基于 Q-learning 机制的网络动态防御研究*

张书钦[1,2], 李凯江[2], 李红[1], 石志强[1]

[1](中国科学院 信息工程研究所, 北京 100093)

[2](中原工学院 计算机学院, 河南 郑州 450007)

通讯作者: 张书钦, E-mail: zhangsq@zut.edu.cn

摘 要: 在网络安全防御中,考虑到部分服务和功能的修复代价比较大.网络防御的决策并不是盲目的进行漏洞修复和防火墙的重设置.本文通过对网络主机的资产重要性以及脆弱性的影响进行建模生成属性攻击图,结合攻击者的攻击动作决策和 Q-learning 机制,提出针对不同安全风险的动态防御策略.该方案通过 Q-learning 算法的收敛状态对不同安全风险等级、安全防护边界进行训练,可以根据防护目标的安全风险,及当前攻击的行为 Q 函数值选取最佳的防护策略. 最后,通过实验对本文的方法进行了有效性验证.

关键词: 网络安全; 攻击防御; Q 学习;攻击图

中图法分类号: TP393.09

Dynamic Defense Based on Q-learning

ZHANG Shuqin[1,2], LI Kaijiang[2], LI Hong[1], SHI Zhiqiang[1]

[1](School of Computer Science and Technology, Nantong University, Nantong 226019, China)

[2](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: In the network security defense, considering repair cost of some service and function relatively large.Network defense decision-making is not simply vulnerabilities repair and firewall resetting.This paper models network host asset importance and vulnerability impact to generate the attribute attack graph, combining with the attacker's action decision-making and Q-learning mechanism, and put forward the dynamic defense strategy for different security risks.The scheme through the convergence of the Q-learning algorithm for different security risk levels, security protection boundary training, The optimal defense policy can be be selected according to security risk of the protection goal and the Q function value of the current attack behavior. Finally, the proposed scheme in the paper is validated by experiments.

Key words: network security; attack defense; Q-learning; attack graph

随着网络信息技术的不断发展,计算机网络应用越来越广泛,由网络系统引发的网络安全问题也越来越多.网络系统作为攻击目标,其安全防御模型也有很多.目前研究安全防御的基础技术主要有 Petri Net、故障树、攻击树、特权图和攻击图等技术.攻击图概念最早由 Phillips 和 Swiler 等人提出[1].到目前为止,攻击图技术已经广泛应用于信息安全领域,特别是 Ammann 提出便于自动化分析的属性攻击图之后,攻击图成为网络脆弱性分析模型中最为流行的一类技术[2].席荣荣等人提出了基于环境属性的网络威胁态势强化的评估方法[4],该方法主要是安全事件的发生可能性和损失的量化进行建模的评估方法,但是在该方法中,无论是资产的分类还是安全事件数据库都是主观定义的,在应用上受到了很大的限制.戚湧等人通过改进漏洞的评分标准,在攻防图的基础

* 基金项目: 国家重点研发计划(2016YFB0800202); 工信部重点科研项目(JCKY2016602B001); 国家自然科学基金项目(U1636120) ; 河南省高新攻关项目(172102210591);郑州市科技攻关项目(153PKJGG131).

Foundation item: National Key Research and Development Program of China (2016YFB0800202); Key Research Program of Chinese MIIT (JCKY2016602B001); National Natural Science Foundation of China (U1636120); Henan Province High-tech Research Project (172102210591); Zhengzhou Science & Technology Project (153PKJGG131).

上结合攻击成功率和主机信息资产两个因素实现了网络安全防御策略的生成[5].但是该方法的防御策略是固定的,当针对具有多个不同主机资产评价标准的网络环境时,该策略的局限性就显露出来了.高妮等人通过漏洞利用的成功率和攻击成功建立了概率攻击图,该模型主要通过构建防护成本和攻击收益提出了基于粒子群的最有安全防护策略[6].但是这种基于粒子群的安全防护策略本身就建立在专家经验数据库上,并且针对不同的应用领域需要不同的领域知识经济指标,这使得防护策略的应用复杂性加大.黄亮等人利用基于历史攻击偏好的方法和熵权法对 DDoS 进行攻击和防护两个方面进行评估属性的计算,提出了防护措施遴选模型[7].该决策模型虽然排除了评估过程主观性的影响,但是算法模型本身针对性太强,只能够对 DDos 攻击提供防御决策.

近些年攻击图技术和智能算法的研究的结合得到了学者的尝试和关注,刘渊等人利用攻击图的攻击动作构建了带权的防御策略集合,利用二进制粒子群算法 BPSO 获取攻击图的最小关键策略集,引入通过最小代价的方式阻止网络恶意攻击[3].戚湧等人构建了一种基于马尔科夫链的攻击图模型[11].本文借助攻击图中的动态规划问题的增强学习机制建立模型.Neri J R F 等人和 Araghi S 等人已经利用 *Q*-learnging 机制解决了 Robocup 2D 类模拟足球机器人和网格系统中智能体的决策问题[10].本文通过构建基于马尔科夫决策模型的攻击图,对整个攻击图的状态进行形式化定义.考虑到攻击图生成过程需要对网络进行评估来选取攻击目标,文中结合目标网络的资产重要性和脆弱性两个因素进行攻击目标的选取.在攻击图的基础上进行 *Q*-learning 机制体系的构建,通过设置安全距离进行防御安全等级的实现.虽然智能体在 *Q*-learning 机制体系的构建中场景的学习比较耗费时间,但是后期决策不需要任何代价,仅通过当前节点的安全风险状态就可以给出当前安全等级下的防御决策建议.

