·热点与综述·　　文章编号：1000-3428(2022)04-0030-09　　文献标志码：A　　中图分类号：TP309

基于演化博弈的NFV拟态防御架构动态调度策略

张青青，汤红波，游　伟，普黎明

（中国人民解放军战略支援部队信息工程大学 国家数字交换系统工程技术研究中心，郑州 450002）

摘　要：构建网络功能虚拟化(NFV)拟态防御架构能够打破防御滞后于攻击的攻防不对等格局，其中动态调度策略是关键实现技术。然而，现有拟态防御架构中的动态调度策略大多根据执行体自身固有的特点进行调度，没有进一步利用裁决机制对异常执行体的定位感知能力做优化调整。通过引入演化博弈理论，设计一种新的NFV拟态防御架构动态调度策略。在NFV拟态防御架构中增加一个分析器，用于对历史裁决信息进行分析研究。根据分析器中得到的反馈信息，从攻防双方的有限理性出发构建多状态动态调度演化博弈模型，并采用复制动态方程求解该博弈模型的演化均衡策略，利用李雅普诺夫间接法对均衡策略进行稳定性分析，提出基于演化博弈的动态调度策略选取算法。仿真结果表明，该策略能够利用裁决机制对异常执行体的定位感知能力，通过深入分析研究和不断调整优化选择具有适应性和针对性的调度策略，有效提升系统的安全收益和防御效能。

关键词：网络功能虚拟化；拟态防御；动态调度；演化博弈；裁决机制

开放科学(资源服务)标志码(OSID)：

中文引用格式：张青青，汤红波，游伟，等. 基于演化博弈的NFV拟态防御架构动态调度策略[J]. 计算机工程，2022，48(4)：30-38，49.

英文引用格式：ZHANG Q Q，TANG H B，YOU W，et al. Dynamic scheduling strategy of NFV mimic defense architecture based on evolutionary game[J]. Computer Engineering，2022，48(4)：30-38，49.

Dynamic Scheduling Strategy of NFV Mimic Defense Architecture Based on Evolutionary Game

ZHANG Qingqing，TANG Hongbo，YOU Wei，PU Liming

（National Digital Switching System Engineering & Research Center，People's Liberation Army Strategic Support Force Information Engineering University，Zhengzhou 450002，China）

【Abstract】Constructing a Network Functions Virtualization（NFV）mimic defense architecture can break the unequal pattern of attack and defense with defense lagging behind attack，where the key implementation technology is the dynamic scheduling strategy.However，most dynamic scheduling strategies in the existing mimic defense architecture schedule according to the inherent characteristics of the executor，and do not further use the adjudication mechanism to optimize and adjust the location perception ability of the abnormal executor.To address this problem，this paper proposes a new dynamic scheduling strategy for the NFV mimic defense architecture by introducing evolutionary game theory.First，an analyzer is added to the NFV mimic defense architecture to analyze and research historical arbitration information.Second，based on the feedback information obtained from the analyzer，a multi-state dynamic scheduling evolutionary game model is built based on the bounded rationality of both offense and defense，using the replication dynamic equation to solve the evolutionary equilibrium strategy，and Lyapunov's indirect method to analyze the stability of the equilibrium strategy.Finally，an optimal dynamic scheduling strategy selection algorithm，based on evolutionary game theory，is proposed.The simulation results show that the method proposed in this paper can effectively use the arbitration mechanism ability to locate and perceive abnormal executive bodies.Through in-depth analysis and continuous optimization，an adaptive and targeted scheduling strategy is selected to further enhance the security benefits and defense effectiveness of the system.

【Key words】Network Functions Virtualization（NFV）；mimic defense；dynamic scheduling；evolutionary game；arbitration mechanism

DOI：10. 19678/j. issn. 1000-3428. 0061282

基金项目：国家自然科学基金(61941114，61521003，61801515)。

作者简介：张青青(1995—)，女，硕士研究生，主研方向为网络功能虚拟化、网络空间安全；汤红波，教授；游　伟，讲师、博士；普黎明，副研究员、硕士。

收稿日期：2021-03-25　　**修回日期：**2021-04-29　**E-mail：**2532356150@qq.com

0　概述

针对传统网络架构部署周期长、运营成本高、网络结构僵化等问题，欧洲电信标准化协会（European Telecommunications Standards Institute，ETSI）提出了网络功能虚拟化（Network Functions Virtualization，NFV）技术[1]。NFV实现了网络功能与专有硬件之间的解耦，通过虚拟化技术实现网络功能的软件化部署，提高了网络的灵活性和资源的利用率，降低了运营商的资本支出（Capital Expenditure，CPEX）和运营开销（Operation Expenses，OPEX）。然而，NFV资源共享的特点也给网络带来了新的安全风险[2]。传统的被动防御手段面向已知特征的威胁，防御滞后于攻击，在应对NFV带来的新的安全风险时缺乏主动性和对攻击的预测能力。

应对上述攻防不对等的格局，可基于拟态防御思想[3]构建NFV拟态防御架构，利用网络系统环境的动态性和不确定性获得防御功能或属性，其基本原理是：通过为每一个网络功能构建多个功能等价的异构体，为系统引入多样性，同时利用动态调度策略定期或不定期地对线上执行体进行替换，为系统引入动态性与随机性，使得攻击者可利用的攻击面不断变化，从而增加攻击者的攻击难度。此外，利用裁决机制对输出结果进行多模裁决，提高系统可靠性与防御鲁棒性。其中动态调度策略是实现拟态防御的关键技术之一，设计合理的调度策略可以有效提升系统的安全性和异构体的利用率。在现有针对调度策略的相关研究中：文献[4]引入了信誉度与相异度2个指标，提出一种基于信誉度与相异度联合优化的执行体选择算法，在保证执行体之间差异性的同时避免了脆弱程度较低的执行体；文献[5]在考虑系统负载因素的情况下，将调度问题转化为安全与负载的动态双目标优化问题，兼顾了系统的安全性与计算性能；文献[6]提出一种随机种子最小相似度的调度策略，首先随机确定种子执行体，然后根据相似度指标选择整体相似度最小的调度方案，实现了动态性和可靠性之间的平衡；文献[7]提出一种基于优先级和时间片的执行体调度算法，首先从时空维度对执行体的相似性进行定义，然后基于相似性指标进行优先级预排序，最后结合时间片策略进行执行体调度。

上述调度策略虽然在一定程度上改善了拟态防御架构的动态性与随机性，但都是根据执行体自身固有的特点进行调度，没有进一步利用裁决机制对异常执行体的定位感知能力做优化调整。拟态裁决机制可以通过对执行体的输出结果进行对比，发现输出异常的执行体，并利用态势感知技术以及大数据分析对异常执行体的系统信息深入研究，从而得到当前环境下攻击的类型和分布。各异构体由于其组成结构不同，暴露的攻击面不同，对不同攻击的防御能力也不同，因此利用拟态裁决机制的反馈结果对调度策略进行适应性地优化调整，可以提高调度策略的有效性和系统的安全性。

为提升NFV拟态防御架构的安全性和调度策略的有效性，本文利用分析器对裁决器的裁决信息进行分析，得到关于攻击状态的反馈信息，并利用演化博弈理论构建一个多状态动态调度演化博弈模型，将在线执行体的不同组合作为NFV拟态防御架构的不同状态。在此基础上，利用复制动态方程和李雅普诺夫间接法对不同状态下攻防策略的演化趋势和稳定性进行分析，提出基于演化博弈的最优动态调度策略选取算法。

