·热点与综述· 文章编号:1000-3428(2021)05-0030-06 文献标志码:A 中图分类号:TP393.08

面向拟态防御系统的高阶异构度大数判决算法

魏 帅[1],张辉华[2],苏 野[3],薛鹏飞[3],闻 亮[3]

(1.战略支援部队信息工程大学信息技术研究所,郑州 450003;2.无锡市新吴区保密技术服务中心,江苏 无锡 214000;
3.天津市滨海新区信息技术创新中心,天津 300450)

摘 要: 异构系统之间的相似性会影响拟态防御系统异构执行体的选择和调度。目前多数针对执行体的异构性分析仍停留在二阶层面,但二阶异构度不适用于执行体数目较多的情况,难以准确评估拟态系统的防御能力。针对该问题,提出高阶异构度的概念,分析高阶异构的性质,解释高阶异构性在执行体调度及判决中所起的重要作用,并将其用于拟态防御系统安全性的量化分析。在此基础上,设计基于高阶异构度的大数判决算法,同时依据容斥原理计算系统失效率。实验结果表明,该算法可有效分析拟态系统的防御能力,并且随着执行体相似度增大以及执行体数目增多,准确性能够进一步提高。

关键词: 高阶异构度;拟态防御;大数判决;执行体调度;冗余执行体

开放科学(资源服务)标志码(OSID):

中文引用格式: 魏帅,张辉华,苏野,等.面向拟态防御系统的高阶异构度大数判决算法[J].计算机工程,2021,47(5):30-35.

英文引用格式: WEI Shuai,ZHANG Huihua,SU Ye,et al.Majority voting algorithm based on high-order heterogeneity for mimic defense system[J].Computer Engineering,2021,47(5):30-35.

Majority Voting Algorithm Based on High-Order Heterogeneity for Mimic Defense System

WEI Shuai[1],ZHANG Huihua[2],SU Ye[3],XUE Pengfei[3],WEN Liang[3]

(1.Information Technology Research Institute,PLA Strategy Support Force Information Engineering University,Zhengzhou 450003,China;
2.Wuxi Xinwu Confidential Technology Service Center,Wuxi,Jiangsu 214000,China;
3.Tianjin Binhai Information Technology Innovation Center,Tianjin 300450,China)

【Abstract】 In mimic defense system, the selection and scheduling of heterogeneous executors is influenced by the similarity between heterogeneous systems, but most of the existing analysis of the heterogeneity of executors is still at 2-order, which often fails to deal with the mimic defense systems with a large number of executors and reduces the accuracy of system performance analysis. To solve the problem, this paper proposes the concept of high-order heterogeneity and its properties, explains the important role it plays in the scheduling and decision of executors, and makes a quantitative analysis of high-order heterogeneity and system security. On this basis, this paper proposes the design of a majority decision algorithm based on high-order heterogeneity, and calculates the system failure rate according to the inclusion-exclusion principle. The experimental results show that the algorithm can accurately analyze the defense performance of mimic defense systems, and its accuracy grows along with the similarity between executors and the number of executors.

【Key words】 high-order heterogeneity;mimic defense;majority voting;executor scheduling;redundant executor

DOI: 10. 19678/j. issn. 1000-3428. 0058718

基金项目: 国家自然科学基金(61521003);国家核高基重大专项(2017ZX01030301);上海市经信委信息化发展专项"大数据发展"(201701046)。

作者简介: 魏 帅(1984—),男,副教授、博士,主研方向为信息安全、嵌入式系统、高性能与分布式计算;张辉华、苏 野、薛鹏飞、闻 亮,工程师、硕士。

收稿日期: 2020-06-22 **修回日期:** 2020-08-14 **E-mail:** weis0906@163.com

0 概述

2010年“震网”病毒的攻击对伊朗核电站造成重大破坏,导致放射性物质泄漏,严重威胁了社会公共安全。这一事件标志着网络攻击对象已从传统的计算机网络拓展到工业控制系统。2013年斯诺登披露的“棱镜门”等项目,也证实了利用信息系统的漏洞后门能够进行相应的窃听和监控,触动国家安全和个人隐私的红线。

