

云虚拟网络安全研究

涂碧波^{1,2}, 孙瑞娜^{1,2,3}, 游瑞邦^{1,2}, 程杰^{1,2}, 陶小结^{1,2}, 张坤^{1,2}

¹中国科学院信息工程研究所, 北京 中国 100093

²中国科学院大学 网络空间安全学院, 北京 中国 100093

³新疆财经大学 信息管理学院, 乌鲁木齐 中国 830012

摘要 云计算以虚拟化技术为基础, 提供了一种按需、灵活分配资源的网络计算模式。在网络虚拟化技术的推动下, 用户的网络变为云服务提供商根据用户需求, 在物理网络之上为其分配的逻辑上相互隔离的虚拟网络。虚拟网络带来了网络架构的动态性, 呈现出网络边界动态模糊、共享底层资源及流量以内部“东西”向交互为主的新特性, 不仅加剧了传统网络固有的攻击威胁(如 ARP 攻击、DoS 攻击等), 还引入了新的安全威胁: 虚拟网络边界防护失效、信息泄露及篡改、流量监控存在盲点等。因此, 虚拟网络的安全问题成为工业界和学术界关注的热点。本文对虚拟网络环境中存在的安全问题进行了归纳, 分析产生的原因, 给出了云虚拟网络的威胁模型; 并针对这些安全问题, 从基于虚拟防火墙、基于安全服务动态部署、基于虚拟网络嵌入、基于虚拟网络隔离强化、基于深度流量监测、基于流量动态控制等类别分别对近年国内外相关防御机制进行了分析和比较, 并指出了当前仍存在的问题; 最后对虚拟网络安全未来研究方向进行了探讨, 给出了基于软件定义边界的动态防御框架。

关键词 网络安全; 云虚拟网络; 软件定义网络; 软件定义边界

中图分类号 TP393 DOI号 10.19363/J.cnki.cn10-1380/tn.2025.03.13

Research on Cloud Virtual Network Security

TU Bibo^{1,2}, SUN Ruina^{1,2,3}, YOU Ruibang^{1,2}, CHENG Jie^{1,2}, TAO Xiaojie^{1,2}, ZHANG Kun^{1,2}

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

³ School of Information Management, Xinjiang University of Finance and Economics, Urumqi 830012, China

Abstract Cloud computing, with virtualization technology as its base, provides a network computing model that allocates resources flexibly on demand. Driven by network virtualization, the traditional user network is transformed into logically isolated virtual networks, which are allocated by cloud vendors on the physical network according to users' needs. The virtual network brings the dynamic and flexible nature of the network architecture, and presents the new characteristics of dynamic blurring boundary, sharing underlying resources and traffic based on internal “east-west” interaction. But it aggravates the inherent attack threats of traditional network, such as ARP attack, DoS attack, etc. Also, new security threats are introduced: virtual network perimeter protection failure, information leakage and tampering, blind spots in traffic monitoring and so on. Therefore, the security of virtual network has become a hot spot in industry and academia. This paper summarizes the security problems in virtual network environment, analyzes the causes, and gives the threat model of the cloud virtual network. In response to these security issues, this paper analyzes and compares the defense mechanisms at home and abroad from categories based on virtual firewall, security service dynamic deployment, virtual network embedding, virtual network isolation enhancement, deep traffic monitoring, traffic dynamic control, and etc. We also point out the existing problems of these schemes. Finally, we discuss the future research direction of virtual network security, and propose the framework of a dynamic defense based on Software Defined Perimeter.

Key words network security; cloud virtual network; software defined networking (SDN); software defined perimeter (SDP)

1 引言

云计算以虚拟化技术为基础, 为了更好的提升网络资源的利用率, 在网络虚拟化技术^[1]的发展推动下, 云底层物理网络资源开始向虚拟化深度延伸。

在云平台中, 网络服务提供商可以根据用户的业务需求, 在物理网络上创建多个逻辑上相互隔离的虚拟网络。云虚拟网络可以按需部署及管理, 在提高资源利用率的同时, 促进了网络部署的动态性和灵活性。然而, 由于云虚拟网络共享底层物理网络资源,

通讯作者: 游瑞邦, 博士, 高级工程师, Email: youruibang@iie.ac.cn。

本论文得到国家重点研发计划课题(No. 2016YFB0801002)和中国科学院先导科技专项(C类)课题(No. XDC02010900)资助。

收稿日期: 2020-09-29; 修改日期: 2020-12-22; 定稿日期: 2025-02-25

并且网络边界模糊动态, 攻击者更容易实施攻击, 并且攻击行为不易被发现。因此, 云虚拟网络攻击相比物理网络攻击所造成的危害更大, 影响更广泛^[2]。目前, 云虚拟网络面临诸多安全挑战, 主要体现在如下四个方面:

(1) 网络边界支离破碎。随着用户大量使用云主机, 网络的通信端由物理服务器变为物理服务器中运行的虚拟机, 虚拟机通过服务器内部提供的软件交换设备(如虚拟网桥、虚拟交换机等)实现与虚拟机和外部网络之间的通信, 网络边界从服务器的物理边界延伸到服务器内的虚拟边界, 传统“基于物理网络边界”的安全防护机制无法直接用于虚拟网络边界防护。

(2) 恶意流量无法检测。由于云主机集群间协同工作, 虚拟机之间的交互流量不断增加, 使得云中 80% 的流量为同一租户虚拟网络内不同虚拟机之间, 或同一租户不同虚拟网络之间的二层通信流量(即“东西”向流量), 现有流量监控机制对“东西”向流量无法感知或无法检测, 难以发挥流量监控设备的作用^[3]。

(3) 攻击手段更隐蔽。云虚拟网络之间共享物理资源的特点, 使得攻击者的攻击手段更加隐蔽且多样。例如攻击者利用恶意网络服务在宿主机内部尝试各种攻击, 窃取机密数据^[4-5], 目前采用一种通用的防护方式难以实现对攻击的有效检测和防护, 需要更多特定、组合的安全防护方式。

