

容器云中基于 Stackelberg 博弈的动态异构调度方法

曾威, 扈红超, 李凌书, 霍树民

(信息工程大学, 河南 郑州 450001)

摘要: 容器技术以其灵活、高效的特性促进了云计算的快速发展, 但同时引入了如同驻攻击、逃逸攻击和共模攻击等安全威胁。针对这些安全威胁, 提出一种容器云中基于 Stackelberg 博弈的动态异构式调度方法。首先, 构建异构镜像资源池以抑制云上基于共模漏洞的攻击扩散; 进而, 将攻防交互过程建模为 Stackelberg 博弈模型; 最后, 对攻防模型进行分析, 将系统调度问题建模为混合整数非线性规划问题以求解系统最优调度策略。实验证明, 所提方法能够提升云平台的防御效果, 降低系统防御开销。

关键词: 云安全; 容器调度; Stackelberg 博弈; 移动目标防御

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021063

Dynamic heterogeneous scheduling method based on Stackelberg game model in container cloud

ZENG Wei, HU Hongchao, LI Lingshu, HUO Shumin

Information Engineering University, Zhengzhou 450001, China

Abstract: Container technology promotes the rapid development of cloud computing with its flexible and efficient features, but it also introduces security threats such as co-resident attacks, escape attacks, and common mode attacks. In response to these security threats, a dynamic heterogeneous scheduling method based on Stackelberg game in the container cloud was proposed. First, a heterogeneous mirrored resource pool is constructed to suppress the spread of attacks based on common-mode vulnerabilities on the cloud. Then, the offensive and defense interaction process is modeled as a Stackelberg game model. Finally, the offensive and defensive model is analyzed, and the system scheduling problem is modeled as a mixed integer non-linear programming problem to solve the system's optimal scheduling strategy. Experiments show that the proposed method can improve the defense effect of the cloud platform and reduce the system defense overhead.

Keywords: cloud security, container scheduling, Stackelberg game, moving target defense

收稿日期: 2020-12-23; 修回日期: 2021-04-23

通信作者: 曾威, zengwei19970605@163.com

基金项目: 国家重点研发计划 (2018YFB0804004); 国家自然科学基金 (62002383)

Foundation Items: The National Key R&D Program of China (2018YFB0804004), The National Natural Science Foundation of China (62002383)

论文引用格式: 曾威, 扈红超, 李凌书, 等. 容器云中基于 Stackelberg 博弈的动态异构调度方法[J]. 网络与信息安全学报, 2021, 7(3): 95-104.

ZENG W, HU H C, LI L S, et al. Dynamic heterogeneous scheduling method based on Stackelberg game model in container cloud[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 95-104.

1 引言

云计算技术将底层物理设施虚拟化,并将大量计算任务分散到网络节点中,提供了强大的计算与存储服务。虚拟化技术对底层资源进行了抽象,屏蔽了底层逻辑和物理环境的差异,使大量网络节点仅需要关注提供服务本身,而不需要关注服务实现的具体细节。虚拟化技术在提供给用户细粒度服务的同时引入了一定的安全威胁。在多虚拟机或多容器共享虚拟化层和底层物理资源的环境下,数据集中存储和集中计算的运行模式会引入新的安全问题,造成数据丢失、数据泄露、越权访问和拒绝服务等安全威胁^[1]。

近年来,移动目标防御(MTD, moving target defense)技术已广泛应用于网络空间安全领域,并用来解决云环境中虚拟化层可能存在的安全威胁。MTD技术致力于构建一种动态、异构的网络环境,通过动态地改变系统暴露出来的攻击面,来增加攻击者的攻击难度。Jin等^[2]在容器云环境中建立了多维攻击图模型来形式化各种复杂的攻击场景,并提出了一种能够动态优化MTD策略的防御框架。Chowdhary等^[3]介绍了一个云环境下基于MTD技术的安全防御框架,并将攻防双方的交互行为建模为二人零和马尔可夫博弈,来识别和推理云环境中可能存在的基于软件漏洞的多阶段攻击。MTD技术与云计算技术相结合可以有效地提高云网络安全性,但防御者防御策略的选取往往具有一定的盲目性,盲目选择节点进行调度会额外增加系统负荷,且很难达到最优的防御效果。

因此,本文提出了一种容器云中基于Stackelberg博弈模型的动态异构式调度方法。在该博弈模型中,防御者(如云平台管理员)基于MTD技术的思想,通过分析攻击者的行为特征与收益函数,继而提出一种动态异构式系统最优调度策略,使用异构镜像容器对系统中可能遭受到攻击的容器进行实时动态调度,可以在不影响正常服务的同时,使攻击者基于先验知识的攻击行为失效。通过模型分析和实验证明,该动态异构式调度方法可以显著增加云系统的安全性,同时降低系统防御开销。

本文的贡献如下。

(1) 基于动态化、随机化和多样化的理念构建了异构镜像资源池,持续改变系统对外暴露的攻击面,以提升云平台的整体安全性。

(2) 提出了一种基于Stackelberg博弈模型的动态异构式调度方法,通过分析攻防策略和攻防行为特征,为云系统安全防御策略的制定提供了依据。

(3) 将容器云中系统调度问题建模成混合整数非线性规划问题进行求解,以最大化防御者收益为目标求解系统最优调度概率。

(4) 通过仿真实验,验证了该动态异构式调度方法在提升云平台安全性、降低系统防御开销上的有效性。

2 相关工作

常见的虚拟化技术包括虚拟机技术和容器技术^[4]。容器技术是一种内核轻量级的操作系统层虚拟化技术,实现了将系统资源分配到多个独立的组中,来平衡各个组之间资源调度的问题。目前,以Docker技术为代表的容器虚拟化技术逐渐发展成为云环境下一一种主流虚拟化技术^[5]。相比虚拟机技术,容器技术为云环境提供了更加高效、低成本的解决方案,但是容器技术的独立管理属性和内核共享机制会给系统带来新的安全威胁,其紧迫性和重要性不容忽视。

