计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200800174

基于多 Agent 联合决策的队组协同攻击规划

周天阳　曾子懿　臧艺超　王清贤
战略支援部队信息工程大学　郑州 450001
国家数字交换系统工程技术研究中心　郑州 450001

摘　要　自动化渗透测试通过将人工找寻可能攻击路径的过程自动化，可大幅降低渗透测试的成本。现有方法主要利用单一 Agent 执行攻击任务，导致攻击动作执行耗时长，渗透效率不高；若考虑多个 Agent 协同攻击，由于每个 Agent 的局部状态有多个维度，总的规划问题的状态空间会呈指数级增长。针对上述问题，提出了基于多 Agent 联合决策的队组协同攻击规划方法。该方法首先将多 Agent 协同攻击路径规划问题转化为联合决策约束下的攻击目标分配问题，建立多 Agent 集中决策模式；然后以 CDSO-CAP 为模型基础，利用联合决策矢量矩阵 JDVM 计算渗透攻击奖励，并采用贪婪策略搜索多 Agent 的最优攻击目标。实验结果表明，与单 Agent 规划方法相比，该方法的收敛性相近，但执行轮次更短，更适合在多目标网络场景内进行快速攻击规划。

关键词：渗透测试；自动化；攻击规划；智能体；联合决策；队组协同
中图法分类号　TP393.08

Team Cooperative Attack Planning Based on Multi-agent Joint Decision

ZHOU Tian-yang，ZENG Zi-yi，ZANG Yi-chao and WANG Qing-xian
Information Engineering University，Zhengzhou 450001，China
National Digital Switching System Engineering & Technological Research Center，Zhengzhou 450001，China

Abstract　Automated penetration testing can greatly reduce the cost of penetration testing by automating the process of manually finding possible attack paths. Existing methods mainly use a single agent to perform attack tasks，which leads to long execution of attack actions and low penetration efficiency. If multi-agent cooperative attack is considered，the state space scale of planning problem will grow exponentially due to the multi-dimensional local state of each agent. Aiming at the above problems，a team cooperative attack planning method based on multi-agent jointdecision is proposed. Firstly，the multi-agent cooperative attack path planning problem is transformed into the attack target assignment problem under the jointdecision constraints，and the multi-agent centralized decision-making mode is established. Secondly，the joint decision vector matrix JDVM is used to calculate the penetration attack reward based on the CDSO-CAP model，and the greedy strategy is used to search the optimal target of attack. The experimental results show that compared with the single agent planning method，the proposed method has similar algorithm convergence with shorter execution rounds. Thus it is more suitable for rapid attack planning in multi-target network scenarios.

Keywords　Penetration test，Automation，Attack planning，Agent，Joint decision，Team collaboration

1　引言

随着互联网深入教育、经济、生活的方方面面，网络安全问题带来的影响越来越深远。及时发现网络安全问题，在事前对网络进行安全性加固是必要的。作为网络安全性测试的一种重要方法，渗透测试通过寻找系统中存在的漏洞，常被用于检验机构或组织的网络防御机制是否正常运作。自动化渗透测试，通过将人工找寻可能的攻击路径过程自动化，可大幅降低渗透测试的成本，近年来成为了相关领域研究的热点。例如，Obes 等[1]将渗透方案以 PDDL 语言描述，并采用了经典的规划算法来进行攻击路径规划；Sarraute 等[2]将概率思想引入经典的规划算法，提出了一种在非确定性攻击场景中找到最佳攻击路径的方法，随后利用部分可观察的马尔可夫决策过程(POMDP)对该方法进行改进[3]；Shmaryahu 等[4]将渗透测试建模为部分可观察配额问题，提出了一种配额规划树算法来规划攻击路径。

在自动化渗透测试的渗透攻击过程中，可能同时存在位于不同网段的多个目标主机。上述方法利用单一 Agent 执行攻击任务，但由于渗透攻击耗时长，难以在规定的时间内完成渗透任务，因此测试效率不高。事实上，可以利用多个 Agent 对目标网络同时展开渗透攻击，使攻击执行过程并行化，但如果多个 Agent 进行自主规划，由于每个 Agent 的局部状态有多个维度，总的规划问题的状态空间会呈指数级增长，同时各自规划的结果极易存在目标冲突、动作冲突等问题，多条

到稿日期：2020-08-27　返修日期：2020-10-30
通信作者：周天阳(aipteamzhouty@aliyun.com)

Agent 攻击路径交叉将使攻击陷入混乱，导致任务失败。

针对上述问题，本文首先从分析实现多 Agent 攻击规划要解决的主要问题及现有方法的不足出发，提出了两级攻击规划解决思路，设计了一种基于“集中决策和同步行动”的多 Agent 协同攻击规划模式 CDSO-CAP；然后，构建了 CDSO-CAP 的形式化模型，并在此基础上提出利用联合决策矢量矩阵对多 Agent 协同攻击规划进行分层优化的求解算法；最后，通过仿真对比实验，从规划步数收敛性和任务分配合理性等方面对模型和算法的有效性进行验证。

