软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2022,33(1):274−296 [doi: 10.13328/j.cnki.jos.006338]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

网络行为仿真综述*

符永铨[1,2], 赵 辉[2], 王晓锋[3], 刘红日[4], 安 伦[5]

[1](国防科技大学 计算机学院, 湖南 长沙 410073)
[2](鹏城实验室 网络空间安全研究中心, 广东 深圳 518055)
[3](江南大学 人工智能与计算机学院, 江苏 无锡 214122)
[4](哈尔滨工业大学 (威海) 计算机科学与技术学院, 山东 威海 264209)
[5](北京邮电大学 网络空间安全学院, 北京 100876)
通信作者: 安伦, E-mail: anlun@bupt.edu.cn

摘 要: 网络行为描述了网络上各类元素对象动态交互过程. 它以各类网络服务协议及应用为运行载体, 形成不断变化的丰富多样的网络行为, 反映出网络拓扑结构给定时间内网络上的场景特点. 网络行为仿真主要包括运行框架、背景流仿真、前景流仿真, 将生产网络环境下网络行为按需映射到测试网络环境, 提供一种按需灵活定制仿真再现能力. 网络仿真应用场景不断发展, 包括性能分析评估、产品和技术验证、网络入侵检测、网络攻防演练与研究发展等. 为总结现有研究成果和存在的不足, 分析未来发展趋势, 梳理了网络行为仿真的相关概念和研究框架, 从框架、背景流、前景流等技术层面总结了网络行为仿真的国内外研究现状, 并对相关商业产品和开源软件工具进行了系统地分析调研, 最后对网络行为仿真的未来发展进行了展望.

关键词: 网络行为; 网络靶场; 背景流; 前景流; 运行框架
中图法分类号: TP393

中文引用格式: 符永铨, 赵辉, 王晓锋, 刘红日, 安伦. 网络行为仿真综述. 软件学报, 2022, 33(1): 274–296. http://www.jos.org.cn/1000-9825/6338.htm
英文引用格式: Fu YQ, Zhao H, Wang XF, Liu HR, An L. State-of-the-art Survey on Network Behavior Emulation. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 274–296 (in Chinese). http://www.jos.org.cn/1000-9825/6338.htm

State-of-the-art Survey on Network Behavior Emulation

FU Yong-Quan[1,2], ZHAO Hui[2], WANG Xiao-Feng[3], LIU Hong-Ri[4], AN Lun[5]
[1](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)
[2](Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China)
[3](School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China)
[4](College of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai 264209, China)
[5](School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The network behavior typically describes the interaction process among different kinds of network elements, which is based on different kinds of network service protocols and applications, formulates evolving and diverse network behavior, and reflects attributes of network scenarios during certain periods on the network topology. Network behavior emulation includes runtime framework, background traffic emulation, and foreground traffic emulation which project network behaviors in the production network environment to the test cyber environment, and provides the mirroring capability of on-demand and flexible design specifications. The application scenarios of network behavior emulation continuously evolve, including performance analysis and evaluation, product and technique evaluation, network intrusion detection, and the research and development of network attack and defense techniques. To summarize existing research results and limitations, and analyze future development trends, this study seeks to category relevant definitions and research frameworks on simulating network behaviors, summarizes the state-of-the-art research progress in terms of the framework, background traffic, and

* 基金项目: 国家重点研发计划 (2018YFB0204300, 2020YFB2104003); 国家自然科学基金 (61972409, 61972182, 61672264)
收稿时间: 2020-11-16; 修改时间: 2021-01-10, 2021-03-03; 采用时间: 2021-03-25; jos 在线出版时间: 2021-04-20

foreground traffic, and systematically surveys both commercial and open-sourced software tools. Finally, this study proposes future research topics on network behavior simulation.

Key words: network behavior; cyber range; background traffic; foreground traffic; orchestration framework

我国《国家网络空间安全战略》指出, 网络空间由互联网、通信网、计算机系统、自动化控制系统、数字设备及其承载的应用、服务和数据等组成[1]. 在我国网络空间安全一级学科中, 网络空间主要包括设备层、系统层、数据层、应用层.

网络行为即网络运行过程中其上各类元素对象动态交互过程中产生的运行时环境. 它以各类用户驱动的各类网络服务协议及应用为运行载体, 用户与这些载体交互时将在网络上形成丰富多样的网络流量, 反映出给定网络拓扑结构上、给定时间内的动态场景特点[2,3].

网络行为仿真通过物理或虚拟的手段支持一定规模网络行为的模拟仿真, 将生产网络环境下可能出现的各类网络行为按需映射到实验网络环境, 提供一种按需灵活定制的仿真复现能力. 理想条件下的网络实验应当能在具备多样化场景和流量条件的大型背景网络下直接测试真实的目标应用.

互联网靶场建设中, 网络行为仿真是提高靶场实验逼真度的最直观有效的指标, 除了对网络基础设施进行仿真, 还需要对网络链路中的流量、端用户的行为等动态内容进行仿真. 考虑到大量私有网络协议尚未公开, 对其进行仿真需要以用户驱动真实软件的方式来实现. 在以场景为牵引的攻防演练中, 用户行为仿真技术作为构建真实场景的有效手段, 能够有效解决靶场场景仿真中"仿不真"的问题.

未来网络行为仿真技术需要支持更为广泛的网络安全相关应用与协议的高效仿真, 支持对用户行为、应用行为、网络行为的多视角、多层级认知要素体系的构建, 形成统一的认知概念模型, 知识库和仿真场景体系; 抽取复杂环境下人机交互的应用类型、攻防手段、场景模型框架, 建模强对抗、不确定的网络安全活动的分析空间需要大规模计算支持; 研制可重构的仿真软件架构, 实现架构、功能、流程的灵活组合与配置, 快速实现仿真运行环境定制; 实时感知任务关键网络行为仿真的关联关系、系统运行状态、环境变化, 支持网络通信智能行为模式的持续演化分析.

本文系统调研了网络行为仿真的发展情况, 在第 1 节介绍了网络行为仿真的背景概况, 对模型、模拟、仿真、实验床等核心概念进行区分, 然后介绍网络行为仿真的发展概况. 在第 2 节梳理了网络行为仿真的研究框架, 包括需求分析、组成框架、核心技术方向等. 第 3 节对网络行为仿真技术现状进行分析, 从运行框架、背景流仿真、前景流仿真 3 个方面对核心技术方向的关键成果进行梳理分析. 第 4 节介绍了网络行为仿真的商业化产品和开源工具. 第 5 节对未来的发展进行展望.

1 网络行为仿真背景

网络行为可以分为背景流和前景流, 背景流对应与用户应用不相关的应用或设备生成的网络流量等; 前景流对应与目标应用相关的网络流量. 网络行为仿真程序主要包括网络行为仿真场景规划, 用于分析与制定用户、应用、网络流程序的行为规律与规则; 网络行为仿真模型设计, 用于构建对用户、应用、网络背景前景和恶意流的计算机仿真模型表示; 网络行为仿真运行环境, 用于调度用户、应用、网络流程序的运行过程等.

网络实验床通过集中或联邦管理的软硬件设施支持数据收集、实验实施、分布协调等复杂功能, 但是财力和人力资源耗费昂贵; 网络模拟通过计算机运行对网络环境进行抽象模拟的数字模型, 模拟器支持算法功能和性能测试, 但是模拟器无法运行真实的程序, 用户必须重写网络模拟代码, 并且无法监测操作系统或硬件导致的影响; 网络仿真在真实操作系统上搭建对网络环境进行抽象的仿真器, 利用仿真器运行测试程序, 收集程序行为数据, 能使用真实的软硬件, 具有可控、可管理、可追溯、全面监测、合理成本等优势. 互联网仿真搭建能反映互联网整体特征的软硬件设施, 是网络仿真的扩展形态.

MIT 林肯实验室的 Damodaran 在国际建模与仿真 2015 年年会上对赛博空间的模型、模拟与仿真做出了概念区分[4], 当一个资产或系统的模型提供与真实资产某种操作上相关的交互接口、协议、或者特征时, 该模型被称为一个代表性模型; 可对接口或协议实施仿真 (emulation). 当真实资产能够与真实系统协议层逼真表示的代表性模型进行交互时, 代表性模型较真实系统具有易于部署、复制、重建和无物理局限的优势.

佛罗里达国际大学 Jason Liu 课题组[5,6]认为背景流对目标应用的流量生成具有显著的影响, 但是真实实验中生成背景流量通常是不真实和无法达成的, 而网络仿真提供了目标应用原生环境和应用实例之间的近似网络环境, 可以对网络链路状况进行调整, 为网络行为研究提供了更多的操作功能, 因此通过网络仿真来正确反映背景流量的影响是客观评测新应用协议的最佳可用选择. 网络仿真提供了运行在虚拟机和虚拟网络的未修改应用实施网络实验的运行环境, 并且能与真实的操作系统接口和库进行交互. 物理网络实验床提供真实的操作环境, 能直接用真实网络流量按需测试应用, 但是这种方式缺乏可控性, 无法对物理实验床不支持的应用进行测试, 并且扩展规模受限, 难以研究应用可扩展性和鲁棒性等重要问题; 网络模拟能有效分析整体设计的各个方面, 能回答“如果-那么”类型的问题, 能揭示复杂系统特征, 并行网络模拟能高效处理大规模细粒度模型, 但是模拟不等于真实, 在模拟环境下通常难以复现真实的网络流量和行为条件; 网络仿真在可控和精确性之间取得较好的折中, 但是类似于物理实验床, 规模和容量受物理条件制约, Jason Liu 教授将网络行为实验系统的核心评测指标分为: (1) 可控: 灵活创建多样化的网络场景, 能够复现不同赛博实验中的背景流量和前景流量场景; (2) 可扩展: 捕捉大规模网络活动, 如支持的拓扑类型, 支持的协议数量, 支持的应用数量, 支持的恶意软件数量, 支持的攻击种类及数量等. 仿真对象通常是大规模的、复杂的、分散化的系统实体, 仿真场景是多方的、频繁竞争的场景, 仿真行为需要收集极限条件、罕有条件下发生的事件; (3) 保真性: 能够重现重要的系统和网络效果, 如空间逼真度 (反映流量 IP 的分布特征), 时间逼真度 (应用流量的时序分布特征). 仿真的逼真度经常因为缺乏真实比对数据而难以定量评测; (4) 性能: 支持高吞吐率数据传输. 仿真性能指标包括数据传输速率、资源利用率、并发处理效率, 流量生成速率 (如比特率、包速率)、支持的网络节点数等.

以 1969 年 ARPANET 诞生为标志算起, 网络行为仿真领域历经多次升级 (如表 1 所示). 随着网络安全技术的不断发展, 网络仿真的应用场景不断发展, 包括性能分析评估、产品和技术验证、网络入侵检测、网络攻防演练与研究发展等.

表 1　网络行为仿真发展历程

版本	第一代	第二代	第三代	第四代
年代	1969–1980	1980–1990	1990–2006	2006–现在
背景	诞生了ARPANET互连网, 主要用于支撑学术研究. 当时网络数据稀少, 缺乏数据收集和分析手段, 因此采取电信网的一些数学理论模型进行数学仿真计算.	随着网络站点逐渐增多, 网络互连传输量不断增长, 出现了Ping、抓包等基本网络收集工具, 中小规模的互联网行为数据统计分析成为热点	万维网(WWW)流行, 互联网变成社会、军事、经济的基础, 互联网行为规律变得越来越复杂, 大规模时空分布的网络行为规律分析成为热点	互联网对物联网、云计算、大数据、人工智能等领域的发展起到了助推的作用, 大规模网络数据回放与合成手段不断丰富
代表仿真模型	泊松分布, 排队论等信息理论模型	Poisson 过程已不能充分反映 Internet 业务流量的特性. 互联网拥塞控制、突发等数据统计规律, 马尔科夫和回归等随机模型	自相似模型、多尺度模型、小波理论、幂律分布、长尾分布、低维表示学习等复杂网络模型	大数据、机器学习、人工智能技术
实验设施	ARPANET (1969–1990)	NSFNET (1984–1998)、ESnet (1986–现在)	物理网络实验床(PlanetLab (2002)), 专用集群仿真实验床(ModelNet (2002), EMULAB (2002), DETER (2004)), 分布式平台(VINI (2006)), 可编程设备(ORBIT (2005)), 中国教育网络CERNET (1994)、IPv6网络CNGI-CERNET2 (2004)、联邦物理实验床GENI (2005)	物理与虚拟混合仿真环境, 云平台(CloudLab (2014)), SENSLAB (2010-2013)、IOT-LAB (2014)
实验设备	大型机/小型机	服务器	服务器、终端结点	服务器、可编程网络、终端结点、物联网终端

表 2 给出当前美国、欧洲、日本的网络仿真实验床的行为仿真情况. 可以看出, 当前仿真环境以虚拟网为主, 仿真形态以虚拟机软件为主, 仿真方法通常支持数据重放/背景流仿真/前景流仿真, 仿真层次一般在网络层、传输层和应用层, 部分仿真实验床支持用户行为仿真, 仿真框架支持高层行为建模, 仿真程序以客户端、服务器的集中式实验为主.