目前,业界对与虚拟化技术相关的安全问题进行了广泛的研究。传统上可以通过虚拟机安全加固^[6]和虚拟化层隔离^[7]来隔离虚拟机或容器的运行环境来抵御已知和未知的攻击,然而这些静态的防御方式具有一定的滞后性,难以适用于真实复杂的大型网络环境。基于动态性防御的理念,Azab等^[8]提出了一种容器云中基于MTD技术的逃逸机制,该机制使用基于主机的行为监测系统来指导容器实时迁移以防御未知的攻击。Huang等^[9]提出了一种基于MTD技术动态重定位策略来防御恶意邻居的攻击,仿真实验证明了该策略用来定位恶意容器的有效性。Sarkale等^[10]提出了一个具有最高特权容器的新安全层来解决容器进程直接调用底层主机内核而引发的安全问题,可以用来防御系统中潜在的威胁。杨爽等^[11]构建了

一种动态虚拟机调度模型,并在调度策略中引入遗传算法,有效解决了在虚拟机调度中出现负载失衡和较大动态迁移开销的问题。Tong 等^[12]提出了一种基于容器技术的安全云平台,实现了不同租户之间的安全隔离,并能够对租户行为进行监控,不足之处是仅使用短序列匹配(STIDE, sequence time-delay embedding)算法来进行威胁检测分析,可能被攻击者识别并刻意规避。动态平台技术(DPT, dynamic platform technology)是MTD技术在平台层面的一种具体方案,主要指系统通过动态调度软硬件平台的属性(如基础架构、操作系统、软件运行环境和平台数据格式等)以提高系统的安全性。蔡雨彤等^[13]对DPT防御攻击过程中瞬态效能进行量化分析,并设计了对比实验,验证DPT的有效性。Zhang 等^[14]提出了一种基于连续时间马尔可夫决策过程的多阶段攻击动态平台防御模型,可以基于系统奖励确定最佳的服务迁移时机。Sourour 等^[15]基于动态平台迁移系统,提出了主动和被动随机迁移两种防御策略,并验证了两种策略在动态平台防御系统中的有效性与局限性。云平台容器调度问题需要考虑较高的性能需求,如网络时延和服务质量等^[16],盲目选择节点进行调度会降低系统运行效率,且无法达到最优的防御效果。

由于攻防双方行为具有相互依存的特性,博弈论技术广泛应用于网络安全领域。博弈论可以充分考虑攻防双方的行为与决策信息,并用来对攻防双方交互过程进行建模,许多现有的成熟攻防博弈理论可以对网络安全防御提供新的解决思路。Xiao 等^[17]基于移动目标防御思想,设计了一种进化博弈模型的算法来解决虚拟机动态布局的问题,该算法能够成功解决虚拟机动态布局问题且能很大程度上优化其能耗。Hasan 等^[18]提出了一种使用 Nash 均衡的博弈论模型以检测同驻攻击,有效解决了虚拟化环境中恶意容器或虚拟机共存的问题。Lei 等^[19]设计了一种基于不完全信息的马尔可夫博弈论方法,用来指导并生成最优移动目标防御策略,拓展了云环境中具有单状态或不完整信息的最优策略选择。季新生等^[20]提出了一种面向安全的动态调度方法,将网络功能虚拟化服务链中的攻防过程建模为斯坦科尔伯格博

弈模型,通过动态地轮换服务链中可能存在安全威胁的节点,有效地提升了服务链的安全性。上述方法一定程度上解决了与虚拟化技术相关的安全问题,但值得注意的是,当前云环境普遍使用同构式容器提供相关服务,同构式容器难以防御针对同一漏洞的安全威胁,且云环境下容器的调度过程需要考虑其较高的性能需求。

针对上述问题,本文提出了一种容器云中基于 Stackelberg 博弈模型的动态异构式调度方法。实验证明,该动态异构式调度方法可以显著增加云系统的安全性,同时减少云系统防御开销。

3 威胁分析

云计算系统中多容器或多虚拟机尽管运行在独立的逻辑空间,但集中存储、集中运算的运行特性和资源共享模式依然会给云环境带来严重的安全风险,且软件层和硬件层中抽象出的虚拟化层容易引入新的安全威胁。本文主要研究与虚拟化技术相关云安全问题,同时避免云环境中基于共模漏洞的大规模攻击扩散。

图1展示了容器云环境中可能存在的安全威胁。云环境中与虚拟化技术相关的安全威胁主要包括同驻攻击和逃逸攻击等。同驻攻击主要指恶意用户在云环境中建立侧通道,并从位于同一服务器上的容器或虚拟机中提取私人信息。研究表明,同驻攻击逐渐成为当前云环境下数据泄露的主要原因。Han 等^[21]研究了如何改进虚拟机分配策略,不仅可以使攻击者难以与目标共同定位,还可以满足负载平衡和低功耗的要求。目前,云环境防御同驻攻击主要有两种方法。第一种方法是对物理主机的软件层和硬件层进行加固,但是这种防御方式不适用于防御未知类型的攻击。第二种方法是对容器应用进行动态迁移或轮换,这类方法能够有效地防御已知和未知的攻击,但这类方法需要综合考虑容器迁移算法和系统迁移成本。逃逸攻击是指攻击者利用虚拟机或者容器中软硬件层的漏洞进行攻击,从而扰乱宿主机上虚拟化层或底层的操作系统^[22]。云平台下虚拟化平台本身可能存在一定的安全漏洞,攻击者可以利用虚拟化平台的交互漏洞或容器本身存在的漏洞对目标系统发起特定类型的攻击,一旦攻击者攻