2 相关工作

多 Agent 自动协同规划是人工智能领域研究的热点问题。一种研究思路是利用合作博弈的方法来解决多 Agent 协同规划问题。该类方法的主要观点是，Agent 独立动作时会对环境产生影响，每个 Agent 在规划决策时要观察和权衡其他的 Agent 动作对自己造成的潜在影响，由此多 Agent 之间的协同规划问题转变为多 Agent 之间的相互决策与策略优化问题。一种经典的解决方法是基于范式(Normal Form)[5]对多 Agent 的策略交互进行形式化描述，假定每个 Agent 都有自己的行动策略集，以矩阵的方式列出所有可能的 Agent 行动策略组合，定义联合行动的效用函数，并以此约束 Agent 的策略选取，每个 Agent 在采取行动使自身收益最大化时必须考虑其他 Agent 的策略，以确保总体收益的最大化。当没有任何一个 Agent 可以通过改变自己的策略获得更大的收益时，整个多方博弈达到纳什均衡，此时每个 Agent 的策略即为最佳的联合行动策略[5]。经典合作博弈描述了多 Agent 的策略选择方式，但是无法详细给出策略的具体生成过程。如果每个 Agent 的策略对应的是一个具体的行为或动作，则博弈只能给出单步规划的结果，仅限于解决“猎鹿博弈”等简单问题。对于更为复杂的多步连续规划问题，通常采用扩展博弈树来进行问题分析。扩展博弈树是将多步规划过程描述为树状结构，每条树枝代表一个 Agent 的策略选择，每个节点代表 Agent 所处的博弈状态，树中的每一层代表一轮次博弈局面，树的叶子节点则代表一个 Agent 的最终的策略选择。可见，随着博弈轮次的不断增加，树的规模将呈指数级增长，这显然不利于复杂问题的求解。

另一种研究思路是利用马尔可夫决策过程的模型来解决多 Agent 协同规划问题。该类研究的基本观点是，多 Agent 在选取联合行动策略时仅与所在系统的当前状态有关，而与历史状态无关，多 Agent 协同规划过程是从一个系统状态向另一个系统状态转移的最优决策过程[6]。由此多 Agent 协同规划问题转变为多 Agent 联合行动的序贯决策问题(马尔可夫决策问题)，这本质上与解决单 Agent 规划问题的思路是一样的。基于多智能体的马尔可夫决策过程(Multi-agent Markov Decision Process，MMDP)模型与 MDP 模型的基本结构和工作原理是一致的，其区别仅在于 MMDP 的动作是多智能体的联合行动，且协同状态信息可被所有 Agent 获取[7]。但是，MMDP 方法也存在着与扩展博弈树相似的问题——“联合状态空间爆炸”，即联合状态空间与 Agent 及其状态、动作的数量相关，每个 Agent 的状态空间取决于相关描述的维度数量，每个 Agent 在局部独立规划时都有多个维度，总的状态空间则会随着 Agent 数量的增加呈指数级增长。

针对多 Agent 协同规划中的“状态空间爆炸”问题，许多规划领域的研究都提出了分解和降维问题规模的策略。一种典型的方法是分层任务网络(Hierarchical Task Network，HTN)规划，该方法将原始任务(Goal Task)分解为复合任务(Compound Task)以及更细粒度的原子任务(Primitive)。其中，原始任务表示最终的目标状态；复合任务是原始任务的有机组件，但需要多个原子任务共同完成；原子任务是由直接可执行的原子动作完成的最小任务。“任务分层”的规划思想通过将整体规划任务转化、分解为不同的抽象层次任务，避免了全局状态组合爆炸问题，有效降低了规划的复杂度，提高了多 Agent 联合规划的效率。例如，Santiago 等[8]在复杂的实时战略游戏中引入了一种对抗分层任务网络算法 AHTN(Adversarial Hierarchical Task Network)来降低两个 Agent 参与的对手博弈的状态空间规模，从而提高规划效率。Carlos 等[9]为降低基于 POMDP 的攻击规划问题规模，引入 4AL 算法将目标网络逐层分解为有向无环图、全连通节点、同子网节点等不同层级，在最低层级利用 POMDP 进行攻击规划，最后整合形成攻击路径。但 4AL 方法仅针对单 Agent 的攻击规划，无法用于多 Agent 的并行执行和动作协调。“任务分层”类方法的缺点是，规划域问题的分解依赖于良好的领域专家知识和对规划问题本身的理解，且分解方案依赖手工定制，自动化程度不高。

另外，多 Agent 协同规划的质量还依赖于系统状态信息的完整性。由于每个 Agent 分布在不同的系统位置，仅能观察到局部或本地的状态信息，为了保证自身与队组行动的一致性，因此必须获得完整的系统状态信息。为此，需要建立良好的多 Agent 交互模型和通信机制来实现 Agent 之间的信息同步，这正是基于联合决策的多 Agent 建模仿真的重要内容。Igor[10]从网络战争建模的角度，分别为 DDoS 攻击者和 DDoS 防御者进行了多 Agent 建模，在双方模型中定义了不同的攻防 Agent 角色，同时根据攻击或防御的技术实现机理，为 Agent 角色的命令控制机制、任务指派和行动响应定义了形式化的行为交互模型与通信协议。由于多 Agent 之间实时的信息同步会产生额外的流量和带宽消耗，对于 DDoS 和渗透攻击都将暴露 Agent 的行踪。为了减少信息交互，POMDP[11-12]、基于成本的通信决策[10]等方法还被用于提高 Agent 在局部信息条件下的联合决策能力。

总的来看，在基于联合决策的队组协同攻击规划问题的求解上，既可以借鉴序贯决策的思想，也可以参考合作博弈的问题求解方法，具体策略是：在多 Agent 整体规划上采取马尔可夫决策状态转移模型方法，而在每一轮的决策上采用与合作博弈中联合策略矩阵类似的方法进行攻击策略选取。同时，可参考 HTN 分层任务的思想分解攻击规划问题来降低多 Agent 协同攻击规划的状态空间。

