基于马尔可夫博弈与多智能体强化学习的云原生移动目标防御决策方法

耿致远，张恒巍，谭晶磊，齐高鑫
（信息工程大学密码工程学院，河南 郑州 450001）

摘　要：随着云原生网络中攻击者的类型多样化与行为智能化趋势加剧，传统移动目标防御方法难以应对攻击者类型分布未知的情形。基于贝叶斯马尔可夫博弈模型对云原生攻防场景进行建模，结合独立多智能体强化学习方法，实现了信息不对称条件下的移动目标防御智能决策。首先，分析了云原生网络环境中移动目标防御的攻防过程，针对攻防双方的不完全信息特征，将攻击类型分布未知的防御决策问题构建为贝叶斯马尔可夫博弈模型。其次，从网络攻防对抗实际出发，针对攻击者和防御者具有同等或不同智能程度的情况，设计了基于独立近端策略优化的配置转换决策算法。最后，通过实验验证了所提模型和方法能够有效应对攻击类型未知的云原生网络攻防场景，相较其他强化学习决策方法具有显著优势。

关键词：云原生；移动目标防御；贝叶斯马尔可夫博弈；独立近端策略优化；最优策略选取
中图分类号：TP309
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025173

Markov games and multi-agent reinforcement learning based decision-making method for cloud-native moving target defense

GENG Zhiyuan, ZHANG Hengwei, TAN Jinglei, QI Gaoxin
School of Cryptography Engineering, Information Engineering University, Zhengzhou 450001, China

Abstract: The increasing diversity and intellectualization of attacker behaviors in cloud-native networks have exposed the limitations of traditional moving target defense methods in handling scenarios with unknown attacker-type distributions. To address this challenge, cloud-native attack-defense interactions were modeled using a Bayesian Markov game framework, and an independent multi-agent reinforcement learning approach was incorporated to achieve intelligent moving target defense decision-making under asymmetric information conditions. Firstly, the attack-defense process of moving target defense in cloud-native environments was thoroughly analyzed. To capture the incomplete information characteristics of both parties, a Bayesian Markov game model was constructed to formalize the defense decision-making problem under uncertain attacker-type distributions. Secondly, grounded in practical network confrontation dynamics, an optimal strategy selection algorithm based on independent proximal policy optimization was designed, accounting for both symmetric and asymmetric intelligence levels between attackers and defenders. Finally, experimental results demonstrate that the proposed model and method can effectively handle cloud-native attack-defense scenarios with unknown attacker types, outperforming other reinforcement learning-based decision-making methods with significant advantages.

Keywords: cloud-native, moving target defense, Bayesian Markov game, independent proximal policy optimization, optimal strategy selection

收稿日期：2025–06–17；**修回日期：**2025–09–16
通信作者：张恒巍，zhw11qd@163.com
基金项目：国家自然科学基金资助项目（No.62502103）
Foundation Item：The National Natural Science Foundation of China (No.62502103)

0 引言

以DevOps（development operations）、持续交付、微服务和容器技术为代表的云原生架构，通过将复杂业务系统解耦为多个功能独立的微服务单元，并借助容器化封装和自动化编排机制，使每个微服务都能实现独立开发部署，进而实现复杂系统的高效扩展，帮助企业快速构建高弹性、易管理、松耦合且具备观测性的应用系统，显著提升资源利用率与交付效率[1]。然而，云原生组件供应链采用的独立开发模式、共享组件依赖及容器基础镜像的高度开放性，也为系统引入了更多漏洞后门威胁[2]。一方面，在人工编码阶段的漏洞无法避免。在快速交付需求的驱动下，云原生系统开发人员缺乏对编写代码的严格审查，导致在编码阶段出现不可避免的安全缺陷。另一方面，云原生系统广泛依赖的开源组件存在较多已知漏洞。尽管云原生计算基金会（CNCF, Cloud Native Computing Foundation）社区提供了大量高质量开源项目以提升开发效率，但其开放性特点也带来了额外的漏洞风险。

尽管目前针对漏洞利用问题已涌现出基于特征匹配的漏洞扫描[3]和漏洞修复[4]等技术手段，但由于云原生系统架构高度复杂且运行环境动态多变，网络安全威胁治理难度显著增加。从技术实施层面看，传统安全技术（如认证授权、访问控制等）在动态复杂的云原生应用中难以有效落地；从配置管理层面看，云原生架构的复杂性进一步加剧了安全配置的动态管理难度，安全配置的不当设置可能招致更多的网络攻击。

为了扭转这种不对称的攻防态势，移动目标防御（MTD, moving target defense）通过主动改变系统配置的防御方式动态变换系统攻击面，为解决云原生环境下的安全挑战提供了新思路[5]。现有的移动目标防御方法研究主要有基于博弈论和深度强化学习两大方向。博弈论中的非合作博弈可以为研究博弈参与者在对抗冲突条件下的策略选取提供有效的建模工具，通过将攻防双方建模为博弈的局中人，在构建攻防双方策略的基础上通过求解均衡得到最优的移动目标防御策略。曾威等[6]通过构建容器云的异构镜像提出一种基于Stackelberg博弈模型的动态异构式调度方法，并将调度问题建模成混合整数非线性规划问题求解系统最优调度概率。Zhang等[7]针对无标度网络中实时安全防御决策问题，基于微分博弈理论构建网络攻防微分博弈模型，通过求解鞍点均衡策略实现最优防御策略选择，为复杂网络环境下的实时防御决策提供了新框架。文献[8]针对移动目标攻防中的有限理性问题，提出一种基于Wright-Fisher过程的演化决策方法，通过引入理性参数构建攻防策略演化博弈模型以求解最优防御策略。深度强化学习结合深度学习的感知能力和强化学习的决策能力，通过不断地决策探索与反馈学习训练，逐渐逼近并最终实现最优防御策略。针对有限资源约束场景下的移动目标防御部署优化问题，文献[9]结合攻击图理论构建非线性数学模型并提出深度强化学习框架，利用深度Q网络（DQN, deep Q-network）与近端策略优化（PPO, proximal policy optimization）算法求解最小化系统安全损失的部署策略。张帅等[10]提出一种基于DQN和移动目标防御的微服务动态清洗方案，将大规模安全配置空间条件下清洗周期求解难的问题转化为马尔可夫决策过程，利用DQN算法求解最优的防御策略。Kim等[11]在软件定义网络环境中融合流量检测与移动目标防御技术，利用深度确定性策略梯度（DDPG, deep deterministic policy gradient）算法优化流量检测资源分配和基于IP地址随机化的移动目标防御策略，通过动态适应网络状态提升恶意流量捕获效率并降低系统漏洞暴露风险。

博弈论出色的策略评估能力使其在移动目标领域应用广泛，但其需要专家预设完整的奖励函数和状态转移模型，因此现有研究尝试结合深度强化学习技术实现策略的在线学习[12-17]。然而，目前基于博弈论与深度强化学习的移动目标防御方法在应用于云原生环境时仍有2个主要问题有待解决。一是攻击类型未知下的博弈建模问题。云原生环境中攻击者类型多样、分布未知，可能包括国家级黑客、高级持续性威胁（APT, advanced persistent threat）组织、脚本小子等，其攻击意图与技术能力差异显著。即使采用相同的攻击动作（如利用同一漏洞），不同类型攻击者也会带来不同的攻击效果，从而导致防御者获得不同的收益，但防御者无法通过单一攻击动作来判断攻击者的真实类型。现有研究多假设攻击类型服从已知分布，该强约束假设使防御策略效果严重依赖先验信息准确性。二是基于深度强化学习的模型求解问题。现有研究往往将攻击策略内置于环境中，然后将双人博弈的决策问题简化为