击成功,则整个系统内的其他租户会受到严重的安全威胁。Wu 等^[23]通过对比传统的攻击模型和云环境下的攻击场景,提出了一种基于自适应的访问控制模型来防止虚拟机逃逸攻击。

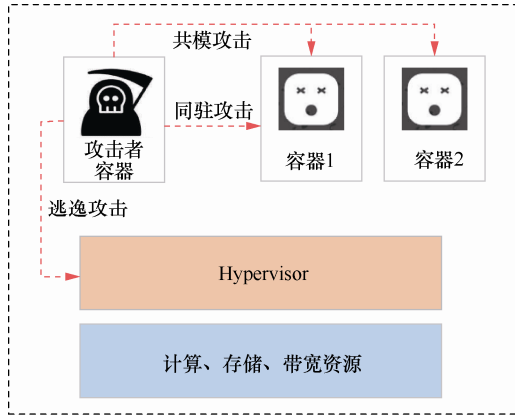


图1 容器云威胁模型
Figure 1 Container cloud threat model

共模攻击是指攻击者使用同一种攻击方式或不同的攻击方式同时对系统中的多个子系统进行攻击,如果目标系统采用基于大数表决机制或共识机制等策略时,攻击者通过使多个子系统呈现相同的输出状态可以实现攻击逃逸的目的。王禛鹏等^[24]基于拟态防御思想设计了一种具有内生安全特性的SDN控制层架构,降低了共模漏洞的安全威胁,并提出了基于博弈论模型的裁决机制,该裁决机制综合考虑了历史裁决信息,提升了系统裁决结果的正确性和整个系统的安全性。

4 模型设计与分析

本节首先提出了一种动态异构式容器调度模

型;然后将容器云环境下攻防双方的交互行为建模为Stackelberg博弈过程,并对博弈模型进行分析,最后对该博弈攻防双方收益进行量化与分析。

4.1 动态异构调度模型

为了保证云资源环境下系统和服务的正常运行,本节基于MTD技术的思想提出了一种动态异构式云资源池容器调度模型。图2展示了该动态异构式容器调度模型,包括云管理与调度层、虚拟化层资源、异构资源池和应用容器实体。系统中存在功能等价的容器异构镜像库,各异构容器实例之间具有不同的基础架构、操作系统、编程语言、函数库依赖等。在该动态异构调度模型中,当云管理与调度层接收到防火墙或入侵检测系统的安全告警信息时(如容器1存在被攻击的风险),云平台管理者依据一定的调度策略,从异构资源池中选择一种异构镜像容器(1a、1b或1c)对容器1进行实时动态轮换。本文主要使用的数学符号与其含义归纳见表1。

(1) 云资源池及执行实体特征向量表示

假设云资源池 V 中有 n 个容器,记作 $V=(V_1, V_2, \dots, V_n)$,容器集群中第 i 个容器记作 V_i 。对于容器应用执行实体 i ,对其 p 元特征进行抽象,如基础架构、操作系统、软件环境、函数库依赖等方面, p 元特征表示为 $(f_{i1}, f_{i2}, \dots, f_{ip})$ 。两容器应用执行实体的特征向量相同则代表两者同构,否则为异构,特征向量数值越大,容器执行体间异构度越大。在云资源系统上构建异构容器镜像库,使用 m_i 表示容器 i 对应的异构镜像数, $M=(m_1, m_2, \dots, m_n)$ 记作该云系统异构镜像数。容

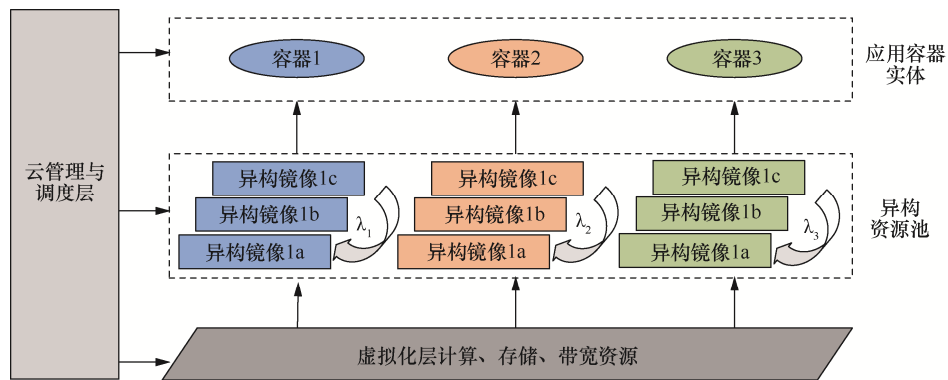


图2 动态异构式云资源池容器调度模型
Figure 2 Dynamic heterogeneous cloud resource pool container scheduling model

器应用在云环境中所占的价值权重记作 $\omega=(\omega_1, \omega_2, \dots, \omega_n)$ 。

表1 主要数学符号及含义
Table 1 Mathematical symbols and their meanings

数学符号	含义
V	云资源池
T	系统调度周期
M	容器异构资源池大小
ω	容器的价值权重
D	云资源池的异构度
λ	容器调度概率
p	异构容器元特征
k	攻击者的组数
s	攻击者最大攻击能力

