基于不完全信息随机博弈与 Q-learning 的防御决策方法

张红旗[1,2]，杨峻楠[1,2]，张传富[1,2]
（1. 信息工程大学三院，河南 郑州 450001；2. 河南省信息安全重点实验室，河南 郑州 450001）

摘 要：针对现有随机博弈大多以完全信息假设为前提，且与网络攻防实际不符的问题，将防御者对攻击者收益的不确定性转化为对攻击者类型的不确定性，构建不完全信息随机博弈模型。针对网络状态转移概率难以确定，导致无法确定求解均衡所需参数的问题，将 Q-learning 引入随机博弈中，使防御者在攻防对抗中通过学习得到的相关参数求解贝叶斯纳什均衡。在此基础上，设计了能够在线学习的防御决策算法。仿真实验验证了所提方法的有效性。

关键词：网络攻防；随机博弈；Q-learning；贝叶斯纳什均衡；防御决策
中图分类号：TP393.08
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018145

Defense decision-making method based on incomplete information stochastic game and Q-learning

ZHANG Hongqi[1,2], YANG Junnan[1,2], ZHANG Chuanfu[1,2]
1. The Third Institute, Information Engineering University, Zhengzhou 450001, China
2. Henan Province Key Laboratory of Information Security, Zhengzhou 450001, China

Abstract: Most of the existing stochastic games are based on the assumption of complete information, which are not consistent with the fact of network attack and defense. Aiming at this problem, the uncertainty of the attacker's revenue was transformed to the uncertainty of the attacker type, and then a stochastic game model with incomplete information was constructed. The probability of network state transition is difficult to determine, which makes it impossible to determine the parameter needed to solve the equilibrium. Aiming at this problem, the Q-learning was introduced into stochastic game, which allowed defender to get the relevant parameter by learning in network attack and defense and to solve Bayesian Nash equilibrium. Based on the above, a defense decision algorithm that could learn online was designed. The simulation experiment proves the effectiveness of the proposed method.

Key words: network attack and defense, stochastic game, Q-learning, Bayesian Nash equilibrium, defense strategy

1 引言

近年来，信息安全事件日趋频繁，给网络安全带来了巨大的损失[1]。由于资源和能力的限制，防御者无法保证绝对的安全，因此亟需一种能够对攻防行为进行分析和防御策略选取的技术，使防御者在风险与投入之间达成一种均衡。博弈论与网络攻防所具有的目标对立性、关系非合作性和策略依存性高度契合[2]，通过建立网络攻防的博弈模型及对模型的均衡求解来研究攻击行为和指导防御决策逐渐成为一个研究热点[3]。

针对网络防御决策问题，本文首先对网络攻防

收稿日期：2018−04−03；**修回日期**：2018−07−10
通信作者：杨峻楠，624519905@qq.com
基金项目：国家高技术研究发展计划（“863”计划）基金资助项目（No.2014AA7116082, No.2015AA7116040）
Foundation Item: The National High Technology Research and Development Program of China (863 Program) (No.2014AA7116082, No.2015AA7116040)

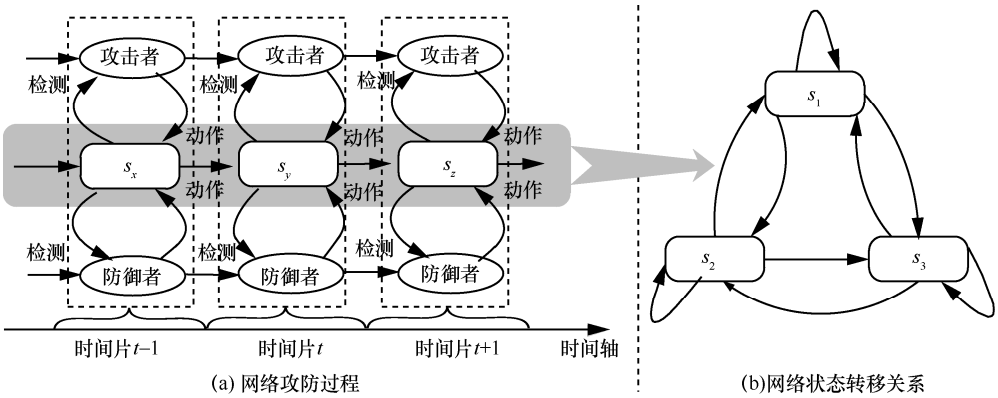
进行分析。将攻防对抗进行离散化处理，整个过程看作一系列时间片，每个时间片都包含且只包含一个网络状态。网络攻防过程可以用图 1(a)进行描述。每个时间片中攻防双方检测当前网络状态，依据状态选择执行攻防动作。网络系统在攻防双方的联合行为作用下由一个状态转移到另一个状态。时间片中的网络状态转移关系如图 1(b)所示，网络状态之间的转移不仅受到攻防动作的影响，还受到网络中其他一些复杂因素的影响，这导致网络状态的转移存在随机性。由于网络攻防的非合作性，攻防双方无法完全掌握对手信息，且只能通过检测网络来了解对手的行动，这会比动作的执行时间至少延迟一个时间片，并导致每个时间片中双方不知道对手当前时间片的行动，因此每个时间片是一个不完全信息静态博弈。但从整个攻防过程来看，当前时间片的攻防动作会影响后续时间片的网络状态，进而影响攻防双方的决策，这又是一个动态变化的过程。

本文采用不完全信息静态博弈与马尔可夫决策结合的随机博弈[4]来对网络攻防进行建模分析，其博弈系统在参与人的联合行动下由一个状态转移到另一个状态，符合上述分析的网络攻防对抗过程。

目前，国内外基于随机博弈的网络安全研究已取得一定进展，但是应用在网络安全领域的随机博弈模型大部分基于完全信息假设。文献[5]将攻击者的特权状态作为网络状态，构建完全信息随机博弈模型。文献[6]通过完全信息随机博弈分析攻击者与防御者之间的策略交互，利用纳什均衡指导防御决策。文献[7]提出一种完全信息随机博弈分析方法，保护电网免受协调攻击。文献[8]将随机博弈应用于物联网中用于平衡网络性能与安全水平，其所提方法比其实验中的基准方法更有效。由于网络攻防的竞争关系，掌握对方博弈信息非常困难，上述文献中的完全信息假设不能满足实际需求。因此部分学者开始采用更符合实际的不完全信息随机博弈进行研究。文献[9]针对移动目标防御决策问题，构建不完全信息随机博弈模型，并通过对比分析说明了此模型比其之前建立的完全信息随机博弈模型[10]实用性更高。但是，上述成果均未讨论随机博弈现有收益函数在网络攻防中的适用问题。首先，上述文献的收益函数都以已知转移模型为前提，但在实际中防御者往往无法确定网络状态转移概率，导致无法利用上述文献模型中的收益函数求解均衡。另外，网络具有动态性，其状态转移概率也会不定期变化，但上述文献没有提出相应的解决方案而是假设状态转移概率为定值。以上 2 点使上述文献的模型实用性较低。

为解决上述问题，借鉴强化学习思想，在不完全信息随机博弈中引入 Q-learning[11]算法。据统计，阿里云在 2017 年仅每天就要遭受 16 亿次左右的攻击，对于不同攻击者，可能每个攻防场景只会出现一次，但对于以阿里云为代表的防御者来说，其每天都要面对大量相同的攻防场景。通过引入 Q-learning 算法并将其由一个参与者扩展为可用于博弈的 2 个参与者，使防御者在大量相同的攻防场景中以增量求和的方式对收益进行在线学习和更新，不需要确定转移概率，就可以求解相应场景的贝叶斯纳什均衡，从而进行防御决策。

目前，Q-learning 在攻防领域已得到广泛应用。文献[12]提出一种基于 Q-learning 的 LDoS 攻击实时防御机制，该方法具有较好的实时性和灵敏性。文献[13]将 Q-learning 嵌入软件中，提供一种安全机制，该方法能够较快学会阻止恶意行为。文献[14]基于 Q-learning 设计了一种有效的智能电网脆弱性分析方法。上述研究取得了一定进展，但其将攻击者作为环