针对网络空间安全易守难攻的局面,有学者提出移动目标防御[1]的方法,其采用主机突变[2]、有效地址突变[3]、IP地址随机化[4]、端口随机化[5]和加密随机化[6]等技术,并且成功应用于软件定义网络[7-9]中。借鉴移动目标防御思路,邬江兴等人进一步提出了拟态防御的思想[10-12],并将其应用于网络操作系统[13]、路由器[14]和工业控制设备[15]等。攻击者利用漏洞通常依赖相似、确定和静态的执行环境,而拟态技术采用动态异构冗余的思路打破系统的相似性、确定性和静态性,使得后门和漏洞难以被利用[16]。拟态技术通过异构冗余执行体执行相同任务,并对执行结果进行比较,以此来确保结果的正确性,其依据的原理就是异构执行体不会或者较少产生共模错误。事实上,随着开源和敏捷开发等技术的发展,不同的异构执行体难免存在相似性。以应用软件为例,开源社区如github上有许多优质的软件,经过许多用户的实用和测试,代码比较稳定可靠。为节省时间和降低成本,越来越多的开发者倾向于采用开源代码进行系统开发,这就造成了不同系统应用之间存在相似性。硬件也是如此,如在设计CPU的过程中也采用了大量的公用IP,使得不同的硬件之间可能存在相似性。

现有对异构性和拟态调度之间关系的研究,基本都是采用二阶相似性进行评估。文献[17]采用二阶相似度累加即$\sum_i\sum_j \mathrm{dif}_{ij}$($i$、$j$为异构冗余系统中的执行体且$i\neq j$,$\mathrm{dif}_{ij}$代表执行体$i$和$j$之间的异构度)的方法评判系统的异构性。文献[18]提出了2种相异性组件选择算法,分别是最长相异性(Maximum Dissimilarity, MD)距离算法和最佳平均相异性(Optimal Mean Dissimilarity, OMD)距离算法。文献[19]提出了基于随机种子的调度方法,利用木桶原理确保被调度执行体的异构度不超过某一个值。文献[20]参考多样性指标,结合物种之间的差异二次熵给出一种新的异构性评价标准,并得出异构性不随执行体数目增加而减小的结论。

现有拟态防御系统的判决策略大多只引入异构性指标并以此建立调度模型[21],在调度策略中仅能保证执行体的异构性而缺乏对异构性和安全性之间的量化分析,并且没有考虑高阶共生漏洞给安全性带来的影响。本文分析影响系统异构性的关键因素,定义高阶异构度的概念,指出高阶异构性在多执行体中起到重要作用,并通过对攻击模型的分析,依据容斥原理给出系统失效率计算方法,从而实现对拟态系统防御能力的准确分析。

1 拟态防御系统执行体的高阶相似性

典型的拟态防御架构如图1所示,其中,外界输入通过输入代理分发给异构执行体,异构执行体执行相同的任务并将结果发送给调度器,调度器采用大数判决策略对执行结果进行判决,最终产生一个认为是正确的输出结果,调度器可以根据各执行体的表现进行相应的反馈控制,如某个执行体产生错误则对其进行清洗,待清洗完成后再将其加入工作队列。在此过程中,判决相关参数和系统运行状态信息可通过反馈控制器进行控制和查看。