(2) 异构度表示及安全性量化

异构技术广泛应用于 MTD 技术，异构度用来衡量多个执行体实例之间的差异程度。Wu 等^[25]使用了系统熵这一特征来度量整个系统的异构性。Garcia^[26]使用了共模漏洞数来衡量操作系统的异构度。张杰鑫等^[27]在量化生物多样性的基础上，提出了一种量化 Web 服务器异构度的方法，实验证明了该异构性量化方法的有效性。上述方法仅考虑了系统一维特征，不适用于云环境中容器高度虚拟化的情况，本文对容器执行体 p 元特征进行异构度量，使用 $D_i=(\bar{d}_{i1}, \bar{d}_{i2}, \dots, \bar{d}_{ip})$ 来衡量容器 i 与对应异构镜像容器的异构度差异。

(3) 基于异构度的调度策略

系统未监测到异常告警消息时，系统采取随机调度策略，以时间 T 为调度周期，调度模块以 $\lambda=(\lambda_1, \lambda_2, \dots, \lambda_n)$ 的概率对相应容器进行动态轮换；当系统检测到异常告警消息时，系统立即从异构容器资源池中选择一个异构应用容器对可能遭受攻击的容器进行实时动态轮换和清洗操作。

云资源环境中由于存在大量的未知漏洞和复杂的虚拟化边界，使云环境面临巨大的安全威胁。攻击者往往只需要攻破系统中一个容器，就可能造成云服务中断，甚至获得管理权限，相比之下，防御者需要对系统中每一个容器进行防御。针对这种攻防双方极度不平衡的态势，研究云环

境中系统最优调度策略具有重要的意义。

4.2 基于 Stackelberg 博弈的模型分析

在网络空间安全领域中，博弈论已广泛用于对云环境中攻防双方的交互行为进行建模^[28-29]。本节将容器云环境下攻防交互行为建模为双人 Stackelberg 博弈，将云平台管理者建模为领导者，攻击者建模为跟随者。云平台管理者对容器节点的调度策略看作防御者的防御策略，攻击者的攻击行为看作攻击者的攻击策略。图 3 结合概率性、学习型攻击者、动态防御者模型 (PLADD 模型, probabilistic, learning attacker, dynamic defender model) 展示了容器云环境下一次完整的攻防双方交互流程^[30]。通过将安全性抽象为资源的使用权，防御者首先获取资源的使用权，并使用一定的调度策略维持对资源的使用权，一旦攻击者获取对该资源的使用权，即代表攻击者攻击成功。

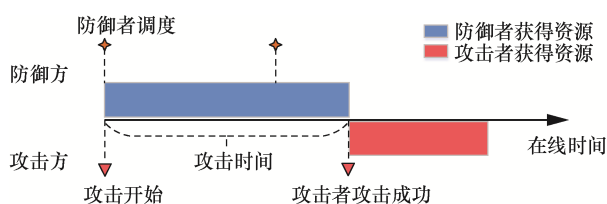


图3 攻防双方交互流程示意
Figure 3 Schematic of the interaction process between offense and defense

基于以上分析，对攻防博弈双方做出如下假设。

(1) 假设攻击者和防御者都是理性的，攻防双方都会采取最大化自身收益的策略。这里仅考虑攻击者针对云中的容器应用实施攻击，暂不考虑攻击物理层基础设施的情况。

(2) 假设系统中存在 k 组独立的攻击者，攻击者在某一个时间片可以选择攻击或者不攻击。攻击者在某一时刻最大攻击能力记作 s ，且随着时间的推进，攻击者能够不断获取系统防御知识，为下一步攻击决策提供依据。

(3) 攻击者攻破容器中任一 p 元特征即视为攻击者成功对目标系统进行攻击，攻击过程中攻击行为能够继续进行的概率与容器的异构度有关，并与其成正比。

由于攻防双方都是理性的，攻击者可以通过

多次探测继而推断出云资源池异构度 D 、云系统调度概率 λ 和各容器价值权重 ω 。因此, λ 、 D 、 ω 对于攻防双方来说是共同知识。

基于以上模型假设, 将博弈模型定义为三元组 $G(H, A, U)$ 。

(1) $H = \{\emptyset, H_1, H_2, \dots, H_n\}$, 表示云资源环境下防御者的防御策略集, H_i 表示防御者对容器集群中容器 i 进行动态轮换, $\emptyset$ 表示防御者不进行任何调度操作。

(2) $A = \{\emptyset, A_1, A_2, \dots, A_n\}$, 表示攻击者的攻击策略集, A_i 表示攻击者选择对容器 i 进行攻击, $\emptyset$ 表示攻击者不进行任何攻击操作。

(3) $U = (U_D, U_A)$ 表示为攻防双方的收益函数, 其中 U_D 为防御者的收益, U_A 为攻击者的收益。

4.3 攻防收益分析

在博弈过程中, 攻防策略的选择至关重要, 相比混合策略, 纯策略具有一定的限制性, 难以适用于复杂的云网络环境, 因此防御者通常采用混合策略以获得最大收益^[20], 该混合概率的分布即为容器集群中各节点的调度概率 λ 。通过分析不同策略下攻防双方的收益函数, 可推导出博弈过程中防御者的最优调度概率, 继而为防御者调度策略提供依据。攻防双方收益函数推论如下。

首先, 根据 4.2 节模型分析, 对攻击者的攻击成功率进行研究。在初始时刻, 攻击者未进行任何攻击操作, 此时攻击成功率为 0。随着时间的推进, 每个单位时间攻击者都会选择一种攻击方式, 防御者也会选择一种防御方式。攻击者获取到的防御知识随着时间增加, 参数 k 代表系统中存在 k 组攻击者。在单位时间片中, 攻击者攻击成功的概率与容器间的异构度均值有关, 为 $1 - \prod_{j=1}^p (\bar{d}_{ij})$, 攻击不成功的概率为 $\prod_{j=1}^p (\bar{d}_{ij})$, 则在 t 个单位时间片内, 攻击者首次对云平台攻击成功的条件概率 $P_i^k(t)$ 为

$$P_i^k(t) = \begin{cases} 0, & t=0 \\ \left(1 - \prod_{j=1}^p (\bar{d}_{ij})\right) \left(\prod_{j=1}^p (\bar{d}_{ij})\right)^{t-1}, & t=1, 2, \dots, n \end{cases} \quad (1)$$

