计算机科学
COMPUTER SCIENCE

基于分层任务网络的攻击路径发现方法

王子博, 张耀方, 陈翊璐, 刘红日, 王佰玲, 王冲华

引用本文

王子博, 张耀方, 陈翊璐, 刘红日, 王佰玲, 王冲华. 基于分层任务网络的攻击路径发现方法[J]. 计算机科学, 2023, 50(9): 35-43.

WANG Zibo, ZHANG Yaofang, CHEN Yilu, LIU Hongri, WANG Bailing, WANG Chonghua. Hierarchical Task Network Planning Based Attack Path Discovery[J]. Computer Science, 2023, 50(9): 35-43.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230500025

基于分层任务网络的攻击路径发现方法

王子博[1] 张耀方[1] 陈翊璐[1] 刘红日[1,2] 王佰玲[1] 王冲华[3]
1 哈尔滨工业大学(威海)计算机科学与技术学院 山东 威海 264209
2 威海天之卫网络空间安全科技有限公司 山东 威海 264209
3 国家工业信息安全发展研究中心 北京 100040
(wzb_inet_hitwh@hotmail.com)

摘 要 攻击路径发现是辅助网络资产安全评估的一项关键任务。现有基于智能规划的攻击路径发现方法因建模语言丰富和规划算法完备而深受安全从业者青睐,但其存在的扩展性问题不容忽视。为此,提出一种基于分层任务网络的攻击路径发现方法。具体而言,围绕网络规模逐步扩展、路径发现任务愈加复杂和安全推演场景频繁变化所引发的扩展性问题,将所提方法分解为3个阶段。第一阶段,针对对路径生成性能差的问题,引入面向目标拓扑的多层级K路划分算法;第二阶段,针对领域问题描述难的问题,构建融入专家经验的路径规划分层任务网络;第三阶段,针对路径更新效率低的问题,设计应对局部信息更替的攻击路径维护方案。实验结果表明,所提方法适用于大规模网络,执行效率更高,具备良好的扩展性。

关键词:智能规划分层任务网络;多层级K路算法;攻击路径发现;攻击路径扩展
中图法分类号 TP393

Hierarchical Task Network Planning Based Attack Path Discovery

WANG Zibo[1],ZHANG Yaofang[1],CHEN Yilu[1],LIU Hongri[1,2],WANG Bailing[1] and WANG Chonghua[3]
1 School of Computer Science and Technology,Harbin Institute of Technology(Weihai),Weihai,Shandong 264209,China
2 Weihai Cyberguard Technologies Co. Ltd,Weihai,Shandong 264209,China
3 China Industrial Control Systems Cyber Emergency Response Team,Beijing 100040,China

Abstract Attack path discovery is a critical task for cyber asset security assessment. The existing artificial intelligence-based planning for attack path discovery method is favored by security practitioners due to its rich modeling language and complete planning algorithm,but its scalability problem cannot be ignored. For that reason,a hierarchical task network-based attack path method is proposed. Specifically,concerning the scalability problem,the proposed method is decomposed into the following three stages:first and foremost,focusing on undesirable attack path generation performance caused by expanding network scale,a multi-level K-way partitioning algorithm is introduced into the target topology;subsequently,focusing on the difficulty of domain problem description caused by complex discovery tasks,an attack path-oriented hierarchical task network is constructed with the combination of expert experiments;and finally,focusing on low attack path updating efficiency caused by demands on what-if security analysis,a maintenance scheme is designed for local information changes of assets. Experimental results show that the proposed method is suitable for attack path discovery in large-scale network which has an advantage over efficiency and scalability.

Keywords Artificial intelligence planning,Hierarchical task network,Multi-level K-way partitioning algorithm,Attack path discovery,Attack path extension

1 引言

现有网络资产存在的安全漏洞数量逐年递增,为潜在攻击者实施兼具靶向性、持续性、隐蔽性和多步性等特点的网络攻击提供了便利,个人、企业和国家的资产管理者正面临严峻挑战。攻击路径在狭义上用于描述攻击者从初始入口逐步攻陷目标资产的一系列利用行为及渗透过程,从广义上则反映资产间的脆弱性依赖关系。由此,攻击路径发现被视为透过攻击者行为探寻真实威胁意图,辅助网络资产安全评估的关键任务,近二十年来引起了国内外学者的广泛关注[1-2]。

智能规划作为人工智能研究领域的重要分支,旨在指导智能体根据一组来自环境的初始条件寻找实现特定目标的一组动作序列。通过上述描述不难发现,如果将智能体看作一个攻击者,动作序列寻找与攻击路径发现两者从任务本质上

到稿日期:2023-05-05 返修日期:2023-07-06
基金项目:国家重点研发计划(2021YFB2012400)
This work was supported by the National Key R&D Program of China(2021YFB2012400).
通信作者:王佰玲(wbl@hit.edu.cn)

来看十分相似，因而智能规划领域研究成果得以在网络安全领域中应用[3]。丰富的领域问题描述语言保障了攻击场景建模的全面性，完备的规划算法能够满足高效搜索攻击路径的需求。

然而，相较于同时期基于逻辑推理的攻击路径发现方法在安全评估领域内的成功应用[4]，扩展性问题一直制约着基于智能规划的攻击路径方法在现实应用场景中的推广。扩展性问题具体表现在3个方面：1)随着网络规模扩大，有待分析的网络资产数量递增，规划算法的攻击路径生成效率问题凸显；2)现有攻击路径发现任务逐步复杂化，致使部分领域问题模型难以描述具体任务，亟需融入专家经验；3)应对网络资产安全推演需求，如新增设备、漏洞更新等局部信息更替，使得攻击路径维护问题凸显。

针对上述问题，本文提出了基于分层任务网络的攻击路径发现方法，构建基于智能规划的攻击路径发现框架，开展3项关键技术相关研究，即基于分层任务网络的攻击路径规划模型、融合多层级 K 路算法的拓扑图划分方法，以及面向局部信息更替的攻击路径维护方案，从而改善基于智能规划的攻击路径发现方法的扩展性。

2 相关工作

近年来，攻击图建模是攻击路径发现方法研究的主要技术途径之一。探寻攻击路径发现方法的扩展性问题，离不开对主流攻击图模型扩展性的多方衡量，如形式化表述、路径生成和模型更新等环节的处理方式和运行效率。本章重点关注逻辑推理和智能规划两项技术在攻击路径发现研究中取得的成果。此外，日趋成熟的图划分算法也为解决攻击路径发现方法的扩展性问题提供了新思路，因而，本章进一步梳理图划分算法在攻击图建模领域的应用成果。