表 2 典型网络仿真实验床项目的行为仿真概况

序号	测试床	仿真环境	仿真对象	仿真层次	仿真框架	仿真程序	仿真形态
1	Air Force Research Laboratory Cyber Experimentation Environment (CEE)	虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	高层建模框架	客户端-服务器	物理、虚拟
2	USC-ISI DeterLab	虚拟网	提供数据重放、背景流仿真、前景流仿真	用户、应用、传输、网络	高层建模框架	客户端-服务器	物理、容器虚拟化软件
3	Department of Transportation Connected Vehicle TestbedS	无线网	提供真实网络流	真实应用	无	无	物理
4	European Union FIRE Initiative	虚拟网	提供数据重放、背景流仿真、前景流仿真	、应用、传输、网络	高层建模框架	客户端-服务器	物理、虚拟机软件
5	National Science Foundation GENI	虚拟网	提供数据重放、背景流仿真、前景流仿真	用户、应用、传输、网络	高层建模框架	客户端-服务器	物理、虚拟机软件
6	NICT Japan StarBed3、JGN2+	虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	高层建模框架	客户端-服务器	物理、虚拟机软件
7	MIT Lincoln Laboratory (MIT-LL)	虚拟网	提供数据重放、背景流仿真、前景流仿真	用户、应用、传输、网络	高层建模框架	客户端-服务器	物理、虚拟化软件
8	Test Resource Management Center (TRMC) National Cyber Range (NCR)	虚拟网	提供数据重放、背景流仿真、前景流仿真	用户、应用、传输、网络	高层建模框架	客户端-服务器	物理、容器虚拟化软件
9	George Mason University (GMU) OCTAN	无线网	提供真实网络流量	无	无	无	Windows软件库
10	Open Networking Lab (ON. Lab) Software Defined Networking (SDN) Testbed	虚拟网	提供数据重放、背景流仿真、前景流仿真	网络层	SDN框架	客户端服务器	虚拟化交换机软件
11	Intel Labs Open Cirrus	虚拟网	提供真实网络流量	无	Hadoop框架	云服务	物理、虚拟机软件
12	Rutgers ORBIT	无线网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	OMF管理框架	客户端服务器	物理、虚拟化软件
13	University of Buffalo PhoneLab	Android测试床	提供真实网络流量	无	单Android手机	客户端服务器	物理
14	Iowa State PowerCyber	虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	ISEAGE建模框架	客户端服务器	物理、虚拟化软件
15	Pacific Northwest National Laboratory (PNNL) PowerNet	虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	PowerNET建模框架	客户端服务器	物理、虚拟化软件
16	Skaion Traffic Generator	物理、虚拟网	提供数据重放、背景流仿真、前景流仿真	用户、应用、传输、网络	Skaion建模框架	客户端服务器	物理、虚拟化软件
17	Department of Energy (DOE) TCIPG	物理、虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	TCIPG实验框架	客户端服务器	物理、虚拟化软件
18	Applied Communications Sciences (ACS) Virtual Ad Hoc Network (VAN)	虚拟网	提供数据重放、背景流仿真、前景流仿真	应用、传输、网络	VAN实验框架	客户端服务器	虚拟化软件

2 研究框架

DARPA 在美国国家靶场 (NCR) 项目起始阶段, 对网络行为仿真的要求是: (1) 提供易于配置的技术, 仿真整个网络的活动, 生成代表性的网络流量信息, 即背景流量; (2) 人类角色复制与程序激活 (基于主机的流量生成), 即前景流量和恶意流量. 要求提供健壮的技术, 仿真所有靶场结点的人类行为: 要求复制体能提供真实的多人之间的事件链而不需要明确的脚本活动; 复制体能实现真实网络发现的多个用户角色; 复制体行为能够随着网络环境的改变、复制的外部环境改变及网络活动的改变而改变; 复制体能模拟物理设备的物理交互活动 (如键鼠、键盘等); 复制体能驱动桌面环境的所有通用应用, 复制体能与认证系统交互等.

美国思博伦公司在 2014 年发布了网络靶场仿真能力白皮书[7], 指出靶场应具备对整个互联网的仿真能力, 支持所有在互联网上的操作, 支持对来自全世界所有国家流量的仿真, 具备对真实 IP 地址 (主机+网络地址) 的仿真; 网络靶场应具备加载关键目标基础设施的能力, 生成对这些目标的无害攻击. 流量生成器 (traffic generator) 应具备导入数据以完成目标流量生成模版的配置; 音视频占据互联网流量的绝大比重, 网络靶场应具备生成真正的 (可被互联网客户端软件所感知的) 音视频流量的能力, 对此类多媒体服务的仿真应该作为下一代网络靶场的组成部分; 网络靶场需要流量生成器完成多项功能, 仿真来自全球的真实 IP 地址, 仿真成千上万的服务器和客户机以及其上运行的各类协议和无害攻击行为, 仿真主流的互联网应用, 如主流音视频应用、主流社交软件、主流搜索引擎、主流游戏和其他娱乐软件等, 仿真大量移动设备与主流移动应用之间的流量交互, 仿真卫星信号等. 英国剑桥大学的 Schatzmann 等人[8]在研究如何高效评价对话系统中使用的用户行为模拟技术时, 提出需要评估用户模型是否能够生成类似人类的输出, 是否会产生真实用户在相同对话上下文中可能给出的响应, 评估仿真是否能够再现真实用户行为的多样性.

随着互联网技术的更新换代, 网络行为仿真领域也在不断发展, 网络行为仿真产品不断丰富. 图 1 给出了网络行为仿真领域在不同技术维度上的发展变化. 仿真环境从早期的以太网、广域网环境扩展到虚拟网络、软件定义网络、物理与虚拟混合仿真网络环境; 仿真方法从早期的数据重放、虚拟背景流量仿真扩展到对复杂网络系统行为的前景流量仿真; 仿真层次从早期无状态的数据报文流量仿真扩展到有状态的复杂交互仿真; 仿真控制模式从早期的模型独立于系统的开环控制扩展到模型与系统交互的闭环控制; 仿真框架从命令行参数、脚本文件发展到高层建模框架; 仿真程序从早期的单机单进程、单机多进程发展到客户端服务器、全分布式等系列产品; 仿真模式形态已经包括纯软件模拟、纯物理硬件模式、基于物理硬件加虚拟化软件的仿真模式等多样化的形式. 网络仿真应用的场景不断发展, 包括性能分析评估、产品和技术验证、网络入侵检测、网络攻防演练与研究发展等.

网络行为仿真程序主要包括 3 个部分: (1) 网络行为仿真运行框架, 用于调度用户、应用、网络流程序的运行过程等; (2) 背景流仿真, 构建背景流仿真模型, 依托运行框架提供背景流仿真服务功能; (3) 前景流仿真, 构建前景流仿真模型, 依托运行框架提供前景流仿真服务功能.

2.1 运行框架

仿真运行框架规划了网络行为仿真实验的组织过程, 是网络行为仿真的基础技术. 框架技术主要研究控制平面和传输平面协同的软件体系架构, 仿真流量的配置模板, 仿真过程的调度, 仿真场景的配置, 并发传输控制等, 需要适应不同流量条件、不同测试场景、多样化的网络拓扑.

已有的框架衡量指标主要包括[3]: (1) 运行框架场景编排模型能力. 该指标是指网络行为仿真场景的流程化编排能力, 系统通过预先建立好的编排模型框架为仿真场景设计用户提供场景编排服务, 便于按需组装已经封装好的前景流和背景流模板, 形成运行框架模板. (2) 运行框架有向无环图拓扑种类. 该指标是指表征运行框架的图形拓扑模式的多少, 种类越多表示运行框架的仿真流程场景构建能力越强. (3) 运行框架场景随机生成. 该指标是指运行框架是否支持运行框架场景的自动化一键式随机生成, 这是一个自动化能力核心指标, 可简化场景设计人员的工作量. (4) 运行框架并发执行能力. 该指标是考察运行框架对多用户和多运行框架同时使用系统资源的支持能力, 是否支持并发地在系统中运行, 并发度越高, 并发执行能力越强.

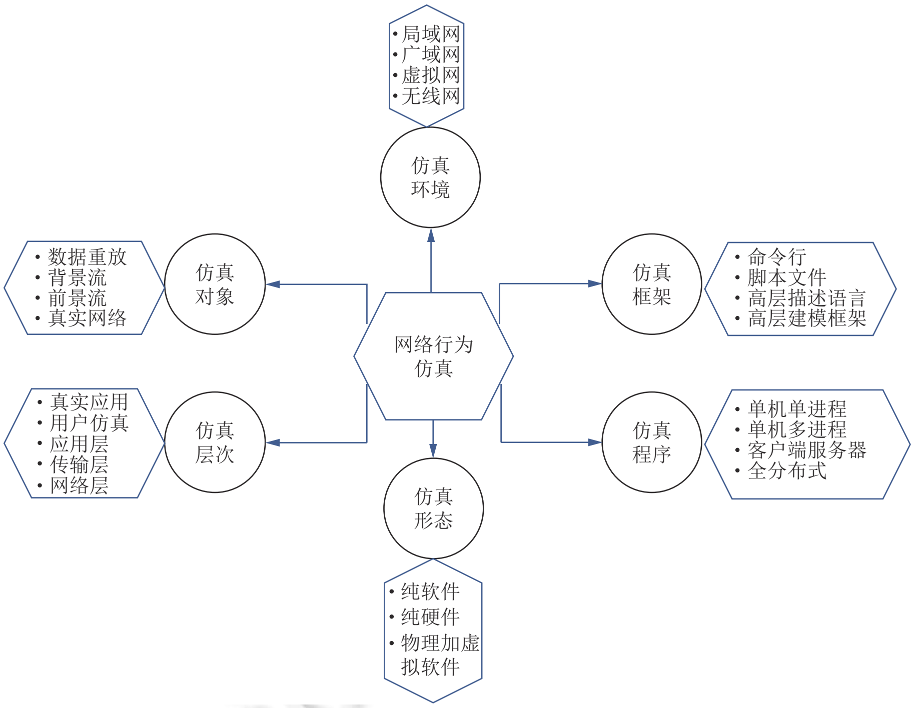

图 1　网络行为仿真框架

目前运行框架主要包括配置脚本、事件语言、智能体等类型, 配置脚本简单灵活, 具有易于实现的优势, 是最为常见的运行框架形式. 但是配置脚本难以支持计算机的自动化解析, 难以组装大型的配置脚本, 并且通常配置静态的参数, 仿真程序运行后难以动态调整参数赋值, 可控性不高; 配置脚本下发和更新缺乏通信协议支持, 难以扩展到大规模或者异构的仿真环境; 事件语言具有结构化模型的表达优势, 通常是封闭世界假设, 难以支持异构化的仿真环境和仿真场景, 预先未知的仿真应用和仿真场景; 智能体方法有助于建模预先未知的仿真场景, 较事件语言的方法更为灵活, 可以支持预先未知的仿真应用和仿真. 然而基于智能体的方法需要采用统一的语言模型构建, 对于异构化的仿真应用和仿真环境的适应能力有待提升.

2.2　背景流仿真

背景流仿真指的是通过可编程的软硬件设施向目标网络注入真实捕获或人工合成的网络流. 背景流通常定义为非目标应用产生的网络流量. 背景流将消耗大量网络资源, 影响整个网络的行为表现. 背景流仿真可用于网络性能评估、网络安全分析、服务质量评估、网络协议设计等.

已有的背景流评价指标主要包括[3]: (1) 支持模拟的应用协议模型数量、攻击流量模型数量、恶意软件流量模型数量. 该指标是指生成的流量包中可模拟多少种应用协议、攻击行为以及恶意软件的数量和行为模式. 支持的数量越多, 说明仿真模拟能力越强. (2) 不同 IP 地址的模拟用户数量. 该指标是指在进行背景流仿真时, 仿真程序最多可构建多少个不同 IP 地址的客户端, 以模拟大量来自不同地点的用户. 支持的数量越多, 说明仿真模拟能力越接近真实环境. (3) 每秒新建会话连接数量、并发连接数量、聚合流量峰值. 该指标是指生成背景流量时系统每秒能新建多少会话连接. 该指标是指在生成背景流量时系统能同时保持多少活动会话连接. 该指标是指在生成背景流量时, 所有流量生成节点在某时刻产生的流量总和.

目前背景流仿真主要包括流量回放和解析仿真等类型. 流量回放操作简单, 流量真实, 但是难以混合大规模的网络行为, 回放速度受限于网卡性能, 回放参数难以动态调整; 解析仿真不依赖于流量数据的优势, 但是生成速度通常低于流量报文回放, 仿真的逼真度难以保障, 而且难以建模混合的大规模网络行为.

2.3 前景流仿真

网络用户的行为指用户在外界环境影响下所产生的应用活动和表现出来的与应用有关的态度, 包括显性行为和隐性行为. 从空间范围延拓后显示为集群行为, 而从时间范围延拓后显示为时序行为. 目前国内外赛博实验设施大都缺乏对用户行为的仿真建模. 用户行为仿真的目的在于能够产生符合真实网络用户行为规律的群体用户, 能够灵活逼真地生成符合真实个体用户行为特征的个体用户, 以构建更加逼真的互联网场景, 提高互联网靶场的真实度, 为研究群体用户行为提供数据支撑.

已有的前景流评价指标主要包括[3]: (1) 前景应用模板数量. 该指标是指系统能仿真模拟用户操作多少种不同应用的行为. 数量越多, 说明仿真模拟能力越强. (2) 前景应用插件式接口开发能力. 该指标是指系统对模拟应用开发用户提供的开发接口格式, 以便于前景模拟应用的种类扩展. (3) 前景流序列生成能力. 该指标是指为仿真用户操作应用而基于用户输入的序列来构建操作步骤的过程, 其输入是用户的行为模式, 其输出是用户操作行为模板文件, 或称时间线文件. 生成过程越快, 说明生成能力越强.

目前前景流仿真主要包括应用行为仿真和用户行为仿真等类型, 应用行为仿真具有操作简单, 易于实现的优势, 但是难以构建反映大规模用户群体的操作分布规律, 仿真逼真度有限; 用户行为仿真包括马尔可夫模型、攻击向量模型或者博弈论工具建模等, 马尔可夫模型能够构建应用和活动的状态迁移图, 每个状态具有一个持续区间, 区间完成后基于概率转移到其他状态来模拟用户的动态行为; 攻击向量模型或者博弈论工具建模能够描述恶意用户行为, 较应用行为仿真更适用于面向群体行为的仿真场景, 但是这种方法缺乏真实数据集支持, 采集到的数据或数学模型难以反映真实场景下的用户行为.