则攻击者在 $t=n$ 时间段内攻击成功的总概率

P_i 为

$$\begin{aligned} P_i^k &= \sum_{t=1}^n P_i^k(t) = P_i^k(1) + P_i^k(2) + \dots + P_i^k(n) = \\ &C_n^0 \cdot \left(\lambda_i \cdot \prod_{j=1}^p (\bar{d}_{ij}) \right)^n + C_n^1 \cdot (1 - \lambda_i) \cdot \\ &\left(\lambda_i \cdot \prod_{j=1}^p (\bar{d}_{ij}) \right)^{n-1} + \dots + C_n^n \cdot (1 - \lambda_i)^n = \\ &\left(1 - \left(1 - \prod_{j=1}^p (\bar{d}_{ij}) \right) \cdot \lambda_i \right)^n \end{aligned} \quad (2)$$

若容器集群中存在 m 个容器, 第 i 个容器在容器集群中对应的价值权重为 ω_i , σ 为攻击者固定的攻击开销, 则攻击者在 $t=n$ 时间段内攻击收益函数为

$$U_A(n) = \sum_{i=1}^m (\omega_i \cdot P_i^k - \sigma) \quad (3)$$

在 Stackelberg 博弈中, 攻击者作为跟随者具有后手优势, 可以通过探测攻击分析出一定的防御知识, 如容器集群中各容器实例的价值权重 ω_i 、异构镜像库中容器间平均异构度 D 和云平台调度策略 λ 。基于以上信息, 理性的攻击者通过对不同策略下收益函数进行分析, 会倾向于选择使自身收益最大的攻击策略, 理性的攻击者在 $(\omega_i \cdot P_i^k - \sigma) > 0$ 时才会对容器 i 进行攻击, 在 $(\omega_i \cdot P_i^k - \sigma) \leq 0$ 时不会对容器 i 进行攻击操作。

防御者的防御开销由攻击者的攻击开销和容器实例调度开销两部分组成, 对防御开销取反即为防御者的防御收益。攻击者的开销可通过计算 $\sum_{i=1}^m (\omega_i \cdot P_i^k - \sigma)$ 获得, 容器实例调度开销

即为容器动态轮换时系统的切换开销, 可通过系统调度概率 λ 求出其期望值。系统调度开销与异构资源池中不同容器异构度正相关, 则在一轮博弈中, 防御者调度总开销为 $\sum_{i=1}^n \sum_{j=1}^p \lambda_i \bar{d}_{ij}$ 。在系统调

度开销和攻击开销中设置权重系数 μ , 在一轮博弈过程中, μ 取值越大, 即防御者调度开销在防御总开销中占的比重越大。此时防御者的防御收益为

$$U_D(\lambda, t) = -\sum_{k=1}^k \sum_{i=1}^m (\omega_i \cdot P_i^k - \sigma) - \mu \cdot t \cdot \sum_{i=1}^n \sum_{j=1}^p \lambda_i \bar{d}_{ij} \quad (4)$$

为了避免服务中断、多个容器同时轮换导致的系统产生瞬态效应等问题，这里仅考虑防御者在一个调度周期只能选择一种容器进行调度的情况。由于攻防双方都是理性的，攻击者会选择最大化自身收益的攻击策略，理性的防御者需要选择最优的混合策略以最小化防御开销。上述攻防收益直接求解比较困难，本节将容器云中系统调度问题转化成混合整数非线性规划问题进行求解，这里使用 CPLEX 求解器^[31]对其进行求解。目标函数与相关约束条件为

$$\begin{aligned} U_D(\lambda, t) &= -\sum_{k=1}^k \sum_{i=1}^m (\omega_i \cdot P_i^k - \sigma) - \mu \cdot t \cdot \sum_{i=1}^n \sum_{j=1}^p \lambda_i \bar{d}_{ij} \\ \text{s.t.} \quad &\forall i \in [1, n]: \lambda_i \in [0, 1], 0 \leq \sum_{i=1}^n \lambda_i \leq 1 \\ &\forall k \in [1, q], \forall i \in [1, n]: (\omega_i \cdot P_i^k - \sigma) \geq 0 \end{aligned} \quad (5)$$

5 仿真实验与分析

为评估该动态异构式容器调度方法的有效性，本节使用蒙特卡洛仿真，分析不同调度策略、异构度、调度周期和攻击能力下的攻防行为与攻防收益。

5.1 实验环境与参数设置

云资源环境包括 3 台 x86 服务器 (28 cores 2.00 GHz, 256 GB RAM, 10 GB NIC) 和 1 台 ARM 服务器 (16 cores 1.50 GHz, 32 GB RAM)，其中 1 台 x86 服务器为 master 控制节点，其余 3 台服务器为 node 计算节点。采用 Kubernetes 来管理云平台，Docker 镜像仓库存放异构镜像。仿真工具使用 Matlab R2018a。其余实验参数设置为 $w = (0.2, 0.3, 0.5)$ ， $\sigma = 0.01$ ， $\mu = 0.01$ 。

