

云环境下基于动态异构的拜占庭系统

陈扬, 扈红超, 刘文彦, 霍树民, 梁浩

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘要: 拜占庭系统被广泛应用于提高云平台的入侵容忍能力。但现阶段的拜占庭系统难以有效防御一种潜伏式攻击, 即攻击者通过累积控制个数超过容忍上限的执行体来破坏拜占庭系统。针对此问题提出云环境下具有动态异构性质的拜占庭系统, 即将拜占庭系统中的执行体进行最大化多样性配置, 同时利用系统漏洞和攻击技能水平来衡量执行体生命周期, 将执行体进行动态轮换。仿真结果表明, 提出的动态异构拜占庭系统能有效提升系统安全性。

关键词: 动态异构; 拜占庭; 入侵容忍; 攻击面; 平均攻击时间

中图分类号: TP302.8

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2018036

Dynamic and heterogeneous Byzantine system in cloud

CHEN Yang, HU Hongchao, LIU Wenyan, HUO Shumin, LIANG Hao

National Digital Switching System Engineering & Technological R&D Center, Zhangzhou 450002, China

Abstract: In order to improve the intrusion tolerance performance of cloud platforms, the Byzantine system model has been widely used. But the current Byzantine system is difficult to prevent a latent attack effectively, which can destroy the Byzantine system by accumulating the number of executive modules which the attacker controls and making it exceed the tolerance upper limit. To solve the problem, a dynamic and heterogeneous Byzantine system was proposed. In which the executive modules were maximized diversified in configurations such as different operating systems and so on. Meanwhile, every executive module was assigned a lifetime based on the attacker's ability and the vulnerabilities of the module. The novel system can significantly enhance the intrusion tolerance of the cloud service by diverse and dynamic attack surface, which is verified by simulations.

Key words: dynamic heterogeneous, Byzantine, intrusion tolerance, attack surface, MTTC

1 引言

云计算 (cloud computing) 是一种基于互联网的計算方式, 通过这种方式, 共享的软硬件资

源和信息, 可以按需求提供给计算机和其他设备。它提供可用的、便捷的、按需的网络访问。通过可配置的计算资源 (包括网络、服务器、存储、应用软件、服务) 共享池, 只需投入很少的管理

收稿日期: 2018-01-07; 修回日期: 2018-02-04

通信作者: 陈扬, bedivere14@outlook.com

基金项目: 国家自然科学基金资助项目 (No.61602509); 国家自然科学基金创新群体资助项目 (No.61521003); 国家重点研发计划基金资助项目 (No.2016YFB0800100, No.2016YFB0800101)

Foundation Items: The National Natural Science Foundation of China (No.61602509), The Innovative Research Groups of the National Natural Science Foundation of China (No.61521003), The National Key Research and Development Program of China (No.2016YFB0800100, No.2016YFB0800101)

工作就可快速获得这些资源。例如,国内的阿里云、腾讯云以及在国外已经非常成熟的微软 Azure、谷歌云平台等,各种“云计算”的应用服务范围正日渐扩大,影响力也无可估量。但云计算存在以下 2 个不可忽视的问题。

1) 数据的入侵容忍能力低。云计算最重要的 2 个问题是保密性和安全性。如企业把企业信息、客户信息等敏感的商业数据存放于云计算服务提供商,通过减少对某些数据的控制节约经济成本,但是企业也必然会担心数据放在云端是否安全以及会不会被泄密和窃取等问题。

2) 云计算基础设施的入侵容忍能力低。云计算模式下,由于所有的业务处理都在服务器端完成,所以云计算平台成为入侵者攻击的核心。另外,云计算平台自身也面临各式各样无法预料的错误。例如,2010 年上半年,Amazon 云计算服务因为人为失误和意外停电事故出现故障,致使美国东部的少量用户失去服务并导致部分数据丢失。因此云计算平台一旦被入侵或出现问题,将导致云服务提供商和用户利益受到严重损害^[1]。

为了解决上述 2 个问题,拜占庭系统模型被大量应用。拜占庭问题是构建可靠的分布式计算体系的基本问题。在这类问题中,有 $3f+1$ 个执行体,其中,最多有 f 个可能失效(拜占庭错误或被破坏),拜占庭问题的目标是让所有正常执行体在即使系统中有拜占庭错误的情况下也能达成一致。