本文第 1 节对文中使用的基本技术和算法进行描述,第 2 节构建了基于攻击图的马尔科夫链模型,利用攻击图对动态防御系统中防御目标间的攻击依赖关系进行建模,直观、方便地表达攻击者的攻击过程,并利用 *Q*-learnging 机制来表达防御动态性所导致的攻击不确定性(成功率).第 3 节结合攻击图和 *Q*-learnging 机制的特点,对 Q-learning 算法进行改进,通过算法的改进可生成针对不同安全等级实现不同的防护区域内任意状态节点的最佳防护策略选取.第 4 节通过实验对攻击图和 Q-learning 机制的结合进行应用性的验证,任意选取安全距离为 2 对攻击图环境进行训练学习,并对学习后的结果进行分析,得出该防御区域内的节点最佳选取策略.第 5 节对本文做了总结分析,并且阐述了本文的不足和下一步研究工作.

1 基础概念及描述

本文中使用的方法主要是将攻击图技术和增强学习中的 Q-learning 机制进行结合研究的,下面对攻击图技术和 Q-learning 做简要介绍.

1.1 攻击图技术

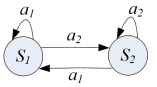
攻击图技术主要是根据网络的配置信息和脆弱性信息进行逻辑分析,生成攻击的逻辑的序列.如果对所有的序列进行相关性的关联,最终这些逻辑依赖序列形成一张有向图.这张有向图就是攻击图.目前对于攻击图的存储特性以及其表示形式主要分为状态攻击图和属性攻击图两种.

1.2 *Q*-learning算法思想

Q-learning 是强化学习主要算法之一,是模型无关的学习算法.*Q*-learning 假设是把智能体和环境的交互看作为一个马尔可夫(Markov)决策过程(MDP),即智能体当前的状态和下一步选择的动作,决定一个状态转移概率分布、下一个状态、并得到一个即时回报值.*Q*-learning 的目标是通过对客观世界采样,寻找一个策略可以最大化将来获得的报酬.

如图 1 所示,假设现在智能体处于 s_1 状态(只有采取指定次数 a_1 行为才能达到目标状态 G),而且智能体以前并没有尝试过 a_1 和 a_2 动作的同时采用,所以现在智能体有两种选择分别是 a_1 和 a_2.由于之前没有被惩罚过,所以智能体选择 a_2,然后智能体现在的状态变成了 s_2,最后智能体又选了继续 a_2 动作,接着智能体还是选择看 a_2,最后通过对环境的检测,发现智能体在没有得到固定要求的收益目标就一直采取动作 a_2 了,然后给出了反馈信

息,智能体得到反馈信息后,学习到"没得到一定的收益积累就采取动作 a_2"这种行为为负面行为.Q-learning 学习过程包含的两个重要过程分别是 Q-learning 决策和 Q-learning 更新.



