面向主动防御的多样性研究进展

仝 青[1], 郭云飞[1], 霍树民[1], 王亚文[1]
[1]战略支援部队信息工程大学 郑州 中国 450002

摘要 不同软件或执行过程通常存在不同的脆弱性, 多样性技术基于该前提应用于系统的可靠性、安全性设计中, 显著增强了系统的防御能力和入侵容忍能力, 然而也存在系统代价高、复杂性高等不足。已有研究中出现了大量的多样性技术实现、系统设计以及相关的评估工作, 覆盖范围广泛。针对主动防御领域内的多样性应用, 围绕多样性应用性价比的问题, 本文梳理了多样性研究中的典型工作和最新进展。首先对多样性综述研究工作进行了对比分析, 讨论了多样性研究的主要内容和研究侧重点。其次对多样性概念进行了梳理, 给出了时、空多样性的定义。再次, 按照时空多样性的分类方法, 对基于多样性的主动防御系统的架构和实现技术进行介绍, 分析了时、空多样性系统的特点和实现方式。然后, 对多样性度量和有效性评估方法进行了分类总结, 分析了不同度量、评估方法的优势和不足。最后, 提出了多样性技术的下一步研究方向。

关键词 多样性; 主动防御; 分类; 度量; 评估
中图法分类号 TP393.1 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2022.05.08

Research Advances of Diversity Facing the Active Defense

TONG Qing[1], GUO Yunfei[1], HUO Shumin[1], WANG Yawen[1]
[1] Strategic Support Force Information Engineering University, Zhengzhou 450002, China

Abstract Different software or execution processes usually have different vulnerabilities. Based on that premise, diversity technology is applied to the design of system reliability and security, which significantly enhances the defense capability and intrusion tolerance capability of systems. However, it also has the shortcomings of high cost and high complexity. There are a lot of diversity technology implementation, system design and related assessment work in the existing research, covering a wide range. Focusing on the diversity application in the field of the active defense and the cost performance of applying diversity, this paper reviews the typical work and the latest progress in diversity research. Firstly, the diversity review research work is compared and analyzed, and the main contents and emphases of diversity research are discussed. Secondly, the concept of diversity is combed, and the definitions of temporal and spatial diversity are given. Thirdly, according to the classification method of temporal and spatial diversity, the architecture and implementation technology of diversity based active defense system are introduced, and the characteristics and implementations of temporal and spatial diversity systems are analyzed. Then, the diversity measurement and effectiveness evaluation methods are classified and summarized, and the advantages and disadvantages of different measurement and evaluation methods are analyzed. Finally, the future research direction of diversity technology is proposed.

Key words diversity; active defense; classification; metric; evaluation

1 引言

目前, 网络空间攻防博弈面临着“易攻难守”的不对称态势。这种不对称性主要体现三个方面: 1)工作量方面, 攻击者只需要针对信息系统的一个脆弱点发起有效攻击, 就能够达成一定的目的, 而防御者则需要对信息系统进行全方位、全时域的防护, 才能够避免安全性受损; 2)信息量方面, 攻击者能够通过正常访问或离线漏洞挖掘探测系统可能存在的防御缺陷, 采用扫描、探测、踩点等方式对目标系统进行信息收集和分析, 而防御者的信息系统或服务通常具有公开性, 广泛接受未知来访者的访问, 对攻击者的身份和目的缺少探测手段甚至一无所知; 3)后果方面, 攻击者攻击失败时, 对其自身没有明显的损害, 而防御者防御失败则标志着安全性或多或少的损失, 而类似于数据泄露、敏感信息泄露之类的损

通讯作者: 仝青, 博士生, Email:szbnlllskd@163.com。
本课题得到国家自然科学基金(No.62072467), 国家重点研发计划课题(No.2018YFB0804004), 国家自然科学基金创新研究群体项目(No.61521003)资助。
收稿日期: 2021-02-19; 修改日期: 2021-04-15; 定稿日期: 2022-03-22

失还具有不可恢复性。

传统防御以防火墙、入侵检测、防病毒系统等为主要技术, 这些方法依赖于对攻击行为或特征的掌握, 具有一定的被动性和滞后性, 难以应对未知威胁。

主动防御技术针对网络空间攻防的不对称性提出, 其防御理念旨在对攻击达成事先防御的效果。相较于被动防御, 主动防御不针对特定的攻击类型, 不针对特定的行为特征, 也不依赖于先验知识[1]。已有的主动防御技术包括: 1)入侵容忍, 通过提高系统的弹性使得系统在部分组件被攻击的情况下仍然能够保证服务的可用性; 2)移动目标防御, 通过动态改变系统的攻击面, 扰乱攻击者对系统信息的收集、将攻击者针对的脆弱点转移, 达到防御效果; 3)拟态防御, 通过增加异构性、冗余性和动态性改变系统单一静态的特性, 起到增大漏洞、后门不确定性的效果; 4)其他主动防御技术, 如蜜罐, 通过刻意构造脆弱环境诱捕攻击实现欺骗防御; 沙箱通过隔离不可信的操作保护执行环境的安全性; 可信计算通过建立信任根并向上逐级构建信任链, 进而将信任扩展到整个计算机系统达到增强系统安全性的目的[2]。

具有等价功能的网络空间元素在设计、实现等环节的差异性, 使得具备多样性的系统能够应对各种难以预期的问题, 如未知的错误或攻击等。多样性最早应用于信息系统的容灾、备份、容错和可靠性设计等方面。伴随着 1999 年美国 DARPA (Defense Advanced Research Projects Agency)“信息保障和可生存性”项目的启动和 2011 年美国国家科学技术委员会《可信网络空间: 联邦网络空间安全研究战略规划》的发布, 国内外陆续出现了入侵容忍、可信计算、移动目标防御、拟态防御等主动防御技术的研究, 在学术界和工业界兴起了“以改变游戏规则”为核心的网络空间安全技术研究热潮。

由于不同组件具有不完全相同的脆弱点, 多样性技术在该前提下改变了攻击者所依赖的特定攻击条件, 从而达到主动防御的效果。多样性被广泛应用于主动防御系统的设计和实现中, 在入侵容忍、移动目标防御和拟态防御等主动防御技术的发展中均扮演了重要角色。

多样性在主动防御的应用中不可避免地需要引入冗余组件或动态化运行, 这些特性相较于传统的单节点静态系统额外增大了系统的实现和运行代价, 对已有的应用和服务的稳定性也带来一定的挑战。多样性系统在应用多样性技术的同时, 需要针对传统应用权衡安全性、性能、代价和服务质量等方面。

围绕多样性在主动防御技术中的应用, 本文对多样性的概念、分类、度量和有效性等方面的研究成果进行了综述, 旨在分析多样性技术的性价比方面值得关注的问题。

已有研究对网络空间范围内的多样性研究进行了综述, 如表 1 所示。文献[3]针对可执行代码层面的多样性, 基于漏洞类型和攻击者的漏洞利用方法, 分析了可执行代码多样性对不同漏洞的防御有效性和性能影响。文献[4]从软件开发流程的不同阶段, 包括需求分析、系统设计和实现、质量保证和维护管理等方面对软件多样性系统的实现技术进行了综述。文献[5]在更广泛的多样性含义下对多样性的应用目标、实现方法和分析技术进行了讨论, 不仅仅关注了面向安全性、可靠性的多样化技术, 也包含了代码复用、多态性相关的广义多样性研究。文献[6]集中讨论了自动化软件多样性, 按照多样化的对象, 多样化发生的时间和层面对不同的多样性技术进行

表 1 典型的多样性综述对比

Table 1 Comparison of the typical reviews of diversity

	研究对象	概念讨论	分类	度量	有效性评估
文献[3]	可执行代码层多样性	/	面向软件脆弱性的多样性技术分类	/	基于攻击类型的分类分析
文献[4]	软件多样性	多样性	面向多样性系统建模方法的分类	/	/
文献[5]	多样性	多样性, 异构性, 冗余性, 动态性, 随机性	多角度分类	/	/
文献[6]	自动化软件多样性	/	基于软件开发流程的分类	/	基于评估方法的分类分析 (4 种)
文献[7]	混淆技术与多样化技术	/	基于应用目标和环境的技术分类	/	/
本文	面向主动防御的多样性	多样性, 异构性, 冗余性, 动态性, 随机性, 时间多样性, 空间多样性	基于应用维度的分类	√	基于评估方法的分类分析 (3 种)

了分类分析, 并讨论了多样化技术的安全性增益和性能代价问题。文献[7]讨论了混淆技术和多样化技术的应用目标和应用环境, 主要介绍了两类技术的实现。表 1 给出了典型的综述研究在概念讨论、系统分类、多样性度量和有效性评估 4 个方面的研究情况, 给出了本文工作与已有综述研究的异同点。其中“/”表示未在相应内容上集中讨论分析, 但不代表文献中未涉及该类工作。

本文在已有研究基础上, 对多样性的概念进行了梳理, 并着重对多样性度量和有效性评估方法进行了分类分析。文章剩余部分组织如下: 第 1 节介绍了多样性的概念和分类方法; 第 2 节按照时间多样性和空间多样性的分类方式对基于多样性的主动防御系统架构和实现技术进行描述; 第 3 节对现有的多样性的度量方法进行归纳和分析; 第 4 节总结了多样性有效性的评估方法; 最后总结全文并给出多样性技术的研究方向展望。