(a) 网络攻防过程　　(b)网络状态转移关系

图 1　网络攻防

境的一部分固定到系统中，不能体现攻防的对抗特点，也无法较好地满足网络攻防的策略依存性。

2 不完全信息随机博弈模型

实际的网络攻防对抗是一个复杂的过程，针对防御策略选取问题，对其进行一些必要的假设和简化。

对参与博弈的攻防双方做出如下假设。

1) 攻防双方都是理性的，追求最大的收益。

2) 防御者对攻击者收益函数的不确定性视为对攻击者类型的不确定性，且防御者对攻击者类型的概率分布有一个判断[11]。

对每个时间片中博弈所需的“信息”和“行动顺序”这 2 个基本要素做进一步假设。设当前博弈处于时间片 t 中，则在 t 中攻击者类型为其私有信息，双方的共同知识为：①时间片 t 中的状态和对应状态攻防双方可采取的动作；②攻击者在 t 采取的动作，在 t+1 会被防御者观察到，防御者在 t 采取的动作，在 t+1 会被攻击者观察到；③防御者对攻击者类型分布的概率判断。行动顺序：在每个时间片，攻防双方是同时行动的。这里的“同时”是一个信息概念，非时间概念，即尽管从时间概念上攻防双方的选择可能不在同一时刻，但只要攻防双方在行动时不知道对手的选择就认为是同时行动。

建立时间片之间的网络状态转移模型，即说明网络攻防是如何演变的。网络状态之间的转移存在随机性，因此选择概率方式对其进行描述。由于下一时间片的网络状态只与当前状态有关，与之前状态无关，即状态的转移符合一阶马尔可夫假设，因此转移概率为 $P(x_{t+1}\mid x_{0:t},a_{0:t},d_{0:t})=P(x_{t+1}\mid x_t,a_t,d_t)$。由于网络具有动态性，从长期看，转移概率是变化的，但在短期内可以认为其是一个定值。

常用变量及其含义如表 1 所示。

在上述基础上构建博弈模型。

定义 1 不完全信息随机博弈模型（II-SGM, incomplete information stochastic game model）是一个八元组，即 $II\text{-}SGM=(N,S,\Theta,P_A,A,D,\pi,U)$，其中，各变量定义如下。

1) $N=(attacker,defender)$ 为参与博弈的局中人，分别代表网络攻击者和防御者。

2) $S=(s_1,s_2,\cdots,s_n)$ 为随机博弈状态集合，每个博弈状态代表一个网络状态。

3) $\Theta=\left(\theta_i\mid i=1,2,\cdots,n\right)$ 为攻击者的类型空间。

4) $P_A=(p_A(s_i,\theta_1),p_A(s_i,\theta_2),\cdots,p_A(s_i,\theta_n))$ 为防御者对攻击者类型分布的概率判断。

5) $A=(A_1,A_2,\cdots,A_n)$ 为攻击者动作集合，其中，$A_k=\{a_1,a_2,\cdots,a_m\}$ 为攻击者在博弈状态 s_k 的动作集合。

6) $D=(D_1,D_2,\cdots,D_n)$ 为防御者动作集合，其中，$D_k=\{d_1,d_2,\cdots,d_m\}$ 为防御者在博弈状态 s_k 的动作集合。

7) $\pi=(\pi_a,\pi_d)$ 为攻防策略集合。

8) U 为攻防双方的收益函数集合。

攻防策略与攻防动作是 2 个不同的概念，攻防策略是攻防动作的规则，而不是动作本身。攻防策略以概率的形式规定攻防双方在每个网络状态选

表 1 主要符号及其含义

符号	含义	符号	含义
N	局中人集合	$II\text{-}SGM$	不完全信息随机博弈模型
S	博弈状态集合	$\pi_a(s_k,\theta_i)$	在状态 s_k 类型为 θ_i 的攻击者的策略
Θ	攻击者类型空间	$\pi_d(s_k)$	在状态 s_k 的防御者策略
A	攻击者动作集合	$\sigma_a(s_k,a_m,\theta_i))$	在状态 s_k 类型为 θ_i 的攻击者选择 a_m 的概率
D	防御者动作集合	$\sigma_d(s_k,d)$	在状态 s_k 防御者选择 d 的概率
π	攻防策略集合	$Q_a(s_i,a,d,\theta_j)$	双方采取动作（a,d）后攻击者的期望累积收益
θ_i	攻击者类型	$Q_a(s_i,a,d,\theta_j)$	双方采取动作（a,d）后防御者的期望累积收益
a_m	攻击动作	$V_a(s_i,\pi_a(s_i,\theta_j),\pi_d(s_i),\theta_j)$	双方采取策略 $(\pi_a(s_i,\theta_j),\pi_d(s_i))$ 后攻击者的期望累积收益
d_m	防御动作	$V_d(s_i,\pi_a(s_i,\theta_j),\pi_d(s_i),\theta_j)$	双方采取策略 $(\pi_a(s_i,\theta_j),\pi_d(s_i))$ 后防御者的期望累积收益
π_a	攻击者策略	$p_A(s_i,\theta_n)$	防御者认为在状态 s_i 攻击者类型为 θ_n 的概率
π_d	防御者策略	P_A	防御者对攻击者类型分布的概率判断
α	学习率	π^{ε}	ε-greedy 策略
γ	折扣因子	Q^*	正确的状态−动作收益

择什么动作，如 $\pi_a(s_k,\theta_i)=(\sigma_a(s_k,a_1,\theta_i),\cdots,\sigma_a(s_k,a_m,\theta_i))$ 表示类型为 θ_i 的攻击者在网络状态 s_k 的策略，$\sigma_a(s_k,a_m,\theta_i)$ 为其选择动作 a_m 的概率。

收益函数 $U=(Q,V)$ 为攻防双方策略制定的依据，其中，$Q=(Q_a,Q_d)$ 为攻防双方的状态—动作收益函数，如 $Q_d(s_i,a,d,\theta_j)$ 为在状态 s_i 下攻击者类型为 θ_j 时双方采取动作 (a,d) 后防御者的收益函数。网络攻防对抗是一个序列式决策问题，决策不仅影响当前的收益，还会影响未来的收益，故 Q 应该包含立即回报和未来回报，是一种累积收益函数。本文采用折扣因子 $\gamma(0<\gamma<1)$ 表示攻防双方对未来回报的偏好。由于网络的随机性，导致同一状态下相同的攻防动作也会引向不同的攻防过程，因此无法对具体的收益进行度量，只能通过期望来表示其效用。综上，Q 应该是期望累积收益函数。$V=(V_a,V_d)$ 为攻防双方的状态收益函数，如 $V_d(s_i,\pi_a(s_i,\theta_j),\pi_d(s_i),\theta_j)$ 为在状态 s_i 下当攻击者类型为 θ_j 时，攻防双方采取策略 $(\pi_a(s_i,\theta_j),\pi_d(s_i))$ 后防御者的期望收益函数，其中，有

$$V_h(s_i,\pi_a(s_i,\theta_j),\pi_d(s_i),\theta_j)=\sum_{a\in A_i}\sigma_a(s_i,a,\theta_j)\sum_{d\in D_i}\sigma_d(s_i,d)Q_h(s_i,a,d,\theta_j),h\in N \tag{1}$$

采用海萨尼转换[15]来处理不完全信息博弈，引入虚拟参与人“自然”按照相应概率选择转移的状态和攻击者类型，使攻防双方对状态转移的不完全信息和防御者对攻击收益的不完全信息转换成对“自然”行动的不完美信息。在此基础上，依据博弈模型构建一个时间片的攻防博弈树，如图 2 所示。其中，N 为虚拟参与人“自然”，A 为攻击者，D 为防御者。博弈由上一个时间片转移到当前时间片后，“自然” N 按照转移概率先选择当前时间片的网络状态，再按防御者对攻击者类型分布的概率判断选择攻击者的类型，攻击者 A 和防御者 D 都能观察 N 对状态的选择，但只有攻击者能观察到 N 对攻击者类型的选择。A 和 D 观察后依据策略选择自己的动作并获得回报。当前时间片的博弈结束后，博弈转移到下一个时间片。

3　Q-learning 与贝叶斯纳什均衡求解

贝叶斯纳什均衡是攻防双方的最优策略，双方无法再通过单方面改变自己的策略来提高收益。本节主要对随机博弈在网络攻防中求解均衡存在的问题进行研究。

3.1　网络攻防中的均衡求解参数问题

定义 2　贝叶斯纳什均衡。在博弈状态 s_i 上，对于所有 θ_j 的所有攻击策略 $\pi_a(s_i,\theta_j)$，有

$$V_a(s_i,\pi_a^*(s_i,\theta_j),\pi_d^*(s_i),\theta_j)\geqslant V_a(s_i,\pi_a(s_i,\theta_j),\pi_d^*(s_i),\theta_j) \tag{2}$$