(*a*)状态迁移

	a_1	a_2
s_1	1	-2
s_2	1	-2

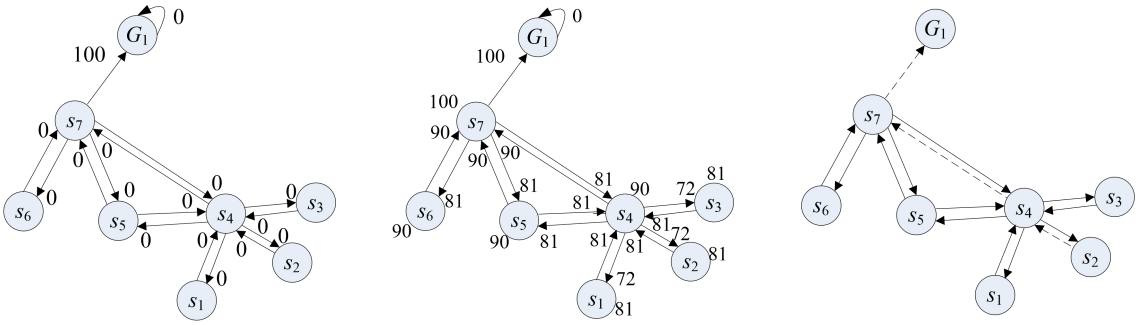
(*b*)Q 表

Fig.1 Illustration of Q-learning

图 1 Q-learning 算法示意

1.3 基于攻击图的Q-learning机制示例

为了更进一步对攻击图环境下的 Q-learning 机制进行应用,本节做一个简单的举例说明.图 2 示例了攻击图环境下的 Q-learning 机制的运算过程,图中每一个节点代表网络所处于的安全状态,有向连接代表在当前状态可供选择的行为.如果有向的连接方向是指向攻击目标则为攻击行为,相反则为防御行为.图 2(a)表示攻击图环境中采取动作时获取的立即奖罚函数值.这里只对到达攻击目标的直接行的立即奖罚值设为+100,其他的行为奖罚值为 0.这样根据贝尔曼方程中的 Q 函数的运算中会使整个智能体的抉择向攻击目标的方向做最佳的决策.这里设折扣因子 γ=0.9,经过强化学习算法收敛后 $V^*(s)$和$\pi^*(s)$的值为图 2(b).这样看来,在折扣因子的作用下 Q-learning 机制使得对于立即奖罚值为 0 的节点也具有了一定的远观性.智能体在进行动作选择时如果不考虑 ε *-greedy* 策略,从初始节点集合中通过状态-动作值函数中按照最大的值进行攻击行为的选取,当到达收敛状态 G_1 时这样的决策过程也就是其中的一个最优策略.图 2(c)描述的就是当初始状态为 s_2 时,智能体根据图 b 中的 Q 函数选取最大值的原则的一次最优策略所对应的攻击路径,其攻击路径为: $s_2 \rightarrow s_4 \rightarrow s_7 \rightarrow G_1$.



(a)初始的立即奖罚值 (b)Q 函数值和值函数值 (c)最优攻击路径

Fig. 2 Example of Q-learning mechanism in attack graph environment

图 2 攻击图环境下 Q-learning 机制应用示例

2 基于攻击图的马尔科夫链模型

防御目标是攻击者试图侵入、修改、破坏的防御实体.这些实体可以是软件、通信链路等,通过配置参数、运行状态等各种信息进行描述.利用攻击图技术对动态防御系统中防御目标的攻击依赖关系进行建模,直观、方便地表达攻击者为达到其攻击目标需要实施有步骤的攻击过程,并利用 Q-learnging 机制引擎来表达移动目标的动态性所导致的攻击不确定性(成功率).

2.1 攻击图基础模型构建

攻击者的每一步攻击都可以用网络中安全属性的迁移来进行描述.一次成功的安全攻击事件中,攻击者需要从先决条件开始进行网络攻击,并利用当前网络中的脆弱性依赖关系进而逐步达到攻击目标.攻击者可以通过先决条件根据脆弱性依赖关系到达攻击目标,在这个过程中的所有攻击路径生成的拓扑图称为攻击图.对于某一个特定的攻击目标,攻击者攻击过程中的攻击行为序列称为到达该攻击目标的攻击路径.为方便 Q-learning

机制的应用,本文进行了如下定义.

定义 1:网络安全状态主要有当前网络中各个主机的状态(包括各个权限)、连接关系等信息组成.*Node* 是表示网络安全状态的原子节点,用四元组(*nodeID*, *stateString*,*vValue*)描述,其中 *nodeID* 表示状态节点序号,*stateString* 表示当前安全节点的描述,如 *vulExists*(192.168.1.23,CVE-2016-0833,general,anyExploit,Android allows users to cause a denial of servic)表示当前节点 IP 为 192.168.1.23 的安卓设备存在漏洞 CVE-2016-0833,攻击者不需要获取内网访问权限即可进行远程攻击,*vValue* 表示当前网络节点的评估时的值函数值(*Q*-learning 机制中的值函数).

定义 2:*Trans* 表示网络安全状态转移关系,主要描述从初始节点到目标节点的变迁的集合.用五元组(*src*, *dst*, *bprop*, *fprop*, *bv*, *fv*)描述,其中 *src* 表示当前网络安全状态转换的源节点,*dst* 表示网络安全状态转换的目的节点,一对(*src*, *dst*)分别代表的一次正向的攻击和反向的防御.*fprop* 表示正向的攻击的概率,*bprop* 表示反向的防御的概率.*bv* 表示当前转换关系为防御时的 *Q* 函数值(*Q*-learning 机制中的 *Q* 函数),*fv* 表示当前变迁为攻击时的 *Q* 函数值.

定义 3: *Action* 表示对于当前网络状态转移中的节点行为的集合,节点有攻击、防御和维持三种行为.

定义 4:攻击图定义为一个网络的安全状态的变迁描述,记为 *G*=(*Node*, *Trans*, *Action*, $Node_{init}$, $Node_{goal}$).其中, $Node_{init}$ 表示当前目标网络中的扫描后获得的先决条件的集合,$Node_{goal}$ 表示网络中的维护的核心目标(即网络的攻击路径的目标节点集合).$Node_{init}$ 集合中的节点只有两种行为,分别为攻击和维持.$Node_{goal}$ 集合中的节点同样也只有一种行为,为维持当前状态(因为已经出于攻击模型中的收敛的状态).

定义 5:攻击路径定义为从初始节点到目标节点的攻击行为序列集合,记为 *P*=(*Node*, *Trans*, *Action*, $Node_{init}$, $Node_{goal}$).攻击路径为攻击图的子图,主要是针对某个攻击节点进行攻击状态转移序列的集合描述.

2.2 基于*Q*-learning机制的攻击图模型的优化

Q-learning 机制是解决动态规划问题的一种算法思想,攻击图的拓扑结构就是具有若干个初始节点和终端节点的有向图的描述.对于具有特定目标的攻击,可以利用 *Q*-learning 机制攻击路径进行优化改进.本文采用马尔科夫决策模型构建基础攻击图模型,进而通过 *Q*-learnging 机制对某个防御节点提供动态防御策略.