3 多 Agent 协同攻击规划模式

本节首先讨论了自动化渗透测试中多目标网络攻击的场景特点，提出了两级攻击规划的解决思路；在此基础上，介绍了

一种多 Agent“集中决策和同步行动”的协同攻击规划模式。

3.1 面向多目标网络攻击场景的两级规划

渗透型网络攻击旨在获得对目标网络中特定计算机的控制。一般而言，Agent 以受控制的计算机开始，通过网络扫描方式收集可达主机信息。收集信息后，Agent 选择一台可达主机进行攻击。若该主机能够被攻击者成功控制，那么攻击者将进一步寻找新的可达主机进行攻击。图 1 给出了多目标渗透型网络攻击的典型示例场景，攻击者按照下标数字从小到大逐步展开攻击。在这个过程中，由于原子攻击动作具有较强的偏序性，如果多个 Agent 同时在原子攻击层面进行规划，则攻击规划将会面临大量的原子攻击动作冲突和状态冲突问题，增加了规划的技术难度和挑战，且难以保证在限定的时间内完成多目标攻击规划任务。

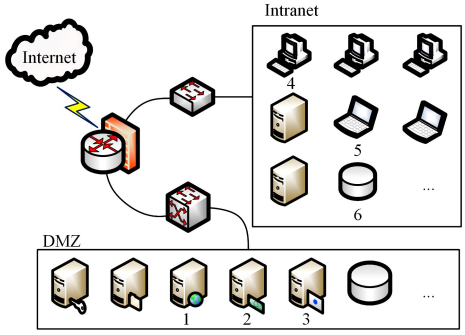

图 1 多目标渗透型网络攻击场景

Fig. 1 Multi-target penetrating network attack scenario

针对多 Agent 协同攻击规划的问题特点，可以将多目标攻击规划问题进行分解，从攻击目标和攻击动作两个层面分别进行规划。

(1)攻击目标规划

从宏观角度看，渗透型网络攻击的过程可以被视为一种逐个攻破内网主机的过程。如图 1 所示，主机的下标数字表示一种可能的攻击目标顺序。在这个过程中，每一次的成功攻击都会改变攻击者对网络的访问能力(例如在成功攻击主机 1 前无法访问主机 2)。特别是在完全信息条件下，这个层面规划的目的是找到一条从外网到内网目标主机尽可能短的“通路”。

(2)攻击动作规划

从原子攻击的角度看，当确定了每一步需要攻击的主机后，攻击者需要根据该主机的状态(例如操作系统、开放端口、安装的应用程序)规划攻击动作序列。这个层面的规划是以完全控制目标主机为前提的，得以到一个尽可能短的原子攻击动作序列。

在多 Agent 渗透攻击场景中引入分层规划策略有以下两个好处：

1)能够有效避免多 Agent 直接进行原子攻击规划而带来的动作冲突问题。例如，在图 1 的场景中，假设存在 3 个不同的 Agent——A，B，C 分别以主机 4，5，6 为目标。主机 1 具有进入内网的攻击路径，当 A 成功控制主机 1 后，B 和 C 也同样具有访问内网的权限，因此可以直接对 5，6 开展攻击而无须对主机 1 进行重复攻击。在网络攻击中，如果对同一主机进行多次攻击，则极有可能导致目标主机运行异常，如蓝屏或宕机。

2)能够有效降低攻击规划的问题规模。原子攻击规划问题的状态空间规模与目标主机、操作系统、应用软件等的数量和种类密切相关。例如，假设某个网络攻防场景中，原子攻击规划的状态空间大小为 $|S|$，主机数量为 $|H|$，操作系统数量类型为 $|O|$，每种操作系统版本数量为 $|V_o|$，应用软件类型数量为 $|A|$，应用软件版本类型数量为 $|V_A|$，则 $|S|=(|V_o|^{|O|}\times|V_A|^{|A|})^{|H|}\times 2^{|H|\times|H|}$。随着目标主机数量、参与 Agent 数量等不同维度影响因素的增加，$|S|$ 会呈现出指数级增长。此时，多目标攻击规划问题的求解变得十分困难。如果将多目标攻击规划问题进行抽象分解，如变为两级规划，由于不同抽象层次规划问题的规模有限，则可大大降低整体攻击规划的实现难度。在多目标网络攻击中，首先在攻击目标层面进行规划，为不同的 Agent 分配不同的目标主机；然后，每个 Agent 在对目标主机进行攻击规划时，可以充分利用 NIG-AP 方法[13]，得到单目标的原子攻击序列；最后将两个层级的规划结果进行组合，从而得到整体规划的近似解。

3.2 面向目标层级规划的多 Agent 协同攻击模式

根据参与决策的 Agent 的行为，多 Agent 决策通常可以分为两类：一类是所有 Agent 之间都没有利益冲突的多 Agent 集中决策；另一类是这些 Agent 之间既有共同利益，又存在攻击资产竞争关系的多 Agent 群体决策。在多 Agent 群体决策中，既要妥善解决不同 Agent 之间合作博弈的关系，还要同时满足决策的多个目标。这类群体决策的问题在于博弈本身是用于刻画 Agent 之间的竞争的，各 Agent 出于自身利益最大化进行决策，因此引导每个 Agent 选择全局最优策略的计算是十分困难的，很可能无法得到全局最优的结果。在多 Agent 集中决策的场景中，多个 Agent 的目标利益一致且各自攻击资产独立，不存在 Agent 之间因为资产冲突而发生的动作冲突，因此更易于求解全局最优解。

本文采用“集中决策、同步行动”的协同模式。对于具有 n 个 Agent 并行攻击的场景，设置第 $n+1$ 个 Agent 用于决策。具体实施攻击的 Agent 称为行动 Agent，负责规划决策的 Agent 称为决策 Agent。决策 Agent 与行动 Agent 之间的协同控制关系如图 2 所示。