对于所有防御策略 $\pi_d(s_i)$，有

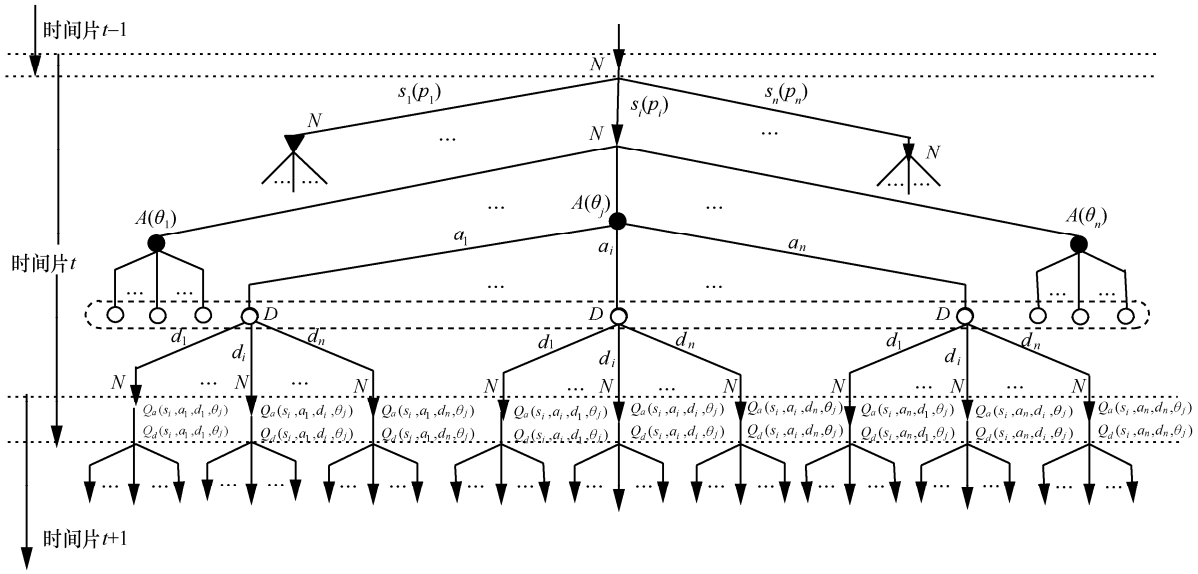


图 2　攻防博弈树

$$\sum_{j=1}^{n} p_A(s_i,\theta_j)V_d(s_i,\pi_a^*(s_i,\theta_j),\pi_d^*(s_i),\theta_j)$$
$$\geqslant \sum_{j=1}^{n} p_A(s_i,\theta_j)V_d(s_i,\pi_a^*(s_i,\theta_j),\pi_d(s_i),\theta_j) \quad (3)$$

则策略 $(\pi_a^*(s_i,\theta_1),\cdots,\pi_a^*(s_i,\theta_n),\pi_d^*(s_i)$ 是网络状态 s_i 上的一个贝叶斯纳什均衡。

本文 II-SGM 的均衡解就是每个状态贝叶斯纳什均衡解的集合。其中，状态 s_i 上贝叶斯纳什均衡求解问题可以归纳为

$$f:(Q_a(s_i,\theta_1),\cdots,Q_a(s_i,\theta_n),\ Q_d(s_i,\theta_1),\cdots,$$
$$Q_d(s_i,\theta_n),P_A)\rightarrow(\pi_a^*(s_i,\theta_1),\cdots,\pi_a^*(s_i,\theta_n),\pi_d^*(s_i)$$

函数 f 是一个从收益和概率判断到均衡的映射，是一个二次规划问题，旨在满足某些约束条件下寻找最大收益。目前，有很多成熟的算法可以表示 f，如 Lebg-plex[16]等，这里不再做进一步研究。本文重点对函数 f 的参数 Q 进行讨论。

随机博弈中收益函数 Q 经典的定义方式为

$$Q_h(s,a,d,\theta_j)=R_h(s,a,d,\theta_j)+$$
$$\gamma\sum_{s'\in S} p(s'\mid s,a,d)V_h(s',\pi_a^*(s',\theta_j),\pi_d^*(s'),\theta_j) \quad (4)$$

收益函数 Q 包含立即回报 $R_h(s,a,d,\theta_j)$ 和未来回报 $\gamma\sum_{s'\in S} p(s'\mid s,a,d)V_h(s',\pi_a(s',\theta_j),\pi_d(s'),\theta_j)$，并以期望表示收益，符合第 2 节对网络攻防收益函数的分析。但在实际中，防御者往往无法确定其参数中的网络状态转移概率 $p(s'\mid s,a,d)$ [17]，这导致防御者无法利用式(4)得到准确的收益，将其设为 $Q*$，也就不能求解贝叶斯纳什均衡；现有文献都假设网络状态转移概率为定值，这与实际不符，实际中的网络具有动态性，网络状态转移概率是一个变化的值，收益 $Q*$ 应该依据转移概率的变化而变化。以上 2 点造成网络防御决策中现有的随机博弈模型的实用性较低。

3.2 基于 Q-learning 的均衡求解参数确定方法

为了在网络状态转移概率未知的情况下得到求解贝叶斯纳什均衡所需的参数 $Q*$，同时为了适应网络的动态性，能够实时对参数 $Q*$ 进行更新，借鉴强化学习的思想，引入基于数据驱动的 Q-learning 算法，从攻防对抗中对 $Q*$ 进行在线学习。

Q-learning 是一种被广泛应用的典型免模型强化学习算法，能够求解马尔可夫决策的最大回报和最优策略问题。虽然 Q-learning 和随机博弈的基础理论都是马尔可夫决策，但是 Q-learning 中只有一个参与人，其决策只受环境的影响，每个状态下参与人最大收益的动作是固定的，而随机博弈中有 2 个参与人，其决策不仅受环境的影响，还要依赖于对手的决策，每个状态下对手采取不同策略，自己的最大收益策略也不同，将 Q-learning 应用在随机博弈中还需要对其进行改进。

Q-learning 中 agent 通过与环境的交互可以获得回报和环境状态转移的知识，算法中知识用收益函数 Q 来表示，通过更新 Q 来进行学习。其收益函数 Q 可定义为

$$Q(s,d)=(1-\alpha)Q(s,d)+\alpha(R(s,d)+\gamma V(s',\pi^q(s')) \quad (5)$$
$$\pi^q=\arg\max_d Q(s,d) \quad (6)$$

式(5)的收益函数 Q 包含立即回报 $R(s,d)$ 和未来回报 $\gamma V(s',\pi(s'))$。通过引入学习率 α 以增量求和的方式对收益进行更新，其本质是一种求平均的期望收益，符合第 2 节对网络攻防收益函数的分析，而且学习率 α 的引入使收益 $Q*$ 的获取不再需要转移概率，解决了现有定义方式存在的问题。不足的是 Q-learning 定义的收益函数 Q 只与环境和参与人本身行动有关，且其策略 π^q 也不适用于双人博弈，所以对式(5)和式(6)进行改进，将 Q-learning 的收益函数由一个参与者扩展为 2 个参与者，同时策略由 π^q 变为贝叶斯纳什均衡策略，改进后的 Q 为

$$Q_h(s,a,d,\theta_j)=(1-\alpha)Q_h(s,a,d,\theta_j)+$$
$$\alpha(R_h(s,a,d,\theta_j)+\gamma V_h(s',\pi_a^*(s',\theta_j),\pi_d^*(s'))) \quad (7)$$

本文 II-SGM 的状态—动作收益函数 Q 由式(7)定义，式(7)中学习率 α 的引入决定了其需要通过学习机制来寻找 $Q*$。下面，对 Q-learning 的学习机制进行改进，使其能够满足攻防博弈的需求。

防御者在进行学习或决策时，不仅要考虑网络系统，还要考虑攻击者行为，改进后的学习机制如图 3 所示。防御者检测网络状态 s 并从动作空间选择动作执行，网络系统接收攻防双方的动作后，给予双方相应奖赏反馈 R 同时更新共同知识，并转移到下一个状态 s'，防御者根据收到的奖赏和共同知识利用式(1)和式(7)来更新状态—动作收益 (Q_a,Q_d)、状态收益 (V_a,V_d) 和贝叶斯纳什均衡 $(\pi_a^*(s,\theta_1),\cdots,\pi_a^*(s,\theta_n),\pi_d^*(s))$ 以完成学习，同时检测新的网络状态做出新的决策。需要注意的是，这里