2 相关工作

近年来,针对入侵云容忍研究的拜占庭系统,主要是在拜占庭容错(BFT, Byzantine fault tolerance)协议上进行优化改进和应用。文献[2]提出了一种拜占庭容错框架拜占庭容错云(BFT cloud),用于提高云系统的顽健性。文献[3]提出了一种名为 BFT-MCDB (Byzantine fault tolerance multi-clouds database) 的拜占庭容错模型,将拜占庭协议以及 Shamir 的秘密共享方法结合起来,以检测多云计算环境中的拜占庭故障,并确保存储在云中数据的安全性。

为了基于拜占庭提高云的入侵容忍能力,文献[3]以 Hadoop 为平台实现了 MinBFT 和 MinZyzyva

最小拜占庭算法原型。在对 2 种算法性能进行比较的基础上,改进了容忍入侵算法性能,实现了在 $2f+1$ 个副本中容忍 f 个副本的错误;尽量减少内部通信资源开销和副本数量,降低云计算资源消耗,提高系统容忍入侵性能。文献[4]提出了资源有效的拜占庭容错(REBFT, resource-efficient Byzantine fault tolerance),这种方法通过将 f 个副本保持在被动模式,在正常情况下通过最小化 BFT 系统的资源使用情况。与主动副本相比,被动副本既不参与协议也不执行客户端请求;而是通过活动副本提供的验证状态更新来提升速度。在怀疑或检测到故障的情况下,被动副本以一致的方式被激活。文献[5]提出了一个新的 BFT 中间件系统——Thema,它扩展了 BFT 和 Web 服务技术,以提供一种结构化的方法来构建应用程序开发人员可以使用的拜占庭容错,其可生存的 Web 服务和其他 Web 服务一样。它们允许拜占庭容错服务: 1) 支持 Web 服务的多层次需求; 2) 为自己的客户提供标准化的 Web 服务支持。文献[6]介绍了 BFT-WS 的设计和实现, BFT-WS 是一种用于 Web 服务的拜占庭容错中间件框架。它使用标准的 Web 服务技术构建拜占庭式容错服务,因此更适合实现互操作性。文献[7]提出了 Perpetual——一种支持拜占庭容错执行复制服务的协议,同时强制执行严格的故障隔离,永久性地实现了在长时间运行的线程中异步调用和处理远程请求的复制服务之间的交互。

但现在大多数拜占庭系统都是镜像复制的执行体,这样的系统具有一定的可靠性,但是安全性不足,往往攻击者攻破一台主机就能很容易攻破其他主机。本文围绕基于动态异构的拜占庭系统提高云服务入侵容忍能力进行研究,主要贡献如下。

1) 针对入侵云容忍下的拜占庭系统描述了潜伏式攻击模型。

2) 为提高拜占庭系统安全性提出了动态异构的拜占庭系统(DHB, dynamic heterogeneous Byzantine system)模型。

3) 为进一步提高异构系统的安全性,提出利用系统漏洞和攻击技能水平衡量执行体生命周期,并将量化结果作为系统切换时间。

4) 实验部分模拟攻击方式对改进动态异构拜占庭模型进行防御效果测试,有效评估系统安全性。

3 潜伏攻击模型

拜占庭执行体在云环境下以虚拟机的形式存在,随着时间的推移,一个虚拟机的攻击面会逐渐扩大^[8]。即使是有一定入侵容忍的拜占庭系统,执行体也可能在扩大的攻击面下被攻击者逐个攻破被攻击者而使系统失效。针对此问题构建潜伏式攻击模型。结合实际情况,做出如下假设。

假设 1 假设攻击者以虚拟机作为攻击目标,暂不考虑针对 Hypervisor、管理器以及物理设施的攻击。

假设 2 对于一些重要的云中资源,如军事机密、银行账户,攻击者已经不是独立个人,而是团体,为了攻破拜占庭系统,他们有足够的资源和资源支撑其行动。

假设 3 攻击者攻破各个执行体是相互独立事件。因为采用异构执行体,所以暂时忽略执行体之间的相互影响。

假设 4 因为拜占庭系统针对个别(不多于 f 个)的单独攻击有较好的容侵性能,所以在考虑攻击的时候仅考虑潜伏式攻击。

基于以上假设,提出一种潜伏式攻击模型。当攻破一台执行体的时候并不立刻进行破坏(因为拜占庭的入侵容忍机制),而是继续感染攻击其他执行体。此时攻击者可以知道有多少执行体,因为各个拜占庭执行体之间会进行通信,一旦攻击者控制了一台执行体就可以获得关于其他拜占庭执行体的很多信息。如果攻击者逐个依次攻击,则定义为“串行”攻击。基于攻击者组织规模,攻击者也可能进行“并行”攻击,即多个执行体