2.1 基于逻辑推理的攻击路径发现研究

Ou 等[4]于2005年率先关注到攻击路径发现方法的扩展性问题，提出了基于逻辑推理的攻击图模型，形成了一款用于分析多主机、多阶段安全漏洞的端到端框架，并将其命名为MulVAL(Multi-host, multistage Vulnerability Analysis)。该框架采用逻辑推理语言 Datalog 对攻击场景进行建模，并借助 XSB 推理引擎生成逻辑攻击图，在漏洞信息自动化和应对大规模网络场景的路径生成效率两方面均有显著改善。

但是，原生 MulVAL 框架内置推理规则有限，导致其在刻画复杂异构网络和攻击场景时表达能力不足。为此，Inokuchi 等[5-6]给出交互规则扩展的具体设计过程，从而丰富了关于通用网络协议脆弱性和无线、总线协议通信交互模型的推理规则，使 MulVAL 可以适配更多场景，如工业控制系统，完成特定攻击路径发现任务。此外，由于环境发生部分变化而重新生成全局攻击图是对计算资源的一种浪费，同时，在大规模网络场景中上述方式更是极不可取。针对这一问题，Saha[7]从 XSB 执行逻辑推理查询环节着手，基于推导图生成过程中子目标请求和查询结果的依赖关系，提出了一种增量式攻击图维护算法，用来识别已构建的图模型所需更新部分，完成局部攻击图的重新生成。

2.2 基于智能规划的攻击路径发现研究

如前所述，智能规划技术之所以能够运用在攻击路径发现任务中，除了源于智能体找寻动作序列与模拟攻击者发现攻击路径两项任务的高度相似性，还得益于丰富的领域建模语言和完备的路径规划算法(简称为规划器)。其中，规划领域定义语言(Planning Domain Definition Language, PDDL)作为主流建模语言，通过“领域”和“问题”形式化地描述攻击场景[8-10]。

在路径规划算法方面，Ghosh 等[8]首先基于 SGPlan 规划器对当前攻击场景下的最短攻击路径进行求解，随后设计外部枚举算法，通过多次修改编码在“问题”中的攻击场景而后调用规划器，以发现全部攻击路径；Gao 等[9]提出基于双向蚁群的最优攻击路径发现方法，实现了在较大规模网络下的攻击路径高效发现，同时引入重规划机制，在攻击场景中设备发生变化时，对局部攻击路径进行调整；Zang 等[3]面向自动化渗透测试领域的攻击路径发现应用，对相关的领域独立智能规划技术展开综述，包括确定性规划、非确定性规划和博弈规划3类算法。其中，确定性规划算法又进一步划分为规划图、偏序规划和分层任务网络。

在攻击路径维护方面，Bezawada 等[10]设计了一款名为 AGBuilder 的工具，该工具支持攻击图的自动生成、更新和调整。在 PDDL“领域”更新时，该工具通过比对场景变化前后生成规划图的差异，对规划路径进行增量式维护。

2.3 图划分算法在攻击图建模领域的应用

随着网络规模的逐步扩大，各攻击图模型难免会遇到不同程度的状态空间爆炸、算法复杂度攀升和结果可展示性降低等问题。将图划分算法引入攻击图模型有助于缓解上述问题。目前图划分算法主要应用于划分目标网络中资产(如设备或服务)的可达性关系以及攻击图分割。

资产的可达性关系划分是以并行方式发现攻击路径的一个重要前序步骤。Kaynar 等[11]把服务的可达性视为一个超图，其中超图内顶点表示单个服务并赋有一定权重，边则表示一组具有相互访问关系的服务，设计超图划分算法以实现各子图负载均衡，同时跨越不同子图的超图边的数量减少，从而提升各代理端的生成效率。Cao 等[12]运用基于多层级 K 路(Multi-level K-way)划分算法的工具 METIS 对网络拓扑图进行划分，结合 Spark 集群计算框架实现了攻击图的并行生成，同时给出各代理端生成攻击图的更新方法。针对攻击图呈现过于复杂的问题，Liu 等[13]提出了利用分支节点的攻击图划分方法，其优势在于可以在不改变攻击图的基本结构的情况下，将攻击图划分为多个小规模子攻击图，既保留了攻击图的全部信息，也提升了其可读性。

总体来看，基于逻辑推理的攻击路径发现方法扩展性较好，其构建体系尤为完整，值得借鉴。但考虑基于智能规划的攻击图模型通过漏洞利用组合的形式呈现攻击路径更为直观，且相关领域建模语言和路径规划算法的成熟度俱佳，因此本文对基于智能规划的攻击路径方法进行研究。文献[3]提及的分层任务网络(Hierarchical Task Network, HTN)算法可融入专家经验，任务网络设计者仅需对规划问题做顶层分析，而忽略底层具体实现细节，从而降低攻击路径规划算法

复杂度,因此本文将其作为实现攻击路径发现的基础算法,并拓展其维护攻击路径的能力。同时,文献[12]和文献[13]的启发,将融合的多层级 K 路划分算法引入所提规划方法,改善其在大规模场景下的任务执行效率。

3 基于智能规划的攻击路径发现框架

本文所提框架由要素层、表征层和规划层构成,如图 1 所示。在要素层主要完成网络建模和漏洞分析两项任务,收集有待评估的资产及其可达性关系信息,同时分析资产自身潜在漏洞,如组件安全弱点和安全策略缺失。在表征层,提取网络模型内涵盖的信息形成问题描述,提取漏洞分析结果形成领域定义。根据规划层所采用的规划算法,选择领域建模语言,实现对"问题"和"领域"的映射与编码。在规划层,首先对编码进行解析,调用规划器,发现当前环境下的攻击路径。其中,表征层和规划层是所提框架的核心,要素层所需信息可借助安全厂商提供的资产探测工具收集。目前,所提框架在表征层可支持 PDDL 模型和 HTN 模型,在规划层可支持两种调用规划器的方式,即设计外部枚举算法和多次调用规划器,每次发现一条路径,直至完成全部攻击路径的枚举,以及调用一次规划器完成全部攻击路径的发现。第一种方式参见文献[8],下一章将对第二种方式进行阐述。

围绕扩展性问题,本文研究路线分解为如下 3 个阶段:1)在划分阶段,关注网络规模扩展,提出融合多层级 K 路算法的拓扑图划分模型;2)在求解阶段,关注专家经验融入,提出基于 HTN 的攻击路径规划方法;3)在维护阶段,关注局部信息更替,提出面向局部信息更替的攻击路径扩展方案。其中,在第二阶段提出的方法是本文的基础,而在其余两个阶段提出的模型和方案分别从路径的生成效率及维护功能角度出发,优化上述框架。