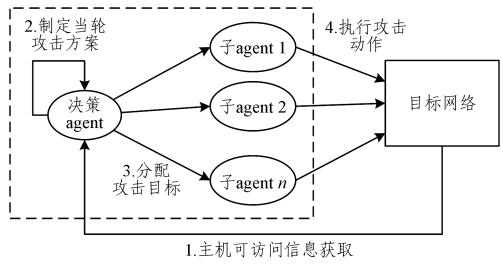

图 2 多 Agent 的集中决策、同步行动模式

Fig. 2 Multi-agent centralized decision making and synchronous action mode

多个 Agent 攻击任务规划可以看作一个由决策 Agent 主导的多轮次“决策-行动-评估”过程。在每个攻击轮次中，决策 Agent 首先获取目标网络中的主机可访问信息，确定目标

网络的当前状态，然后制定当前轮次的攻击方案，即由哪个行动 Agent 对目标网络中的哪台主机发起攻击。行动 Agent 在接受到分配的攻击任务后自主发起对指定目标的攻击。行动 Agent 之间的信息共享实际由决策 Agent 在制定当轮攻击方案时完成，从而尽可能地降低了通信消耗。优化的目标是使攻击所有目标主机所经历的轮次尽可能小。本文将这种多 Agent 的规划模式称为基于集中决策和同步行动的协同攻击规划(Centralized Decision-making and Synchronous Operating based Coordinated Attack Planning，CDSO-CAP)。

4 CDSO-CAP 模型与规划算法

为了实现本文提出的 CDSO-CAP，本节首先对集中决策和同步行动涉及的基本概念进行形式化定义，逐步建立其完整的 CDSO-CAP 规划模型；然后，面向最佳攻击目标序列的生成，提出了基于贪心策略的 CDSO-CAP 问题的求解算法。

4.1 CDSO-CAP 概念模型

用 $A=\{a_1,a_2,\cdots,a_z\}$ 代表多 Agent 的集合，其中 a_i 是每个具有特定攻击能力的 Agent。Agent 所在场景中的网络节点以及节点之间的攻击依赖关系可由一个有向图 $G=(N,L)$ 描述。其中，$N=\{n_1,n_2,\cdots,n_x\}$ 是目标网络中节点的集合，节点类型包括主机、服务器、防火墙、路由器等逻辑主机；$L=\{l_1,l_2,\cdots,l_y\}$ 代表节点之间的攻击依赖关系集合，用 $l=(n_i,n_j)\in L$ 表示节点 n_i 是节点 n_j 的攻击前置节点。图 3 给出了图 1 渗透测试场景的一组攻击依赖关系，其中实线代表显性的攻击依赖关系，虚线代表隐性的攻击依赖关系。

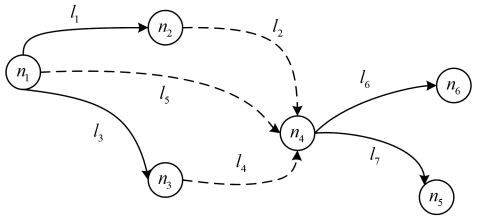

图 3 多目标网络及其攻击依赖关系

Fig. 3 Network with multiple targets and its attack dependency

图 3 中的节点 1 和节点 2 同属于 DMZ 区，则两者之间具有显性攻击依赖关系，即 Agent 可以利用 DMZ 内部的 IP 连接关系从节点 1 发起到节点 2 的主动攻击(如远程缓冲区溢出攻击)；根据图 1，节点 1 所在的 DMZ 和节点 4 所在的内网之间被防火墙(应用代理)隔离，节点 1 和节点 4 之间不存在直接的 IP 连接，但若节点 4 被授权可以访问节点 1 的服务(如 Web)或节点 1 向节点 4 用户开放了登录 Web 后台所需的 ssh 服务，则节点 1 和节点 4 存在隐性的攻击依赖关系，即攻击者可利用这种上层业务连接关系从节点 1 向节点 4 发起被动攻击。可见，节点 1 既是节点 2 的攻击前置节点，也是节点 4 的攻击前置节点。本质上，L 反映的攻击依赖关系描述了节点之间的攻击可达性。

定义 1(攻击距离) 其用于计算节点间攻击可达性的量化指标，记为 d。有向图 G 中具有直接攻击依赖关系(包括显性和隐性关系)的节点间的攻击距离记为 1。

定义 2(距离矩阵) 其指场景中节点之间攻击可达性的表示方法，记为 D。D 根据有向图 G 计算得到，D 中的每个元素代表任意两个节点之间最短的攻击距离，该距离可由目标主机进行广度优先搜索的方式确定。若节点之间不存在直接或间接的攻击依赖关系(即攻击路径)，则该元素标记为 ∞。

为清晰阐述多 Agent 协同攻击规划过程，下面分别给出了目标向量、受控节点向量、可攻击节点向量以及任务向量等相关概念的定义。

定义 3(目标向量) 其指节点集 N 中关于目标节点和非目标节点描述的一个记录，由一个 x 维的布尔向量 $\boldsymbol{g}$ 表示。对于 $1\leqslant i\leqslant x$，目标向量的第 i 维分量 $\boldsymbol{g}^{(i)}\in\{0,1\}$，表示节点集合 N 中的节点 n_i 是否为目标节点。当 $\boldsymbol{g}^{(i)}=1$ 时，节点 n_i 为目标节点；否则节点 n_i 不是目标节点。例如，若多 Agent 选中图 3 中的节点 5 和节点 6 作为攻击目标，则目标向量可表示为(0,0,0,0,1,1)。

定义 4(受控节点向量) 其是 Agent 基于网络攻击状态进行规划的主要依据，由一个 x 维的布尔向量 $\boldsymbol{c}$ 表示。对于 $1\leqslant i\leqslant x$，受控节点向量的第 i 维分量 $\boldsymbol{c}^{(i)}\in\{0,1\}$。当 $\boldsymbol{c}^{(i)}=1$ 时，节点 n_i 已被控制；当 $\boldsymbol{c}^{(i)}=0$ 时，节点 n_i 未被控制。例如，在某个时刻图 3 中的节点 1、节点 2 和节点 4 已被成功控制，则受控节点向量为(1,1,0,1,0,0)。

