基于平均任务失效时间和任务完成时间的移动目标防御技术效能量化分析

陈 志[1], 常晓林[1], 杨润垲[1], 韩 臻[1]

[1]北京交通大学计算机与信息技术学院 智能交通数据安全与隐私保护技术北京市重点实验室 北京 中国 100044

摘要 移动目标防御(MTD)通过不断的变换系统攻击面, 增加系统的不确定性, 限制攻击者探索系统的弱点, 从而有效降低系统被攻击的可能。随着信息系统的发展和新漏洞的不断增加, 且传统防御方法存在天然的时间劣势无法抵御新型攻击, MTD 越来越受到关注。本文旨在量化分析 MTD 环境中关键任务的安全性和性能。本文使用攻击者攻击成功概率作为系统安全性评估指标。使用长期任务平均失效时间(MTTF)和短期任务平均完成时间(JCT)作为评估 MTD 系统性能指标。本文中的系统由多个物理机(PM)组成, 每个 PM 中托管一个虚拟化环境(容器或虚拟机), 关键任务运行在虚拟化环境中并受攻击者影响。系统中部署了基于动态平台技术(DPT)的 MTD 来减少攻击行为对任务运行的影响, 动态平台技术通过将任务的运行主动划分为多个阶段, 并且通过随机选择每一阶段的运行平台的方式降低任务被攻击者发现和破坏的概率。本文我们使用马尔可夫模型抽象表示系统中的任务运行行为, 并在此基础上量化分析 MTD 防御效能。相对于现有的分析模型要求所有时间均服从指数分布, 我们的方法允许任务阶段运行时间和迁移时间服从任意分布。本文分别以长期任务 MTTF 和短期任务 JCT 为评估指标并给出了对应的解析解公式。同时, 我们使用仿真实验验证了我们的模型和公式的准确性。此外, 本文还提出了一个 MTD 系统的总成本预测方案, 用来帮助管理员更有效合理的部署防御系统。

关键词 动态平台技术; 移动目标防御; 马尔可夫链; 平均失效时间; 性能

中图法分类号 TP309 DOI 号 10.19363/J.cnki.cn10-1380/tn.2022.03.02

Effectiveness and Performance Analysis of Moving Target Defense System: MTTF and Job Completion Time Perspectives

CHEN Zhi[1], CHANG Xiaolin[1], YANG Runkai[1], HAN Zhen[1]

[1] Beijing key Laboratory of Intelligent Transporation Data Security and Pavacy Protection Technology School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

Abstract Moving Target Defense (MTD) technology protects a target system by creating asymmetric uncertainty of the target system to confuse the adversaries and increase the complexity of attacks. It has been gaining more and more attention with the massive growth of vulnerabilities and the widespread deployment of critical network services and traditional defense technology has a natural time disadvantage. This paper aims to quantitatively analyze both the effectiveness and performance of an MTD enabled system. We use the probability of successful attack as the security metric. As for performance metrics, Mean Time To Failure (MTTF) and Job Completion Time (JCT) are used to evaluate long-term and short-term running job in the MTD protected system, respectively. The system in this paper consists of multiple Physical Machines (PM) and each PM hosts a virtualized environment (containers or virtual machines), each of which can run a critical job under attack from adversaries. It applies Dynamic Platform Technique (DPT), a kind of MTD implementation techniques, to reduce the impact of attacks on job performance. The DPT actively divides the running process of a critical job into multiple stages, and randomly selects the operating platform of each stage to reduce the probability of the job being discovered and destroyed by the attackers. We propose a stochastic model which captures job execution behaviours in the system. Our model-based approach allows both job residency/execution time at a PM and job migration time to be generally distributed which releases the exponential distributed time assumption in other related analysis models. We derive the closed-form solutions of job MTTF (for long-term jobs) and JCT (for short-term jobs) which are the main evaluation metrics in this paper. Simulation experiments are carried out to validate our model and formulas. Moreover, a formula is proposed to predict the total cost of the system, which helps administrators manage the system effectively.

Key words dynamic platform technique; moving target defense; Markov chain; mean time to failure; performance

通讯作者: 常晓林, 博士, 教授, Email: xlchang@bjtu.edu.cn。

本课题得到国家自然科学基金(No. U1836105)资助。

收稿日期: 2020-12-22; 修改日期: 2021-03-26; 定稿日期: 2022-01-10

1 引言

随着用户设备和网络规模的不断扩大, 原本相对独立自主的信息系统之间的关联变得越来越紧密, 这使得修改更新网络系统的配置或者参数信息的代价越来越大, 导致大规模的系统变得越来越固化。不幸的是, 信息系统的这种相对固化的静态属性为攻击者提供了天然的时间优势, 攻击者在发起攻击之前有足够的时间来探索系统的弱点和漏洞, 而防御者只能处于被动等待攻击的状态。尽管研究人员在过去几年中解决了很多信息系统安全防护问题, 但是新的漏洞仍然在不断的出现[1-2]。并且为产业和社会带来巨大的经济损失[3]。而传统的防御机制(例如恶意行为检测和识别)很难防御这些新出现的攻击方式。为了弥补传统防御方法的天然缺陷, 研究人员提出了移动目标防御(MTD)。作为一种主动保护技术, MTD 通过不停的改变系统某些参数和属性来增加攻击者探索系统的难度, 提高攻击者的攻击成本以保护目标系统[4]。动态平台保护技术(DPT)[5]是 MTD 技术的一种具体实现方式, 主要通过动态更改计算平台的属性, 包括硬件或者操作系统(OS)等, 以增加攻击的复杂性[6]。如图 1 所示, 有一个关键任务运行在受 DPT 保护的系统中。系统为中有 5 台物理机(PM)用来构成动态平台。关键任务将在这些 PM 之间动态迁移, 从而使攻击者无法轻松地发现和攻击破坏关键任务, 因为攻击者很难知道哪个 PM 正在作为目标任务当前运行的底层平台。尽管 MTD 的使用提高了系统和服务的安全性, 但同时也会降低服务可用性和运行性能。因此, 有必要对其防御效果和性能进行定量分析。当前关于 MTD 的研究很多探索了如何使用 MTD 技术来解决各种安全问题[7-15], 或者评估了部署 MTD 技术的系统的安全性[16-20]。也有一些工作分析了 MTD 技术对于系统性能的影响, 但是评估模型中假设所有时间间隔都呈指数分布[21], 或者在特定威胁模型场景下评估了 MTD 系统的可靠性[6]。

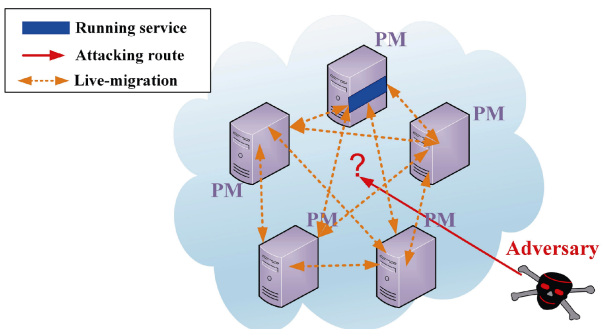

图 1 动态平台技术

Figure 1 The principle of DPT

本文旨在分析一个运行在部署了移动目标防御技术的系统中的关键任务的平均失效时间(MTTF)或(Job Completion Time, JCT), 并以此评估移动目标防御系统的防御效果和系统性能。本文所考虑系统中使用基于 DPT 的 MTD 技术来减少攻击对关键任务的影响。本文中我们将量化评估的目标分成长期任务和短期任务两种。长期任务意味着任务作为一个服务始终在系统中运行。短期任务意味着需要固定的时间才能完成执行获得期望结果。平均失效时间(MTTF)是评估任务或者服务的长期运行能力的一个重要的可靠性指标, 本文中我们只考虑攻击者的攻击破坏导致的任务失效。因此在本中我们使用 MTTF 作为衡量动态平台技术对于长期任务防御效能的量化指标。而对于一个脆弱系统中运行的短期任务, 平均任务完成时间则更加适合用来衡量防御技术或者策略的效果。本文考虑的系统中存在多个 PM, 每个 PM 托管一个可进行实时迁移的虚拟环境(Virtual Environment, VE, 即 Docker 容器[22]或虚拟机[23])。关键任务在 VE 中运行, 为了实现 MTD 防御技术, 任务的运行被分成多个阶段(每一阶段的运行时间本文称之为阶段时间或周期时间), 并且可以通过动态迁移技术在不同 PM 之间迁移以达到任务的持续运行。在关键任务运行的过程中, 存在多个攻击者独立随机地攻击系统, 并且尽可能的破坏目标任务的运行, 以降低 MTTF 或尽可能推迟总的 JCT。