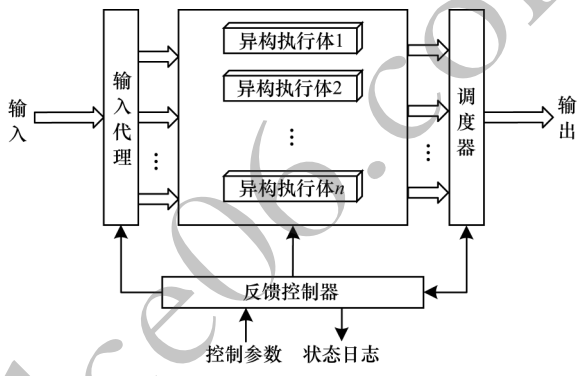

图1 拟态防御典型架构

Fig.1 Typical mimic defense architecture

虽然构建拟态系统的最佳方法是采用完全异构的异构执行体,但是随着开源技术的发展,越来越多的软硬件采用敏捷开发技术进行系统集成,通过复用成熟的构件,可以有效地降低开发成本,提升系统的可靠性。然而,代码复用技术的发展也对异构执行体的选择造成了困扰。对拟态防御系统而言,不同的执行体使用同样的构件,有可能带来同源漏洞。通过源码分析、溯源(即软件基因解析)等技术可以分析不同执行体组件之间的相关性,并以此预估相同漏洞出现的概率。一般而言,复用的共同代码越多,相似性越大,出现的共同漏洞就越多。

假定执行体由一系列组件构成,如CPU、操作系统、中间件、应用程序等,每个组件又由若干个构件组成,如应用程序可分为多个模块,构件是执行体组成的最小单元,具有原子性(不可分割性),并且实现方式各不相同。而在实现过程中,不可避免地会出现不同执行体中的组件采用相同构件的情况,这种情况下执行体之间就存在相似性。由于现实情况中各个组件之间需要整合,因此在执行体组成时需要

考虑组件的成熟度、整体运行效率等因素。此外，组件之间的组合是受限的，如某种应用程序只适配特定类型的中间件，此时执行体的组合受限，难免存在相似性。文献[17-19]以矩阵来表示构件之间的相似性。在高阶相似度中采用矩阵进行存储虽然容易查找，但存储密度太小，如有N个构件的系统，其N阶相似性指标只有1个，但是按照矩阵方式存储，则其矩阵大小为$N\times N$。本文以向量的方式进行相似性表示，n阶相似向量$\boldsymbol{\rho}^n=\{\rho_1^n,\rho_2^n,\cdots,\rho_{C_N^n}^n\}$所对应的组合为：

$$C=\begin{pmatrix} Z_1 & Z_2 & \cdots & Z_{n-1} & Z_n \\ Z_1 & Z_2 & \cdots & Z_{n-1} & Z_{n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ Z_{N-n+1} & Z_{N-n+2} & \cdots & Z_{N-1} & Z_N \end{pmatrix}$$

其中，ρ_1^n代表C中第1行组合的相似性，如ρ_1^2代表执行体Z_1和执行体Z_2的相似性，ρ_1^3表示执行体Z_1、执行体Z_2和执行体Z_3的共有相似性，ρ_i^n代表C中第i行组合的相似性。

不同阶的相似性不存在必然的联系，可以根据基因分析和代码统计等技术进行代码复用率估计，进而计算相似性，但其中存在着严格的不等式关系，即高阶相似性小于其低阶子集的相似性，如组件集合{A,B,C}的三阶相似性必然小于{A,B}、{A,C}或者{B,C}的相似性，因此，可以得到高阶相似性的以下性质及推论：

性质1 高阶相似性小于其低阶子集的相似性，如$\rho_1^3<\rho_1^2$。

推论1 如果高阶相似性不为0，则低阶相似性必不为0；如果低阶相似性为0，则高阶相似性必为0。

2 基于高阶异构度的大数判决算法

本文提出基于高阶异构度的大数判决算法(MVAHH)，其判决策略如下：

1)将输出结果一致的执行体划分为一个组G_k，所有的k组成集合K。

2)如果存在G_i，对于所有$j\in K,j\neq i$，都有$|G_i|>|G_j|$，则选择G_i的结果作为最终结果。

3)选取集合中元素数目最多的集合组G_{m_p}，评估$|G_{m_p}|$相似性，选取集合组中相似性最低的作为最终输出结果。

4)对产生错误的执行体进行清洗恢复。

如果采用大数判决，使得系统中大多数执行体相似性为0，没有共同漏洞/后门，则系统被攻破的概率为0。假定采用大数判决的拟态系统中有N个执行体，其$\left\lfloor\frac{N+1}{2}\right\rfloor$阶相似性均为0，则系统安全性为1。如三模冗余系统二阶相似性为0，如果五模冗余系统三阶相似性为0，则系统安全性为1。