防御者单智能体的优化任务，忽略了攻击者的策略学习与动态调整能力。尽管部分研究尝试采用多智能体强化学习进行联合策略训练，但其通常依赖全局状态信息，导致算法在实际应用中受限。

针对上述问题，本文提出一种基于贝叶斯马尔可夫博弈与独立近端策略优化（IPPO, independent proximal policy optimization）的云原生移动目标防御决策方法。首先，构建的贝叶斯马尔可夫博弈模型能够精确刻画攻防双方不完全信息特征，防御者可以利用贝叶斯更新机制，通过历史交互信息动态更新对攻击者类型分布概率的信念，从而摆脱对先验知识的依赖，提升模型对不完全信息环境的适应能力。其次，采用IPPO多智能体强化学习算法求解最优防御策略。与多智能体近端策略优化（MAPPO, multi-agent proximal policy optimization）等依赖全局信息的多智能体算法不同，IPPO中各智能体仅依赖自身局部观测独立更新策略，不需要共享对方私有信息，不仅更符合实际攻防中信息不透明的特点，还降低了算法复杂度和训练成本，更适用于对实时性要求较高的云原生网络环境[18]。本文的主要工作如下。

1)根据云原生系统配置特点构建系统异构资源池，基于国家信息安全漏洞库与通用漏洞评分系统搭建以漏洞利用为核心的攻防实验环境，实现云原生攻击面的动态建模。

2)考虑攻击者不完全信息特征与策略学习能力的攻防场景，构建贝叶斯马尔可夫博弈模型，重点刻画攻击类型分布不确定下的防御决策过程，并形式化其收益函数与策略更新机制。

3)提出基于IPPO算法的最优配置转换策略决策方法，通过多组对照实验验证算法的收敛性与有效性，并与其他强化学习方法进行性能对比，实验结果证明了所提方法的优越性。

1　云原生移动目标防御攻防策略分析

1.1　攻击策略分析与设计

云原生系统构成的复杂性导致系统面临多种安全威胁，攻击者可借助镜像库、容器及其运行时存在的漏洞，直接或间接地对相关组件发动镜像仓库攻击、容器提权与逃逸攻击以及容器网络攻击等。同时，攻击者还会利用编排工具或微服务的漏洞，实施服务对外暴露攻击、业务节点攻击、应用程序接口（API, application programming interface）攻击和服务网格攻击等，以此实现入侵目的。国家信息安全漏洞库（CNNVD, China National Vulnerability Database of Information Security）详细记录了云原生系统配置供应链组件中各类已知漏洞。这些记录涵盖了从漏洞影响程度到攻击者可能利用的攻击技术等丰富信息，其中潜在攻击者的攻击渗透途径，为构建攻击策略集提供了理论依据。基于此，本文构建云原生系统漏洞库$V=\{v_{i,j}\}$，其中$v_{i,j}$表示系统配置中第i类组件中第j个漏洞。若定义系统攻击面状态空间为$E=\{e_c\}$，其中e_c表示系统在当前配置下所有漏洞集合，则e_c满足$e_c \subseteq V$且随系统配置动态变化。结合攻击者行为特征与技术能力差异构建多维度攻击策略空间，具体而言，攻击策略被形式化为四元组模型{type, app, ability, prob}，各维度定义如下。

type表示攻击者类型，反映其组织背景（如脚本小子、APT组织、国家级黑客等）与攻击意图差异，不同类型攻击者策略偏好与资源投入程度不同。

app表示攻击者能够影响的关键技术单元，具体对应云原生供应链中的异构组件（如Doker容器运行时、Redis数据库等）。

ability表示对应的技术能力，通过这一数值具体量化攻击者利用特定漏洞的实际效能，数值越高表示攻击成功概率越大。

prob表示攻击类型对应分布概率，表征不同类型攻击者在攻防交互中的策略选择倾向，攻击者能够在博弈过程中实时动态更新。

1.2　防御策略分析与设计

云原生系统的防御体系构建于其技术架构的天然异构性之上，核心技术栈涵盖容器镜像库、容器运行时、编排工具、服务网格、数据库等关键领域，支撑这些技术实现的供应链组件具有显著异构特征：容器镜像库包含亚马逊ECR、阿里云ACR与威睿Harbor等多元化实现方案；容器运行时则包括Docker、Cri-o、Containerd等差异化组件，为实施云原生移动目标防御提供了现实基础。因此，本文构建了一个异构配置集负责提供云原生系统的异构配置，具体如图1所示。一个有效配置c可以表示为$c=T_0\times T_1\times\cdots\times T_n$，$T_i$表示云原生系统所需的第$i$类技术，$K_i=\{t_i^0,t_i^1,\cdots,t_i^m\}$表示$m$种可以实现技术$T_i$

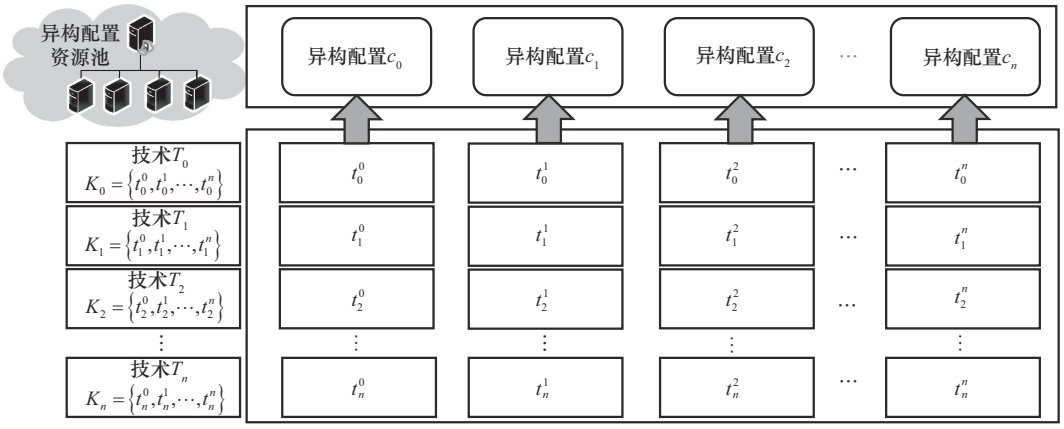


图1 云原生系统异构配置示意

的不同组件集合，理论上，配置空间（$C=\{c_0,c_1,\cdots,c_n\}$）规模为各技术模块组件数的笛卡尔积，即$|C|=|K_0|\times|K_1|\times\cdots\times|K_n|$。防御智能体依据防御策略从系统异构配置集实时动态选择目标配置进行转换，通过改变系统当前技术组件组合实现攻击面的动态变换。这一机制在确保系统在维持合法用户正常使用的前提下，增加攻击者执行侦察攻击的不确定性和复杂性，从而提升系统整体安全性。

2 基于贝叶斯马尔可夫博弈的多阶段移动目标防御模型

2.1 博弈模型构建

根据以上云原生网络环境攻防双方的策略分析，本节建立了一个基于贝叶斯马尔可夫博弈的移动目标防御模型。模型相关要素定义为一个五元组(N, D, A, S, U)，具体定义如下。