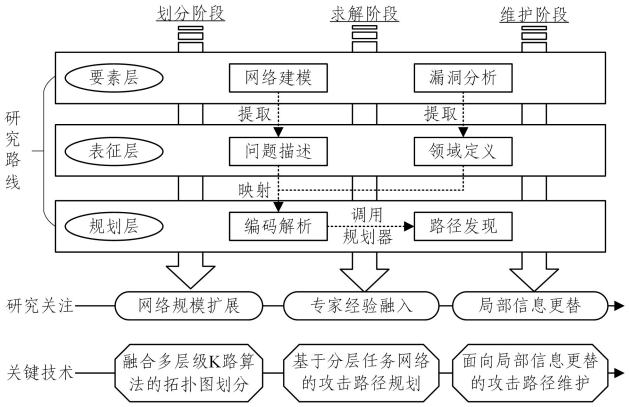


图 1 本文方法整体框架

Fig. 1 Overview of the proposed method

4 基于分层任务网络的攻击路径规划

在介绍基于 HTN 的攻击路径规划方法前,先需明确本文中攻击路径和对应攻击图模型的定义。

定义 1(攻击路径(Attack Path,AP)) 在给定初始条件下,面向特定目标资产的一组利用动作序列 $e_i(i=1,2,\cdots,n)$,即 $ap=\{e_1,e_2,\cdots,e_n\}$。

定义 2(最小攻击图(Minimal Attack Graph,MAG)) 由面向同一目标资产的攻击路径集合所组成的有向图,即 $MAG=\{ap_j\mid j=1,2,\cdots,n\}$。在图模型中,节点为 $e_i,(i=1,2,\cdots,n)$,有向边则表示满足利用动作前、后置条件的有效转移。

定义 2 延续了文献[8]关于 MAG 的描述定义,并假设所有攻击路径具有相同的起点和终点。本文的研究目标为发现固定的起点和终点之间的全部攻击路径。

4.1 分层任务网络形式化表述

HTN 由"领域"和"问题"组成,用于完成特定规划问题求解。相较于经典规划,HTN 引入了"任务网络"和"方法"两个重要概念。HTN 相关定义总结如下[14]:

1)任务网络 TN 由一组任务 T 和任务间的约束 C 组成,记为二元组 $TN=\langle T,C\rangle$。其中,任务又分为复合任务 T_C 和原子任务 T_P,复合任务可以被分解为多个子任务,不能再分解的任务即为原子任务。

2)方法 M 为分解复合任务的策略,由复合任务 T_C、方法前提 Pre_M 和分解后的任务网络 TN_C 组成,记为三元组 $M=\langle T_C,Pre_M,TN_C\rangle$。

3)HTN 领域 D 由状态 S、操作 O 和方法 M 组成,记为三元组 $D=\langle S,O,M\rangle$。其中,状态为描述对象的一组事实谓词,本文对象为网络资产,状态则可以是运行服务、控制权限或访问凭证等;操作是利用动作 $e_i(i=1,2,\cdots,n)$ 的模板,包含利用动作的前置、后置条件。

4)HTN 问题 P 由初始状态 S_0、初始任务网络 TN_0 和领域 D 组成,记为三元组 $P=\langle S_0,TN_0,D\rangle$,用于描述攻击场景。

5)规划 π 是一组当前问题下的利用动作序列,即一条攻击路径 ap。

4.2 攻击路径规划模型

本文提出基于 HTN 的攻击路径规划模型,如图 2 所示。

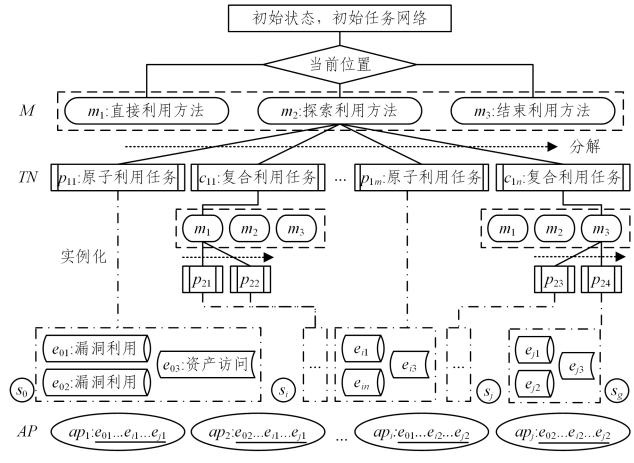


图 2 攻击路径规划模型

Fig. 2 Attack path planning model

分析攻击者所掌握的网络资产信息,得到初始状态和初始任务网络作为模型的输入,输出为面向目标资产的全部攻击路径。该模型包含 3 种方法,即直接利用、探索利用和结束利用。上述方法的前提是已获得攻击者目前所处位置,用可达性关系表征。直接利用表示攻击者已获得目标资产的可达

性信息，可直接开展利用任务；探索利用表示攻击者利用相连资产的漏洞，通过多次横移最终到达目标资产；结束利用表示攻击者已经到达目标资产并完成漏洞利用，无须开展后续利用任务。其中，探索利用方法对任务网络中满足其前提的任务进行分解，得到多个复合利用任务和原子利用任务。

上述分解任务的方法是对目标资产进行“白盒”安全测试过程的一种描述，体现了渗透专家结合网络场景利用资产上已知漏洞的测试经验。此外，相较于现有基于智能规划的攻击路径发现方法，所提方法无须借助外部枚举算法对规划器进行多次调用，而是在所提模型中设置漏洞利用导致资产可被访问的操作，并基于记录的资产访问关系判别是否完成全部路径规划，因此仅需调用一次 HTN 规划器即可获得全部攻击路径。

5 融合多层级 K 路算法的拓扑图划分

本文目标网络中资产之间的服务可达性关系和漏洞分布通过拓扑图模型进行定义。

定义 3（目标网络拓扑图（Target Topology Graph，TTG））表示为一个赋有权重的有向图，记为 $TTG=\langle A, Conn\rangle$。其中 A 为目标网络中资产集合，在图中表示为顶点；$Conn$ 为资产间的服务可达性关系集合，在图中表示为有向边；W_{Vul} 为各资产上运行服务的漏洞数量，表示节点权重；W_{Conn} 为有漏洞服务的可达性关系数量，表示有向边权重。