1　NFV拟态防御架构

NFV拟态防御架构是一个基于拟态防御理论的内生安全防御体系。图1为NFV拟态防御架构的示意图，可以看出该架构由异构元素池、异构体池、调度器、输入代理、裁决器、输出代理和分析器组成。

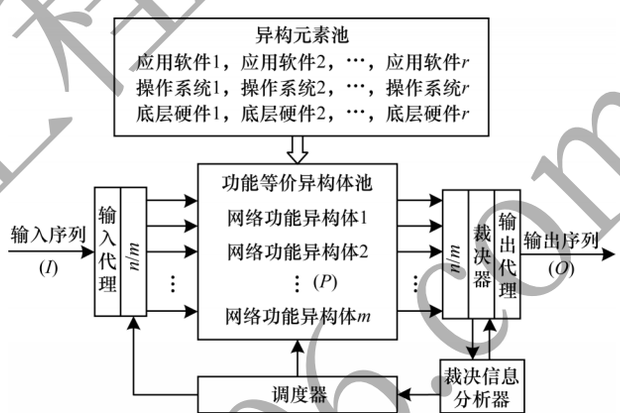

图1　NFV拟态防御架构

Fig.1　NFV mimic defense architecture

在NFV拟态防御架构中，每一个网络功能异构体由上而下分为3层：

1）应用软件层：采用不同编程语言或算法实现的虚拟网络功能（Virtual Network Function，VNF），是处理服务请求的逻辑功能。

2）操作系统层：包含不同的操作系统和虚拟化软件，操作系统可以采用Windows、Ubuntu、Centos等，虚拟化软件可以采用OpenStack、XEN等，为应用软件层提供必要的运行环境。

3）底层硬件层：可以使用不同厂家的服务器，如X86和ARM，其作用是搭载整个操作系统。

这些异构体在功能上是等效的，但是具有不同的组成结构，因此降低了共模漏洞的可能性。M个异构体组成一个异构体池，负反馈控制器依据动态调度策略从异构体池中选择N个异构体作为执行体处理网络请求，并不定期地对当前运行的执行体集合进行变换，对外呈现结构的随机性和不可预测性。输入代理将输入请求复制N份分发给执行体进行处理。裁决器对N个执行体的输出结果进行判决，最终形成一致性的输出。拟态裁决机制将单一确定目标攻击难度增强为多目标协同一致攻击难度。攻击者只有在同时攻破$k((N+1)/2 \leqslant k \leqslant N)$个执行体并产生相同的错误输出时，才能成功攻击NFV拟态防御架构。分析器对输出异常的执行体做进一步深入分

析，结合态势感知技术[8]和大数据分析技术，从中获取当前环境下攻击者的分布和攻击的类型，并将这些信息反馈给调度器，对调度策略进行优化调整。

2 多状态动态调度演化博弈模型

2.1 模型定义

演化博弈[9]是把传统博弈理论和生物进化理论结合起来的一种理论，其以有限理性作为理论分析的基础。演化博弈可以对博弈双方长期稳定趋势进行分析，符合实际网络攻防对抗动态演化的现实，能够有效提高利用博弈论分析网络对抗行为的准确性和可信度[10]。本节以演化博弈理论为基础构建多状态动态调度演化博弈模型。与模型相关的前提假设如下：

假设1 NFV拟态防御架构是基于“输入-处理-输出”的IPO（Inputs Process Outputs）模型[3]，对于相同的输入，会产生一致的输出。因此，本文仅针对由系统漏洞和后门引发的系统异常或者输出错误，而不考虑DDoS（Distributed Denial of Service）[11]等资源消耗型攻击，这样裁决器才能对输出结果进行比较，以判决执行体是否被攻击。

假设2 系统输出异常可能是由内部故障或者外部攻击导致。本文的研究目标是制定适应性的调度策略防御攻击者的攻击，因此，在导致执行体输出异常的因素中不考虑非攻击因素，如环境等原因导致的系统崩溃。

假设3 由于网络环境复杂多变，攻防双方获得的攻防信息是不完全的。此外，由于分析能力和计算能力有限，攻防双方无法通过一次博弈就能找到最优策略，因此本文假设攻防双方都是有限理性的，具有一定的统计分析能力和事后判断能力，但没有事先的预测能力，需要通过不断学习和调整来找到合适的策略。攻击者可以通过网络扫描和渗透等手段对NFV拟态防御架构进行探测，获取防御者的漏洞和后门信息；NFV拟态防御架构可以通过裁决机制感知异常执行体，并利用分析器得到攻击者的分布和攻击的类型。

基于以上假设，下文给出多状态动态调度演化博弈模型（Multi State Dynamic Scheduling Evolutionary Game Model，MSDSEGM）的相关定义。

定义1 多状态动态调度演化博弈模型可用一个五元组表示：$\mathrm{MSDSEGM}=(N,\Phi,S,P,U)$。

1）$N=(N_\mathrm{A},N_\mathrm{D})$是博弈的参与者集合。其中：$N_\mathrm{A}$为攻击者；$N_\mathrm{D}$为防御者（NFV拟态防御架构）。

2）$\Phi=\{\Phi_1,\Phi_2,\cdots,\Phi_\phi\}$是NFV拟态防御架构的状态集合，当前在线执行体的不同组合即为NFV拟态防御架构的不同状态。其中：$\phi=C_M^N$为状态总数，与异构体数量M和执行体数量N有关。当NFV拟态防御架构处于不同的状态时，攻防双方采取不同的博弈策略得到的博弈收益也不同，因此，需要对不同状态下的NFV拟态防御架构进行博弈分析。

3）$S=(S_\mathrm{A},S_\mathrm{D})$是博弈参与者的策略集合。其中：$S_\mathrm{A}=\{S_{\mathrm{A}1},S_{\mathrm{A}2},\cdots,S_{\mathrm{A}m}\}$表示攻击者可采取的攻击策略集合，即利用各种漏洞对NFV拟态防御架构进行攻击，m为攻击策略总数；$S_\mathrm{D}=\{S_{\mathrm{D}1},S_{\mathrm{D}2},\cdots,S_{\mathrm{D}n}\}$表示NFV拟态防御架构可采取的防御策略集合，即动态调度策略集合。动态调度策略就是NFV拟态防御架构不同状态之间的转换，因此，防御策略总数$n=\varnothing$。

4）$P=(p,q)$是博弈参与者的策略概率集合。其中：$p=\{p_1,p_2,\cdots,p_m\}$表示攻击者采取的各攻击策略的概率集合，且$\sum_{i=1}^{m}p_i=1(0\leqslant p_i\leqslant 1,m\geqslant 2)$；$q=\{q_1,q_2,\cdots,q_n\}$表示NFV拟态防御架构采取的各调度策略的概率集合，且$\sum_{j=1}^{n}q_j=1(0\leqslant q_j\leqslant 1,n\geqslant 2)$。