3 技术现状

3.1 运行框架

运行框架按照框架的自动化程度可以分为 3 类, 如图 2 所示, 分别为: (1) 基于配置脚本的方法; (2) 基于事件语言的方法; (3) 基于智能体的方法等.

(1) 配置脚本

基于配置脚本的方法采用参数赋值的脚本文件控制和编排仿真程序, 具有易于实现的优势, 是最为常见的运行框架形式. 但是配置脚本难以支持计算机的自动化解析, 难以组装大型的配置脚本, 并且通常配置静态的参数, 仿真程序运行后难以动态调整参数赋值, 可控性不高; 配置脚本下发和更新缺乏通信协议支持, 难以扩展到大规模或者异构的仿真环境.

2009 年, 美国犹他大学的几位研究人员[9]针对大部分网络仿真器局限于网络链路特性仿真问题, 提出了一种基于路径的端到端流量建模方法. 通过对 Dummynet 扩展实现路径仿真, 部署到 Emulab 上执行. 2010 年, 日本情报通信研究所学者 Miyachi 等人[10]在 TridentCom 2010 会议论文集上发表了流量生成平台 XBurner, 该平台基于日本的国家靶场 starbed 定制环境, 基于场景生成背景流量, 通过运行真实的应用 (如 Apache、wget、FTP、ping 等) 生成背景流量. 软件架构包括基础层、邻居层 (构建用于插入网络特征的拓扑) 和应用触发层 (控制应用软件). 通过 SpringOS 触发应用软件, 由真实的应用软件生成真实流量. 利用 XENbula 配置虚拟网络实验环境, 利用 AnyBed 配置网络实验参数. 在 Starbed 运行评测的结果显示吞吐量可以达到约 375 MB/s. 2011 年, 美国佛罗里达国际大学 Jason Liu 团队[11]在 GENI 实验床上增加了实时网络模拟, 虚拟网络能与真实实现的网络协议交互. 软件由物理资源层、元资源层、模拟与仿真执行层和会聚管理器组成. 通过 ProtoGENI 控制框架管理、控制和访问资源. 通过 Python 脚本配置网络模型, 利用仿真的端结点生成仿真流量. 实验能够与真实网络进行交互, 但是不能回放现实网络中的流量, 无法还原现实网络的原场景. 2017 年, 美国普渡大学 Cao 等人[12]将网络仿真实验映射到由多个 (可能是异构的) 物理机器组成的集群, 通过自动配置信息以及实验拓扑的特征来确定一个有效的映射以保持性能保真度. 2017 年, 美国伊利诺伊理工学院的 Wu 等人[13]提出了基于容器和物理设备混合搭建实验床, 支持高可信和重复的网络实验. 该框架基于容器环境运行真实的网络应用, 通过控制信道执行定制脚本来配置网络

实验. 2018 年, 西班牙 Pastor 等人[14]开发了基于 NFV/SDN 场景下的用于机器学习训练和验证的综合网络测试框架 MouseWorld, 通过组合不同类型的网络安全威胁、网络服务来生成旨在解决机器学习相关网络安全调整的方案. 2019 年, 美国海军研究实验室利用 MGEN[15]生成网络流量, 实现了支持大多数的流量整形功能. 通过定义 RAPR 输入脚本、行为表、RAPR 运行时接口进行配置. RAPR 输入脚本设置应用行为仿真的时间线; 运行时接口注入和调度额外的行为, 行为表定义应用生成的事件行为. 可以通过额外的词典用来翻译应用和脚本使用的键-值对. 该方法无法还原真实网络中的流量. 美国波音公司开源的 CORE[16]建立了实时运行的真实计算机网络的表示形式, 与使用抽象模型的模拟相反, 具有直观的可视化图像界面. 实时运行的 CORE 仿真可以连接到真实网络和路由器中. 它利用 Linux 操作系统提供的虚拟化功能为运行实际的应用程序和协议提供仿真环境.

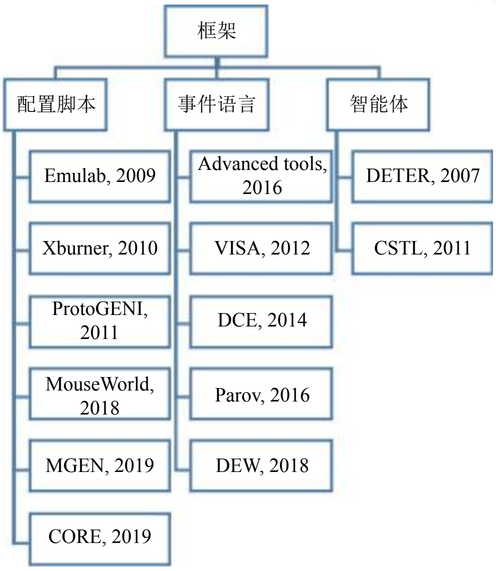

图 2 框架研究内容分类

未来基于配置脚本的运行框架需要支持动态的参数赋值, 支持可控灵活的配置脚本分发和更新, 扩展到大规模和异构的仿真环境, 优化计算机的自动化解析和模块化组装能力.

(2) 事件语言

基于事件语言的方法试图建立运行框架的形式化描述、资源和约束表示, 仿真交互过程的语法和语义表示模型等, 支持计算机的自动化解析和组件化处理, 可控性和可扩展性优于配置脚本方法, 但是事件语言通常是封闭世界假设, 难以支持异构化的仿真环境和仿真场景, 预先未知的仿真应用和仿真场景.

美国林肯实验室 MIT-LL 开发了基于标准的事件描述语言, 基于该语言开发了一系列用于实现靶场功能组件的工具集合[17]. 在互联网仿真方面开发了根域名服务仿真, 在靶场内实现域名解析, 自动构建虚拟用户, 并将其分布于不同节点, 在靶场内部实现离线网站副本仿真互联网内容服务, 通过自动化控制虚拟用户与靶场内预装的互联网应用交互生成网络流量. 2012 年, 德国 Detken 等人[18]提出了面向基于系统模型的半自动部署实验的仿真环境, 通过仿真编译器 (SC) 将 TE 参数集转换为仿真定义, 这种仿真定义可以包括用于自动化实验的特定数据源、虚拟网络接口的定义、虚拟映像的配置数据以及监视仿真的请求. SC 将仿真定义传递给控制器, 以根据给定的仿真定义执行所有操作 (可能包括启动 OMF). 2014 年, 法国 INRIA 机构 Camara 等人[19]利用 NS-3 的直接代码执行 DCE 环境, 在模拟器中集成 Linux 内核和应用代码, 能运行不修改的网络系统软件, 用于模拟网络环境. 该框架由内核层 (虚拟化原语, 同步和调度 NS-3 模拟进程); kernel 层 (将真实的网络协议栈嵌入到 NS-3); POSIX 层 (重新实现标准的 socket API)3 部分组成. 将吞吐率、执行时间和处理报文速度作为评测指标. 该环境无法运行 Window 程序. 2016 年, 伊朗沙里夫大学 Momeni 和 Kharrazi[20]利用 Partov 拓扑语言定义虚拟网络, 标识执行核心

模拟 (如路由包) 的网络节点, 指定节点的类型 (例如无线接入点), 配置指定的协议, 使用链路延迟、包丢失概率和最大带宽进行注释, 采用马尔可夫模型和时序蒙特卡罗模型构建数据模型元素. 2018 年, 美国南加州大学 Mirkovic 等人[21]提出了实验描述框架的结构化通用语言 DEW, 包括场景 (what、who)、绑定 (how)、约束 (where), 利用高层表示的脚本工具生成实验描述, 支持从实验到高层表示的反向表示. 2020 年, 中国国防科技大学王帅等提出了基于容器的仿真程序封装方法[22], 提升了仿真的模块化水平. 中国国防科技大学刘彪等人提出了容器化工作流框架方法, 拓展了工作流的部署灵活性[23]. 中国陆军工程大学纪一方等提出了用户行为仿真程序的部署优化方法[24], 提升了部署效率.

未来基于事件语言的方法需要扩展语言的覆盖能力, 易于增加预先未知的仿真应用和仿真场景, 支持异构化的仿真环境和仿真场景.

(3) 智能体

基于 Agent (智能体) 的方法采用 Agent 技术封装实验行为模型, 采用面向 Agent 的调度器部署和运行 Agent 模型, 智能体模块支持标准的方法, 如配置、启动、停止等方法, 智能体角色可以是执行者, 如 Web 服务器智能体, 也可以是传感器, 如对特定链路监测. 不同智能体通过消息总线交互. 较事件语言的方法更为灵活, 可以支持预先未知的仿真应用和仿真. 然而基于智能体的方法需要采用统一的语言模型构建, 对于异构化的仿真应用和仿真环境的适应能力有待提升.

美国南加州大学的 DETER 平台包括校园的 204 结点和在伯克利大学的 96 结点, 利用 Purdue Tool Suite 进行配置[25], 支持的数学模型有: 流量合成类 (SEER, Tcpreplay, Performance Test Tools, Webstone, NTGC, TCP Opera, Harpoon 等); 拓扑生成类 (Rocketfuel-to-ns, Inet, Brite, GT-ITM 等). 美国南加州大学 DETER 项目开发了 MAGI 网络行为仿真框架. 该框架由消息总线, 智能体模块, 调度器模块组成. DETER 项目提供一个智能体库, 实现了基本的功能函数, 如流量生成与监控, 包含了生成 Web、FTP、SSH、VoIP、IRC 流量和恶意 Dos 攻击流量的智能体. MAGI 框架支持定制开发新的智能体. 调度器模块利用代理和消息总线实现实验的控制流, 提供确定性实验控制. 实验的过程表示为一个事件与触发器流, 描述了正确执行实验的偏序关系. 一个事件能够激活智能体的特定行为, 而触发器提供了基于事件和基于时间的控制流同步点. MAGI 框架通过 YAML 格式的代理激活语言 AAL 表示一个实验的控制流. AAL 语言模板由分组指令、代理指令、事件流指令组成. 分组指令定义了实验结点上消息总线的分组情况, 代理指令定义了实验结点的功能行为情况, 事件流指令是 AAL 模板的主体, 包含了一个事件和触发器的列表. 2011 年, 洛克希德·马丁公司为美国国家靶场 NCR 项目专门开发了科学测试方法 (cyber scientific method) 和基于 OW2 Web 本体语言的赛博科学测试语言 (cyber scientific test language, CSTL)[26], 二者共同对待测试系统提供形式化描述, 描述内容包括测试注入、测试环境、测试假设与目标以及测试结果等. 靶场软件利用 CSTL 语言的测试规格说明自动化构建测试床, 通过测试事件插入开展实验. 在测试结束后, 通过 CSTL 描述的测试结果能够进行事后分析处理. 通过 CSTL 的网络流量生成模板 (recipe) 调度管理网络中的控制结点生成人工网络流量; 同时调度待测试系统主机上的注入软件合成网页浏览或者邮件的流量. 该软件利用流量生成模版来生成流量, 但并不具有还原真实网络流量的功能. 2019 年, 意大利罗马大学计算机、控制与管理工程系与意大利网络安全国家实验室研究常用协议 (HTTP, HTTPS, SSH, SMB, SFTP) 仿真技术[27], 通过定制化 Agent 软件仿真用户行为. Agent 软件由一组流量仿真任务构成, 每个任务作为一个工作线程, 由调度器统一管理, 调度器按照预先设定好的流量生成模型 (包括协议类型、开始及结束时间、频率, 执行节点等设定) 将任务调度到指定的结点执行, 但 Agent 的工作效率完全依赖调度器, 并且调度器在研究中没有指明具体的工作原理.

未来基于智能体的方法需要进一步提升对异构化的等仿真应用环境的支持, 支持不同语言的仿真程序集成运行, 支持虚机、容器等虚拟化环境的快速部署.

综上比较, 可以看出, 基于配置脚本的框架和基于事件语言的框架比例最高, 具有通用性好、实现简单的优势; 基于智能体的方法、基于本体语言和智能体架构实现智能化的框架架构, 具有适应性强的优势.

3.2 背景流仿真

背景流仿真的研究按照是否生成虚拟流量可以分为两类: (1) 真实互联网背景流回放仿真技术, 是研究如何提

取和回放真实互联网流量时空特征的回放仿真技术, 能高效回放真实的网络流记录、网络流量矩阵, 适应互联网流量行为持续变化; (2) 时空关联的背景流解析仿真技术, 是研究如何提取和回放真实互联网流量的时空特征的解析模型仿真技术, 能通过解析模型复现单独路径/链路的突发流量行为, 而且在目标网络拓扑上放置的网络流量能复现全局拓扑范围的网络流量分布.

(1) 流量回放

基于流量回放的方法将预先的流量文件修改流量报文头部信息, 通过物理或者虚拟网卡重新发送和接收流量报文, 具有操作简单, 流量真实的优势, 但是难以混合大规模的网络行为, 回放速度受限于网卡性能, 回放参数难以动态调整.

2004 年, 美国加州大学 Cheng 等人设计了一套流量回放系统 Monkey[28]. 该系统首先捕获接近服务器端通信的 TCP 流量, 进而提取流量特征 (如时延、带宽、响应延迟等), 最后并在另一台节点中回放. 通过此方式实现服务器性能测试. 实验证明了该系统的正确性以及可重复性. 2015 年, MIT 研究人员提出一种 HTTP 应用流量精准采集与重放框架 Mahimahi[29]. 更加精准地仿真 Web 应用的多服务器特征, 采用多个 Mahimahi 实例并行仿真多个不同的网络, 通过一套可组合 shell 套件提供易用性和扩展性. 该框架的评测指标为页面载入时间、速度索引-页面展示区域完成度. 论文中指出该框架不能仿真任意网络拓扑 (比如 transit-stub 结构). 2015 年, 美国密歇根大学 Durumeric 等人[30]提出一个公共搜索引擎和数据处理工具 Censys, 旨在帮助研究人员回答与安全性相关的问题, 支持在协议标语上进行全文搜索并查询各种派生字段, 可以识别特定的易受攻击的设备和网络, 并生成有关广泛使用模式和趋势的统计报告. 该工具可作为识别和统计工具使用, 但并不能实现模拟功能. 2021 年, 中国江南大学黄宁等提出了面向 OpenStack 环境的流量回放方法[31], 利用延迟优化技术提高了多点流量回放的逼真性.