据统计，平均1 000行~1 500行代码程序员就会留下一个漏洞，可见信息系统有巨大的安全隐患。假设网络攻击针对不同漏洞/后门的攻击会产生不同的输出表现，只有针对高阶漏洞/后门的攻击才会产生共用的输出表现，则当N模执行体$\left\lfloor\frac{N+1}{2}\right\rfloor$阶相似性为0时，就不会有$\left\lfloor\frac{N+1}{2}\right\rfloor$个以上执行体产生同样的错误结果，根据大数判决策略，一定会判决出正确的结果，所以，系统的安全性为1。

在选取执行体数目时，N通常取单数，如果N为偶数，则其要求和$(N-1)$模执行体相同，但是多了一个执行体。假如拟态系统中存在4个冗余执行体，不存在三阶相似性，但存在二阶相似性，当发生针对二阶漏洞的攻击时，如针对执行体1和执行体2的共同漏洞，执行体1和执行体2的输出结果一致，执行体3和执行体4的输出结果一致，投票比例2∶2，则无法通过大数判决判断出哪个才是可信结果。

本文以5个执行体的拟态架构为例，采用上文提出的MVAHH算法。输入代理将输入分为5份发给5个执行体，5个执行体同时进行数据处理，并将处理结果输出到仲裁器进行结果表决，当系统中所有执行体结果均不相同时，按照平均1 000行~1 500行代码程序员就会留下一个漏洞的规律，优先信任代码量较小的执行体，系统整体执行流程如下所示：

以S1~S4表示五执行体拟态系统分别在5个、4个、3个、2个执行体有效情况下的仲裁算法，以及根据仲裁结果的跳转情况。

1)初始时默认3个执行体工作，N=5，清洗集合C为空。若存在5个执行体输出结果一致，跳转至S1；若存在4个执行体输出结果一致，将1个结果不同的执行体加入到清洗集合C，跳转至S2；若存在3个执行体输出结果一致，将2个结果不同的执行体加入到清洗集合C，跳转至S3；若存在2个执行体输出结果一致，且存在2个以上的二元素集合G_i、G_j比较，则选取异构度小的作为最终结果进行输出，将3个结果不同的执行体K加入到清洗集合C，跳转至S2；若所有执行体输出结果一致，选取代码量较小的执行体作为最终结果进行输出，将4个结果不同的执行体K加入到清洗集合C，跳转至S5。

2)配置执行体集合C中的执行体进行清洗，若某个执行体清洗完成，跳转至S1；若存在4个执行体输出结果一致，跳转至S2；若存在3个执行体输出结果一致，将1个结果不同的执行体加入到清洗集合C，跳转至S3；若存在2个执行体输出结果一致，且存在2个以上的二元素集合G_i、G_j比较，则选取异构度小的作为最终结果进行输出，将2个结果不同的执行体加入到清洗集合C，跳转至S4；若所有执行体输

出结果一致,选取代码量较小的执行体作为最终结果进行输出,将3个结果不同的执行体加入到清洗集合C,跳转至S5。

3)配置执行体集合C中的执行体进行清洗,若某个执行体清洗完成,跳转至S2;若存在3个执行体输出结果一致,跳转至S3;若存在2个执行体输出结果一致,将1个结果不同的执行体加入到清洗集合C,跳转至S4;若所有执行体输出结果一致,选取代码量较小的执行体作为最终结果进行输出,将2个结果不同的执行体加入到清洗集合C,跳转至S5。

4)配置执行体集合C中的执行体进行清洗,若某个执行体清洗完成,跳转至S3;若存在2个执行体输出结果一致,跳转至S4;若所有执行体输出结果一致,选取代码量较小的执行体作为最终结果进行输出,将1个结果不同的执行体加入到清洗集合C,跳转至S5。

5)配置执行体M进行清洗,若某个执行体清洗完成,则跳转至S4。输出执行体输出结果。

3　系统失效率计算

采用大数判决算法的拟态系统被攻破的概率等于虽然产生错误但是未被判决出来的错误概率之和,也等于高阶相似构件所产生的概率之和。为简要说明问题而不失一般性,假定所有执行体实现的代码量和产生漏洞的概率基本相同,均为1。根据执行体的n阶相似性,结合容斥原理公式,可以计算得到采用大数判决策略时拟态系统被攻破的概率。

