计算机科学
COMPUTER SCIENCE

自动化红队测试中强化学习策略的实现与验证

陈宇飞, 李赛飞, 张丽杰, 赵越

引用本文

陈宇飞, 李赛飞, 张丽杰, 赵越. 自动化红队测试中强化学习策略的实现与验证[J]. 计算机科学, 2023, 50(11A): 230200162-6.

CHEN Yufei, LI Saifei, ZHANG Lijie, ZHAO Yue. Implementation and Verification of Reinforcement Learning Strategy in Automated Red Teaming Testing [J]. Computer Science, 2023, 50(11A): 230200162-6.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230200162

自动化红队测试中强化学习策略的实现与验证

陈宇飞[1]　李赛飞[1]　张丽杰[2]　赵　越[3]
1 西南交通大学信息科学与技术学院　成都 611756
2 北方激光研究院有限公司信息技术中心　成都 610041
3 中国电子科技集团公司第三十研究所保密通信重点实验室　成都 610041
(cyfllab@163.com)

摘　要　红队测试是一种通过模拟真实黑客攻击行为来对网络系统进行安全测评的方法。然而,目前人工测试存在成本较高与适应性较差的问题。红队测试智能化与自动化是当前研究的热点问题,旨在降低红队测试的成本,提高网络安全测评的测试性能与测试效率。自动化攻击策略是自动化红队测试的核心,其作用是替代安全专家进行攻击技术的决策。文中将红队攻击技术映射到强化学习,从而将红队测试过程建模为马尔可夫决策模型,通过有限状态机模型实现了固定策略与强化学习策略;在真实网络环境中对不同的强化学习策略进行训练和测试,验证了强化学习策略的收敛性和可行性。实验结果表明,基于SARSA(λ)算法的强化学习策略优于其他强化学习策略,收敛速度最快;3种强化学习策略均能在测试实验中稳定完成测试目标,且性能远优于固定策略。

关键词:网络安全;红队;自动化攻击策略;渗透测试;强化学习
中图法分类号　TP393

Implementation and Verification of Reinforcement Learning Strategy in Automated Red Teaming Testing

CHEN Yufei[1],LI Saifei[1],ZHANG Lijie[2] and ZHAO Yue[3]
1 College of Information Science and Technology,Southwest Jiaotong University,Chengdu 611756,China
2 Norla Institute of Technical Physics,Chengdu 610041,China
3 Science and Technology on Communication Security Laboratory,Chengdu 610041,China

Abstract　Red teaming testing is a method to evaluate the security of network system by simulating real hacker attack behavior. However,manual test has the problems of high cost and poor adaptability at present. Red teaming testing intelligence and automation is currently a hot research topic,aiming at reducing the cost of red teaming testing and improving the test performance and efficiency of cybersecurity assessments. Automated attack strategy is the core of automated red teaming testing,it is designed to replace security experts in the attack technology decision-making process. In this paper,the red teaming attack technique is mapped to reinforcement learning,the red teaming testing process is modeled as a Markov decision process model,and the fixed strategy and reinforcement learning strategy are implemented through the finite state machine. Reinforcement learning strategy is trained and tested in the real network environment to verify the convergence and feasibility. Experimental results show that the SARSA(λ) algorithm is superior to other reinforcement learning algorithms and has the fastest convergence speed. The three reinforcement learning strategies can achieve the test objective stably in the test experiment,and the performance is much better than that of the fixed strategy.

Keywords　Cybersecurity,Red teaming,Automated attack strategy,Penetration testing,Reinforcement learning

1　引言

随着计算机网络的不断发展,安全事件的发生频率提高,网络安全已成为一个不可忽视的问题。目前解决这个问题的一种方法是通过渗透测试来评估系统的安全性能。渗透测试是指在获得授权的情况下,通过发现与利用系统的漏洞来获得系统的控制权,从而修复安全漏洞,提高系统安全性[1]。另一种方法是红队测试,即通过模拟黑客真实的攻击行为,构建可行的攻击策略,从而对被测系统进行安全测评,以达到评估整个被测系统的状态的目的。但是目前红队测试存在成本

基金项目:四川省科技计划项目(2021YJ0372);四川省重大科技专项项目(2019ZDZX0007,2021YFQ0056);保密通信重点实验室基金(61421030201022108)
This work was supported by the Sichuan Science & Technology Planning Project(2021YJ0372) and Sichuan Science & Technology Major Special Project(2019ZDZX0007,2021YFQ0056) and Science and Technology on Communication Security Laboratory Foundation(61421030201022108).
通信作者:李赛飞(lisaifei@swjtu.edu.cn)