基于攻击图的 MDP 模型主要包括状态、攻击目标、行为,折扣因子,收益五个部分进行定义[12][13],做如下定义.

定义 6:基于攻击图的 MDP 模型可以形式化定义为{*State*, *Goal*, *Action*, γ, *refund*},其中各元素含义如下.

(1) *State* 表示网络系统中攻击图脆弱性状态的集合,用攻击图中的 *Node* 进行描述;

(2) *Goal* 表示在攻击图环境中的目标状态,也就是 *Q*-learning 机制中的收敛状态,用攻击图中的 $Node_{goal}$ 进行描述;

(3) *Action* 表示智能体可以采取的动作集合,用攻击图中的 *Action* 进行描述,唯一不同的是根据安全距离的不同(安全距离见 3.2 节)定义不同的收敛状态,收敛状态为最终的状态,在没有检测到外部攻击时只采取停留当前状态的行为.

(4) $\gamma \in [0,1)$: 折扣因子表示随着状态节点时序的推移采用的回报率.

(5) *refund* 回报函数,主要根据脆弱性因素以及攻击和防护措施进行构建回报值.

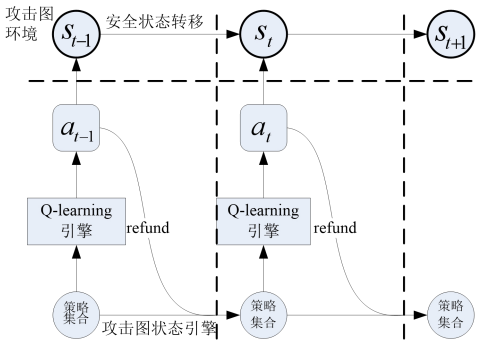


Fig.3 *Q*-learning architeture

图 3 *Q*-learning 学习过程

在基于 *Q*-learning 机制的攻击图决策中,智能体通过多次攻击模拟防御学习对网络安全状态的节点进行值函数计算.针对成对的攻击和防御节点对通过攻击和防御行为的对 Q 函数进行计算.策略学习的一次场景过程描述:从 $Node_{init}$ 的某个初始状态开始,存在到 $Node_{goal}$ 的一组攻击防御序列 $Node_{init}$, $Node_i$, …, $Node_{goal}$ 这样一组网络安全状态转移序列称为一次策略学习过程.

2.3 基于*Q*-learning机制的动态防御策略算法

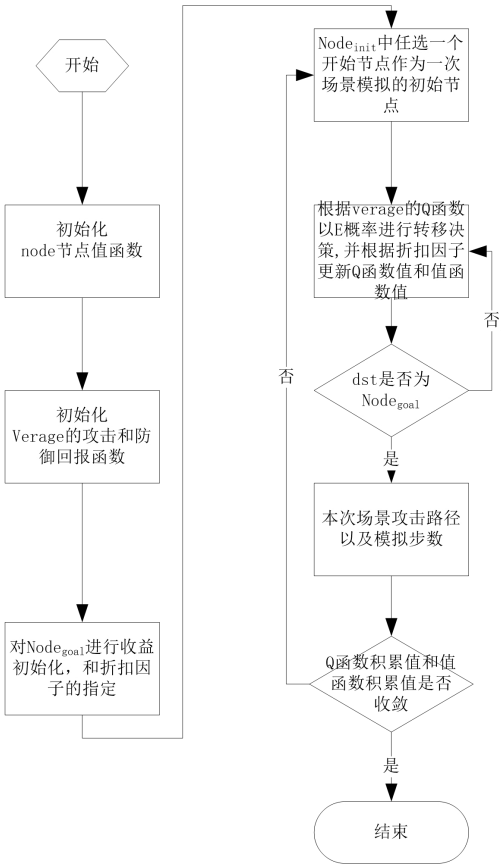


Fig.4 Dynamic defense policy algorithm based on *Q*-learning

图 4 *Q*-learning 机制的动态防御策略算法

本文采用对 *Q*-learning 的机制对攻击图的模拟攻击防御来进行策略学习,最终针对某个状态节点给出攻击或者防御的最优决策行为.相对于广度和深度优先的攻击图遍历计算评估值后再进行防御的策略推荐,本算法在用机器学习中 *Q*-learning 机制进行马尔科夫决策模型的求解,学习结束后,可在线性时间内给出最佳的攻击和防御的决策.

算法中,通过收敛状态 $Node_{goal}$ 进行初始化收益、对折扣因子进行取值和对整个攻击图进行初始化.然后在攻击防御的模拟场景中进行策略的学习,对 *Trans* 中的攻击、防御的行为回报 *Q* 函数和攻击图中的 *Node* 中的值函数进行更新.直至算法中的 *Q* 函数积累值和值函数的积累值收敛后结束.*Q*-learning 机制的动态防御策略算法描述如图 4.

3 基于资产分析的防护目标选取与安全距离建模

本节从资产分析的角度对网络进行建模,选取出维护网络区域的防护目标.最后,结合攻击图和 *Q*-learning 特点,对 *Q*-learning 算法进行改进.算法中将不同的安全风险对应不同的收敛状态进行攻击环境的学习训练,最终针对设定的安全风险可生成防护节点的最佳防御策略.