5）$U=(U_\mathrm{A},U_\mathrm{D})$是博弈参与者的收益函数集合。其中：$U_\mathrm{A}$表示攻击者的收益；$U_\mathrm{D}$表示NFV拟态防御架构的收益；$U_\mathrm{A}^{ij}$和$U_\mathrm{D}^{ij}$为攻击者和防御者分别采取策略$S_{\mathrm{A}i}$和$S_{\mathrm{D}j}$时的收益函数。由此得到博弈收益矩阵，如式（1）所示：

$$\begin{pmatrix} U_\mathrm{A}^{11},U_\mathrm{D}^{11} & U_\mathrm{A}^{12},U_\mathrm{D}^{12} & \cdots & U_\mathrm{A}^{1n},U_\mathrm{D}^{1n} \\ U_\mathrm{A}^{21},U_\mathrm{D}^{21} & U_\mathrm{A}^{22},U_\mathrm{D}^{22} & \cdots & U_\mathrm{A}^{2n},U_\mathrm{D}^{2n} \\ \vdots & \vdots & & \vdots \\ U_\mathrm{A}^{m1},U_\mathrm{D}^{m1} & U_\mathrm{A}^{m2},U_\mathrm{D}^{m2} & \cdots & U_\mathrm{A}^{mn},U_\mathrm{D}^{mn} \end{pmatrix} \tag{1}$$

MSDSEGM模型对应的攻防博弈树如图2所示，由图可知攻击者和NFV拟态防御架构采取不同攻防策略时得到的攻防收益也会不同。该收益不仅与其自身的策略有关，而且还与对方的策略密切相关。

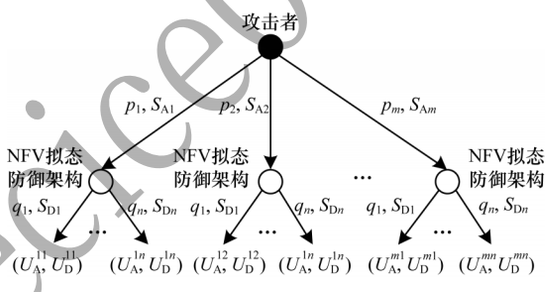

图2 网络攻防博弈树

Fig.2 Network attack and defense game tree

2.2 博弈收益量化

对攻击者和防御者的博弈收益进行量化是MSDSEGM模型演化均衡求解和稳定性分析的基础，其量化是否合理直接决定最终博弈结果的准确性。文献[12]在总结多种攻击防御策略分类的基础上，提出了成本/回报量化方法，即收益为回报减去成本。基于此，在对MSDSEGM模型进行分析之前，结合NFV拟态防御架构的特点给出以下关于攻防双方收益的定义。

定义2（攻击成本A_C） 攻击者发动一次攻击所需耗费的时间开销、软硬件资源开销、风险开销等。

定义3（攻击回报A_R） 攻击者发动一次攻击所得到的好处，用攻击者给NFV拟态防御架构带来的损失表示。

定义4（防御成本D_C） 由式（2）可知防御成本

由操作成本和负面成本构成：

$$D_C=\alpha\times O_C+N_C \tag{2}$$

其中：O_C是操作成本，表示NFV拟态防御架构置换一个异构体所需耗费的时间开销和计算资源开销；α为调度策略置换的异构体数量；N_C是负面成本，表示执行调度策略给系统带来的一段时间内的性能损失。

定义5（防御损失D_L）　防御方因遭受攻击方攻击而造成的损失。防御损失D_L等于攻击回报A_R，计算公式如下：

$$D_L=\beta\times\mu\times V_L+F_L \tag{3}$$

由式(3)可知，防御损失由漏洞损失V_L和失效损失F_L构成。其中：V_L表示攻击者利用某个漏洞攻击单个异构体造成的损失；β为被攻击的异构体数量，μ为漏洞的攻击成功概率；F_L表示系统中超过一半数量异构体被攻破时的损失，此时NFV拟态防御架构失效。

定义6（防御回报D_R）　每一个异构体为NFV拟态防御架构带来的安全收益。

定义7（底层负载U_L）　NFV技术将网络功能和硬件实体解耦分离，每一个异构体都是一个网络功能实例，与其他网络功能实例共享底层物理资源。底层负载影响着网络功能的运行效率[13]，应尽量选取底层负载较小的异构体上线。

根据上述定义，可以得到博弈收益函数的具体表达式：

$$U_A^{ij}=A_R-A_C \tag{4}$$

$$U_D^{ij}=N\times D_R-D_C-D_L-U_L \tag{5}$$

本节涉及的相关变量及含义如表1所示。

表1　主要变量及含义

Table 1　Main variables and their significance

变量	含义
A_C	攻击成本
A_R	攻击回报
D_C	防御成本
α	防御操作置换的异构体数量
O_C	操作成本
N_C	负面成本
D_L	防御损失
β	被攻击的异构体数量
μ	漏洞攻击成功概率
V_L	漏洞损失
F_L	失效损失
D_R	防御回报
U_L	底层负载
U_A^{ij}	采取策略S_{Ai}和S_{Dj}时的攻击收益
U_D^{ij}	采取策略S_{Ai}和S_{Dj}时的防御收益

2.3　演化均衡求解

本节使用复制动态（Replicator Dynamics，RD）方法对MSDSEGM模型进行演化均衡求解。复制动态方程就是策略选取概率的动态变化速率，描述的是博弈过程中攻防策略随时间的动态调整过程，是目前应用最为广泛的一种动力学演化机制[14]。由于在有限理性的条件下博弈参与者掌握的初始知识是有限的，因此采取的策略并非完全理性条件下的最优策略，在博弈的过程中通过不断调整和改进，收益低的策略的选取概率逐渐降低，收益高的策略的选取概率逐渐提高，最终达到策略均衡的结果。

采用复制动态方程求解MSDSEGM模型演化均衡的具体过程为：

1）根据攻防策略概率集合p、q和收益矩阵，计算攻击者不同攻击策略的期望收益U_{Ai}和平均收益U_A：

$$U_{Ai}=q_1U_A^{i1}+q_2U_A^{i2}+\cdots+q_nU_A^{in}=\sum_{j=1}^{n}q_jU_A^{ij} \tag{6}$$

$$U_A=p_1U_{A1}+p_2U_{A2}+\cdots+p_mU_{Am}=\sum_{i=1}^{m}p_iU_{Ai} \tag{7}$$

2）计算攻击者的复制动态方程：

$$A(p_i)=\frac{\mathrm{d}p_i(t)}{\mathrm{d}t}=p_i(t)(U_{Ai}-U_A) \tag{8}$$

其中：$p_i(t)$表示随时间t变化选择攻击策略S_{Ai}的概率，且满足$\sum_{i=1}^{m}p_i(t)=1$。由式(8)可知，攻击策略S_{Ai}的动态方程取值与选择该策略的概率p_i成正比，与其期望收益U_{Ai}和平均收益U_A的差值成正比。

3）根据攻防策略概率集合p、q和收益矩阵，计算NFV拟态防御架构不同调度策略的期望收益U_{Dj}和平均收益U_D：

$$U_{Dj}=p_1U_D^{1j}+p_2U_D^{2j}+\cdots+p_mU_D^{mj}=\sum_{i=1}^{m}p_iU_D^{ij} \tag{9}$$

$$U_D=q_1U_{D1}+q_2U_{D2}+\cdots+q_nU_{Dn}=\sum_{j=1}^{n}q_jU_{Dj} \tag{10}$$

4）计算NFV拟态防御架构的复制动态方程：

$$D(q_j)=\frac{\mathrm{d}q_j(t)}{\mathrm{d}t}=q_j(t)(U_{Dj}-U_D) \tag{11}$$

其中：$q_j(t)$表示随时间t变化选择选择调度策略S_{Dj}的概率，且满足$\sum_{j=1}^{n}q_j(t)=1$。由式(11)可知，调度策略S_{Dj}的动态方程取值与选择该策略的概率q_j成正比，与其期望收益U_{Dj}和平均收益U_D的差值成正比。