未来基于流量回放的方法需要支持混合大规模的网络行为数据, 提升回放速度, 支持回放的参数动态灵活定制, 支持大规模的多点实时回放.

(2) 解析仿真

基于解析仿真的方法利用预先构建的网络流量行为模型直接生成网络流量报文, 通过物理或者虚拟网卡发送合成的流量报文, 具有不依赖于流量数据的优势, 但是生成速度通常低于流量报文回放, 仿真的逼真度难以保障, 而且难以建模混合的大规模网络行为.

2004 年, 美国维斯康星大学 Sommers 等人提出了流一级的开源流量生成器 Harpoon[32], 用于 3–4 层网络协议的背景流量生成. Harpoon 软件架构采用 C/S 模式, 通过脚本配置控制软件的运行. 在于真实流量进行对比验证时, Harpoon 采用分层的流量模型, 分别为文件级、会话级和用户级. 传统的五元组流模型类似于用户级模型, 作为 Harpoon 的最高层模型, 用户会参与到很多会话当中, 每个会话有会话级模型, 每个会话又由两个不同 IP 之间传输的一组文件构成, 会话模型由 IP 空间分布特性、会话开始时间和会话持续时间 3 方面刻画, 文件级模型是最底层模型, 该模型由文件尺寸和相邻文件请求时间间隔两方面构成. 通过与现网流量对比评估测试, 发现该模型生成的流量在生成包数量和流持续时间上与真实情况有出入, 模型逼真度还有待改进. 2006 年, 美国克莱姆大学 Weigle 等人[33]提出针对 TCP 应用负载的流量生成工具 Tmix, 并基于著名的网络模拟器 NS-2 给出了模块化实现, 集成到 NS-2 平台之中. 该工作只针对 TCP 应用行为进行模拟, 运行于模拟环境而非仿真网络环境, 无法直接在仿真实验床环境下工作. 2006 年, 美国 UCSD 大学 Vishwanath 等人[34]设计并实现一种基于网络响应的流量生成工具 Swing. 该工具通过将捕获的应用软件中交互过程中产生的数据包使用一种结构化模型, 导出其中的用户, 使用应用软件以及用户网络行为的分布, 并利用该分布使用合适网络协议生成流量. 并且, 第一次实现了使用支持大多数硬件设备的模型在多种时间尺度下复现流量的突发性. Swing 通过一个简单的结构化模型实现对大量网络应用交互性流量包的精确捕捉. 2008 年, 德国慕尼黑大学的研究人员 Khayari 等人[35]于 International Symposium on Performance Evaluation of Computer and Telecommunication Systems 会议上提出了一个基于人工参数合成的流量生成器. 研究考虑了 WWW 流量的多种统计特征, 指出这种人工合成的方法比基于真实流量回放方法更灵活可控. 通过与基于现网流量生成流量的软件进行对比, 证明了人工方法可以拟合现网流量. 2012 年, 德国汉堡大学学者 Kolesnikov[36]于发表在 MMB & DFT 2012 会议论文集上提出了 UniLoG, 采用通用的基于有限用户状态自动机

(UBA) 的负载抽象表示模型, 每个负载生成节点部署一个 UniLoG agent 实例, 负责成负载, 中心管控节点通过命令发布抽象负载生成命令到每个 Agent 节点. 2012 年, 中国西安交通大学褚伟波等人[37]对互动式网络场景再现进行深入分析, 认为网络流量生成方法包括基于仿真的流量再现方法和基于真实网络流量还原回放的方法. 基于仿真的流量再现方法控制灵活, 但是仿真流量和实际流量差异较大; 网络流量回放能直接考察被测系统的功能和性能. 褚伟波等人提出了一种网络流量回放的速率控制方法, 将网络场景再现过程中的流速控制问题转化为目标跟踪控制问题, 然后采用基于函数近似器的方法描述系统的输入输出关系, 通过动态调整系统输入流量对回放过程输出流量进行跟踪. 2014 年, 美国佛罗里达国际大学 Li 和 Liu[38]使用数据聚类技术将终端主机的行为描述为多维属性函数, 并将其分组为不同的类, 然后将这些类映射到模拟路由器, 以便可以根据群集级别的统计信息生成流量, 它还能够缩放生成的流量, 以便可以相应地更改流量强度, 以便在不同而又现实的网络条件下测试应用程序. 2015 年, 美国佛罗里达国际大学 Jason Liu[6]设计了网络队列模型-仿真中再生同样的模拟流量条件, 模型缩放方法-降低网络模型计算复杂性, 仿真流量行为再生方法-模拟中捕捉目标应用的流量需求, 创建对应的模拟网络流, 将网络应用和协议的真实实例运行在真实操作环境和模拟网络环境. 2015 年, 德国慕尼黑工业大学 Emmerich 等人[39]在 ACM IMC 会议上提出称为 MoonGen 的高速流量包生成器. 基于数据报文处理框架 DPDK 和 LuaJIT 架构生成高速数据报文, 支持单机运行模式, 支持常见网络协议, 软件架构包括: 用户脚本、配置 API 及 MoonGen 核心模块. 2015 年, 中国国家计算机网络与信息安全管理中心唐积强等人提出了基于数据流特征的网络流量生成方法和基于数据包特征的网络流量生成方法[40], 能生成随时间而波动变化的、符合单位时间内特定属性特征的网络数据流量. 2017 年, 美国海军研究生院 Fortner 等人[41]设计了一个平衡协议语义与流速的流量回放工具 DSSR, 将流量按照流向分为两类, 内向流量和外向流量. 在保证协议语义相关的数据包顺序正确的情况下, 通过调整外向流量的发送时间戳来控制流速, 进而模拟虚拟网络延时. 局限在于该方法仅适用于通信网络协议较少且通信过程完整的场景, 不具备通用性. 2017 年, 澳大利亚阿德莱德大学 Tune 等人[42]提出了可控的流量矩阵生成方法, 利用球形加性噪声模型 (SANM) 和模型结合迭代比例拟合方法 (IPF) 在围绕预测的流量矩阵快速生成合成的流量矩阵集合. 作者从理论上分析了该方法的渐进收敛性, 特别是能收敛到著名的重力模型 (一种关注空间特性的基于流量矩阵的流量模型). 2019 年, Javali 和 Revadigar 提出了一种面向靶场的 Web 流量生成器[43]. 采用马尔可夫模型、Dirichlet 分布和混合分布构建混合行为模型, 将真实数据和输出数据的熵以及交叉熵作为评测指标, 评测结果显示生成器和真实数据比较接近. 2019 年, 美国海军研究实验室推出了更新版的开源软件 MGEN[15], 主要采用 TCP 和 UDP/IP 流量对 IP 网络性能进行测试评估. 基于脚本记录流量特征并生成真实的流量模板, 支持单播、组播、TCP、UDP 应用. 通过命令行脚本控制程序的运行, 以 Agent 方式部署在节点上, 生成并传输报文数据. 局限在于该软件仅适用于 NS-2、NS-3 这类网络模拟环境, 仅支持单节点背景流量生成, 不具备大规模分布式流量生成能力, 主要是缺乏分布式管控框架的支持.

未来基于解析仿真的方法需要提升报文生成速度, 借鉴流量回放的数据统计规律提升大规模流量仿真的逼真度.

3.3 前景流仿真

按照前景流的协议层次类型, 可以将前景流仿真技术分为应用行为仿真和用户行为仿真两类. 应用行为仿真当前相对成熟, 而用户行为仿真是目前前景流仿真技术的研究热点.

(1) 应用行为仿真

应用行为仿真方法通过真实运行或者仿真模拟的手段操作应用程序, 产生应用程序的操作序列, 具有操作简单, 易于实现的优势, 但是难以构建反映大规模用户群体的操作分布规律, 仿真逼真度有限.

2013 年, 澳大利亚昆士兰科技大学的 Bhatia 等人提出了流量生成和测试平台框架来综合生成各种现实的 DDoS 攻击和其他良性流量跟踪[44]. 软件流量生产器 (Botloader) 由 Bots, clusters, modules (攻击模块和数据模块) 组成, 使用 IP 别名在一台计算机上创建数千个交互式 UDP/TCP 端点, 同时模拟大量攻击者或正常者的客户机, 模拟正常流量部分和原始 CAIDA 数据集的攻击. 2016 年, MIT 林肯实验室研制了集成化仿真技术, 其中包括: (1) 用户仿真, 通过创建大量虚拟用户来仿真互联网上的真实用户[45]; (2) DNS 服务仿真, 仿真互联网根域名服务系统结

构提供 Web 站点名称到 IP 地址的映射; (3) Web 内容服务仿真, 通过采样数万个 Web 站点, 对其内容进行动态镜像, 实现对 Web 内容服务的仿真; (4) Web 应用仿真, Web 应用占据互联网绝大部分流量, 通过服务代理方式获取用户与 Web 应用之间的交互模型; (5) 背景流量仿真, 使靶场中每个虚拟用户通过程序控制系统中已安装的网络应用, 让其自动运行操作, 以此来仿真互联网环境下真实用户所产生的网络流量. 该方法比简单的协议流量仿真器更加逼真, 但由于增加了复杂性, 其扩展规模受到限制, 且需要预先对各类用户的不同行为模式进行人工建模, 很难实现批量自动化建模. 2016 年, 日本高科技研究所提出前景流仿真系统 CyRIS[46], 系统提供无害恶意仿真程序 (dummy malware). 该程序支持两种运行模式, 模式一是启动一个计算任务并控制其消耗的 CPU 资源; 模式二是将恶意软件运行为一项服务, 通过任意端口监听网络流量, 用户通过设定名称和模式来定义和部署这种无害恶意程序, 部署后该程序将出现在后台运行程序列表, 为系统仿真一种无害的威胁. 此外, 系统支持插件方式自定义恶意软件, 以扩展恶意软件的仿真能力. 2017 年, 美国 CMU 软件工程研究所 Dobson 等人提出了真实-环境-对手-通讯-战术-角色 R-EACTR 设计框架[47], 重点强调真实性, 支持 30 多个真实的演习场景. 该框架利用真实的基于互联网的 HTTP&DNS 网络流量生成数据. 该框架覆盖 Windows 办公环境、Linux 应用环境、Linux BIND DNS、HBSS/McAfee ePolicy 调度器、思科路由器、防火墙、流量分析、流量审计等网络设备. 能够模拟 BGP 路由, 互联网站点, 根 DNS 服务器, 预定义和动态的红方 IP 地址段. 局限性在于该框架实现的协议有限, 难以涵盖真实网络环境中的各种协议类型. 2018 年, MIT 林肯实验室开发了一种方法来生成符合网络靶场要求的真实的电子邮件代理[48], 该方法涉及一个分布式线程模型, 以避免中央协调. 它通过可编辑模板和可编辑线程模型进行配置, 从而允许对网络范围事件进行可能需要的微调. 它是数据驱动的, 使用从语料库中学习的模板、主题和线程图, 这比手工编写方法更省时. 但是在代理的发件行为中, 主题映射是根据电子邮件的词汇特征确定的, 在复杂的情况下, 无法确定邮件的主题, 也没有提及使用何种查找方法来应对该问题. 2017 年, 美国 UIUC Babu 通过合成高时态可信的数据集[49], 描述正常状态下电网网络空间元素和物理元素的交互, 系统架构包括仿真主机运行仿真驱动、重放驱动、PowerSim 驱动-应用层线程、攻击层线程、网络层线程. 仿真控制结点发出控制指令, 指令传送到目的主机, 主机通过代理将指令传给电力模拟器. 主机的仿真驱动相互通信生成测试床流量, 重放驱动在特定时间点重放流量, PowerSim 驱动传输 PowerWorld 电力状态数据.

未来应用行为仿真需要集成化大规模、异构的用户群体在不同空间、不同时间、不同规模下的应用的操作行为, 提升仿真的逼真度和仿真性能.

(2) 用户行为仿真

用户行为仿真通过真实采集或者模型合成的方法构建用户的操作序列, 较应用行为仿真更适用于面向群体行为的仿真场景, 但是这种方法缺乏真实数据集支持, 采集到的数据或数学模型难以反映真实场景下的用户行为.