$N=\{N_D, N_A\}$表示云原生网络攻防对抗博弈模型中的局中人，其中，D表示防御方，A表示攻击方。在该模型中，防御方只存在一个防御者（云原生服务提供商），攻击方有n种不同类型的攻击者$N_A=\{A_0,A_1,\cdots,A_n\}$。

$S=\{s_0,s_1 s_n\}$定义为云原生系统的有限配置状态集合，每个状态s_i对应唯一的异构配置，与2.2节构建的异构配置集$C=\{c_0,c_1 c_n\}$形成逐一映射关系。

$D=\{d_i|i=0,1,\cdots,n\}$表示防御者动作集合，与状态空间严格对应，即每个动作d_i表示转换至配置状态s_i的操作。不同配置间的转换存在相应的转换成本$W=\{w_{ss'}\}$，其中$w_{ss'}$表示从当前状态配置s转换至目标状态s'的资源消耗（如计算开销、服务中断时延等）。防御策略π_D定义为状态依赖的分布概率$\pi_D(s_i)=\{p(d_0|s_i), p(d_1|s_i),\cdots, p(d_n|s_i)\}$，表示防御者在状态$s_i$下选择各配置转换动作的分布概率，满足$\sum_{j=1}^{n} p(d_j|s_i)=1$。

$A=\{a_i|i=0,1,\cdots,n\}$表示攻击者动作集合，对应云原生系统漏洞库中的可利用漏洞，不同类型攻击者具有差异化的可用漏洞子集（如APT组织可利用零日漏洞，脚本小子依赖公开已知漏洞）。攻击策略π_A定义为$\pi_A(s_i)=\{p(a_0|s_i), p(a_1|s_i),\cdots, p(a_k|s_i)\}$，表示攻击方在状态$s_i$下选择不同类型攻击者实施攻击的分布概率，满足$\sum_{j=1}^{k} p(a_j|s_i)=1$。

$U=\{U^D, U^A\}$表示攻防双方的收益，其中，$U^{D/A}\{s, d(s), a(s)\}$表示防御者N_D或攻击者N_A在状态s下攻防双方分别采取动作$d(s)$和$a(s)$获得的收益。

2.2 攻防收益分析

结合文献[19-20]，本文基于通用漏洞评分系统（CVSS, common vulnerability scoring system）v3.1标准定义攻防双方的收益函数。CVSSv3.1通过可利用性评分（EM）与影响评分（IM）两大维度评估漏洞风险[21]。可利用性评分用来量化漏洞被成功利用的技术难度，由攻击途径（AV）、攻击复杂度（AC）、特权需求度（PR）、用户交互度（UI）4项指标构成，反映攻击者实施漏洞利用所需的资源投入与操作条件。影响评分用来评估漏洞利用对系统安全的危害程度，通过机密性影响（C）、完整性影响（I）、可用性影响（A）3项指标综合衡量，体现漏洞被利用后对系统核心安全属性的破坏程度。IM和EM的具体计算式分别为

$$\mathrm{EM}=8.22\times\mathrm{AV}\times\mathrm{AC}\times\mathrm{PR}\times\mathrm{UI} \tag{1}$$

$$\mathrm{IM}=6.42\times[1-(1-C)\times(1-I)\times(1-A)] \tag{2}$$

攻防双方根据系统状态生成当前各自策略并依据策略选择自身动作，然后计算动作的交互收益，在计算动作交互的收益前，需先判定攻击是否成功。具体来说，如果攻击者选定的攻击动作中包含防御动作选择配置中存在的组件漏洞，且该类型攻击者对此技术组件的攻击能力大于漏洞的可利用性评分，当且仅当2个条件同时满足时判定攻击成功，记$g(s, a, d)$=1；否则判定攻击不成功，记$g(s, a, d)$=0。判定函数的数学表达式可表示为

$$g(s,a,d)=\begin{cases}1, e_d \in a \cap \text{ability}_a \geqslant \text{EM}_{e_d} \\ 0, \text{其他}\end{cases} \tag{3}$$

对于攻击者的收益U^A，当攻击成功（即判定函数$g(s,a,d)=1$）时，攻击回报为漏洞的影响评分IM；当攻击失败（$g(s,a,d)=0$）时，回报为零。此外，不同类型攻击者进行攻击时调用资源的不同导致攻击成本存在较大差异，本文利用攻击者的技术能力量化这种不同的攻击成本，技术能力值越高代表调用的资源越多，付出的攻击成本越大。综合攻击回报和成本因素，环境状态s下的攻击者收益可以表示为

$$U^A(s,a,d)=\begin{cases}\text{IM}_{e_d}-\text{ability}_a, g(s,a,d)=1 \\ -\text{ability}_a \quad\quad , g(s,a,d)=0\end{cases} \tag{4}$$

对于防御者的收益U^D，同样要考虑攻击是否成功2种情况：当攻击成功时，防御回报同样用漏洞的IM表示，代表漏洞被成功利用后对系统的影响大小，此时回报值为负值；当攻击失败时，防御者并不会因为防御成功而获得正值奖励，因此回报设置为零。防御者执行配置转换动作时需要承担状态转换成本$w_{ss'}$，综合防御回报和成本因素。其中，转换成本$w_{ss'}$表示从当前状态配置s转换至目标状态s'的资源消耗（如计算开销、服务中断时延等），在实际网络防御应用中可以采用CPU占用率Δ_{CPU}、服务重启时间T_{rs}、服务响应时间T_{sa}、服务完成时间T_{sd}度量计算开销和对合法用户的影响，为方便计算，将$w_{ss'}$的值归一化后映射到[0,100]，环境状态s下防御者收益可以表示为U^D。

$$w_{ss'}=\Omega(\Delta_{\text{CPU}}+T_{\text{rs}}+T_{\text{sa}}+T_{\text{sd}})$$

$$U^D(s,a,d)=\begin{cases}-\text{IM}_{e_d}-w_{ss'}, g(s,a,d)=1 \\ -w_{ss'} \quad\quad , g(s,a,d)=0\end{cases} \tag{5}$$

在多阶段移动目标防御模型中，攻防双方在各阶段获得的收益不仅与当前所处状态s直接相关，还涉及对未来阶段预期收益的动态权衡。为求解长期最优防御策略，引入折扣因子γ以量化长期收益，构建攻防双方的目标函数为

$$R^A(s_0)=U^A(s_0,d_0,a_0)+\sum_{t=1}^{\infty}\gamma^t U^A(s_t,d_t,a_t) \tag{6}$$

$$R^D(s_0)=U^D(s_0,d_0,a_0)+\sum_{t=1}^{\infty}\gamma^t U^D(s_t,d_t,a_t) \tag{7}$$

3　云原生移动目标防御决策方法

在攻防双方贝叶斯马尔可夫博弈模型构建完成后，需要解决的是如何设计高效灵活的策略生成方法，以支持防御者能够在攻击类型分布概率未知的动态环境中实现策略优化。

不同于单智能体的马尔可夫决策过程，本文模型因涉及攻防双方的多智能体动态交互，防御方的收益同时依赖于自身行为和攻击方策略，导致攻防策略的动态演化仅能收敛至纳什均衡状态。本文基于博弈论中最优响应和纳什均衡[22]相关概念对最优响应策略进行说明。