构建分析模型是有效评估防御效能和任务运行可靠性的方法之一。对于移动目标防御技术的量化分析, 目前已经有很多研究人员提出了不同的量化评估模型方案。但是当前的评估模型主要集中在评估移动目标防御机制的防御效果上, 忽视了被保护任务自身运行的情况。本文我们提出了一种新颖的能够反映任务运行和迁移行为的分析模型。并且分别对于长期任务和短期任务给出了平均失效时间和平均完成时间的解析解以及不同参数的实验分析。此外, 我们提出了系统的总成本开销的分析预测方法。为了验证我们模型和解析解的准确性, 我们还通过仿真实验的方式, 利用仿真解与解析解相互对比佐证。利用我们的模型和实验分析, 系统管理员或者防御方在部署动态平台方案的时候可以更合理的配置防御系统, 选择最优的系统配置参数。本文是我们先前工作(发表在 IEEE GlobeCom 2019)的延伸扩展[24]。我们将在第二节相关部分给出本文与之前工作的详细区别。本文的主要贡献如下:

• 本文提出了一个移动目标防御效能量化分析模型从被保护平台中任务运行的角度分析了移动目标防御技术的防御有效性以及对开销成本和系统性能的影响;

• 本文提出的量化分析模型分别以无穷状态马尔可夫链和吸收态马尔可夫链的形式刻画了动态平台中长期运行任务和有固定时间需求的短期任务的运行逻辑;

• 基于我们的模型, 我们分别推导了长期任务的平均失效时间和短期任务的平均完成时间的闭合解公式并以此为指标评估了动态平台技术的防御效能。此外, 我们进行了仿真实验, 并利用评估指标的仿真解验证了我们闭合解公式的准确性。最后, 我们的实验结果展示了不同配置参数对于系统整体效能的影响。

本文的其余部分安排如下。第 2 节介绍了相关工作; 第 3 节给出了具体的系统描述, 模型介绍和相关评估指标计算公式; 第 4 节展示了我们的数值分析以及相关的实验结果; 第 5 节介绍了本文的结论和未来工作展望。

2 相关工作

MTD 技术近年来引起了学术界和工业界的广泛关注。很多学者和研究组织对 MTD 技术进行了一系列的探索和研究。总体上, MTD 技术可分为以下几类: 动态网络, 动态平台, 动态运行时环境, 动态软件和动态数据[25]。在本文中, 我们重点介绍基于 DPT 的 MTD 技术相关研究。

技术使用上, MTD 技术被用来防御各种层面的攻击, 例如 SQL 注入攻击, 渗漏攻击, DDoS 攻击和 APT 攻击等等, 针对这些攻击方式学术界也在一直不断的提出适应各种场景的 MTD 技术。Steinberger 等[7]提出了一种基于 MTD 技术的 DDoS 防御解决方案, 并对提出的解决方案进行了定性和定量评估。Albanese 等人[8]使用 MTD 技术有效缓解了僵尸网络的问题。Sengupta 等人[9]提出了一种基于 MTD 的方法来提高 Web 应用程序的安全性。Tian 等人[10]提出了一种隐藏式 MTD 保护智能电网的方法, 他们的方法能够在避免敌手探索防御安全策略的同时而不会影响智能电网的性能。Zeitz 等人[11-12] 提出了一种基于 IPv6 场景的微型移动目标防御技术, 并详细分析研究了其功耗开销和防御策略的安全优势。Lakshminarayana 等人[13]提出了一种在智能电网环境中部署的 MTD 的方案, 并给出了相应的有效性的量化评估方法。他们分析了防御有效性和成本之间的权衡。并使用仿真实验验证了其评估结果的准确性。同样, Zhang 等人[14]也使用 MTD 技术来保护智能电网系统。不同之处在于它们的方案仅针对虚假数据注入攻击。Griffioen 等人[15]提出了一种混合 MTD 方案, 用于检测和隔离对物理信息系统的攻击。他们的工作演示了如何将不同的 MTD 技术结合使用。与我们的工作不同, 以上这些研究[7-15]专注于开发新的 MTD 方案或如何应用 MTD 技术来解决特定的安全问题。

除上文介绍的 MTD 技术应用开发研究之外, 还有很多研究旨在分析 MTD 系统的有效性和性能。他们中的一些人使用博弈论方法来研究MTD的防御优势。Zhou 等人[16]将三边博弈与马尔可夫决策过程相结合, 以研究基于变换的 MTD 方案的性能和防御成本之间的权衡。Huang 等人[17]在五个评估指标和一个评估框架下系统地评估了混合MTD的方案的安全优势和成本。Stout 等人[18]开发了一种在真实网络攻击环境中对 MTD 系统进行评估的实验平台分析方法。Anderson 等人[19]提出了两种方法来评估 DPT 对攻击成功概率的影响, 他们展示如何最有效的配置 DPT 系统使攻击成功概率最小。Alavizadeh 等人[20]提出了一个三种 MTD 技术的比较评估方案, 包括基于变换、基于冗余以及变换和冗余的相组合的 MTD 技术, 他们的方案主要目的是评估不同 MTD 技术对提高系统安全性和可靠性的能力区别。以上[16-20]工作通过分析攻击成功概率, 系统风险和可靠性等来评估各种类型的 MTD 技术的防御能力和效果。尽管这些研究工作各自具有优势, 但这些研究都主要关注于MTD的防御效果而忽略了其对于系统性能的影响。本文介绍的工作旨在分析基于 DPT 的 MTD 技术对系统中的运行的关键任务的 MTTF 和平均完成时间JCT的影响并以此为指标综合评估MTD的防御效能。

效能量化分析可以用于指导具体MTD技术的设计以及比较不同技术优缺点。对于 MTD 的量化分析可以采用测试平台、理论分析或者使用随机模型等方法。测试平台方案无法支持大规模的分析场景, 此外, 由于目前没有公认的 MTD 理论系统导致理论分析方案通常只适用于分析自身提出的具体 MTD 机制。因此现有很多研究工作采用随机模型方法对 MTD 技术的性能和可靠性进行了定量分析。Connell 等人[26-27]评估了使用基于系统重配置方案的MTD技术对系统中任务运行性能的影响。他们的研究专注于分析 MTD 对任务请求的平均等待时间的影响, 属于单纯的性能分析。而本文考虑的是 MTD 技术对系

统中运行的关键任务 MTTF 的影响, 同时考虑防御效果和防御性能。Okhravi 等人[6]开发了一个基于离散时间马尔可夫链的模型, 以分析预测 DPT 系统中服务从开始持续运行直到被攻击者破坏的时间。Chang 等人[21, 28]使用基于随机 Petri 网的建模分析方法来研究 MTD 对受保护平台上任务运行的影响。并在文献[29]中进一步开发了一种可以分析大型系统的建模方法。以上工作与本文的研究目标比较类似, 都是从受保护平台上任务运行情况的角度对 MTD 技术进行效能量化分析。他们的分析模型同样将关键任务的运行分割成多个阶段, 每一阶段随机选择一个任务运行平台, 最后通过模型计算得到任务完成时间和攻击成功的概率并以此为指标分析 MTD 防御效能。以上这些分析模型有一个共同的缺点是需要假设系统中的所有时间分段都服从指数分布。而本文提出的分析方案允许任务在不同平台不同阶段的运行时间和迁移时间为任意分布。本文是我们先前工作[24] (发表在 IEEE GlobeCom 2019)的延伸扩展。在之前的工作中, 我们展示了 MTTF 的初步分析过程。在本文中, 我们将从以下两个方面对之前工作进行改进:

• 之前分析方案中, 我们只考虑了在一个任务运行周期内只能有一次攻击能够命中任务所在平台的情况。但是实际上同一运行周期内也可能发生多个以任务所在平台为目标的攻击, 即使只造成一次破坏。本文中我们考虑了一个运行周期内所有可能命中任务所在平台的攻击情况, 提出了一个更加精确的 MTTF 计算方法。

• 我们重建了之前的分析模型使其既能支持对长期任务 MTTF 的分析也能支持对短期任务 JCT 的分析, 我们分别给出了两个指标闭合解的详细计算推导过程。

3 系统描述和分析模型

本节首先对要研究的系统进行介绍, 然后介绍本文考虑的攻击模型。本文提出的分析模型见第 3.3 节。之后我们给出本文所考虑的指标的解析解计算公式。本文所使用的变量的定义以及部分变量缺省值如表 1 所示。