美国林肯实验室 MIT-LL 为 LARIAT 开发了系列的用户仿真工具[45]. 早期基于增强马尔可夫模型的用户建模工具“Petri Nets”, 能够执行简单的机器操作指令, 如切换任务, 利用人类行为统计模型驱动真实的、图形化应用程序运行; 利用自适应重放技术生成远程托管的应用, 客户端可以和远程应用对话; 利用微软 Detours 工具收集 Windows 桌面用户的使用数据, 用这些数据配置马尔可夫链参数; 用户与应用交互的模型使用层次马尔可夫链通过, 记录真实的人类用户生成的输入, 然后为每个应用程序的事件训练一个层次马尔可夫链模型. 在实验期间, 从马尔可夫链模拟生成与实际用户生成的事件流分布相似的新事件流, 利用这些相似的新事件流驱动测试台上的应用程序. 同时, MIT-LL 开发了 Koala 工具, 通过图像识别方式直接处理 VGA 输出, 然后通过 USB 端口注入键盘点击和鼠标移动, 具有较好的通用性. 这两种方式尽管能够创建多种网络流量, 但是模型针对复杂场景的预测能力比较受限, 导致难以评估用户达成正常目标的效果, 也无法预测用户环境适应性行为的自然变化. 针对这些问题, MIT-LL 在 2013 年开发了一种面向目标的场景建模机器人 (GOSMR), 采用经济学和行为决策研究工具仿真人类行为, 而不是传统的马尔可夫模型, 通过在虚拟机中创建了“机器影子”虚拟社区, 能够支持用户建模和执行复杂的实验场景. 在上述基础上 MIT-LL 将用户行为仿真改进为 CASCADE (cyber adversarial scenario modeler and artificial intelligence decision engine) 来实现, CASCADE 有两个智能引擎, 一个负责搜索决策空间寻找可能采取的动作, 另一个负责评估决策的有效性, 通过两者协同工作, 逼真的实现用户的仿真. 2014 年, 爱尔兰 Limerick 大学

Gleeson 等人[50]在 PNAS 上提出了在线网络信息共享的行为生成模型, 通过对社交关系网络中信息驱动的基本法机制进行仿真, 可以重现关于标签使用的特征 (如依赖时间的模因流行度的重尾分布). 该模型具有分析易处理性, 相对于其他模型, 可以区分用户的记忆时间和社交网络的连接结构对信息模因流行度的影响. 美国南加州大学 DeterLab 赛博实验平台[21,51−57]研发了 DASH (deter agents simulating human) 工具构建预测模型来仿真人类的网络行为. 模型使用了双处理 (dual-process), 即代理可以通过“本能”模式匹配或通过理性思考来选择下一个动作. DASH 还支持心理模型, 通过预测可能的行为及其影响来推动其理性行为, 支持用户在信息不充分或者压力及疲劳情况下作出的次优决策进行建模, 支持对用户行为为防守方、攻击方以及终端用户的行为进行建模, 为网络中人的行为实验提供真实且重复场景. DASH 支持人类邮件行为仿真, 直觉决策采取对外部世界的激活层次的声明+对激活层次调整的条件规则; 慎思推理采取自动规划选择合理的动作. 首先直觉推理, 其次确定是否触发慎思. 2018 年, 美国 TiER1 Performance Solutions 的 Warwick 等人[58]在研究中描述了模拟网络操作者的人的维度的工作, 一个受不同动机直接导致网络行为差异的用户, 最终导致遭受“驱动”恶意软件感染风险的差异. 2019 年, 清华大学 Wang 等人提出贝叶斯混合模型[59]以捕获何时, 何地以及使用什么应用程序, 并预测未来的应用程序使用情况. 在复杂的使用行为和非常有限的个人数据的情况下, 对应用程序使用的时空模式进行建模挑战比较大. 为了解决数据稀疏性的挑战, 使用了一个分层的 Dirichlet 过程来利用共享的时空模式来精确地建模数据不足的用户. 但是没有对位置上下文和用户兴趣进行进一步的研究, 所提出的模型在用户仿真程度上有所欠缺, 移动应用的使用的模型应该更加完善. 2019 年, 美国 CMU 软件工程研究所开发了 Ghost 系统[60], 能够自动调度虚拟用户活动和网络流, 其目的是通过建立和构建行为准确、自主的非玩家角色 (NPC), 在网络战演习中创建高水平的现实性, 但是 NPC 只对确定的行为进行反馈, 并表现出人类行为, 对于一些不确定性以及无把握的行为, NPC 是没有能力去应对的, 需要进一步的研究. 2020 年, 中国哈尔滨工业大学刘红日等人提出了面向用户群体的 Web 行为仿真方法[61], 能够较好拟合 Web 行为分布规律.

未来用户行为仿真需要采集大规模的真实用户行为数据, 建立用户行为的精准画像, 通过画像模型合成出更为逼真的用户操作序列, 支持群体的网络行为仿真场景需求.

4 网络行为仿真产品现状

本节介绍网络行为仿真的商业化产品和开源工具. 商业化产品集成大量的背景流或者前景流的行为模型, 支持完备的参数配置和可视化显示, 但是通常不支持二次开发或灵活部署. 开源工具通常集成少量的行为模型, 但是具有开放开发接口, 支持物理环境或虚拟环境的灵活部署.

4.1 商业产品

目前市场上的网络行为商业产品主要包括网络流量测试仪表系统和应用行为仿真仪表系统两种类型, 能够提供基本的背景流量、丰富的多领域应用模板库, 支持灵活的二次开发接口. 1) 网络流量测试仪表系统提供高速的流量引擎, 可以为靶场攻防演练环境或新技术验证实验环境的骨干网络注入大量的背景流量. 网络流量测试仪表系统包括测试回放模块、测试引擎模块、客户端软件模块、授权管理模块等. 测试回放模块主要提供真实流量回放、合成流量生成等, 支持多层网络协议的流量测试. 测试引擎模块主要提供高性能测试引擎、多引擎组合等. 客户端软件模块主要提供协议测试、任务管理、运行监控、用户管理等. 授权管理模块主要提供网络流量测试仪表使用测试授权管理. 目前主流的网络流量测试仪表系统厂家以美国为主, 如 IXIA、思博伦等厂家, 国内厂家如触点互动、信而泰公司开始进入网络行为仿真测试设备行业, 提供集成开发平台. 2) 应用行为仿真仪表系统提供网络应用回放模板库, 可以为靶场攻防演练环境或新技术验证实验环境定制大量的仿真场景配置和场景混合流量. 应用行为仿真仪表系统包括应用模板库模块、任务调度引擎模块、任务运行管理模块、授权管理模块等. 应用模板库模块主要提供应用模板管理、应用行为模板设计、应用模板升级、网络应用数据管理等. 任务调度引擎模块主要提供任务设计、任务调度、任务日志管理等. 任务运行管理模块主要提供应用仿真任务配置、任务运行监控、用户账户管理等. 授权管理模块主要提供应用行为仿真仪表使用测试授权管理. 目前主流的应用行为仿真仪表系

统网以美国为主, 如 IXIA、思博伦等厂家.

按照商业产品的市场定位, 可以将网络行为仿真相关的商业产品分为 3 类: (1) 综合产品; (2) 背景流产品; (3) 前景流产品. 商业产品以综合产品和前景流产品为主, 背景流产品以开源型商业产品为主. 图 3 给出了当前网络行为仿真相关主流商业产品的分类汇总, 基于每个种类的产品, 我们逐个给出调研分析结果. 最后, 我们对 3 类产品的优势和不足进行对比分析.

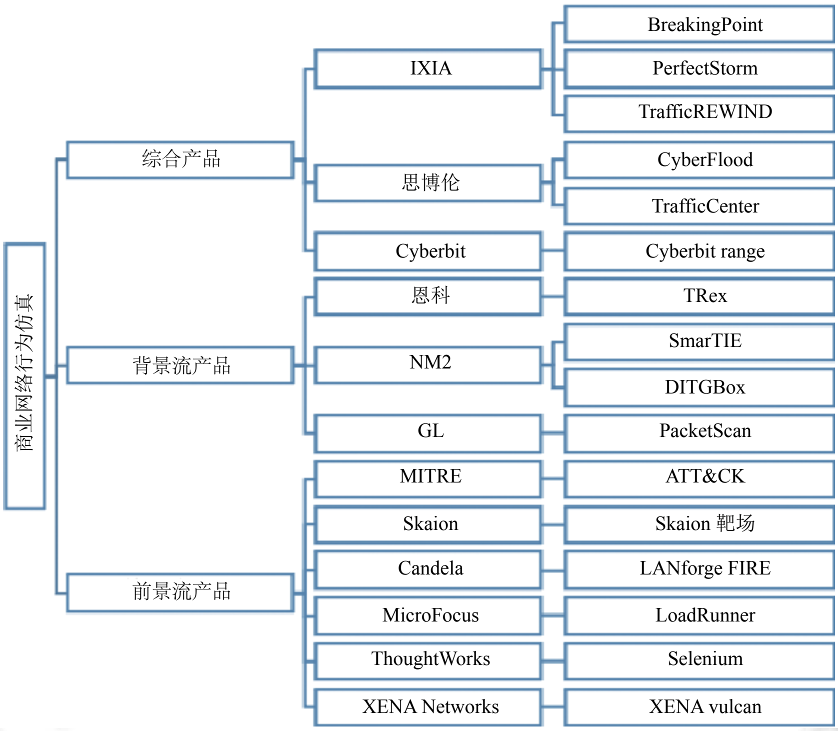

图 3　商业网络行为仿真产品分类

(1) 综合产品

美国的是德科技和思博伦公司、以色列的 Cyberbit 公司占据了网络行为仿真领域主要份额. 美国是德科技 (Keysight Technologies) 公司旗下 Ixia 公司的 BreakingPoint 产品[62]全方位重现真实网络条件, 支持马可夫模型、字典、动态文档生成以及多种语言能力. 其应用和用户沟通动作能加以编制. BreakingPoint 支持 OSI-7 层协议, 模拟 400 多个实际应用程序协议, 允许自定义和操作协议, 可仿真 4500 多个真实网络安全攻击和 28000 多个恶意软件攻击. 不足是无法与模拟的用户交互. 该公司 PerfectStorm[63]模拟数百万种终端用户的实际环境, 可测试聚合多元服务、应用业务交付和网络安全平台; TrafficREWIND[64]将生产网络洞察力转化为高度真实的测试流量配置. 所生成的测试配置可以用于 IXIA BreakingPoint 应用与安全测试解决方案. 通过 BreakingPointLiveAppSim 重建更具应用组合的生产流量; 然后借助 ATIP NETFlow 记录网络流量动态. 美国思博伦公司开发的 CyberFlood 仿真平台[65], 可生成真实的应用流量和攻击, 对当今各类应用感知网络基础设施的性能、扩展能力和安全性进行测试. 支持 15 000 多种互联网场景, PC 场景有 3412 种, 安卓有 4466 种, iOS 有 4790 种. 系统可以支持有状态的 200 万 IP 地址同时运行 HTTP 应用仿真. 支持多种应用按照指定比例混合, 内置企业, 高校, 运营商等典型网络模型. 可实现每秒 700 万个 HTTP 连接, 支持导入私有应用场景 (TCP/UDP). 此外, Spirent TrafficCenter[66]提供网络流量流配置所需的所有工具. 利用直观的测试管理仪表板为企业用户提供丰富的测试配置能力. 用户可以根据自己所需进行虚拟场景的仿真, 以软件即服务应用的方式提供, 可运行在任何常用的浏览器和操作系统上, 并提供其直观的测试管理仪表板. 以色列的 Cyberbit 公司开发的一个网络靶场 Cyberbit Range[67]提供的流量生成器能够模仿普

通企业网络的自然背景流量和“威胁流量”, 可生成电子邮件和 Web 浏览流量, 为学员提供检测、调查和响应网络攻击所需的工具. 随着云计算和计算机虚拟化技术的发展, 网络测试仪表领域涌现出以美国思博伦公司的 Cyberflood Virtual 系列、美国是德科技公司 IXIA 的 BreakPoint VE 系列为代表的虚拟化测试设备. 目前网络测试设备市场主要分为专用硬件型测试仪表和虚拟硬件型测试仪表, 专用硬件型测试仪表功能固化, 难以适配通用架构服务器、KVM 虚拟化环境, 虚拟硬件型测试仪表支持通用架构服务器、KVM 虚拟化环境, 满足灵活开发需求.

鹏城实验室的网络技术仿真验证平台课题组对接国家重大需求, 研究开发了关键指标达到世界先进水平的泛在网网络仿真平台, 在网络安全人才培养、攻防演练、安全技术评测和新技术验证领域成功获得应用, 产生重大的社会和经济效益, 提升我国网络安全水平. 在网络行为仿真方向, 课题组构建了大规模多源网络流模型模板库, 通过机器学习抽取互联网应用行为模板, 通过复杂业务场景仿真工作流模型, 实现了交互式背景流回放以及用户网络仿真, 通过业务场景合成仿真和编译技术快速再现背景流量、前景流量和靶标行为仿真, 通过分布并行调度技术有效实现了远景、中景、近景融合仿真.

(2) 背景流仿真产品

国外的背景流仿真产品主要包括思科公司的 TRex、意大利 NM2 公司的 SmarTIE 和 DITGBox、GL 公司的 PacketScan 思科公司的 TRex[68]是基于 Intel dpdk 开发的开源流量发生器, 提供有状态和无状态流量生成, 支持预处理和智能化回放实时流量模板; 支持用户自定义功能扩展; 流量测试规模可以达到 200 Gb/s. 通过命令行、Python 接口、YAML 配置文件方式下发配置和流量模板, 模拟真实、自定义的流量, 主要用于交换机、路由器压力测试、RFC2544 相关测试. 发包性能弱于思博伦等商业级流量测试设备, 产生的流量不能保持长连接. 意大利 NM2 公司提供 SmarTIE[69]和 DITGBox[70]监测大规模复杂网络状态, 结合主动和被动技术, 为网络基础设施及其上应用的使用、性能和安全情况提供一个唯一视图, 为大量应用和协议生成有状态和无状态流量集合, 为每个流提供性能评测, 包括 3–7 层网络 QoS 和 QoE 评测、RFC 兼容测试和网络设备压力测试. 单服务器的吞吐量最高可达 100 Gb/s. 支持数据平面开发套件 (DPDK), 缺少应用层流量场景模型支持, 应用交互仿真逼真度有待考证. GL 公司的 PacketScan[71]具备对 IP 网络上的实时信号和流量捕捉和监测能力, 可捕捉、分割、监控和收集 IP 网络上实时传输的各种类型的语音信号和流量, 几乎所有的 VoIP、无线协议和各类流量都可以被捕获并解码, 支持在双向实时传输协议 (RTP) 信道上的对 2000 个并发语音呼叫进行监控, 在双向 RTP 信道上处理 5000 个并发语音呼叫, 数据传输率从 1 Gb/s 到10 Gb/s.