5）联立攻击者与NFV拟态防御架构的复制动态方程进行演化均衡求解：

$$\begin{cases}A(p_1)=\dfrac{\mathrm{d}p_1(t)}{\mathrm{d}t}=p_1(t)(U_{A1}-U_A)=0\\ \quad\vdots\\ A(p_m)=\dfrac{\mathrm{d}p_m(t)}{\mathrm{d}t}=p_m(t)(U_{Am}-U_A)=0\\ D(q_1)=\dfrac{\mathrm{d}q_1(t)}{\mathrm{d}t}=q_1(t)(U_{D1}-U_D)=0\\ \quad\vdots\\ D(q_n)=\dfrac{\mathrm{d}q_n(t)}{\mathrm{d}t}=q_n(t)(U_{Dn}-U_D)=0\end{cases} \tag{12}$$

对式(12)进行求解，可以得到MSDSEGM模型的演化均衡策略，即在该策略下各个博弈行为的选择概率不变。但是其中的一些演化均衡策略存在不稳定性，即一旦发生博弈双方的策略偏离该均衡状态的情况，复制动态方程就会使演化结果不再收敛于该策略。因此，需要对演化均衡策略进行稳定性分析，得到演化均衡策略中的稳定策略，实现最优动态调度策略选取。

2.4　演化稳定策略

演化稳定策略（Evolutionarily Stable Strategy，ESS）是演化博弈中的一个核心概念[15]，是对纳什均衡的改

进，具有较强的稳定性和预测能力，其在受到少量干扰后仍能恢复。演化稳定策略的数学定义为：

定义8 如果 $\forall y \in S, y \neq x$，存在一个 $\varepsilon_y \in (0,1)$，使式(13)对任意 $\varepsilon \in (0,\varepsilon_y)$ 都成立，则 $x \in S$ 为演化稳定策略。

$$U\left[x,\varepsilon_y+(1-\varepsilon)x\right] > U\left[y,\varepsilon_y+(1-\varepsilon)x\right] \tag{13}$$

其中：S 为策略集；y 表示突变策略；ε_y 是一个与突变策略 y 相关的常数，称为入侵界限；$\varepsilon_y+(1-\varepsilon)x$ 表示演化稳定策略与突变策略按比例组合而成的混合策略；$U(x,y)$ 为策略 x 与策略 y 相遇时策略 x 的收益函数。由定义8可知，演化稳定策略的基本思想是：在给定环境下，如果一个策略被群体大部分个体所采用，并且由于其他策略无法产生比使用该策略更高的收益，该策略无法被其他策略所代替，则称该策略为演化稳定策略。

针对2.3节采用复制动态方程求出的演化均衡策略，利用李雅普诺夫间接法[16]对其进行稳定性分析。由于式(12)是一个非线性微分方程组并且二阶连续可微，因此可以通过研究其在均衡点 $\boldsymbol{O}=[p_{O1},p_{O2},\cdots,p_{Om},q_{O1},q_{O2},\cdots,q_{On}]^{\mathrm{T}}$ 处的一次近似方程组的稳定性来实现演化均衡策略的稳定性分析。将式(12)在演化均衡点处进行泰勒展开得到式(14)：

$$\begin{cases}
A(p_1)=A(p_1)\big|_{\boldsymbol{O}}+\dfrac{\partial A(p_1)}{\partial p_1}(\boldsymbol{O})(p_1-p_{O1})+\cdots+\dfrac{\partial A(p_1)}{\partial p_m}(\boldsymbol{O})(p_m-p_{Om})+\dfrac{\partial A(p_1)}{\partial q_1}(\boldsymbol{O})(q_1-q_{O1})+\cdots+\\
\qquad\dfrac{\partial A(p_1)}{\partial q_n}(\boldsymbol{O})(q_n-q_{On})+\xi_1(p_1,p_2,\cdots,p_m,q_1,q_2,\cdots,q_n)\\
\quad\vdots\\
A(p_m)=A(p_m)\big|_{\boldsymbol{O}}+\dfrac{\partial A(p_m)}{\partial p_1}(\boldsymbol{O})(p_1-p_{O1})+\cdots+\dfrac{\partial A(p_m)}{\partial p_m}(\boldsymbol{O})(p_m-p_{Om})+\dfrac{\partial A(p_m)}{\partial q_1}(\boldsymbol{O})(q_1-q_{O1})+\cdots+\\
\qquad\dfrac{\partial A(p_m)}{\partial q_n}(\boldsymbol{O})(q_n-q_{On})+\xi_m(p_1,p_2,\cdots,p_m,q_1,q_2,\cdots,q_n)\\
D(q_1)=D(q_1)\big|_{\boldsymbol{O}}+\dfrac{\partial D(q_1)}{\partial p_1}(\boldsymbol{O})(p_1-p_{O1})+\cdots+\dfrac{\partial D(q_1)}{\partial p_m}(\boldsymbol{O})(p_m-p_{Om})+\dfrac{\partial D(q_1)}{\partial q_1}(\boldsymbol{O})(q_1-q_{O1})+\cdots+\\
\qquad\dfrac{\partial D(q_1)}{\partial q_n}(\boldsymbol{O})(q_n-q_{On})+\zeta_1(p_1,p_2,\cdots,p_m,q_1,q_2,\cdots,q_n)\\
\quad\vdots\\
D(q_n)=D(q_n)\big|_{\boldsymbol{O}}+\dfrac{\partial D(q_n)}{\partial p_1}(\boldsymbol{O})(p_1-p_{O1})+\cdots+\dfrac{\partial D(q_n)}{\partial p_m}(\boldsymbol{O})(p_m-p_{Om})+\dfrac{\partial D(q_n)}{\partial q_1}(\boldsymbol{O})(q_1-q_{O1})+\cdots+\\
\qquad\dfrac{\partial D(q_n)}{\partial q_n}(\boldsymbol{O})(q_n-q_{On})+\zeta_n(p_1,p_2,\cdots,p_m,q_1,q_2,\cdots,q_n)
\end{cases} \tag{14}$$

其中：ξ_i 和 ζ_j 为级数展开式中二阶以上各项之和。记：

$$\boldsymbol{x}=[p_1,p_2,\cdots,p_m,q_1,q_2,\cdots,q_n]^{\mathrm{T}} \tag{15}$$

$$\boldsymbol{f}=[A(p_1),A(p_2),\cdots,A(p_m),D(q_1),D(q_2),\cdots,D(q_n)]^{\mathrm{T}} \tag{16}$$

$$\boldsymbol{f_O}=[A(p_1)\big|_{\boldsymbol{O}},\cdots,A(p_m)\big|_{\boldsymbol{O}},D(q_1)\big|_{\boldsymbol{O}},\cdots,D(q_n)\big|_{\boldsymbol{O}}]^{\mathrm{T}} \tag{17}$$