5.2 实验分析与讨论

5.2.1 攻击成功率与防御者收益

考虑云资源环境下防御者的 3 种调度策略 (最优调度策略、随机调度策略和静态调度策略) 随着时间的推进，攻击者能够不断获得防御者防

御知识。仿真实验设置 50 个单位时间片 (即在线时间)，每个单位时间片进行 100 次蒙特卡洛仿真。本节仅考虑容器间具有一维异构度 $D = (0.6, 0.8, 0.7)$ ，系统中存在一组攻击者且每组攻击者在一个调度周期只能攻击一种容器的情况。仿真实验对比了 3 种不同调度策略下攻击成功率和防御者收益。

如图 4 所示，攻击者攻击成功率随在线时间的增加而上升，相比之下动态调度策略能够显著降低攻击者攻击成功率，提高云平台的安全性。如图 5 所示，动态调度策略下系统的防御开销明显低于静态调度策略下的防御开销，且最优调度策略的收益优于随机调度策略。仿真实验证明了该最优调度策略在降低系统攻击成功率和防御开销方面的有效性。

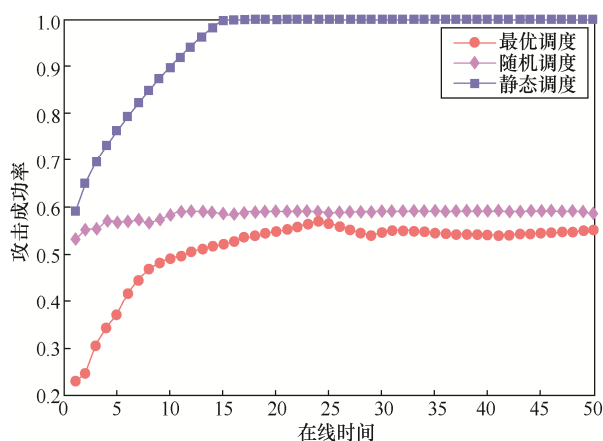


图4 不同调度策略下攻击成功率对比
Figure 4 Comparison of attack success rate under different scheduling strategies

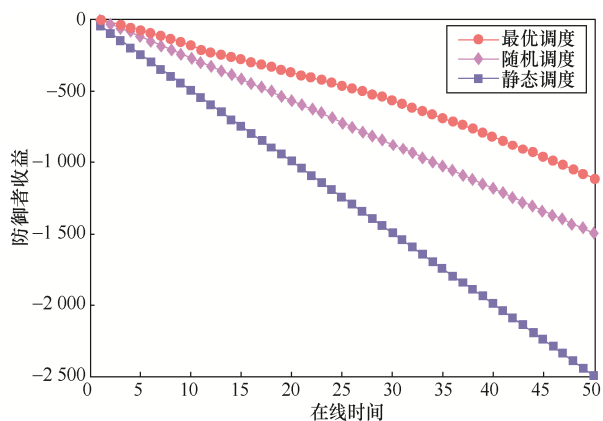


图5 不同调度策略下防御者收益对比
Figure5 Comparison of defender benefits under different scheduling strategies

5.2.2 异构维度对防御效果的影响

异构维度是衡量异构系统安全性的重要指标, 容器间异构度越大, 则攻击者的攻击难度越大, 系统的安全性越高。本节从基础架构、操作系统、软件环境 3 个不同特征来考虑不同应用容器间异构维度对系统防御效果的影响。一维异构度考虑基础架构之间的差异, 一维异构度参数设置为 $D=(0.6,0.8,0.7)$, 二维异构度考虑基础架构和操作系统之间的差异, 二维异构度参数设置为 $D=(\langle 0.6,0.6 \rangle, \langle 0.8,0.6 \rangle, \langle 0.7,0.6 \rangle)$, 三维异构度考虑考虑基础架构、操作系统和软件环境之间的差异, 三维异构度参数设置为 $D=(\langle 0.6,0.6,0.5 \rangle, \langle 0.8,0.6,0.5 \rangle, \langle 0.7,0.6,0.5 \rangle)$ 。

图 6 对比了不同异构维度下攻击者的攻击成功率, 容器间异构维度数越大, 攻击者对目标容器的攻击难度越大, 系统的防御效果越好。三维异构度下攻击者攻击成功率收敛于 0.45, 防御效果最好, 二维异构度下防御效果次之, 一维异构度下防御效果较差。

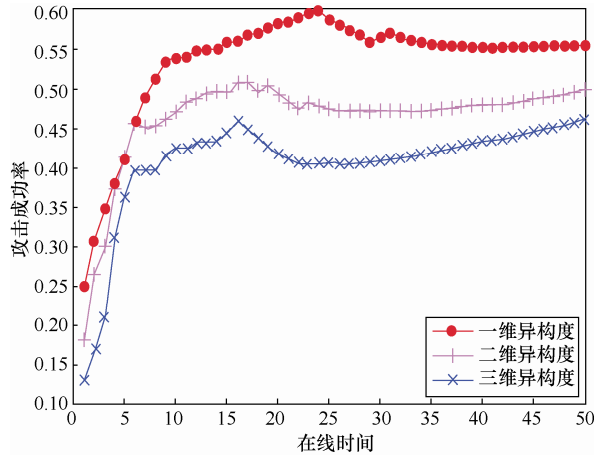


图 6 不同异构维度下攻击成功率对比
Figure 6 Comparison of attack success rates in different heterogeneous dimensions

5.2.3 调度周期对防御效果的影响

这里考虑最优调度策略下, 云平台系统调度周期 T 对系统防御效果的影响。改变防御者收益函数中权重系数 μ 的取值, 即可调整系统调度周期 T 。权重系数 μ 取值越大, 即代表系统调度开销所占比重大, 系统动态调度频率加快, 系统调度周期 T 越小。调整参数 μ 的取