2 多样性相关概念介绍

2.1 多样性

多样性在不同领域有不同的内涵, 因而存在多种不同的多样性概念, 如: 生态系统多样性、物种多样性、遗传多样性、文化多样性、软件多样性等。网络空间中对多样性目前尚未出现明确的定义。文献[4]将软件多样性阐述为: 在当今的软件系统中, 通常同时开发不同的系统变体, 以满足广泛的应用环境或客户需求。这种广泛存在的变体被称为软件多样性。该定义仅针对软件多样性, 范围较小。文献[5]指出无论在哪个领域, 多样性均被认为对恢复力、稳定性或创新性的出现至关重要。多样性的概念在网络空间安全不同的研究群体中较分散, 因而该文献[5]也未对网络空间中的多样性进行明确定义。文献[8]提出网络空间多样性指一个计算机系统不同层面引入的多样性。虽然网络空间多样性的概念较模糊, 但不同文献中对网络空间多样性的理解仅在范围上有所不同, 其他方面并无矛盾。

网络空间中多样性的实现和应用非常广泛, 不仅包括常见的软件多样性, 也包括体系结构多样性、网络拓扑多样性、协议多样性、数据结构多样性等。

从网络空间多样性的应用方向和目的上, 多样性可分为可靠性、安全性、代码重用等方面。可靠性方向的多样性研究如容错计算[7]、数据库容错[9]等, 主要用于应对难以预期的错误、故障的发生, 保障系统的可用性、可靠性; 安全性方向如入侵容忍、移动目标防御等, 主要通过采取多样化组件冗余执行或动态改变攻击面等多样性策略容忍攻击或增大攻击难度, 达到提高系统防御能力的效果; 代码重用方向如面向对象程序设计、代码封装与模块继承[10], 基于软件多态性构建多样化的程序结构, 增强程序的可理解性、可维护性和可扩展性等。

在多样性的产生和来源上, 包括设计过程中、实现中和运行时造成的多样性。设计多样性[5]是在设计过程中采用不同的架构或者流程实现功能等价的应用或结构; 实现多样性是指在实现过程中采用不同的程序设计语言、代码实现方式或不同的组件、插件应用; 运行多样性是指功能载体在运行过程中等价功能的发挥采用的是不同的执行单元或运行环境参数。多样性的实现层次覆盖了网络、平台、组件、数据、运行环境, 呈现形式上可分为时间多样性和空间多样性[5]。

本文所提多样性主要指网络空间中包含多种多样具有等价功能的元素的属性, 这些元素既包括硬件、软件等网络实体, 也包括体系架构、网络拓扑、数据结构等逻辑结构。在网络空间多样性的广泛内涵下, 本文主要针对应用于主动防御系统的多样性进行进一步的研究和分析。

2.2 其他相关概念

在面向主动防御系统的多样性技术的实现应用中, 与之相关的概念包括异构性、冗余性、动态性、随机性等。

异构性[11]在信息技术中的含义通常指异构计算, 是指包含不同类型的计算机的网络, 这些计算机的内存大小、处理能力甚至基本的底层架构都可能存在很大差异。在更大的范围上, 异构性不仅指计算机的异构, 也包括数据结构、网络结构、体系结构等方面的不同。相对于多样性的概念而言, 异构性更强调不同对象的差异和不同, 以拟态防御为主的研究成果中对异构性的实现及其关键作用做出了详细阐述和约束[6,12]。另外, 本文中异构性主要指两个对象之间的差异性, 而多样性更多地反映 3 个或 3 个以上的对象集合所呈现出的特性。从该角度来看, 多样性的容错、容侵特性主要来自于异构性。异构性使得攻击者无法将针对某目标的成功攻击经验直接应用于异构的其他目标。因而异构性的大小决定了各类攻击在两个功能等价的目标之间的重现率。

冗余性[13]是为了提高系统的可靠性而对系统的关键部件或功能进行备份, 多个备份在同一系统的同一时间片内并行、共存。其中, “备份”不仅包含完全相同的拷贝版本, 也包括能够起到相同功能的相似版本。容错理论的冗余性通常伴随着异构性而

存在, 通过两者的结合以减少共模错误的发生。在冗余系统中, 一个关键的问题是如何在分布式的多个执行体之间达成“共识”。共识问题主要存在以下解决方案, 如状态机复制、集群管理、原子广播等方案, 主要针对分布式系统的一致性问题而提出。拜占庭容错协议是分布式系统共识协议的一种, 也是入侵容忍系统常用的共识协议。文献[14]对拜占庭容错系统的定义为: 拜占庭容错系统是由 n 个部件组成的系统, 即使 $f < n$ 个部件发生任意故障, 也能正常工作。入侵容忍系统除了要求冗余度, 还需配合一定的异构性、恢复机制以及其他机密操作才能实现[14]。

动态性[15]描述的是一个依赖于时间的概念, 通常指信息系统随时间而改变的特性。在移动目标防御系统中, 主要通过在不同层面、采用不同方式实现动态化, 对系统的参数、配置甚至组件随时间进行调整, 起到迷惑攻击者、加大攻击者信息收集的难度的作用。同样, 拟态防御中通过引入了动态化调度机制, 改变系统静态特征, 扭转攻击者具有的时间优势, 缩小攻击者探测系统和利用漏洞的时间窗。从攻击面理论[16]看, 动态化机制将攻击表面要素动态化, 或转移系统攻击表面, 使攻击者无法准确锁定目标的攻击表面。如果系统攻击表面变化足够快, 即使在低熵或存在暴力攻击的情况下, 动态防御也能够有效地保护系统。但动态化引入的开销及由此导致对正常服务的影响也是必须要解决的问题。

随机性[5]通常与动态性同时出现, 也是动态性的一种实现方式, 以实现动态化的不可预测性、增强系统的不确定性为主要目的, 避免攻击者对动态性结果的预测, 从而增大动态化的有效性。典型的随机化技术包括地址空间布局随机化、指令集随机化、内核数据随机化[17]以及网络层随机变化[18]和平台层随机化变迁技术[19]。随机变化的调度时间[20-21]和调度对象[22-24]均能够达到增加动态性的不可预测性的效果。但动态性实现也包含其他方式, 例如基于事件触发的调度策略[25-26], 调度结果往往更具有针对性, 能够有效提高调度的效率。在不同的应用场景中, 随机性并不一定是最好的选择, 具有导向性的策略如以调度异构性[27]或调度对象安全性[28]等为主导的调度策略更有利于防御效果的提升, 同时避免调度的盲目性。

时空多样性的概念在已有研究中较少涉及, 最早在文献[29]中被提出, 该研究认为多样性在时间维度的反映可定义为时间多样性。基于该理念, 本文定义时间多样性为同一系统或结构随时间改变, 从而采用不同的执行过程或对外呈现不同特征的属性。时间多样性结合了动态性和异构性, 是异构性在时间维度上的反映。现有研究中时间多样性主要体现在以移动目标防御为主的研究领域中。相对地, 空间多样性是指对一个输入同时采取多个执行过程, 并采取一致性协议对不同执行过程的执行结果进行表决以获得最终输出的运行特性。空间多样性是异构性和冗余性的有机结合。多数采用冗余副本实现的入侵容忍系统与空间多样性的定义相吻合。如图 1 所示, 时间多样性主要反映多样性在不同时间分片上的呈现, 而空间多样性主要是指多样性在同一时间切片上的呈现形式。需要说明的是, 本文所定义的时间、空间多样性特指系统在运行过程中呈现的多样性, 不包含多样性在时间、空间维度上普遍存在的广义概念。

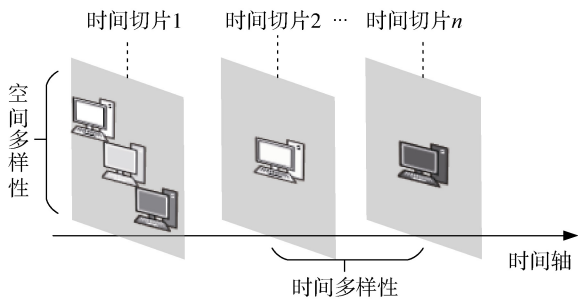

图 1　时、空多样性示意图

Figure 1　Schematic diagram of spatial and temporal diversity

上述概念在多样性系统设计和实现中时常交替出现, 在不同的研究中随侧重点的不同而使用不同的概念。如文献[30]设计了 SCIT (Self-Cleansing Intrusion Tolerance)系统, 该系统主要工作机制是动态轮换服务器执行体, 多使用“动态性”的概念; 文献[31]设计了多层次异构化 web 服务器, 更强调“异构性”的概念; 文献[5]针对软件多样性进行了综述研究, 将动态性和冗余性、异构性均归入多样性的范畴。

在以多样性为主要概念的文献中, 通常将涉及到网络空间安全的多样性划为“软件多样性”的研究范围中, 而从研究方向上, 软件多样性侧重于在软件层面实现设计或实现上的多样性, 来保证软件的可靠性。近年来, 随着主动防御技术的兴起, 以移动目标防御和入侵容忍等为代表的技术围绕制造不同层面的多样性达成主动防御效果, 包括在网络层、数据层、组件层、平台层、运行环境层等多个层次。软件多样性的研究层次主要处于组件层和运行环境层, 因而本文中对多样性的概念进行扩展, 后文中所提多样性包含了以软件多样性为代表的网络空间