较高、耗时较长的问题，测试效果也受限于测试人员的专业知识水平；另一方面，不同的被测系统的复杂性也给红队测试带来了挑战，测试人员需要根据不同的系统环境设计不同的攻击方案，如被测系统测试范围，测试可采用的战术与技术，测试目标等。以上问题增加了开展红队测试的难度[2]。

为了解决上述问题，国内外学者提出了自动化红队测试；与此同时，人工智能技术的高速发展也为自动化红队测试带来了新的方向——强化学习。强化学习需要代理与环境不断交互，根据目标状态获得反馈，从而获得学习经验，而红队测试也是根据测试目标状态进行决策，因此两者结合是可行的。强化学习使得自动化红队测试具有了自适应性，可以面对不同的测试环境智能地做出决策，构建最优的策略来达成测试目标。

近年来，强化学习与自动化测试的结合成为了研究热点。Gangupantulu 等[3]提出了将网络地形构建为攻击图的马尔可夫决策模型，并应用强化学习进行渗透测试；Hu 等[4]提出了一个自动渗透测试框架，目标是为给定的网络拓扑寻找最佳攻击路径；Pozdniakov 等[5]将安全审计过程建模为马尔可夫决策过程，优化了 Q-Learning 算法并将其应用于自动化渗透测试；文献[6-7]将渗透测试过程建模为更全面的部分可观测马尔可夫决策模型，但是带来了模型计算求解复杂度过高的问题；Li 等[8]提出了基于 Q-Learning 的最优攻击路径生成方法，根据网络拓扑与漏洞信息模拟攻击环境；Maeda 等[9]在 PowerShell Empire 的基础上应用了深度强化学习算法，从而实现了自动化后渗透。以上研究工作的对比如表 1 所列。

表 1 已有研究的比较

Table 1 Comparison of previous researches

文献	属性		
	可观测性	测试类型	实验环境
[3]	完全	漏洞利用	仿真
[4]	完全	漏洞利用	仿真
[5]	完全	漏洞利用	真实
[6-7]	部分	漏洞利用	仿真
[8]	完全	漏洞利用	仿真
[9]	完全	后渗透	仿真

综上所述，目前使用强化学习进行自动化红队测试的研究仍处于初步阶段，大多数研究通过漏洞信息进行攻击策略的规划；同时，受限于强化学习代理与环境的强交互性，实验环境还处于仿真阶段，没有与真实环境进行交互。本文的主要工作是将红队测试与强化学习相互映射，以攻击技术对应动作，攻击结果对应状态，将攻击场景建模为马尔可夫决策过程，通过有限状态机模型实现强化学习策略，在真实实验环境中完成对强化学习代理的训练与测试，从而验证强化学习策略的收敛性与可行性。

本文第 2 节介绍强化学习理论基础与自动化红队测试的模型结构；第 3 节设计并实现了两种自动化攻击策略，分别是固定策略与强化学习策略；第 4 节进行自动化红队测试实验，测试了两种自动化攻击策略在真实实验环境中的表现；最后总结全文。

2 理论基础与自动化红队测试模型

2.1 自动化对手仿真平台

MITRE ATT&CK[10]是一个基于真实世界高级持续性威胁(Advanced Persistent Threat，APT)攻击事件构建的对手战术和技术知识库。该知识库可作为红队测试、开发特定威胁模型和威胁情报分析等技术的基础。ATT&CK 总结归纳了大量 APT 攻击事件，将其划分为战术和技术两个维度，战术代表对手行动的战术目标，技术是实现战术目标所采用的具体方式。

CALDERA[11]是一个基于 MITRE ATT&CK 开发的自动化对手仿真平台，可以协助红队测试。CALDERA 将 ATT&CK 下的技术进行了具体实现，将其称为能力，其包含能力名、能力描述、所适用的操作系统、具体的命令实现等属性，可用于真实攻击。本文通过 CALDERA 构建对手，在真实环境中基于不同的攻击策略进行自动化红队测试。