贝叶斯马尔可夫博弈中的最优响应指的是在初始状态s_0确定的情况下，给定攻击方的策略π_A°，若防御策略π_D^*是使防御方获得最大收益的策略，满足

$$R^D(s_0;\pi_A^{\circ},\pi_D^*)=\max_{\pi_D} R^D(s_0;\pi_A^{\circ},\pi_D) \tag{8}$$

则称防御策略π_D^*就是对攻击策略π_A°的最优响应。若双方的策略互为对方策略的最优响应，同时满足

$$R^D(s_0;\pi_A^*,\pi_D^*)=\max_{\pi_D} R^D(s_0;\pi_A^*,\pi_D) \tag{9}$$

$$R^A(s_0;\pi_A^*,\pi_D^*)=\max_{\pi_A} R^A(s_0;\pi_A,\pi_D^*) \tag{10}$$

即任何一方均不能通过单方面改变己方策略以获得更高收益，则称此时的攻防策略组合(π_A^*,π_D^*)为纳什均衡策略，由于任何具有有限状态空间和不完全信息的静态博弈都存在贝叶斯纳什均衡[23]，意味着基于本文收益函数和攻防策略，通过求解式(9)与式(10)可以得到最优的移动目标防御策略。

根据本文构建的贝叶斯马尔可夫博弈模型，为求解最优的移动目标防御策略，本文设计了基于独立近端策略优化的配置转换决策算法（CTG-IPPO, configuration transition generation method based IPPO）。与依赖全局信息的其他多智能体强化学习

算法不同，IPPO 算法作为 PPO 算法在多智能体环境下的扩展，其核心特征是每个智能体配备独立的演员–评论家（Actor-Critic）网络。各智能体仅依据自身观测和经验独立更新策略参数，不需要共享对手私有信息，通过环境反馈的奖励信号自主优化策略，这一机制与实际攻防中双方在不完全信息下独立决策的博弈特征高度契合。由于攻击者采用与防御者完全相同的网络结构，图 2 以防御者智能体为视角，直观展示了 IPPO 算法的策略更新训练过程。

首先攻击者和防御者的 Actor 网络分别从环境中获取当前状态 s，再由 Actor 网络生成攻击动作与防御动作的分布概率（攻防策略 $\pi_{\theta A}(s)$、$\pi_{\theta D}(s)$），根据该分布概率经过随机动作采样得到动作 a、d，再由环境执行动作 a、d 并计算攻防双方奖励函数，然后将得到的奖励 r_A、r_D 以及下一个状态 s' 分别反馈给两个智能体，智能体再将训练样本（s, a, r_A, s'）和（s, d, r_D, s'）存储到各自的经验缓冲区中。随后，环境状态更新到 s'，双方智能体再次获取状态 s' 输入各自的 Actor 网络中，重复上述步骤不断收集训练样本。当经验缓冲区中的训练样本数达到阈值后，开始进行 Actor 网络与 Critic 网络参数更新。

下面以防御者智能体为例介绍网络参数的更新过程，首先是 Critic 网络参数。防御者智能体从缓冲区中提取到存储的训练样本，对于每个时间步 t，计算从 t 到回合结束的累积奖励

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots + \gamma^{T-t-1} r_{T-1} \tag{11}$$

其中，γ 表示对未来收益的折扣系数（$0 < \gamma < 1$），T 是每回合的最大步数。同时，根据经验池中的状态集合，使用 Critic 网络可以计算得到所有状态估计的状态价值 $V_{\omega D}(s_T)$。进一步，可以得到对优势函数的估计

$$\hat{A} = G_t - V_\omega(s_T) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots + \gamma^{T-t-1} r_{T-1} - V_\omega(s_T) \tag{12}$$

优势函数通过将当前防御策略与平均水平进行对比，可以更准确地评估当前策略的优劣。根据优势函数，能够得到 Critic 网络的损失函数

$$L(\omega) = E\left[\left(\sum_{i=t}^{T} \gamma^{i-t} r_i - V_\omega(s_t)^2\right)\right] \tag{13}$$

然后依据式(13)对 Critic 网络进行权重参数的更新 $\omega' \leftarrow \omega + \alpha \nabla L(\omega)$，其中，$\alpha$ 代表学习率。

接下来介绍 Actor 网络参数的更新。值得一提的是，当单独使用一个 Actor 网络进行样本收集时，需要每次参数更新后重新开始采样，极大降低了训练样本的利用效率[24]，增加了算法的训练时长，因此引入重要性采样和剪切机制[25]，设置了 2 个 Actor 网络：Actor 网络 θ 与旧 Actor 网络 θ_{old}，根据输入的环境状态，Actor 网络与旧 Actor 网络分别输

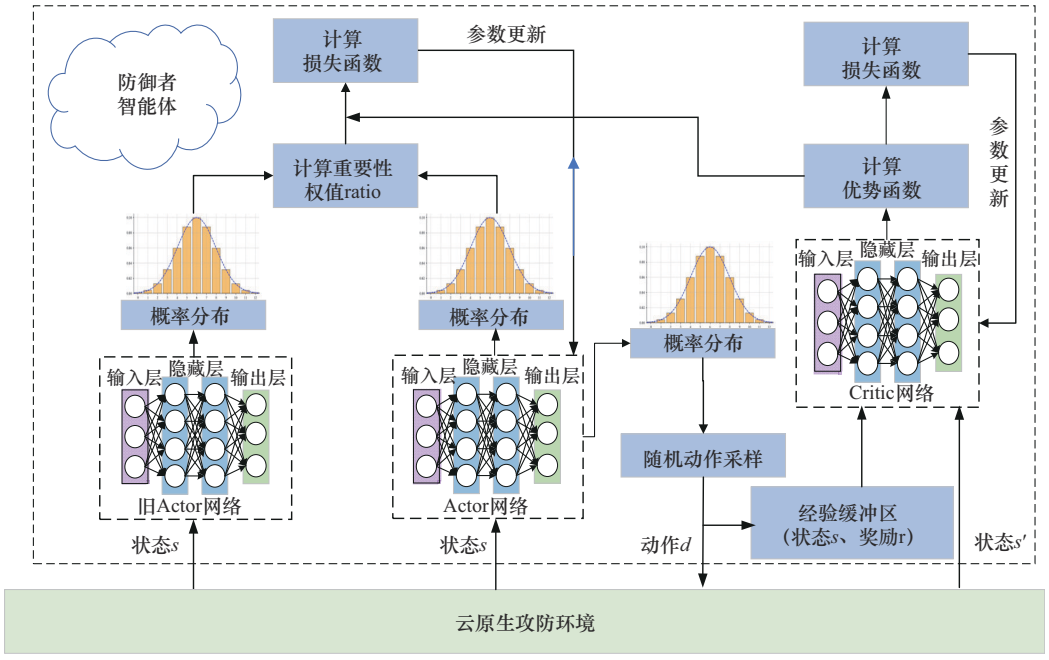


图2　防御者智能体 IPPO 算法的策略更新训练过程

出动作概率，计算2者动作概率的比值得到重要性权值，并将其与每个动作的优势函数相乘，得到Actor网络的损失函数

$$L(\theta) = E\left[\left(\frac{\pi_\theta(d_t|s_t)}{\pi_{\theta\,\mathrm{old}}(d_t|s_t)}\right)\hat{A}_t\right] \tag{14}$$