5.1 网络拓扑图划分流程

本文选用多层级 K 路算法对 TTG 进行划分，如图 3 所示，划分流程由 3 个阶段组成，包括粗化阶段、划分阶段和还原阶段。

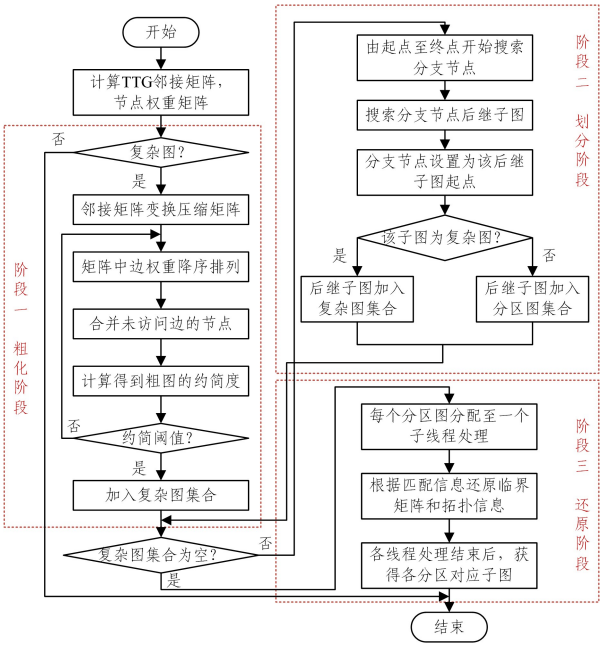


图 3 TTG 划分流程图

Fig. 3 Flow chart of TTG partitioning

在各阶段根据划分需求，可灵活采用不同算法[15]。所提算法在粗化阶段选择重边匹配策略算法，在划分阶段选择分支节点划分算法。粗化后的 TTG 规模更小，分支节点划分算法的计算负担随之减轻。分支节点划分不破坏原有拓扑图的结构，有助于 HTN 的求解。在还原阶段，所提算法可减少子图数量，尽量避免各子图间节点和有向边重复。

5.2 多层级 K 路算法设计

本节对融合多层级 K 路算法的拓扑图划分模型在粗化阶段和划分阶段采用的重边匹配策略算法和分支节点划分算法分别进行设计。

1）重边匹配策略算法

重边匹配策略是在 TTG 粗化过程中多次选择权重较高的边，合并两端节点，压缩有向图的过程。选择重边匹配策略的目的是在粗化后的图中保留权重相对较小的边，这意味着合并后的节点对应的一组漏洞服务对外可达性关系总体数量更少，有助于减少还原后各子图间重复的有向边数量，进而避免规划时对同一有向边重复搜索。

如图 4(a)所示，以文献[8]中示意拓扑图为例，展示重边匹配策略算法的大致过程。在图 4(b)中随机选择一个节点 D_0 作为重边匹配的起点，随后在图 4(c)中通过边排序选取与 D_0 相连的权重最高的有向边 $D_0\rightarrow D_1$，合并得到 D_{01}，同时分别重新计算合并后节点和有向边的权重，最终得到图 4(d)。针对上述过程需指出两点：(1)合并后的节点权重是对应原 TTG 中两个节点权重的求和，合并后有向边权重同理；(2)TTG 中表示攻击者的初始节点(如图 4 中 AT)和目标节点(如图 4 中 D_3)不参与重边匹配过程。

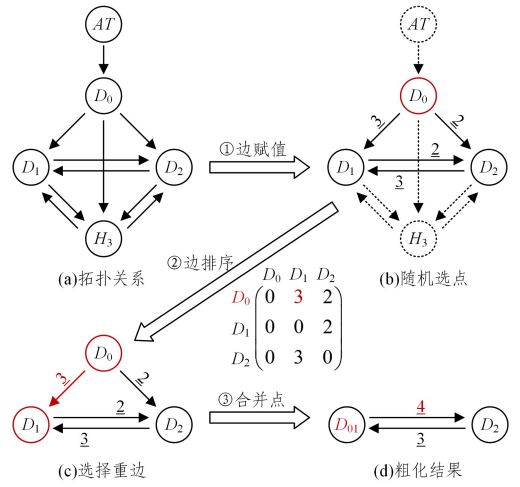


图 4 重边匹配策略示意图

Fig. 4 Heavy edge matching strategy diagram

2）分支节点划分算法

在划分阶段，本文参考文献[13]的分支节点划分算法划分粗化后的图，得到多个子图。分支节点为 TTG 出度数量大于 1 的节点。在攻陷分支节点对应资产后，攻击者有多种利用选择，对应图中多条攻击路径。考虑 HTN 的回溯机制，重新定义图的复杂度 ϕ，如下：

$$\phi=\begin{cases}|A|+\sum\limits_{i\in A}W_{\text{Vul}}(i)+\sum\limits_{j\in A\times A}W_{\text{Conn}}(j), & \theta\leqslant 2\\ |A|\times\sum\limits_{j\in A\times A}W_{\text{Conn}}(j), & \theta>2\end{cases}\tag{1}$$

$$\theta=\frac{1}{|A|}\sum_{j\in A\times A}W_{\text{Conn}}(j)$$

当 θ 较大时，说明 HTN 规划过程中需对子图中多个

节点及有向边进行回溯，执行多次“领域”中定义的操作，此时复杂度与节点和有向边的全部排列数量有关；反之，当 θ 较小时，复杂度仅由子图对应 HTN“问题”的规模决定，即资产、漏洞和服务可达性关系三者总数。子图是否需要再划分同样由式(1)计算的复杂度决定。

需指出，不同于传统多层级 K 路算法求解问题，在划分阶段引入分支节点划分算法会让部分节点和有向边被同时划分至不同子图。此外，粗化阶段未处理的攻击初始节点和目标节点也需分配至各子图。上述两步处理的结合，使得对还原后的子图进行攻击路径规划结果具备完备性，同时与文献[13]中多个子攻击图的相互独立以提升攻击图可读性的目标一致。

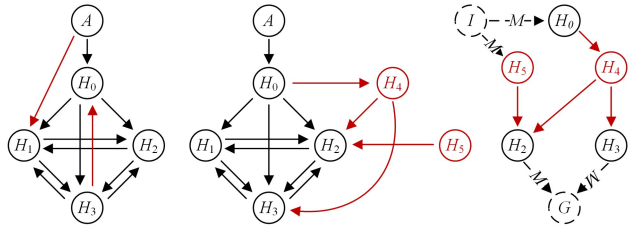
6 面向局部信息更替的攻击路径维护

由定义 2 可知，MAG 是典型的状态攻击图，其优势在于它显式地展示全部攻击路径。然而随着目标网络规模扩大，攻击路径数量也呈指数型增长，导致目标网络中资产相关信息变化时，扩展攻击路径变得愈加困难。针对上述问题，本文首先将 MAG 转换为扩展性更好的属性攻击图，随后给出攻击路径维护方案。

属性攻击图包含两类节点，即属性节点和利用节点，其中属性节点表示攻击者可以利用的条件，指向利用节点的有向边表示前置条件被满足，指向属性节点的有向边则表示利用完成得到后置条件[16]。属性攻击图相较于状态攻击图更紧凑，同时在攻击图模型中引入属性节点为所提局部攻击路径扩展方法提供了可能性。