值为 0.02、0.05 和 0.08, 对比不同调度周期对系统调度开销的影响。

不同调度周期下防御者平均开销对比如图 7 所示, 当权重系数 μ 取值较大时, 防御者系统调度周期 T 值减少, 此时系统动态调度的频率加快, 防御者调度开销占总开销的比重增大, 系统的平均开销逐渐增大。

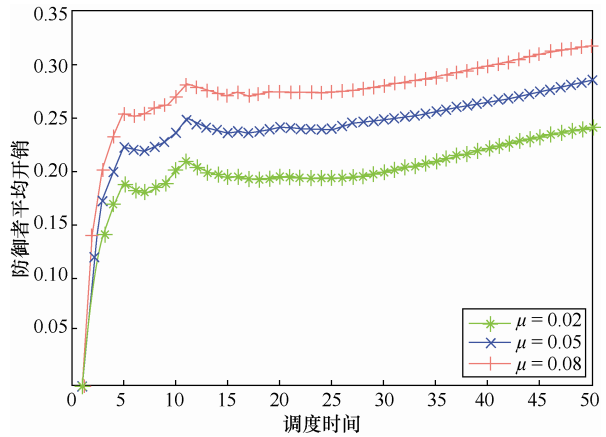
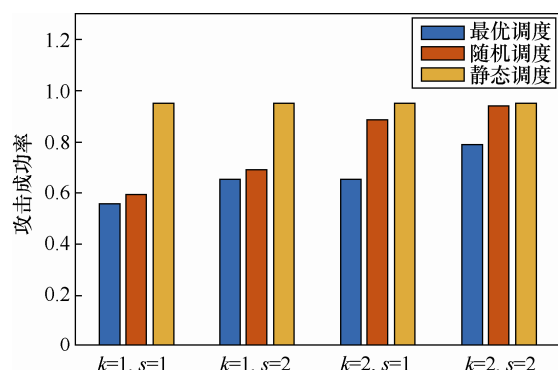


图 7 不同调度周期下防御者平均开销对比
Figure 7 Comparison of the average cost of defenders under different scheduling periods

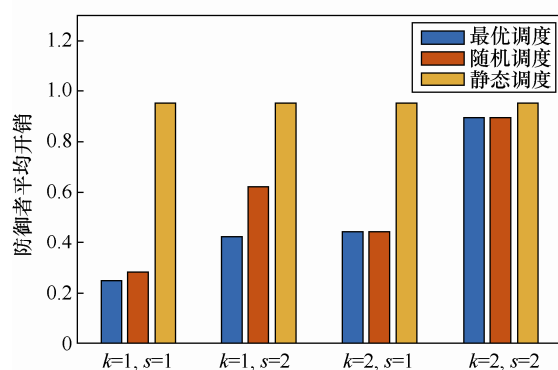
5.2.4 多攻击者对防御效果的影响

由于云环境的复杂性和不确定性, 系统中可能存在多个攻击者和多种攻击行为。本节考虑系统面临多个攻击者, 且单个攻击者能够同时攻击多个容器的情况。通过改变攻击者组数 k 和一组攻击者最大攻击能力 s , 观察不同攻击场景下攻击成功率和单个调度周期防御者的平均收益, 以确定多攻击者情况下系统的防御效果。由于防御者收益为负值, 这里取防御者收益绝对值记作防御者的防御开销。

多攻击者条件下的防御效果对比如图 8 所示, 在多攻击者且攻击能力加强的条件下, 系统采取最优调度策略时, 系统防御效果明显优于随机调度策略和静态调度策略, 此时攻击者攻击成功率最低且防御者平均开销最低。相比之下, 随机调度策略未考虑到攻防过程中的动态因素, 防御效果次之。静态调度策略由于未更改系统资源调用方式, 攻击者有足够时间获取有效的防御信息, 攻击成功率最高, 系统防御效果最差。



(a) 多攻击者条件下的攻击成功率



(b) 多攻击者条件下的防御者的平均开销

图8 多攻击者条件下防御效果对比

Figure8 Comparison of defense effects under multiple attackers

6 结束语

本文提出了一种基于 Stackelberg 博弈的动态异构式容器调度方法。该方法首先通过构建异构容器资源池,以降低云上基于大规模共模漏洞的攻击扩散;然后云平台依据一定的调度策略周期性地选择异构容器进行动态轮换和清洗操作以防御系统中可能存在的安全威胁;最后,将攻防交互过程建模为 Stackelberg 博弈,通过对攻防双方行为特征和攻防策略进行分析,以最大化防御者收益为目标求解出系统最优调度概率,以提升容器云环境的整体安全性。仿真实验验证了该动态异构式调度方法的有效性,能够在提高云平台安全性的同时,降低系统防御开销。

参考文献:

[1] JITHIN R, CHANDRAN P. Virtual machine isolation[C]//International Conference on Security in Computer Networks and Distributed Systems. 2014.

[2] JIN H, LI Z, ZOU D, et al. DSEOM: a framework for dynamic

security evaluation and optimization of MTD in container-based cloud[J]. IEEE Transactions on Dependable and Secure Computing, 2019, (99):1-1.

[3] CHOWDHARY A. Adaptive MTD security using Markov game modeling[C]//2019 International Conference on Computing, Networking and Communications (ICNC). 2019: 577-581.

[4] WANG J W, ZHANG X L, LI Q, et al. Network function virtualization technology: a survey[J]. Chinese Journal of Computers, 2019, 42(2): 415-436.

[5] LI S Y, LI Q, LI B. Research on isolation of container based on docker technology[J]. Computer engineering & Software, 2015, (4): 110-113.