表 1 变量定义

Table 1 Variable definition

变量	定义	期望
N	物理机数量	6
X	攻击者数量	—
A	攻击者攻击一个物理机所需要的平均时间	12 d
A'	目标任务到达某物理机后剩余的攻击平均时间	—
$H_{s,m}$	攻击者 s 第 m 次攻击所花费的时间	—
A_j	攻击者 j 攻击一个物理机所需要的平均时间	—
R	任务在每台物理机上的停留时间(阶段时间或周期时间)	5 d
R_j	任务迁移 j 次之后在下一台物理机上的停留时间	—
L	任务动态迁移所消耗的平均时间	30 min
L_j	任务第 j 次热迁移所消耗的时间	—
θ	X 个攻击者中某一个最先成功攻击物理机的速率	—
p	关键任务所在物理机不被攻击的概率	—
$failure\_time$	从任务最后一次迁移完成到被攻击破坏的间隔时间	—
I	R 时间间隔内发生的攻击次数	—
K	任务被攻击之前的迁移次数	—
$MTTF_k$	已知任务能够成功迁移 k 次情况下的平均失效时间	—
TC	系统每运行 1000 d 的成本开销	—
$loss$	任务失败一次的损失	\$100000
$C_{migration}$	迁移一次的成本开销	\$10
$C_{security}$	延长任务被攻击时间一天的额外防御成本	\$2000
C_{PM}	单个虚拟机的成本开销	\$1000

3.1 系统描述

如图 1 所示, 本文所考虑的是一个典型的基于动态平台技术的移动目标防御系统。系统中包含多个物理机, 每个物理机上可以运行独立的虚拟环境, 虚拟环境根据具体虚拟化技术的不同, 可以是虚拟机(VM)也可以是容器(docker)。任务运行在虚拟环境当中, 并且可以通过动态迁移技术选择跳转到不同的物理机继续运行任务。本文中我们根据任务的运行需求将任务分成长期任务和短期任务两种。长期任务以服务类程序为代表需要长期在系统中运行。短期任务代表需要固定执行时间完成运行获得目标结果的任务。此外, 系统中存在多个攻击者持续通过对物理机发起攻击的方式来干扰任务的正常运行。为保护任务的正常运行, 系统采用基于动态平台的移动目标防御技术。具体实施起来如下, 首先在任务开始时, 系统会随机选择一台物理机作为底层硬件环境部署虚拟环境, 然后任务在虚拟环境中开始运行。运行一段预先设置好的时间之后, 系统重新选择一台物理机并启动虚拟环境的动态迁移, 当动态迁移完成之后, 任务随着虚拟环境在新的物理机中继续运行下一个时间段。依此类推, 任务一直在迁移状态和运行状态中切换, 根据任务类型不同, 直到执行完成或者一直长期运行下去。过程中, 如果任务运行所在的物理机被成功攻击, 任务被破坏, 则需要退回到开始状态, 重新执行任务。需要强调的是, 本文我们假设攻击者的目标只有运行中的任务, 而动态迁移过程不受攻击影响。

对于攻击者来说, 攻击开始时每个攻击者会独立随机选择一台物理机作为目标机器进行攻击。一旦成功攻破某一台物理机, 攻击者会判断目标任务是否正运行在当前物理机上。如果发现目标任务在运行的话则会执行破坏程序, 使得任务运行失败并退回开始状态重新选择新的物理机和虚拟平台重新执行。如果攻击者成功攻击物理机但是没有发现目标任务在运行, 则立刻退出并重新选择目标物理机进行下一次攻击。本文中, 任务每个阶段的运行时间以及动态迁移时间可以服从一般分布。攻击者的攻击时间服从指数分布。

由于多个攻击者的存在, 我们使用图 2 中的例子具体描述任务运行过程中可能发生的任务事件和攻击事件。图 2 以系统中存在两个攻击者并且任务有四台物理机资源为例子。其中, $adversary-x$ 代表编号为 x 的攻击者, $x\in\{1,2\}$。PM_y 表示编号为 y 的物理机, $y\in\{A,B,C,D\}$。R_j 表示任务成功完成 j 次动态迁移之后的下一阶段运行过程。L_j 表示任务的第 j 次动态迁移过程和时间。$H_{s,m}$ 则表示编号为 s 的攻击者的第 m 次攻击所花费的时间。图 2 所示的例子中初始状态, 任务选择物理机 PM_A 开始第一阶段的执行。在 PM_A 上运行 R_0 时间之后, 任务重新选择下一个物理机平台 PM_B 并迁移到 PM_B 上进行第二阶段的运行。迁移消耗了 L_1 时间。同样的, 经过 R_1 时间后, 任务在 PM_B 上完成第二阶段的运行并迁移到 PM_C 上继续下一阶段的运行、迁移等等……。图中 A_1 和 A_2 分别表示两个攻击者的攻击过程和时间, 根据前面的假设, 攻击时间 A_1 和 A_2 服从指数分布, 并且一旦攻击成功物理机后, 攻击者能够立刻判断目标任务是否在当前物理机上运行。因此我们可以看到图 2 的示例中, 两个攻击者的前 4 次攻击以及 1 号攻击者的第 5 次攻击都没有命中任务运行所在的物理机, 任务的运行没有被破坏。然而在 2 号攻击者的第五次攻击(也就是 $H_{2,5}$)中,命中了任务所在的物理机 PM_C, 因此任务在 $H_{2,5}$ 的结尾被破坏。根据上面的描述, 我们将整个过程分成了, 两个阶段, 阶段 1 表示任务的正常运行和迁移, 没有被攻击者破坏。阶段 2 则表示, 任务在此阶段被攻击者破坏, 需要重新执行。

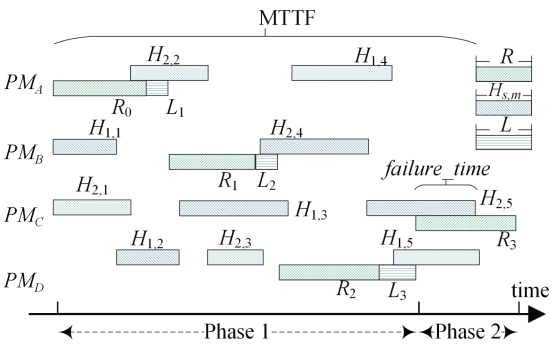

图 2 任务执行过程和攻击过程描述

Figure 2 The events occur in the job execution process

3.2 攻击模型

上一节我们描述了本文所讨论的系统环境和攻防双方的行为细节。本文考虑的攻防场景中, 防御方提供多个底层平台用来运行一个关键任务。攻击方掌握了一个或者多个针对防御方平台的漏洞, 并且是可利用实施攻击的漏洞。本文所讨论的威胁模型中, 攻击者的目的是成功攻击渗透进入防御方的平台并尝试探索和破坏关键任务的运行, 使得关键任

务的完成时间尽可能的延长。这里我们将攻击者实施攻击时对掌握漏洞的利用时间称为攻击时间。我们假设攻击者每轮攻击都会随机选择目标平台。对于攻击者来说, 所有的平台, 包括当前没有运行关键任务的平台都是处于活动状态并且是可访问的。

3.3 效能量化分析模型

介绍我们的分析模型之前, 首先需要介绍一下对于多攻击存在情况的处理。根据上一节中的描述, $A_1, A_2, A_3, \ldots$ 和 A_X 分别表示攻击者 $1,2,3,\ldots$ 和攻击者 X 对于目标平台的攻击时间。根据我们的假设, 它们是服从指数分布的独立事件。我们用 $\lambda_1, \lambda_2, \lambda_3, \ldots$ 和 λ_X 表示相应的攻击者攻击时间服从指数分布的参数。由于我们考虑的是任务在系统中的运行时间情况, 并且任务一旦被攻击破坏, 则需要回到初始状态重新运行, 因此对于多攻击者存在的情况, 我们只需要关注多个攻击者的多个攻击中最先成功破坏目标任务的情况。我们用 A 表示多攻击者中最先成功攻击目标任务所在平台的平均攻击时间。根据指数分布的性质, 我们可以知道随机变量 $A=\min\{A_1, A_2, \ldots, A_X\}$ 也服从指数分布[23], 并且参数为 $\theta=\lambda_1+\lambda_2+\lambda_3+\ldots+\lambda_X$。

为了评估任务的失效时间和完成时间我们用 K 表示任务被攻击者破坏之前成功迁移的次数, 也就是成功完成执行的阶段数。那么, $\{K=0\}$ 意味着任务在第一阶段, 也就是在第一台物理机上的运行过程就被攻击者破坏了。这种情况下, 平均失效时间 MTTF 可以用上一节图 2 中的 *failure_time* 的均值来表示。也就是说这种情况下, $MTTF=E[failure\_time]$。而 $\{K>0\}$ 则表示任务在被破坏之前成功执行了 K 个时间阶段, 也就是说, 在前 K 阶段, 攻击者都没有成功攻击到任务所在的物理机。具体我们可以将任务的完整运行情况表示为一个具有以下属性的离散时间马尔科夫过程:

具有状态空间 $0,1,2,\ldots,K,(K\geqslant 0)$

状态 k $(1\leqslant k\leqslant K)$, 表示任务在迁移 k 次之前没有被破坏过

初始状态是 0 状态, 并且一旦任务被攻击者破坏, 则回到 0 状态

图 3 展示了我们本文所提出的分析模型。图中 p 表示任务在当前平台运行过程中没有被攻击的概率。模型从 k 状态到 $k+1$ 状态表示任务成功完成了当前阶段的执行, 并迁移到下一平台。从 k 状态到 0 状态则表示任务在第 k 阶段的运行中被攻击者破坏, 需要重新开始运行。需要注意的是, 针对短期任务, 也就是有固定执行时间需求的任务, 我们的模型是一个具有吸收态的有限状态马尔科夫模型。状态 K 作为模型的吸收态, 处于此状态则模型结束, 表示短期任务成功完成了所有阶段的运行。而对于一直需要运行的长期任务, 我们的模型则是一个无穷状态马尔科夫模型。以图 2 中的任务运行情况为例, 该情况下, 任务在第三阶段被攻击, 也就是说任务成功完成了两次迁移, 对应到模型中 $K=2$。

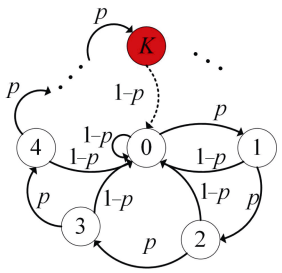

图 3 马尔可夫模型状态转移

Figure 3 State transitions of Markov model

3.4 指标量化公式

在本节中, 我们将根据前面描述的系统环境和模型定义, 给出本文主要评估指标的闭合解计算和证明过程。

3.4.1 长期任务平均失效时间 MTTF

对于一个长期任务, 由于没有固定的运行需求时间, 没有总的完成时间, 因此计算任务由于攻击者导致的平均失效时间可以作为衡量防御效果的一个很好的方式。本节我们将详细介绍如何利用我们的模型计算长期任务的 MTTF。根据图 2 所描述的任务运行情况和图 3 中我们定义的模型, 我们将 MTTF 的计算过程分为 4 个步骤。首先, 计算任务所在平台能够成功迁移的期望概率, 也就是图 3 中的 p。第二步, 计算攻击者成功攻击破坏任务时, 当前阶段任务已经运行的时间, 也就是任务上次迁移完成到失效的时间, 图 2 中我们用 *failure_time* 来表示。第三步, 根据前面两步得到的 p 和 *failure_time* 的期望以及任务迁移次数 k 我们可以得到在 $K=k$, 也就是任务成功运行的 k 阶段情况下 MTTF 的条件期望。最后, 使用条件希望, 计算出最终的任务平均失效时间 MTTF。本节剩余部分, 我们分别给出具体的计算过程和公式。

引理 1: 任务免受攻击破坏成功迁移的期望概率 p, 如下公式 1 所示:

$$p=\sum_{i=0}^{\infty}\left(\frac{N-1}{N}\right)^i\left(\frac{(\theta E[R])^i}{i!}\right)e^{-(\theta E[R])},N\geqslant 1 \qquad (1)$$

证明: 首先我们考虑在任务某一运行阶段 R 时间内发生的一系列攻击事件, 如图 4 所示, 下方绿色条纹段显示的是任务的运行阶段 R, 上方蓝色显示的是一系列的攻击时间 A_1, A_2, A_3, A_4, A_1' 表示的是任务开始当前阶段运行后第一个攻击事件剩余的时间。根据我们前面的假设, 攻击者的攻击时间服从指数分布, 因此, 一系列的攻击事件的发生可以表示成一个泊松过程, 由于具有无记忆性, 因此无论攻击完成后任务运行到哪一个时间, 当前阶段内平均发生的攻击事件次数是固定的。因此计算任务免受攻击破坏成功迁移的期望概率 p 的时候, 我们可以先计算任务运行阶段时间 R 内发生 i 次攻击的概率 $Prob(I=i)$, 如下公式 2 所示:

$$Prob\left(I=i\right)=\frac{\left(\theta E\left[R\right]\right)^{i}}{i!}e^{-\left(\theta E[R]\right)} \tag{2}$$

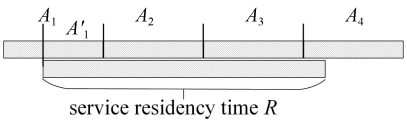

图 4 任务运行某一阶段 *R* 发生的攻击事件

Figure 4 The number of attacks in a *R*

根据我们的假设, 如果任务能够成功迁移, 意味着在当前运行阶段 R 时间内所发生的所有攻击事件, 都没有命中任务当前所在的物理机, 也就是说, R 时间内发生的攻击都是以其他物理机为目标的。因此我们可以得到条件概率 $Prob\left(p|\ i\right)$, 表示任务运行阶段发生 i 次攻击条件下能够成功迁移到下一阶段的概率, 如下公式 3 所示:

$$Prob\left(p\,|\,i\right)=\left(\frac{N-1}{N}\right)^{i} \tag{3}$$

将公式 2 和公式 3 相乘, 我们得到了任务免受攻击破坏成功迁移的期望概率 p, 如公式 4 所示, **引理 1** 得证。

$$\begin{aligned} p&=\sum_{i=0}^{\infty}Prob\left(I=i\right)\cdot Prob\left(p|\ i\right)\\ &=\sum_{i=0}^{\infty}\left(\frac{N-1}{N}\right)^{i}\left(\frac{\left(\theta E\left[R\right]\right)^{i}}{i!}\right)e^{-\left(\theta E[R]\right)} \end{aligned} \tag{4}$$

引理 2: 计算攻击者成功攻击破坏任务时, 当前阶段任务已经运行的时间 $failure\_time$ 的期望值 $E[f\_t]$, 如下公式 5 所示:

$$\begin{aligned} E\left[f\_t\right]=&\sum_{j=1}^{\infty}\sum_{i=1}^{j}k\cdot E\left[A\right]\cdot\left(\frac{N-1}{N}\right)^{i-1}\\ &\cdot\frac{1}{N}\cdot\frac{\left(\theta\cdot E\left[R\right]\right)^{j}}{j!}\cdot e^{-\theta E[R]} \end{aligned} \tag{5}$$

证明: 首先, 我们考虑在已知 R 时间段内发生 j 次攻击的条件下 $failure\_time$ 的条件期望 $E[f\_t|\ J=j]$。根据我们的假设, 任意一次攻击都有可能命中任务当前所在的物理机。我们定义 J 表示直到第 J 攻击才命中到目标任务所在物理机, 意味着之前的 $J-1$ 次攻击都是以其他物理机为目标的。用 $E[A]$ 表示平均攻击时间, 我们可以得到公式 6 如下:

$$\begin{aligned} &E\left[f\_\ t|J=j\right]=\\ &\sum_{i=1}^{j}i\cdot E\left(A\right)\cdot\left(\frac{N-1}{N}\right)^{i-1}\cdot\frac{1}{N} \end{aligned} \tag{6}$$

结合公式 2, 我们可以求出 $E[f\_t]$, **引理 2** 得证。需要注意的是, 与**引理 1** 中计算 p 时求和从 0 开始不同的是, 在**引理 2** 公式 5 中求和从 1 开始, 这是因为要造成任务被攻击失效, 至少需要有一次攻击事件发生。

有了 p 和 $E[f\_t]$ 之后我们便可以计算在已知 $K=k$ 情况下, MTTF 的条件期望。本文中, 我们用 $MTTF_k$ 来表示 $K=k$ 时的条件期望值。在此已知条件下, 任务成功完成了 k 阶段的运行, 迁移了 k 次, 并且在下一阶段的运行过程中被攻击者命中破坏。因此整个任务的运行时间可以分为两部分。第一部分是任务的正常运行部分, 包括 k 次运行时间段和 k 次迁移时间, 也就是 $k(E[R]+E[L])$, 其中 $E[R]$ 表示任务单阶段运行平均时间, $E[L]$ 表示任务平均迁移时间。第二部分则是最后一阶段被破坏前的任务运行时间, 也就是 $E[f\_t]$。将两部分求和我们可以得到 MTTF 条件期望如下公式 7 所示:

$$MTTF_k=k\cdot\left(E\left[R\right]+E\left[L\right]\right)+E\left[f\_t\right] \tag{7}$$