2.2 马尔可夫决策模型

马尔可夫决策过程(Markov Decision Process，MDP)是强化学习的理论基础，其具有马尔可夫性，即系统下一个状态不仅和当前的状态有关，也和当前采取的动作有关。MDP 由五元组(S,A,R,T,γ)定义。本文将红队测试过程建模为马尔可夫决策过程，以 S 表示代理观察到的目标主机的状态；A 表示代理可选的攻击动作的集合；$R(s,a)$表示奖励函数，指代理在状态 s 下执行攻击动作 a 后获得的奖励；$T(s,a)$为状态转移函数，指代理在状态 s 下执行攻击动作 a 后可能的状态分布；γ 代表用于计算整个过程累计奖励收益的折扣率。

2.3 强化学习

与监督学习和无监督学习不同，强化学习不需要依赖大量的静态数据，而是通过与环境不断地进行交互，学习在一个连续决策问题中寻找最大累积奖励的策略[12]。图 1 给出了强化学习代理与环境的交互过程。

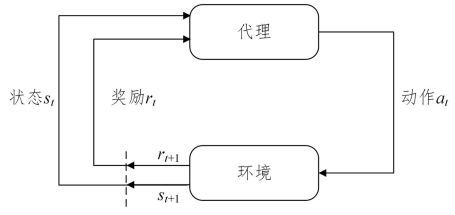


图 1 强化学习交互过程

Fig. 1 Reinforcement learning interactive process

在时刻 t，代理基于观察到的环境状态 $s_t \in S$ 与动作选择策略 $\pi(a_t|s_t)$选择并执行动作 $a_t \in A$；在下一个时刻 $t+1$，代理收到奖励 r_t 并观察到环境产生新的状态 $s_{t+1} \in S$。代理学习的目标是最大化未来的累积奖励，其奖励函数见式(1)，其中 $\gamma(0<\gamma<1)$为折扣率，表示代理对未来奖励的期望程度，γ 越大表示代理越倾向于未来奖励。

$$R_t=\sum_{k=0}^{\infty}(\gamma^k r_{t+k}) \tag{1}$$

本文一共构建了 3 种强化学习算法作为攻击策略，包括 Q-Learning，SARSA 以及 SARSA(λ)。

Q-Learning 是一种基于价值的离线策略强化学习算法[13]。Q-Learning 采用表格的方式来存储 Q 值。ε-greedy 是最常见的动作选择策略。每次迭代过程中，代理首先根据 ε-greedy 策略在状态 s_t 选择动作 a_t，执行动作 a_t 后转移到状态 s_{t+1}，同时代理获得奖励 r_t，在新状态 s_{t+1}，代理选取当前

状态下最大的 Q 值用于更新价值函数 $Q(s_t,a_t)$。更新公式如式(2)所示：

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha[r_t+\gamma\max_a Q(s_{t+1},a)-Q(s_t,a_t)] \tag{2}$$

其中，$\alpha(0<\alpha<1)$为学习率，$\gamma(0<\gamma<1)$为折扣率。

SARSA 是一种基于价值的在线策略强化学习算法[13]。SARSA 算法与 Q-Learning 算法在策略上是一致的，也采用 ε-greedy 策略，但是 SARSA 在更新 Q 表时与 Q-Learning 不同，SARSA 根据 ε-greedy 选择动作 a_t 并执行，转移到状态 s_{t+1} 后，再次根据 ε-greedy 策略选择动作 a_{t+1} 并执行，并使用 $Q(s_{t+1},a_{t+1})$ 值来更新价值函数 $Q(s_t,a_t)$。更新公式如式(3)所示：

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha[r_t+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \tag{3}$$

其中，$\alpha(0<\alpha<1)$为学习率，$\gamma(0<\gamma<1)$为折扣率。

SARSA(λ)是在 SARSA 的基础上进行了改进，它的学习效率更高[14]。与 Q-Learning 和 SARSA 不同，SARSA(λ)获取奖励后不仅会更新当前的价值函数，还会更新之前已经更新的价值函数。SARSA(λ)在时刻 $t+1$，对于所有的状态 s 与动作 a，价值函数 $Q_{t+1}(s,a)$ 的更新如式(4)所示：

$$Q_{t+1}(s,a)=Q_t(s,a)+\alpha\delta e(s,a) \tag{4}$$

其中，$\delta=r_t+\gamma Q_t(s_{t+1},a_{t+1})-Q_t(s_t,a_t)$，$\alpha$ 为学习率，$e(s,a)$ 为资格迹，其记录了到达当前状态前经历的所有状态和动作。对于状态 s_t 与任意动作 a，$e(s,a)$ 的计算方式如式(5)所示：

$$e(s_t,a)=\begin{cases}\gamma\lambda e(s_t,a)+1, & \text{if } a=a_t\\ \gamma\lambda e(s_t,a)*0, & \text{otherwise}\end{cases} \tag{5}$$