可能同时受到攻击。随着时间积累,当攻击者攻破的执行体超过了入侵容忍的限度 f ,再发起总攻,就能彻底破坏拜占庭系统。定义 n_a 为攻击者控制的执行体个数。

1) 当 $f < n_a < 2f$ 时,攻击者能使系统无法正常工作(达不成一致)。其危害性较低。

2) 当 $n_a > 2f$ 时,攻击者可以使系统一致输出错误信息。其危害性较高,这是最佳攻击目标。

4 动态异构拜占庭模型

现在大多数拜占庭系统都是镜像复制的执行体,这样,往往一台主机被攻破就能感染其他主机。所以异构执行体是拜占庭系统的发展方向。

即使采用异构方法,如果执行体一直保持静态不变,系统也存在如下安全性问题。

1) 没有完全异构的执行体,即执行体之间仍然存在共同攻击面,如共同漏洞,如表 1 所示。

2) 潜伏式攻击手段在时间积累下仍能攻破异构系统。

所以,构建上述系统需要解决 2 个问题:① 如何配置最大异构度的拜占庭执行体集;② 如何最优切换,即切换时机问题(太快会影响服务以及成本较高,慢了又可能给攻击者带来足够的攻击时间)。为解决上述问题,一个完善有效的系统应该包含执行体的异构选择和执行体的动态切换功能。

为了提高容忍性能,本文采用执行体个数为 $2f+1$ 的拜占庭模型,即系统加入了消息签名。

4.1 系统框架

本文设计了一个动态异构的拜占庭系统框架,如图 1 所示。在云中为拜占庭系统部署了切换管理、资源控制器、执行体状态、执行体选择 4 个模块。

表 1 系统的共同漏洞数

系统名称	与 Ubuntu 的共同漏洞数/个	与 Debian 的共同漏洞数/个	与 RedHat 的共同漏洞数/个	与 Solaris 的共同漏洞数/个
Ubuntu	840	303	179	1
Debian	303	995	220	1
RedHat	179	220	1 518	1
Solaris	1	1	1	100

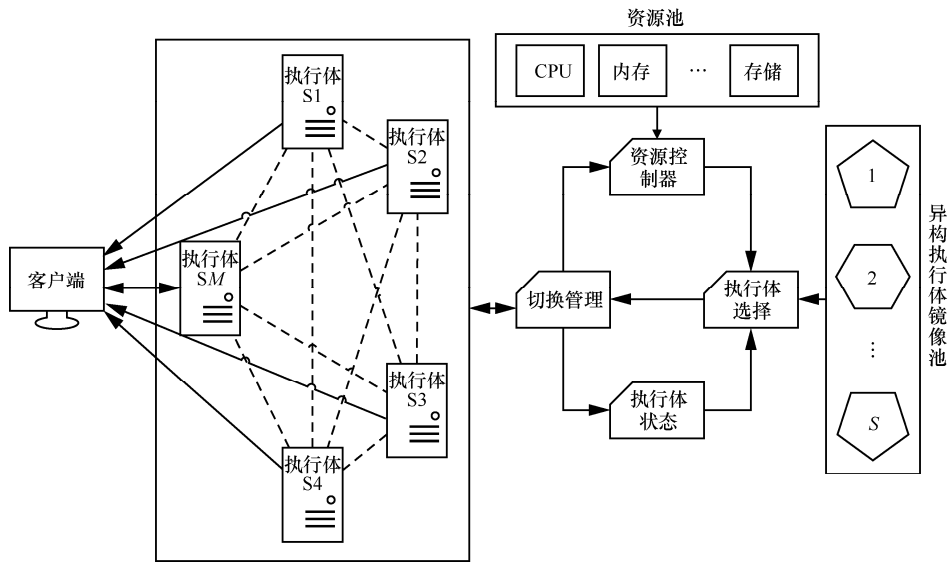


图1 动态异构的拜占庭系统框架

1) 切换管理

负责管理、生成和销毁执行体。可以用云中的系统调度模块实现，如基于事先创建的镜像快速生成可使用的执行体。

2) 资源控制器

负责管理和分配系统资源，如 CPU、内存、存储等。这是云的资源管理模块，根据用户申请进行分配。

3) 执行体状态

维持每个执行体状态表，监视每个执行体状态，可通过云管理系统实现。

4) 执行体选择

负责从异构执行体镜像池中选择执行体，通过不同的算法，自适应地使系统的异构度最大，此部分需要在云管理系统（如 OpenStack）中开发相应模块实现。

4.2 异构性最大化部署策略

在云环境中，拜占庭系统的执行体可以在多个层面上实现异构，如操作系统异构、软件异构。应用程序堆栈的配置集合表示为 $C = L_1 \times L_2 \times \dots \times L_n$ ，其中，存在 n 个技术堆栈。这里， L_i 表示可以在应用栈第 i 层使用的技术集合。