安全领域所涉及的多个层次的多样性。

3 面向主动防御的多样性系统分类

主动防御技术主要目标是达到压制威胁、未雨绸缪的防御效果[1]。多样性技术由于其“内生安全”属性成为主动防御技术的主要选择。面向主动防御的多样性系统的实现方案较多, 根据多样性呈现方式的不同, 可以将基于多样性的主动防御系统分为空间多样性系统, 时间多样性系统以及混合多样性系统。多样性技术的实现在其他综述[3,5-6]中已有较清晰的介绍, 本节以多样性系统为主体, 介绍了包括多样性技术在内的系统实现和应用相关的多方面问题。

3.1 时间多样性系统

时间多样性系统主要以移动目标防御(Moving Target Defense, MTD)为代表。MTD 技术通过在系统运行过程中动态改变网络属性、平台、组件、运行环境或数据等方式, 改变攻击面, 增大系统的不确定性, 从而提高攻击门槛[16]。

文献[30]首次提出 SCIT 架构, 基于虚拟化技术, 每个轮换周期内同时提供多个服务器虚拟机对外提供服务, 轮换周期为分钟级, 将在线的服务器逐个下线进行清洗, 恢复到未受攻击的“纯净”状态。通过自清洗策略, 缩短一个服务器节点的暴露时间, 从而增强服务器系统的可靠性。测试结果显示, 较低的轮换周期下服务的响应时延略高于较高的轮换周期, 而轮换周期在 4 min 时, 响应时延与无动态性的系统相当, 总体上轮换周期对响应时延的影响微小。因而 SCIT 架构的设计在较小的性能损失下, 显著提高了系统的动态性。同时 SCIT 的特点还包括不依赖于入侵检测, 采用的是固定周期的前摄式轮换; 不依赖于入侵防御, 仅通过清洗恢复将可能发生的入侵进行清除。在防御效果上, 该文献中 SCIT 主要应用于入侵容忍提高系统可用性, 未对异构性提出需求。

后续研究中, 该研究团队从服务质量角度对 SCIT 进行了改进和评估, 并将 SCIT 架构应用于云环境和入侵检测。文献[32]通过对入侵容忍服务质量(IT-QoS, Intrusion Tolerance Quality of Service)进行建模分析提出了一种能够保障不同级别可靠性的 SCIT 系统动态频率调节机制, 使得系统能够根据运行环境的安全性变化重新调节动态频率和在线节点数量以维持一定等级的 IT-QoS。文献[33]通过调节 SCIT 系统的动态轮换时间来补偿系统运行过程中由于攻击面的扩张而导致的可用性下降。文献[34]提出了云环境下的 SCIT 部署方案(C-SCIT, Cloud-based SCIT), 该方案能够提供满足特定 IT-QoS 的 SCIT 实现。同时, 该文献指出云环境内在的异构性, 如来自不同供应商的云平台, 使得 C-SCIT 相对于 SCIT 架构能够提供额外的容忍能力。除此之外, 该团队将 SCIT 与入侵检测结合, 结果表明 SCIT 有助于降低入入侵检测系统漏检的代价[35]。

SCIT 架构虽然最初设计目的是为了达成入侵容忍, 但其采用的动态性方式却属于时间多样性。SCIT 的动态性主要基于清洗、恢复机制, 虽然提出异构性具有额外的增益, 但相关研究中未对异构性的作用进行深入挖掘。

其他时间多样性系统的设计主要以 MTD 系统为主, 解决攻击面的转移问题。文献[36]将攻击面的转移归纳为以下三个方面的问题: 1)移动什么, 2)如何移动, 以及 3)何时移动。

“移动什么”主要是指动态性在哪个层面上发生以及可改变的要素。已有的 MTD 系统广泛地应用于数据、软件、平台和网络等多个层面[16], 例如, 数据层可改变数据结构、存储方式等, 软件层可改变地址空间、指令集、代码结构等, 平台层可改变虚拟机、容器、执行体等, 网络层可改变端口号、IP 地址、网络拓扑等。

“如何移动”主要指移动前后的攻击面如何变化以达到提高攻击门槛的目的。文献[36]将该问题分为变换、多样和冗余 3 种实现形式进行分类讨论, 而本文从导向性和随机性两个策略设计方向对该问题进行讨论。随机性策略以增大动态性的不可预测性为主要目的。如 OF-RHM (OpenFlow Random Host Mutation)[18]随机改变主机的虚拟 IP 地址, 使攻击者无法基于前一 IP 地址猜测变换后的地址信息。文献[19]将用户的服务请求随机转发至后台的异构 web 服务器, 达到迷惑攻击者的目的。文献[37]将虚拟机进行随机迁移, 降低共存攻击的发生率。随机化效果较明显的应用更多集中于软件层, 例如地址空间随机化技术 ASLR (Address Space Layout Randomization)[17]将攻击者通过猜解获取地址的可能性降低为接近于 0。导向性移动策略基于一定的评估结果, 采取针对性的变化。如文献[27]将对执行体的异构性进行了量化, 按照异构性大小决定执行体调度上线的概率, 以异构性最大化为导向。文献[26]基于系统对当前受攻击状态和程度等实时状态信息的评估, 通过机器学习算法产生安全性更高的配置。文献[38]通过自学习恶意侦查策略进行变异策略的选择以最大化变异结果的不可预测性。

“何时移动”主要指变化发生的时机。包括了前摄式, 应答式和混合策略三种[36]。前摄式方法以时间驱动为主, 例如按照一定的变化频率变化。文献[18]基于主机身份和运行时间生成下一次变化时间, 进一步增强了前摄式变化时机的不可预测性。应答式方法以事件驱动为主, 驱动事件包括了入侵发生、系统受损、预警等[26,39]。其中文献[26]采用攻防博弈分析和机器学习对攻击进行预测, 针对预测结果采取一定的变换。虽然攻击事件可能为发生或不发生, 但预测结果表明存在一定的风险, 因而属于一种基于预警信息驱动的应答式移动。混合策略[37,40]融合了前摄式与应答式变化, 一方面能够增大动态性的不可预测性, 另一方面有助于提高调度的效率, 避免频繁而盲目的变化。

其他时间多样性系统在文献[16,36,41]等综述中有详尽的讨论, 为避免重复, 本节仅对典型系统进行了讨论。

3.2 空间多样性系统

空间多样性系统以具有冗余副本的入侵容忍系统为代表。该类系统通常利用冗余备份的方式, 同一请求同时被多个节点处理, 对所有节点的处理结果进行比较和表决。本节中空间多样性系统的冗余性主要指并行处理相同输入的冗余, 不包含同时运行但处理不同输入(如负载均衡的工作方式)或轮换运行(如基于清洗策略的入侵容忍系统)的冗余形式, 以区分于时间多样性研究中涉及的冗余概念[36]。

空间多样性系统中冗余节点达成共识的方法包括: 高可靠的一致性协议和简易一致性协议。

高可靠的一致性协议以拜占庭容错协议为主。拜占庭容错协议所需冗余节点数量通常较多, 主要目的是保证表决结果的可靠性, 即保证 $3f+1$ 个节点中有至多 f 个失效节点的情况下仍能得到正确结果。经典的拜占庭协议中需要保证 $n \geqslant 3f+1$, 为了减少冗余代价, 文献[42-44]通过提高资源利用率将拜占庭一致性协议的副本数量从 $3f+1$ 降低至 $2f+1$, 甚至 $f+1$。但上述研究通常针对于软件或通信中的不确定错误。对于面向网络攻击的拜占庭容侵系统, 例如 Prime[45], 通常采用典型的拜占庭容错协议, 为主节点部署了 $3f$ 个备份节点, 每个备份节点上都连接了基础服务库, 用于与其他分节点之间传递事件, 保证即使在有 f 个备份节点受到攻击而失效的情况下系统依然运行正确。文献[14]对拜占庭容错和容侵进行了对比研究, 讨论了拜占庭容侵应用中已解决的、未解决的和开放性问题。

拜占庭容错和容侵能够有效提高可靠性, 然而对机密性的保障需要配合其他机制, 如门限加密等。OASIS(Organically Assured & Survivable Information Systems)[46]设计了全面的防护机制, 防御层针对良性网络/计算机故障、被动攻击、低强度主动攻击、拒绝服务和内部攻击等基本系统威胁均能够提供保护。在保护数据机密性上, 采用了门限加密的方式, 基于冗余数据片段恢复出原信息。

拜占庭容错协议主要保障协商结果的可靠性, 也有针对协商的可终止性问题提出的其他一致性协议, 如随机一致性协议[47]。RITAS (Randomized Intrusion-Tolerant Asynchronous Services)[48]描述了一种随机入侵容忍协议栈, 实现了多种随机一致性协议, 通过在局域网和广域网环境等实际条件下的性能评估, 表明不同于直观上对随机一致性协议迭代次数较高的认识, 随机入侵容忍协议在实际条件下能够高效实现分布式系统的容侵和容错。

在对一致性协商结果的可靠性要求不高的场景下, 不少入侵容忍系统采用简易的一致性协议, 如大数表决。文献[49]对比研究了各种不同的表决方法, 包括大数表决、全体一致表决、复数表决、中值表决等。SITAR(scalable intrusion-tolerant architecture)[30]采用多层次冗余的架构, 在代理层、表决层、校验层和服务器层均进行了异构冗余的节点, 并支持大数表决和拜占庭表决多种表决方式。多版本执行环境-Varan[50]提出了一种多版本执行的实现方法, 通过创建多个子线程, 并行执行不同版本的二进制代码, 对每个子进程的执行结果进行大数表决, 提高软件的可靠性。多变体系统[51]通过创建两个不同的 http 服务软件版本, 对不同版本的响应进行比较来判断是否有攻击发生。