(4) 攻击破坏性更大。对于传统网络中的安全问题, 例如拒绝服务攻击^[6-8]、ARP 攻击、脚本攻击等, 在云虚拟网络环境中增无减, 且相比传统物理网络下受到的攻击而言, 这种攻击带来的破坏性更大。

为了确保云虚拟网络环境的安全性, 建立有效的虚拟网络防护体系及其重要。为此, 本文聚焦于云虚拟网络安全研究, 通过分析云平台内部虚拟网络环境特有的攻击和漏洞, 将虚拟网络安全问题归纳为虚拟网络边界安全问题、信息泄露及篡改问题和“东西”向流量安全问题, 同时针对以上三个方面的安全问题, 归纳整理了近年来工业界和学术界所提出的安全机制和解决方案, 并比较分析了现有方案的不足之处, 展望了未来虚拟网络安全的研究方向, 并给出了基于软件定义边界的动态防御框架。

本文组织结构如下: 第 2 节对云虚拟网络相关技术进行简要介绍, 第 3 节对虚拟网络的安全问题进行梳理和总结, 第 4 节对第 3 节的安全问题阐述近年工业界和学术界所提出的防护方案, 第 5 节展望虚拟网络安全未来发展方向, 第 6 节总结全文工作。

2 云虚拟网络相关技术

在云环境中, 用户通过创建虚拟网络(Virtual Network, VN)来使用网络服务, 虚拟网络是虚拟机间相互通信的网络节点和链路的集合, 其网络实现基于网络虚拟化技术。网络虚拟化作为云基础架构的核心网络技术, 基于该技术可以不受数据中心物理网络拓扑的限制, 根据不同用户的业务需求, 在物理网络基础之上为用户提供所需的一个至多个虚拟网络, 同时保持其隔离性。目前, 云中的虚拟网络服务以 VLAN 和 VxLAN 技术为主, 其基础架构如图 1 所示。

VLAN 为二层网络技术, 通过向每个网段添加 VLAN 标识, 实现虚拟网络隔离。处于不同物理服务器上的虚拟机通过 VLAN 机制可划分到同一虚拟网络中, 服务器内部同一 VLAN 的虚拟机之间通过虚拟机监控器(Virtual Machine Monitor, VMM)提供的软件交换机进行通信, 而服务器之间同一 VLAN 中的虚拟机流量交互通过内部软件交换机经服务器的物理网卡流出到外部交换机进行。

随着云环境中虚拟机数量的增长, 基于业务灵活性变更需求, 虚拟机迁移成为常态业务, 基于 VLAN 的虚拟网络划分存在虚拟机迁移受限的问题, 同时, 大规模租户的部署使得虚拟网络数量增长, 而 VLAN 虚拟网络数量受限于 4096 个, 不能满足部署需求。为此, 业界提出的 overlay 网络技术可在对现有基础网络不进行大规模修改的条件下, 在基础网络之上构建虚拟网络, 支持虚拟机的灵活迁移。当前对云环境虚拟网络服务构建最具代表性的 overlay 技术是 VxLAN, VxLAN 通过三层网络来搭建虚拟的二层网络, 其扩展的隔离标识位数, 很好的解决了 VLAN ID 数量受限问题。

对于 VxLAN 技术, 在多个计算节点相互连接的场景中, 位于不同计算节点同一 VxLAN 虚拟机之间的通信由虚拟交换机 br-tun 实现虚拟三层网络的连接, 对收到的数据包进行标记转换后经隧道(VxLAN Tunnel)送出。VxLAN 技术以全局 VxLAN Tunnel ID 和局部 VLAN TAG 两套机制来对不同的虚拟网络予以标识区分。

由以上云虚拟网络的相关技术, 结合文献[9-10]给出的虚拟网络定义, 在此将本文研究的云虚拟网络定义为: 根据用户业务需求, 在底层物理网络的基础上按需构建的逻辑拓扑网络, 同时保持它们的隔离性, 是实现云平台虚拟机间通信的网络节点和链路的集合。本文后续内容所提到的虚拟网络, 均指云平台内的虚拟网络(即云虚拟网络)。