一般情况下的容斥原理公式如下所示:

$$\left|\bigcup_{i=1}^{n}A_i\right|=\sum_{i=1}^{n}\left|A_i\right|-\sum_{i,j:\ 1\leqslant i<j\leqslant n}\left|A_i\cap A_j\right|+\sum_{i,j,k:\ 1\leqslant i<j<k\leqslant n}\left|A_i\cap A_j\cap A_k\right|-\cdots+(-1)^{n-1}\left|A_1\cap A_2\cap\cdots\cap A_k\right| \tag{1}$$

以五模执行体为例,根据高阶相似性矩阵推出系统失效概率。其中,一阶以上错误概率可以参照式(1)求出,二阶漏洞后门可以根据递归原理看作C_5^2个元素$A_1\cap A_2$、$A_1\cap A_3$、$A_1\cap A_4$、$A_1\cap A_5$、$A_2\cap A_3$、$A_2\cap A_4$、$A_2\cap A_5$、$A_3\cap A_4$、$A_3\cap A_5$、$A_4\cap A_5$,并利用式(1)求解,得到漏洞后门数$\sum_i\rho_i^3-3\sum_i\rho_i^4+6\rho_1^5$。同理可以得到五模执行体各阶漏洞数和判决结果,如表1所示。其中,3个以上结果错误会导致系统产生错误结果,按照大数判决算法,会对系统整体产生影响,则判决出现错误的总概率为:

$$P=\frac{\sum_i\rho_i^3-3\sum_i\rho_i^4+6\rho_1^5}{5-\sum_j\rho_j^2+\sum_k\rho_k^3-\sum_l\rho_l^4+\rho_1^5}$$

表1　五模执行体各阶漏洞数和判决结果

Table 1　Number of vulnerabilities and voting result for each level of 5-executors

结果	可能存在的漏洞数	判决结果
5个执行体错误相同	ρ_1^5	错误
4个以上执行体错误相同	$\sum_i\rho_i^4-4\rho_1^5$	错误
3个以上执行体错误相同	$\sum_i\rho_i^3-3\sum_i\rho_i^4+6\rho_1^5$	错误
2个以上执行体错误相同	$\sum_i\rho_i^2-2\sum_i\rho_i^3+6\sum_i\rho_i^4-20\rho_1^5$	正确
1个以上执行体错误相同	$5-\sum_j\rho_j^2+\sum_k\rho_k^3-\sum_l\rho_l^4+\rho_1^5$	正确

4　实验验证

实验对MVAHH判决算法和基于二阶异构度的最大异构度算法进行比较,实验程序通过MATLAB编写,执行体之间的相似度随机生成。利用蒙特卡洛方法进行模拟测试,观察拟态系统的失效概率。为简化结果并不失一般性,本文采用相对错误率来衡量系统的失效概率,即系统多阶漏洞占执行体内总漏洞的比重,这个比重可能偏高,因为通常发现漏洞都较为困难,真实的攻击成功概率需要再乘以一定的系数(漏洞发现率),但不影响分析高阶漏洞在整体系统中的作用。

假定执行体数目为3,二阶相似度为50%、25%、10%、5%情况下的系统失效率如图2~图5所示(彩色效果见《计算机工程》官网HTML版),其中,MVAHH判决算法结果用蓝色表示,基于二阶异构性的最大异构度算法用红色表示。可以看出,随着相似性的增大,最大差异度算法和本文算法的差距逐渐增大,这是因为当相似性较大时,更容易出现三阶相似性不为0的情况,此时采用基于二阶异构度的最大异构度算法,就会造成三阶异构度的重复计算,造成结果产生较大的失真。由此可知,当相似性增加时,需要更多地考虑高阶相似性,否则会造成较大失真。

