较低时，还可自适应地减少采集项，降低采集频率。

虽然基于策略模板和混合优化计算的方式是一种有效的策略精化方法，但本文缺乏对计算、存储和传输等成本的高精度量化。因此，进一步的研究工作将重点分析各采集项的采集成本，并增加对能耗成本的量化计算。

参考文献：

[1] 李凤华, 熊金波. 复杂网络环境下访问控制技术[M]. 北京: 人民邮电出版社, 2015.
LI F H, XIONG J B. Access control technology in complex network environment[M]. Beijing: Posts & Telecom Press, 2015.

[2] PADHY P, DASH R K, MARTINEZ K, et al. A utility-based sensing and communication model for a glacial sensor network[C]//The 5th International Joint Conference on Autonomous Agents and Multi-agent Systems. 2016: 1353-1360.

[3] 曾文序, 库少平, 郑浩. 基于旋转门算法的自适应变频数据采集策略[J]. 计算机应用研究, 2018, 35(3): 769-772.
ZENG W X, KU S P, ZHENG H. Strategy of self-adaptive frequency conversion data acquisition based on swing door trending algorithm[J]. Application Research of Computers, 2018, 35(3): 769-772.

[4] 张斌, 朱建涛, 徐罡. 基于动态频率算法的远程监控系统数据采集优化策略[J]. 微电子学与计算机, 2016, 33(8): 86-91.
ZHANG B, ZHU J T, XU Z. Data gathering optimization strategy based on dynamic frequency algorithm for remote monitoring system[J]. Microelectronics & Computer, 2016, 33(8): 86-91.

[5] MAULLO M, CALO S. Policy management: an architecture and approach[C]//The 1st International Workshop on Systems Management. 1993: 13-26.

[6] MONT M C, BALDWIN A, GOH C. Power prototype: towards integrated policy-based management[C]//IEEE/IFIP Network Operations and Management Symposium. 2000: 789-802.

[7] ALBUQUERQUE J P D, KRUMM H, GEUS P L D, et al. Scalable model-based configuration management of security services in complex enterprise networks[J]. Software: Practice and Experience, 2011, 41(3): 307-338.

[8] ALBUQUERQUE J P D, KRUMM H, GEUS P L D. Formal validation of automated policy refinement in the management of network security systems[J]. International Journal of Information Security, 2010, 9(2): 99-125.

[9] ROMEIKAT R, BAUER B. Specification and refinement of domain-specific ECA policies[C]//International Conference on Advanced Information Systems Engineering Workshops. 2011: 197-206.

[10] NEISSE R, STERI G, GENEIATAKIS D, et al. A privacy enforcing framework for Android applications[J]. Computers & Security, 2016, 62: 257-277.

[11] RUDOLPH M, FETH D, DOERR J, et al. Requirements elicitation and derivation of security policy templates—an Industrial case study[C]//The 24th International Requirements Engineering Conference. 2016: 283-292.

[12] GUARDA P, RANISE S, SISWANTORO H. Security analysis and legal compliance checking for the design of privacy-friendly information systems[C]//The 22nd ACM on Symposium on Access Control Models and Technologies. 2017: 247-254.

[13] RUDOLPH M, MOUCHA C, FETH D. A framework for generating user and domain-tailored security policy editors[C]//The 24th International Requirements Engineering Conference Workshops. 2016: 56-61.

[14] BEIGI M S, CALO S, VERMA D. Policy transformation techniques in

policy-based systems management[C]//The 5th IEEE International Workshop on Policies for Distributed Systems and Networks. 2004: 1-10.

[15] HAN W, FANG Z, YANG L T, et al. Collaborative policy administration[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 498-507.

[16] UDUPI Y B, SAHAI A, SINGHAL S, A classification-based approach to policy refinement[C]//The 10th IFIP/IEEE International Symposium on Integrated Network Management. 2007: 785-788.

[17] RIEKSTIN A C, JANUÁRIO G C, RODRIGUES B B, et al. Orchestration of energy efficiency capabilities in networks[J]. Journal of Network and Computer Applications, 2016, 59: 74-87.