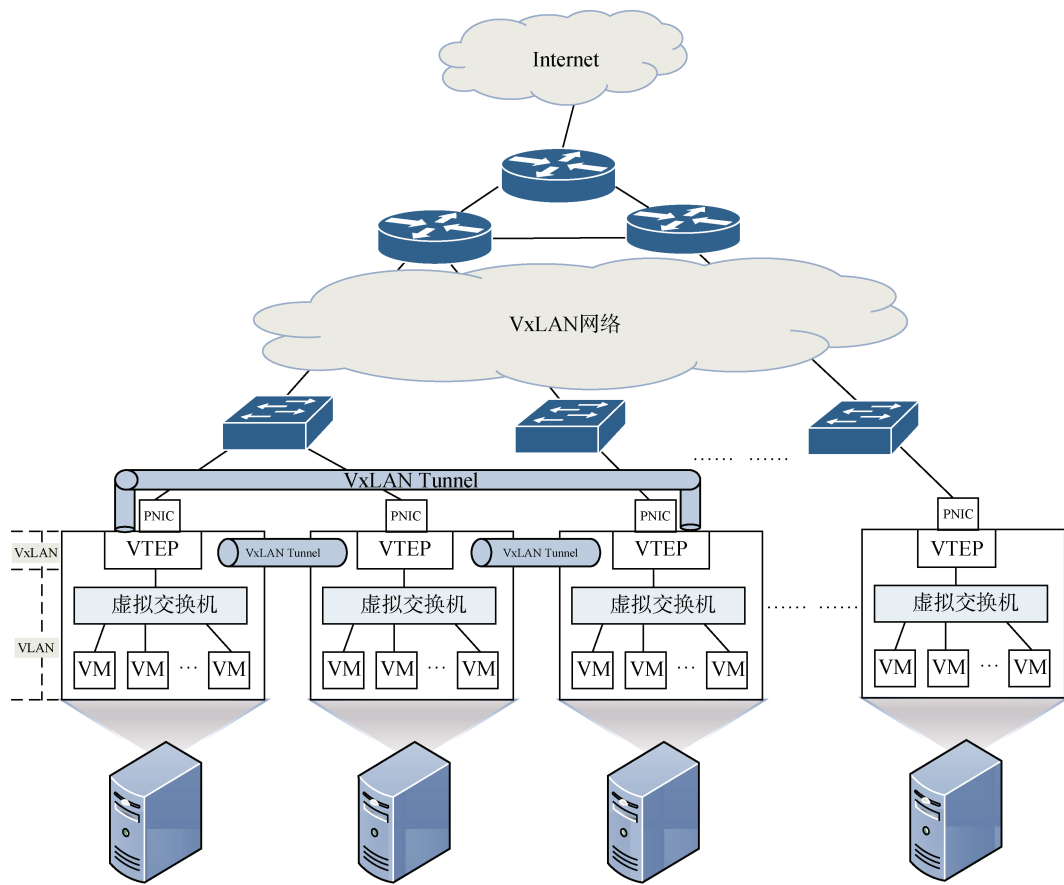


图 1 虚拟网络环境及基础架构

Figure 1 Virtual network environment and basic architecture

3 云虚拟网络的安全问题

为了形象地描述虚拟网络的安全问题, 本节对虚拟网络环境中可能存在的安全问题进行分析, 将其归结为三大类 安全问题, 构建了如图 2 所示的虚拟网络场景安全模型。

3.1 虚拟网络边界安全问题

3.1.1 传统边界防护机制失效

在传统物理网络环境中, 网络边界固定, 安全防护系统基于固定边界(如核心交换机处)部署, 实现内外网的安全隔离。但虚拟网络到实际物理网络拓扑的映射是相对分散的, 同一虚拟网络的虚拟机很可能分布在多台物理服务器上, 且虚拟机因资源调度及业务需求的动态迁移性, 使得虚拟网络的映射动态变化, 网络边界变得动态、模糊缺乏明确定义, 同时, 虚拟网络环境软件化、动态多变的新特性, 使得攻击方式更隐蔽、多样^[1]。因此, 这种基于固定边界部署且配置后长期不变的防护方式难以实现对攻击的有效检测和防护, 需要使用更多可按需组合动态部署的安全防护措施。

3.1.2 安全策略部署方式无法有效实施

对于传统边界防护机制, 安全策略通常由运营商根据安全需求手动规划部署, 在策略集数量不大且安全策略改动次数较少时, 这种方式具有适用性。但在虚拟网络环境中, 网络攻击的传播速度更快, 如数据泄露攻击在分钟到小时级即可完成^[12], 当安全机制发现恶意攻击时, 这种依靠人工配置和更新安全策略的方式, 一方面难以根据安全需求及时、有效地响应, 会使虚拟网络处于易受攻击的状态; 另一方面策略管理方式缺乏灵活性, 存在安全策略配置错误(或冲突)的问题, 易造成严重的数据泄露及防火墙漏洞^[13-14]。对此, 需要将安全策略手动的配置方式提升到软件层面半自动化(或自动化)动态的配置方式。同时, 在安全策略部署前, 既要满足单个安全策略的需求, 也要确保策略组合的有效性, 以降低安全策略配置后出现冲突或导致错误的概率。

3.2 信息泄露及篡改问题

3.2.1 拓扑泄露

云环境物理网络拓扑的静态特性, 以及服务器内部同驻虚拟机通常共享数字相近的内部 IP 地址、

同驻虚拟机之间数据包往返时间更短^[4]等网络特征, 易被攻击者利用实施网络拓扑探测。扫描探测虚拟机以及重复的虚拟网络嵌入请求是最常见的拓扑探测手段, 例如攻击者可通过多次重复的虚拟网络嵌入请求来获取有关物理网络的相关信息, 进而推断网络拓扑结构以发现潜在目标宿主机(或虚拟机)的位置以及可以利用的漏洞^[15]。这种拓扑泄露易成为攻击者在网络通信端(如虚拟机)实施同驻攻击, 或在网络级别实施 DoS、ARP 等攻击的基石。

3.2.2 数据泄露

虚拟网络环境复杂, 任何漏洞都可能被恶意用户利用, 通过受害主机实现对目标虚拟网络的攻击。在虚拟网络环境中, 虚拟化技术提供的逻辑隔离边界与传统网络中的物理隔离边界相比, 隔离性并不强, 易导致数据泄露^[16]。因此, 一旦攻击者通过泄露的网络拓扑信息, 进一步实现其恶意虚拟机与目标虚拟机的同驻(同一物理主机或同一子网), 可在

成功入侵目标虚拟机后, 通过目标虚拟机, 利用虚拟化技术自身存在的安全漏洞, 对其他虚拟机及内部网络实施更有针对性的攻击, 以达到窃取或破坏受害敏感数据的目的^[17]。例如, 对图 2 所示场景中的虚拟机跳跃(VM Hopping)^[18]攻击, 外部攻击者在成功入侵目标虚拟机之后, 可利用目标虚拟机做“跳板”, 通过共享内存或者 hypervisor 漏洞建立隐蔽信道, 从而绕过隔离机制, 在受害虚拟网络内部虚拟机之间, 或相邻虚拟网络虚拟机之间实施违规通信, 这种虚拟机跳跃攻击会导致数据从一个虚拟网络泄露到另一个虚拟网络。同样, 对于侧信道(Side-channel)^[19-20]攻击, 攻击者可通过探测共享资源的状态信息, 使得虚拟化技术提供的逻辑边界失效, 从而实施跨虚拟机的侧信道攻击, 以实现窃取相关私钥、证书等数据的目的, 并进一步通过虚拟网络传播其恶意行为, 导致对整个云虚拟化网络环境造成严重威胁。