定义 5(可攻击节点向量) 其是 Agent 基于网络攻击状态进行规划的补充依据，由一个 x 维的布尔向量 $\boldsymbol{r}$ 表示。对于 $1\leqslant i\leqslant x$，受控节点向量的第 i 维分量 $\boldsymbol{r}^{(i)}\in\{0,1\}$。当 $\boldsymbol{r}^{(i)}=1$ 时，节点 n_i 可作为攻击对象；当 $\boldsymbol{r}^{(i)}=0$ 时，节点 n_i 不可作为攻击对象。特别地，当一个节点被控制后，该节点不再作为可攻击对象。因此，在任意时刻，总有 $\boldsymbol{c}\cdot\boldsymbol{r}=0$。例如，若图 3 中受控节点的向量为(1,1,0,1,0,0)，则可攻击节点向量为(0,0,1,0,1,1)。

定义 6(任务向量) 其是对 Agent 攻击目标的一个指派，可由一个 z 维的布尔向量 $\boldsymbol{t}$ 表示。对于 $1\leqslant j\leqslant z$，任务向量的第 j 维分量 $\boldsymbol{t}^{(j)}\in\{0,1,2,\cdots,x\}$，表示对第 j 个 Agent 指派的攻击对象。假设 $\boldsymbol{t}^{(j)}=i$，当 $i>0$ 时表示为 a_j 指派攻击对象 n_i；当 $i=0$ 时表示不为 a_j 指派任何攻击对象。例如，图 3 所示的场景中存在两个 Agent，为 a_1 指派的攻击对象为节点 1 和节点 4，为 a_2 指派的攻击对象为节点 2 和节点 3，则此时任务向量 $\boldsymbol{t}$ 可用一个二维矩阵表示：

$$\boldsymbol{t}^T=\begin{pmatrix}1&0&0&1&0&0\\0&1&1&0&0&0\end{pmatrix}\tag{1}$$

为了进行多 Agent 协同攻击规划，需要计算在单 Agent 理想条件下所有 Agent 从当前攻击状态出发完成最终渗透攻击目标所需的联合“行动”次数(此处“行动”为节点级行动，而非原子攻击级行动)，该行动次数可由联合决策矢量矩阵计算得到。

定义 7(联合决策矢量矩阵(Joint Decision-making Vector Matrix，JDVM)) 其记为 $\boldsymbol{M}$，是当前所有 Agent 已控节点到目标节点的联合距离表示方法，矢量代表了每个 Agent 攻击的具体方向，其数值等于攻击距离。

$\boldsymbol{M}$ 由以下函数得到：

$$\boldsymbol{M}=\mathrm{SUB}(\boldsymbol{D},\boldsymbol{c},\boldsymbol{g})\tag{2}$$

其中，$\mathrm{SUB}(\boldsymbol{D},\boldsymbol{c},\boldsymbol{g})$ 函数分别按向量 $\boldsymbol{c}$ 和 $\boldsymbol{g}$ 中为 1 的元素标号

抽取两两节点最短距离矩阵 $\boldsymbol{D}$ 中对应的行与列（顺序保持不变）形成的新矩阵（$\boldsymbol{D}$ 的子矩阵）。若有 z 个 Agent 待分配任务，则 $\boldsymbol{M}=(\boldsymbol{m}_1,\boldsymbol{m}_2,\cdots,\boldsymbol{m}_z)$。

定义 8（最小化运算符） 其记为 MIN($\boldsymbol{v}$)，是对任意一个向量 $\boldsymbol{v}=(v^{(1)},v^{(2)},\cdots,v^{(k)})$ 中的每个元素取最小值的操作，其表达式为：

$$MIN(\boldsymbol{v})=\min\{\boldsymbol{v}^{(1)},\boldsymbol{v}^{(2)},\cdots,\boldsymbol{v}^{(k)}\} \tag{3}$$

定义 9（联合决策矢量距离（Joint Decision-making Vector Distance，UDVD）） 其记为$|\boldsymbol{M}|$，是一个计算多 Agent 联合行动效能的度量，由在单 Agent 理想攻击条件下从当前状态出发完成最终攻击目标所需的联合行动次数表示。

$|\boldsymbol{M}|$由步数计算函数 $STEP(\boldsymbol{c},\boldsymbol{t})$得到：

$$|\boldsymbol{M}|=\mathrm{STEP}(\boldsymbol{M})=\sum_{i=1}^{z}\mathrm{MIN}(\boldsymbol{m}_i) \tag{4}$$

定义 10（$AVAL(\boldsymbol{r})$） 其是以可攻击节点向量 $\boldsymbol{r}$ 所表示的网络攻击状态下可执行的任务集合。对于任意 $\boldsymbol{t}\in AVAL(\boldsymbol{r})$，有$\boldsymbol{t}^{(j)}\in\{i|\boldsymbol{r}^{(i)}=1\}$。

定义 11（$ACT(\boldsymbol{t})$） 其是任务向量 $\boldsymbol{t}$ 中指派的实际有动作的节点数。$ACT(\boldsymbol{t})=|\{\boldsymbol{t}^{(j)}|\boldsymbol{t}^{(j)}>0,1\leqslant j\leqslant z\}|$。

定义 12（$NEXT(\boldsymbol{c},\boldsymbol{t})$） 其是在受控节点向量 $\boldsymbol{c}$ 所表示的网络攻击状态基础上执行任务向量 $\boldsymbol{t}$ 所表示的攻击任务后所得到的新的受控节点向量。