[18] SCHEID E J, MACHADO C C, FRANCO M L, et al. INSpIRE: integrated NFV-based intent refinement environment[C]//IFIP/IEEE Symposium on Integrated Network and Service Management. 2017: 186-194.

[19] RANA A I, JENNINGS B. Semantic aware processing of user defined inference rules to manage home networks[J]. Journal of Network and Computer Applications, 2017, 79: 68-87.

[20] MACHADO C C, WICKBOLDT J A. GRANVILLE L Z, et al. ARKHAM: an advanced refinement toolkit for handling service level agreements in Software-Defined Networking[J]. Journal of Network and Computer Applications, 2017, 90: 1-16.

[21] RIEKSTIN A C, JANUÁRIO G C, RODRIGUES B B, et al. A survey of policy refinement methods as a support for sustainable networks[J]. IEEE Communications Surveys and Tutorials, 2016, 18(1): 222-235.

[22] AZIZ B, Modelling fine-grained access control policies in grids[J]. Journal of Grid Computing, 2016, 14(3): 477-493.

[23] BANDARA K, LUPU E C, RUSSO A. Using event calculus to formalise policy specification and analysis[C]//The 4th International Workshop on Policies for Distributed Systems and Networks. 2003: 26-39.

[24] CRAVEN R, LOBO J, LUPU E, et al. Policy refinement: decomposition and operationalization for dynamic domains[C]//The 7th International Conference on Network and Services Management. 2011: 115-123.

[25] RUBIO-LOYOLA J. Towards the policy refinement problem in policy-based management systems[M]. Saarbrücken: VDM Verlag, 2008.

[26] LEIGHTON G, BARBOSA D. Access control policy translation, verification, and minimization within heterogeneous data federations[J]. ACM Transactions on Information and System Security, 2011, 14(3): 1-28.

[27] JOHNSON M, KARAT J, KARAT C M, et al. Usable policy template authoring for iterative policy refinement[C]//IEEE International Symposium on Policies for Distributed Systems and Networks. 2010: 18-21.

[28] CHUNG L, NIXON B A, YU E, et al. Non-functional requirements in software engineering[M]. New York: Springer Publishing Company, 2000.

[29] LIPPMANN R, HAINES J W, FRIED D J, et al. The 1999 DARPA off-line Intrusion Detection Evaluation[J]. Computer Networks, 2000, 34(4): 579-595.

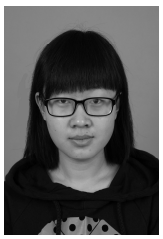
[30] AFFLECK A, KRISHNA A. Supporting quantitative reasoning of non-functional requirements: a process-oriented approach[C]// International Conference on Software and System Process. 2012: 88-92.

[31] KAMRAN S, ZAHID A, FARRUKH A K, et al. KDD cup 99 data sets: a perspective on the role of data sets in network intrusion detection research[J]. Computer, 2019, 52(2): 41-51.

[32] 窦全胜, 陈姝颖. 演化计算方法及应用[M]. 北京: 电子工业出版社, 2016.
DOU Q S, CHEN S Y. Evolutionary calculation method and application[M]. Beijing: Publishing House of Electronics Industry, 2016.

[作者简介]

李凤华（1966− ），男，湖北浠水人，博士，中国科学院研究员、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。



李子孚（1992− ），女，内蒙古赤峰人，中国科学院博士生，主要研究方向为访问控制、安全策略管理。

李凌（1993− ），女，湖南浏阳人，中国科学院硕士生，主要研究方向为安全策略管理。

张铭（1997− ），男，浙江宁波人，主要研究方向为访问控制。

耿魁（1989− ），男，湖北红安人，博士，中国科学院助理研究员，主要研究方向为网络安全。

郭云川（1977− ），男，四川营山人，博士，中国科学院副研究员、博士生导师，主要研究方向为访问控制、形式化方法。