考虑一个具有 4 层 ($n=3$) 的云 Web 服务执行体，即拜占庭执行体承载的是云 Web 服务，其中，第 1 层表示 Web 服务的运行系统，第 2 层表示服务器软件，第 3 层表示其应用脚本，第 4 层表示存储处理数据的数据库。例如，第 1 层使用

的技术 $L_1 = \{\text{Windows10}, \text{RedHat}, \dots, \text{Ubuntu}\}$ 。通过表 2^[9]能够看出，一个有效的配置可以是 $c_1 = (\text{Ubuntu}, \text{Apache}, \text{PHP}, \text{MySQL})$ ，异构化配置在各个层面可以不同。

表 2

异构执行体

数据库	应用脚本	服务器软件	虚拟操作系统
MySQL	PHP	Tomcat	Windows 10
PostgreSQL	Perl	Apache	RedHat
...	...	...	...
MariaDB	ASP	IIS	Ubuntu
Oracle	JSP	Nginx	Centos

对于 $|C|$ 种配置，执行体集合 $S = \{s_1, s_2, \dots, s_{2f+1}\}$ 之间有相似性存在，如共同漏洞。定义一个相关性矩阵 R ， r_{ij} 表示 c_i 、 c_j 这 2 种配置系统之间的关联程度，即攻破操作系统 c_i 后，操作系统 c_j 被攻破的概率，且 $r_{ii}=1$ 。关联程度由共同漏洞数刻画，共同漏洞数越大，关联程度越高，对应的元素值越大，如表 1 所示。设 c_i 、 c_j 间共同的漏洞数为 K_{ij} ， c_i 本身的漏洞数为 V_i ，其中 $K_{ij} = K_{ji}$ ，令

$$r_{ij} = \frac{K_{ij}}{V_i}, \quad 1 \leq c_i, c_j \leq |C|$$

借鉴生物多样性中基于物种相似度的多样性度量方法评价配置系统之间的异构性^[10]。对于任

意配置 c_i ，令

$$w_i = \sum_{j=1}^{|C|} r_{ij} p_j, i=1,2,\dots,|C|$$

$$p_j = \frac{\left| \left\{ s_n : c_j \in s_n \right\} \right|}{2f+1}, 1 \leq n \leq 2f+1$$

p_i 为 c_i 在系统中出现的频率。则定义操作系统的多样性度量^[11]为

$$D(c) = \frac{1}{\prod_{i=1}^{2f+1} w_i^{p_i}}$$

其中， c 代表拜占庭系统配置方法。在布置异构执行体时，应该尽量使系统的异构性最大化，即求得 $D(c)_{\max}$ 。

4.3 动态执行体切换策略

执行体的动态切换是一种很好的提高安全性的方法，这种思想在国外被充分应用在移动目标防御上。云服务的资源调度为执行体的切换提供了很好的条件。当执行体运行一段时间后就需要启动新的异构执行体来替换。

为了获得最大的安全性，系统还必须满足以下要求。

1) 不考虑将执行体重置成初始状态。因为执行体如果已经被攻破，攻击者第二次攻破相同系统就会变得非常容易。

2) 在执行体切换过程中，先启动替换执行体，等其正常工作后再销毁原有执行体。

- ① 不会破坏原系统 $2f+1$ 的协议原则。
- ② 不会影响用户的正常使用。

从不同的角度考虑执行体的动态化有不同的方法。本文根据 McQueen 团队^[12]的研究成果，考

虑攻击者的能力以及每个执行体漏洞情况，基于单个执行体平均攻破时间（MTTC, mean-time-to-compromise），为每个执行体设置不同的生命周期。

4.3.1 单个执行体平均攻破时间

本文定义攻击者在执行体上获得某种级别的权限所需的平均时间为平均攻破时间。MTTC 取决于漏洞的性质和攻击者的技能水平。MTTC 被建模为由以下 3 个攻击者子进程组成的随机进程^[12]。