本文针对目标网络中资产的局部信息更替，包括设备、漏洞和服务可达性的增加或删减，提出了基于属性攻击图的路径维护方案。该方案对局部信息进行增加和删减情形，处理方式不同，面向局部信息增加的维护描述如下：1)将 MAG 转换为属性攻击图；2)分析信息更替的资产，在 TTG 中定位所涉及的节点和有向边；3)围绕上述节点和有向边抽取多个局部拓扑图；4)分析各局部拓扑图，更新 HTN 的“领域”和“问题”，判别是否需要划分，再调用规划器，获得局部攻击路径；5)分析上述攻击路径的利用节点的前置、后置条件后，将新增条件扩展至原有属性攻击图中。针对局部信息删减的情形，仅需删除原属性攻击图中的属性节点和相应有向边即可。如同时出现增加和删减信息的复杂情形，优先处理删减相关任务，随后处理增加相关任务。

上述局部拓扑图的抽取是所提方案的关键步骤。本章仍采用文献[8]的示意拓扑为例展示抽取过程。由于删减局部信息处理过程较为简单，此处着重展示新增局部信息情形下的局部拓扑图抽取，说明扩展攻击路径过程。在图 5(a)中，新增 $AT \rightarrow D_1$ 和 $D_3 \rightarrow D_0$ 这两处服务可达性关系，因未构成联通的局部图，故仅需分别调用规划器，获得攻击路径，扩展至原属性攻击图中；在图 5(b)，新增设备 D_4 和 D_5，因增加了 5 处服务可达性关系，故抽取的局部拓扑图中节点相互联通，但存在多个起点和终点。考虑 HTN 规划器的求解特点，为上述局部拓扑图新增虚拟的共同起点、终点，以及虚拟的有向边，并标记为“M”，如图 5(c)所示。算法 1 给出了该情形下攻击路径扩展的具体步骤。通过上述局部拓扑图的处理，可减少维护过程中调用规划器的次数。



(a)新增服务可达关系 (b)新增威胁设备 (c)局部拓扑抽取处理

图 5 局部拓扑抽取示意图

Fig. 5 Local topology graph extraction diagram

算法 1 局部扩展方法

输入：新增设备 devices，属性攻击图 att_ag，网络拓扑图 topo

输出：扩展后网络拓扑图 exp_topo，扩展后攻击图 exp_ag

```
1. init_info(exp_topo,exp_ag) / * 初始化 * /
2. While true do
3.     devices←getUpd_Dev(att_ag,devices) / * 设备信息更新 * /
4.     topo,part_topos←getPart_Topo(devices,att_ag,topo) / * 针对
       新增设备生成局部拓朴图 * /
5.     sub_graphs←multi_K_way(part_topos) / * 划分局部拓扑图 * /
6.     part_att_graphs←genAtt_Ag(sub_graphs) / * 生成局部攻击图 * /
7.     mergeAtt_ag(att_ag,part_att_graphs) / * 合并攻击图 * /
8.     If not hasNew_att(att_ag) then
9.       break / * 设备无新增属性信息，扩展结束 * /
10. exp_ag←att_ag
11. exp_topo←topo
12. return exp_ag,exp_topo
13. Function getPart_Topo(devices,att_ag,topo)
14.     devices,vuls,conns←init_info(devices) / * 获取设备、漏洞和服
        务可达性关系 * /
15.     pre_devices,succ_devices,vuls←addNeigh_ devices(topo,de-
        vices,att_ag) / * 获取新增设备的前驱设备和后继设备，存储后
        继设备的漏洞，同时更新现有拓扑图内设备信息 * /
16.     pre_devices,vuls←updVul(M_label,pre_ devices,vuls) / * 为
        前驱设备漏洞对应有向边添加 M 标记 * /
17.     init(new_comb_start,new_comb_end)
18.     If getNum(pre_devices) > 1 then
19.       devices,conns,new_comb_start←addNew _Start(devices) /
          * 为多个前驱设备添加共同起点 * /
20.     If getNum(succ_devices) > 1 then
21.       devices,conns,new_comb_end←addNew _End(devices) / *
          为多个后继设备添加共同终点 * /
22.     part_topo←updTopo(new_comb_start,new_comb_end,devices,
        vuls,conns) / * 获取局部拓扑图 * /
23. return topo,part_topo
```

7 实验验证

本文从攻击路径发现的正确性和扩展性两方面开展相关实验验证。首先，以一个典型工业控制网络场景为例，借助图形化的形式，展示攻击路径规划和面向局部信息更替的攻击路径维护结果。其次，构造大规模验证网络，通过与现有基于智能规划的攻击路径发现方法对比，评估所提方法的扩展

性能。实验运行环境为 VMware ESXI 平台上创建 Ubuntu 20.04.1 LTS 实例，配置为 4vCPU 和 16GB 运行内存。

7.1 攻击路径发现

本文验证攻击路径发现方法正确性的网络拓扑如图 6 所示，这是一个典型的工业控制网络，由过程控制网络、自动控制网络和现场控制网络组成，包括 OPC(OLE for Process Control)服务器、工程师站、操作员站、人机接口设备(Human Machine Interface，HMI)和可编程逻辑控制器(Programmable Logic Controller，PLC)，用于对现场控制网络中设备进行数据采集与控制。