定义 13（$EXT(\boldsymbol{c})$） 其是受控节点向量 $\boldsymbol{c}$ 所示的网络攻击状态下生成的可攻击节点向量。

4.2 基于贪心策略的协同规划算法

由第 4.1 节中的概念模型的定义过程可知，多 Agent 协同攻击规划的过程是一个联合决策矢量距离不断优化的过程，具体为：

Step1 从初始状态出发，决策 Agent 对行动 Agent 进行任务指派；

Step2 行动 Agent 采取攻击动作后，会产生新的受控节点会，且这些新的受控节点会对网络中主机的攻击可达性产生影响；

Step3 决策 Agent 根据新的受控节点的信息反馈，重新量化评估网络的攻击可达性，求解联合决策矢量距离，该距离值可看作上一轮规划结果（任务指派）的即时收益，该收益直接反映了网络攻击状态向最终目标状态逼近的程度；

Step4 决策 Agent 根据求解得到的最优距离，按照联合决策矢量矩阵进行新一轮的任务指派，从 Step2 开始重复执行，直至达到最终攻击目标。

多 Agent 协同攻击的每一轮次规划问题可以表示为分层最优化问题，每一轮次的优化算法如下：

$$\max\ (\mathrm{STEP}(\mathrm{SUB}(\boldsymbol{D},\boldsymbol{c},\boldsymbol{g}))-STEP(SUB(\boldsymbol{D},\boldsymbol{c}',\boldsymbol{g}))) \tag{5}$$

$$\min\ \mathrm{ACT}(\boldsymbol{t}) \tag{6}$$

$$\text{s.t. } c'=\mathrm{NEXT}(\boldsymbol{c},\boldsymbol{t}) \tag{7}$$

$$\boldsymbol{r}=\mathrm{EXT}(\boldsymbol{c}) \tag{8}$$

$$\boldsymbol{t}\in\mathrm{AVAL}(\boldsymbol{r}) \tag{9}$$

式(5)和式(6)是算法优化的目标函数。其中，式(5)的作用是确保当前分配的任务 t 执行后能够获得最大的收益，即获得最大的攻击步数跨度，该收益将引导 Agent 在学习中做出更优选择，使多个 Agent 的联合行动尽可能以最短的攻击路径达成攻击目标；式(6)的作用是控制活跃 Agent 的个数，尽可能地以最少的 Agent 来获取最大收益，从而确保整体活动的低耗性。在以目标和效果优先为原则的规划中，式(5)先于式(6)被考虑。式(7)－式(9)为算法的约束条件，分别限制了新受控节点向量、可攻击节点向量和任务向量的取值。

采用贪心策略对多 Agent 协同攻击规划进行求解，主要包括 3 种算法，分别为基于贪心策略的多 Agent 最佳协同攻击序列生成算法、受控节点向量更新算法和可攻击节点向量更新算法。

算法 1 基于贪心策略的多 Agent 最佳协同攻击序列生成算法

输入：$\mathbf{c},\mathbf{t},\mathrm{L}$

输出：最佳攻击序列$\mathbf{t}_{\mathrm{list}}$

```
1. t_list ← [] / * 最佳攻击序列 * /
2. s̄ ← ∞ / * 记录执行 t̄ 后仍需要的步数 * /
3.     while s̄ > 0:
4.         r ← EXT(c, L) / * 计算可攻击节点向量 * /
5.         T ← AVAL(r) / * 计算合法的任务集合 * /
6. t̄ ← 0 / * 记录当前找到的最佳任务 * /
7.  ¯ c ← 0 / * 记录 t̄ 后预计的受控制节点向量 * /
8. b̄ ← −1 / * 记录 t̄ 的活跃 Agent 数 * /
9. for t in T :
10.    c' ← NEXT(c, t)
11. s' ← STEP(c', g)
12.    b' ← ACT(t)
13. if s' < s̄ or (s' = s̄ and b' < b̄):
14.        t̄ ← t
15. c̄ ← c'
16. s̄ ← s'
17. b̄ ← b'
18.    t_list. add(s̄)
19.    return t_list
```

$NEXT()$和 $EXT()$的执行流程如算法 2 和算法 3 所示。

算法 2 受控节点向量更新法

输入：$\mathbf{c},\mathbf{t}$

输出：新受控节点向量$\mathbf{c}'$

```
1. c' ← c
2. for j in {1, 2, ⋯, z}:
3.    if t^(j) > 0 and c^(t^(j))' = 0:
4.        c^(t^(j))' ← 1
5.    return c'
```

算法 3 可攻击节点向量更新算法

输入：$\mathbf{c},\mathrm{L}$

输出：可攻击节点向量 $\mathbf{r}$

```
1. r ← 0
2. for i in {1, 2, ⋯, x}:
3.    for j in {1, 2, ⋯, x}:
4.        if (n_i, n_j) ∈ L and c^(i) = 1 and c^(j) = 0:
5.            r^(j) ← 1
6. return r
```

5 实验验证

本节将通过实验来验证本文提出的 CDSO-CAP 方法的有效性。协同路径规划算法通过 Python 语言实现，运行于 Windows 10 系统，其主机配置为 Intel 酷睿 i7 处理器、16 GB 内存。实验网络拓扑场景选用真实世界中的典型企业网络，共设置 4 个渗透攻击目标。因此，共设置 5 组对照实验，即将 CDSO-CAP 方法分别在 Agent 数为 1，2，3，4 时的实验结果和使用 NIG-AP 方法所得到的结果进行对比，观察完成渗透攻击目标所需要的轮次、剩余步数和活跃 Agent 数随轮次的变化。

5.1 仿真实验场景

本文实验使用的网络拓扑图如图 4 所示。企业网络由两个部分组成，即 DMZ 区与内网区域。防火墙规则允许 Internet、企业网络的内网区分别访问 DMZ 区，而不允许 Internet 直接访问企业内网区。渗透测试的目标主机共有 4 个，分别是 DMZ 区中的一台 Ubuntu 主机，以及企业内网区的一台 Ubuntu 服务器和两台主机。渗透攻击目标已在图 4 中圈出。