其中，$\gamma(0<\gamma<1)$为折扣率，$\lambda(0<\lambda<1)$为衰减率，如果 $\lambda=0$，SARSA(λ)就是 SARSA。

2.4 自动化红队测试模型结构

自动化红队测试模型主要由以下两个部分组成(见图2)。

(1)对手模型：根据 CALDERA 构建对手作为决策模型的输入。

(2)决策模型：根据攻击策略与真实环境进行交互，执行对手的能力。

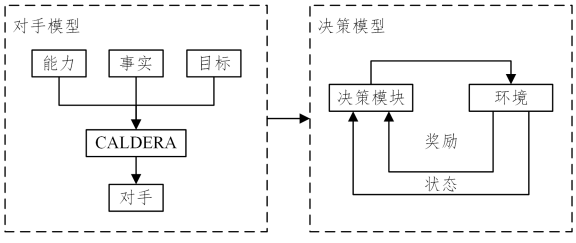


图 2 模型结构

Fig. 2 Model architecture

2.4.1 对手模型

对手由三元组 (A,F,g) 构成，其中 A 表示能力的集合，F 表示事实的集合，g 表示对手的目标。在 CALDERA 中根据三元组则可构建出对手，作为决策模型的输入。

能力 $a\in A$ 是一种特定的 ATT&CK 技术实现，可以通过代理在目标主机上进行执行。能力属性包括执行命令、对应的平台(如 Windows)、执行器(如 PowerShell)、包含的有效负载等。部分能力 $a_i\in A$ 拥有前置事实 $f_{\text{pre}}\in F$ 与后置事实 $f_{\text{post}}\in F$。其中，f_{pre} 是执行 a_i 前必须存在的事实，如果在执行能力 a_i 时 f_{pre} 未知，则 a_i 无法执行，如执行窃取敏感文件能力前必须知道敏感文件的存储路径这一前置事实；f_{post} 是执行 a_i 后生成的事实，其可以作为其他能力的前置事实，如发现敏感文件能力执行后可生成敏感文件路径的后置事实，此后置事实可作为窃取敏感文件能力的前置事实。事实 $f\in F$ 是关于给定环境的可识别信息，包括事实名与事实值，前者用于能力匹配事实，后者用于能力加载事实。目标 g 是攻击者定义的在本次测试中所要达到的目标。

2.4.2 决策模型

图3所示为决策模型的具体攻击流程。决策模块根据攻击策略选择对手的能力发送给代理，代理接收能力后在目标环境进行能力(命令)的执行，环境会反馈给代理执行结果，之后代理将能力执行结果返回到决策模块，决策模块根据返回结果来观测环境的状态，例如发现敏感文件能力在目标环境执行成功，则代理会返回敏感文件存在的路径，可认为环境的状态发生了改变，由当前状态转移至目标环境存在敏感文件的状态。如果能力的执行对达到目标有推进作用，那么决策模块会获得奖励；反之，决策模块获得惩罚。通过不断的训练，决策模块会偏向于选择对达到目标有推进作用的能力，从而获得最大累计奖励。

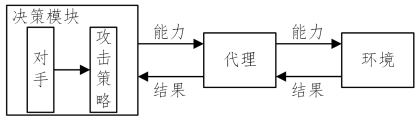


图 3 攻击流程

Fig. 3 Attack process

3 自动化攻击策略实现

自动化攻击策略的实现依赖于决策模块，其通过控制不同能力的执行顺序来达到测试目标。本文设计了两种攻击策略，一种是固定策略，另一种是强化学习策略。

3.1 对手构建

本文以窃取者作为对手，其事实如表2所列，能力与事实的关联如表3所列，能力命令如表4所列。窃取者的目标为窃取主机敏感文件。初始状态下，窃取者只拥有敏感文件后缀事实(file. sensitive. extension)。

表 2 事实列表

Table 2 Fact list

事实名	事实值
file. sensitive. extension(f_1)	png
host. dir. staged(f_2)	目录路径
host. file. path(f_3)	文件路径
host. dir. compress(f_4)	压缩路径