在训练过程中，策略更新幅度太大或太小都会降低收敛速度，因此使用裁剪函数clip可以将策略更新幅度限制在一定范围内，有效防止策略更新幅度过大，优化后的Actor网络的损失函数为

$$L(\theta) = E\left[\min\left(\frac{\pi_\theta(d_t|s_t)}{\pi_{\theta\,\mathrm{old}}(d_t|s_t)}\hat{A}_t,\ \mathrm{clip}\left(1-\varepsilon,\frac{\pi_\theta(d_t|s_t)}{\pi_{\theta\,\mathrm{old}}(d_t|s_t)},1+\varepsilon\right)\hat{A}_t\right)\right] \tag{15}$$

其中，ε为设置的裁剪参数，clip函数具体定义为

$$\mathrm{clip}(l,x,h) = \begin{cases} l, & x < l \\ x, & l \leqslant x \leqslant h \\ h, & x > h \end{cases} \tag{16}$$

然后根据Actor网络损失函数对Actor网络进行更新$\theta' \leftarrow \theta + \alpha\nabla L(\theta)$，重复上述步骤直到循环迭代结束，更新旧Actor网络参数。具体的算法过程如算法1所示。

算法1　基于IPPO的配置转换决策算法

输入　trains, steps, batch, γ, α, ε

输出　攻防均衡策略对(π_A^*,π_D^*)

1)初始化攻击者Actor网络参数θ^A, θ_{old}^A和Critic网络参数ω^A

2)初始化防御者Actor网络参数θ^D, θ_{old}^D和Critic网络参数ω^D

3) for episode in range(trains)

4)　初始化环境和状态s

5)　初始化经验缓冲区$D_A \leftarrow \varnothing$, $D_D \leftarrow \varnothing$

6)　for step in range (steps)

7)　//收集训练样本

8)　根据系统状态s输出策略$\pi_D \leftarrow \theta_{\mathrm{old}}^D(s)$，$\pi_A \leftarrow \theta_{\mathrm{old}}^A(s)$

9)　根据当前策略采样动作$d \leftarrow \pi_D(s)$，$a \leftarrow \pi_A(s)$

10)　环境中执行动作得到奖励r^D、r^A以及下一状态s_{next}

11) 将训练样本存入经验回放池$D_D \leftarrow (s, d_t, r_t^D, s')$，$D_A \leftarrow (s, a_t, r_t^A, s')$

12) 环境状态$s \leftarrow s_{\mathrm{next}}$

13)　if step % batch ==0 or step ==steps

14)　//更新Critic网络参数ω^A、ω^D

$$L(\omega) = E\left[\left(\sum_{i=t}^{T}\gamma^{i-t}r_i - V_\omega(s_t)^2\right)\right]$$

15)　for i =1,2,⋯,D do

16)　$L(\theta) = E\left[\min\left(\frac{\pi_\theta(d_t|s_t)}{\pi_{\theta\,\mathrm{old}}(d_t|s_t)}\hat{A}_t,\mathrm{clip}\left(1-\varepsilon,\frac{\pi_\theta(d_t|s_t)}{\pi_{\theta\,\mathrm{old}}(d_t|s_t)},1+\varepsilon\right)\hat{A}_t\right)\right]$//更新Actor网络参数$\theta^A$、$\theta^D$

17)　end for

18)　$\theta_{\mathrm{old}}^A \leftarrow \theta^A$、$\theta_{\mathrm{old}}^D \leftarrow \theta^D$

19)　end if

20)　step← step+1

21) end for

4　实验设计及结果分析

4.1　实验设计

为评估该策略选取方法的防御效能，本文使用kubernetes与Docker Swarm编排平台搭建了容器云集群实验环境。集群共包括5台x64服务器，分为1个管理节点和4个计算节点。计算节点主要负责完成对容器服务的创建、部署、调度和删除等，从而构建云原生系统配置资源池。管理节点负责系统配置的调度管理，主要是根据防御动作完成对应配置的转换工作。本文选择了云原生架构主要的5类技术栈：容器运行时T_0、网络管理T_1、服务网格T_2、编排工具T_3和数据库T_4进行系统配置构建，每类技术栈包括2种不同组件K_0= {Containerd, Docker}、K_1= {Open vSwitch, Cilium}、K_2= {Istio, Linkerd}、K_3= {Kubernetes, Docker Swarm}、K_4= {Redis, PostgreSQL}，生成如图3所示的云原生系统异构配置资源池，因此，防御者的防御动作数量（即异构配置总数C=$\{c_0,c_1,\cdots,c_{31}\}$）共有32个。

为了构建不同类型的攻击者，本文将攻击者按照从高到低的攻击能力划分为3种类型：type= {Low, Medium, High}，通过CNNVD筛选了近年来上述10种应用组件的120个高危漏洞，为每种类型的攻击者构建相应的攻击动作集，它们针对各应用组件的攻击能力如表1所示。

实验平台采用的服务器CPU型号为Intel Core i9-10900K CPU、3.70GHz、RAM为64 GB、显卡为GeForce GTX 2080 Ti，同时使用Python3.7和Py-

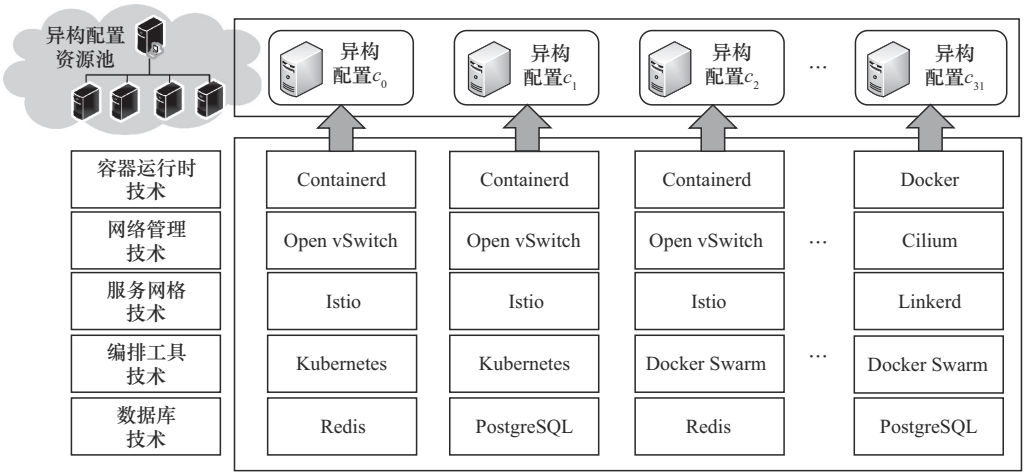


图3　云原生系统异构配置资源池

torch1.12.0 软件构建仿真环境。本文算法的神经网络参数、折扣系数、学习率等具体参数设置如表 2 所示。

表 1　不同类型攻击者的攻击能力

应用组件	Low	Medium	High
Containerd	1.8	3.0	4.0
Docker	0	1.6	2.4
Open vSwitch	1.8	2.6	3.2
Cilium	0	1.5	2.6
Istio	1.6	2.4	3.4
Linkerd	0.8	1.8	3.2
Kubernetes	1.4	2.4	3.0
Docker Swarm	2.0	2.8	4.0
Redis	1.2	2.8	4.0
PostgreSQL	0	1.8	3.4

表 2　智能体参数设置

参数名称	数值
trains	20 000
steps	100
隐藏层维度	128
学习率 α	1×10^{-4}
折扣系数 γ	0.95
裁剪参数 ε	0.2
Batch Size	512