令 $\boldsymbol{g}=\boldsymbol{f}-\boldsymbol{f_O}$，$\boldsymbol{y}=\boldsymbol{x}-\boldsymbol{O}$，可以得到一次近似方程组的矩阵向量形式 $\boldsymbol{g}=\boldsymbol{J}\boldsymbol{y}$，其中，$\boldsymbol{J}$ 为向量函数的雅可比矩阵，如式(18)所示：

$$\boldsymbol{J}=\begin{pmatrix}
\dfrac{\partial A(p_1)}{\partial p_1} & \cdots & \dfrac{\partial A(p_1)}{\partial p_m} & \dfrac{\partial A(p_1)}{\partial q_1} & \cdots & \dfrac{\partial A(p_1)}{\partial q_n}\\
\vdots & & \vdots & \vdots & & \vdots\\
\dfrac{\partial A(p_m)}{\partial p_1} & \cdots & \dfrac{\partial A(p_m)}{\partial p_m} & \dfrac{\partial A(p_m)}{\partial q_1} & \cdots & \dfrac{\partial A(p_m)}{\partial q_n}\\
\dfrac{\partial D(q_1)}{\partial p_1} & \cdots & \dfrac{\partial D(q_1)}{\partial p_m} & \dfrac{\partial D(q_1)}{\partial q_1} & \cdots & \dfrac{\partial D(q_1)}{\partial q_n}\\
\vdots & & \vdots & \vdots & & \vdots\\
\dfrac{\partial D(q_n)}{\partial p_1} & \cdots & \dfrac{\partial D(q_n)}{\partial p_m} & \dfrac{\partial D(q_n)}{\partial q_1} & \cdots & \dfrac{\partial D(q_n)}{\partial q_n}
\end{pmatrix} \tag{18}$$

当带入均衡点 $\boldsymbol{O}$ 时，雅可比矩阵 $\boldsymbol{J}$ 的所有特征根均有负实部，则该点对应的策略为演化稳定策略。

3 基于演化博弈的动态调度策略选取算法

基于演化博弈的动态调度策略工作流程如图3所示。

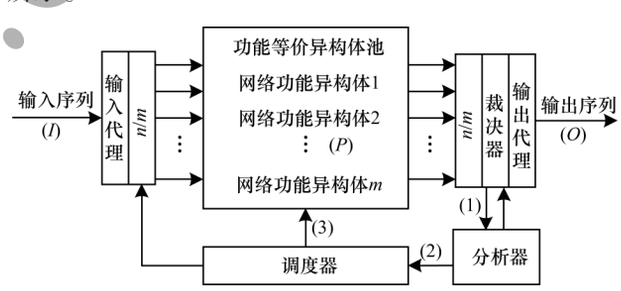

图3 基于演化博弈的动态调度策略工作流程

Fig.3 Workflow of dynamic scheduling strategy based on evolutionary game

1)分析器根据裁决结果，利用漏洞扫描、数据采集、查杀病毒木马、沙箱隔离等防护手段对异常执行体进行故障查找和安全问题排查，或利用态势感知和大数据分析技术对异常执行体的系统信息深入研究，得到当前环境下的攻防状态信息(攻击的类型和

分布、各异构体对不同攻击的防御情况）。

2）调度器利用博弈论方法构建一个MSDSEGM模型，并根据拟态防御架构的系统组成（例如异构体的数量、异构体的组成构件、执行体的数量）和分析器中的反馈信息，得到攻防双方的策略集合、策略概率集合和收益函数集合，以及NFV拟态防御架构的状态集合。

3）调度器根据演化博弈理论，利用复制动态方程方法对异构体动态调度策略进行不断优化调整。

基于演化博弈的最优动态调度策略选取算法描述如下：

算法1　基于演化博弈的最优动态调度策略选取算法

输入　攻防双方状态信息

输出　动态调度策略

1.while 攻防博弈未结束 do

2.MSDSEGM = (N, Φ, S, P, U);//初始化MSDSEGM

3.$\Phi = \{\Phi_1, \Phi_2, \cdots, \Phi_j\}$;//根据拟态防御实现架构的系统//组成得到状态集合

4.$S_A = \{S_{A1}, S_{A2}, \cdots, S_{Am}\}$;//构建攻击策略集合

5.$S_D = \{S_{D1}, S_{D2}, \cdots, S_{Dn}\}$;//构建拟态防御实现架构的//调度策略集合

6.$p = \{p_1, p_2, \cdots, p_m\}$;//攻击策略选取概率集合

7.$q = \{q_1, q_2, \cdots, q_n\}$;//调度策略选取概率集合

8.if Φ == Φ_k//根据拟态防御实现架构当前的状态求解//异构体调度策略

9.$U_{Dk}^{ij} = D_R - D_C - D_L - U_L$;//计算拟态防御实现架构收益

10.$U_{Dk}^{j} = \sum_{i=1}^{m} p_i U_{Dk}^{ij}$;//计算每个调度策略的期望收益

11.$U_{Dk} = \sum_{j=1}^{n} q_j U_{Dk}^{j}$;//计算拟态防御实现架构的平均收益

12.$D_k(q_j) = q_j(U_{Dk}^{j} - U_{Dk})$;//计算拟态防御实现架构的//复制动态方程

13.$q_j = q_j + \int D_k(q_j)dt$

14.$q = \{q_1, q_2, \cdots, q_n\}$;//输出异构体调度策略

15.end;

16.end;

4　仿真结果与分析

4.1　仿真设置

为验证MSDSEGM模型和最优调度策略选取算法的可行性和有效性，在MATLAB仿真平台上对博弈过程进行仿真与分析，仿真平台主机配置为Intel® Core™ i7-7700 CPU 3.60 GHz，8 GB RAM。为便于展示分析，本节仅对异构体的操作系统层进行异构化。设定异构体数量为4，操作系统分别采用Windows Server 2016、Ubuntu 18.04、CentOS和Windows Server 2008。由通用漏洞披露（Common Vulnerabilities & Exposures，CVE）[17]和通用漏洞评分系统（Common Vulnerability Scoring System，CVSS）[18]得到异构体操作系统配置参数及相关漏洞信息，如表2所示，其中CVSS分值反映了漏洞的严重程度，漏洞的得分最大为10，最小为0。CVSS得分在7~10的是高级漏洞，得分在4~6.9之间的是中级漏洞，得分在0~3.9的则是低级漏洞。Exploitability为CVSS中的可利用度指标，反映了漏洞受攻击的难易程度。从上述4个异构体中选择3个上线作为执行体，则NFV拟态防御架构共有$C_4^3=4$种状态$\Phi=\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$。其中：状态$\Phi_1$表示异构体$H_1$、$H_2$和$H_3$为在线执行体；状态$\Phi_2$表示异构体$H_1$、$H_2$和$H_4$为在线执行体；状态$\Phi_3$表示异构体$H_1$、$H_3$和$H_4$为在线执行体；状态$\Phi_4$表示异构体$H_2$、$H_3$和$H_4$为在线执行体。防御策略可以等价为系统各状态之间的转换，由此可以得到防御策略集合$S_D=\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$，同时可以得到攻击策略集合$S_A=\{vul1, vul2, vul3, vul4, vul5, vul6, vul7\}$。根据文献[19]中的漏洞利用成功概率计算方法，结合表2中的Exploitability指标信息得到各个漏洞的攻击成功概率：$\mu(vul1)=0.39$，$\mu(vul2)=0.39$，$\mu(vul3)=0.39$，$\mu(vul4)=0.34$，$\mu(vul5)=0.39$，$\mu(vul6)=0.34$，$\mu(vul7)=0.39$。参考文献[20]中的成本量化方法，取$A_C=5$，$O_C=5$，$N_C=5$，$F_L=30$，$D_R=10$。D_L根据漏洞等级从高到低分别取30、20、10。U_L根据异构体底层设备具体的负载量用0~10之间的数值表示，取$U_L(H_1)=5$，$U_L(H_2)=4$，$U_L(H_3)=7$，$U_L(H_4)=2$。