假定执行体数目为5,二阶相似度为50%、25%、10%、5%情况下的系统失效率如图6~图9所示(彩色效果见《计算机工程》官网HTML版),其中,MVAHH判决算法结果用蓝色表示,基于二阶异构性的最大异构度算法用红色表示。可以看出,与三阶情况类似,随着相似性的增大,最大差异度算法和本文算法的差距逐渐增大,这是因为当相似性较大时,更容易出现三阶以上相似性不为0的情况,此时采用基于2阶异构度的最大异构度算法会造成结果产生较大的失真。比起相同相似性的三执行体系统,可以看出其系统被攻破的概率明显减小,当二阶相似度为5%时,有多种选择可以使系统被攻破概率为0。

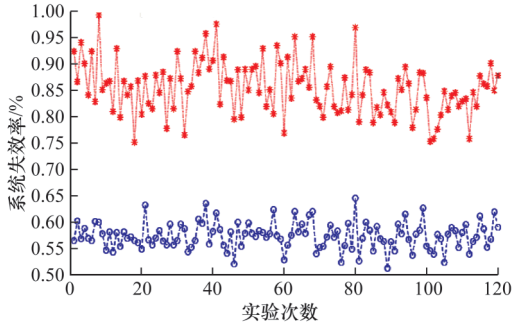

图2 二阶相似度为50%的三执行体系统失效率

Fig.2 System failure probability of 3-executor system whose 2-level similarity is 50%

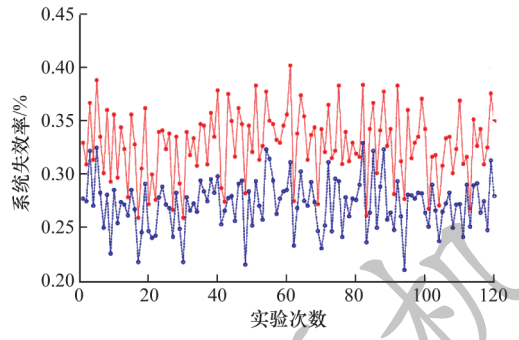

图3 二阶相似度为25%的三执行体系统失效率

Fig.3 System failure probability of 3-executor system whose 2-level similarity is 25%

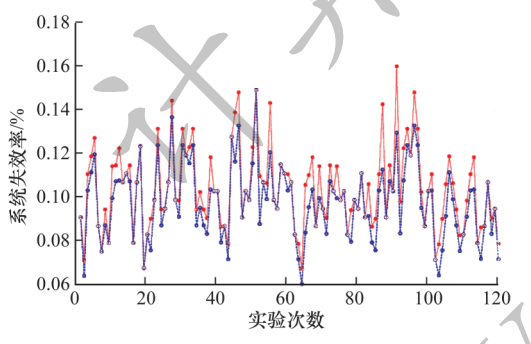

图4 二阶相似度为10%的三执行体系统失效率

Fig.4 System failure probability of 3-executor system whose 2-level similarity is 10%

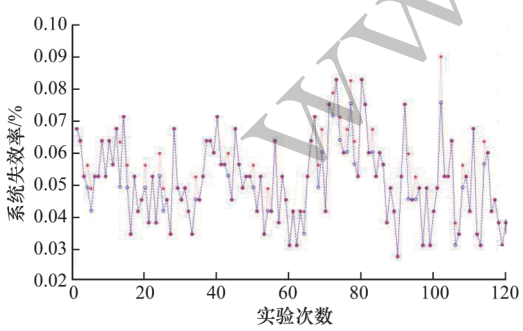

图5 二阶相似度为5%的三执行体系统失效率

Fig.5 System failure probability of 3-executor system whose 2-level similarity is 5%

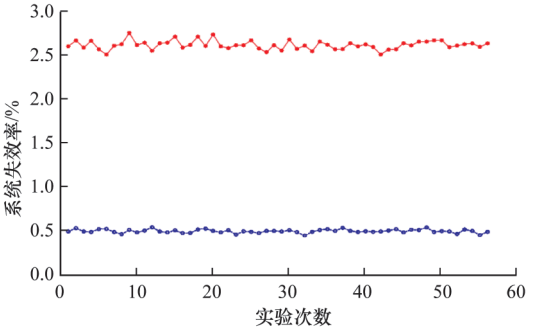

图6 二阶相似度为50%的五执行体系统失效率

Fig.6 System failure probability of 5-executor system whose 2-level similarity is 50%