并没有限制攻击者必须采用相同的学习机制，防御者的学习不依赖于攻击者是否有学习机制或采用何种学习机制。

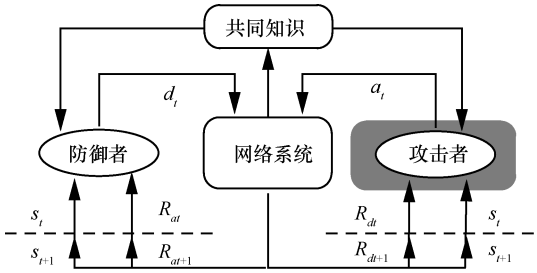


图 3　防御者学习机制

Q-learning 中的环境采用本文 II-SGM 中的博弈状态进行建模，参与者的行为采用 II-SGM 中的攻防动作集合来定义。

Q-learning 中探索与利用问题经典的解决方法为 SoftMax 算法和 ε-greedy 算法[18]。SoftMax 算法所需计算量较大，为满足决策实时性需求，本文选用 ε-greedy 算法进行探索和利用的折中。算法以 $\varepsilon(0\leqslant\varepsilon\leqslant 1)$ 的概率进行探索，探索时随机选择下一步动作；以 $1-\varepsilon$ 的概率进行利用，利用时按照贝叶斯纳什均衡策略选择下一步动作。ε-greedy 算法为

$$\pi^{\varepsilon}(s_i)=\begin{cases}\pi_d^*(s_i),\ \text{以概率}1-\varepsilon\\ \text{从}D_i\text{中以均匀概率选取动作,以概率}\varepsilon\end{cases}$$

立即回报 $R_h(s,a,d,\theta_j)$ 采用文献[5-6]中的方法对其进行量化。

使用改进的 Q-learning 通过增量求和的方式定义 Q，再通过学习得到 Q^*，不再需要确定网络状态转移概率 $p(s'|s,a,d)$，而且当网络中转移概率发生变化时，通过在线学习能够对 Q^* 进行实时调整。依据改进的 Q-learning 学习得到的 Q^*，可以解得更准确的贝叶斯纳什均衡，得到更合理的防御策略。具体的学习算法和收敛性分析在第 4 节进行详细介绍。

4　网络防御策略选取

本节首先对将贝叶斯纳什均衡作为攻防策略的合理性以及均衡的存在性进行分析，然后提出结合 II-SGM 与改进 Q-learning 的防御策略选取算法并从理论上对算法的收敛性进行分析。

4.1　贝叶斯纳什均衡策略分析

网络攻防具有策略依存性：①防御者在考虑策略时，会对攻击者做出预测并根据预测结果选择收益最高的策略；②攻击者会预测到防御者对自己的预测进而调整自己的策略；③作为理性的防御者，还应该进一步考虑到攻击者会因预测到自己对攻击者的预测而做出调整，此时防御者会进一步优化防御策略；④作为理性的攻击者，也应该考虑到防御者会因预测到自己对防御者的预测而重新调整策略。这是一个螺旋上升的过程，最终攻防双方的最优策略会达到一种稳定状态，此时双方的策略都是对方策略的最优响应，没有人有积极性再去偏离这个结果。根据第 3 节贝叶斯纳什均衡的定义，均衡中每一个参与人的策略，都是在其他参与人策略下使自己获得最高收益的策略，是针对其他人做出的最优响应，所以贝叶斯纳什均衡是攻防对抗稳定状态的最优策略，防御者选择均衡作为防御策略可以获得最高的期望收益，是合理的。

贝叶斯纳什均衡存在性：攻防动作集 A 和 D 是由攻防双方可采取的攻防动作组成的有限集，攻击者类型也是一个有限集，所以本文的攻防随机博弈在每个状态 S 都是一个有限不完全的信息静态博弈，而任何有限不完全的信息静态博弈都存在贝叶斯纳什均衡[4]。

4.2　防御策略选取算法

在每个时间片，算法使用 II-SGM 对网络攻防进行分析，依据现有的 Q 求解贝叶斯纳什均衡，进行防御决策。决策后利用改进的 Q-learning 对本次对抗进行学习，并对 Q 进行更新。算法 1 第 1)~5)步是依据攻防场景对 II-SGM 初始化，第 6)~8)步是利用现有知识求解贝叶斯纳什均衡，第 9)~17)步是进行防御策略选取和在线学习，其中，第 10)~11)步是依据 ε-greedy 对探索和利用进行折中并完成策略选取，第 12)~16)步是依据攻防对抗的阶段结果对 Q 和 π^* 进行更新来完成学习，第 17)步是收敛条件，第 18)步返回收敛后的防御策略。具体算法如算法 1 所示。

算法 1　自适应防御策略选取算法

输入　II-SGM，奖赏折扣 γ，探索概率 ε，更新步长 α，收敛精度 δ，稳定时长 z

输出　防御动作 d，防御策略 π_d^*

begin

1)初始化 $S=(s_1,s_2,\cdots,s_n)$

2)初始化 $\Theta=\left(\theta_i\middle|i=1,2,\cdots,n\right)$

3)初始化 $P_A=(p_A(s,\theta_1),p_A(s,\theta_2),\cdots,p_A(s,\theta_n))$

4)初始化 $A=(A_1,A_2,\cdots,A_n)$ 和 $D=(D_1,D_2,\cdots,D_n)$

5)初始化 $Q=(Q_a,Q_d)$ //利用先验知识对其初始化

6) $(\pi_a^*(s,\theta_1),\cdots,\pi_a^*(s,\theta_n),\pi_d^*(s)=Lebg-plex(Q(s),P_A),s\in S$ //求解贝叶斯纳什均衡

7) $V_h(s_i,\pi_a^*(s_i,\theta_j),\pi_d^*(s_i),\theta_j)=\sum_{a\in A_i}\sigma_a^*(s_i,a,\theta_j)\sum_{d\in D_i}\sigma_d^*(s_i,d)Q_h(s_i,a,d,\theta_j),h\in N,\theta_j\in\Theta$

8) $s=get(E)$ //从网络 E 中获取当前网络状态

9)repeat:

10) $d=\pi^{\varepsilon}(s)$ //利用 ε-greedy 选取防御动作

11) output d ;//将防御动作反馈给防御者

12) $s'=get(E)$ //从 E 中获取新的状态

13) $Q_h(s,a,d,\theta_j)=(1-\alpha)Q_h(s,a,d,\theta_j)+\alpha(R_h(s,a,d,\theta_j)+\gamma V_h(s',\pi_a^*(s',\theta_j),\pi_d^*(s')),\theta_j),h\in N,\theta_j\in\Theta$ //根据阶段性结果对 Q 进行更新学习

14) $(\pi_a^*(s,\theta_1),\cdots,\pi_a^*(s,\theta_n),\pi_d^*(s)=Lebg-plex(Q(s),P_A),s\in S$ //对贝叶斯纳什均衡进行更新

15) $V_h(s_i,\pi_a^*(s_i,\theta_j),\pi_d^*(s_i),\theta_j)=\sum_{a\in A_i}\sigma_a^*(s_i,a,\theta_j)\sum_{d\in D_i}\sigma_d^*(s_i,d)Q_h(s_i,a,d,\theta_j),h\in N,\theta_j\in\Theta$

16) $s=s'$

17)until $\left|\pi_d^{t-i^*}-\pi_d^{t-i-1^*}\right|\leqslant\delta,\forall i\in[0,z]$

18)output π_d^*

end

4.3 算法分析

4.3.1 收敛性分析

算法 1 是一个学习算法，能否通过学习收敛到正确的收益 Q^* 并得到贝叶斯纳什均衡，直接影响算法的可用性，所以这里对其收敛性进行分析。

算法 1 是对 Q-learning 的改进，改进后的算法要想收敛到 Q^*，首先需要满足 Q-learning 算法收敛的无限抽样假设和关于学习率的假设。

假设 1 每个状态 $s\in S$ 和每个行为 $a\in A$ 都要经常被访问到。

假设 2 对所有的 s、t、a，学习率 α_t 要满足如下条件。

1) $0\leqslant\alpha_t(s,a)<1,\sum_{t=0}^{\infty}\alpha_t(s,a)=\infty,\sum_{t=0}^{\infty}[\alpha_t(s,a)]^2<\infty$。

2)如果 $(s,a)\neq(s_t,a_t)$，则 $\alpha_t(s,a)=0$。其表示如果某个行为不是当前所采取的行为，则此行为的学习率为 0。

在满足上述假设的基础上对算法的收敛性进行分析。

Szepesvari 等[19]已经证明，如果满足以下 3 个条件，那么由式 $Q_{t+1}=(1-\alpha_t)Q_t+\alpha_t[P_tQ_t]$ 通过迭代进行的学习过程，能够以概率 1 收敛到 Q_t^*。