国内的背景流仿真产品主要有北京触点互动信息技术有限公司的 XPRO、北京信而泰公司的 DarYu 产品. XPRO 产品单机支持数百万级新建和数千万级并发, 可模拟 L2-L7 DDoS 和应用漏洞攻击等. DarYu 3000 产品提供 OSI2-7 层协议高性能测试, 支持大规模路由和交换协议仿真和 HTTP 等应用测试, 最大支持 12 个 400GE 端口, 支持菊花链的形式进行多框级联, 便于扩展测试规模.

(3) 前景流仿真产品

前景流仿真产品市场主要被国外公司占据. 美国 MITRE 公司提出的 ATT & CK[72]是一个反应各个攻击生命周期攻击行为知识库和模型, 主要应用于评估攻防能力覆盖、APT 情报分析、威胁狩猎及攻击模拟等领域. 该框架包含攻击者渗透网络、入侵主机、提升权限、隐秘移动和渗漏数据的攻击链. ATT & CK 框架的应用比较少. 美国 Skaion 公司[73]的 Skaino 网络靶场提供大规模前景网络靶场测试场景和对抗模型来测试受测系统, 提供桌面级应用程序仿真和大型网络的仿真, 可生成达到 100 Gb/s 实时有状态流量. 恶意流量的仿真包括蠕虫、基础攻击、电子邮件钓鱼、扫描及模拟真实网络场景进行红蓝对抗测试. Candela 技术公司的 LANforge FIRE[74]可生成和接收 2–7 层网络协议 (包括无线协议), 用于在被测试网络上创建流量负载, 同时报告各类网络统计特征, 支持真实协议和有状态 TCP 连接, 可针对 Web 服务器、VOIP 网关、防火墙、负载均衡器等各类网络组件生成流量负载并支持 IPv4 和 IPv6. 支持十几种网络协议. MicroFocus 公司的 LoadRunner[75]是一种预测系统行为和性能的负载测试工具, 通过模拟实际用户的动作行为, 测试服务性能与稳定性, 支持记录用户活动, 并生成测试脚本, 模拟用户行为产生真实流量, 完成软件性能测试, 支持的应用程序环境和协议性能测试. ThoughtWorks 公司的 Selenium 产品[76]采用 Javascript 来管理整个测试过程, 包括读入测试套件、执行测试和记录测试结果. 通过模拟真实用户操作, 包

括浏览页面、点击链接、输入文字、提交表单、触发鼠标事件等, 用来对 Web 应用程序进行自动化测试, 主要用于 Web 前端测试. Selenium 录制回放方式的稳定性和可靠性有限; 对于复杂的页面逻辑其处理能力有限. XENA Vulcan[77]是 Xena Networks 开发的商业流量生成器和分析仪, 使用预定义和可自定义的混合模板设置各种应用程序的大规模实际流量, 支持生成有状态的 4–7 层以太网流量, 综合分析防火墙, 交换机, 路由器, NAT 路由器, 代理, 负载均衡器, 带宽整形器等在各种实际场景中的表现. 高性能 VulcanBay 支持测试最高 40GE 的网络吞吐率, 最高 2 400 万并发连接数, 以及最高 700 万并发 HTTP 事务数.

国外厂商仿真全应用层业务流量的同时支持安全攻击仿真测试, 病毒攻击仿真测试, 僵尸网络, 逃避技术以及 APT 等复杂测试场景安全测试. 而目前国内测试仪表生产厂家主要通过用户抓包回放实现网络行为仿真, 缺乏足够的应用模板库和网络安全样本库, 通常仅支持 HTTP, FTP, SIP, DNS 等少量应用协议测试以及简单的攻击测试, 应用层业务协议覆盖度较低, 攻击种类非常简单, 不支持复杂测试场景安全测试、工控协议测试及工控攻击攻击测试.

综合上述产品分析, 商业产品在背景流仿真方面具有完备的技术体系支撑, 在前景流仿真的技术能力和规模上存在不足, 特别是难以支持面向群体的前景流仿真.

4.2 开源产品

按照开源工具的功能分类定位, 可以将网络行为仿真的开源工具分为背景流仿真工具和前景流仿真工具. 开源工具以背景流仿真为主, 前景流仿真以特定协议测试为主. 图 4 给出了开源工具分类梳理.

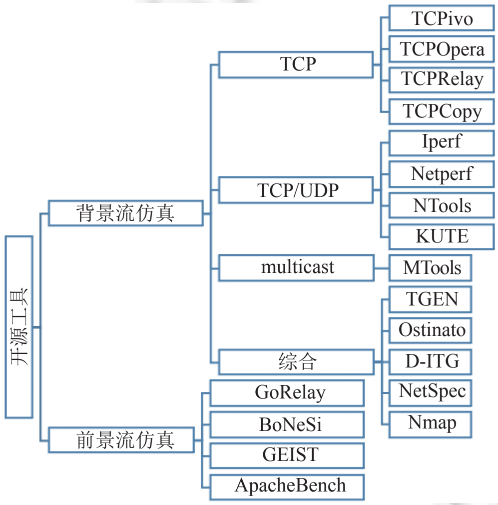

图 4 开源工具分类

(1) 背景流仿真产品

TCP 协议是最为常见的传输协议, 因此多种 TCPivo、TCPOpera、TCPRelay、TCPCopy 开源工具专门支持 TCP 协议的生成. TCPivo 开源工具[78]支持在单个主机上重放 PCAP 文件, 针对高速日志利用预取技术保持时间戳精度, 在报文复杂可忽略时, 支持将报文载荷替换为空以提升重放速率. TCPOpera 开源工具[79]从单个报文日志文件能够重新创建 TCP 序列和会话层事件 (TCP 握手和 Web 对象流量). TCPOpera 从报文日志抽取事件的统计模型, 然后基于模型生成合成的网络流事件, 能够复制网络事件的分布, 而不是保持数据报文的时间戳精度. TCPOpera 利用独立的线程管理不同的网络流, 对流间的顺序并不考虑. TCPOpera 可以利用多个结点来模拟一组客户端与真实服务器的交互过程, 但是使用单一的控制结点和带外控制消息生成并发网络流. TCPRelay[80]是在 GNUv3 下开源的工具套件, 主要用于在 UNIX 或安装 Cygwin 的 WIN32 系统下编辑和重放抓取到的流量包, 将流量按客户端或服务器分类, 支持对 2–4 层流量包的重写, 最终将流量重放到被测网络或网络设备 (如交换机、路由器、防火墙、IDS、IPS 等), 支持单网卡和双网卡两种模式. TCPReplay 能重放捕获的流量, 允许修改 IP 地址、

MAC 地址, 并选择性重放部分报文. TCPRelay 的目的是对网络设备进行压力测试, 并不能够连接服务. 该软件不足主要体现在不支持全协议栈测试, 缺少对应用层协议的支持. TCPCopy[81]是网易公司开发的一种基于 TCP 的包请求复制工具, 可以把在线流量导入到测试系统中去. 曾经应用于网易的广告投放系统, URS 系统, Nginx hmux 协议等系统, 避免了上线带来的很多问题. 现在此工具已经广泛应用于各大互联网公司. TCPCopy 采用了 C/S 架构, 其中客户端运行在在线服务器上面, 用来捕获在线请求数据包; 服务端 (监听端口为 36524) 运行在测试机器上面, 在测试服务器的响应包丢弃之前截获测试服务器的响应包, 并通过客户端和服务端之间的 TCP 连接传递响应包的 TCP 和 IP 头部信息给客户端, 以完成 TCP 交互.

针对 TCP 和 UDP 协议的开源仿真工具发展早, 目前已经较为成熟, 涌现出 Iperf、Netperf、NTools、KUTE 等知名仿真工具. Iperf[82]是一个通过网络合成流量对网络性能测试的工具. Iperf 可以测试最大 TCP 和 UDP 带宽性能, 具有多种参数和 UDP 特性, 可以根据需要调整, 可以报告带宽、延迟抖动和数据包丢失. Iperf 采用 C/S 软件架构, 支持调整时间、缓冲、协议配置参数, 支持零拷贝. Netperf[83]是一种网络性能的测量工具, 主要针对基于 TCP 或 UDP 的传输. Netperf 支持批量数据传输 (bulk data transfer) 模式和请求/应答 (request/response) 模式. Netperf 测试结果所反映的是一个系统能够以多快的速度向另外一个系统发送数据, 以及另外一个系统能够以多快的速度接收数据. Netperf 工具以 client/server 方式工作. 首先建立一个控制连接, 传递有关测试配置的信息, 以及测试的结果; 在控制连接建立后再建立一个测试连接, 用来传递特殊的流量模式, 以测试网络的性能. NTools[84]是一个 Linux 平台下用于流量生成、分析和网络仿真的工具包, 可用于仿真各种丢包、延迟和抖动等网络损伤特性. 支持任意数量的 UDP 和 TCP 流; 以大数据帧方式在千兆以太网接口卡仿真; 支持配置端口、IP 地址、包尺寸和速率和 VLAN 协议、文件大小分布,评估包丢失、乱序、延迟、抖动等特性, 支持在动态 TCP 流上对下载时间和吞吐量进行评估; 支持网路损伤特性仿真. KUTE[85]是一个高性能的 Linux 内核级的 UDP 流量引擎. 在应用性能测试领域, 包生成速率是响应性能的重要因素, 已有的流量生成器多在用户空间运行, 由于上下文切换带来了效率损失, 尤其针对高速千兆以太网环境. KUTE 采用内核实现模式, 在发送端和接收端都可以更精准地刻画包传输的时间特性. 该软件采用 C/S 架构模式, 评测指标采用包间隔到达时间和包发送速率.

综合型背景流如 Ostinato、TGEN、D-ITG、NetSpec、NPing 综合了常见的传输层和网络层的协议生成模式, 具有较好的通用性, 能满足多种网络测试需求. Ostinato[86]是一个开源跨平台网络包生成器和分析工具, 通过直观的图形化界面合成 L1–L5 层常见标准协议, 支持速率定制, 并发流发送. 主控点通过远程过程调用控制软件平台的执行. 支持随机参数模型. TGEN[87]是一个基于 C 语言开发的流量生成软件, 用户可以自主配置流量生成参数 (规模、时序、并发度等), 支持复杂的流量生成模式. 用户通过编写相对简单的 Python 脚本生成 graphml 文件, 作为 TGEN 的配置参数文件, 用于指导流量生成. 技术上利用行动依赖图构建复杂的行为模拟, 支持常见网络协议, 支持常见用户动作定义. 开源网络模拟器 Shadow 用 TGEN 进行网络流量模拟, 开源洋葱路由性能监控工具 OnionPerf 用 TGEN 监控开源洋葱路由网络 Tor 的性能. D-ITG[88,89]可在数据包一级精准刻画流量的时序和规模特征, 支持 IPv4 和 IPv6 流量, 生成网络层、传输层和应用层流量. 支持多种操作系统. 软件的控制过程为主控点通过 TSP (traffic specification protocol) 通道集中远程控制, 或者其他流量发送/接收端通过 TSP 通道分布远程控制. 数据传输过程为网络流发送端用独立的线程并发向不同的接收端发送, 网络流接收端用独立的线程接收并发的网络流. NetSpec [90]由美国堪萨斯州立大学学者研发, 用于 ACTS ATM Internetwork (AAI) 项目. 采用脚本语言支持用户定义多种流量类型. NetSpec 支持多操作系统平台, 支持多测试并发, 支持主动、被动两种运行模式, 假设网络状态不变的情况下, 可重现网络特性, 支持按需灵活添加新组建. 控制过程通过命令行脚本配置. 数据传输过程为发送用户自定义配置生成的报文. 压力测试案例中支持自定义的 TCP, UDP, ICMP 和 ARP 报文生成, 对 ARP 欺骗、dos 攻击、路由追踪、防火墙进行了测试. Nmap 团队开发的 NPing[91]可作为生成网络包、响应分析和测量响应时间的工具, 主要功能包括支持自定义的 TCP, UDP, ICMP 和 ARP 报文生成; 支持多目标主机与多目标端口; 支持以太网帧生成、支持 IPv6 协议. 该方法可作用于多平台, 支持以太网、TCP、UDP、ICMP、ARP、IPv6 等协议; 可用于安全审查, 防火墙测试, 小规模攻击测试、追踪探测等场景. 拥有独立的客户端/服务端程序, 通过命令行脚本配置实现发送用户自定义配置生成的报文.

组播协议仿真工具较少见. MTools [92]是一个通过测量评估网络数据报文的单项时延和往返时延数据来分析网络性能的工具. 用于基于 UDP 协议的组播测试, 运行于 Linux 环境, 采用分布式软件架构, 通过命令行参数配置软件的运行, 数据传输方式为流量生成节点加入组播组, 然后发送 UDP 报文. 评价指标为消息速率.

(2) 前景流仿真产品

多款开源工具支持 Web 协议的前景流仿真. GEIST 是 Intel 公司[93]发布的 Web 服务器压力测试工具, 运行与 Linux 平台, 支持 Web 相关协议, 采用自动构建真实数据的参数模型. 支持的数据模型包括流量时间分布、用户关联分布、网络动态分布、流量非稳态分布及事务组合分布. 评测指标为吞吐率、响应时间、间隔到达时间. ApacheBench[94]是一个针对 Apache HTTP 服务器进行压力测试的工具, 用于测试 Web 服务器性能, 即服务器每年能处理的请求数. 采用命令行参数配置和控制程序运行. 通过创建很多的并发访问线程, 模拟多个访问者同时对某一 URL 地址进行访问. 采用并发数据模型, 评测指标为页面载入时间. GoReplay[95]是一个开源的网络监控工具, 可用于采集指定的现网流量, 采集的流量可用于流量镜像、负载测试或详细分析监控. 软件是用 Golang 语言编写, 支持 http 实时流复制, 支持流量按需放大, 缩小, 频率限制. 专业版提供云端存储和重放流量, 提升重放精度, 支持 TCP 会话级跟踪和重放, 支持二进制协议, 免费开源社区版本仅支持 HTTP 协议, 专业版额外提供了 Thrift 和 ProtocolBuffers 等云服务协议. 网络安全协议的开源仿真得到关注. BoNeSi[96]是一个 DDoS 僵尸网络模拟器, 在有线测试床环境下模拟生成僵尸网络流量. 可生成符合 ICMP、UDP 和 TCP (HTTP) 协议的洪水攻击流量, 参数可高度灵活配置. 性能方面, 在一台 AMD Opteron with 2 GHz 的机器下每秒生成的数据包可达 150 000 个.