3.1 基于资产分析的防护目标选取

攻击图中的基于资产的防护目标的选择核心问题在于在网络环境中的主机的资产 *P* 和修复成本 C 以及漏洞的影响度 S 一定的条件下,选择最优的目标防护主机.假设维护网络空间共有 N 个主机(M_1, M_2, …, M_n),根据 IP 或者 MAC 可做唯一标识.其中主机 M_i 的漏洞数量为 k_i.本文中综合考虑网络主机的资产 pc_i 以及脆弱性本身的代价 cv_k 进行网络中主机进行重要性的量化评估 V_k.根据当前主机的资产 P 和修复成本 C(默认为 P>C)的差值做归一化处理归一化范围为 0–π/3,如式(1)所示:

$$pc_i = \frac{\frac{\pi}{3}[(P_i - V_i) - \min(P - V)]}{\max(P - V) - \min(P - V)} \tag{1}$$

脆弱性的影响度 cv_k 表示本系统中漏洞 *k* 的影响参数,计算方式如式(2).

$$cv_k = \ln(1 + \frac{\text{cvs}}{10}) \tag{2}$$

主机中资产的重要性比脆弱性的影响度要高,比如对于相同的脆弱性,位于普通客户端主机的重要性没有维护核心服务器主机的重要性高.综合考虑我们可经验性的设置 $V_k = \tan(pc_i)cv_k$,将防护目标选取为主机重要性值最大的一个或者若干个主机.

3.2 基于攻击图技术的网络防护安全距离体系

在网络安全系统中,安全和威胁是相对的.安全性能设置太高会造成一些服务无法正常使用,特别是涉及到工业控制或者基础设置的传感器的网络,但是设置太低又容易引起网络攻击者的攻击.特别是在动态的防御系统中需要当前的攻击状态进行监控,在不影响系统功能和安全底线的攻击行为可以只进行监控,当达到系统的安全防御范围内再进行防御措施的采取.这样就可以达到系统性能和安全性的兼顾.

本文通过对主机的资产重要性对防御区域进行构建基于一定安全距离和防御等级区域的模型,一般来说,用安全距离可以区分网络系统中根据业务要求和安全要求区分的不同安全区域,不同区域内的防御目标对应用任务会有不同的影响,如 SCADA 系统中,RTU、PLC 设备一旦被攻击甚至会导致灾难性的后果.再如企业服务器中的软件,tomcat、IIS、mysql 和服务端程序一旦被入侵则会引起商业机密的泄漏.因此我们构建了基于攻击图技术的网络防护安全距离体系.该体系以防御目标为中心,可以设置根据安全距离设置安全的防护级别.具体运行方式如图 5 所示.

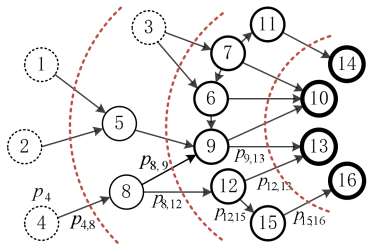


Fig.5 Illustration of security distance for defense based on attack graph

图 5 基于攻击图的网络防御安全距离示意

这里假定对于网络监控范围内以防护主机为目标进行攻击图进行构建,可以根据检测目标网络中的目标节点{10, 13, 14, 16}根据不同的安全距离定义防御等级{dl_i},防御等级越高意味着该节点的动态性越强,则该目标被成功入侵的概率越低.目标节点日常运行在 dl_i,若发现被侵入时,就相应提升其前向节点的防御等级.对于 $i \in Node$,当且仅当 i 的所有直接前驱节点在时刻 t 已被成功入侵,则 i 在时刻 t+1 才有非零的攻击成功概率.以上定义的攻击图如图 5 所例示(仅在部分边标出了攻击概率),其中 $Node_{init}$ ={1, 2, 3, 4}, $Node_{goal}$ ={10, 13, 14, 16}.攻击者若要侵入目标 5,必须首先侵入目标 1 和 2；若要若要侵入目标节点 13,必须首先侵入目标 9 和 12.在对网络的监控中,若检测到攻击者已经到达 8 号节点的状态,如果当前的安全距离设置为 3,那么则进行防御(防御的措施通常为漏洞修复,防火墙设置以及服务的关闭)；如果检测到当前的安全距离设置小于 3,那么则认为目前网络状态仍为安全的.目前看来安全等级越高越好,但是需要注意的是较高的防御等级也会增加防御代价,影响正常业务的运行.

3.3 基于*Q*-learning机制的动态防御策略算法改进

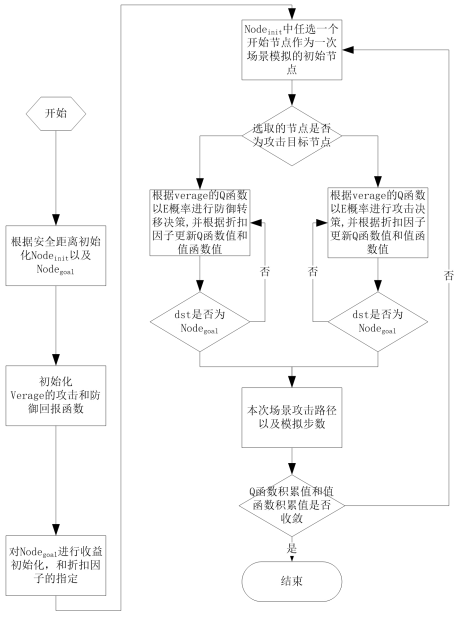


Fig. 6 Dynamic defense policy based on *Q*-learning

图 6 *Q*-learning 机制的动态防御策略算法

对于 *Q*-learng 机制的攻击图中节点的策略学习过程本文已做了详细的算法说明.考虑到网络安全中防御级别的设定,我们构建了基于攻击图的安全距离的体系.该体系可根据不同的安全级别进行动态的调整防御的规则.算法中我们需要对网络的收敛状态进行重新的定义,在攻击和防御的场景中,根据制定的网络安全级别相