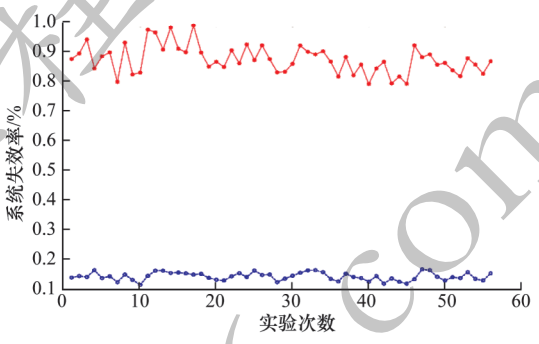

图7 二阶相似度为25%的五执行体系统失效率

Fig.7 System failure probability of 5-executor system whose 2-level similarity is 25%

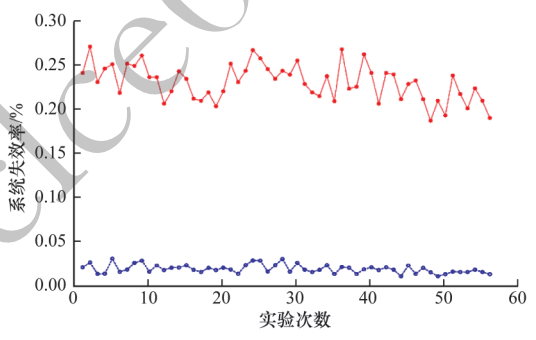

图8 二阶相似度为10%的五执行体系统失效率

Fig.8 System failure probability of 5-executor system whose 2-level similarity is 10%

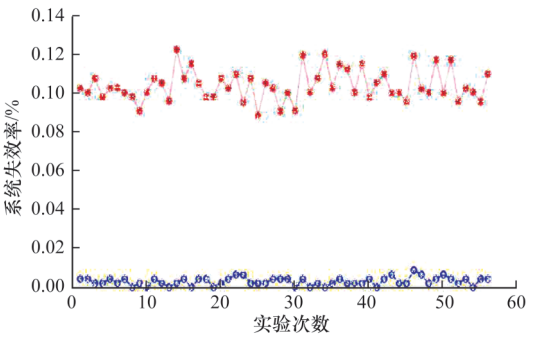

图9 二阶相似度为5%的五执行体系统失效率

Fig.9 System failure probability of 5-executor system whose 2-level similarity is 5%

5 结束语

本文基于拟态防御架构,结合执行体的异构特点分析拟态防御系统中裁决器的判决算法,总结现有基于二阶异构度的调度算法存在的不足,提出一种基于高阶异构度的大数判决算法用于评估拟态系统性能。实验结果表明,该算法能够准确分析系统的安全性和可靠性。本文对于攻击模型的分析和假设还不够完善,后续将进一步分析网络攻击模型及其对判决结果的影响,提出更具通用性的判决调度算法。

参考文献

[1] ZHENG J J, NAMIN A S. A survey on the moving target defense strategies:an architectural perspective[J]. Journal of Computer Science and Technology, 2019, 34(1):207-233.

[2] JAFARIAN J H, AL-SHAER E, DUAN Q. OpenFlow random host mutation:transparent moving target defense using software defined networking[C]//Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks. New York, USA:ACM Press, 2012:127-132.

[3] JAFARIAN J H, AL-SHAER E, DUAN Q. An effective address mutation approach for disrupting reconnaissance attacks[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12):2562-2577.

[4] JAFARIAN J H, AL-SHAER E, DUAN Q. Adversary-aware IP address randomization for proactive agility against sophisticated attackers[C]//Proceedings of 2015 IEEE Conference on Computer Communications. Washington D. C., USA:IEEE Press, 2015:738-746.

[5] LUO Yuebin, WANG Baosheng, CAI Guilin. Effectiveness of port hopping as a moving target defense[C]//Proceedings of the 7th International Conference on Security Technology. Washington D. C., USA:IEEE Press, 2014:7-10.