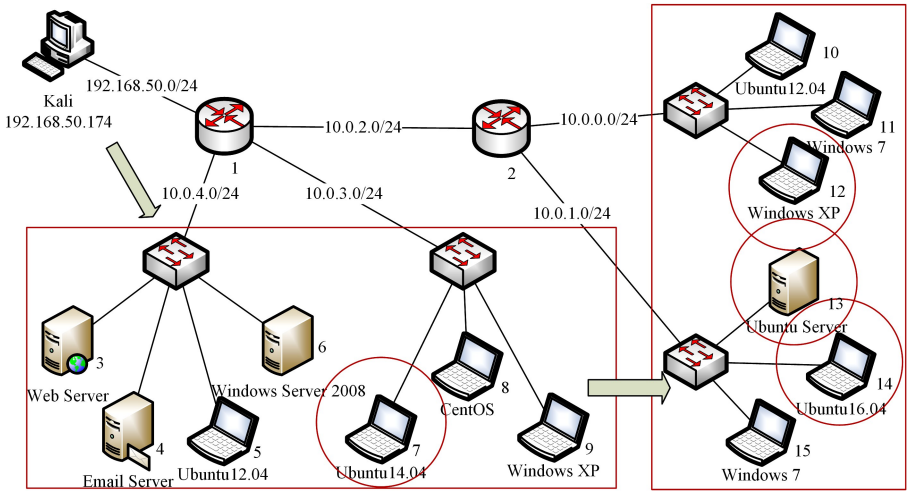

图 4 对典型企业网络的多目标渗透攻击

Fig. 4 Peneration towards multiple targets in a typical enterprise network

(1)攻击者首先对 Email Server 发起攻击，利用漏洞攻击来控制该服务器；借由该服务器向 Windows Server 2008 R2 发起攻击；通过查询记录找到 Ubuntu12.04 的用户名登录密码；Ubuntu12.04 存有路由器 1 的登录名密码，通过登录该路由器可以开放向内网的数据转发的权限。由于路由器 2 进行了限制，攻击者此时只能访问 10.0.0.0/24 网段。

(2)攻击者通过 SQL 注入方式攻击 Web Server；借由该服务器向 Windows Server 2008 R2 发起攻击；通过查询记录找到 Ubuntu14.04 的用户名登录密码；Ubuntu14.04 存有路由器 2 的登录名密码，通过登录该路由器可以开放向 10.0.1.0/24 网段的访问权限。

可以发现两条攻击路径中存在相同的攻击对象，即借由 Email Server 或 Web Server 向 Windows Server 2008 R2 发起攻击。实验结果将展示本文提出的方法如何消除攻击路径中对重复对象发起的攻击。

5.2 实验结果分析

表 1 列出了使用 NIG-AP 方法边探测边规划和使用基于贪心策略的多 Agent 协同攻击规划(CDSO-CAP)的结果。其中使用 NIG-AP 方法需要的轮次最多，这是由于该方法并未掌握攻击所需的全局信息，在攻击过程中试探性的行为将导致对非必要攻击对象开展攻击。而 CDSO-CAP 使用了全局信息，因此其结果全面优于 NIG-AP 方法。当 Agent 的数量为 1 时，使用 CDSO-CAP 方法共需要 9 个轮次才能完成渗透攻击，而当 Agent 的数量为 4 时，其只需要 5 个轮次就能完成渗透攻击。因此，在该方法中，多 Agent 相比单 Agent 能够更快地完成渗透攻击过程。

表 1 不同攻击规划方法的结果对比

Table 1 Comparison of results of different attack planning methods

方法	任务分派序列	轮次
NIG-AP	(3),(4),(6),(5),(7),(8),(9),(1),(2),(10),(11),(12),(13),(14)	14
CDSO-CAP(1)	(3),(6),(7),(2),(5),(13),(14),(1),(12)	9
CDSO-CAP(2)	(0,3),(0,6),(5,7),(1,2),(12,13),(0,14)	6
CDSO-CAP(3)	(0,0,3),(0,0,6),(0,5,7),(0,1,2),(12,13,14)	5
CDSO-CAP(4)	(0,0,0,3),(0,0,0,6),(0,0,5,7),(0,0,1,2),(0,12,13,14)	5

进一步观察实验结果可以发现，本文方法的 Agent 数为 3 时所需的轮次和 Agent 数为 4 时所需的轮次数相同。分析两者对应的任务分派序列可知，Agent 数为 4 时对应的任务分派序中每一个任务分派都包含了至少 1 个静默的 Agent。这就意味着这个 Agent 在整个攻击过程中都未被分配任务，因此在实际攻击过程中仅使用 3 个 Agent 就能达到最优的效果。确定 Agent 数量的方法是：1)将渗透攻击目标数作为 Agent 个数；2)运行基于贪心策略的多 Agent 协同攻击规划方法；3)在确保需求轮次数最少的同时，找到最少的 Agent 个数。

图 5 和图 6 分别给出了在不同 Agent 数量的条件下(仅考虑 Agent 数为 1，2，3)，剩余步数和活跃 Agent 数随轮次变化的变化情况。从图 5 可以发现，在第 1，2 轮任务分配后，3 条曲线下降的程度相同，在第 3，4 轮任务分配后，Agent 数为 2 和 3 的曲线下降程度相同，快于 Agent 数为 1 的曲线。在第 5 轮时，Agent 数为 3 的曲线率先降至 0，而后 Agent 数量