定理 1: 长期任务平均失效时间 MTTF, 如下公式 8 所示:

$$\begin{aligned} MTTF=&\\ &\sum_{k=0}^{\infty}\left(\begin{array}{l} k\cdot(E[R]+E[L])+\sum_{j=1}^{\infty}\sum_{i=1}^{j}k\cdot E[A]\cdot\\ \left(\frac{N-1}{N}\right)^{i-1}\cdot\frac{1}{N}\cdot\frac{(\theta\cdot E[R])^{j}}{j!}\cdot e^{-\theta E[R]} \end{array}\right). \end{aligned}$$

$$\left(\sum_{i=0}^{\infty}\left(\frac{N-1}{N}\right)^{i}\left(\frac{(\theta E[R])^{i}}{i!}\right)e^{-(\theta E[R])}\right)^{k}\cdot \left(1-\sum_{i=0}^{\infty}\left(\frac{N-1}{N}\right)^{i}\left(\frac{(\theta E[R])^{i}}{i!}\right)e^{-(\theta E[R])}\right) \tag{8}$$

证明: 根据我们前面的描述, 任何一次攻击都有可能命中任务当前所在平台, 因此任务在被攻击失效之前的迁移次数服从几何分布, 据此我们可以计算出任务在正常运行 k 阶段之后被攻击者破坏的概率是 $p^k(1-p)$。结合前面任务平均失效时间的条件期望, 我们可以计算出任务从 0 阶段开始运行到被攻击失败的平均时间 MTTF, 如下公式(9)所示:

$$MTTF=\sum_{k=0}^{\infty}\left(MTTF_k\cdot p^k\cdot(1-p)\right) \tag{9}$$

将公式(1), (5), (7)带入公式(9), 我们可以得到最终的 MTTF 值, **定理 3.1** 得证。

3.4.2 系统总成本计算

由于被攻击导致的任务失效无疑会给系统带来成本上的损失。在部署安全防护系统的时候, 也会造成新的成本开销。本节我们讨论如何计算动态平台系统的整体成本开销。我们以 1000 天系统开销作为评估基准。TC 表示系统运行 1000 天的总开销。我们将系统总开销分成以下 4 个部分:

(1) 系统失效造成的损失(1000 天内平均失效次数*每次失效的损失);

(2) 购买物理机的成本(物理机数目*每台物理机的价格成本);

(3) 额外安全机制成本(攻击者对物理机的攻击时间*安全机制每天的成本);

(4) 任务迁移成本(1000 天内平均 MTTF 区间数*每个 MTTF 区间内的任务迁移次数)。

将 4 个部分相加得到我们系统从开销评估指标 TC 如以下公式(10)所示:

$$TC=1000/\mathrm{MTTF}\cdot loss+N\cdot C_{PM}+E[A]\cdot C_{security}+1000/\mathrm{MTTF}\cdot\sum_{k=0}^{\infty}k\cdot C_{migration}\cdot p^k\cdot(1-p) \tag{10}$$

3.4.3 短期任务平均完成时间 JCT

前一节我们讨论了如何计算长期任务的平均失效时间, 对于短期任务来说, 尽快也存在因为攻击失效需要重新执行任务的情况, 因此, MTTF 也能在一定程度上分析动态平台系统的防御效果。但是, 由于短期任务具有一个固定的运行时间需求, 因此, 分析动态平台对于短期任务的防御能力, 使用任务平均完成时间 JCT 要比使用 MTTF 作为指标更为恰当。此外, 与计算 MTTF 不同的是, 对于一个有固定时间需求的短期任务, 我们在将任务分成多阶段多平台运行的时候, 每一阶段的运行时间以及总的任务阶段数都是一个固定值, 而不是长期任务时候的随机变量。也就是说, 在本节对于 JCT 的计算过程中, 任务单阶段的运行时间 $E[R]=R$。为了表述方便, 我们在本节的计算过程中用 J 来表示任务在面对攻击情况下的从开始运行到最终完成运行的时间。因此, 我们最终的计算目标是在已知任务阶段数 $K=k$ 情况下求 $E[J]$ 的值。很明显 $E[J]$ 由以下两个部分构成:

(1) $(kE[R]+(k-1)E[L])\cdot p^k$, 表示的在整个任务执行 k 阶段过程中都没有发生被攻击失效的情况, 任务正常的完成了所有运行和迁移。其中 $kE[R]+(k-1)E[L]$ 代表任务的总运行时间和迁移时间。p^k 表示的任务在整个运行过程中没有被攻击者命中破坏的概率;

(2) $\sum_{j=0}^{k-1}\begin{pmatrix}(E[R]+E[L])\cdot j+\\ E(f\_t)+E[J]\end{pmatrix}\cdot p^j(1-p)$, 表示任务在不同阶段数失效重新执行总完成时间的期望总和, 其中求和下标 j 表示任务正常完成了 j 阶段的运行和迁移, 并且在第 $j+1$ 阶段被攻击者破坏返回到初始阶段重新运行。式中 $(E[R]+E[L])\cdot j+E(f\_t)$ 表示任务正常运行和迁移 j 阶段的时间以及下一阶段被攻击之前已运行的时间和, $E[J]$ 则表示任务被破坏后回到起点需要重新运行一个完整的时间需求。$p^j(1-p)$ 表示任务前 j 阶段运行迁移都成功并且在第 $j+1$ 阶段运行中被攻击者破坏的概率。

将两部分求和得出公式 11 中的 $E[J]$ 的表达式如下:

$$\begin{aligned}E[J]&=(kE[R]+(k-1)E[L])\cdot p^k+\sum_{j=0}^{k-1}\begin{pmatrix}(E[R]+E[L])j+\\ E(f\_t)+E[J]\end{pmatrix}p^j(1-p)\\&=(kE[R]+(k-1)E[L])\cdot p^k+(E[R]+E[L])\cdot\sum_{j=0}^{k-1}j\cdot p^j(1-p)+\left(E(f\_t)+E[J]\right)\cdot\sum_{j=0}^{k-1}p^j(1-p)\end{aligned} \tag{11}$$

又已知:

$$\sum_{j=0}^{k-1} jp^j = \frac{(k-1)p^{k+1} - kp^k + p}{(1-p)^2}$$

和:

$$\sum_{j=0}^{k-1} p^j = \frac{1-p^k}{1-p}$$

我们很容易得到:

$$\begin{aligned}&\left(E[R]+E[L]\right)\sum_{j=0}^{k-1} jp^j(1-p) = \\ &\left(E[R]+E[L]\right)\frac{p\left(1-p^k\right)}{1-p} \\ &-\left(E[R]+E[L]\right)kp^k\end{aligned} \tag{12}$$

以及:

$$\begin{aligned}&\sum_{j=0}^{k-1}\left(E[f\_t]+E[J]\right)p^j(1-p) = \\ &E[f\_t]\left(1-p^k\right)+E[J]\left(1-p^k\right)\end{aligned} \tag{13}$$

将公式(12)和（13）带入公式 11 我们可以得到:

$$\begin{aligned}E[J] = &\left(kE[R]+(k-1)E[L]\right)\cdot p^k + \\ &\left(E[R]+E[L]\right)\frac{p\left(1-p^k\right)}{1-p} - \\ &\left(E[R]+E[L]\right)kp^k + \\ &E[f\_t]\left(1-p^k\right)+E[J]\left(1-p^k\right)\end{aligned} \tag{14}$$

最后将公式(14)中 $E[J](1-p^k)$ 移到等式左边再除去 p^k, 化简得到最后的任务平均完成时间 JCT 如下所示:

$$\begin{aligned}E[J] = &(E[R]+E[L])\frac{p\left(1-p^k\right)}{1-p} \\ &+E[f\_t]\frac{\left(1-p^k\right)}{p^k} - E[L]\end{aligned} \tag{15}$$

4 数值分析与讨论

本节我们应用上文提出的分析模型和评估指标的解析解, 通过数值分析的形式评估不同防御配置参数和攻击场景对动态平台系统的防御效能的影响。此外, 我们还使用了蒙特卡洛仿真的方法分析验证了前文给出的任务成功迁移概率、MTTF 和 JCT 解析解的准确性。

为了与实际系统环境吻合, 我们的仿真实验由攻击事件、任务运行事件和任务迁移事件 3 个事件构成, 仿真实验持续 10 万个时间单位, 每个仿真实验重复 10 次最后取 10 次实验结果的平均值用来和数值解进行比较分析。对于 MTTF 的计算中, 我们使用均值分别为 5 天和 30 min 的指数分布来分别生成 R 和 L 的值, 而对于 JCT 的计算, 如上文所述则是直接赋予一个常数值。本文所有的数值计算和仿真实验都是在 Maple 18[30]中进行的。实验过程和计算中的参数的缺省值参照 Table 1 设置。