表 3 能力与事实的关联

Table 3 Correlation of abilities and facts

能力名	前置事实	后置事实
Create staging directory(a_1)	无	f_2
Find files(a_2)	f_1	f_3
Stage sensitive files(a_3)	f_2，f_3	无
Compress staged directory(a_4)	f_2	f_4
Exfil staged directory(a_5)	f_4	无

表 4 能力命令列表

Table 4 Ability command list

能力	Windows 命令	Linux 命令
a_1	New-Item-Path "."-Name "staged"-ItemType "directory"-Force | foreach { $_.FullName} | Select-Object	mkdir-p staged &&. echo $PWD/staged
a_2	Get-ChildItem C:\\Users-Recurse-Include *.#{f_1}-ErrorAction 'SilentlyContinue' | foreach { $ _. FullName} | Select-Object-first 5;exit 0;	find/-name '*.#{f_1}'-type f-not-path '*/\\.*'-size-500k 2 >/dev/null | head-5
a_3	Copy-Item #{f_3[filters(technique=T1005,max=1)]} #{f_2[filters(max=1)]}	cp #{f_3[filters(technique=T1005,max=1)]} #{f_2[filters(max=1)]}
a_4	Compress-Archive-Path #{f_2}-DestinationPath #{f_2}.zip-Force; sleep 1;ls #{f_2}.zip | foreach { $_. FullName} | select	tar-P-zcf #{f_2}.tar.gz #{f_2} &&. echo #{f_2}.tar.gz
a_5	$ErrorActionPreference='Stop'; $fieldName="#{$f_4$}"; $filePath="#{$f_4$}"; $url="server/file/upload";	curl-F "data=@#{f_4}"--header " X-Request-ID:'hostname'-paw" server/file/upload

3.2 固定策略

固定策略被用来作为强化学习策略的性能评估基线。最简单的固定策略是随机策略(Random),对于任意给定对手,它采用完全随机的方式从对手能力列表中弹出一个能力并执行(无论能力是否能执行),直到对手能力列表中所有能力全部弹出。

另一个策略是贪婪策略(Greedy),它的执行顺序与预定义的对手能力列表一致,如果最优先的能力可以执行,它总是选择那个能力,如果不能执行,则检查下一个能力,以此类推。它总是尽可能地将对手的能力全部执行,其根据前置事实与后置事实对能力执行顺序进行了调整,使得能力在满足条件的情况下一定能得到执行。部分能力即使存在前置事实,也会在执行过程中加入到等待队列中,一旦其他能力执行后所更新的后置事实与其前置事实匹配,则处于等待的能力立即会被执行。可见,贪婪策略的性能一定程度上依赖于对手预定义的能力列表次序。

3.3 强化学习策略

采用强化学习策略的决策模块在训练中可以根据自身所获得的奖励来更新动作选择策略,从而得到最优的动作序列,即代理执行该动作序列可以获得最大的累积奖励。

3.3.1 构建马尔可夫决策过程

决策模型通过输入的对手信息构建动作空间与状态空间,构建过程如图 4 所示。

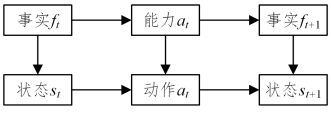


图 4 动作空间与状态空间的构建

Fig. 4 Construction of action space and state space

马尔可夫决策过程中,动作空间由对手的能力映射而来,每个能力对应一个动作,如能力 a_1($ability_1$)映射为动作 a_1($action_1$);状态空间则由对手的事实构建,在时刻 t,对手拥有事实 $f_t \in F$,决策模块观测到的环境状态映射为 $s_t \in S$,此时决策模块选择能力 $a_t \in A$,如果事实 f_t 为能力 a_t 的前置事实,那么在能力 a_t 执行成功后,对手会获得新的事实 f_{t+1},同时决策模块观测的环境状态转移到状态 $s_{t+1} \in S$,其中 f_{t+1} 为能力 a_t 的后置事实,否则决策模块无法执行能力 a_t,此时决策模块观测的环境状态不会发生改变。

根据窃取者的信息与上述状态空间和动作空间的构建过程,可将攻击场景建模为图 5 所示的马尔可夫决策过程。图中逆时针回环箭头表示在该状态下代理选择的相应动作无法执行或执行后对达到最终目标没有推进作用,如在初始状态 s_0 执行动作 a_4(压缩暂存目录),此时 a_4 的前置事实 f_2(暂存目录路径)不存在,动作 a_4 无法执行。