[6] AZAB M, ELTOWEISSY M. ChameleonSoft:software behavior encryption for moving target defense[J]. Mobile Networks & Applications, 2013, 18(2):271-292.

[7] CHIANG M L, HSIEH H C, WANG C W. Improving the fault-tolerance under software-defined network based on new sight of agreement protocol[J]. IEEE Access, 2018, 6:40898-40908.

[8] SAKIC E, DERIC N, KELLERER W. MORPH:an adaptive framework for efficient and Byzantine fault-tolerant SDN control plane[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(10):2158-2174.

[9] YUAN Bin, JIN Hai, ZOU Deqing, et al. A practical Byzantine-based approach for faulty switch tolerance in software-defined networks[J]. IEEE Transactions on Network and Service Management, 2018, 15(2):825-839.

[10] HU Hongcha, WU Jiangxing, WANG Zhenpeng, et al. Mimic defense: a designed-in cybersecurity defense framework[J]. IET Information Security, 2017, 12(3):226-237.

[11] WU Jiangxing. Construction of endogenous security in national information network space with mimic defense technology[J]. Information and Communication Security, 2019, 13(6):4-6. (in Chinese)
邬江兴. 拟态防御技术构建国家信息网络空间内生安全[J]. 信息通信技术, 2019, 13(6):4-6.

[12] WU Jiangxing. Research on cyber mimic defense[J]. Journal of Cyber Security, 2016, 1(4):1-10. (in Chinese)
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.

[13] HU Hongchao, WANG Zhenpeng, CHENG Guozhen. MNOS:a mimic network operating system for software defined networks[J]. IET Information Security, 2017, 11(6):345-355.

[14] MA Hailong, JIANG Yiming, BAI Bin, et al. Tests and analyses for mimic defense ability of routers[J]. Journal of Cyber Security, 2017, 2(1):43-53. (in Chinese).
马海龙, 江逸茗, 白冰, 等. 路由器拟态防御能力测试与分析[J]. 信息安全学报, 2017, 2(1):43-53.

[15] WEI Shuai, YU Hong, GU Zeyu, et al. Architecture of mimic security processor for industry control system[J]. Journal of Cyber Security, 2017, 2(1):54-73. (in Chinese)
魏帅, 于洪, 顾泽宇, 等. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报, 2017, 2(1):54-73.

[16] WU Jiangxing. Meaning and vision of mimic computing and mimic security defense[J]. Telecommunications Science, 2014, 30(7):2-7. (in Chinese)
邬江兴. 拟态计算与拟态安全防御的原意和愿景[J]. 电信科学, 2014, 30(7):2-7.

[17] SHEN Conglin, CHEN Shuangxi, WU Chunming, et al. Adaptive mimic defensive controller framework based on reputation and dissimilarity[J]. Journal of Communications, 2018, 39(S2):173-180. (in Chinese)
沈丛麒, 陈双喜, 吴春明, 等. 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(S2):173-180.

[18] YAO Wenbin, YANG Xiaozong. Design of selective algorithm for diverse software components[J]. Journal of Harbin Institute of Technology, 2003, 35(3):261-264. (in Chinese)
姚文斌, 杨孝宗. 相异性软件组件选择算法设计[J]. 哈尔滨工业大学学报, 2003, 35(3):261-264.

[19] LIU Qinrang, LIN Senjie, GU Zeyu. Heterogeneous redundancies scheduling algorithm for mimic security defense[J]. Journal of Communications, 2018, 39(7):188-198. (in Chinese)
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):188-198.

[20] ZHANG Jiexin, PANG Jianmin, ZHANG Zheng. Quantification method for heterogeneity on Web server with mimic construction[J]. Journal of Software, 2020, 31(2):564-577. (in Chinese)
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, 31(2):564-577.

[21] WU Zhaoqi, ZHANG Fan, GUO Wei, et al. A mimic arbitration optimization method based on heterogeneous degree of executors[J]. Computer Engineering, 2020, 46(5):12-18. (in Chinese)
武兆琪, 张帆, 郭威, 等. 一种基于执行体异构度的拟态裁决优化方法[J]. 计算机工程, 2020, 46(5):12-18.

编辑 金胡考