4.2　算法收敛性分析

为检验 CTG-IPPO 的收敛性以及对不同超参数的敏感性，设计了两组对照实验分别考察学习率参数 α 与裁剪参数 ε 对算法训练过程的影响。图 4 和图 5 分别呈现了不同学习率和裁剪参数配置下的防御收益随训练步数的收敛变化曲线。其中，图 4 是在固定裁剪参数 ε=0.2 的条件下，测试学习率参数 $\alpha=5\times10^{-5}$、1×10^{-4}、5×10^{-4} 时的方法性能；图 5 是在学习率参数 $\alpha=1\times10^{-4}$ 的条件下，测试裁剪参数 ε=0.1、0.2、0.3 时的方法性能。实验结果表明，不同参数设置下，本文方法均可达到收敛并且最终收敛后的防御收益基本保持一致，但是当学习率参数 $\alpha=1\times10^{-4}$、裁剪参数 ε=0.2 时收敛速度最快，参数设置的偏高或者偏低均降低了方法的收敛速度。基于上述分析，本文最终选定神经网络学习率参数 $\alpha=1\times10^{-4}$ 与裁剪参数 ε=0.2 作为实验的参数选择。

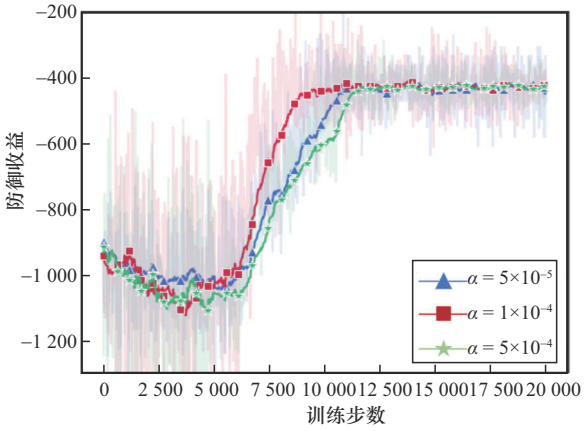


图4　不同学习率参数下的收敛情况

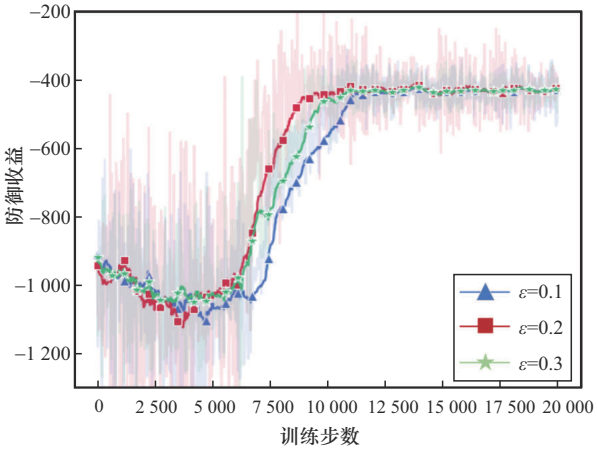


图5　不同裁剪参数下的收敛情况

4.3　策略防御效能分析

首先，为验证CTG-IPPO所求策略的有效性，引入移动目标防御经典策略的固定循环策略（FCS, fixed circle strategy）与平均随机策略（URS, uniform random strategy）作为基线进行实验对比。在FCS下，防御者按照预定义的固定顺序循环切换系统配置（如$c_0 \to c_1 \to \cdots \to c_n \to c_0$）；在URS下，防御者在每一时间步以均匀概率从所有可用配置中随机选择某一配置，即各配置被选中的概率相等，各策略在相同实验环境下的防御效能对比如图6所示。从图6可以看出，FCS因配置转换过程具备确定性，攻击者能够通过历史交互数据快速识别其转换规律，并据此优化攻击策略，导致防御效能急剧下降，最终收敛时的防御收益显著偏低；URS通过引入随机性在一定程度上增加了攻击者推断防御策略的难度，延缓了其策略学习过程，但由于其分布概率在训练过程中保持静态，攻击者仍可逐渐通过大量交互样本估计出防御行为分布，并最终逼近最优攻击策略，因此URS的防御效果仍存在明显上限。相比之下，CTG-IPPO通过持续进行梯度优化，始终保持策略空间的高不确定性，使防御策略分布有效偏离攻击者的预期从而实现出色的防御效果。实验结果表明，在面对具备策略学习能力的攻击者时，仅依靠随机性或固定模式的移动目标防御机制难以维持长期有效的防御能力，而基于博弈均衡的动态策略优化是提升策略有效性的关键。

其次，为了证明CTG-IPPO不需要攻击类型分布概率先验知识的假设依赖，本节设置了完全信息条件下CTG-PPO的对照实验。具体来说，在CTG-PPO中，假设攻击类型分布概率对于防御者为已知信息，根据Low、Medium、High分布概率的不同，实验设置了3个混合攻击者（0.5, 0.3,0.2）、（0.2,0.5,0.3）、（0.2,0.3,0.5），分别记为CTG-PPO1、CTG-PPO2和CTG-PPO3。CTG-IPPO仅知晓攻击者类型空间{Low, Medium, High}，对其具体分布不具备任何先验知识。从理论上看，由于信息劣势，CTG-IPPO所求策略的防御性能必定弱于CTG-PPO，然而从图7可以看到，虽然在训练初期CTG-PPO凭借先验知识实现了更快的收敛速度，但CTG-IPPO通过持续的策略探索与优化，其最终获得的防御收益值与各CTG-PPO结果均十分接近，证明了CTG-IPPO面对混合攻击时不需要其分布概率的先验知识，也能逼近最优防御策略。

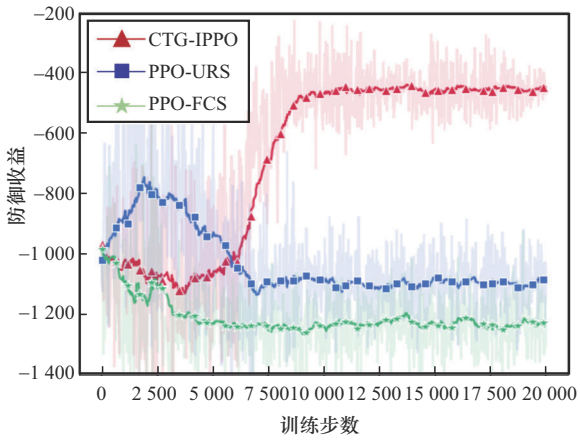
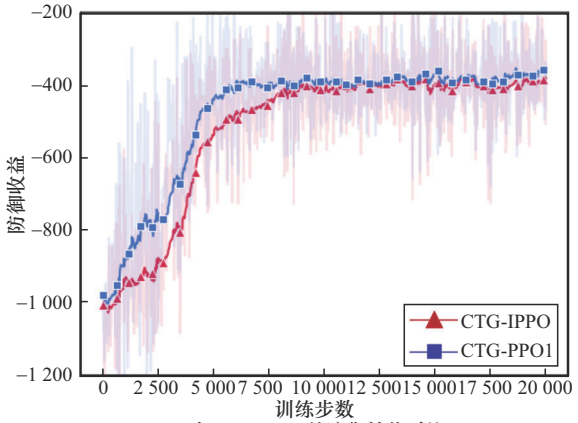
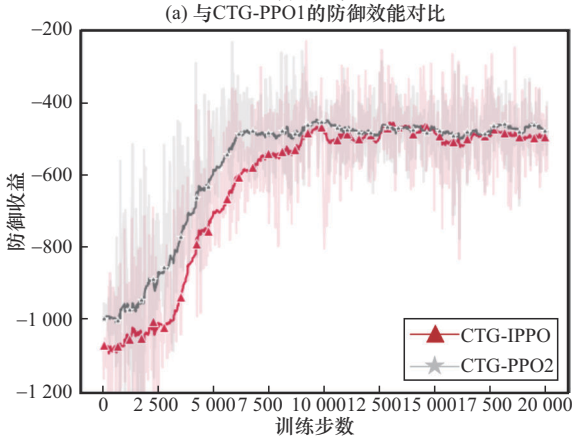