高可靠的一致性协议更多应用于分布式系统或环境下, 虽然在实现中也能够容忍恶意攻击的威胁, 但存在较高的冗余代价, 更注重协商过程的设计。大数表决也是其中的协商环节之一。在面向应用、服务等较高软件栈层次的入侵容忍设计中, 通常采用以大数表决为主的简易一致性协议, 不强调表决结果的高可靠性、表决的可终止性等问题, 以异构性面向攻击存在的差异性表现为前提, 重点探讨异构冗余的组件在高层应用中对入侵的发现、容忍能力。大数表决对冗余度的要求较低, 文献[12]讨论分析了不同余度下的基于差模的大数表决策略的可靠性, 结果表明冗余度为 3 时大数表决的结果具有较高的可靠性。

3.3 混合多样性系统

时间、空间多样性均存在一定的代价问题, 两种

多样性的混合设计研究虽然较少, 但也存在一些代表性研究。如, 文献[52]提出了一种结合了消息队列服务的入侵容忍防火墙模型——SieveQ。该防火墙采用两层过滤机制, 第一层过滤机制采用时间多样性机制, 通过动态替换失效节点实现容侵; 第二层过滤机制采用空间多样性机制, 基于冗余的备份节点在拜占庭容错协议下工作。由于无效消息在第一层过滤中被有效清除, 因而相比已有的入侵容忍防火墙, SieveQ 降低了一定的代价。该防火墙模型提供了一种空间多样性和时间多样性的结合方案, 两种多样性分别应用于两层不同粒度的过滤机制中, 通过第一层时间多样性的过滤减轻了第二层空间多样性果过滤层的业务量。

拟态防御提出了 DHR (Dynamic Heterogeneous Redundancy)架构[53]。该结构基于异构冗余节点并行处理后通过裁决输出最终结果, 同时, 在系统运行过程中, 对在线的冗余节点进行动态轮换。文献[31]基于 DHR 架构实现了拟态防御 web 服务器, 基于 COTS (Commercial off the Shelf)在软件栈的各个层次实现了异构化。该架构同时也在 DNS 服务器等的拟态化设计中得到了应用[54]。DHR 架构直接叠加了两种多样性, 虽然防御效果更好, 适用于强安全性需求的应用, 但不可避免地叠加了两种多样性的缺点。

鉴于时间、空间多样性两种系统对防御效果、防御代价等方面的影响不同, 通过一定的策略设计联合两种多样性策略, 有可能达到优劣互补的效果。但目前仍缺少更经济的时空多样性联合方案。

3.4 小结

无论是联合了动态性和异构性的时间多样性系统, 还是联合了冗余性和异构性的空间多样性系统, 相对于传统的单节点、静态系统, 均增加了一定的代价。表 2 给出了不同多样性系统的特点和典型实现。

表 2 不同类型多样性系统的特点与典型实现

Table 2 The characteristics and typical implements of different diversity systems

分类	动态性	异构性	冗余性	运行策略特点	典型系统或实现
时间多样性系统	√	√		动态调度	[17-19, 26-27, 30, 36-40]
空间多样性系统		√	√	基于一致性协议的表决	[14,30, 45-46, 48-51]
混合多样性系统	√	√	√	一致性表决为主, 动态调度为辅	[31,54]
				一致性表决与动态调度分层工作	[52]

动态频率的大小对时间多样性系统的防御效果起主要决定作用。直观而言, 动态频率越高, 系统留给攻击者用于探测脆弱点、发起攻击的时间窗越小, 系统的防御能力越强。但较高的动态频率对系统的稳定性或服务质量会造成一定的影响, 为了消除这些影响, 不可避免地需要引入其他机制(如状态的保存与恢复), 也会增加额外的代价和不确定因素。

冗余度是空间多样性系统的代价来源之一, 冗余度的大小取决于应用场景。拜占庭容侵系统的设计通常对冗余度的要求较高, 但保证了共识结果的可靠性。相对宽松的大数表决策略在共模漏洞率较低的前提下也能够保证较高的表决正确率, 同时降低了系统的复杂度。冗余度所带来的代价主要包括时延和吞吐量, 通常会受限于所有冗余节点中性能最差的节点, 造成短板效应。而采用性能较高的冗余节点则容易导致异构性的下降, 无法很好地保证“低共模漏洞率”的前提条件。

异构性对两种多样性系统均有一定影响。通过增强异构性, 可以增大时间多样性系统变化的不可预测性, 同时使攻击者针对该系统已掌握的攻击经验失效。对空间多样性系统而言, 冗余节点之间的异构性越大, “低共模漏洞率”的前提越能得到保证, 防御能力就越强。但异构性受限于系统所采用的多样性来源, 部分基于 COTS 的异构组件可能出现如代码复用等造成的共模漏洞, 难以保证其异构性。同时, 随着攻击者探测时间的延长, 对系统的掌握会逐渐增大, 在有限的变换空间内, 系统的异构性有可能被攻击者逐渐学习掌握, 文献[33]也提出了随着不断的变换, 存在攻击面扩张问题, 有限的异构性随着运行存在熵降低的问题。

4 多样性度量方法

在高代价的情况下, 多样性系统的应用部署存在一定的阻力。面对该问题, 近年来, 有研究者从安全性评估方面开展了对多样性的度量与评估。当前已掌握的文献中, 对多样性的度量多出于评估安全

性的需求, 而忽略了多样性度量对于优化多样性系统设计的必要性。目前已掌握的度量工作大致包括信息熵的度量, 基于共模漏洞的度量和基于近似重复检测的度量三种度量方法。

4.1 基于信息熵的度量

由于概念上的相似性, 生物多样性的度量方法常被用于信息系统中多样性的度量。生物多样性包括多样性(不同物种的数量)、平衡性(个体在这些物种之间的分布)和差异性(物种之间的差异)。由于物种间的差异过于细粒度且难以获得, 因而在大型复杂系统上, 通常参考物种多样性和物种平衡性的度量, 以基于香农熵的香农指数为基本度量方法。

Forrest[55]是第一位通过生物多样性概念度量软件多样性的研究者。文献[56]基于香农指数设计了度量方法, 用于软件开发者多样性的度量。文献[57]提出了一种网络多样性形式化描述模型, 设计和评估多样性度量指标用于评估安全性。该研究设计了基于生物多样性启发的资源多样性度量方法。该方法考虑了网络中资源的不均匀分布和资源之间的相似性等因素。文献[58]提出了一种量化安全评估方法 QSAME (Security Assessment Method based on Entropy)。基于香农熵提出了漏洞熵、攻击熵和衰减熵, 并提出相应的计算方法。分析结果显示, 移动目标防御系统的安全性与漏洞熵呈正相关, 与攻击熵和衰减熵呈负相关关系。但该研究以两个执行体的完全异构为前提假设。文献[59]以香农指数度量架构多样性, 并提供了一种基于约束求解的方法来静态估计不同架构的最小适应成本。该研究的多样性度量主要是对系统内部架构呈现的多样性的度量, 不仅限于同类型组件的异构性, 更接近于生态多样性的概念。

文献[60]基于操作系统所占市场份额和香农指数度量了操作系统的多样性, 显示目前商业软件操作系统具有较强的多样性, 且这种多样性随着时间的推移趋于增大。然而该研究以不同版本操作系统不存在共模漏洞和同源性为假设, 多样性度量粒度较粗。

除了基于生物多样性启发的度量方法, 文献[61]采用瑞丽熵来代表多样性, 通过给出不同操作系统的时长占有率和漏洞数量, 从两种数据角度计算了多样性并进行了参数的分析, 但缺少对度量结果的验证。

4.2 基于共模漏洞的度量

基于共模漏洞的度量是一种基于相同漏洞的数量来衡量不同软件体或系统的异构性的方法。

在早期的可靠性研究中, 采用共模错误对软件多样性进行度量。文献[62-63]通过基于共模错误度量软件多样性或软件设计多样性。通过统计一组多版本程序或软件的相同错误或相同失效点衡量多样性。该度量方法主要针对早期软件设计与实现中可能存在的错误, 多应用于容错系统。多样性的度量以准确反映可靠性为目的。文献[64]提出失效率作为鲁棒性指数, 通过枚举所有输入下不同版本软件的输出, 度量了每个版本的失效率, 并将各个版本两两组合进行了失效率测试, 结果显示多版本系统的失效率比构成该系统的任意版本的失效率都要低。失效率也从一定程度上反映了多样性, 然而需要通过枚举所有输入进行测试才能得出结果。

与共模错误类似, 在网络攻击日益增多的环境下, 多样性度量以共模漏洞为基础。文献[65-66]提出了 CVI (Common Vulnerability Indicator)表示两个操作系统之间的多样性。度量时考虑了相同漏洞的数量以及漏洞曝出的时间, 赋予不同年限内曝出的漏洞以不同的权重, 据此计算某一年的两个操作系统质量的 CVI 值。该研究对多样性的度量主要针对两个操作系统之间的异构性, 研究结果说明了基于操作系统的多样化能够提高系统的可靠性。