进程 1：当攻击者识别出一个或多个已知漏洞并且有一个或多个可利用漏洞。

进程 2：当攻击者识别出一个或多个已知的漏洞，但没有可用的可利用漏洞。

进程 3：识别新的漏洞和可利用漏洞。

进程 1

简单假设可用的漏洞均匀分布在所有漏洞中，则攻击者具有容易获得的漏洞概率 P_1 为

$$P_1 = 1 - e^{-\frac{mV}{K}}$$

$$V = 0.1V_{LR} + 0.5V_{MR} + V_{HR}$$

参数详情如表 3 所示。表 3 中数据为 McQueen 等得出的经验值，本文以此参数为案例进行分析，不同的云环境下可以以此为参照，修改相应数值以适应不同系统。规定 1 天作为进程 1 中成功攻击的平均时间^[12]。

进程 2

由于进程 1 和进程 2 是互斥的，即

$$P_2 = e^{-\frac{mV}{K}} = 1 - P_1$$

表 3

MTTC 中参数定义和取值

变量	定义	值
V	目标执行体漏洞个数	—
V_{LR}, V_{MR}, V_{HR}	分别代表低、中、高危漏洞个数	—
m	m 是攻击者已经拥有的可利用漏洞数	150（初级），250（中级），450（专家）
K	不重复漏洞的总数	9 447
V_A	攻击者根据技能水平可以利用开发的漏洞	—
V_N	攻击者技能级别无法使用的漏洞数	$V - V_A$
$\frac{V_A}{V}$	可利用漏洞率	0.3（初级），0.55（中级），1（专家）

进程 2 的平均时间估计应该取决于已知漏洞的数量以及攻击者能够找到或编写自己的漏洞利用代码以利用漏洞的概率。这被建模为一个串行过程,攻击者随机选择一个已知的漏洞,然后尝试查找或创建可利用漏洞。依据尝试所需基准平均值为 5.8 天。过程 2 的平均时间被建模为尝试次数乘以 5.8 天的预期值为

$$t_2 = 5.8N$$

其中,进程 2 预期的尝试次数 N 为

$$N = \frac{V_A}{V} \left[1 + \sum_{n=2}^{V_N+1} \left(n \prod_{i=2}^n \frac{V_N - i + 2}{V - i + 1} \right) \right] \quad (1)$$

进程 3

进程 3 会持续到“成功”。发现可用漏洞的平均时间的合理估计为 30.42 天。根据每个攻击者级别,漏洞数估计将按 $\frac{V}{V_A}$ 进行缩放。从而进程 3 的预期攻破时间 t_3 为

$$t_3 = \left(\frac{V}{V_A} - 0.5 \right) 30.42 + 5.8$$

4.3.2 切换时间

平均攻破时间,即 3 个过程的总时间为

$$t_i = t_1 P_1 + t_2 (1 - P_1)(1 - u) + t_3 (1 - P_1)u \quad (2)$$

$$u = \left(1 - \frac{V_A}{V} \right)^V$$

其中, u 是进程 2 失败的概率(如果 $V=0$,则 $u=1$)。

通过式(2)可以计算出每个执行体在不同攻击水平下的平均攻破时间 t 。本文将这个时间设置为该执行体的切换时间,即生命周期。因为在 1 个执行体的生命周期 t 内,必须至少同时有其他 f 个被攻破执行体才能对系统造成破坏。即攻击者必须在 t 内攻破 $f+1$ 个以上的执行体。这是极其困难的,故系统的安全性可以得到很好的保证。

4.3.3 切换策略

在此系统上,结合上文的分析,本文设计了拜占庭执行体的切换策略。

步骤 1 首先切换管理模块部署多个异构执行体,并为每个执行体通过式(2)计算设置一个生

命周期 t ,不同执行体的生命周期是不同的。

步骤 2 执行体状态模块监视每个执行体状态。如果拜占庭系统发现不一致的执行体,或者某个执行体生命周期即将到期,执行体状态模块要求进行执行体更换。

步骤 3 切换管理模块联系资源控制器从资源池分配新的资源,执行体选择器从异构执行体镜像池中选择执行体镜像,结合 4.2 节的异构度计算方法,它要保证其选择使系统的异构性最大。执行体状态模块为其设置一个生命周期。

步骤 4 切换管理模块将新旧执行体进行切换并保证业务不中断。

拜占庭系统具有较高的安全性,并且系统切换时间越短,消耗资源越多。所以本文将安全划分成 3 个等级:Level 1 (初级)、Level 2 (中级)、Level 3 (专家),依据攻击者的 3 个水平计算切换时间。定义执行体 i 的生命周期 $t_i = \{z_1, z_2, z_3\}$, z_i 分别对应 3 个安全等级的值。这样管理者可以依靠网络状况、安全需求等具体环境进行选择。