1) $0\leqslant\alpha_t(s,a,d)<1,\sum_{t=0}^{\infty}\alpha_t(s,a,d)=\infty$。

2)存在一个数 $0<\gamma<1$ 和一个序列 $\lambda_t\geqslant 0$（λ_t 以概率 1 收敛到 0），使所有的 Q_t 有

$$\|P_tQ_t-P_tQ_t^*\|\leqslant\gamma\|Q_t-Q_t^*\|+\lambda_t \tag{8}$$

3) $Q^*=E[P_tQ^*]$。

下面，对本文算法是否满足上述 3 个条件进行分析。

本文算法算子为

$$P_tQ=R_t(s,a,d,\theta_j)+\gamma V(s',\pi_a(s',\theta_j),\pi_d(s'),\theta_j) \tag{9}$$

条件 1)

由于本算法满足假设 2，因此条件 1)成立。

条件 2)

定义 3 对于所有 Q_t，定义范数 $\|Q-Q^*\|$ 为

$$\begin{aligned}\|Q_t-Q_t^*\|&=\max_j\max_s\|Q_t^j(s)-Q_t^{j*}(s)\|_{(j,s)}\\&=\max_j\max_s\max_{a,d}|Q_t^j(s,a,d,\theta_j)-Q_t^{j*}(s,a,d,\theta_j)|\end{aligned} \tag{10}$$

在上述基础上证明如下引理。

引理 1 对于所有 Q_t，有

$$\|P_tQ_t-P_tQ_t^*\|\leqslant\gamma\|Q_t-Q_t^*\| \tag{11}$$

证明 由定义 3 可知，范数 $\|P_tQ_t-P_tQ_t^*\|$ 为

$$\begin{aligned}\|P_tQ_t-P_tQ_t^*\|&=\max_j\max_s|P_tQ_t^j(s)-P_tQ_t^{j*}(s)|\\&=\max_j\max_s\gamma|\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)-\\&\quad\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)|\end{aligned} \tag{12}$$

联立式(10)和式(12)可知，欲证引理 1，只需证明

$$\max_j\max_s\gamma|\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)-\sum_{a\in A_s}\pi_a^*(s)\cdot\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)|\leqslant\gamma\max_j\max_s\max_{a,d}|Q_t^j(s,a,d,\theta_j)-Q_t^{j*}(s,a,d,\theta_j)|$$

$$\Rightarrow$$

$$|\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)-\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)|$$

$$\leqslant\max_{a,d}|Q_t^j(s,a,d,\theta_j)-Q_t^{j*}(s,a,d,\theta_j)| \tag{13}$$

即可。

当 j 为攻击者时，分 2 种情况进行证明。

①当 $\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)\geqslant\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)$ 时，有

$$\begin{aligned}&|\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)-\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)|\\&\leqslant\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^j(s)-\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)\\&\leqslant\max_{a,d}|Q_t^j(s,a,d,\theta_j)-Q_t^{j*}(s,a,d,\theta_j)|\end{aligned} \tag{14}$$

②当 $\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)<\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)$ 时，有

$$\begin{aligned}&|\sum_{a\in A_s}\pi_a(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)-\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d^*(s)Q_t^{j*}(s)|\\&\leqslant\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d(s)Q_t^{j*}(s)-\sum_{a\in A_s}\pi_a^*(s)\sum_{d\in D_s}\pi_d(s)Q_t^j(s)\\&\leqslant\max_{a,d}|Q_t^j(s,a,d,\theta_j)-Q_t^{j*}(s,a,d,\theta_j)|\end{aligned} \tag{15}$$

由式(14)和式(15)可知，式(13)成立。

当 j 为防御者时，式(13)也成立，其证明过程与 j 为攻击者时类似，这里不再叙述。综上可知，引理 1 成立。又因为 $\lambda_t\geqslant 0$，故式(6)成立，因此本文算法满足条件 2)。

条件 3)

$$\begin{aligned}&Q^{j*}(s,a,d,\theta_j)\\&=R^j(s,a,d,\theta_j)+\gamma\sum_{s'\in S}p(s'|s,a,d)V^{j*}(s',\pi_a(s',\theta_j),\pi_d(s'),\theta_j)\\&=\sum_{s'\in S}p(s'|s,a,d)[R^j(s,a,d,\theta_j)+\\&\gamma\sum_{a\in A_{s'}}\pi_a(s',\theta_j)\sum_{d\in D_{s'}}\pi_d(s')Q^{j*}(s',a,d,\theta_j)]\\&=\mathrm{E}[P_t\ Q^{j*}(s,a,d,\theta_j)]\end{aligned} \tag{16}$$

由式(16)可知，本文算法满足条件 3)。

综上可知，本文算法满足 3 个条件，所以本文算法收敛。

4.3.2　复杂性分析

空间复杂度：设 $|S|$ 为状态数，$|A|$ 为每个状态攻击者的措施数，$|D|$ 为每个状态防御者的措施数，攻击者类型数为 n，则需要维护 n+1 个 Q 表，故空间复杂度为 $O((n+1)|S||A||D|)$。

时间复杂度：每次决策的时间复杂度主要是在策略选取后对 $\pi^*(s)$ 的更新，更新需要对一个不完全信息静态博弈进行求解，本文采用 Lebg-plex 算法对其进行计算，Lebg-plex 算法的平均时间复杂度为 $O((\max(|A|,|D|))^3)$。

5　仿真实验与分析

5.1　仿真实验场景

采用图 4 所示的典型网络信息系统场景进行仿真实验。Web 服务器和堡垒主机部署在非隔离区（DMZ, demilitarized zone），内网由文件服务器和数据库服务器组成。防火墙的安全策略为外部用户仅允许访问 Web 服务器的 FTP、HTTP 服务和堡垒主机上的 SMTP 服务，其他网络节点和端口均进行阻断。所有的攻击均来自外网。参考 NVD 数据库给出目标网络脆弱性信息，如表 2 所示。参考美国麻省理工大学的攻防行为数据库[20]给出防御动作，如表 3 所示。

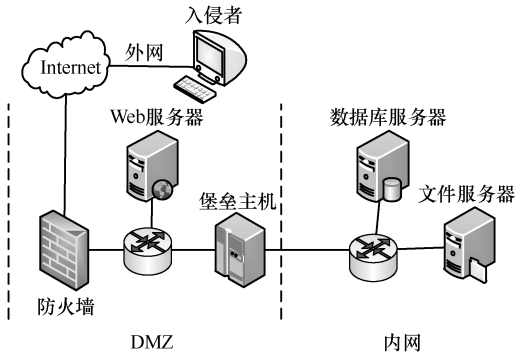


图 4　仿真网络信息系统场景

表 2　网络脆弱性信息

序号	主机	CVE 编号	服务
a_1	Web 服务器	CVE-2015-1635	HTTP
a_2	Web 服务器	CVE-2017-7269	IIS
a_3	Web 服务器	CVE-2014-8517	FTP
a_4	堡垒主机	CVE-2014-3556	SMTP
a_5	文件服务器	CVE-2014-4877	FTP
a_6	数据库服务器	CVE-2013-4730	FTP
a_7	数据库服务器	CVE-2016-6662	MySQL

表 3 防御动作描述

原子防御动作	d_1	d_2	d_3	d_4	d_5	d_6
Renew root data	√		√		√	√
Limit SYN/ICMP packets		√				
Install Oracle patches	√					
Reinstall Listener program	√				√	
Uninstall delete Trojan		√				√
Limit access to MDSYS		√		√		
Restart Database server			√	√	√	
Delete suspicious account		√				
Add physical resource	√			√	√	√
Repair database			√	√		
Limit packets from ports	√	√	√			√

5.2 构建仿真场景的攻防随机博弈模型

参与人 $N=(attacker,defender)$ 为外网攻击者与内网防御者。

随机博弈状态 $S=(s_1,s_2,s_3,s_4,s_5,s_6,s_7)$。设 H_1 为入侵主机，H_2 为 Web 服务器，H_3 为堡垒主机，H_4 为文件服务器，H_5 为数据库服务器。攻击者在各主机的权限有 3 种：不具任何权限（none），具有普通用户权限（user），具有 root 用户权限（root）。网络状态由攻击者在各个主机的权限组成，其中，H_1(root)表示攻击者在入侵主机具有 root 权限。网络状态具体含义如表 4 所示。网络状态转移关系如图 5 所示。