[6] ABDELRAHEM O, BAHAA-ELDIN A M, TAHA A. Virtualization security: a survey[C]//International Conference on Computer Engineering & Systems. 2016: 32-40.

[7] ZHENG Z. Virtual machine security isolation and protection based on cloud platform[J]. China Computer & Communication, 2018, 417(23): 174-177.

[8] AZAB M. Toward smart moving target defense for Linux container resiliency[C]//2016 IEEE 41st Conference on Local Computer Networks (LCN). 2016: 619-622.

[9] HUANG R. RELOCATE: a container based moving target defense approach[C]//Proceedings of The 7th International Conference on Computer Engineering and Networks — PoS(CENet2017). 2017: 8.

[10] SARKALE V V, RAD P, LEE W. Secure cloud container: runtime behavior monitoring using most privileged container (MPC)[C]//IEEE International Conference on Cyber Security & Cloud Computing. 2017: 351-356.

[11] 杨爽. 基于改进遗传算法的动态虚拟机调度策略研究[D]. 哈尔滨: 哈尔滨工程大学. 2019.

YANG S. Research on dynamic virtual machine scheduling strategy based on improved genetic algorithm[D]. Harbin: Harbin Engineering University. 2019.

[12] TONG K, LIMING W, ZHEN X, et al. Design of a container-based security cloud computing platform[J]. e-Science Technology & Application, 2017, 8(1): 10-18.

[13] 蔡雨彤, 常晓林, 石禹, 等. 动态平台技术防御攻击的瞬态效能量化分析[J]. 信息安全学报, 2019, (4):59-67.

CAI Y T, CHANG X L, SHI Y, et al. Quantitative analysis of transient effectiveness of dynamic platform technology in defense against attacks[J]. Journal of Information Security, 2019, (4):59-67.

[14] ZHANG Y. Cost-effective migration-based dynamic platform defense technique: a CTMDP approach[J]. Peer-to-Peer Networking and Applications, 2021: 1-11.

[15] SOUROUT D. Platform moving target defense strategy based on trusted dynamic logical heterogeneity system[C]//Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science. 2019: 643-648.

[16] CAI Z Y, XIE X L. An improved container cloud resource scheduling strategy[C]//Proceedings of the 2019 4th International Conference on Intelligent Information Processing. 2019: 383-387.

[17] XIAO Z, JIANG J, ZHU Y, et al. A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory[J]. Journal of Systems & Software, 2015,

- 101(3):260-272.
- [18] HASAN M G M M, RAHMAN M A. Protection by detection: a signaling game approach to mitigate co-resident attacks in cloud[C]// Proc IEEE 10th Int Conf Cloud Comput (CLOUD) 2017: 552-559.
- [19] 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(1): 184-199.
- [20] 季新生, 徐水灵, 刘文彦, 等. 一种面向安全的虚拟网络功能动态异构调度方法[J]. 电子与信息学报, 2019, 41(10).
JI X S, XU S L, LIU W Y, et al. A security-oriented dynamic heterogeneous scheduling method of virtual network functions[J]. Journal of Electronics & Information Technology, 2019, 41(10).
- [21] HAN Y, CHAN J, ALPCAN T, et al. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing[J]. IEEE Transactions on Dependable & Secure Computing, 2017, 14(1):95-108.
- [22] ALNAIM A, ALWAKEEL A, FERNANDEZ E B. A misuse pattern for compromising VMs via virtual machine escape in NFV[C]//The 14th International Conference. 2019: 1-6.
- [23] WU J. An access control model for preventing virtual machine escape attack[J]. Future Internet, 2017, 9(2): 20.
- [24] 王祺鹏. 拟态网络操作系统调度与裁决机制研究及实现[D]. 郑州: 信息工程大学, 2017.
WANG Z P. Research and implementation of mimic network operating system scheduling and adjudication mechanism[D]. Zhengzhou: Information Engineering University, 2017.
- [25] WU M, GUAN H, ZANG B, et al. POSTER: quantitative security assessment method based on entropy for moving target defense[J]. ACM SIGPLAN Notices, 2017, 52(8): 457-458.
- [26] GARCIA M. Analysis of operating system diversity for intrusion tolerance[J]. Software-Practice and Experience, 2014, 44(6): 735-770.
- [27] 张杰鑫, 庞建民, 张铮. 拟态构造的 Web 服务器异构性量化方法[J]. 软件学报, 2020(2): 564-577.
ZHANG J X, PANG J M, ZHANG Z. Quantification method for heterogeneity on Web server with mimic construction[J]. Journal of Software, 2020(2): 564-577.
- [28] PARUCHURI P, PEARCE J P, TAMBLE M, et al. An efficient heuristic approach for security against multiple adversaries[C]// Proc 6th Int Joint Conf Auton Agents Multiagent Syst. 2007: 1-8.
- [29] PARUCHURI P, PEARCE J P, MARECKI J, et al. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games[C]//Proc 7th Int Joint Conf Auton Agents Multiagent Syst. 2008: 895-902.
- [30] PLADD: deterring attacks on cyber systems and moving target defense[P]. 2017.
- [31] IBM Ins. CPLEX optimization studio 12.10[EB]. 2019

[作者简介]



曾威 (1997-), 男, 河南信阳人, 信息工程大学硕士生, 主要研究方向为移动目标防御和拟态安全防御。



扈红超 (1982-), 男, 河南商丘人, 信息工程大学研究员, 主要研究方向为云计算和网络安全。



李凌书 (1992-), 男, 湖北恩施人, 信息工程大学博士生, 主要研究方向为拟态安全防御和网络欺骗。



霍树民 (1985-), 男, 山西长治人, 信息工程大学副研究员, 主要研究方向为网络空间安全和人工智能安全。