5 理论分析

5.1 场景设定

本节尝试分析动态异构的拜占庭系统的安全性。为了便于分析,假设除设定的防御机制外,被保护系统没有采取其他安全措施,如入侵检测、防火墙等。对于 $2f+1$ 的拜占庭系统,攻击者必须在执行体生命周期 t_i 内攻破 $f+1$ 个执行体才能攻击成功。因为并行攻击对攻击团队要求非常高,投入的资源也是极大的。所以这里仅讨论潜伏式串行攻击。

5.2 安全评估

因为攻击者成功攻破一个执行体的概率密度在零时刻为零,随着时间的增加,趋向峰值然后衰减但从未达到零。所以将 1 个执行体被攻击成功的概率密度函数建模为伽马函数^[12]。因此专家级别攻破单个执行体的概率密度函数如图 2 所示 ($t_{\text{MMTC}} = 4.5$)。

对于执行体 $i \in [1, N]$, $N = 2f + 1$, 其生命周期根据式(2)计算为 t_i ,攻击者攻破执行体 i 的时间为 T_i 。

$$T_i \sim \Gamma(k_i, \theta)$$

$$t_i = k_i \theta$$

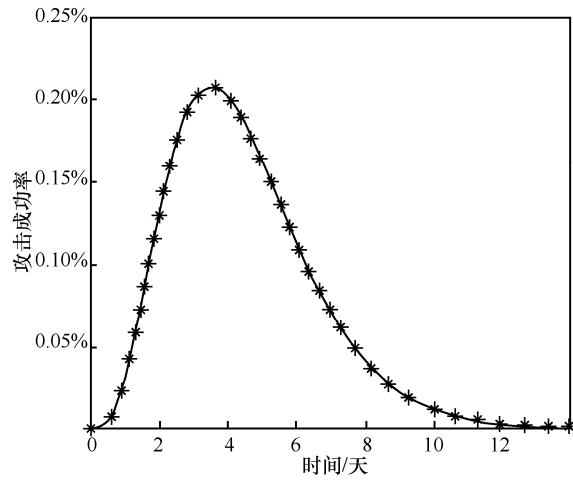


图2 专家攻破单个执行体的概率密度函数

其中, $\theta > 0$ 是刻度参数, $k_i > 0$ 是形状参数, $\Gamma(\cdot)$ 是伽马函数。因为是同一攻击者, 本文设置对各执行体攻击成功概率分布的 θ 值一样。

1) 对于单纯镜像复制拜占庭系统, 从 4.3.1 节可知, 当系统攻破一个执行体后, 攻破其他镜像执行体就只有进程 1, 即攻破时间为 $t_i = 1 (2 \leq i \leq f+1)$ 天。所以攻破总时间 T^* 为

$$T^* = T_j + f$$

由于没有执行体的切换, 随着时间的积累, 系统最终会被攻破。

2) 对于静态异构的拜占庭系统, 攻击者攻破 $f+1$ 个执行体的时间 T 服从

$$T = \sum_{j=1}^{f+1} T_j \sim \Gamma\left(\sum_{j=1}^{f+1} k_j, \theta\right)$$

明显, $T > T^*$, 即异构拜占庭系统被攻破需要的时间更长。

3) 对于动态的拜占庭系统, 其被攻破的概率分布函数为

$$F(t) = \begin{cases} \int_0^t f(t) dt, & 0 \leq t < t_i \\ 0, & t \geq t_i \end{cases} \quad (3)$$

$$t_i = \min(t_1, t_2, \dots, t_{f+1})$$

t_i 为攻击者攻击的目标执行体集合 $S^* = \{s_1, s_2, \dots, s_{2f+1}\}$ 中生命周期最小值, $f(t)$ 分别对应同构系统和异构系统被攻破时间 T^* 和 T 的概率密度函数。要想攻破动态的拜占庭系统, 攻击者必须要在 t_i 内至少攻破 $f+1$ 个执行体, 因为一旦超过

t_i , 生命周期对应为 t_i 的执行体就会被系统切换, 攻击者攻破的执行体数量小于 $f+1$, 则攻击失败。从式(3)可以看出, 切换时间越短, 被攻破的可能性越小。