表2　操作系统配置参数及漏洞信息

Table 2　Operating system configuration parameters and vulnerability information

异构体	操作系统	漏洞信息	漏洞编号	CVSS	Exploitability
H_1	Windows Server 2016	CVE-2019-0543权限许可和访问控制问题漏洞	vul1	7.8	AV:L/AC:L/PR:L/UI:N
		CVE-2019-0554 Windows内核信息泄露漏洞	vul2	5.5	AV:L/AC:L/PR:L/UI:N
H_2	Ubuntu 18.04	CVE-2020-8832 Linux内核信息泄露漏洞	vul3	5.5	AV:L/AC:L/PR:L/UI:N
		CVE-2018-6557 Ubuntu后置链接漏洞	vul4	7.0	AV:L/AC:H/PR:L/UI:N
H_3	CentOS	CVE-2018-17977 Linux内核安全漏洞	vul5	4.4	AV:L/AC:L/PR:H/UI:N
		CVE-2019-16295 Control Web Panel跨站脚本漏洞	vul6	4.6	AV:L/AC:L/PR:L/UI:R
H_4	Windows Server 2008	CVE-2019-0543权限许可和访问控制问题漏洞	vul1	7.8	AV:L/AC:L/PR:L/UI:N
		CVE-2018-8589 Windows跨站脚本漏洞	vul7	7.8	AV:L/AC:L/PR:L/UI:N

4.2　仿真结果分析

本节假设攻防双方在初始时刻对对方的信息均不了解，分别以1/7和1/4的概率随机选择各攻击策略和调度策略，后期通过学习调整双方都趋向于选择收益较高的策略。基于该假设对不同状态下攻防双方的策略演化趋势进行分析。

图4和图5分别为Φ_1状态下攻击者和NFV拟态防御架构的策略演化趋势图。由于异构体H_1和H_4存在共模漏洞vul1，当H_1和H_4同为线上执行体时，攻击者就可能会利用漏洞vul1将H_1和H_4同时攻破，使NFV拟态防御架构输出错误的结果，从而获得更高的收益，因此vul1的选择概率不断增大，最终收敛到1。NFV拟

态防御架构可以通过分析器中得到的反馈信息，发现攻击者的策略演化趋势。为防止攻击者同时攻破 H_1 和 H_4，NFV拟态防御架构不断学习调整，优先选择 Φ_1 和 Φ_4 为下一调度状态，但由于NFV拟态防御架构此时就处于 Φ_1 状态，继续选择 Φ_1 可以避免调度操作带来的开销，因此 Φ_1 的选择概率逐渐增大。

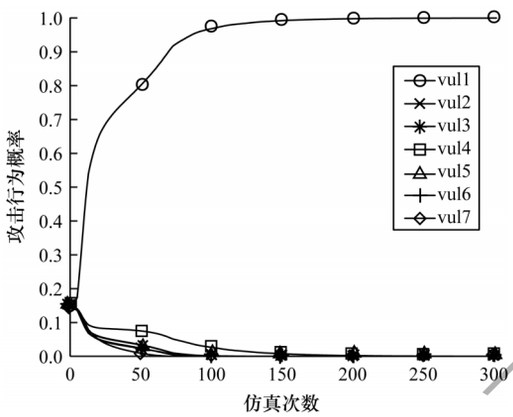

图4 Φ_1 状态下的攻击策略演化趋势

Fig.4 Evolution trend of attack strategy in Φ_1 state

图6和图7分别为 Φ_2 状态下攻击者和NFV拟态防御架构的策略演化趋势图。同理，攻击者为得到更高的收益，最终会趋向于选择利用漏洞vul1进行攻击。NFV拟态防御架构为避免攻击者同时攻破 H_1 和 H_4，会优先选择 Φ_1 和 Φ_4 为下一调度状态，但由于NFV拟态防御架构此时处于 Φ_2 状态，无论选择 Φ_1 还是 Φ_4 都无法避免调度操作带来的开销，但 Φ_1 状态表示异构体 H_1、H_2 和 H_3 为在线执行体，Φ_4 状态表示异构体 H_2、H_3 和 H_4 为在线执行体，由仿真参数设置可知，H_4 的底层设备负载量更低，底层负载影响着网络功能的运行效率，系统会优先调度底层负载较小的异构体上线作为执行体，因此 Φ_4 的选择概率不断增加。

图8和图9分别为 Φ_3 状态下攻击者和NFV拟态防御架构的策略演化趋势图，由于具体演化趋势分析与 Φ_2 状态一致，因此本文不再赘述。

图10和图11分别为 Φ_4 状态下攻击者和NFV拟态防御架构的策略演化趋势图。由图10可知，在博弈前期，攻击者利用vul1进行攻击得到的收益较大，因此vul1漏洞的概率逐渐提高。但是当图11中NFV拟态防御架构选择 Φ_4 状态的概率收敛到1时，根据数值计算可知此时攻击者利用漏洞vul1和漏洞vul7得到的收益相同且最大，vul1和vul7的选择概率都在增加，但是由于vul1的整体收益较大，因此vul1的选择概率增长更快。最终攻击者的策略会收敛到以0.890 7的概率利用vul1漏洞进行攻击，以0.109 3的概率利用vul4漏洞进行攻击。而NFV拟态防御架构为避免攻击者同时攻破 H_1 和 H_4 以及调度操作带来的开销，最终会趋向于保持 Φ_4 状态。