类似的, 文献[9]研究了 4 种商业数据库软件的共模漏洞数量, 文献[64]采用共模漏洞数量作为操作系统的异构性指标。文献[67]不仅研究了共模漏洞的数量, 同时分析了相同漏洞利用代码在不同软件版本、相同软件不同平台环境下的可利用性。

4.3 基于近似重复检测的度量

基于近似重复检测的度量主要来自于对文本、代码等的相似度检测方法。

文献[8]基于哈希值比较了二进制文件是否相同, 并进一步通过度量操作系统中每种二进制文件的变体比例分布和变体种类比例分布等, 分析了当前操作系统的多样性的存在性。对二进制异构性的比较仅表明两个文件是否相同, 但未给出相似性的大小。文献[68]利用数据挖掘和哈希算法, 对同一恶意代码的不同二进制文件进行了相似性度量, 以检测恶意代码的变种。文献[69]比较了不同的近似重复检测算法在度量软件多样性上的有效性。

文献[70]度量了控制流图差异性和多样化编译算法差异性, 得到代码异构性。文献[71]通过不同次执行时子路径相异指令的数量度量子路径差异性, 采用相邻子路径的差异性均值作为一个时间区间内的时间多样性。该方法在所掌握文献中首次对时间多样性的进行了度量。

文献[72]文献采用了非结构化文本分析中的常见方法, 如欧几里得距离、余弦距离等, 对比分析了不同的软件多样化策略的差异性, 其中包含了对多样性策略所产生的变体的多样性的度量。该研究同时提出, 在未来研究中有必要将多样性策略的差异性与攻击难度形成映射关系。文献[73]基于海明距离度量了代码的句法冗余度, 并基于句法冗余分析了代码的唯一性。

4.4 小结

3 种多样性度量方法适用于不同的应用场景, 表 3 给出了各种多样性度量方法的优缺点对比。

基于信息熵的度量易于把握多个(3 个或 3 个以上)执行体的群体多样性特征, 然而, 对于具有局部共同点的系统而言, 度量粒度较粗。

基于共模错误和共模漏洞的度量方法可以从较细的粒度上对两个执行体的异构性进行较精确的度量。基于共模漏洞的度量方法是直接面向攻击的度量, 更容易集中反映执行体面向攻击的多样性, 度量结果在较大程度上代表了多样性所能发挥的安全性, 在主动防御系统的设计中, 更具有实际指导意义。但已有研究中该度量方法通常用于验证现存的异构性足以提供容错、容侵能力, 同时, 需要统计分析已知的漏洞信息, 工作量大, 且难以覆盖未知漏洞。

基于近似重复检测的度量方法从代码或执行过程本身出发, 在较细粒度上度量了异构性, 可借鉴的方法和工具较多, 具有相对完善的理论基础, 但该方法侧重于两个执行体之间的异构性度量。已有研究的方法同时需要对大量的代码或二进制文件进行分析, 不适用于大型系统的多样性度量。在对多样性系统的安全性设计上, 有时代码本身的异构性或多样性并不足以说明系统安全性的大小, 例如与攻击目标不相关的代码异构性对多样化设计并不构成安全增益。

表 3 多样性度量方法优缺点对比

Table 3 Comparison of advantages and disadvantages of diversity measurement methods

度量方法分类	特点	优势	不足	相关工作
基于信息熵	借鉴生物多样性的度量理念或直接基于信息熵的相关概念评价多样性	易于把握 3 个或 3 个以上执行体的群体多样性特征	粗粒度度量, 难以避免局部相似性对度量结果的影响	[55-61]
基于共模漏洞	基于已知漏洞数据进行统计分析相同攻击在不同执行体上的可利用性	细粒度度量; 面向攻击成功概率的度量, 有利于指导多样性系统设计	统计漏洞信息工作量大; 难以覆盖未知漏洞信息	[9,62-67]
基于近似重复检测	基于执行体自身组成或执行过程的差异性刻画执行体的相似性	细粒度度量; 理论和方法相对完善	侧重于 2 个执行体之间的异构性度量; 不适用于大型系统; 度量结果与安全性的相关性有待验证分析	[8,68-73]

5 多样性的有效性评估

多样性的有效性评估主要分析多样性在达成防御目的方面的效果, 相关研究形成了基于理论模型、仿真实验和实证实验的 3 种主要的评估方法。

5.1 基于理论模型的评估

理论分析方面, 研究者主要通过构建攻防模型或抽象系统模型, 从理论上分析系统的安全性或可靠性。

文献[74-75]创建了一种基于马尔科夫链的分析模型, 讨论了攻击间隔、调整间隔、节点数以及调整间隔内调整的节点数量 4 个关键参数对攻击成功率的影响, 评价了网络层 MTD 应用的有效性。文献[76]针对文献[74-75]存在的状态爆炸的问题, 提出基于攻击图的马尔科夫模型构建方法作为改进。然而该模型因假设过多, 与实际情况可能有所偏离。

文献[33]使用半马尔可夫过程建模进行定量分析, 指导调整 SCIT 系统参数以补偿服务攻击面的扩展。文献[77]基于马尔科夫到达过程定量评估了基于虚拟机的入侵容忍系统, 通过 Laplace-Stieltjes 变换给出了系统的失效时间。

文献[78]利用攻击图模型推导出攻击者达到目标所需时间的精确公式。通过分析证明了更大的多样性和更快的变迁频率会使预期的攻击成功时间更长, 因而可以说明时间多样性系统的防御有效性。但该研究依赖于具体的攻击行为并需要大量概率相关的细节数据。

文献[79]将 MTD 技术分为基于网络、主机和工

具等 3 类, 利用网络传播动力学和度量评估方法研究了 MTD 的最佳策略, 但该模型中存在参数难以获取的问题。

文献[80]提出了 PLADD (Probabilistic Learning Attacker, Dynamic Defender)博弈模型。PLADD 模型将攻击者和防御者的博弈抽象为对关键资源的抢占, 攻击者通过发起攻击抢占资源, 防御者通过驱逐攻击和动态变化两种抽象行为获取资源, 通过计算双方在博弈期间内对资源的占有时间得到双方的收益, 通过给定双方每种行为的代价给出双方的支付, 以此建立了基本的博弈模型。在仿真分析中, 研究者研究了无学习能力和有学习能力两种类型的攻击模式, 通过参数假设得到了一部分能够验证时间多样性系统有效性的分析结果。文献[81]提出了一种分层博弈模型, 分析了防御者与攻击者、攻击者与系统“内鬼”之间的双层博弈, 并通过对静态和动态两种博弈进程下纳什均衡的求解, 分析了防御者的最优策略。

文献[82]从可靠性指标和基于服务请求优先级的服务部署成本两个方面分析了冗余资源利用的有效性。通过综合分析模型描述了不同冗余方案下可靠性、成本和安全性之间的权衡。

5.2 基于仿真实验的评估

基于仿真实验的评估主要通过搭建仿真实验环境, 测试仿真条件下多样性系统呈现的安全性、防御能力、成本、性能表现等指标。

文献[83]采用一个基于网络安全模拟器 NeSSi2[84]用于评估网络层 MTD 系统中执行体的设计有效性。该平台由“防御”、“攻击”及“基本事实”三大组件构成, 可以提供数据包级别的仿真环境, 主要用于仿真测试安全相关的算法、插件和网络架构。文献[83]以虚拟机刷新操作作为 MTD 技术实现方式, 基于攻击图模拟攻击路径。通过评价不同的动态频率对攻击成功率的作用, 验证了虚拟机动态变化的 MTD 系统的防御有效性。该仿真实验的环境假设较简易, 需要在构建更复杂攻击仿真环境上进行进一步的探索。

文献[85]介绍了一种可用于 MTD 安全评估的网络仿真虚拟基础设施 VINE (Virtual Infrastructure for Network Emulation)。该设施为网络安全研究人员提供了一种可以快速实施大规模重复实验的工具。该研究通过仿真实验, 对比了没有 MTD 和有 MTD 两种情形下服务的可用性, 得出“移动目标防御能够有效提高安全性”的结论。

文献[38]基于 Mininet 构建了仿真网络拓扑, 通过主动扫描和被动窃听实验分析验证了所提动态变异方法在抗侦查方面的有效性, 同时对性能和代价也进行了评估分析。

文献[86]面向互联网服务供应商提出了一种路由变异方法, 使得攻击者难以对网络进行攻击侦察或获取网络拓扑信息, 从而防御可能的 DDoS (Distributed Denial of Service)攻击。该研究在 Mininet 仿真环境中实现了路由变异方法, 并通过攻击成功率和防御开销两方面评估了路由变异方法的有效性。其中开销衡量了防御成本, 例如存储成本和端到端延迟, 但仿真结果与实际网络环境下的情形存在一定差异。

5.3 基于实证实验的评估

实验分析方面, 研究者通常针对已实现的真实多样性系统或软件进行信息数据分析、攻防测试、性能测试等, 依据分析或测试结果对多样性系统的有效性进行评估。

文献[9]研究了 4 种商业数据库软件的漏洞报告, 包含了 181 种漏洞, 其中仅有 4 个漏洞同时出现在 2 个版本的数据库软件中, 而没有一个漏洞同时出现在超过 2 个版本的软件中, 该研究证明了多样性能够有效地实现容错。文献[87]统计分析了在 2007 年发布的 6427 个漏洞信息, 发现 98.5%以上的商业软件存在“替代品”, 而且绝大多数没有共同的漏洞, 或者不能被相同的攻击代码攻击。通过该研究验证了多样化的商业软件有效抵御共模攻击。类似地, 文献[88]通过采集国家漏洞数据库公共漏洞的数据, 检查了不同操作系统中的漏洞是如何暴露和消除的, 并通过评估它们的严重性, 分析了这些操作系统在不同的入侵容忍架构中部署对可用性、机密性和完整性的影响。