6 仿真分析

基于第 5 节的 4 种系统 (镜像复制拜占庭系统、静态异构拜占庭系统、动态同构拜占庭系统、动态异构拜占庭系统) 架构下在潜伏式攻击中安全性能的数值解, 假设系统为 5 个执行体的拜占庭系统, 仿真专家级攻击者对该系统的攻击。

本文假设攻击者目标执行体集合 $S^* = \{s_1, s_2, s_3\}$, 其中, s_1 的生命周期最短。由式(3)计算得 $t_i = \{4.5, 10, 20\}$, 专家攻击者对于 S^* 执行体集合的攻击成功率如图 3 所示。在 level2 的安全等级下, 专家级攻击者必须在 10 天内攻破 3 个执行体 S^* , 一旦超时, s_1 会被系统切换, 攻击者必须更换目标集合, 即攻击失败。从图 3 可以看出, 安全等级越高, 系统被攻破的可能性越小。

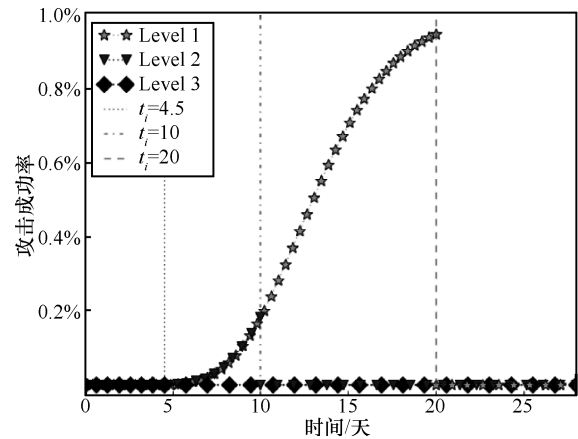


图3 专家在不同安全等级下攻破系统的概率分布

图 4 比较了在安全等级 Level 1 环境下专家级别攻击者对有 5、7、9 个执行体的动态异构拜占庭系统的攻击成功率。可以看出, 异构执行体越多, 系统的安全性越高。所以管理者应该根据自身需求选择拜占庭系统配置。

图 5 给出了 4 种系统的攻破概率分布。可以看出, 在安全等级 Level 3 环境下专家级别攻击者对 S^* 进行进攻, 动态异构的防御性能有较高的安全性, 攻破概率几乎为 0, 这个与预期的结果一

致。HB 和 B 系统的 2 条线所包围的面积代表了系统的异构度范围。镜像复制拜占庭系统异构度最小, 图中 HB 系统代表“理想”的完全异构(任何 2 个系统不可能完全不同)。随着异构度的增大, 攻击成功曲线向右移动, 即攻破需要的时间也随之增加。

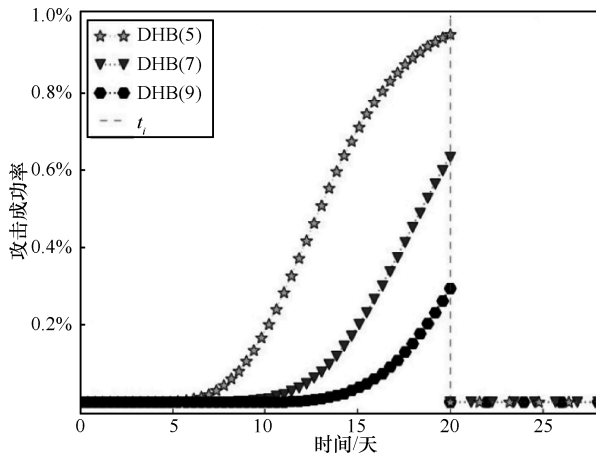


图4 执行体个数对系统安全性影响

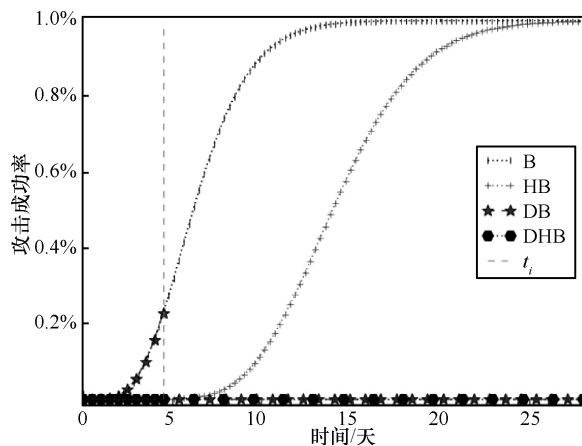


图5 专家攻破4种系统的概率分布

7 结束语

本文针对现有云环境下的拜占庭系统构建了一种潜伏式攻击模型, 并提出了一种有效应对此攻击的防御系统, 即动态异构拜占庭系统。DHB 系统主要由异构执行体组成, 运行拜占庭协议, 由控制器进行动态切换。针对攻防过程的复杂性, 简化了攻防模型, 在潜伏式攻击下对系统的安全性进行了理论分析。理论与实践结果均表明, 在一定程度上增大系统资源开销的情况下, DHB 系统能够显著提升被保护系统的安全性, 提升入