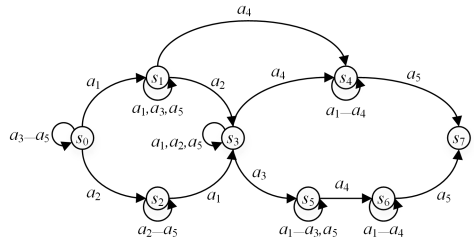


图 5 攻击建模过程

Fig. 5 Attack modeling process

奖励值 R 是引导决策模块学习的基础,窃取者的目标是窃取到敏感文件,因此只有达到最终目标才能获得 $R=10$ 的奖励,即 $R(s_6,a_5)=10$;其余状态转移获得 $R=-1$,作为执行动作的消耗,如 $R(s_0,a_1)=-1$;若选择的动作无法执行或对达到最终目标没有推进作用,则给予 $R=-10$ 的惩罚,如 $R(s_0,a_4)=-10$。

3.3.2 强化学习攻击策略实现

本文以有限状态机模型为基础,根据窃取者的信息与马尔可夫决策过程来实现强化学习攻击策略。

Q-Learning 攻击策略的训练过程如图 6 所示。该策略一共包含 8 个状态机,其中决策状态机主要根据强化学习算法与参数选择动作;5 个动作状态机分别对应 5 个能力,作用是根据决策状态机选择的动作来将对应的能力(命令)发送给远程代理进行执行,然后接收代理执行结果,再根据执行结果与马尔可夫决策过程判定环境状态的转移与对应的奖励;更新状态机负责将环境状态的转移与奖励根据强化学习更新公式来更新 Q 表;当更新状态机发现强化学习代理已到达最终状态或本次训练回合超过了最大步数,则会转移至终止状态机,终止状态机会记录本次训练的结果,将 Q 表的值进行保存,然后判断是否满足停止训练条件(如训练次数),若不满足,则初始化环境状态为初始状态 s_0,再开始下一次训练,读取保存的 Q 表,并以此为基础继续训练。当强化学习策略训练完成后,可以直接读取训练好的 Q 表进行测试。

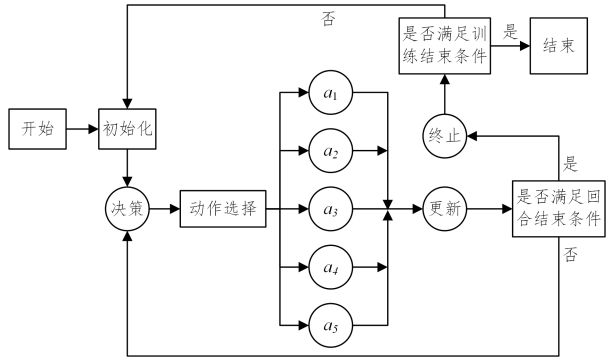


图 6 训练过程

Fig. 6 Training process

SARSA 与 SARSA(λ)攻击策略的实现与 Q-Learning 攻击策略基本一致,只是在对 Q 表进行更新时有所不同,SARSA 攻击策略会在更新状态机处选择下一次的动作,并将其

用于 Q 表的更新；而 SARSA(λ)则是在 SARSA 的基础上加入了资格迹的更新，同时对已经完成的行动的 Q 值也进行了更新。

4 实验

本文以窃取目标主机敏感文件的红队测试过程为实验场景，在真实实验环境中对强化学习策略的收敛性与可用性进行测试：首先，在相同的实验场景下训练不同的强化学习算法，对比它们在限定的训练次数下的收敛速度；其次，在完成强化学习策略的训练后，对比固定策略与强化学习策略在同一环境下的测试性能。

4.1 实验场景

实验场景如图 7 所示，Kali linux 被设定为攻击机，采用 Caldera v4.1.0 自动化对手仿真平台来进行自动化红队测试；CentOS 7 和 Windows 10 被设定为目标机，目标机各设置 1 张图片作为敏感文件，文件后缀为 png。