为 2 的曲线在第 6 轮次降至 0，最后 Agent 数量为 1 的曲线在第 9 轮降至 0。结合图 6 可以发现，在最初两轮中，最优的任务分配只允许 1 个 Agent 进行攻击。对于 4 个最终目标而言，单 Agent 对单目标每前进一步，总体则前进 4 步。在中间两轮中，最优的任务配置允许 2 个 Agent 进行攻击。因此，Agent数量 2 和 3 的对照组的剩余步数下降得更快，且在第 3 轮时作为渗透攻击目标的 Ubuntu14.04 被控制，故第 4 轮任务分配后其剩余步数只比第 3 轮任务分配后的剩余步数减少了 3。

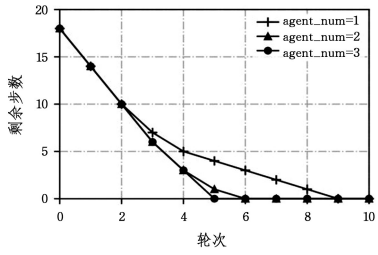

图 5 Agent 数改变时的轮次-剩余步数变化

Fig. 5 Changes of turns and remaining steps when number of agents changes

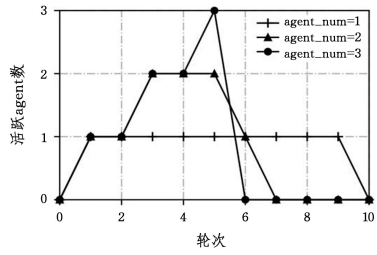

图 6 Agent 数改变时的轮次-活跃 Agent 数变化

Fig. 6 Changes of turns and the number of Active agents when number of agents changes

结束语 队组协同攻击是自动化渗透测试的重要模式，通过对多目标网络同步展开多路攻击规划，可以实现对防御者的全方位检验。本文以此为研究背景，提出了一种基于 CDSO-CAP 模型的多 Agent 协同攻击路径规划方法。首先，在区分渗透攻击的两级规划基础上，提出了一种“集中决策、同步行动”的协同控制模式，将多 Agent 协同攻击任务规划问题转化为分层优化问题，优化的第一个目标是攻击步数降低的最大化，第二个目标是在第一个目标的基础上最小化活跃 Agent 数目。然后，面向多目标并行规划求解，提出了联合决策矢量矩阵 JDVM，以联合决策矢量距离为奖励计算依据，采用贪婪策略迭代求解最优攻击目标及其分配方案。实验结果表明，CDSO-CAP 算法的复杂度低、收敛性好，与单 Agent 规划相比，其对大规模目标网络场景的攻击规划效率更高。

本文所提的两级攻击规划和分层问题优化的方法为解决多 Agent 协同攻击规划提供了较好的思路，有效控制了多 Agent 规划的问题规模，为渗透测试的目标选择、Agent 组成、攻击实施提供了决策依据。本文是基于联合决策的多 Agent 协同规划，现实攻击中还要考虑对抗条件下的对手策略对攻击成功率的影响，后续工作将对这一问题进行深入研究。

参 考 文 献

[1] OBES J L, SARRAUTE C, RICHARTE G. Attack planning in the real world[J]. arXiv: 1306.4044, 2013.

[2] SARRAUTE C, RICHARTE G, OBESJ L. AN algorithm to find optimal attack paths in nondeterministic scenarios[C]// Proceedings of th 4th ACM Workshop on Security and Artificial Intelligence. ACM, 2011: 71-80.

[3] SARRAUTE C, BUFFET O, HOFFMANN J. POMDPs make better hackers: Accounting for uncertainty in penetration testing [C]// Twenty-Sixth AAAI Conference on Artificail Intelligence. 2012.

[4] SHMARYAHU D, SHANI G, HOFFMANN J, et al. Partially observable contingent planning for penetration testing[C]// IWAISe: First International Workshop on Artificail Intellijgence in Security. 2017, 33.

[5] MCLENNAN A . The expected number of Nash equilibria of a normal form game[J]. Econometrica, 2005, 73(1): 141-174.

[6] BOUTILIER C. Planning, learning and coordination in multiagent decision processes[C]// Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge. Morgan Kaufmann Publishers Inc. 1996: 195-210.

[7] MUSLINER D J, DURFEE E H, WU J, et al. Coordinated Plan Management Using Multiagent MDPs[C]// AAAI Spring Symposium: DIstributed Plan and Schedule Management. 2006: 73-70.

[8] ONTANON S, BURO M. Adversarial hierachical-task network planning for complex real-time games[C]// Twenty-Fourth International Joint Conference on Artificail Intelligence. 2015.

[9] SARRAUTE C, BUFFET O, HOFFMANN J. Penetration Testing==POMDP Solving? [J]. arXiv: 1306.4714, 2013.

[10] KOTENKO I. Agent-based modeling and simulation of cyber-warfare between malefactors and security agents in Internet [C]// the 19th European Simulation Multiconference “Simulation in wider Europe”. 2005.

[11] ROTH M, SIMMONS R, VELOSO M. What to communicate? Execution-time decision in multi-agent POMDPs[M]// Distributed Autonomous Robotic Systems 7. Springer, Tokyo, 2006: 177-186.

[12] ZHANG C, LESSER V. Coordinated multi-agent reinforcement learning in networked distributed POMDPs[C]// Proceedings of the 25th AAAI Conference on Artificial Intelligence. San Francisco, America, 2011: 764-770.

[13] ZHOU T Y, ZANG Y C, ZHU J H, et al. NIG-AP: a new method for automated penetration testing[J]. Frontiers of Information Technology & Electronic Engineering, 2019, 20(9): 1277-1298.

ZHOU Tian-yang, born in 1979, associate professor. His main research interests include software vulnerability analysis, virtualization-based security technology and application, penetration test, fundamental study of network modeling and simulation, and cyber security assessment.