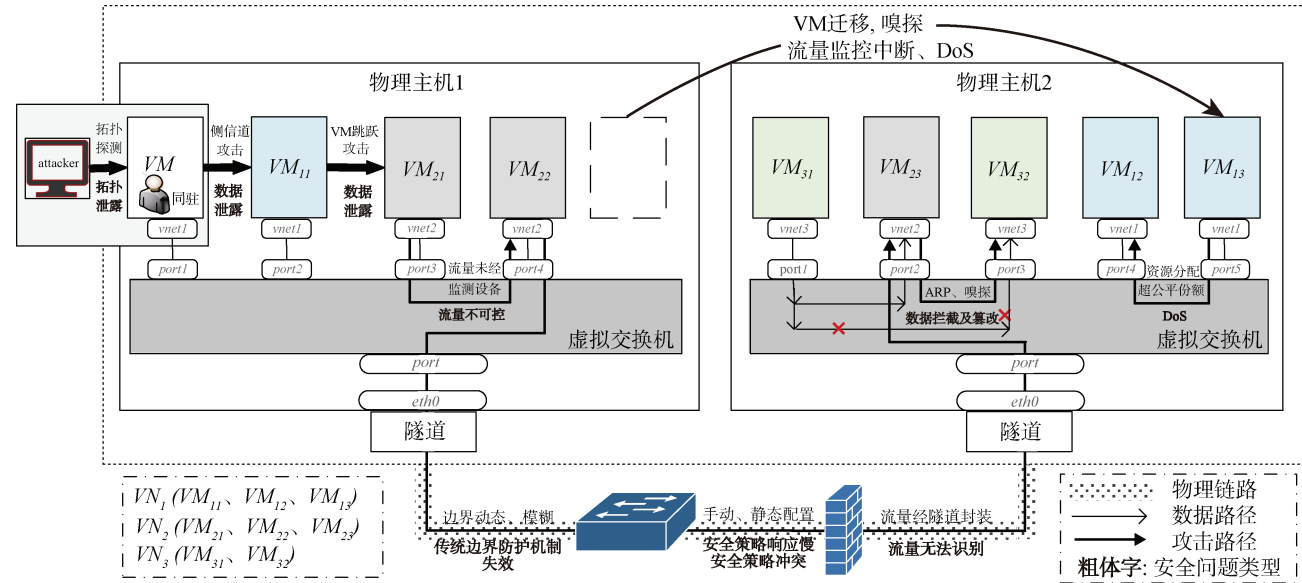


图 2 虚拟网络安全模型
Figure 2 Security model of virtual network

3.2.3 数据拦截及篡改

虚拟网络中也存在传统网络固有的安全威胁, 如 ARP 攻击、嗅探攻击等二层网络攻击, 例如, 对于常见的 ARP 攻击, 攻击者可通过 ARP 欺骗^[21]技术修改受害者本地的 ARP 缓存表(添加、更新缓存项), 使得恶意主机的 MAC 地址与目标主机的 IP 地址相关联, 进而将数据包重定向到攻击者, 实现截获机密数据以及伪造数据的注入。在 ARP 攻击基础上, 攻击者还可以实施更高级别的攻击(会话劫持、DoS 攻击、重放攻击等)造成更严重的安全危害。例如, 对于重放攻击, 攻击者捕获传输的合法数据包后, 可

通过恶意的重复传输, 导致其他实体认为消息已发送多次, 来达到欺骗系统的目的。在基于虚拟设备互联、大二层架构的虚拟网络环境中, 虚拟交换机相比物理交换机缺少复杂的二层安全控制, 因此, 在虚拟网络中对常见的二层网络攻击防护难度更大, 特别是随着二层网络范围的扩大化, 虚拟网络环境中二层网络攻击的危害性远超对传统网络的影响。

3.3 “东西”向流量安全问题

3.3.1 流量绕过安全机制

虚拟网络环境中, 同一服务器内的虚拟机之间,

通过内部虚拟交换机相连, 虚拟机之间存在直接交换的“东西”向流量, 因这种“东西”向流量不经过外部物理安全设备, 而难以做到有效防护。例如在图 2 所示场景中, 属于同一虚拟网络(Virtual network, VN) VN_2 的虚拟机 VM_{21} 和 VM_{22} 之间的二层网络流量, 通过宿主机内部的虚拟交换机完成交换, 流量只在宿主机内部传输, 不再通过真实的物理网卡到达部署在核心(或汇集)物理交换机处的安全设备, 使得安全设备无法对这些“东西”向流量进行监控, 一旦虚拟机受到攻击, 则会导致内网虚拟机及同驻虚拟机受到威胁。

3.3.2 流量无法识别

虚拟网络环境中, 不同服务器内虚拟机之间的流量交换虽然经过了物理交换机上部署的安全设备(如防火墙), 但无法被安全机制所理解。例如图 2 所示的场景, 虚拟网络 VN_2 中的虚拟机 VM_{22} 和虚拟机 VM_{23} 分别位于物理主机 1 和物理主机 2 中, 主机之间通过 VxLAN 隧道相连, VM_{22} 到 VM_{23} 的流量通过物理主机 1 的虚拟交换机, 经隧道封装到物理网络中, VM_{22} 到 VM_{23} 的流量虽然经过了安全防护设备, 但封装后无法解析, 导致恶意流量无法识别。

3.3.3 流量监控中断

虚拟网络中虚拟机部署灵活, 可动态迁移, 由于现有安全监测设备的固定性, 安全策略无法跟随虚拟机迁移状态及时、动态部署, 易造成原监控节点流量监控的中断, 虚拟机在迁入到新节点的过程中, 其内存会暴露在网络中, 易受中间人、拒绝服务等攻击(如图 2 所示), 造成迁移数据被嗅探、篡改或破坏^[22], 而因原监控数据的中断, 新监控节点难以

监测到迁入虚拟机的异常数据流。

(4) 拒绝服务

在虚拟网络环境中, DoS 攻击会产生更大影响。攻击者入侵虚拟机后, 可通过内部虚拟网络对同驻虚拟机或其他节点的虚拟机进行大流量攻击, 导致网络瘫痪^[23]。攻击者还可以通过对路由器和链路攻击造成当前使用其资源的虚拟网络服务中断^[24]; 对共存在同一网络基础设施中的多个虚拟网络, 攻击者可使其中一个虚拟网络的资源超过其公平份额来破坏其他虚拟网络^[25]。虚拟化网络环境也为攻击者实施 DDoS(Distributed Denial of Service, DDoS)攻击提供了便利, 攻击者可以在不同位置批量租用虚拟机或者利用受控虚拟机组成僵尸网络, 从多个攻击源发送大量恶意流量, 消耗网络带宽, 从而使正常服务请求被拒绝^[26-27], 这种攻击在虚拟网络环境中难以检测和过滤使得防御问题更加复杂。