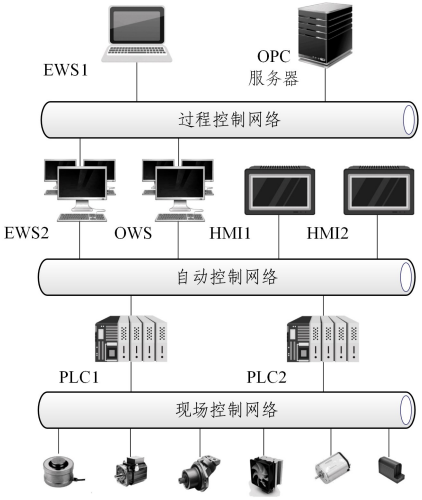


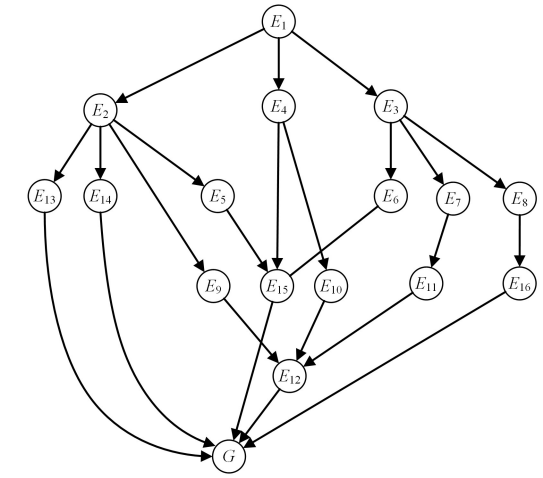
图 6 工业控制网络拓扑示例

Fig. 6 Example topology of typical industrial control network

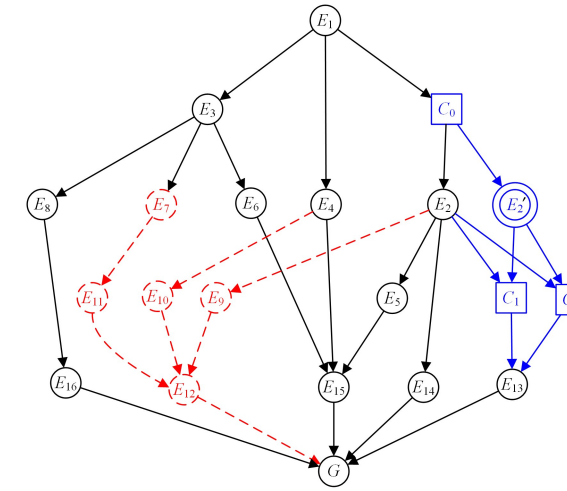
假设攻击者渗透进目标网络的入口为 OPC 服务器，以破坏现场控制网络的仪器仪表进而影响物理生产过程为攻击目标。本文提出的基于 HTN 的攻击路径规划方法基于一款 Python 语言实现的 SHOP(Simple Hierarchical Ordered Planner)规划器，即 Pyhop[17](V1.2.2)。通过将图 6 示例网络的设备、服务可达性关系、漏洞及利用等相关信息进行编码，输入规划器，得到 9 条攻击路径，调用 Graphviz(V2.43.0)展示上述路径组成的攻击图，如图 7(a)所示。

此外，为展示基于属性攻击图的扩展结果，本文考虑如下两个信息更替场景，即已有漏洞修复和新增设备，如图 7(b)所示。由于 PLC1 设备上的漏洞(ID:CVE-2022-38465)被修复，因此阻断了 3 条攻击路径，如图中深红色虚线区域所示；新增与 EWS1 相同的设备后，局部扩展的攻击路径，如图中蓝色实线区域所示。

在图 7(b)中，仅漏洞(ID:CVE-2021-37172)对应的节点 E_2 和 E_2 按照属性攻击图的节点形式给出其属性节点(如图中的属性节点 C_0，C_1 和 C_2)，为清晰展示，余下利用的属性节点已经省略。表 1 详细列了图 6 中设备、漏洞和相关利用等信息，并对图 7 中攻击路径上的利用节点进行注释，其中结合影响组件和漏洞信息给出利用完成后的设备访问关系，综合体现在攻击路径的节点上，如 $E2(D1,D2)$ 表示设备 $D1$ 利用 $E2$ 实现对设备 $D2$ 的访问。



(a)攻击路径规划



(b)攻击路径维护

图 7 攻击图示例(电子版为彩图)

Fig. 7 Example of attack graph

表 1 设备漏洞分布及图 7 中攻击路径节点说明

Table 1 Device vulnerability distribution and illustration of nodes on all attack paths

设备	简称	漏洞编号	影响组件	攻击路径节点
OPCServer	D_1	CVE-2020-29457	OPC UA. NET 1.4.363.107	$E_1(A,D_1)$
EWS1	D_2	CVE-2020-0796	Windows SMBv3，TIA Portal V13	$E_2(D_1,D_2)$
EWS2	D_3	CVE-2019-10918	SIMATIC WinCC V15	$E_3(D_1,D_3)$
OWS	D_4	CVE-2021-44228	ApacDe Log4j2 2.0	$E_4(D_1,D_4)$，$E_5(D_2,D_4)$，$E_6(D_3,D_4)$
HMI1	D_5	CVE-2020-15798	SIMATIC HMI Comfort Panels V15.0	$E_7(D_3,D_5)$
HMI2	D_6	CVE-2020-15798	SIMATIC HMI Comfort Panels V15.1	$E_8(D_3,D_6)$
PLC1	D_7	CVE-2020-15782	SIMATIC S7-1500	$E_9(D_2,D_7)$，$E_{10}(D_4,D_7)$，$E_{11}(D_5,D_7)$
		CVE-2022-38465	SIMATIC S7-1500	$E_{12}(D_7,D_7)$
PLC2	D_8	CVE-2021-37172	SIMATIC S7-1200	$E_{13}(D_2,D_8)$
		CVE-2021-44694	SIMATIC S7-1200	$E_{14}(D_2,D_8)$，$E_{15}(D_4,D_8)$，$E_{16}(D_6,D_8)$

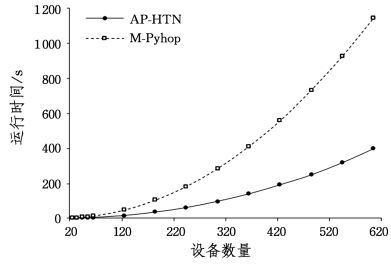
7.2 扩展性能评估

本文验证攻击路径发现方法扩展性的网络拓扑仍取自文献[8]，以此为基础进行多倍扩展，生成多种规模的测试网络，其中最大规模的网络包含 1 005 台设备。在第 3 章所述的基于智能规划的攻击路径发现框架下，通过集成封装 5 类规划算法的规划器，用于对不同规模测试网络进行攻击路径发现。除了 Pyhop，典型的规划器还包括 FF(Fast Forward)[18]，SGPlan5[19]，FD(Fast Downward)[20] 和 POPF2[21]，其中前三款规划器已被用于攻击路径发现相关任务。此外，基于偏序规划的模型同样可用于攻击路径发现任务[3]，因而本文补充 POPF2，分析其运行结果。

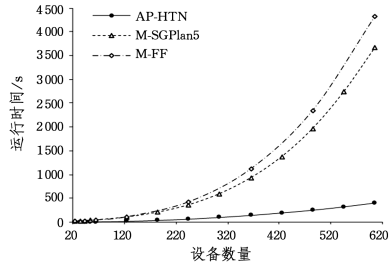
如第 3 章所述，本文所提框架支持两种规划器的调用方式，其中需要借助外部枚举算法的规划器包括 FF，SGPlan5，FD(A^*)和 POPF2。FD(A^*)表示规划过程中选用 A^* 算法找寻最短攻击路径。而 HTN 规划器也可以通过设计任务网络实现单次找寻最短攻击路径，进而本文实现了一种采用外部枚举算法调用 Pyhop 的攻击路径发现方法。另一方面，FD(K^*)表示 FD 选用 K^* 算法找到 K 条攻击路径，更接近本文第 4 章所提方法。当 K 设置为较大阈值时，可用于发现测试网络下的更多攻击路径。本文中 $K=1\,000$。