文献[64]提出失效率作为系统鲁棒性指数。首先测试了 POSIX 的 13 种实现版本, 发现失效率在 6%~19%之间; 又通过两两组合不同的版本构成多版本系统, 测试系统的失效率; 结果显示多版本系统的失效率比构成该系统的任意版本的失效率都要低, 因而验证了多版本程序能够提高系统安全性。文献[89]利用 UVa Online Judge 网站收集了数千个实现相同功能的不同来源的程序, 通过测试程序集中两两程序之间共模错误的发生几率, 发现不同版本的程序出现相同错误的概率较低, 得出多版本设计对提升安全性的有效性。文献[90]实现了一种基于三模冗余的多版本系统。每个版本的程序是在同样功能需求下采用不同的开发流程和开发语言实现的, 通过表决 3 个程序的输出得出最终输出结果, 该系统在黑盒实验中的可靠性超过了 3 个版本中可靠性最

高的程序, 证明了多样性在提高可靠性方面的有效性。

文献[91]提出了一种基于主机的入侵检测系统的多样性配置方法, 并通过实验证明了多样性配置下, 能够有效减少入侵漏报率。

文献[52]通过性能测试以及不同的攻击场景实验, 评估验证了基于拜占庭容侵的消息服务 SeiveQ 有效提高了现有的防火墙的恢复能力和容侵能力, 并通过负载测试验证了 SeiveQ 的高负载能力。

文献[92]提出了一种增强型移动目标防御系统——Morpheus, 并通过实验分析了 Morpheus 系统在不同类型的控制流攻击下的安全性和基于标准测试集的性能损失。

基于拟态防御 DHR 架构的拟态防御系统均进行了安全性测试、性能测试等实验验证, 显示了多样性技术的可行性和有效性, 如文献[31,54,93], 其中文献[93]在系统实现和有效性验证的基础上, 通过实验进一步分析了拟态防御 web 服务器在使用不同的表决策略和异构性不同的执行体的情形下系统的安全性变化。

5.4 小结

表4给出了各类多样性有效性评估方法的优缺点与主要方法汇总。通过对比以上多样性有效性评估方法, 可以发现:

1) 基于理论模型的评估能够对一种或一类技术进行抽象评估, 对多样性分析的覆盖面较广, 但是涉及到参数选定的难题, 例如对攻击成功率的假设、状态转移概率的假设等。

2) 基于仿真平台的评估能够在短时间内模拟大量的不同配置下多样性系统的运行, 比较不同配置对多样性系统的影响。然而仿真平台的配置通常较理想化, 因而在性能、代价等方面的数据可能与实际环境存在误差。

表 4 多样性有效性评估方法优缺点对比

Table 4 Comparison of advantages and disadvantages of evaluation methods for diversity effectiveness

有效性评估方法分类	主要方法	优势	不足
理论分析	马尔科夫模型[33,74-77] 博弈模型[80-81] 网络传播动力学模型[79] 其他概率分析模型[78,82]	问题覆盖面广 可操作性强, 不需要搭建环境或设备配合实施	假设条件较多, 参数难以选定
仿真实验	NeSSi2[83] VINE[85] Mininet[38,86]	贴近真实测试同时评估效率高	部分仿真数据与实际环境可能存在误差
实证实验	共模漏洞率分析[9,87-88] 多版本失效率验证[64,89-90] 攻击实验检验[31,52-54,91-93]	评价方法直观; 数据和结果真实, 具有说服力; 真实反映实际防御机理	分析问题范围有限

3) 基于实证实验的评估验证有效性相较于理论分析和仿真分析是最直接、最具有说服力的方法。所获取的数据和结果均具有真实性, 其中, 基于攻击的实验检验方法能够直接验证某种技术的有效性, 同时反映出攻击发生时, 多样性技术的防御机理和细节。然而该种方法通常针对特定的系统配置、攻击案例或攻防交互环境, 能够反映和分析的问题范围有限。

6 结束语

多样性作为主动防御的关键特点之一, 对扭转网络空间“易攻难守”态势发挥了重要价值。随着网络空间攻击行为的智能化、复杂化演进, 多样性技术的应用与评估问题需要得到进一步的研究与探索。针对上述问题, 本文首先梳理了多样性的概念, 并按照时空多样性的分类方法对多样性系统进行了介绍, 随后, 分类分析了不同的多样性的度量以及评估方法, 并对各种方法的优缺点进行了比较。

随着多样性技术的普及与应用, 后续研究值得关注的问题主要包括:

1) 面向网络攻击的多样性度量。针对多样性度量的必要性以及该工作对多样性系统设计的指导意义, 可以开展面向网络攻击的多样性度量, 使得多样性的度量结果能够更确切地反映多样性能够产生的安全增益, 增强多样性度量的对多样性系统设计和优化的指导作用。

2) 多样性与安全性的量化关系研究。针对该问题, 可研究多样性能够发挥有效性的范围, 分别讨论“多样性能够发挥安全增益的最小值如何”, “多样性达到多大以后不再有安全增益的提升”等问题。另一方面, 可以研究多样性的提高与安全增益增长之间的量化关系, 在多样性度量的基础上, 通过多样性的提升直接反应安全性的变化幅度, 有助于设计更高效的多样性策略或技术。

3) 多层面、多维度的多样性技术联合设计与评估。在多样性技术的最新发展中, 已有部分研究开始对不同层面的多样性技术进行组合设计, 如移动目标防御系统中代码层、数据层等层面的组合, 但仍缺少不同维度的多样性联合设计, 如时间多样性与空间多样性的组合。不同的多样性技术(包括不同层面和不同维度)在安全性、代价等方面的影响不同, 面向相同的防御目标, 多种多样性技术存在联合优化的可能。进一步地, 可以开展对多层面、多维度的多样性策略优化与评估研究。

4) 多样性系统的代价收益比评估。已有研究对多样性的代价评估比较模糊, 造成主观上认为多样性技术性价比较低的现象。已有的有效性评估工作肯定了多样性在提供安全性上的意义, 但缺少对多样性系统安全性的代价收益比的评估。针对该问题, 可以研究多样性技术在实际部署中的代价收益比评估方法, 以促进多样性技术的应用和改进。

致 谢 本文工作受到国家自然科学基金(No.62072467), 国家重点研发计划课题(No.2018YFB0804004), 国家自然科学基金创新研究群体项目(No.61521003)资助。

参考文献

[1] Chen F C, Hu H C, Liu W Y. Cyberspace active defense [M]. Beijing: Science Press, 2018.
(陈福才, 扈红超, 刘文彦. 网络空间主动防御技术[M]. 北京: 科学出版社, 2018.)

[2] *Shen C X, Zhang H G, Wang H M, et al.* Research and development of Trusted Computing *[J]*. *Scientia Sinica* (*Informations*), 2010, 40(2): 139-166.
(沈昌祥, 张焕国, 王怀民, 等. 可信计算的研究与发展[J]. *中国科学: 信息科学*, 2010, 40(2): 139-166.)

[3] Just J E, Cornwell M. Review and Analysis of Synthetic Diversity for Breaking Monocultures[C]. *The 2004 ACM workshop on Rapid malcode*, 2004: 23-32.

[4] Schaefer I, Rabiser R, Clarke D, et al. Software Diversity: State of the Art and Perspectives[J]. *International Journal on Software Tools for Technology Transfer*, 2012, 14(5): 477-495.

[5] Baudry B, Monperrus M. The Multiple Facets of Software Diversity[J]. *ACM Computing Surveys*, 2015, 48(1): 1-26.

[6] Larsen P, Homescu A, Brunthaler S, et al. SoK: Automated Software Diversity[C]. *2014 IEEE Symposium on Security and Privacy*, 2014: 276-291.

[7] Hosseinzadeh S, Rauti S, Laurén S, et al. A Survey on Aims and Environments of Diversification and Obfuscation in Software Security[C]. *The 17th International Conference on Computer Systems and Technologies 2016*, 2016: 113-120.

[8] Kravvaritis K, Mitropoulos D, Spinellis D. Cyberdiversity: Measures and Initial Results[C]. *2010 14th Panhellenic Conference on Informatics*, 2010: 135-140.

[9] Gashi I, Popov P, Strigini L. Fault Diversity among Off-the-Shelf SQL Database Servers[C]. *International Conference on Dependable Systems and Networks*, *2004*, 2004: 389-398.

[10] Pohl K , Günter Böckle, Linden F J V D . Software product line engineering : foundations, principles, and techniques[C]. *The First Intl Workshop on Formal Methods in Software Product Line Engineering*, 2005, 49(12):29-32.

[11] Segev E, Ahituv N, Barzilai-Nahon K. Mapping Diversities and Tracing Trends of Cultural Homogeneity/Heterogeneity in Cyberspace[J]. *Journal of Computer-Mediated Communication*, 2007, 12(4): 1269-1297.

[12] Hu H C, Chen F C, Wang Z P. Performance Evaluations on DHR for Cyberspace Mimic Defense[J]. *Journal of Cyber Security*, 2016, 1(4): 40-51.
(扈红超, 陈福才, 王禛鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. *信息安全学报*, 2016, 1(4): 40-51.)