综上所述, 开源产品在 TCP、UDP 和 HTTP 协议的背景流仿真方面具有成熟的技术体系, 但是在应用层协议方面缺乏数据模型的技术支持, 在前景流仿真的技术能力和规模存在不足, 同样难以支持面向群体的用户行为仿真.

5 网络行为仿真发展展望

在当前的国内外研究成果基础上, 未来有望如下方面取得突破进展.

(1) 前景流仿真

在以场景为牵引的攻防演练中, 前景流仿真作为构建真实场景的有效手段, 能够有效解决靶场场景仿真中“仿不真”的问题. 国际上大多数的赛博实验设施目前都缺乏对用户行为的仿真建模. 前景流仿真集中在单个行为模型, 无法模拟大规模用户行为中广泛存在的人类动态行为. 未来需要在实验中集成真实和合成的人类行为, SRI 和南加州大学科学家领衔发布的 2015 年未来赛博实验报告[97]提出.

1) 增强合成行为: 对于 IT 领域, 应该增强人工合成活动生成器, 以支持所生成活动的行为边界规范, 从而支持可重复性. 支持复杂的网络环境以及系统参与者 (如正常用户、攻击者和管理员), 逐步构建智能化交互式网络应用行为, 通过较高智能化的用户行为模型, 支持马尔可夫模型、知识图谱、深度学习、强化学习、图神经网络, 支持数据预训练和在线学习, 支持用户行为随环境变化而自适应调整.

2) 自动提取特征: 应该努力专注于提供能够自动提取其他域的特征和创建模型的网络, 可能会被注入到一个测试环境和实例化的现实世界.

3) 提供行为转换模型: 应该开发工具来支持将人纳入科学有效的实验. 需要使用这些工具来捕获和测量人员的性能以及他们与系统的交互. 应该提供工具来帮助研究人员将捕获的人类活动转换为人类行为模型, 用于后续的模拟.

4) 支持复杂交互: 实验能力必须支持极端规模的实验, 使用经过验证的仿真和仿真工件的混合. 系统的组合将在系统之间创建复杂的交互行为, 因此有必要使用方法来推断这些交互

(2) 背景流仿真

传统的网络流量模拟方法通常采取统计模型合成报文或者回放互联网采集数据, 这些方法具有计算开销低、易于部署测试的优势, 但是由于对互联网行为模型存在简化假设, 缺乏对复杂网络场景的仿真建模手段, 难以全面再现远景、中景和近景的互联网行为特征. 背景流量仿真工具通常面向网络设备测试的单臂或者双臂测试, 并不适应复杂网络结构的网络靶场环境. 未来背景流仿真需要复现互联网环境下的大规模网络应用协议流量特征[92].

1) 多级流量模型: 真实流量文件逼真度高, 但是存储和回放成本高昂, 并且难以动态调整, 而解析流量模型灵活性好, 但逼真度有限. 未来需要研究流量模型的组合机制, 灵活支持不同等级逼真度的流量模板组合, 满足不同仿真对象和建模层级对逼真度要求的差异性.

2) 大规模背景流调度: 面向大规模虚拟机或者虚拟容器网络仿真平台环境, 构建大规模虚拟机背景流量编排语言和编排框架, 生成不同时间长度的背景流量规划, 通过编排框架调度大规模虚拟机集合按需生成复杂的背景流注入事件, 支持大规模高并发的网络流量仿真.

3) 流量模板抽取: 目前网络行为仿真平台的网络流量模型依托人工手段外包获取, 构建成本高昂, 缺乏对新型模型模板的按需抽取能力. 未来需要研究基于人工智能的流量分类和应用识别方法, 将未知的应用协议流量自动化或半自动化转换为确定类型的应用流量模板, 不断提升互联网应用的覆盖率.

(3) 运行框架

目前国内外的网络技术仿真验证平台主要采取手工脚本配置仿真业务过程, 业务仿真过程描述能力低, 仿真过程配置时间长, 与物理平台环境绑定, 业务仿真过程的透明性、可移植性低., 难以综合仿真远景、中景和近景的互联网行为特征.

1) 复杂场景合成框架: 目前新型的运行框架通过工作流语言提升了场景的表达能力, 但是随着工作流增大, 如何提升随机化参数合成的工作流场景和互联网场景的近似程度成为新的问题.

2) 网络仿真组网框架: 目前运行框架需要手工配置大量物理机、虚拟机网络环境参数, 面向大规模多租户的虚拟机或容器环境, 未来需要研究新型的分布并行编排框架, 实现架构、功能、流程的灵活组合与配置, 实现自动化的场景部署、运行、还原步骤管理, 降低仿真的手工配置成本, 提升对大规模互联网场景的复现能力.

3) 智能化运行框架: 目前运行框架面向纯净的仿真环境, 缺少网络路由异常或失效事件, 未来需要构建更为智能化的数字孪生式运行框架, 实时感知任务关键网络行为仿真的关联关系、系统运行状态、环境变化, 建模强对抗、不确定的网络行为, 向仿真环境注入各类网络异常或失效行为, 复现动态化的运行环境.

References:

[1] China Netcom. Full text of “National cyberspace security strategy”. 2016 (in Chinese). http://www.cac.gov.cn/2016-12/27/c_1120195926.htm

[2] Fang BX, Jia Y, Li AP, Zhang WZ. Cyber Ranges: State-of-the-art and research challenges. Journal of Cyber Security, 2016, 1(3): 1–9 [doi: 10.19363/j.cnki.cn10-1380/tn.2016.03.001]

[3] Group standard of China cyberspace emerging technology Security Innovation Forum. 2021 (in Chinese). https://www.ccsia.com.cn/

[4] Damodaran SK, Couretas JM. Cyber modeling & simulation for cyber-range events. In: Proc. of the Conf. on Summer Computer Simulation. San Diego: Society for Computer Simulation International, 2015. 1–8.

[5] Erazo MA, Liu J. Leveraging symbiotic relationship between simulation and emulation for scalable network experimentation. In: Proc. of the 1st ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation. New York: ACM, 2013. 79–90. [doi: 10.1145/2486092.2486103]

[6] Erazo MA, Rong R, Liu J. Symbiotic network simulation and emulation. ACM Trans. on Modeling and Computer Simulation, 2015, 26(1): 1–25. [doi: 10.1145/2717308]

[7] Spirent Communications, sprient. Operational impact of cyber range elements, simulations and realism white paper. 2014. https://www.spirent.com/~/media/white%20papers/broadband/pab/cyber_range_whitepaper

[8] Schatzmann J, Young S. The hidden agenda user simulation model. IEEE Trans. on Audio, Speech, and Language Processing, 2009, 17(4): 733–747. [doi: 10.1109/TASL.2008.2012071]

[9] Sanaga P, Duerig J, Ricci R, Lepreau J. Modeling and emulation of internet paths. In: Proc. of the 6th USENIX Symp. on Networked Systems Design and Implementation. Boston: USENIX Association, 2009. 199–212.

[10] Miyachi T, Miwa S, Shinoda Y. XBurner: A XENebula-based native traffic-generation platform. In: Proc. of the Int’l Conf. on Testbeds and Research Infrastructures. Berlin: Springe, 2010. 629–631. [doi: 10.1007/978-3-642-17851-1_62]

[11] Van Vorst N, Erazo MA, Liu J. PrimoGENI: Integrating real-time network simulation and emulation in GENI. In: Proc. of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation. Nice: IEEE 2011. 1–9. [doi: 10.1109/PADS.2011.5936747]

[12] Cao LJ, Bu XY, Fahmy S, Cao SY. Towards high fidelity network emulation. In: Proc. of the 26th Int’l Conf. Computer Communication

and Networks (ICCCN). Vancouver: IEEE, 2017. 1–11. [doi: 10.1109/ICCCN.2017.8038453]
[13] Wu XL, Yang Q, Liu X, Jin D, Lee CW. A hardware-in-the-loop emulation testbed for high fidelity and reproducible network experiments. In: Proc. of the 2017 Winter Simulation Conf. Las Vegas: IEEE, 2017. 408–418. [doi: 10.1109/WSC.2017.8247803]
[14] Pastor A, Mozo A, López DR, Folgueira J, Kapodistria A. The Mouseworld, a security traffic analysis lab based on NFV/SDN. In: Proc. of the 13th Int'l Conf. on Availability, Reliability and Security. New York: ACM, 2018. 1–6. [doi: 10.1145/3230833.3233283]
[15] Laboratory UNR. Networks and communication systems branch. https://github.com/USNavalResearchLaboratory/mgen
[16] Ahrenholz J. Comparison of CORE network emulation platforms. In: Proc. of the 2010-Milcom 2010 Military Communications Conf. San Jose: IEEE, 2010. 166–171. [doi: 10.1109/MILCOM.2010.5680218]
[17] Braje TM. Advanced tools for cyber ranges. Lincoln Laboratory Journal, 2016, 22(1): 24–32.
[18] Detken KO, Oberle A, Kuntze N, Eren E. Simulation environment for mobile virtualized security appliances. In: Proc. of the 1st IEEE Int'l Symp. on Wireless Systems (IDAACS-SWS). Offenburg: IEEE, 2012. 113–118. [doi: 10.1109/IDAACS-SWS.2012.6377646].
[19] Câmara D, Tazaki H, Mancini E, Turletti T, Dabbous W, Lacage M. DCE: Test the real code of your protocols and applications over simulated networks. IEEE Communications Magazine, 2014, 52(3): 104–110. [doi: 10.1109/MCOM.2014.6766093]
[20] Momeni B, Kharrazi M. Partov: A network simulation and emulation tool. Journal of Simulation, 2016, 10(4): 237–250. [doi: 10.1057/jos.2014.22]
[21] Mirkovic J, Bartlett G, Blythe J. DEW: Distributed experiment workflows. In: Proc. of the11th USENIX Workshop on Cyber Security Experimentation and Test. Baltimore, 2018.
[22] Wang S, Zhu F, Yao YP, Tang WJ, Xiao YH. Container-based automatic packaging technology for complex system simulation application. Journal of System Simulation, 2020, 32(11): 2199–2207 (in Chinese with English abstract). [doi: 10.16182/j.issn1004731x.joss.20-fz0423]
[23] Liu B, Wang BS, Deng WP. Containerized workflow framework supporting elastic scaling in cloud environment. Computer Engineering, 2019, 45(3): 7–13 (in Chinese with English abstract). [doi: 10.19678/j.issn.1000-3428.0049811]
[24] Ji YF, Zhang GM, Xing CY. Research on local deployment and scheduling problem of user behavior simulation. Information Techology and Network Security, 2020, 39(1): 78–82 (in Chinese with English abstract). [doi: 10.19358/j.issn.2096-5133.2020.01.015]
[25] Faber T. DETER testbed for security experimentation. ACSAC keynote speech, 2007.
[26] Haglich P, Grimshaw R, Wilder S, Nodine M, Lyles B. Cyber scientific test language. In: Proc. of the Int'l Semantic Web Conf. Bonn: Springer, 2011. 97–111. [doi: 10.1007/978-3-642-25093-4_7]
[27] Tanasache FD, Sorella M, Bonomi S, Rapone R, Meacci D. Building an emulation environment for cyber security analyses of complex networked systems. In: Proc. of the 20th Int'l Conf. on Distributed Computing and Networking. New York: ACM, 2019. 203–212. [doi: 10.1145/3288599.3288618]
[28] Cheng Y C, Hölzle U, Cardwell N, Savage S, Voelker GM. Monkey see, monkey do: A tool for TCP tracing and replaying. In: Proc. of the Annual Conf. on USENIX Annual Technical Conf. Boston: USENIX Association, 2004. 7.
[29] Netravali R, Sivaraman A, Das S, Goyal A, Winstein K, Mickens J, Balakrishnan H. Mahimahi: Accurate record-and-replay for HTTP. In: Proc. of the 2015 USENIX Annual Technical Conf. Santa Clara: USENIX Association, 2015. 417–429.
[30] Durumeric Z, Adrian D, Mirian A, Bailey M, Halderman JA. A search engine backed by internet-wide scanning. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. New York: ACM, 2015. 542–553. [doi: 10.1145/2810103.2813703]
[31] Huang N, Liu Y, Wang XF. User behavior emulation technology based on interactive traffic replay. Computer Engineering, 2021, 47(10): 103–110 (in Chinese with English abstract).
[32] Sommers J. jsommers/harpoon. Github. https://github.com/jsommers/harpoon
[33] Weigle MC, Adurthi P, Hernández-Campos F, Jeffay K, Smith FD. Tmix: A tool for generating realistic TCP application workloads in NS-2. ACM SIGCOMM Computer Communication Review, 2006, 36(3): 65–76. [doi: 10.1145/1140086.1140094]
[34] Vishwanath KV, Vahdat A. Realistic and responsive network traffic generation. In: Proc. of the 2006 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM, 2006. 111–122. [doi: 10.1145/1159913.1159928]
[35] Khayari R E A, Rucker M, Lehmann A, Musovic A. ParaSynTG: A parameterized synthetic trace generator for representation of WWW traffic. In: Proc. of the 2008 Int'l Symp. on Performance Evaluation of Computer and Telecommunication Systems. Edinburgh: IEEE, 2008. 317–323.
[36] Kolesnikov A. UniLoG: A unified load generation tool. In: Int'l GI/ITG Conf. on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. Kaiserslautern: Springer, 2012. 253–257. [doi: 10.1007/978-3-642-28540-0_21]
[37] Chu WB, Guan XH, Cai ZM, Tao J. System and method for real-time volume control in reproducing network scenario. Chinese Journal of Computers, 2012, 35(7): 1485–1497 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2012.01485]

[38] Li T, Liu J. Cluster-based spatiotemporal background traffic generation for network simulation. ACM Trans. on Modeling and Computer Simulation, 2015, 25(1): 1–25. [doi: 10.1145/2667222]

[39] Emmerich P, Gallenmüller S, Raumer D, Wohlfart F, Carle G. MoonGen: A scriptable high-speed packet generator. In: Proc. of the 2015 Internet Measurement Conf. New York: ACM, 2015. 275–287. [doi: 10.1145/2815675.2815692]

[40] Tang JQ, Wang DB, Zou XX, Dong L, Yan JH, Zhang X, Zhi H, Zhang JH, Wu KX, Jin HL. Network flow generation method based on data flow features. CN, 105049277A. 2015-11-11 (in Chinese).