图6　不同防御策略的防御效能对比

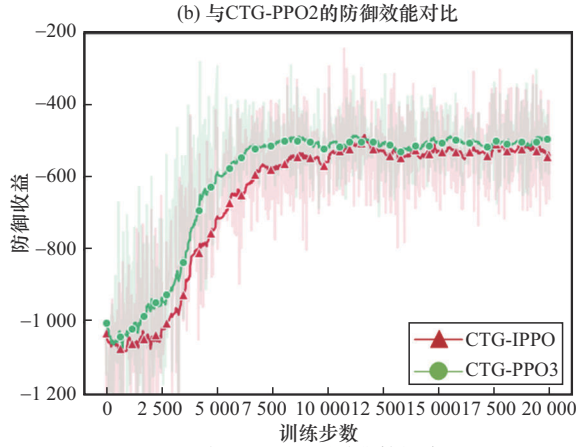
虽然CTG-PPO因已知攻击类型分布概率，其收敛速度更快且收敛收益更高，但该方法高度依赖先验知识的准确性，而且一旦攻击类型的分布概率发生变化或与先验知识不符时，将导致防御策略的效能快速下降。图8所示的实验结果证实了这一结论。该实验将一个攻击类型分布概率为（0.3,0.4, 0.3）的混合攻击者分别输入训练好的各个模型中，然后进行多次攻防交互得到防御收益均值。从实验结果可以看出，CTG-PPO1、CTG-PPO2和CTG-PPO3的防御效果大大降低，而CTG-IPPO得益于训练过程中自适应博弈机制，可以快速适应攻击策略的变化，因此得到的防御收益值远远优于其他模型。



(a) 与CTG-PPO1的防御效能对比



(b) 与CTG-PPO2的防御效能对比



(c) 与CTG-PPO3的防御效能对比

图7　与CTG-PPO的防御效能对比

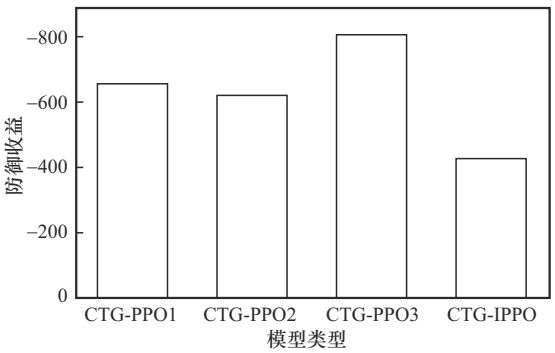
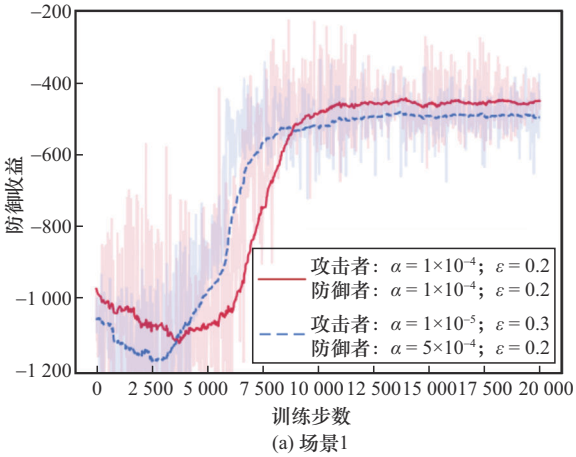
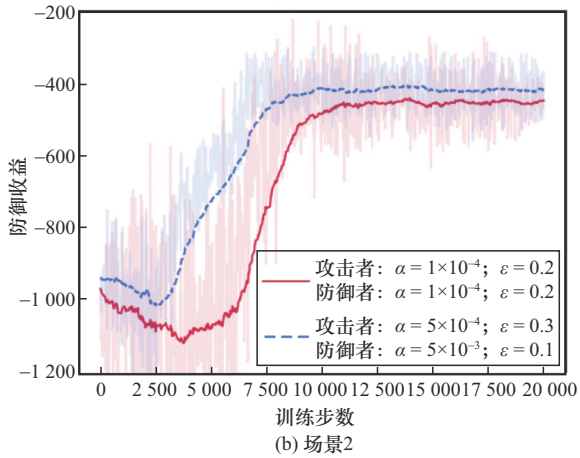


图8　不同模型的防御效能测试

为进一步验证本文方法在更具一般性的“非对等智能”场景下的防御性能，通过调节攻击方与防御方智能体的学习率与裁剪系数，构建了2种不同的非对等智能实验场景。实验结果如图9所示，其中，实线代表攻防智能体在“对等智能”设定下的基线性能，虚线则对应2种“非对等智能”条件下的结果。从图9(a)可以看出，攻击者智能体的策略学习能力高于防御者，策略收敛时防御收益有所降低。从图9(b)可以看出，防御者智能体的策略学习能力高于攻击者，防御收益较基线有所提升。从图9可以看出，尽管非对等智能条件下防御收益会根据防御者智能程度的高低出现一定波动，但最终均能达到收敛状态，证明本文方法在非对等智能环境下的鲁棒性与适应性。



(a) 场景1



(b) 场景2

图9　“非对等智能”条件下的防御效能对比

4.4　算法性能对比

为进一步验证CTG-IPPO在云原生不完全信息博弈决策场景中的优越性，本文选取了2种具有代表性的深度强化学习算法进行对比，分别为基于

Advantage Actor-Critic 架构的 A2C 算法和基于信任域策略优化方法的 TRPO 算法，并结合本文实验环境对其进行适应性改造，设计了 CTG-IA2C 与 CTG-ITRPO，如图 10 所示，3 类算法在相同训练周期内的防御收益收敛曲线表现出较大差异。相较 CTG-IA2C 与 CTG-ITRPO，CTG-IPPO 防御收益分别提升约 42.87% 与 34.71%。首先，CTG-IA2C 采用直接策略梯度更新机制，其更新步长固定且缺乏对策略更新幅度的有效约束，因此在面对高度随机的多智能体环境时，容易出现策略梯度估计方差过大、更新不稳定等问题，从而导致收敛性能受限，易收敛于次优解。其次，尽管 CTG-ITRPO 通过引入 KL 散度约束以保证策略更新的单调性，理论上具备更稳定的收敛特性，但其每一迭代步均需计算 Fisher 信息矩阵并求解约束优化问题，计算开销显著提高。相比之下，CTG-IPPO 利用裁剪函数与重要性采样机制，不仅避免了策略更新幅度过大导致的收敛效果较差的问题，也通过重复利用训练样本提升了学习的效率，能够更好地适应对实时性要求较高的云原生攻防环境的决策场景。