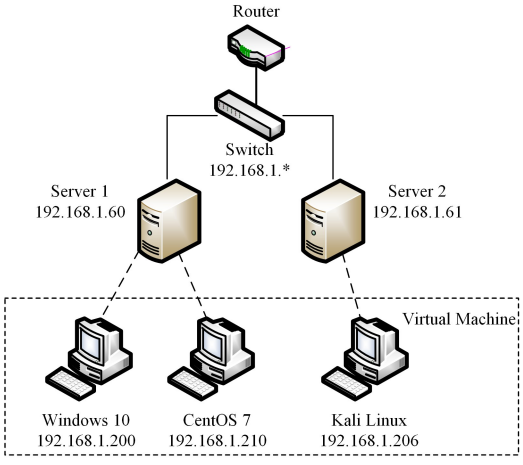


图 7 实验场景

Fig. 7 Experiment scenario

首先在目标主机上部署代理，然后选择对手为窃取者，选择决策策略为强化学习策略，分别对 3 种强化学习策略进行训练，训练回合数为 500，每个训练回合允许的最大步数为 20，超过 20 步没有达到最终状态则视作代理没有完成目标，结束该回合。

训练完毕后，将上述强化学习策略与固定策略在相同实验场景下进行测试，测试回合数为 1 000，目标是窃取到目标主机敏感文件。测试实验保留了强化学习策略的探索率 ε，其值与训练阶段一致；贪婪策略的预定义对手列表由程序随机生成。强化学习算法的其他参数设置如表 5 所列。

表 5 算法参数列表

Table 5 Algorithm parameters

参数	参数意义	参数值
Episodes	训练回合数	500
Max Steps	每个训练回合允许的最大步数	20
α	学习率	0.01
ε	探索率	0.1
γ	折扣率	0.8
λ	衰减率	0.9

4.2 实验结果

图 8—图 10 分别给出了 3 种强化学习算法奖励值随训练回合数的变化。

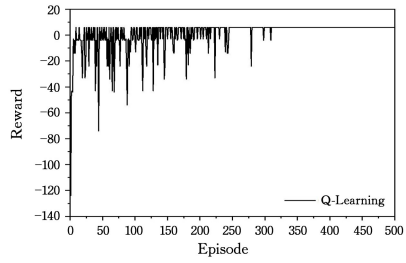


图 8 Q-Learning 算法奖励值随训练回合数的变化

Fig. 8 Reward of Q-Learning algorithm versus training episodes

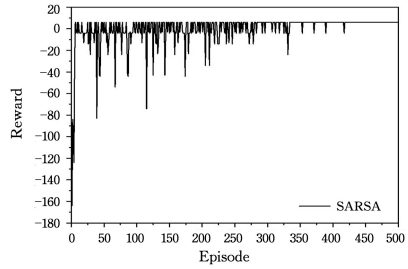


图 9 SARSA 算法奖励值随训练回合数的变化

Fig. 9 Reward of SARSA algorithm versus training episodes

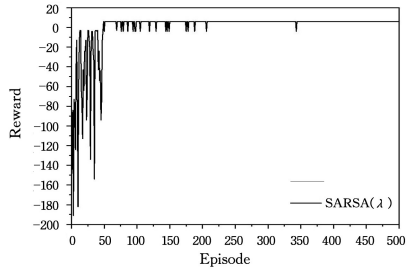


图 10 SARSA(λ)算法奖励值随训练回合数的变化

Fig. 10 Reward of SARSA(λ) algorithm versus training episodes

上述 3 种强化学习策略的训练结果如表 6 所列，测试结果如表 7 所列。

表 6 训练结果

Table 6 Training results

算法	收敛回合	稳定收敛回合	最大奖励	平均奖励
Q-Learning	245	310	6	6
SARSA	335	418	6	5.9
SARSA(λ)	51	344	6	6

表 7 测试结果

Table 7 Test results

攻击策略	测试次数	成功窃取次数	目标完成率/%
Random	1 000	17	1.7
Greedy	1 000	383	38.3
Q-Learning	1 000	956	95.6
SARSA	1 000	955	95.5
SARSA(λ)	1 000	961	96.1

4.3 实验结果分析

由图 8、图 9、图 10 以及表 6 可知，3 种强化学习算法均能收敛到最优解，即获得最大奖励；SARSA(λ)趋向于收敛所用的回合最少，Q-Learning 次之，SARSA 最多；Q-Learning 达到稳定收敛所用的回合最少，SARSA(λ)次之，但与 Q-Learning 差距较小，SARSA 最多。总体而言，SARSA(λ)表现出来的性能更好，这是因为 SARSA(λ)记录了每个回合的“行动轨迹”，更新 Q 值时不仅更新当前状态的 Q 值，而且对已经完成