表 4 网络状态

状态	描述
s_1	H_1(root),H_2(user),H_3(none),H_4(none),H_5(none)
s_2	H_1(root),H_2(user),H_3(user),H_4(none),H_5(none)
s_3	H_1(root),H_2(root),H_3(user),H_4(none),H_5(none)
s_4	H_1(root),H_2(user/root),H_3(root),H_4(none),H_5(none)
s_5	H_1(root),H_2(user/root),H_3(root),H_4(user),H_5(none)
s_6	H_1(root),H_2(user/root),H_3(root),H_4(none),H_5(user)
s_7	H_1(root),H_2(user/root),H_3(root),H_4(none/user),H_5(root)

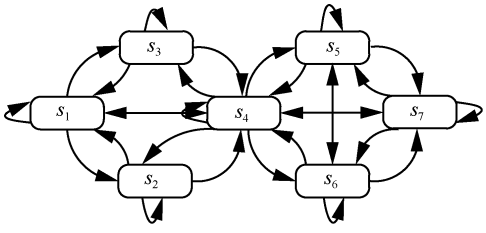


图 5 网络状态转移关系

攻击者类型 $\Theta=(\theta_1,\theta_2)$，其中，$\theta_1$ 为高能力攻击者，θ_2 为低能力攻击者。

攻击者类型分布的先验概率为

$$(p_A(s_1,\theta_1),p_A(s_1,\theta_2);p_A(s_2,\theta_1),p_A(s_2,\theta_2);p_A(s_3,\theta_1),p_A(s_3,\theta_2);p_A(s_4,\theta_1),p_A(s_4,\theta_2);p_A(s_5,\theta_1),p_A(s_5,\theta_2);p_A(s_6,\theta_1),p_A(s_6,\theta_2);p_A(s_7,\theta_1),p_A(s_7,\theta_2))=(0.1,0.9;0.2,0.8;0.15,0.85;0.3,0.7;0.2,0.8;0.4,0.6;0.5,0.5)$$

攻击者动作集合 $A=(A_1,A_2,A_3,A_4,A_5,A_6,A_7)$。其中，$A_1=\{a_1,a_2,a_3,a_4\}$、$A_2=\{a_2\}$、$A_3=\{a_4\}$、$A_4=\{a_5,a_6,a_7\}$、$A_5=\{a_6,a_7\}$、$A_6=\{a_7\}$、$A_7=\{\}$。

防御者动作集合 $D=(D_1,D_2,D_3,D_4,D_5,D_6,D_7)$。其中，$D_1=\{\}$、$D_2=\{d_1\}$、$D_3=\{d_3\}$、$D_4=\{d_2,d_5,d_6\}$、$D_5=\{d_1\}$、$D_6=\{d_1,d_3\}$、$D_7=\{d_2,d_5,d_6\}$。

状态—动作收益函数 Q 用式(7)进行定义，状态收益函数 V 用式(1)进行定义。

参考文献[5-6]的方法对其进行量化，以状态 s_4 为例，其量化结果为

$$R(s_4,\theta_1)=\begin{bmatrix}R_{a,d}(s_4,a_5,d_2,\theta_1) & R_{a,d}(s_4,a_5,d_5,\theta_1) & R_{a,d}(s_4,a_5,d_6,\theta_1)\\ R_{a,d}(s_4,a_6,d_2,\theta_1) & R_{a,d}(s_4,a_6,d_5,\theta_1) & R_{a,d}(s_4,a_6,d_6,\theta_1)\\ R_{a,d}(s_4,a_7,d_2,\theta_1) & R_{a,d}(s_4,a_7,d_5,\theta_1) & R_{a,d}(s_4,a_7,d_6,\theta_1)\end{bmatrix}=\begin{bmatrix}(31,-30) & (37,-40) & (22,-20)\\ (39,-40) & (54,-50) & (32,-30)\\ (48,-50) & (36,-40) & (35,-40)\end{bmatrix} \tag{17}$$

$$R(s_4,\theta_2)=\begin{bmatrix}R_{a,d}(s_4,a_5,d_2,\theta_2) & R_{a,d}(s_4,a_5,d_5,\theta_2) & R_{a,d}(s_4,a_5,d_6,\theta_2)\\ R_{a,d}(s_4,a_6,d_2,\theta_2) & R_{a,d}(s_4,a_6,d_5,\theta_2) & R_{a,d}(s_4,a_6,d_6,\theta_2)\\ R_{a,d}(s_4,a_7,d_2,\theta_2) & R_{a,d}(s_4,a_7,d_5,\theta_2) & R_{a,d}(s_4,a_7,d_6,\theta_2)\end{bmatrix}=\begin{bmatrix}(26,-15) & (25,-30) & (17,-20)\\ (32,-30) & (39,-38) & (27,-29)\\ (37,-42) & (32,-34) & (32,-36)\end{bmatrix} \tag{18}$$

对仿真场景中的状态转移概率进行设定，使仿真网络依此进行演进，以状态 s_4 为例，其状态转移概率为

$$p(s_4)=\begin{bmatrix}p(s_1|s_4,a_5,d_2) & p(s_1|s_4,a_5,d_5) & \cdots & p(s_1|s_4,a_7,d_6)\\ p(s_2|s_4,a_5,d_2) & \vdots & \ddots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ p(s_7|s_4,a_5,d_2) & \cdots & \cdots & p(s_7|s_4,a_7,d_6)\end{bmatrix}$$

$$=\begin{bmatrix} 0.6 & 0 & 0 & 0.5 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0.7 & 0 & 0 & 0.6 & 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0.6 & 0 & 0 & 0.7 & 0 & 0 & 0.7 \\ 0.3 & 0.1 & 0.1 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.1 \end{bmatrix} \tag{19}$$

在已知转移概率的前提下，利用式(1)和式(4)对本场景的贝叶斯纳什均衡进行求解，以状态 s_4 为例，其求解结果为

$$\sigma_d^*(s_4,d_2)=0.1,\sigma_d^*(s_4,d_5)=0.6,\sigma_d^*(s_4,d_6)=0.3$$
$$\sigma_a^*(s_4,a_5,\theta_1)=0.2,\sigma_a^*(s_4,a_6,\theta_1)=0.4,\sigma_a^*(s_4,a_7,\theta_1)=0.4,\ \sigma_a^*(s_4,a_5,\theta_2)=0.6,\sigma_a^*(s_4,a_6,\theta_2)=0.1,\ \sigma_a^*(s_4,a_7,\theta_2)=0.3$$

接下来，在防御者不知道转移概率的前提下，对本文算法的有效性进行测试，仿真实验使用 Python2.7 实现了本文的防御策略选取算法。

5.3　测试与结果分析

5.3.1　收敛性测试与分析

能否在合理的时间内学习到正确收益的 Q^* 解，得到正确的贝叶斯纳什均衡对于算法是否有效至关重要。对收敛性从 2 个方面进行仿真实验，分别测试不同参数设置及不同攻击策略对算法收敛性的影响。

1)不同参数设置对算法收敛性的影响

在学习率 α 满足假设 2 的条件下，对 3 种参数设置进行测试，具体如表 5 所示。

表 5　　实验参数设置

序号	ε	α	γ
设置 1	0.1	0.6	0.3
设置 2	0.2	0.4	0.6
设置 3	0.3	0.5	0.5

实验中，使用立即回报 R 对 Q^* 进行初始化，不引入其他先验知识。不同类型攻击者均采用贝叶斯纳什均衡策略进行攻击，以状态 s_4 为例，有

$$\sigma_a^*(s_4,a_5,a_6,a_7,\theta_1)=(0.2,0.4,0.4)$$
$$\sigma_a^*(s_4,a_5,a_6,a_7,\theta_2)=(0.6,0.1,0.3)$$

对状态 s_4 的防御策略变化进行记录，每当算法 1 第 16)步中的 $s=s_4$ 时就记录一次，结果如图 6 所示。其中，防御次数是指在状态 s_4 防御者决策的次数，即算法 1 第 16)步中 $s=s_4$ 的次数。

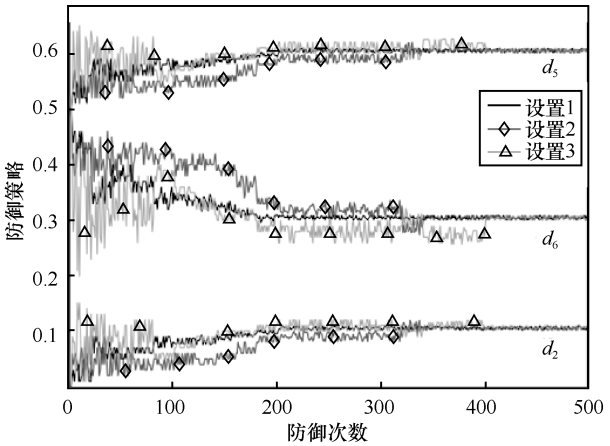


图 6　不同参数设置下状态 s_4 的防御策略变化

从图 6 可以看到，3 组参数设置都能够收敛，通过与已知转移概率计算的贝叶斯纳什均衡结果对比可以发现，3 组参数设置都能够收敛到正确的贝叶斯纳什均衡，即

$$\sigma_d(s_4,d_2)=\sigma_d^*(s_4,d_2)=0.1$$
$$\sigma_d(s_4,d_5)=\sigma_d^*(s_4,d_5)=0.6$$
$$\sigma_d(s_4,d_6)=\sigma_d^*(s_4,d_6)=0.3$$