4.1 MTTF 数值分析

本节内容展示了不同攻击速率对任务 MTTF 和任务成功迁移概率的影响。同时, 我们将本文结果与之前工作[24]做出比较并展示了我们对之前工作的提升。图 5 和 6 展示了我们数值分析的结果, 其中标注“*Previous*”的蓝色虚线显示的是之前工作的参考结果。标注为“*Improved*”的实线表示的是本文我们模型的分析结果。标注为“*Simulation*”的三角形散点表示的是我们仿真实验的结果, 用来和模型的数值分析结果进行比较, 验证模型的准确性。从图 5 和 6 中都可以看出, 在其他参数保持默认的情况下, 攻击时间大于 5 天的时候, 我们的数值结果和之前的结果以及仿真实验结果都很接近。而随着攻击时间越来越短, 之前的结果则越来越偏离仿真的结果, 而本文模型的结果则依然和仿真实验结果比较吻合。也就是说, 本文的模型解在各种攻击速率下都是准确的, 而先前的工作在攻击速率较快的情况下会有较大的分析误差。我们分析造成这种误差的原因, 是因为在之前的工作中, 作者忽略了在某一个任务执行阶段 R 时间内可能会有多次攻击命中任务所在平台的情况。特别是当攻击速度相对快的时候, R 时间内多次攻击命中任务所在平台的概率也会相对较大, 所以导致了在攻击时间较短的时候, 之前工作结果的不准确性。而本文模型的计算则将所有可能的攻击情况都考虑了。

验证了我们模型结果的准确性之后, 我们再回到我们的实验结果中。从图 5 和 6 中我们可以看到, 随着攻击时间从 0.5 天上升到 15 天, 任务迁移成功的概率也变得越来越大, 显然是由于攻击时间变长, 攻击速率变慢, 任务运行周期内攻击者成功攻击物理机破坏任务运行的概率变低。相对来说, 任务的正常运行和迁移变的相对更加安全, 平均失效时间也会变长, 我们可以看到 MTTF 从 3 天增长到了 89 天。也就是说, 在保持其他条件不变的情况下, 物理机被攻击的概率越大, 平均失效时间 MTTF 越短。对于网络系统管理员来说, 虽然攻击者的攻击时间或者速率是未知的, 但是可以通过

对于物理机部署更加先进的安全技术和策略来提高系统的平均失效时间。

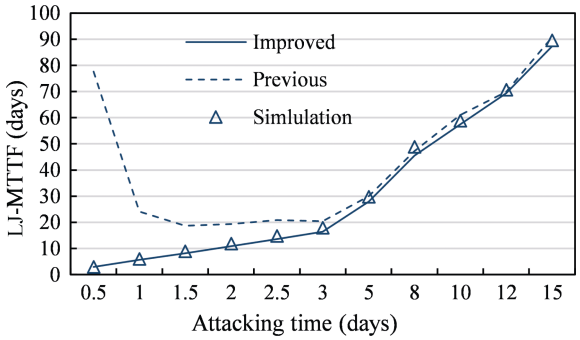

图 5　不同攻击速率下的任务平均失效时间

Figure 5　MTTF under different attack time

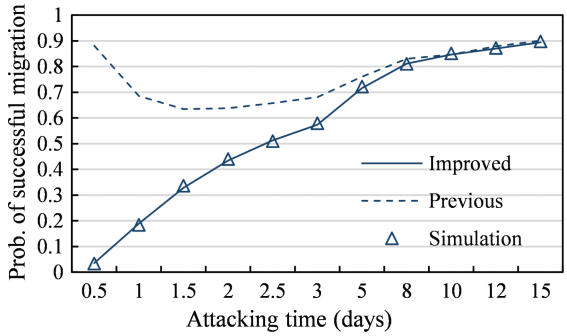

图 6　不同攻击速率下的任务迁移成功概率

Figure 6　Prob. of successful migration under different attack time

上文我们介绍了不同攻击时间对于任务 MTTF 和迁移成功概率的影响。那么系统中防御方能够主动设置的参数对防御效能有何影响？图 7 展示了我们实验结果。如图所示，标记为“*num_MTTF*”和“*num_p*”的直线和虚线分别表示我们使用模型解析解得到的不同任务周期时间对于平均失效时间和迁移成功概率的影响情况。标记为“*sim_MTTF*”和“*sim_p*”的散点图形则表示与解析解相对应的仿真实验结果。从图中我们可以看出当任务运行周期时间增加时，成功迁移的概率变低。这是由于运行周期时间增加导致同一周期内发生攻击次数变多，攻击者命中目标任务所在物理机概率变大。而对于任务平均失效时间，即使任务运行周期时间增加。MTTF 也不会有明显的变化，在我们的实验参数配置下，MTTF 维持在 72 天左右，变化较小。也就是说，尽管任务成功迁移概率减小，但是也会也会因为单周期运行时间更长导致 MTTF 不发生明显变化。因此我们可以得出结论，任务平均失效时间 MTTF 不受任务单阶段运行时间的明显影响。

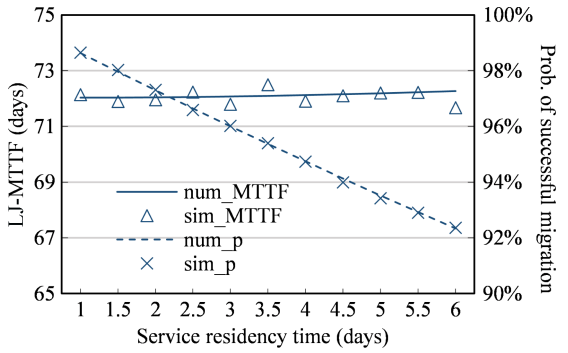

图 7　不同任务周期时间下的任务平均失效时间和成功迁移概率

Figure 7　MTTF and the probabilility of successful migration under different residency time

图 5 和 6 分别显示了场景 1 中 N 取不同值时成功攻击次数的瞬态值(m_1)和累积值(m_2)。从图 5 可观察到成功攻击次数先增加然后趋于稳定。因为当服务刚开始在平台上运行时，攻击者需要一定时间才能进行感染。图 6 表明 N 的取值不同时，成功攻击的累积数量与图 4 具有相似的趋势。

除任务运行周期时间外，另一个防御方能够主动配置的系统参数是 PM 的数量。图 8 展示了我们实验中不同物理机数量对防御效能结果的影响。从图中我们可以明显的看出，保持其它参数不变，对着物理机数量的增多，任务成功迁移的概率 p 以及平均失效时间 MTTF 都迅速变大。显然，在攻击者能力不变的情况下，更多的物理机选择，能够使攻击者更难发现任务所在的物理机平台，任务的运行变得更加安全。通过比较多物理机和单物理机的分析结果我们可以很容易的看出动态平台技术的显著防御效果。当然，更多的物理机也会显著增加防御成本，我们将在接下来的系统总开销成本上做进一步分析解释。

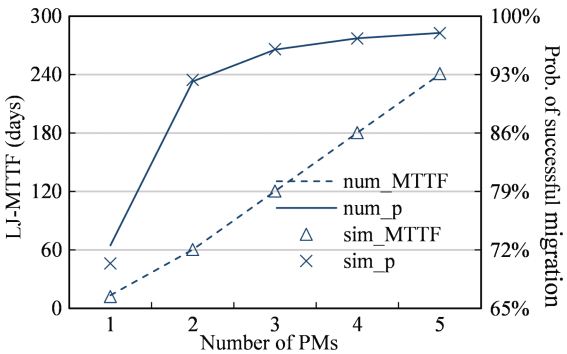

图 8　不同物理机数量下的任务平均失效时间和成功迁移概率

Figure 8　MTTF and the probabilility of successful migration under different number of PMs

4.2 系统总成本预测分析

在本小节中, 我们分析不同系统参数与系统总开销成本之间的关系。我们用四元组 $(loss, C_{migration}, C_{PM}, C_{security})$ 来演示实验中与成本有关的系统参数设置。参数 $loss, C_{migration}, C_{PM}, C_{security}$ 的缺省值参照表 1。与上一节分析 MTTF 类似, 本节主要研究不同任务周期时间、不同攻击速度以及不同物理机数对系统总的成本开销的影响。我们的数值分析主要依靠公式 10 中的成本计算公式完成。具体来说, 不同任务周期时间影响会产生不同的任务迁移频率, 导致迁移成本 $C_{migration}$ 的改变。而不同的攻击速率和物理机数目则分别会造成额外安全配置成本 $C_{security}$ 和物理机成本 C_{PM} 的改变, 并最后反应到总的开销成本上。需要注意的是, 本节所介绍的数值分析主要是为了展示我们模型解析解对于系统成本预测的帮助, 在实际部署系统的时候, 防御方或者管理员可以依据自身的任务需求和物理机购买成本等修改分析参数值。