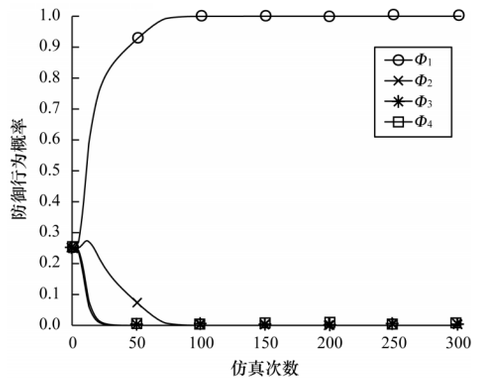

图5 Φ_1 状态下的防御策略演化趋势

Fig.5 Evolution trend of defense strategy in Φ_1 state

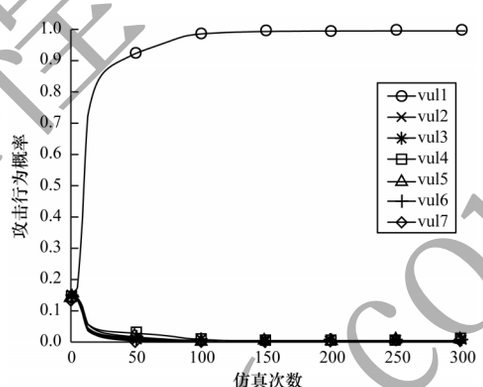

图6 Φ_2 状态下的攻击策略演化趋势

Fig.6 Evolution trend of attack strategy in Φ_2 state

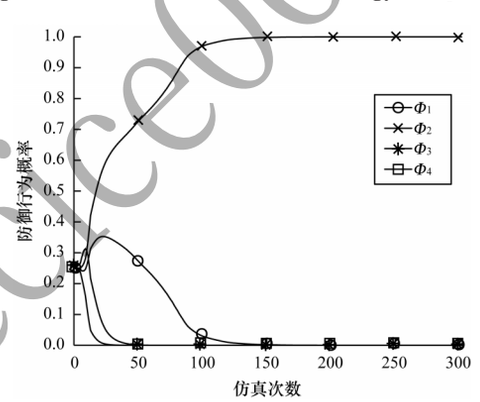

图7 Φ_2 状态下的防御策略演化趋势

Fig.7 Evolution trend of defense strategy in Φ_2 state

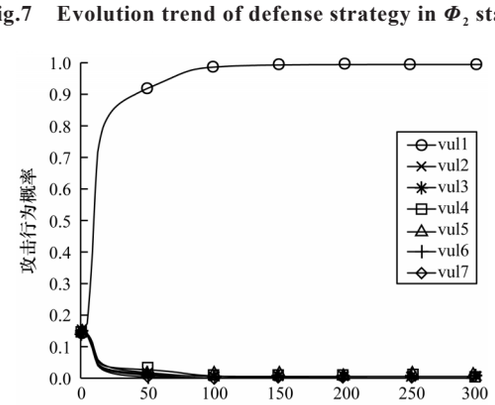

图8 Φ_3 状态下的攻击策略演化趋势

Fig.8 Evolution trend of attack strategy in Φ_3 state

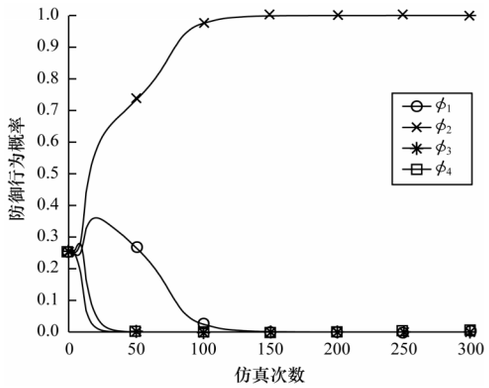

图9　Φ_3状态下的防御策略演化趋势

Fig.9　Evolution trend of defense strategy in Φ_3 state

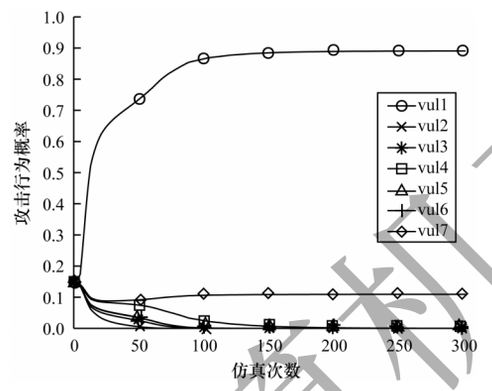

图10　Φ_4状态下的攻击策略演化趋势

Fig.10　Evolution trend of attack strategy in Φ_4 state

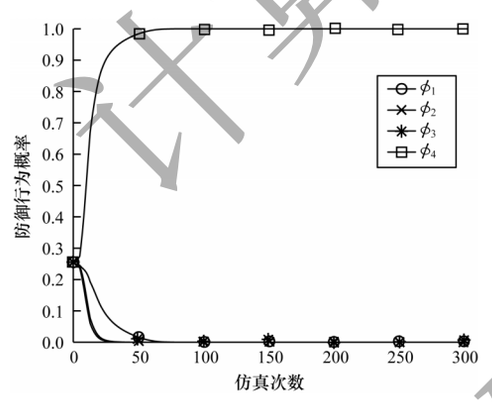

图11　Φ_4状态下的防御策略演化趋势

Fig.11　Evolution trend of defense strategy in Φ_4 state

为展现本文提出的调度策略所带来的安全收益和防御效能情况，将本文提出的基于演化博弈的调度策略（EG）与随机调度策略[21]（RANDOM）、基于执行体异构度的调度策略（HETERO）[6]以及基于Bayesian-Stackelberg博弈的调度策略[22]（BSG）中攻防双方的累计博弈收益和攻击者累计攻击成功的次数进行对比，如图12和图13所示。

图12为攻防累计收益对比图，其中：实线表示攻击者（Att）的累计收益；虚线表示防御者（Def）的累计收益。如图所示，EG和BSG都是基于博弈论的调度策略，NFV拟态防御架构可以根据实际的网络攻防环境找到有针对性的调度策略，实现防御收益的大幅提升，同时将攻击者的收益降低到负值。BSG在基于攻防双方都是完全理性的前提下，可以在得到更多的防御收益的同时进一步降低攻击者的收益。EG需要通过不断学习与调整逐渐找到最佳调度策略，防御收益相比BSG略低，但更符合实际网络中攻防双方的认知规律。HETERO和RANDOM虽然增加了系统的动态性，但是不具有针对性，防御收益相对较低。HETERO是基于执行体异构度的调度策略，异构度越大的执行体调度上线的概率就越大，一定程度上降低了共模漏洞存在的可能性，因此，HETERO的防御收益高于RANDOM。

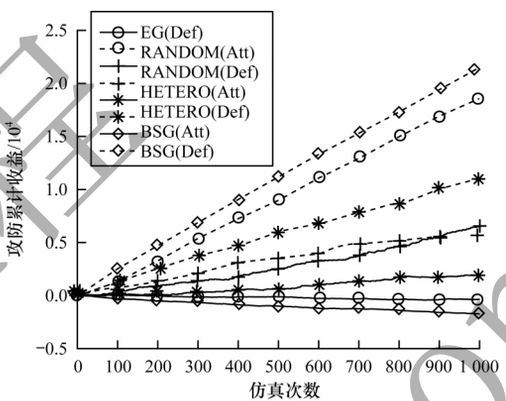

图12　攻防累计收益对比

Fig.12　Comparison of cumulative benefits of offense and defense

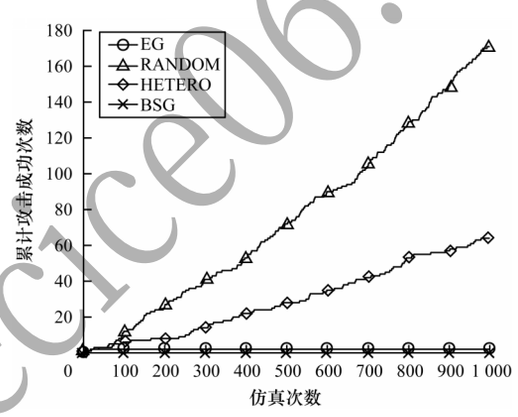

图13　攻击者累计攻击成功次数

Fig.13　Cumulative number of successful attacks by attacker

NFV拟态防御架构将静态空间的单一确定目标攻击难度增强为动态异构空间、多元目标协同一致攻击难度，难度等级呈非线性提升，使得攻击者利用共模漏洞攻击成功防御系统成为小概率事件。但为便于展示基于演化博弈的调度策略对系统防御效能的影响，本文将攻击者攻击成功NFV拟态防御架构的条件放宽为利用同一漏洞攻破半数以上的执行体，但不考虑是否输出相同的错误结果，并对不同调度策略下攻击者的累计攻击成功次数进行对比。如图13所示，经过1 000次仿真后，HETERO策略下攻击者的累计攻击成功次数为64次，RANDOM策略下攻击者的累计攻击成功次数为171次，同时由于HETERO和RANDOM没有充分利用裁决机制对异常执行体的感知定位功能对调度策略进行有针对性的调整优化，因此随着仿真