不同参数设置的收敛速度不同，设置 1 在防御 200 次左右达到贝叶斯纳什均衡，而设置 2 和设置 3 在防御 350~400 次左右达才达到收敛，且设置 1 在学习过程中的振荡幅度也明显小于设置 2 和设置 3，由此可知，设置 1 比设置 2 和设置 3 更适用于本场景。

上述的仿真实验中仅使用了立即回报 R 对收益 Q 进行初始化。如果引入其他先验知识，那么还可以进一步增加算法的收敛速度。

2) 不同攻击策略对算法收敛性的影响

选择设置 1 为本实验参数，仍使用立即回报 R 对 Q 进行初始化。仿真分 2 种情况进行，情况 1：攻击者为理性，不同类型攻击者均采用贝叶斯纳什均衡策略进行攻击；情况 2：模拟现实中的复杂情况，假设攻击者为非理性，采用随机选择策略进行攻击，以状态 s_4 为例，其攻击策略为 $\pi_a(s_4,\theta_i)=\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)$，$i=1,2$，算法 1 第 16)步中状态 s= s_4 的防御策略变化如图 7 所示。

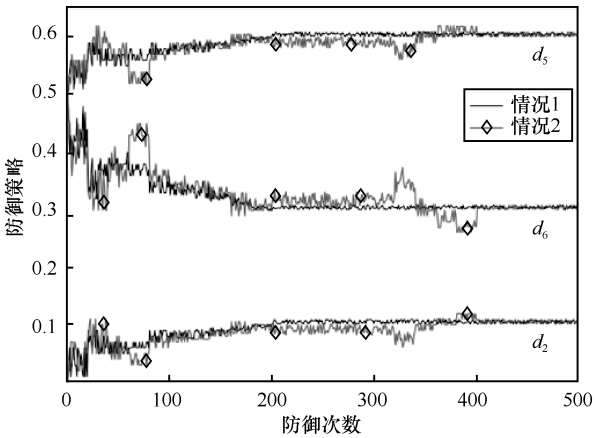


图 7 不同攻击策略下状态 s_4 的防御策略变化

从图 7 可以看到，当面对理性攻击者时，本文方法能够在 200 次左右的学习后快速收敛到贝叶斯纳什均衡策略；当面对非理性攻击者时，防御者仍能够在与攻击者对抗中进行学习，虽然学习过程中振荡幅度比情况 1 稍大且收敛速度比情况 1 稍慢，但在 400 次左右的学习后也能够达到收敛。综上可知，攻击者的策略会对本文方法的学习速度和振荡程度有一定影响，但不会影响最终的学习结果，这说明算法在实际的复杂环境中也能进行有效的防御决策。

5.3.2 适应性测试与分析

网络具有动态性，网络状态转移概率会不定期地发生变化，算法需要具有对状态转移概率变化做出适应性调整的能力才能进行有效的决策。选择设置 1 为本实验参数，仍使用立即回报 R 对 Q 进行初始化。不同类型攻击者均选用贝叶斯纳什均衡策略进行攻击，初始网络状态转移概率为式(19)，在经过一段攻防过程后转移概率发生变化，以状态 s_4 为例，转移概率变为式(20)。

$$p(s_4)=\begin{bmatrix} p(s_1|s_4,a_5,d_2) & p(s_1|s_4,a_5,d_5) & \cdots & p(s_1|s_4,a_7,d_6) \\ p(s_2|s_4,a_5,d_2) & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ p(s_7|s_4,a_5,d_2) & \cdots & \cdots & p(s_7|s_4,a_7,d_6) \end{bmatrix}$$
$$=\begin{bmatrix} 0.3 & 0 & 0 & 0.5 & 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0.7 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 0.3 & 0 & 0 & 0.4 \\ 0.2 & 0.3 & 0.1 & 0.2 & 0.1 & 0.3 & 0.05 & 0.2 & 0.3 \\ 0.5 & 0.3 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.2 & 0.3 \end{bmatrix} \tag{20}$$

算法第 16)步中状态 $s=s_4$ 的防御策略变化如图 8 所示。

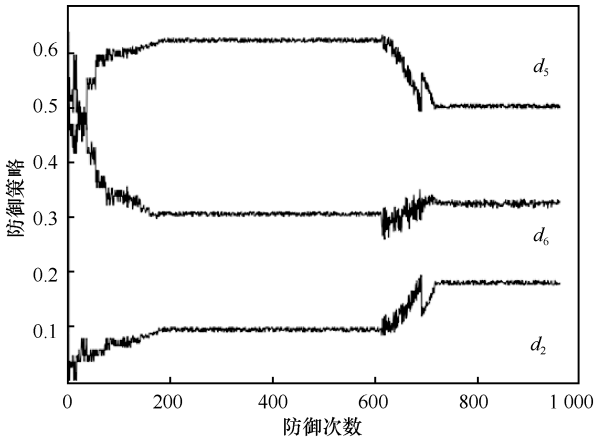


图 8 状态 s_4 的防御策略变化

从图 8 可以看到，初始状态下算法进行了第一次学习并达到收敛，当网络状态转移概率发生变化后，算法能够及时做出反应，进行第二次学习并再次达到收敛。第一次收敛经历了 200 次左右的学习，而第二次收敛只经历了 100 次左右的学习，这是因为初始状态只引入了较少的先验知识，所以需要较长的学习过程，而第二次由于网络的动态变化一般比较平稳，通常不会突然发生剧烈改变，由此转移概率的变化幅度较小，利用前一阶段的学习成果相当于引入了较多的先验知识，因此收敛速度较快。综上可知，算法能够在网络动态变化时及时做出调整，适应新的环境。

5.3.3 对比测试与分析

将本文方法与 4 种典型方法进行对比测试。不同类型攻击者采用均衡策略发起攻击，不同方法在状态 s_4 的防御策略变化如图 9 所示。为了直观地显示 5 种方法的不同，这里对比了选取防御动作 d_5 的概率。不同方法在状态 s_4 的防御收益变化如图 10 所示，为了避免只反映一次博弈的结果，在此取 50 次博弈的平均值，同时这也提供了一个更易理解的平滑曲线。

从图 9 可以看到，本文方法在经过 200 次左右的学习后收敛到了客观的均衡策略；文献[5-6,9]的方法由于初始计算时无法得到准确的转移概率导致其结果与客观的均衡存在误差，其中，文献[5-6]又由于以完全信息假设为前提，导致误差比文献[9]要稍大一些；单纯的 Q-learning 方法由于只能选择纯策略，而客观均衡是一个混合策略，导致其防御策略一直在振荡。

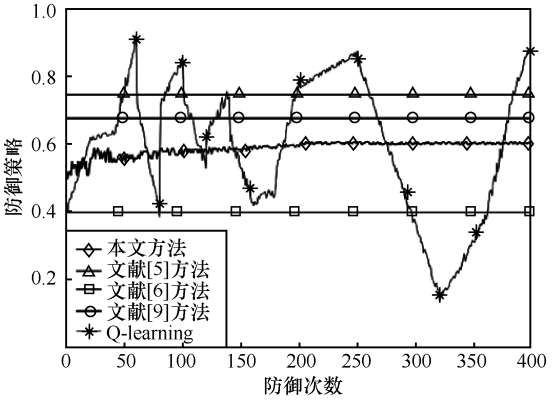


图 9　不同方法在状态 s_4 的防御策略变化

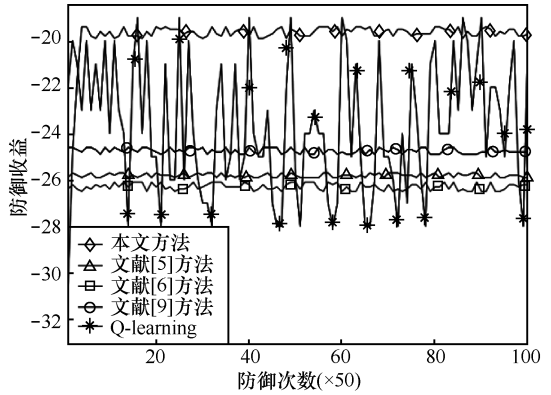


图 10　不同方法在状态 s_4 的防御收益变化

从图 10 可以看到，本文方法在初始阶段收益稍低于其他方法，但是当收敛到均衡策略以后，本文方法收益明显高于其他 4 种方法，这表明本文方法的防御策略在此场景要优于其他方法。