3.4 小结

综上所述, 虚拟网络不仅存在传统网络固有的安全问题(如 ARP 攻击、DoS 攻击、重放攻击等), 其新特性又产生了新的安全问题: 传统边界防护机制及安全策略配置方式无法满足虚拟网络防护需求; 网络拓扑结构泄露易成为同驻攻击的基石, 攻击者易于利用虚拟化技术安全漏洞对目标虚拟网络实施攻击; 虚拟网络“东西”向流量存在监控盲点等。本节对虚拟网络存在的安全问题进行了总结, 如表 1 所示, 在第 4 节中对近年工业界和学术界所提出的应对解决方案进行详细阐述。

表 1 虚拟网络安全问题分类
Table 1 Classification of virtual network security problems

安全问题	安全问题类型	问题描述
虚拟网络边界安全问题	传统边界防护机制失效	(1) 传统固定边界的防护机制无法满足虚拟网络动态边界防护需求 (2) 虚拟网络环境攻击方式隐蔽多样, 传统边界防护机制难以检测
	安全策略响应慢	(3) 安全策略无法随虚拟网络的变化及时、动态配置
	安全策略冲突	(4) 传统安全策略配置方式易造成策略冲突, 导致攻击漏洞
	拓扑泄露	(1) 网络拓扑探测导致拓扑结构泄露 (2) 系统漏洞被攻击者利用实施 VM hopping 攻击
信息泄露 及篡改问题	数据泄露	(3) 攻击者实施侧信道攻击 (4) 虚拟机间 ARP 攻击、嗅探攻击
	数据拦截及篡改	(5) 攻击者持续重放合法数据包
“东西”向流量安全问题	流量不可控	(1) 物理主机内虚拟机间“东西”向流量未通过外部监测设备
	流量不可见	(2) 流经外部监测设备的数据流经隧道技术封装而无法识别
	流量监控中断	(3) 虚拟机迁移导致原节点流量监控中断
	拒绝服务	(4) 虚拟网络共享基础设施使得 DoS/DDoS 攻击效果放大且不易检测

4 虚拟网络安全防御方案

为了增强虚拟网络安全, 本节从上述三大类虚拟网络的安全问题出发, 按上述安全问题的逻辑顺序, 对目前业界主流的安全防护机制进行阐述。

4.1 虚拟网络边界安全防护方案

虚拟网络边界动态、模糊, 因此需重构传统网络的安全防护方式, 以解决虚拟网络边界安全问题。本节将从传统边界防护强化(4.1.1)和软件定义的防护(4.1.2), 对工业界和学术界提出的解决方案进行讨论。

4.1.1 传统边界防护强化

对于传统边界防护强化方案, 本节主要从硬件安全设备和虚拟安全设备之一的虚拟防火墙进行讨论。

(1) 硬件安全设备

在传统基于固定边界处部署硬件安全设备的基础上重用硬件安全设备, 如 FireWall、IDS/IPS、WAF、UTM 等, 通过控制服务器内虚拟机流量到外部硬件安全设备进行检查^[28], 同时在处理数据包时, 增加解隧道和封装隧道的功能实现对虚拟网络流量的有效检测(如图 3 所示)。

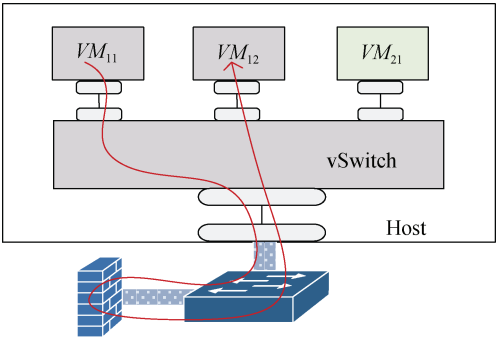


图 3 基于硬件安全设备
Figure 3 Hardware-Based security equipment

虽然这种方式可利用现有物理安全设备的高性能, 但硬件安全设备缺乏与虚拟化管理系统的集成, 不易弹性扩展, 并且随着托管在物理服务器上虚拟机数量的增加, 同一物理服务器上虚拟机之间的通信量会非常大, 不可能将所有的流量都从服务器中路由到硬件安全设备进行检查并返回, 从而导致硬件安全设备存在防护盲点, 且这种发夹式回路易造成网络性能损失, 同时, 该方式无法感知虚拟网络边界的动态变化。因此, 重用硬件安全设备难以实现对虚拟网络的有效防护。为了解决硬件安全设备的不足, 虚拟安全设备应运而生。一些主流的安全供应

商, 如 Juniper、山石网科、vArmour 等分别调整了现有硬件安全产品, 以实现其在虚拟化平台中工作。

(2) 虚拟安全设备之虚拟防火墙

虚拟安全设备, 部署在服务器内部对内部流量进行管控, 避免了将流量重新路由到外部而导致增加额外的流量。本节重点对作为核心虚拟安全设备之一的虚拟防火墙进行分析。目前, 从虚拟防火墙在云中的部署位置来看, 主要有以下两类: 内核级模式的虚拟防火墙和子网级模式的虚拟防火墙^[29]。

① 内核级模式的虚拟防火墙

内核级模式的虚拟防火墙放置在 hypervisor 和虚拟交换机之间, 其安全防护功能作为 hypervisor 的可执行模块, 嵌入 hypervisor 中, 通过虚拟化层开放的 API 拦截进出虚拟机虚拟网卡的流量, 流量的过滤可直接在 hypervisor 中完成, 也可以将截获的流量转发给专用的虚拟机进行处理。例如 VMware 公司推出的安全虚拟设备套件 vShield^[30], 其中的 vShield app 可描述为图 4 所示的虚拟防火墙, 该虚拟防火墙由 hypervisor 的可执行模块和部署在物理主机内的虚拟安全设备(Virtual Security Apparatus, VSA)组成, vShield VSA 为预配置的用于防火墙操作的专用虚拟机, hypervisor 的可执行模块将包过滤器放置在每个虚拟机的虚拟网卡上, 以检查进出的流量, 还可以将流量牵引到 vShield VSA 中进行检查。