[41] Fortner S, Xie GG. DSSR: Balancing semantics and speed requirements in packet trace replay. In: Proc. of the 2017 IEEE Int'l Conf. on Communications. Paris: IEEE, 2017.1–6. [doi: 10.1109/ICC.2017.7997257]

[42] Tune P, Roughan M. Controlled synthesis of traffic matrices. IEEE/ACM Trans. on Networking, 2017, 25(3): 1582–1592. [doi: 10.1109/TNET.2016.2639066]

[43] Javali C, Revadigar G. Network Web traffic generator for cyber range exercises. In: Proc. of the 44th IEEE Conf. on Local Computer Networks. Osnabrueck: IEEE, 2019. 308–315. [doi: 10.1109/LCN44214.2019.8990880]

[44] Bhatia S, Schmidt DA, Mohay GM, Tickle A. A framework for generating realistic traffic for distributed denial-of-service attacks and flash events. Computers & Security, 2014, 40: 95–107. [doi: 10.1016/j.cose.2013.11.005]

[45] Rossey LM, Cunningham RK, Fried DJ, Rabek JC, Lippmann RP, Haines JW, Zissman MA. LARIAT: Lincoln adaptable real-time information assurance testbed. In: Proc. of the IEEE Aerospace Conf. Big Sky: IEEE, 2002. 6. [doi: 10.1109/AERO.2002.1036158].

[46] Pham C, Tang D, Chinen K I, Beuran R. CyRIS: A cyber range instantiation system for facilitating security training. In: Proc. of the 7th Symp. on Information and Communication Technology. New York: ACM, 2016. 251–258. [doi: 10.1145/3011077.3011087].

[47] Dobson GB, Podnar TG, Cerini AD, Osterritter LJ. Geoffrey L JO Dobson B. R-EACTR: A framework for designingrealistic cyber warfare exercises. 2017. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2017_005_001_505226.pdf

[48] Stayton E, Roque A. Data-driven multi-agent email generators. In: Proc. of the IEEE Int'l Symp. Technologies for Homeland Security (HST). Woburn: IEEE, 2018. 1–7. [doi: 10.1109/THS.2018.8574128]

[49] Babu V, Kumar R, Nguyen HH, Nicol DM, Palani K, Reed E. Melody: Synthesized datasets for evaluating intrusion detection systems for the smart grid. In: Proc. of the 2017 Winter Simulation Conf. Las Vegas: IEEE, 2017. 1061–1072. [doi: 10.1109/WSC.2017.8247855]

[50] Gleeson JP, Cellai D, Onnela JP, Porter MA, Reed-Tsochas F. A simple generative model of collective online behaviour. Proc. of the National Academy of Sciences of the United States of America, 2014, 111(29): 10411–10415. [doi: 10.1073/pnas.1313895111]

[51] Mirkovic J, Benzel TV, Faber T, Braden R, Wroclawski JT, Schwab S. The DETER project: Advancing the science of cyber security experimentation and test. In: Proc. of the 2010 IEEE Int'l Conf. Technologies for Homeland Security (HST). Waltham: IEEE, 2010. 1–7. [doi: 10.1109/THS.2010.5655108]

[52] Mirkovic J, Fahmy S, Reiher P, Thomas RK. How to test DoS defenses. In: Proc. of the Cybersecurity Applications & Technology Conf. for Homeland Security. Washington, DC: IEEE, 2009. 103–117. [doi: 10.1109/CATCH.2009.23]

[53] Mirkovic J, Benzel T. Teaching cybersecurity with deterLab. IEEE Security & Privacy, 2012, 10(1): 73–76. [doi: 10.1109/MSP.2012.23]

[54] Benzel T. The science of cyber security experimentation: The DETER project. In: Proc. of the 27th Annual Computer Security Applications Conf. New York: ACM, 2011. 137–148. [doi: 10.1145/2076732.2076752]

[55] Benzel T, Braden B, Faber T, Mirkovic J, Schwab S, Sollins K, Wroclawski J. Current developments in DETER cybersecurity testbed technology. In: Proc. of the Cybersecurity Applications & Technology Conf. for Homeland Security. Washington: IEEE, 2009. 57–70. [doi: 10.1109/CATCH.2009.30]

[56] Wei SJ, Mirkovic J. A realistic simulation of internet-scale events. In: Proc. of the 1st Int 'l Conf. on Performance Evaluation Methodolgies and Tools. New York: ACM, 2006. 28–40. [doi: 10.1145/1190095.1190131]

[57] Schwab S, Wilson B, Ko C, Hussain A. SEER: A security experimentation EnviRonment for DETER. In: Proc. of the DETER Community Workshop on Cyber Security Experimentation and Test on DETER Community Workshop on Cyber Security Experimentation and Test 2007. Boston: USENIX Association, 2007. 2.

[58] Warwick W, Buchler N, Marusich L. An integrated model of human cyber behavior. In: Proc. of the Int 'l Conf. on Applied Human Factors and Ergonomics. Orlando: Springer, 2018. 290–302. [doi: 10.1007/978-3-319-94223-0_28]

[59] Wang HD, Li Y, Zeng SH, Wang G, Zhang PY, Hui P, Jin DP. Modeling spatio-temporal app usage for a large user population. In: Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. New York: ACM, 2019. 1–23. [doi: 10.1145/3314414]

[60] Rob CK Patro. GHOSTS. 2020. https://github.com/cmu-sei/GHOSTS

[61] Liu HR, Zhang X, Li JJ, Wang BL. A novel collective user web behavior simulation method. Computers, Materials & Continua, 2021, 66 (3): 2539–2553. [doi: 10.32604/cmc.2021.012213]

[62] Ixia, A Keysight Business. Breakingpoint systems is now part of Ixia. Ixia, A Keysight Business. https://www.ixiacom.com/breakingpoint-

systems
[63] Ixia, A Keysight Business. PerfectStorm. Industry's highest performing application and security test platform. https://www.ixiacom.com/zh/products/perfectstorm
[64] Ixia, A Keysight Business. TrafficREWIND. https://www.ixiacom.com/products/trafficrewind
[65] Spirent Communications. Security and performance testing for app-aware solutions. https://www.spirent.com/products/security-and-applications-performance-testing-cyberflood
[66] Spirent Communications. Network testing made easy Spirent TrafficCenter. https://www.spirent.com/products/temeva/trafficcenter
[67] Cyberbit. Cyber range. Cyberbit. https://www.cyberbit.com/solutions/cyber-range/
[68] Cisco. TRex realistic traffic generator. https://trex-tgn.cisco.com/
[69] NM2. SmarTIE overview. https://www.nm-2.com/smartie/
[70] NM2. DITGBox overview. https://www.nm-2.com/ditgbox/
[71] GL COMMUNICATIONS INC. PacketScan™-All-IP analyzer. https://www.gl.com/packetscan-all-ip-packet-analyzer.html
[72] The MITRE Corporation. ATT&CK matrix for enterprise. The MITRE Corporation. https://attack.mitre.org/
[73] Skaion. Core competencies. Skaion. http://www.skaion.com/
[74] Candela Technologies. LANforge-FIRE stateful network traffic generator. http://www.candelatech.com/datasheet_fire.php?gclid=EAIaIQobChMI6a6m1oLa5wIVE66WCh3sIAOfEAAYASAAEgJuTvD_BwE
[75] Micro Focus. LoadRunner professional. https://www.microfocus.com/en-us/products/loadrunner-professional/overview
[76] "Selenium Projects. https://www.selenium.dev/projects/
[77] Xena Networks. Vulcan stateful ethernet traffic generation and analysis. https://xenanetworks.com/vulcan/
[78] TCPivo: A high-performance packet replay engine. https://www.eecg.utoronto.ca/~ashvin/publications/TCPivo_SIGCOMM_MoMeTools_2003.pdf
[79] Hong SS, Wu SF. On interactive internet traffic replay. International Workshop on Recent Advances in Intrusion Detection. Springer, 2005. 247–264.
[80] GitHub. Tcpreplay. https://github.com/appneta/tcpreplay
[81] GitHub. TCPCopy-A TCP stream replay tool. https://github.com/session-replay-tools/tcpcopy
[82] Jon D, Seth E, Bruce AM, Jeff P, Kaustubh P. iPerf-The ultimate speed test tool for TCP, UDP and SCTP. https://iperf.fr/
[83] GitHub. netperf. https://github.com/HewlettPackard/netperf
[84] Norbert Vegh. Ntools. http://norvegh.com/ntools/index.php?page=download
[85] Zander S. KUTE-kernel-based traffic engine. http://caia.swin.edu.au/genius/tools/kute/
[86] GitHub. Ostinato. https://github.com/pstavirs/ostinato
[87] GitHub. TGen. https://github.com/shadow/tgen
[88] Avallone S, Guadagno S, Emma D, Pescapè A, Ventre G. D-ITG distributed internet traffic generator. In: Proc. of the 1st Int'l Conf. on the Quantitative Evaluation of Systems (QEST'04). Enschede: IEEE, 2004. 316–317.
[89] Avallone S, Guadagno S, Emma D, Pescape A, Ventre G. D-ITG distributed internet traffic generator. In: Proc. of the 1st Int'l Conf. on the Quantitative Evaluation of Systems. Enschede: IEEE. 2004. 316–317. [doi: 10.1109/QEST.2004.1348045]
[90] Kansas Uo. Introduction to NetSpec. http://www.ittc.ku.edu/netspec/usage/intro.html
[91] Nmap. Introduction. https://nmap.org/nping/
[92] Free Software Foundation. Mtools. 2020. https://www.gnu.org/software/mtools/
[93] Kant K, Tewari V, Iyer R. Geist: A web traffic generation tool. In: Field T, Harrison PG, Bradley J, Harder U, eds. Computer Performance Evaluation: Modelling Techniques and Tools. Berlin, Heidelberg: Springer, 2002. 227–232. [doi: 10.1007/3-540-46029-2_17]
[94] The Apache Software Foundation. Ab-ApacheHTTP server benchmarking tool. http://httpd.apache.org/docs/2.4/programs/ab.html
[95] GitHub. GoReplay. https://github.com/buger/goreplay
[96] GitHub. BoNeSi. https://github.com/Markus-Go/bonesi
[97] David B, Laura T. Cybersecurity experimentation of the future (CEF). 2015. http://www.csl.sri.com/projects/cef/

附中文参考文献:

[1] 中国网信网. 《国家网络空间安全战略》全文. 2016. http://www.cac.gov.cn/2016-12/27/c_1120195926.htm
[2] 方滨兴, 贾焰, 李爱平, 张伟哲. 网络空间靶场技术研究. 信息安全学报, 2016, 1(3): 1–9. [doi: 10.19363/j.cnki.cn10-1380/tn.2016.

03.001]
[3] 中国网络空间新兴技术安全创新论坛团体标准. 2021. https://www.ccsia.com.cn/
[22] 王帅, 朱峰, 姚益平, 唐文杰, 肖雨豪. 基于容器的复杂系统仿真应用自动封装技术. 系统仿真学报, 2020, 32(11): 2199–2207. [doi: 10.16182/j.issn1004731x.joss.20-fz0423]
[23] 刘彪, 王宝生, 邓文平. 云环境下支持弹性伸缩的容器化工作流框架. 计算机工程, 2019, 45(3): 7–13. [doi: 10.19678/j.issn.1000-3428.0049811]
[24] 纪一方, 张国敏, 邢长友. 用户行为仿真的部署与调度问题研究. 信息技术与网络安全, 2020, 39(1): 78–82. [doi: 10.19358/j.issn.2096-5133.2020.01.015]
[31] 黄宁, 刘渊, 王晓锋. 基于交互式流量回放的用户行为仿真技术. 计算机工程, 2021, 47(10): 103–110.
[37] 褚伟波, 管晓宏, 蔡忠闽, 陶敬. 面向互动式网络场景再现的流速控制系统与方法. 计算机学报, 2012, 35(7): 1485–1497. [doi: 10.3724/SP.J.1016.2012.01485]
[40] 唐积强, 王东滨, 邹潇湘, 董琳, 颜靖华, 张勖, 智慧, 张俊辉, 吴可欣, 晋红亮. 一种基于数据流特征的网络流量生成方法. 中国, 105049277A. 2015-11-11.

符永铨(1983－), 男, 博士, 副研究员, CCF 高级会员, 主要研究领域为网络行为大数据仿真, 网络靶场, 人工智能.

刘红日(1982－), 男, 博士, 助理研究员, 主要研究领域为网络行为模拟, 工业互联网安全.

赵辉(1982－), 男, 博士, 工程师, CCF 专业会员, 主要研究领域为大数据分析处理, 网络行为仿真模拟, 网络靶场.

安伦(1985－), 男, 博士生, 工程师, 主要研究领域为云计算, 网络靶场, 网络仿真.

王晓锋(1978－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为网络仿真, 网络安全.