侵容忍能力。

当然, 本文着重介绍动态异构拜占庭基本框架, 下一步是实现了一个基于 DHB 系统的 Web 服务器模型, 并对其展开攻防测试。本文主要思想不局限于仅通过 MTTC 求解切换时间, 而是在确定切换时间时需要考虑攻击模型, 将来可以使用其他攻击模型以研究动态切换。拜占庭系统提高入侵容忍的同时也带来了系统成本的提高, 执行体动态化更是增大了系统开销, 所以将系统开销纳入切换策略中也是个很好的研究方向。

参考文献:

- [1] 王秀苹. 云计算下基于拜占庭算法的容忍入侵技术研究[D]. 包头: 内蒙古科技大学, 2012.
WANG X P. Research on intrusion tolerance based on Byzantine algorithm in cloud computing[D]. Baotou: Inner Mongolia University of Science and Technology, 2012.
- [2] ZHANG Y, ZHENG Z, LYU M R. BFTCloud: a Byzantine fault tolerance framework for voluntary-resource cloud computing[C]// International Conference on Cloud Computing. 2011: 444-451.
- [3] ALZAIN M A, SOH B, PARDEDE E. A Byzantine fault tolerance model for a multi-cloud computing[C]//International Conference on Computational Science and Engineering. 2013: 130-137.
- [4] KAPITZA R, BEHL J, CACHIN C, et al. CheapBFT: resource-efficient Byzantine fault tolerance[J]. IEEE Transactions on Computers, 2016, 65(9): 2807-2819.
- [5] MERIDETH M G, IYENGAR A, MIKALSEN T, et al. Thema: Byzantine-fault-tolerant middleware for Web-service applications[C]// IEEE Symposium on Reliable Distributed Systems. 2005: 131-142.
- [6] ZHAO W. BFT-WS: A Byzantine fault tolerance framework for Web services[C]//Eleventh International IEEE Edoc Conference Workshop. 2007: 89-96.
- [7] PALLEMULLE S L, THORVALDSSON H D, GOLDMAN K J. Byzantine fault-tolerant Web services for n-tier and service oriented architectures[C]//The International Conference on Distributed Computing Systems. 2008: 260-268.
- [8] PENG W, LI F, HUANG C T, et al. A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces[C]//IEEE International Conference on Communications. 2014: 804-809.
- [9] 全青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报, 2017, 28(4): 883-897.
TONG Q, ZHANG Z, ZHANG W H, et al. The design and implementation of mimic defense Web server[J]. Journal of Software, 2017, 28(4): 883-897.
- [10] LEINSTER T, COBBOLD C A. Measuring diversity: the importance of species similarity[J]. Ecology, 2012, 93(3): 477.

- [11] ZHANG M, WANG L, JAJODIA S, et al. Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks[J]. IEEE Transactions on Information Forensics & Security, 2016, 11(5): 1071-1086.
- [12] MCQUEEN M A, BOYER W F, FLYNN M A, et al. Time-to-compromise model for cyber risk reduction estimation[M]//Quality of Protection. Berlin: Springer, 2006: 49-64.



刘文彦（1986-），男，河南周口人，博士，国家数字交换系统工程技术研究中心研究员，主要研究方向为云计算、网络安全。

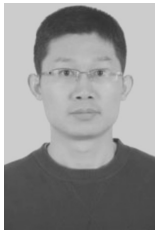
[作者简介]



陈扬（1994-），男，四川南充人，国家数字交换系统工程技术研究中心硕士生，主要研究方向为云计算、入侵容忍、网络安全。



霍树民（1985-），男，山西长治人，博士，国家数字交换系统工程技术研究中心研究员，主要研究方向为云计算、网络安全。



扈红超（1982-），男，河南商丘人，博士，国家数字交换系统工程技术研究中心研究员，主要研究方向为云计算、网络安全。



梁浩（1987-），男，河南郑州人，博士，国家数字交换系统工程技术研究中心研究员，主要研究方向为云计算、网络安全。