对应的节点设置收敛状态(此前攻击目标节点是收敛状态).在学习过成功 Q 函数和值函数的更新规则不变,但是需要对 $Node_{init}$ 进行重新的设置,因为模拟攻击时由于初始节点包含攻击目标节点会引起攻击目标节点到收敛状态之间的节点的 Q 函数和值函数得不到更新学习.所以需要将攻击目标也加入到 $Node_{init}$ 集合中.值得注意的是,在攻击目标节点到收敛状态之间的节点的 Q 函数和值函数更新学习过程中,为了加快 Q 函数积累值和值函数积累值的收敛,攻击和防御的行为的选择概率是相反的.具体算法描述如图 6.

4 实验分析

4.1 网络攻击防御环境

为了验证方法的有效性,构造的网络环境如图 7 所示,其中办公网虚拟了两台主机,一台 linux 主机 ip 为 192.168.1.94,以及一台 win10 主机 ip 为 192.168.1.94.企业隔离服务器由两台服务器组成,分别分配外网 ip 为 222.22.94.97、222.22.94.98.主要开启的服务有 dns 、cifs、smb、rpc、www 和 smtp 服务.办公网络和服务器网络通过路由器和外界进行网络转发,路由器网关分别设置了两个子网映射一个为内网的办公区的局域网为 192.168.1.1,另一个为公网 IP 段的网关为 222.22.94.126.具体的主机信息如表 1.

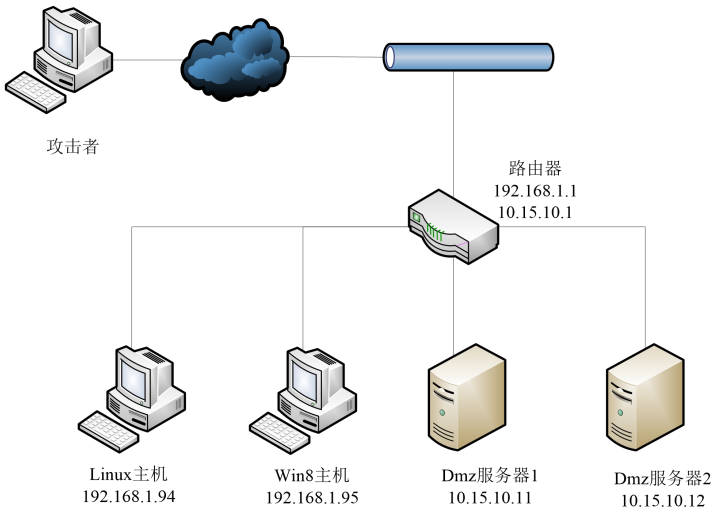


Fig. 7 Experiment network settings

图 7 网络攻击防御环境

Table 1 Host Vulnerablities in Experiment

表 1 主机详细脆弱性信息

主机	服务	cve:cvss	网关
192.168.1.94	Mdns:udp:5353		192.168.1.1
192.168.1.95	Mdns:udp:5353	CVE-2016-3266:10	192.168.1.1
222.22.94.97	Dns:udp:53 |cifs:tcp:443| rpc-nfs:2049|cifs:tcp:445|x11:tcp:6000|smb:tcp:139| rpc-portmapper:tcp:111	CVE-2014-2830:10| CVE-2015-6421:7.8| CVE-2014-0069:6.2	222.22.94.126
222.22.94.98	www:tcp:80|irc:tcp:6667|vnc:tcp:5900|smtp:tcp:25|dns:udp:53|rpc-portmapper:tcp:111|	CVE-2016-0468|3.5	222.22.94.126

4.2 实验结果分析

利用攻击图生成工具对网络环境进行脆弱性的关联分析,该环境中生成的攻击图的节点共有 106 个逻辑推理过程状态节点,逻辑行为关系双向行为共 145 个,考虑到节点比较多,这里不做拓扑的展示.在状态节点中,攻击目标集合为:{1, 4},初始节点状态集合为{3, 10, 11, 12, 13, 14, 15, 16, 19, 25, 28, 29, 30, 31, 34, 42, 43, 44, 45, 48, 50, 54, 55, 56, 57, 61, 62, 63, 66, 70, 71, 72, 75, 85, 88, 89, 97, 98, 99, 100, 101, 102, 104, 106}.强化学习

的 *Q*-learning 机制中的重要参数进行了初始设置,设置到达与安全距离对应的边界状态节点的立即奖罚值为100、折扣因子设置为 0.9.采用 ε =0.7 的贪婪策略进行策略选取.