6　相关工作对比分析

本文方法与一些典型方法的研究对比结果如表 6 所示。文献[5-6,8-9]的理论基础为随机博弈，文献[12,14]的理论基础为 Q-learning，本文方法则是随机博弈与 Q-learning 的结合。与文献[5-6,8]相比，本文方法以不完全信息假设为前提，更符合攻防实际；与文献[5-6,8-9]相比，本文方法不需要已知转移模型，且能够适用于转移概率动态变化的情况；与文献[12,14]相比，本文方法能够充分考虑网络攻防的策略依存性。

将本文方法扩展至实际网络，其攻防对抗仍然符合随机博弈过程，防御决策原理仍然正确，但是需要针对具体网络的环境做一些调整。如果实际网络规模较小，则本文方案可直接使用；如果网络规模较大，一些网络的攻防策略集和攻击者类型数量会增加，导致博弈求解难度加大，还有一些网络采用 Q 表存储 Q 值时可能会消耗过多内存，同时收敛速度也会降低。文献[21]对第一类情况进行了总结与分析，给出了相应的解决方案。如遇到第一类情况，可参考文献[21]中的方法对博弈求解方式进行调整。如遇到第二类情况，可利用神经网络或深度学习等函数拟合的方式存储 Q 值。需要注意的是，并不是本文现有的博弈求解方式和以 Q 表存储 Q 值的方式存在问题，而是实际中不同网络对算法有不同的需求，本文现有的博弈求解方式和以 Q 表存储 Q 值的方式更适用于规模较小的网络，而文献[21]中提供的求解方式和神经网络（深度学习）存储 Q 值的方式则更适用于大规模网络。两者各有优势，由于这不是本文的研究重点，因此没有对其进行深入讨论。

7　结束语

近年来，防御决策问题逐渐成为网络安全领域的研究重点和热点。本文从有限信息和动态对抗的视角对网络攻防实际场景进行了分析，建立了不完全信息随机博弈模型，针对随机博弈中转移概率难以确定，无法得到贝叶斯纳什均衡的问题，将

表 6　　**本文方法与典型方法的评估比较**

方法	理论基础	模型假设	转移模型	转移概率	策略依存性	具体应用
文献[5]方法	随机博弈	完全信息	需已知	定值	已考虑	策略选取
文献[6]方法	随机博弈	完全信息	需已知	定值	已考虑	策略选取
文献[8]方法	随机博弈	完全信息	需已知	定值	已考虑	策略选取
文献[9]方法	随机博弈	不完全信息	需已知	定值	已考虑	策略选取
文献[12]方法	Q-learning	—	免模型	动态变化	未考虑	安全机制
文献[14]方法	Q-learning	—	免模型	动态变化	未考虑	脆弱性分析
本文方法	随机博弈+Q-learning	不完全信息	免模型	动态变化	已考虑	策略选取

Q-learning 引入随机博弈中，使防御者能从与攻击者对抗的过程中学习求解均衡所需的参数 Q^*，进而得到准确的贝叶斯纳什均衡。结合博弈模型和改进的 Q-learning，设计了能够在线学习的防御策略选取算法。本文不仅从理论上对算法的收敛性进行了证明，还通过仿真实验对其进行了验证。研究成果能在网络状态转移概率未知的情况下指导防御决策，比现有方法更具实用性。

参考文献：

[1] HU H, ZHANG H, LIU Y, et al. Quantitative method for network security situation based on attack prediction[J]. Security & Communication Networks, 2017(4): 1-19.

[2] HU H, LIU Y, ZHANG H, et al. Optimal network defense strategy selection based on incomplete information evolutionary game[J]. IEEE Access, 2018, PP(99): 1.

[3] FALLAH M. A puzzle-based defense strategy against flooding attacks using game theory[J]. IEEE Transactions on Dependable & Secure Computing, 2010, 7(1): 5-19.

[4] FILAR J, VRIEZE K. Competitive Markov decision processes[J]. Springer Berlin, 1996, 36(4): 343-358.

[5] 姜伟, 方滨兴, 田志宏, 等. 基于攻防随机博弈模型的防御策略选取研究[J]. 计算机研究与发展, 2010, 47(10): 1714-1723.
JIANG W, FANG B X, TIAN Z H, et al. Research on defense strategies selection based on attack-defense stochastic game model[J]. Journal of Computer Research and Development, 2010, 47(10): 1714-1723.

[6] LYE K W, WING J M. Game strategies in network security[J]. International Journal of Information Security, 2005, 4(1-2): 71-86.

[7] WEI L, SARWAT A, SAAD W, et al. Stochastic games for power grid protection against coordinated cyber-physical attacks[J]. IEEE Transactions on Smart Grid, 2016, PP(99): 1.

[8] ARFAOUI A, LETAIFA A B, KRIBECHE A, et al. A stochastic game for adaptive security in constrained wireless body area networks[C]//Consumer Communications & NETWORKING Conference. 2018: 1-7.

[9] LEI C, ZHANG H Q, WAN L M, et al. Incomplete information Markov game theoretic approach to strategy generation for moving target defense[J]. Computer Communications, 2018, 116: 184-199.

[10] LEI C, MA D H, ZHANG H Q. Optimal strategy selection for moving target defense based on Markov game[J]. IEEE Access, 2017, PP(99): 1.

[11] WATKINS C J C H, DAYAN P. Technical note: Q-learning[J]. Machine Learning, 1992, 8(3-4): 279-292.

[12] 刘陶, 何炎祥, 熊琦. 一种基于 Q 学习的 LDoS 攻击实时防御机制及其 CPN 实现[J]. 计算机研究与发展, 2011, 48(3): 432-439.
LIU T, HE Y X, XIONG Q. A Q-learning based real-time mitigating mechanism against LDoS attack and its modeling and simulation with CPN[J]. Journal of Computer Research and Development, 2011, 48(3): 432-439.

[13] RANDRIANSOLO A S, PYEATT L D. Q-learning: from computer network security to software security[C]//International Conference on Machine Learning and Applications. 2015: 257-262.

[14] YAN J, HE H, ZHONG X, et al. Q-learning-based vulnerability analysis of smart grid against sequential topology attacks[J]. IEEE Transactions on Information Forensics & Security, 2017, 12(1): 200-210.

[15] HARSANYI J C, SELTEN R. A general theory of equilibrium selection in games[M]. Boston: MIT Press, 1988.

[16] CORMEN T H, LEISERSON C E, RIVEST R L, et al. Introduction to algorithms[M]. Boston: MIT Press, 2009.

[17] 张恒巍, 李涛. 基于多阶段攻防信号博弈的最优主动防御[J]. 电子学报, 2017, 45(2): 431-439.
ZHANG H W, LI T. Optimal active defense based on multi-stage attack-defense signaling game[J]. Acta Electronica Sinica, 2017, 45(2): 431-439.

[18] HUNG S M, GIVIGI S N. A Q-learning approach to flocking with UAVs in a stochastic environment[J]. IEEE Transactions on Cybernetics, 2016, 47(1): 186-197.

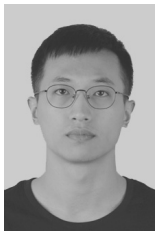
[19] SZEPESVARI C, LITTMAN M. A unified analysis of value-function-based reinforcement-learning algorithms[J]. Neural Computation, 1999, 11(8): 2017-2059.

[20] GORDON L, LOEB M, LUCYSHYN W, et al. 2015 CSI/FBI computer crime and security survey[C]//The 2014 Computer Security Institute. 2015: 48-64.

[21] 王震, 袁勇, 安波, 等. 安全博弈论研究综述[J]. 指挥与控制学报, 2015, 1(2): 121-149.
WANG Z, YUAN Y, AN B, et al. An overview of security games[J]. Journal of Command and Control, 2015, 1(2): 121-149.

[作者简介]

张红旗（1962−），男，河北遵化人，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、风险评估、等级保护和信息安全管理等。



杨峻楠（1993−），男，河北藁城人，信息工程大学硕士生，主要研究方向为网络信息安全、博弈论和强化学习等。

张传富（1973−），男，山东莱芜人，博士后，信息工程大学副教授，主要研究方向为计算机建模与仿真技术等。