[13] Chern M S. On the Computational Complexity of Reliability Redundancy Allocation in a Series System[J]. *Operations Research Letters*, 1992, 11(5): 309-315.

[14] Bessani A N. From Byzantine Fault Tolerance to Intrusion Tolerance (a Position Paper)[C]. *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops*, 2011: 15-18.

[15] Wu Z Z, Chen X Y, Yang Z, et al. Reducing Security Risks of Suspicious Data and Codes through a Novel Dynamic Defense Model[J]. *IEEE Transactions on Information Forensics and Security*, 2019, 14(9): 2427-2440.

[16] Zhou Y Y, Cheng G, Guo C S, et al. Survey on Attack Surface Dynamic Transfer Technology Based on Moving Target Defense[J]. *Journal of Software*, 2018, 29(9): 2799-2820.
(周余阳, 程光, 郭春生, 等. 移动目标防御的攻击面动态转移技术研究综述[J]. *软件学报*, 2018, 29(9): 2799-2820.)

[17] Shacham H, Page M, Pfaff B, et al. On the Effectiveness of Address-Space Randomization[C]. *The 11th ACM conference on Computer and communications security*, 2004: 298-307.

[18] Jafarian J H, Al-Shaer E, Duan Q. Openflow Random Host Mutation: Transparent Moving Target Defense Using Software Defined Networking[C]. *The first workshop on Hot topics in software defined networks*, 2012: 127-132.

[19] Thompson M, Mendolla M, Muggler M, et al. Dynamic Application Rotation Environment for Moving Target Defense[C]. *2016 Resilience Week*, 2016: 17-26.

[20] Clark A, Sun K, Poovendran R. Effectiveness of IP Address Randomization in Decoy-Based Moving Target Defense[C]. *52nd IEEE Conference on Decision and Control*, 2013: 678-685.

[21] Keromytis A, Prevelakis V. A survey of randomization techniques against common mode attacks. Technical Report DU-CS-05-04. Drexel University, Department of Computer Science, 2005.

[22] Xu J, Kalbarczyk Z, Iyer R K. Transparent Runtime Randomization for Security[C]. *22nd International Symposium on Reliable Distributed Systems*, 2003: 260-269.

[23] Schrittwieser S, Katzenbeisser S, Kinder J, et al. Protecting Software through Obfuscation[J]. *ACM Computing Surveys*, 2016, 49(1): 1-37.

[24] Gupta A, Habibi J, Kirkpatrick M S, et al. Marlin: Mitigating Code Reuse Attacks Using Code Randomization[J]. *IEEE Transactions on Dependable and Secure Computing*, 2015, 12(3): 326-337.

[25] Lyu Y Y, Guo Y F, Wang Z P, et al. Negative Feedback Scheduling Algorithm Based on Historical Information in SDN[J]. *Chinese Journal of Network and Information Security*, 2018, 4(6): 45-51.
(吕迎迎, 郭云飞, 王禛鹏, 等. SDN 中基于历史信息的负反馈调度算法[J]. *网络与信息安全学报*, 2018, 4(6): 45-51.)

[26] Villarreal-Vasquez M, Bhargava B, Angin P, et al. An MTD-Based Self-Adaptive Resilience Approach for Cloud Systems[C]. *2017 IEEE 10th International Conference on Cloud Computing*, 2017: 723-726.

[27] Wang Y W, Guo Y F, Liu W Y, et al. A Task Scheduling Method for Cloud Workflow Security[J]. *Journal of Computer Research and Development*, 2018, 55(6): 1180-1189.
(王亚文, 郭云飞, 刘文彦, 等. 面向云工作流安全的任务调度方法[J]. *计算机研究与发展*, 2018, 55(6): 1180-1189.)

[28] Lucas B, Fulp E W, John D J, et al. An Initial Framework for Evolving Computer Configurations as a Moving Target Defense[C]. *The 9th Annual Cyber and Information Security Research Conference*, 2014: 69-72.

[29] John C. Knight. Diversity. In Lecture Notes in Computer Science 6875, 2011.

[30] Bangalore A K, Sood A K. Securing Web Servers Using Self Cleansing Intrusion Tolerance (SCIT)[C]. *2009 Second International Conference on Dependability*, 2009: 60-65.

[31] Tong Q, Zhang Z, Zhang W H, et al. Design and Implementation of Mimic Defense Web Server[J]. *Journal of Software*, 2017, 28(4): 883-897.
(仝青, 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.)

[32] Nguyen Q, Sood A. Realizing S-Reliability for Services via Recovery-Driven Intrusion Tolerance Mechanism[C]. *2010 International Conference on Dependable Systems and Networks Workshops*, 2010: 176-181.

[33] Nguyen Q L, Sood A. Improving Resilience of SOA Services along Space-Time Dimensions[C]. *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, 2012: 1-6.

[34] Nguyen Q L, Sood A. Designing SCIT Architecture Pattern in a Cloud-Based Environment[C]. *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops*, 2011: 123-128.

[35] Nagarajan A, Nguyen Q, Banks R, et al. Combining Intrusion Detection and Recovery for Enhancing System Dependability[C]. *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops*, 2011: 25-30.

[36] Sengupta S, Chowdhary A, Sabur A, et al. A Survey of Moving Target Defenses for Network Security[J]. *IEEE Communications Surveys & Tutorials*, 2020, 22(3): 1909-1941.

[37] Debroy S, Calyam P, Nguyen M, et al. Frequency-Minimal Moving Target Defense Using Software-Defined Networking[C]. *2016 International Conference on Computing, Networking and Communications* (*ICNC*), 2016: 1-6.

[38] Zhang H Q, Lei C, Chang D X, et al. Network Moving Target Defense Technique Based on Collaborative Mutation[J]. *Computers & Security*, 2017, 70: 51-71.

[39] Colbaugh R, Glass K. Predictability-Oriented Defense Against Adaptive Adversaries[C]. *2012 IEEE International Conference on Systems, Man, and Cybernetics*, 2012: 2721-2727.

[40] Jafarian J H H, Al-Shaer E, Duan Q. Spatio-Temporal Address Mutation for Proactive Cyber Agility Against Sophisticated Attackers[C]. *The First ACM Workshop on Moving Target Defense*, 2014: 69-78.

[41] Cai G L, Wang B S, Wang T Z, et al. Research and Development of Moving Target Defense Technology[J]. *Journal of Computer Research and Development*, 2016, 53(5): 968-987.
(蔡桂林, 王宝生, 王天佐, 等. 移动目标防御技术研究进展[J]. *计算机研究与发展*, 2016, 53(5): 968-987.)

[42] Yin J, Martin J P, Venkataramani A, et al. Separating Agreement from Execution for Byzantine Fault Tolerant Services[C]. *The nineteenth ACM symposium on Operating systems principles - SOSP '03*, 2003: 253-267.

[43] Distler T, Kapitza R. Increasing Performance in Byzantine Fault-Tolerant Systems with On-Demand Replica Consistency[C]. *The sixth conference on Computer systems - EuroSys '11*, 2011: 91-106.

[44] Chun B G, Maniatis P, Shenker S, et al. Attested Append-only Memory: Making Adversaries Stick to Their Word[C]. *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles - SOSP '07*, 2007: 189-204.

[45] Amir Y, Coan B, Kirsch J, et al. Prime: Byzantine Replication under Attack[J]. *IEEE Transactions on Dependable and Secure Computing*, 2011, 8(4): 564-577.

[46] J.H. Lala, Organically Assured & Survivable Information Systems (OASIS): Foundations of Intrusion Tolerant Systems[M]. Los Alamitos: IEEE Computer Society Press, 2003.

[47] Chadha R, Sistla A P, Viswanathan M. Verification of Randomized Security Protocols[C]. *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, 2017: 1-12.

[48] Moniz H, Neves N F, Correia M, et al. RITAS: Services for Randomized Intrusion Tolerance[J]. *IEEE Transactions on Dependable and Secure Computing*, 2011, 8(1): 122-136.

[49] Yuan S, Guo Y B, Liu W. Research on Voting Algorithm in NMR and NVP System[J]. *Application Research of Computers*, 2008, 25(11): 3463-3467.
(袁顺, 郭渊博, 刘伟. NMR 及 NVP 系统中表决算法分析与研究[J]. *计算机应用研究*, 2008, 25(11): 3463-3467.)

[50] Hosek P, Cadar C. VARAN the Unbelievable[J]. *ACM SIGARCH Computer Architecture News*, 2015, 43(1): 339-353.

[51] Cox B, Evans D, Filipi A, et al. N-Variant Systems: A Secretless Framework for Security through Diversity[C]. *USENIX Security Symposium*, 2006: 105-120.

[52] Garcia M, Neves N, Bessani A. SieveQ: A Layered BFT Protection System for Critical Services[J]. *IEEE Transactions on Dependable and Secure Computing*, 2018, 15(3): 511-525.

[53] Wu J X. Research on Cyber Mimic Defense[J]. *Journal of Cyber Security*, 2016, 1(4): 1-10.
(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.)

[54] Ren Q, Wu J X, He L. Research on Mimic DNS Architectural Strategy Based on Generalized Stochastic Petri Net[J]. *Journal of Cyber Security*, 2019, 4(2): 37-52.
(任权, 邬江兴, 贺磊. 基于GSPN的拟态DNS构造策略研究[J]. *信息安全学报*, 2019, 4(2): 37-52.)