的行动的 Q 值也进行更新，从而使得其收敛速度远快于其他算法。

由表 7 可知，固定策略中随机策略性能最差，贪婪策略性能较随机策略有所提升。贪婪策略表现更好的原因在于随机策略在发现能力无法执行，即当前状态不满足前置事实时，将能力丢弃，而贪婪策略则将其加入到等待队列中，等待前置事实的满足；强化学习策略的性能远优于固定策略，3 种强化学习策略的性能基本相同，目标完成率均在 95%以上。测试实验验证了强化学习策略在自动化红队测试实验场景中的可行性。

结束语 本文主要研究了在自动化红队测试中应用强化学习算法的可行性，建立了红队攻击技术与强化学习两者属性的映射，从而将红队测试过程建模为马尔可夫决策过程，在真实实验环境中对不同的强化学习策略进行训练和测试，验证了强化学习策略的收敛性和可行性。实验结果表明，基于 SARSA(λ)算法的强化学习策略优于其他强化学习算法，收敛性较好；3 种强化学习策略均能在测试实验中稳定完成测试目标，且性能远大于固定策略。

受限于红队测试技术的多样性与复杂性，目前 MDP 模型需要测试人员手动构建，且构建过程较为复杂，因此本文所构建的自动化红队测试的实验场景较小，所包含的动作较少，未来需要加入更多 ATT&CK 模型中的战术和技术，构建更加复杂的红队测试场景，实现 MDP 模型的自动化构建。除此之外，本文所采用的强化学习算法较为基础，如果扩大红队测试场景规模，加入更多攻击技术后，Q 表的扩大会导致强化学习策略性能受到影响，因此未来需要考虑采用 DQN 和 A3C 等深度强化学习算法。

参 考 文 献

[1] XIONG Y. Design and Implementation of Automatic Penetration Testing Platform[D]. Beijing: Beijing University of Posts and Telecommunications, 2019.

[2] APPLEBAUM A, MILLER D, STROM B, et al. Intelligent, Automated Red Team Emulation[C]// Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016: 363-373.

[3] GANGUPANTULU R, CODY T, PARK P, et al. Using Cyber Terrain in Reinforcement Learning for Penetration Testing [C]// 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). IEEE, 2022: 1-8.

[4] HU Z, BEURAN R, TAN Y. Automated Penetration Testing Using Deep Reinforcement Learning[C]// 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, 2020: 2-10.

[5] POZDNIAKOV K, ALONSO E, STANKOVIC V, et al. Smart Security Audit: Reinforcement Learning with a Deep Neural Network Approximator[C]// 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). IEEE, 2020: 1-8.

[6] SARRAUTE C, BUFFET O, HOFFMANN J. POMDPs Make Better Hackers: Accounting for Uncertainty in Penetration Testing[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2012: 1816-1824.

[7] SHMARYAHU D, SHANI G, HOFFMANN J, et al. Simulated Penetration Testing as Contingent Planning[C]// Proceedings of the International Conference on Automated Planning and Scheduling. 2018: 241-249.

[8] LI T, CAO S J, YIN S W, et al. Optimal method for the generation of the attack path based on the Q-Learning decision[J]. Journal of Xidian University, 2021, 48(1): 160-167.

[9] MAEDA R, MIMURA M. Automating post-exploitation with deep reinforcement learning[J]. Computers & Security, 2021, 100: 102-108.

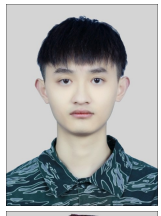
[10] The MITRE ATT&CK. Adversarial Tactics, Techniques, and Common Knowledge[EB/OL]. (2022-10-25)[2022-12-13]. https://attack.mitre.org/.

[11] The MITRE CALDERA. A Scalable, Automated Adversary Emulation Platform [EB/OL]. (2022-09-20) [2022-12-13]. https://caldera.mitre.org/.

[12] QIN Z H, LI N, LIU X T, et al. Overview of Research on Model-free Reinforcement Learning [J]. Computes Science, 2021, 48(3): 180-187.

[13] GAO Y, CHEN S F, LU X. Research on Reinforcement Learning Technology: A Review[J]. Acta Automatica Sinica, 2004, 30(1): 86-100.

[14] CHEN S L, WEI Y M. Least-squares SARSA(Lambda) algorithms for reinforcement learning[C]// 2008 Fourth International Conference on Natural Computation. IEEE, 2008: 632-636.



CHEN Yufei, born in 1997, postgraduate. His main research interests include cyberspace security and reinforcement learning.

LI Saifei, born in 1988, Ph.D, engineer. His main research interests include cyberspace security and so on.