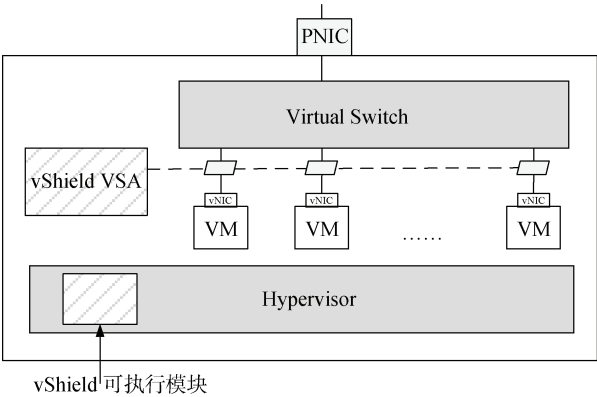


图 4 vShield 虚拟防火墙
Figure 4 vShield virtual firewall

基于内核级模式的虚拟防火墙, 虚拟机流量不需要离开主机即可完成捕获及处理, 具有更高的性能, 但因其安全功能执行模块需要与 hypervisor 紧密耦合, 而受限于特定的虚拟化平台, 且难以与虚拟网络的管理工具集成。

② 子网级模式的虚拟防火墙

子网级模式的虚拟防火墙运行在专用虚拟机上, 消除了对 hypervisor 的完全依赖, 易于同其他相关

安全设备集成, 可与物理防火墙一样, 按需部署在网络的核心检查点。虚拟防火墙通过所在虚拟机上配置的多块虚拟网卡, 分别与虚拟网络的不同子网相连接, 这种防火墙避免了发夹式回路, 同时还可以为虚拟网络提供弹性的防火墙服务。目前, 已有大量学者致力于虚拟防火墙的可伸缩性研究, 文献[31]提出的通用云防火墙框架以满足云平台防火墙动态分配的需求, 该方案通过建立基于事件的集中式检测链机制实现虚拟防火墙的按需动态分配。文献[32]提出一种基于性能和成本分析的集群化防火墙框架, 除考虑防火墙的动态分配外, 还同时考虑满足用户 QoS 级别的要求。因负载均衡在虚拟防火墙创建、管理和部署中发挥着重要作用, 适当的负载均衡可提高系统资源的利用率, 为此, 文献[33]提出的弹性虚拟防火墙体系结构, 除考虑了防火墙的可伸缩性外, 还考虑了负载均衡和虚拟防火墙的处理性能。文献[34]也提出了将强化学习和遗传算法相结合的自动伸缩算法, 用于实现防火墙的负载预测和负载均衡的自适应。

然而, 虽然以上基于集群化、多线程、负载均衡的虚拟防火墙部署方案在解决防火墙可伸缩性方面具有实用性, 但因虚拟网络边界模糊、网络拓扑随虚拟机迁移动态变化的特性, 以上方案在感知虚拟网络边界变化、安全策略协同部署方面仍然有限, 并且基于传统单一模式的防御体系难以应对虚拟网络复杂隐蔽的攻击方式, 将多个安全设备灵活组合的防御模式已成为发展趋势。

4.1.2 软件定义的防护

随着软件定义网络(Software Defined Networking, SDN)和网络功能虚拟化(Network Functions Virtualization, NFV)技术的不断发展, 为虚拟网络边界防护提供了新的解决思路, 首先, 可结合 SDN 技术构建虚拟安全设备, 以解决因虚拟网络边界动态模糊、难以在网络边界处部署安全设备提供检查点的问题。其次, SDN 被扩展到了安全运维领域, 出现了软件定义安全(Software defined security, SDS)的概念^[35], 基于软件定义安全可根据虚拟网络防护需求构建安全服务链, 实现虚拟安全设备按需灵活组合、动态部署的纵深防护模式。

对此, 本小节依次从虚拟安全设备之一的 SDN 防火墙, 以及安全服务动态部署, 对现有防护方案进行深入分析。

(1) 虚拟安全设备之 SDN 防火墙

基于 SDN 技术的网络体系结构, 控制平面与数据平面相分离, 用户通过控制器与整个网络交互,

数据平面只根据控制器定义的流表对传入的数据包进行转发, 这种网络结构易于将控制器实现为无状态防火墙, 强制所有的流量通过控制器进行检查, 以允许或拒绝通过。如文献[36]提出的防火墙解决方案, 控制器充当集中式防火墙, 所有数据包被重定向到控制器, 用户根据需求从控制器外部对防火墙规则的优先级进行切换, 按不同防火墙规则优先级对数据包头进行检测。这种集中式的包过滤防火墙的实现和维护简单, 但易导致控制器过载及控制器和交换机之间通信开销过大, 对此, 可考虑通过控制器将防火墙的所有规则部署到每个交换机中, 将交换机作为分布式的防火墙检查点, 以减轻控制器的工作负载^[37-38]。然而, 由于网络状态及配置的动态变化(如 SDN 流表或防火墙规则的更新)可能导致防火墙规则冲突, 而仅通过检查数据包头字段来判断是否存在违规行为, 且未对网络策略的更新进行实时检测, 存在安全隐患。

为此, FortNOX^[39]对 NOX 控制器进行了扩展, 从控制器内核层面开发了实时规则冲突检测模块, 但该方案不能直接用于防火墙。VeriFlow^[40]与 NetPlumber^[41]等实时策略检查工具, 作为控制器和交换机之间的插件, 使用转发图对网络行为建模, 通过拦截转发规则, 利用增量计算在规则到达网络前进行检测。以上方案虽然能检测防火墙策略冲突, 但不支持有效的冲突解决。而文献[42-45]在策略冲突检测的基础上给出了冲突解决方案, 文献[42]基于头部空间分析(Header Space Analysis, HAS)^[43]算法对防火墙策略新增、策略更新及交换机流表条目新增时可能导致的策略冲突, 提出冲突检测和解决方案。FlowGuard^[44]和 FlowVerifier^[45]提出防火墙策略冲突动态检测及冲突灵活解决的架构方案。FlowGuard 同样基于 HSA 算法建立对整个网络的流追踪机制, 当网络状态或配置发生变化时, FlowGuard 会检查网络流路径空间查看是否与防火墙策略发生了冲突, 并分别对防火墙配置时可能出现的策略冲突、流路径空间与防火墙授权空间可能出现的冲突问题, 提出冲突解决机制。