针对不同测试网络，分别生成“领域”和“问题”编码文件，FF，SGPlan5，FD(A^*)和 POPF2 均采用 PDDL 作为输入，而 Pyhop 遵循 HTN 语法，采用内嵌函数的形式作为输入。文献[8]提供的测试网络内含 5 台设备和 8 个漏洞。首先，在测试网络中设备上的漏洞分布不变，逐步扩展该网络中测试设备数量，每次扩展增加 2 台设备，用于形成不同规模(即设备数量)的测试网络，当测试网络规模达到 605 台时，停止向该网络中添加设备。随机选取其中 8 组测试结果用于评估所提方法的扩展性能并展示。在保证不同规划器输出相同数量攻击路径的前提下，收集规划器运行时间和最大内存占用数据分别如图 8 和表 2 所示。本文第 4 章所提规划算法记为 AP-HTN，其他借助外部枚举算法的规划器记为“M-规划器名字”。实验中各规划器的运行超时时间为 4 700 s。

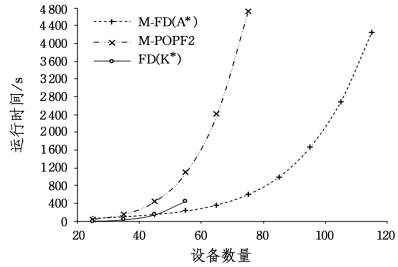
通过对比发现，两种基于 HTN 的攻击路径发现算法相较于其他以 PDDL 作为输入的规划器所需运行时间更短，而本文所提 AP-HTN 在运行时间和内存占用方面均表现良好。反观 M-FD(A^*)，M-POPF2 和 FD(K^*)这 3 款规划器，均无法在规模大于 125 台的测试网络中完成攻击路径发现任务；在测试网络设备规模为 65 时，FD(K^*)的最大内存占用超过了运行环境的内存限制，且无法找到全部攻击路径；M-POPF 的最大内存占用激增，同时运行超时。特别指出，如图 8(a)所示，本文所提规划算法在含 605 台设备的测试网络中所需运行时间也呈小幅度指数型增长，证明了本文引入 TTG 图划分算法的必要性。



(a)基于 HTN 的攻击路径规划算法对比



(b)与基于 PDDL 的攻击路径规划算法对比



(c)与基于 PDDL 的攻击路径规划算法对比

图 8 多款规划器发现攻击路径的运行时间对比

Fig. 8 Running time comparison of multiple planners for attack path discovery

表 2 多款规划器用于发现攻击路径的最大内存占用对比

Table 2 Maximum memory usage comparison of multiple planners for attack path discovery

(单位：MB)

规模	AP-HTN	M-Pyhop	M-SGPlan5	M-FF	M-FD(A^*)	FD(K^*)	M-POPF2
25	0.68	1.80	14.05	12.46	69.83	7.47	52.28
45	2.02	5.95	27.44	24.86	157.67	448.27	451.05
65	4.20	12.46	42.43	39.40	369.41	—	2411.21
125	15.45	46.93	103.30	104.86	—	—	—
245	60.79	182.92	359.81	419.69	—	—	—
365	138.48	410.77	927.70	1 112.08	—	—	—
485	246.63	732.45	1 968.08	2 344.21	—	—	—
605	395.54	1 143.10	3 664.10	4 316.72	—	—	—

其次，面向较大规模的测试网络，本文在 AP-HTN 中引入多层级 K 路算法，其中粗化阶段的重边匹配策略算法参考开源项目[22]的相关实现，粗图的约简阈值设置为 0.6；划分阶段的分支节点算法用于处理粗化后的 TTG，复杂度阈值设置为 24；在还原阶段采用多线程的方式细化各子图并还原节点和有向边的相关信息，最大线程数量设置为 20。

如表 3 所列，选取测试网络的设备数量为 1 005 时，将本文划分方法与文献[13]的分支节点划分方法进行性能比较，

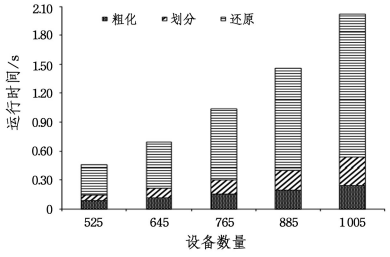
发现本文划分方法可以得到更少的子图，同时各子图间重复节点和有向边的数量大幅缩减。

表 3 TTG 划分性能对比(设备数量:1005)

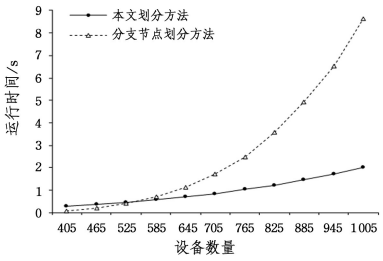
Table 3 TTG partitioning performance comparison (Number of devices:1005)

统计项	分支节点划分方法	本文划分方法
子图个数	1003	252
重复节点个数	2004	503
重复有向边条数	4008	1529

图 9(a)展示了本文划分方法各阶段的运行时间；由图 9(b)可知，当测试网络设备数量少于 525 时，分支节点划分方法在运行时间上占据优势，而随着设备数量递增，在更大规模的测试网络中，本文所提划分方法运行时间增幅不明显。



(a)本文划分方法三阶段运行时间

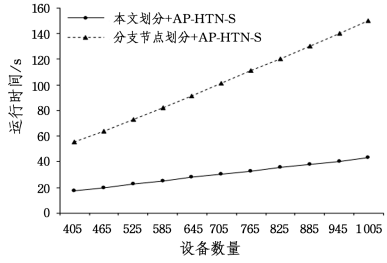


(b)本文划分方法与分支节点划分方法的运行时间对比

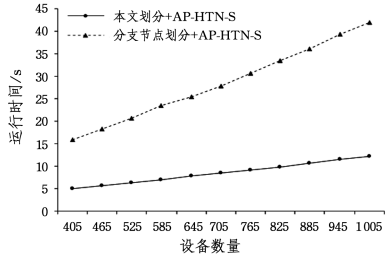
图 9 本文划分算法的性能统计

Fig. 9 Performance statistics of the proposed method

基于上述划分结果，进一步观察上述两种划分方法对攻击路径发现任务的执行效率的影响。如图 10(a)所示，在本文攻击路径规划算法中引入上述两种划分方法，使得攻击路径发现时间整体呈现线性递增趋势，而本文所提划分方法辅助规划器在执行攻击路径发现任务时所需运行时间更少。原因在于本文划分方法尽可能减少子图数和重复节点、有向边的数量，有助于避免规划器的重复求解，进而提升了发现全部攻击路径的执行效率。如相关工作所述，以并行方式发现攻击路径的前提是对 TTG 进行有效的划分。本文利用上述两个划分方法所得结果，采用多线程的形式模拟并行过程，并行攻击路径发现性能结果展示在图 10(b)中，本文所提路径规划算法在近千点规模的测试网络中运行时间缩短至约 10 s。



(a)串行攻击路径发现



(b)并行攻击路径发现

图 10 TTG 划分后攻击路径发现方法的运行时间

Fig. 10 Running time of the proposed attack path discovery method after TTG partitioning