在实验中对该攻击图环境中安全距离与相关的节点(这里列举安全距离 0-4 的相关节点信息)进行计算,详细对应结果如表 2.

Table 2 the relevant node with security distance 0-4

表 2 安全距离 0-4 的节点

安全距离	安全状态节点
0	{1, 4}
1	{103, 2, 5, 105}
2	{102, 3, 101, 6, 106, 104, 15}
3	{51, 35, 20, 67, 7, 76, 58}
4	{68, 32, 71, 64, 36, 8, 77, 72, 46, 73, 14, 44, 15, 45, 17, 16, 21, 52, 59, 57, 63, 62, 31, 30}

当设置安全距离为 2 时,初始节点集合为:{3, 10, 11, 12, 13, 14, 15, 16, 19, 25, 28, 29, 30, 31, 34, 42, 43, 44, 45, 48, 50, 54, 55, 56, 57, 61, 62, 63, 66, 70, 71, 72, 75, 85, 88, 89, 97, 98, 99, 100, 101, 102, 104, 106, 1, 4},收敛状态集合为{102, 3, 101, 6, 106, 104, 15},对于即在初始节点集合又在收敛状态节点集合的节点,表示该节点已经处于"安全防御"区域.如检测到攻击,则立即进行防御.

攻击防御场景 194-205 次模拟过程如表 3 所示,其中 sense 表示训练过程中的场景数,一次场景表示从初始节点中的任意一个节点到达收敛状态的攻击过程,如第 202 次场景分别表示智能体的攻击路径为 19→18→17→7→6.

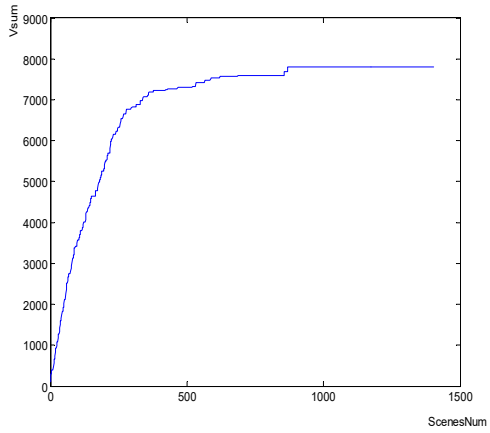
Table 3 part of simulated attack defense scene

表 3 部分攻击防御场景

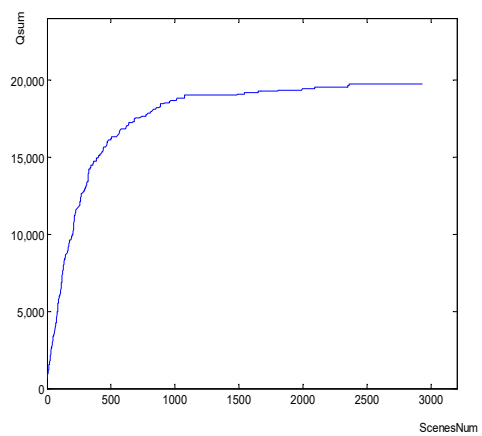
sense	step	qvalueSum	rewardSum	attackPath
194	1480	5911	11333	[4, 5, 102]
195	1489	5984	11399	[25, 39, 38, 37, 38, 37, 36, 35, 6]
196	1504	5984	11458	[63, 90, 79, 78, 72, 78, 79, 78, 77, 78, 77, 76, 77, 76, 6]
197	1519	6032	11458	[85, 84, 83, 84, 83, 82, 81, 80, 79, 80, 79, 78, 77, 76, 6]
198	1522	6032	11558	[71, 76, 15]
199	1529	6032	11558	[12, 9, 8, 9, 8, 7, 6]
200	1536	6032	11617	[28, 27, 26, 22, 21, 20, 6]
201	1547	6071	11617	[88, 87, 86, 82, 81, 80, 79, 78, 77, 76, 6]
202	1552	6152	11617	[19, 18, 17, 7, 6]
203	1565	6152	11660	[85, 84, 83, 82, 81, 80, 79, 80, 79, 78, 77, 76, 6]
204	1568	6152	11660	[1, 2, 3]
205	1577	6152	11660	[50, 49, 46, 49, 46, 47, 46, 35, 6]

对整个训练的值函数值积累 Vsum 和 *Q* 函数积累 Qsum 进行的分析图如图 8,当训练次数 ScenesNum 达到 750 次,智能体决策次数 AgentNum 达到 5708 次时,值函数积累值 Vsum 趋于收敛,收敛值约为 7794；当训练次数 ScenesNum 达到 2010 次,智能体决策次数 AgentNum 达到 15340 次时,值函数积累值 Vsum 趋于收敛,收敛值约为 19723.