由于基于 SDN 构建的无状态防火墙本身不考虑当前网络的连接状态, 虽然在流量高负载的情况下无状态防火墙具有速度快的优势, 但因其缺乏数据包状态信息难以发现来自数据平面的攻击^[46], 而利用连接状态信息构建有状态的防火墙, 可以定义更细粒度的规则, 实现更精确的违规检测机制。对此, 文献[47]提出了 SDN 响应式状态防火墙, 引入了处理 TCP 协议的有限状态机(Finite State Machine, FSM)

算法, 该方案的防火墙使用状态表记录网络连接的不同状态及其属性, 并根据连接的状态来处理流量, 对每个连接的状态, 只接收与此状态转换相对应的通信, 此机制可以限制连接的同步请求数, 并清除与连接状态不一致的通信, 减轻了 DoS 攻击(如 syn flooding)。

对于有状态防火墙, 因控制器必须跟踪交换机上的连接状态, 导致控制器和交换机之间用于连接跟踪的通信开销过大, 使得控制器过载及连接延迟增加。因此, 实现有状态防火墙的同时, 还需进一步对该问题提出解决方案。文献[48]引入一种三层的交换机流表结构, 来区分传入交换机的流量, 避免对控制器发出重复的连接请求。此外, 混合使用“主动”和“被动”流规则, 以最小化控制器配置交换机流表的工作负载。文献[49]提出在流表之外使用构建的 Stable 和 SecPolTable 表来提供状态感知, 以减少交换机和控制器之间的通信开销。除以上解决方案之外, 还可以通过将防火墙从控制平面移动到数据平面来减轻控制器的负载, 避免控制器因接收大量基于状态的流量成为性能瓶颈^[50]。例如 FORTRESS^[51] 防火墙完全在数据平面内运行, 利用数据平面交换机识别和丢弃非法的数据包, 从而最小化数据包在数据平面和控制平面之间转发产生的控制器负载和网络连接开销。

如果只根据特定的包头字段或包的状态过滤流量, 而不对包的有效载荷进行深入分析, 则难以过滤高风险的流量(包含恶意载荷的数据包)。为此, 有学者重点从防火墙对恶意流量的识别提出解决方案, AI-SDNF^[52]提出的智能 SDN 防火墙方案, 设计了一个独立于 SDN 控制器的防火墙代理, 采用二元 logistic 回归模型对提取的数据包有效载荷进行分析, 判断数据包是合法还是恶意的, 对识别的恶意流量, 在其与交换机流表匹配之前进行阻止, 实现对恶意流量的过滤。文献[53]提出了一种防止恶意流量绕过以增强安全性的软件定义防火墙(SDF), 该方案在 SDN 数据平面实现网络流量的检测, 控制平面收集主机信息, 并基于 SDF 可编程语言动态更新数据平面中的安全规则, 以提高恶意流量检测的准确性。

表 2 对以上相关方案进行了小结, 基于 SDN 技术实现的防火墙可解决虚拟网络边界安全防护需求, 但对于防火墙普遍存在的策略规则到流表的转换、策略冲突及策略无法随虚拟机及时迁移、安全策略单一无法实现纵深防御等问题, 还有待深入研究。

(2) 安全服务动态部署

软件定义安全的架构设计, 使得用户按需灵活

组合安全设备, 实现联动防御成为可能。软件定义安全借鉴 SDN 的设计思想, 进行了架构分离, 并将网络安全设备与其接入、部署方式进行解耦, 底层通过 NFV 技术将安全设备虚拟化形成安全资源池, 上层通过软件定义的方式对安全设备进行自动化编排和管理。结合软件定义安全的思想, 可快速重构传统网络的安全防护方式, 解决虚拟网络边界安全防护问题。

表 2 SDN 防火墙相关方案分析

Table 2 Analysis of Relevant Schemes for SDN Firewall Approaches

相关方案	控制器	分布 式	有 状 态	冲 突 检 测	违 规 处 理	有 效 载 荷 分 析
Suh M [36]	POX	×	×	×	×	×
Pena J[37]	POX	√	×	×	×	×
Tran T[38]	POX	√	×	×	×	×
FortNOX [39]	NOX	×	×	√	×	×
Wang J [42]	FloodLight	×	×	√	√	×
FlowGuard[44]	FloodLight	×	×	√	√	×
FlowVerifier[45]	FloodLight	×	×	√	√	×
Stateful FW[47]	RYU	×	√	×	×	×
Tran T [48]	POX	×	√	×	×	×
Nife F [49]	POX	×	√	×	×	×
SDFW[50]	OpenDaylight	√	√	×	×	×
FORTRESS[51]	OpenDaylight	√	√	×	×	×
AI-SDNF[52]	OpenDaylight	×	√	×	×	√
SDF[53]	RYU	×	√	×	×	√

工业界已从不同的侧重点提出了相关安全产品及解决方案。CloudPassage 公司的 Halo 产品、云杉网络为第三方安全厂商提供的接入支持、以及 Phantom Cyber 公司构建的安全应用体系等侧重于安全运维的自动化, 可编排; 而 google 的 BeyondCorp 架构^[54-57]、Skyport Systems 公司的零信任访问控制^[58]、Check Point 的软件定义防护(Software Defined Protection, SDP)^[59]、CSA 的软件定义边界(Software Defined Perimeter, SDP)^[60]、VMWare NSX^[61] 及 Catbird 的防护架构^[62]等方案, 基于零信任和微分段的思想, 侧重于软件定义的访问控制, 提供实时的主动防御。以上安全产品从不同层面表现出了软件定义安全的特征, 但所支持的安全防护, 在安全设备、安全服务等方面均不提供开放接口, 仍然是单一封闭的安全产品, 难以在现有安全产品中添加新的功能。