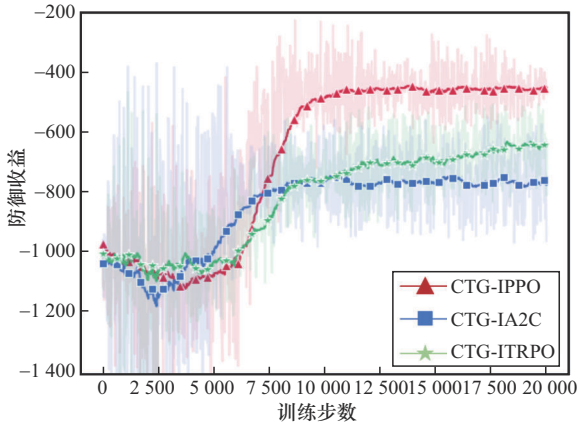


图 10　不同方法的防御效能对比

5　结束语

为应对云原生环境攻击者类型多样化且攻击行为日趋智能化的挑战，本文提出了基于贝叶斯马尔可夫博弈和独立强化学习算法的云原生移动目标防御决策方法。在分析云原生环境攻防过程中考虑了攻击者的策略学习能力，结合 CVSS 定义的收益函数构建贝叶斯马尔可夫博弈的移动目标防御模型，然后构建了 CTG-IPPO 求解最优防御策略。实验验证了 CTG-IPPO 求得策略的防御收益接近攻击类型分布信息已知的防御策略，并且在性能上优于 CTG-IA2C 和 CTG-ITRPO。

参考文献：

[1] ZENG Q Y, KAVOUSI M, LUO Y H, et al. Full-stack vulnerability analysis of the cloud-native platform[J]. Computers & Security, 2023, 129: 103173.

[2] TORKURA K A, SUKMANA M I H, MEINEL C. Integrating continuous security assessments in microservices and cloud native applications[C]// Proceedings of the Proceedings of the10th International Conference on Utility and Cloud Computing. New York: ACM Press, 2017: 171-180.

[3] SHAMEEM AHAMED W S, ZAVARSKY P, SWAR B. Security audit of docker container images in cloud architecture[C]//Proceedings of the 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). Piscataway: IEEE Press, 2021: 202-207.

[4] PINCONSCHI E, BUI Q C, ABREU R, et al. Maestro: a platform for benchmarking automatic program repair tools on software vulnerabilities[C]// Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. New York: ACM Press, 2022: 789-792.

[5] TAN J L, JIN H, ZHANG H Q, et al. A survey: When moving target defense meets game theory[J]. Computer Science Review, 2023, 48: 100544.

[6] 曾威, 扈红超, 李凌书, 等. 容器云中基于 Stackelberg 博弈的动态异构调度方法[J]. 网络与信息安全学报, 2021, 7(3): 95-104.
ZENG W, HU H C, LI L S, et al. Dynamic heterogeneous scheduling method based on Stackelberg game model in container cloud[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 95-104.

[7] ZHANG H W, MI Y, LIU X H, et al. A differential game approach for real-time security defense decision in scale-free networks[J]. Computer Networks, 2023, 224: 109635.

[8] TAN J L, JIN H, HU H, et al. WF-MTD: evolutionary decision method for moving target defense based on wright-fisher process[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(6): 4719-4732.

[9] LI Q X, WU J P. Optimizing the effectiveness of moving target defense in a probabilistic attack graph: a deep reinforcement learning approach[J]. Electronics, 2024, 13(19): 3855.

[10] 张帅, 郭云飞, 孙鹏浩, 等. 云原生下基于深度强化学习的移动目标防御策略优化方案[J]. 电子与信息学报, 2023, 45(2): 608-616.
ZHANG S, GUO Y F, SUN P H, et al. Moving target defense strategy optimization scheme for cloud native environment based on deep reinforcement learning[J]. Journal of Electronics & Information Technology, 2023, 45(2): 608-616.

[11] KIM S, YOON S, CHO J H, et al. DIVERGENCE: deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework[J]. IEEE Transactions on Network and Service Management, 2022, 19(4): 4834-4846.

[12] HE W Z, TAN J L, GUO Y F, et al. Flipit game deception strategy selection method based on deep reinforcement learning[J]. International Journal of Intelligent Systems, 2023, 2023(1): 5560416.

[13] FENG Y M, ZHANG W Z, FENG Z J, et al. An MTD-driven hybrid defense method against DDoS based on Markov game in multi-controller SDN-enabled IoT networks[C]//Proceedings of the 2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2024: 1-6.

[14] ABDEL MESSIH G I. RESONANT: reinforcement learning based moving target defense for detecting credit card fraud[D]. Blacksburg: Virginia Tech, 2023.

[15] SEO S, MOON H, LEE S, et al. D3GF: a study on optimal defense performance evaluation of drone-type moving target defense through game theory[J]. IEEE Access, 2023, 11: 59575-59598.

[16] MA T C, XU C Q, YANG S J, et al. An intelligent proactive defense against the client-side DNS cache poisoning attack via self-checking deep reinforcement learning[J]. International Journal of Intelligent Systems, 2022, 37(10): 8170-8197.

[17] HE W Z, TAN J L, GUO Y F, et al. A deep reinforcement learning-based deception asset selection algorithm in differential games[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 8353-8368.

[18] DE WITT C S, GUPTA T, MAKOVIICHUK D, et al. Is independent learning all you need in the StarCraft multi-agent challenge? [J]. arXiv Preprint, arXiv: 2011.09533, 2020.

[19] JIN H, LI Z, ZOU D Q, et al. DSEOM: a framework for dynamic security evaluation and optimization of MTD in container-based cloud[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1125-1136.

[20] SENGUPTA S, CHOWDHARY A, HUANG D J, et al. Moving target defense for the placement of intrusion detection systems in the cloud[M]// Decision and Game Theory for Security. Berlin: Springer, 2018: 326-345.

[21] ANAND P, SINGH Y, SELWAL A, et al. IVQFIoT: an intelligent vulnerability quantification framework for scoring Internet of Things vulnerabilities[J]. Expert Systems, 2022, 39(5): e12829.

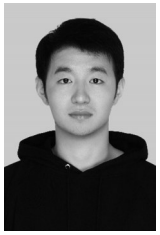
[22] FUDENBERG D, TIROLE J. Game theory [M]. Boston: Massachusetts Institute of Technology Press, 2012.

[23] POVEDA J I, KRSTIĆ M, BAŞAR T. Fixed-time Nash equilibrium seeking in time-varying networks[J]. IEEE Transactions on Automatic Control, 2022, 68(4): 1954-1969.

[24] AI S, KOE A S V, HUANG T. Adversarial perturbation in remote sensing image recognition[J]. Applied Soft Computing, 2021, 105: 107252.

[25] JIN J, XU Y. Optimal policy characterization enhanced proximal policy optimization for multitask scheduling in cloud computing[J]. IEEE Internet of Things Journal, 2021, 9(9): 6418-6433.

[作者简介]



耿致远（1996– ），男，河南孟州人，信息工程大学助理工程师，主要研究方向为网络主动防御。

张恒巍（1978– ），男，河南洛阳人，博士，信息工程大学教授、博士生导师，主要研究方向为网络安全博弈、人工智能对抗攻击与防御。

谭晶磊（1994– ），男，山东章丘人，博士，信息工程大学讲师，主要研究方向为移动目标防御。

齐高鑫（1998– ），男，黑龙江宁安人，信息工程大学博士生，主要研究方向为复杂网络、网络安全博弈。