最后，关注本文所提面向局部信息更替的攻击路径维护性能，运行结果如表 4 所列。参照文献[7]的性能评估方法，将本文所提方法与全局攻击路径维护方法的性能进行对比，分别统计测试网络中包含 405 台和 1005 台设备时，增加或删减一个设备、漏洞和服务可达性关系的运行时间和最大内存占用情况。通过对比可知，本文采用属性攻击图维护攻击路径的局部维护方法在删减和增加方面性能更具优势。

表 4 本文路径扩展方法的性能对比

Table 4 Performance comparison of the proposed attack path extension

规模	扩展方案	性能指标	删减设备	增加设备	删减漏洞	增加漏洞	删减可达关系	增加可达关系
405	全局	运行时间/s	17.62	18.12	17.69	17.67	17.66	17.79
		最大内存占用/MB	171.68	164.61	171.53	171.48	172.38	171.15
	局部	运行时间/s	5.72	0.67	5.97	0.68	5.75	0.69
		最大内存占用/MB	42.08	39.65	41.89	39.58	42.05	39.68
1005	全局	运行时间/s	45.75	45.70	46.09	45.20	45.31	46.04
		最大内存占用/MB	691.47	692.03	692.62	692.17	692.12	697.11
	局部	运行时间/s	35.12	1.38	34.89	1.38	34.78	1.40
		最大内存占用/MB	78.72	72.88	78.14	72.88	78.97	73.01

结束语 本文提出了一种基于 HTN 的攻击路径规划方法，贡献如下：1)引入融合多层级 K 路算法的拓扑图划分模型，提升了大规模网络下的攻击路径生成效率；2)构建攻击路径规划分层任务网络，通过融入专家经验降低了相关算法复杂度；3)设计基于属性攻击图的攻击路径维护方案，避免了全局重新生成攻击图的计算资源消耗。实验结果表明，本文划分方法相较于分支节点划分方法可以得到数量较少的子图，且各子图间重复的节点和有向边的数量更少；所提方法相较于同类基于智能规划的攻击路径发现方法具备良好的扩展性，可用于在大规模网络中完成攻击路径发现任务。然而，

本文所提方法与在相关安全领域得以全面推广的 MulVAL 框架相比,在计算资源消耗和路径维护等方面仍有些许差距。为此,在未来工作中将采用不同的拓扑划分方法,重点突破还原阶段处理子图方法的性能瓶颈;优化 HTN 内部求解算法,通过加入以目标为导向的启发式函数,构建攻击目标任务网络,对利用动作空间进行快速约简。此外,还将考虑待评估网络存在的不确定性,为所提攻击路径规划方法增添"重规划"的能力。

参 考 文 献

[1] LALLIE H S,DEBATTISTA K,BAL J. A review of attack graph and attack tree visual syntax in cyber security[J]. Computer Science Review,2020,35:100219-100259.

[2] ZENITANI K. Attack graph analysis:an explanatory guide[J]. Computers & Security,2022,126:103081-103101.

[3] ZANG Y C,ZHU T Y,ZHU J H,et al. Domain-independent intelligent planning technology and its application to automated penetration testing oriented attack path discovery[J]. Journal of Electronics & Information Technology, 2020, 42(9): 2095-2107.

[4] OU X,GOVINDAVAJHALA S,APPEL A W. MulVAL:A Logic-based Network Security Analyzer[C]// USENIX security symposium. 2005:113-128.

[5] INOKUCHI M,OHTA Y,KINOSHITA S,et al. Design procedure of knowledge base for practical attack graph generation [C]// Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. 2019:594-601.

[6] STAN O,BITTON R,EZRETS M,et al. Extending Attack Graphs to Represent Cyber-Attacks in Communication Protocols and Modern IT Networks[J]. IEEE Transactions on Dependable and Secure Computing,2022,19(3):1936-1954.

[7] SAHA D. Extending logical attack graphs for efficient vulnerability analysis[C]// Proceedings of the 15th ACM Conference on Computer and Communications Security. 2008:63-74.

[8] GHOSH N,GHOSH S K. A planner-based approach to generate and analyze minimal attack graph [J]. Applied Intelligence, 2012,36:369-390.

[9] GAO W L,ZHOU T Y,ZHU J H,et al. Network attack path discovery method based on bidirectional ant colony algorithm [J]. Computer Science,2022,49(S1):516-522.

[10] BEZAWADA B,RAY I,TIWARY K. AGBuilder:an AI tool for automated attack graph building,analysis,and refinement[C]// IFIP Annual Conference on Data and Applications Security and Privacy. Cham:Springer,2019:23-42.

[11] KAYNAR K,SIVRIKAYA F. Distributed attack graph generation[J]. IEEE Transactions on Dependable and Secure Computing,2015,13(5):519-532.

[12] CAO N,LV K,HU C. An attack graph generation method based on parallel computing[C]// Science of Cyber Security:First International Conference. Springer International Publishing,2018:34-48.

[13] LIU Y Z,CHEN Y Z,GUO K,et al. Distributed process mining and graph segmentation for network attack modeling [J]. Journal of Chinese Mini-Micro Computer Systems, 2020, 41(8): 1732-1740.

[14] SHAO T H,ZHANG H J,CHENG K,et al. Review of replanning in hierarchical task network [J]. Journal of Systems Engineering and Electronics,2020,42(12):2833-2846.

[15] HERRMANN J,KHO J,UÇAR B,et al. Acyclic partitioning of large directed acyclic graphs[C]// 2017 17th IEEE/ACM International Symposium on Cluster,Cloud and Grid Computing(CCGRID). IEEE,2017:371-380.

[16] CHEN F,ZHANG Y,SU J S,et al. Two Formal Analyses of Attack Graphs [J]. Journal of Software,2010,21(4):838-848.

[17] Simple Hierarchical Ordered Planner(SHOP) [EB/OL]. https://www.cs.umd.edu/projects/shop/.

[18] Fast Forward(FF) [EB/OL]. https://fai.cs.uni-saarland.de/hoffmann/ff.html.

[19] SGPlan(Version 5.0)[EB/OL]. https://wah.cse.cuhk.edu.hk/wah/programs/SGPlan/.

[20] Fast Downward(FD) [EB/OL]. https://github.com/aibasel/downward.

[21] Partial Order Planning Forwards(POPF)(Version 2.0) [EB/OL]. https://nms.kcl.ac.uk/planning/software/popf.html.

[22] Heavy-edge matching [EB/OL] https://github.com/loukasa/graph-coarsening.

WANG Zibo, born in 1992, postgraduate. His main research interests include industrial Internet security and industrial control system security assessment.

WANG Bailing, born in 1978, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include cyber security, information content security and industrial Internet security.

(责任编辑:何杨)