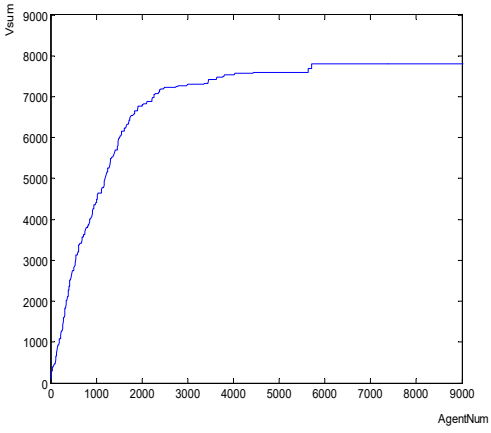
当安全距离为 2 时,安全防御区域内的节点状态集合为{1, 4, 102, 3, 101, 6, 106, 104, 15, 102, 3, 101, 6, 106, 104, 15}.即当检测到攻击者达到以下几个状态节点时,则利用经过训练后的该防御节点对应的行为选取 *Q* 值最大防御行为的来对网络系统进行加固处理.



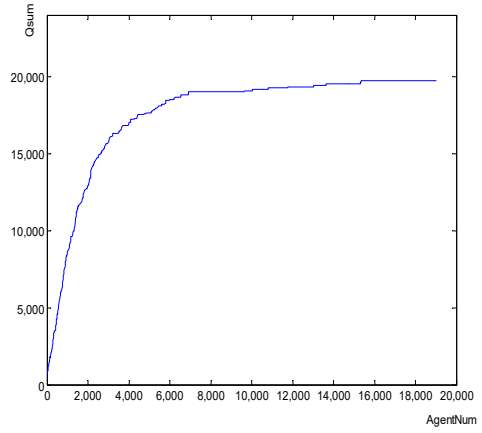
(a)训练场景中值函数积累值的收敛过程



(b) 训练场景中 Q 函数值积累值的收敛过程



(c)智能体决策过程中值函数积累值收敛过程



(d) 智能体决策过程中 Q 函数值积累值的收敛过程

Fig. 8 Value function accumulation and Q function accumulation in Experiments

图 8 实验中值函数积累和 Q 函数积累

5 结束语

在传统的攻击图的使用过程中,往往只是通过这样的网络拓扑图达到评估和网络安全监控的作用.并且攻击图构建过程中需要考虑环的问题,以及采用深度或者广度优先的算法及进行攻击图的后期使用.本文提出了一种利用 Q-learning 机制的基于安全距离的动态防御策略的方法.该方法可以通过不同的防护目标重要性设置不同的网络安全等级.并且通过对攻击图中的节点给出其当前行为的决策参数(Q 函数)的方式进行动态的防御并且通过实验对该方法进行了有效性的验证.但是该算法还存在一些局限性,例如,决策中的方案是针对目前已经暴漏出来的脆弱性进行攻击图生成,其攻击图的好坏对脆弱性扫描的工具依赖比较大.另外,该方法的学习过程中冗余数据量在数据库中比较大,采取合适的算法进行改进是下一步的工作.

References:

[1] Phillips C, Swiler L P. A graph-based system for network-vulnerability analysis[J]. Proceedings of the Workshop on New Security Paradigms, 1998:71-79.

[2] Ammann P, Wijesekera D, Kaushik S. Scalable, graph-based network vulnerability analysis[C]//Proceedings of the 9th ACM Conference on Computer and Communications Security. ACM, 2002: 217-224.

[3] 刘渊, 李群, 王晓锋,等. 基于攻击图和改进粒子群算法的网络防御策略[J]. 计算机工程与应用, 2016, 52(8):120-124.
[4] 席荣荣, 云晓春, 张永铮. 基于环境属性的网络威胁态势量化评估方法[J]. 软件学报, 2015, 26(7):1638-1649.
[5] 戚湧, 莫璇, 李千目. 一种基于攻防图的网络安全防御策略生成方法[J]. 计算机科学, 2016, 43(10):130-134.
[6] 高妮,高岭,贺毅岳,雷艳婷,高全力.基于贝叶斯攻击图的动态安全风险评估模型, 四川大学学报(工程科学版),2016(1):111-118.
[7] 黄亮, 冯登国, 连一峰,等. 一种基于多属性决策的 DDoS 防护措施遴选方法[J]. 软件学报, 2015, 26(7):1742-1756.
[8] 李志, 单洪, 马春来,等. 基于攻防图的网络主动防御策略选取研究[J]. 计算机应用研究, 2015, 32(12):3729-3734.
[9] Watkins C J C H, Dayan P. Q-learning[C]// Machine Learning. 1992:279--292.
[10] Neri J R F, Zatelli M R, Santos C H F D, et al. A Proposal of Q learning to Control the Attack of a 2D Robot Soccer Simulation Team[C]// Brazilian Robotics Symposium and Latin American Robotics Symposium. 2012:174-178.
[11] Araghi S, Khosravi A, Johnstone M, et al. A novel modular Q-learning architecture to improve performance under incomplete learning in a grid soccer game[J]. Engineering Applications of Artificial Intelligence, 2013, 26(9):2164-2171.
[12] 戚湧, 刘敏, 李千目. 基于扩展马尔科夫链的攻击图模型[J]. 计算机工程与设计, 2014(12):4131-4135.
[13] Sheyner O, Haines J, Jha S, et al. Automated Generation and Analysis of Attack Graphs[C]// Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on. IEEE, 2002:273-284.
[14] Roijers D M, Vamplew P, Whiteson S, et al. A Survey of Multi-Objective Sequential Decision-Making[J]. Journal of Artificial Intelligence Research, 2014, 48(1):67-113.