次数的增加,攻击者的累计攻击成功次数会持续增长。EG相比于HETERO和RANDOM提升了NFV拟态防御架构的安全性,在1 000次仿真过程中,攻击者仅在博弈初期攻击成功了2次,之后通过对分析器中的反馈信息不断学习,优化调度策略,使得累计攻击成功次数维持在该水平不变,有效提高了NFV拟态防御架构的防御效能。BSG基于攻防双方完全理性的前提,在博弈初期就可以找到最优调度策略,避免了攻击者成功攻击NFV拟态防御架构的情况,因此,攻击者的累计攻击成功次数始终为零。需要说明的是,图13的仿真结果是基于本文的参数设置条件下产生的,而在实际的网络环境中,异构体之间由于组成结构不同,共模漏洞存在概率较低,攻击者利用共模漏洞攻击成功的次数将小于该值。

5 结束语

现有拟态防御架构中的动态调度策略大多未利用裁决机制对异常执行体的定位感知能力进行优化调整,策略缺乏针对性和适应性。本文在NFV拟态防御架构中增加一个分析器,用于对历史裁决信息进行分析研究。在此基础上,利用分析器得到的反馈信息和演化博弈理论,从攻防双方的有限理性出发构建多状态动态调度演化博弈模型,采用复制动态方程和李雅普诺夫间接法对攻防双方的策略演化趋势和稳定性进行分析,提出一种基于演化博弈的最优动态调度策略选取算法。仿真结果表明,该算法可以根据拟态裁决的反馈结果,通过不断调整优化选择出具有针对性和适应性的调度策略,有效提升NFV拟态防御架构的安全收益和防御效能。本文对异构体的操作系统层做异构化处理,利用操作系统的具体漏洞信息对基于演化博弈的调度策略进行有效性分析。考虑到网络功能异构体由三层异构元素组成,下一步将对异构体的三层异构元素做异构化处理,并通过对漏洞进行分类来降低分析复杂度。

参考文献

[1] ETSI. Network Functions Virtualization(NFV):architectural framework[EB/OL]. (2014-12-01)[2021-03-24]. https://www.etsi.org/technologies/nfv.

[2] FIROOZJAEI M D, JEONG J P, KO H, et al. Security challenges with network functions virtualization[J]. Future Generation Computer Systems, 2016, 67(2): 315-324.

[3] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报, 2016, 1(4): 1-10.
WU J X. Research on cyber mimic defense[J]. Journal of Cyber Security, 2016, 1(4): 1-10. (in Chinese)

[4] 沈丛麒,陈双喜,吴春明. 基于信誉度与相异度的自适应拟态控制器研究[J]. 通信学报, 2018, 39(z2): 173-180.
SHEN C Q, CHEN S X, WU C M. Adaptive mimic defensive controller framework based on reputation and dissimilarity[J]. Journal on Communications, 2018, 39(z2): 173-180. (in Chinese)

[5] 顾泽宇,张兴明,林森杰. 基于安全策略的负载感知动态调度机制[J]. 计算机应用, 2017, 37(11): 3304-3310.
GU Z Y, ZHANG X M, LIN S J. Load-aware dynamic scheduling mechanism based on security strategies[J]. Journal of Computer Applications, 2017, 37(11): 3304-3310. (in Chinese)

[6] 刘勤让,林森杰,顾泽宇. 面向拟态安全防御的异构功能等价体调度算法[J]. 通信学报, 2018, 39(7): 192-202.
LIU Q R, LIN S J, GU Z Y. Heterogeneous redundancies scheduling algorithm for mimic security defense[J]. Journal on Communications, 2018, 39(7): 192-202. (in Chinese)

[7] 普黎明,刘树新,丁瑞浩,等. 面向拟态云服务的异构执行体调度算法[J]. 通信学报, 2020, 41(3): 17-24.
PU L M, LIU S X, DING R H, et al. Heterogeneous executor scheduling algorithm for mimic cloud service[J]. Journal on Communications, 2020, 41(3): 17-24. (in Chinese)

[8] 盖伟麟,辛丹,王璐,等. 态势感知中的数据融合和决策方法综述[J]. 计算机工程, 2014, 40(5): 21-25, 30.
GAI W L, XIN D, WANG L, et al. Overview of data fusion and decision-making methods in situational awareness[J]. Computer Engineering, 2014, 40(5): 21-25, 30. (in Chinese)

[9] FRIEDMAN D. Evolutionary game in economics[J]. Econometrica, 1991, 59(3): 637-666.

[10] 黄健明,张恒巍. 基于随机演化博弈模型的网络防御策略选取方法[J]. 电子学报, 2018, 46(9): 2222-2228.
HUANG J M, ZHANG H W. A method for selecting defense strategies based on stochastic evolutionary game model[J]. Acta Electronica Sinica, 2018, 46(9): 2222-2228. (in Chinese)

[11] 李鹤飞,黄新力,郑正奇. 基于软件定义网络的DDoS攻击检测方法及其应用[J]. 计算机工程, 2016, 42(2): 118-123.
LI H F, HUANG X L, ZHENG Z Q. DDoS attack detection method based on software-defined network and its application[J]. Computer Engineering, 2016, 42(2): 118-123. (in Chinese)

[12] 姜伟,方滨兴,田志宏,等. 基于攻防博弈模型的网络安全测评和最优主动防御[J]. 计算机学报, 2009, 32(4): 817-827.
JIANG W, FANG B X, TIAN Z H, et al. Evaluating network security and optimal active defense based on attack-defense game model[J]. Chinese Journal of Computers, 2009, 32(4): 817-827. (in Chinese)

[13] 海梅生,伊鹏,江逸茗. 基于服务质量与资源约束的服务链部署策略[J]. 计算机工程, 2019, 45(3): 54-59.
HAI M S, YI P, JIANG Y M. Service chain deployment strategy based on service quality and resource constraints[J]. Computer Engineering, 2019, 45(3): 54-59. (in Chinese)

[14] TAYLOR P D, JONKER L B. Evolutionarily stable strategies and game dynamics[J]. Mathematical Biosciences, 1978, 40(1/2): 145-156.

[15] SMITH J M, PRICE G R. The logic of animal conflict[J]. Nature, 1973, 246(5427): 15-18.

[16] ZAK S H. Systems and control[M]. Oxford, USA: Oxford University Press, 2003.

[17] National Vulnerability Database. Common vulnerabilities and exposures[EB/OL]. [2020-12-03]. https://nvd.nist.gov.

(下转第49页)

（上接第38页）

[18] OU X, SINGHAL A. The common vulnerability scoring system(CVSS)[EB/OL]. [2020-12-03]. http://forms. first. org/cvss/cvss_basic-2. 0. pdf.

[19] 高妮,高岭,贺毅岳,等. 基于贝叶斯攻击图的动态安全风险评估模型[J]. 四川大学学报(工程科学版),2016,48(1):111-118.
GAO N, GAO L, HE Y Y, et al. Dynamic security risk assessment model based on Bayesian attack graph[J]. Journal of Sichuan University (Engineering Science Edition), 2016, 48(1):111-118. (in Chinese)

[20] 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. (in Chinese)

[21] 邬江兴,李军飞,张帆,等. 一种异构功能等价体调度装置及方法:CN106161417A[P]. 2016-11-23.
WU J X, LI J F, ZHANG F, et al. A heterogeneous redundancies scheduling equipment and method:CN106161417A [P]. 2016-11-23.

[22] 王晓梅,杨文晗,张维,等. 基于BSG的拟态Web服务器调度策略研究[J]. 通信学报,2018,39(z2):112-120.
WANG X M, YANG W H, ZHANG W, et al. Research on scheduling strategy of mimic Web server based on BSG[J]. Journal on Communications, 2018, 39(z2):112-120. (in Chinese)

编辑　金胡考