[55] Forrest S, Somayaji A, Ackley D H. Building Diverse Computer Systems[C]. *The Sixth Workshop on Hot Topics in Operating Systems* (*Cat. No.97TB100133*), 1997: 67-72.

[56] Posnett D, D'Souza R, Devanbu P, et al. Dual Ecological Measures of Focus in Software Development[C]. *2013 35th International Conference on Software Engineering*, 2013: 452-461.

[57] Zhang M Y, Wang L Y, 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 and Security*, 2016, 11(5): 1071-1086.

[58] Ma D H, Wang L M, Lei C, et al. Quantitative Security Assessment Method Based on Entropy for Moving Target Defense[C]. *The 2017 ACM on Asia Conference on Computer and Communications Security*, 2017: 920-922.

[59] Boehm B W. Software Cost Estimation Meets Software Diversity[C]. *2017 IEEE/ACM 39th International Conference on Software Engineering Companion*, 2017: 495-496.

[60] Yong W, Qiang D, Dick S. Security Evaluation Using Software Diversity Measurement: An Ecological Approach[C]. *2016 International Conference on Software Engineering Research and Practice*, 2016:95-101.

[61] Neti S, Somayaji A, Locasto M E. Software Diversity: Security, Entropy and Game Theory[C]. *the 7th USENIX conference on Hot Topics in Security*, 2012:5.

[62] Mitra S, Saxena N R, McCluskey E J. A Design Diversity Metric and Reliability Analysis for Redundant Systems[C]. *International Test Conference 1999. Proceedings* (*IEEE Cat. No.99CH37034*), 1999: 662-671.

[63] Lyu M R, Chen J H, Avizienis A. Software Diversity Metrics and Measurements[C]. *[1992] Proceedings. The Sixteenth Annual International Computer Software and Applications Conference*, 1992: 69-78.

[64] Koopman P, DeVale J. Comparing the Robustness of POSIX Operating Systems[C]. *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing* (*Cat. No.99CB36352*), 1999: 30-37.

[65] Garcia M, Bessani A, Gashi I, et al. Analysis of Operating System Diversity for Intrusion Tolerance[J]. *Software*: *Practice and Experience*, 2014, 44(6): 735-770.

[66] Garcia M, Bessani A, Gashi I, et al. OS Diversity for Intrusion Tolerance: Myth or Reality? [C]. *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks*, 2011: 383-394.

[67] Han J, Gao D B, Deng R H. On the Effectiveness of Software Diversity: A Systematic Study on Real-World Vulnerabilities[C]. *Detection of Intrusions and Malware*, *and Vulnerability Assessment*, 2009: 127-146.

[68] Azab A, Layton R, Alazab M, et al. Mining Malware to Detect Variants[C]. *2014 Fifth Cybercrime and Trustworthy Computing Conference*, 2014: 44-53.

[69] Coffman J, Chakravarty A, Russo J A, et al. Quantifying the Effectiveness of Software Diversity Using Near-Duplicate Detection Algorithms[C]. *The 5th ACM Workshop on Moving Target Defense*, 2018: 1-10.

[70] Liu H, Zhang Z, Chen Y, et al. Feasibility Evaluation of Diverse Compilation Strategy Based on Heterogeneous Cost-Effectiveness Ratio[J]. *Journal of Information Engineering University*, 2020, 21(2): 200-206.
(刘浩, 张铮, 陈源, 等. 基于异构费效比的多样化编译策略可行性评估[J]. *信息工程大学学报*, 2020, 21(2): 200-206.)

[71] Fang D Y, Zhao Y, Wang H J, et al. Software Protection Based on Virtual Machine with Time Diversity[J]. *Journal of Software*, 2015, 26(6): 1322-1339.
(房鼎益, 赵媛, 王怀军, 等. 一种具有时间多样性的虚拟机软件保护方法[J]. *软件学报*, 2015, 26(6): 1322-1339.)

[72] Gearhart A S, Hamilton P A, Coffman J. An Analysis of Automated Software Diversity Using Unstructured Text Analytics[C]. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, 2018: 79-80.

[73] Gabel M, Su Z D. A Study of the Uniqueness of Source Code[C]. *The eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*, 2010: 147-156.

[74] Zhuang R. A Theory for Understanding and Quantifying Moving Target Defense[D]. Kansas State University, 2015.

[75] Zhuang R, DeLoach S A, Ou X M. Towards a Theory of Moving Target Defense[C]. *The First ACM Workshop on Moving Target Defense*, 2014: 31-40.

[76] Zhuang R, DeLoach S A, Ou X M. A Model for Analyzing the Effect of Moving Target Defenses on Enterprise Networks[C]. *The 9th Annual Cyber and Information Security Research Conference*, 2014: 73-76.

[77] Zheng J J, Okamura H, Dohi T, et al. Quantitative Security Evaluation of Intrusion Tolerant Systems with Markovian Arrivals[J]. *IEEE Transactions on Reliability*, 2021, 70(2): 547-562.

[78] Maleki H, Valizadeh S, Koch W, et al. Markov Modeling of Moving Target Defense Games[C]. *The 2016 ACM Workshop on Moving Target Defense*, 2016: 81-92.

[79] Han Y J, Lu W L, Xu S H. Characterizing the Power of Moving Target Defense via Cyber Epidemic Dynamics[C]. *The 2014 Symposium and Bootcamp on the Science of Security - HotSoS '14*, 2014: 23-33.

[80] Jones S, Outkin A, Gearhart J, et al. Evaluating Moving Target Defense with PLADD[R]. Office of Scientific and Technical Information (OSTI), 2015.

[81] Hu P F, Li H X, Fu H, et al. Dynamic Defense Strategy Against Advanced Persistent Threat with Insiders[C]. *2015 IEEE Conference on Computer Communications*, 2015: 747-755.

[82] Mondal S K, Sabyasachi A S, Muppala J K. On Dependability, Cost and Security Trade-off in Cloud Data Centers[C]. *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing*, 2017: 11-19.

[83] Zhuang R, Zhang S, Deloach S A, et al. Simulation-based Approaches to Studying Effectiveness of Moving-Target Network Defense[C]. *National Symposium on Moving Target Research*, 2013:15111-15126.

[84] Schmidt S, Bye R, Chinnow J, et al. Application-Level Simulation for Network Security[J]. *SIMULATION*, 2010, 86(5/6): 311-330.

[85] Eskridge T C, Carvalho M M, Stoner E, et al. VINE: A Cyber Emulation Environment for MTD Experimentation[C]. *The Second ACM Workshop on Moving Target Defense*, 2015: 43-47.

[86] Aydeger A, Saputro N, Akkaya K. A Moving Target Defense and Network Forensics Framework for ISP Networks Using SDN and NFV[J]. *Future Generation Computer Systems*, 2019, 94: 496-509.

[87] Mijumbi R, Serrat J, Gorricho J L, et al. Network Function Virtualization: State-of-the-Art and Research Challenges[J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(1): 236-262.

[88] Gorbenko A, Romanovsky A, Tarasyuk O, et al. From Analyzing Operating System Vulnerabilities to Designing Multiversion Intrusion-Tolerant Architectures[J]. *IEEE Transactions on Reliability*, 2020, 69(1): 22-39.

[89] van der Meulen M J P, Revilla M A. The Effectiveness of Software Diversity in a Large Population of Programs[J]. *IEEE Transactions on Software Engineering*, 2008, 34(6): 753-764.

[90] Bishop P G, Esp D G, Barnes M, et al. PODS—A Project on Diverse Software[J]. *IEEE Transactions on Software Engineering*, 1986, SE-12(9): 929-940.

[91] Barman D, Chandrashekar J, Taft N, et al. Impact of IT Monoculture on Behavioral End Host Intrusion Detection[C]. *The 1st ACM workshop on Research on enterprise networking*, 2009: 27-36.

[92] Gallagher M, Biernacki L, Chen S B, et al. Morpheus: A Vulnerability-Tolerant Secure Architecture Based on Ensembles of Moving Target Defenses with Churn[C]. *The Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019: 469-484.

[93] Li W C, Zhang Z, Wang L Q, et al. The Modeling and Risk Assessment on Redundancy Adjudication of Mimic Defense[J]. *Journal of Cyber Security*, 2018, 3(5): 64-74.
(李卫超, 张铮, 王立群, 等. 基于拟态防御架构的多余度裁决建模与风险分析[J]. *信息安全学报*, 2018, 3(5): 64-74.)

仝青 于 2017 年在信息工程大学计算机科学与技术专业获得硕士学位。现在信息工程大学网络空间安全专业攻读博士学位。研究领域为网络空间安全。研究兴趣包括: 网络空间主动防御、移动目标防御、拟态防御等。Email: szbnlllskd@163.com

郭云飞 现任信息工程大学信息技术研究所教授、博士生导师。研究领域为网络空间安全。研究兴趣包括: 网络空间主动防御、内生安全等。Email: gyf@ndsc.com.cn

霍树民 于 2015 年在国防科技大学信息与通信工程专业获得博士学位。现任信息工程大学信息技术研究所副研究员。研究领域为网络空间安全。研究兴趣包括: 内生安全、人工智能安全等。Email: huoshumin123@163.com

王亚文 于 2019 年在信息工程大学网络空间安全专业获得博士学位。现任信息工程大学信息技术研究所助理研究员。研究领域为网络空间安全。研究兴趣包括: 云计算安全、网络空间主动防御等。Email: 15738321455@163.com