图 9 展示的是不同系统周期时间对系统总成本的影响。我们保持攻击速率、物理机数目不变, 计算了三组实验结果。三组结果分表表示不同的成本参数值四元组为 $(10k,10,1k,2k)$, $(10k,50,1k,2k)$ 和 $(10k,100,1k,2k)$。从图中可以看出, 随着任务周期时间变长总的系统开销变小。并且相互比较三组结果可以看出, 迁移成本 $C_{migration}$ 更大的时候, 总成本的降低速度更明显。主要原因是, 更短的运行周期时间会造成更频繁的迁移操作, 导致更大的系统开销, 并且更大的 $C_{migration}$ 放大了这种影响。在实际部署系统的时候, 管理员可以通过适当的降低任务迁移频率降低成本。

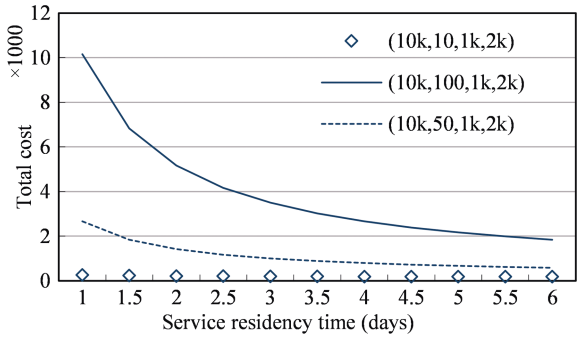

图 9 不同任务周期时间下的系统总开销

Figure 9 The total cost of the system under different residency time

图 10 展示了不同攻击时间对系统总成本的影响。同样的, 我们给出了三组参考数据。保持其他参数不变, 观察成本参数四元组为 $(10k,10,1k,2k)$ (菱形散点图)和 $(10k,10,1k,10k)$ (虚线图)的结果我们可以看到, 在额外安全成本比较低的情况下, 系统的成本随着攻击时间变大而降低。也就是说, 部署更多的防御机制即使会消耗一定的成本, 但是由于任务被攻击失败次数的减少, 最后总的系统成本反而会降低。而对于额外安全配置成本比较高的情况 $(10k,10,1k,20k)$, 我们可以看到随着攻击时间的增加, 不同于另外两个曲线一直递减的趋势, 系统总的成本以 9 天为分界点先降低再增高。原因是因为虽然更长的攻击时间能够带来任务更安全的运行, 但是由于产生更长攻击时间需要的额外防御成本更多并且占比更大, 因此总的系统成本反而会变大。

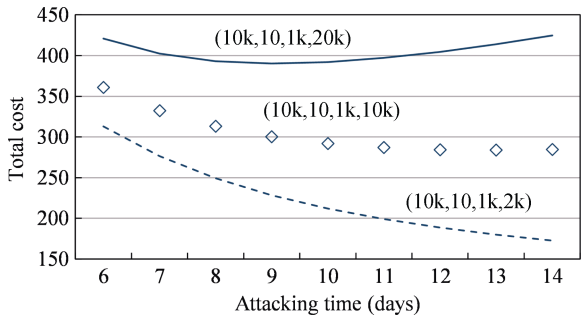

图 10 不同攻击时间下的系统总开销

Figure 10 The total cost of the system under different attack time

图 11 演示的是不同物理机数量对系统总开销的影响。上一节我们分析了, 更多的物理机数目会带来更好的防护效果, 任务被攻击破坏的概率越低, 但是显然更多的物理机数量也会带来更大成本开销。图中我们可以看到系统总的开销成本曲线在物理机

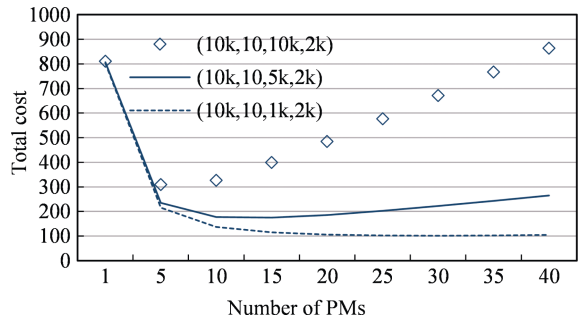

图 11 不同物理机数下的系统总开销

Figure 11 The total cost of the system under different number of PMs

数目为 5 左右的时候有一个很明显的折点。折点左边表示的是物理机数量较少, 因为攻击者对任务破坏导致的成本开销比较明显, 因此, 而折点右边则表示, 物理机数量过多, 任务虽然相对安全, 但是物理机本身的成本造成了明显的系统开销。防御者在部署动态平台机制时, 可以根据任务本身的安全需求选择一个最优的平台数。

4.3 JCT 数值分析

本节我们展示的是不同系统配置参数对于一个短期任务总的完成时间的影响。同样的, 我们使用控制变量的方式来分析不同参数的影响, 并且使用仿真实验的结果来佐证模型解析解的准确性(本节结果图中“*Num*”和“*sim*”标注的曲线和散点分别表示我们的解析解结果和仿真结果)。需要注意的是实验采用的默认参数设置与上一节相同, 但是由于是分析短期任务的总完成时间, 因此, 任务会有一个固定的运行时间需求, 本节中我们将短期任务的固定运行时间需求设置为 30 天。

首先我们分析的是不同的任务阶段数划分对总的完成时间的影响。我们通过将 30 天任务分成 1 到 30 个阶段数(也就说说每阶段运行时间从 30 天到 1 天), 观察总的完成时间影响。图 12 显示任务总完成时间 JCT 随着任务阶段数增加而缓慢的增加。显然, 总的任务需求时间固定的情况下, 阶段数越多会造成更多的迁移时间延迟, 但由于迁移时间相对运行时间来说比较短, 因此多阶段的划分不会对任务总完成时间造成明显的延迟。同时, 更多阶段数的划分也有益于提高任务运行的安全性。

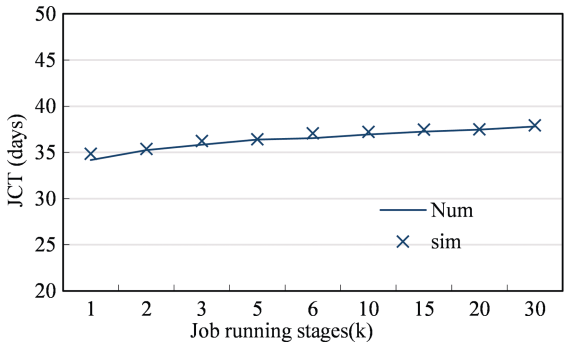

图 12 不同阶段数下的任务完成时间
Figure 12 JCT under different stages

图 13 和 14 分别演示了不同攻击时间和不同物理机数目对任务总完成时间的影响。从图中我们可以看出攻击时间和物理机数目对总完成时间都有很明显的影响。在保持其他条件不变的情况下, 将攻击时间从 1 天增加到 10 天, 可以看到任务总的完成时间从 810 天降低到了 38 天时间。显然, 由于攻击一台 PM 需要的时间越长, 任务运行相对越安全, 也能越快的完成总的运行。同理, 更多的物理机也会增加攻击的难度, 使得完成时间越短。需要注意的是, 虽然总完成时间是逐渐递减的, 但是递减的速率也在放缓, 而总的成本却在不断增加。

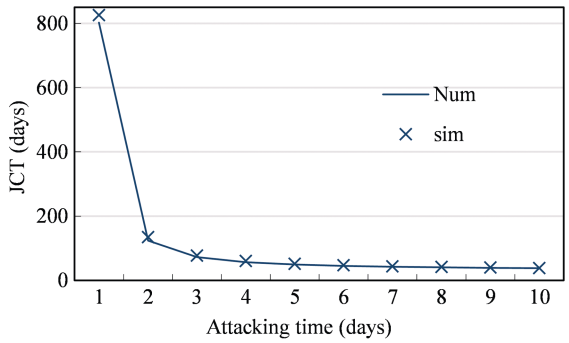

图 13 不同攻击时间下的任务完成时间
Figure 13 JCT under different attack time

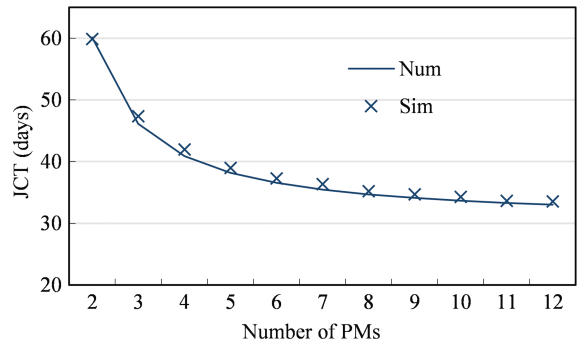

图 14 不同物理机数下的任务完成时间
Figure 14 JCT under different number of PMs

4.4 数值分析总结

根据本节的实验分析, 我们可以得到以下总结:

(1) 我们的模型和公式可以准确地计算任务 MTTF 和 JCT 等动态平台防御效能指标。

(2) 随着任务运行周期时间的增加, MTTF 几乎不发生明显变化。因此, 可以适当的延长任务运行周期时间, 以减少迁移成本。

(3) 迁移频率的增加不会对整个 JCT 产生重大影响, 但会带来更高的安全利益。

(4) MTTF 和 JCT 都随着攻击时间和 PM 数量的增加而增加。因此, 系统管理员可以在成本允许下部署更多的安全机制以延长攻击者平均攻击时间并增加系统中物理机的数量。

5 本文总结与未来工作

本文我们从任务运行时间的角度分析了移动目

标防御技术的防御效能。我们以一个用来运行关键任务并且易受攻击的信息系统为分析场景，该系统部署了基于动态平台技术的移动目标防御机制以减少攻击对任务运行的影响。系统中有多台物理机，每台物理机上都部署了虚拟环境用来满足任务的运行和迁移。通过将任务运行划分为多阶段进行并虚拟机动态迁移技术来达到在不同物理机上的跳转的方式来实现动态平台技术。在此场景基础上，我们提出了一个新的分析模型，通过量化分析长期运行的任务的平均失效时间 MTTF 和短期任务的平均完成时间JCT来评估动态平台技术的防御效果和性能损失。并且，我们还通过仿真实验的方式验证了我们提出的模型和相关计算公式的准确性。我们的模型和结果可以帮助防御方或者系统管理员在配置动态平台防御策略的时候选择更合理的配置参数。

本文中我们的模型基于一个特定的攻击模型假设。在实际应用中，由于网络攻击本身是个很复杂的过程，涉及多种技术，实际攻击模式可以会因为特定场景产生不同变化。因此在未来工作中，我们可以尝试从复杂的攻击行为中总结出统一的攻击模型，并以此为基础扩充我们的分析模型使其适用范围更加广泛。

参考文献

[1] C. N. Authorities. https://cve.mitre.org/cve/ .Common vulnerabilities and exposures, 2020.

[2] KENNA security. https://www.kennasecurity.com/, 2019.

[3] David Molloy, Joe Tidy: Twitter hack: 130 accounts targeted in attack. BBC News, BBC, 2020.

[4] Jajodia S, Ghosh A K, Swarup V, et al. Moving Target Defense - Creating Asymmetric Uncertainty for Cyber Threats[C]. *Advances in Information Security 54*, 2011.

[5] Xu J, Guo P Y, Zhao M Y, et al. Comparing Different Moving Target Defense Techniques[C]. *The First ACM Workshop on Moving Target Defense - MTD '14*, 2014: 97-107.

[6] Okhravi H, Riordan J, Carter K. Quantitative Evaluation of Dynamic Platform Techniques as a Defensive Mechanism[C]. RAID 2014: 405-425.

[7] Steinberger J, Kuhnert B, Dietz C, et al. DDoS Defense Using MTD and SDN[C]. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018: 1-9.

[8] Albanese M, Jajodia S, Venkatesan S. Defending from Stealthy Botnets Using Moving Target Defenses[J]. *IEEE Security & Privacy*, 2018, 16(1): 92-97.

[9] Sengupta S, Vadlamudi S G, Kambhampati S, et al. A Game Theoretic Approach to Strategy Generation for Moving Target Defense in Web Applications[C]. *AAMAS*. 2017: 178-186.

[10] Tian J, Tan R, Guan X H, et al. Enhanced Hidden Moving Target Defense in Smart Grids[J]. *IEEE Transactions on Smart Grid*, 2019, 10(2): 2208-2223.

[11] Zeitz K, Cantrell M, Marchany R, et al. Designing a Micro-Moving Target IPv6 Defense for the Internet of Things[C]. *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation*, 2017: 179-184.

[12] Zeitz K, Cantrell M, Marchany R, et al. Changing the Game: A Micro Moving Target IPv6 Defense for the Internet of Things[J]. *IEEE Wireless Communications Letters*, 2018, 7(4): 578-581.

[13] Lakshminarayana S, Yau D K Y. Cost-Benefit Analysis of Moving-Target Defense in Power Grids[J]. *IEEE Transactions on Power Systems*, 2021, 36(2): 1152-1163.

[14] [14] Zhang Z Y, Deng R L, Yau D K Y, et al. On Hiddenness of Moving Target Defense Against False Data Injection Attacks on Power Grid[J]. *ACM Transactions on Cyber-Physical Systems*, 2020, 4(3): 1-29.

[15] Griffioen P, Weerakkody S, Sinopoli B. A Moving Target Defense for Securing Cyber-Physical Systems[J]. *IEEE Transactions on Automatic Control*, 2021, 66(5): 2016-2031.

[16] Zhou Y Y, Cheng G, Jiang S Q, et al. Cost-Effective Moving Target Defense Against DDoS Attacks Using Trilateral Game and Multi-Objective Markov Decision Processes[J]. *Computers & Security*, 2020, 97: 101976.

[17] Huang C, Zhu S C, Yang Y. An Evaluation Framework for Moving Target Defense Based on Analytic Hierarchy Process[J]. *ICST Transactions on Security and Safety*, 2018, 4(13): 153527.

[18] Stout W M S, van Leeuwen B, Urias V E, et al. Leveraging a LiveNirtual/Constructive Testbed for the Evaluation of Moving Target Defenses[C]. *2018 International Carnahan Conference on Security Technology*, 2018: 1-5.

[19] Anderson N, Mitchell R, Chen I R. Parameterizing Moving Target Defenses[C]. *2016 8th IFIP International Conference on New Technologies*, *Mobility and Security*, 2016: 1-6.

[20] Alavizadeh H, Kim D S, Hong J B, et al. Effective Security Analysis for Combinations of MTD Techniques on Cloud Computing (Short Paper)[C]. *Information Security Practice and Experience*, 2017: 539-548.

[21] Chang X L, Shi Y, zhang Z J, et al. Job Completion Time under Migration-Based Dynamic Platform Technique[J]. *IEEE Transactions on Services Computing*, 2020:1-1.

[22] Docker Inc. https://docs.docker.com/glossary/?term=container, 2019.

[23] Hayes B. Cloud Computing[J]. *Communications of the ACM*, 2008, 51(7): 9-11.

[24] Yang R K, Chang X L, Misic J, et al. Exploiting Dynamic Platform Protection Technique for Increasing Service MTTF[C]. *2019 IEEE Global Communications Conference*, 2019: 1-6.

[25] Okhravi H, Hobson T, Bigelow D, et al. Finding Focus in the Blur of Moving-Target Techniques[J]. *IEEE Security & Privacy*, 2014, 12(2): 16-26.

[26] Connell W, Menascé D A, Albanese M. Performance Modeling of Moving Target Defenses with Reconfiguration Limits[J]. *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(1): 205-219.

[27] Connell W, Menascé D A, Albanese M. Performance Modeling of Moving Target Defenses[C]. *The 2017 Workshop on Moving Target*

Defense, 2017: 53-63.

[28] Chen Z, Chang X L, Han Z, et al. Numerical Evaluation of Job Finish Time under MTD Environment[J]. *IEEE Access*, 2020, 8: 11437-11446.

[29] Chen Z, Chang X L, Mišić J, et al. Model-Based Performance Evaluation of a Moving Target Defense System[C]. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020: 1-6.

[30] Maplesoft Inc. Maple 18, http://www.maplesoft.com.

陈志 于 2014 年在北京交通大学信息安全专业获得学士学位。现在北京交通大学信息安全专业攻读博士学位。研究领域为系统安全、移动目标防御。研究兴趣包括: 云计算、网络安全。Email:chenzhi@bjtu.edu.cn

常晓林 于 2005 年在香港科技大学计算机科学技术专业获得博士学位。现任北京交通大学计算机与信息技术学院教授。研究领域包括: 网络空间安全和人工智能安全。Email: xlchang@bjtu.edu.cn

杨润垲 于 2016 年在北京交通大学, 计算机技术专业获得硕士学位, 现在北京交通大学网络空间安全攻读博士学位, 研究领域为云计算、移动目标防御、区块链安全等。Email: 18112049@bjtu.edu.cn

韩臻 于 1991 年在中国工程物理研究院北京研究生部获博士学位。现任北京交通大学计算机与信息技术学院教授。研究领域包括: 计算机安全、系统安全、保密技术。Email:zhan@bjtu.edu.cn