学术界则提出了动态部署安全服务的边界防护方案。早期的经典方案 FRESKO^[63]作为一种基于 OpenFlow 协议部署在 SDN 控制器上的安全应用程序开发框架,可方便快速地设计和模块化安全功能。基于该方案可将安全监视和威胁检测逻辑编码为模块化库,用脚本语言将模块进行定义组织,通过链接模块实现安全模块的灵活组合。然而,这种方式需要在控制器内增加安全执行内核,因与控制器紧耦合,难以实现迁移复用,且下发的安全策略可能会出现冲突。为此,AVANT-GUARD^[64]在数据平面上实现安全功能,可减少控制平面和数据平面之间的数据交互,并能对动态数据流进行监测和响应。但该方案所提供的数据平面扩展功能仅针对 TCP 连接,对于 UDP、ICMP 等协议均不适用。同样,文献[65]也考虑将安全服务部署在数据平面上,所不同的是该方案引入了安全服务编排中心,对用户发出的安全需求进行分析后,通过对防护策略进行分解,获取所需的安全元功能,并从安全元函数库提取安全元函数,在规则引擎中组成所需的安全服务部署到数据平面,但该方案的安全服务组合缺乏灵活性。对此,文献[66]提出一种基于策略的动态安全服务功能组合设计和管理方法,操作员可通过编写服务等级协议(Service Level Agreement, SLA)来指导服务功能链的构建。

在对安全服务模块组合时,要确保模块之间不同类型策略之间的正确组合并非易事,不仅要避免组合过程中出现策略冲突,而且还需仔细考虑安全模块的组合顺序。对此,学者提出了安全服务模块组合时策略的自动化、无冲突快速组合的解决方案。PGA^[67]是一个高级网络策略(ACL、FW、LB 等)无冲突组合方案,实现了策略与底层网络的解耦,PGA 可根据安全服务构建需求,由输入图计算所有策略的并集,结合考虑组合约束以生成组合图。该方案在策略下发到底层设备进行部署前自动完成策略的无冲突组合,可减少系统运行时策略出现冲突(或错误)的几率。文献[68]提出安全策略的自动化实施模型,该模型定义了一种策略细化机制,用于将高级安全需求转换为网络安全功能的配置,通过定义的高级策略语言对于给定的目标网络安全控制制定安全策略(如包过滤、IDS 等),并由定义的中级策略语言对安全功能的配置(如规则、条件、操作等)进行抽象,从而使安全策略得以实施。但目前该方案不具实用性,只停留在理论验证阶段。

以上方案在安全服务链组合时均未考虑当前的网络状态和性能,容易造成资源浪费,还需要进一

步从性能优化的角度考虑安全服务组合的有效性。比如在设计安全服务链的同时,能结合考虑网络安全需求和 QoS(Quality of Service)参数^[69],以及将特定的安全级别和资源需求(CPU、带宽、存储)作为安全服务组合必须考虑的因素,将模型优化为求解最优安全服务组合的问题^[70]。同时,为确保安全服务组合的可靠性,还可以基于机会约束规划和服务备份策略来优化安全服务组合,以主动应对在安全服务组合中可能出现的安全服务需求变动及基础设施服务中断^[71]等。相比早期方案 FRESKO^[63],以上安全服务组合的优化方案达到的效果更优。

除了考虑优化安全服务组合之外,安全服务如何在虚拟网络防护中实现最优部署,以满足特定目标的安全防护需求,也是需要进一步解决的问题。文献[72-75]分别从不同的侧重点提出了解决方案。文献[72]提出操作成本最小化的安全服务部署算法,以最小化服务器内部交换机 CPU 的开销,但该方案未在真实的云环境中进行性能测试,如何运用在网络监控中还有待进一步研究。文献[73]以降低总体部署成本为目标,提出安全服务放置的优化方案,该方案通过贪心算法和线性规划舍入算法,在最优的对数因子内得到成本解,实现满足所需安全服务组合的约束下,总体部署成本最低的安全服务放置方案。文献[74]提出边缘网络安全服务部署的优化方案,该方案利用最优停止理论提出一个时间优化调度程序,在网络与用户需求动态变化时,重新部署安全服务到最佳位置,以减少安全服务与用户终端之间端到端的延迟。文献[75]提出的安全服务链编排器 Octans,以最大化安全服务链的总吞吐量优化安全服务的部署,该方案采用非线性整数规划模型来描述问题,以了解该优化问题中的关键因素,对服务器输入的安全服务部署请求,由 Octans 部署引擎中的启发式在线布局算法确定最优的部署方案。

以上解决方案快速重构了传统网络的安全防护方式,以网络安全需求为出发点,可对安全服务功能集中管理、按需组合和动态部署,缓解了虚拟网络边界防护失效的问题。但目前如何快速、灵活构建安全服务链,实现对虚拟网络边界动态变化的有效感知,还有待于进一步深入研究。表 3 对以上方案进行了总结分析。

4.2 信息泄露及篡改防护方案

拓扑泄露、数据拦截及篡改(ARP 攻击、嗅探攻击)为常见的网络安全问题,对该类安全问题的研究工作,在传统网络中业界已有大量详细的分析和总结^[76-80],且大多数防御方案^[81-84]同样满足于虚拟网

表 3 安全服务动态部署相关方案分析

Table 3 Analysis of relevant schemes for dynamic deployment of security services		
分类	方案	研究目标
安全服务模块划分及重组	FRESCO[63]	安全服务按需灵活组合
	AVANT-GUARD[64]	
	Wang Z[65]	
安全服务策略自动化组合	PBNM[66]	减少策略冲突
	PGA[67]	
	Basile C[68]	
安全服务编排优化	Hao Z[69]	QoS
	Liu Y[70]	提升资源利用率
	Chemodanov D[71]	QoS
安全服务部署优化	Luizelli M C[72]	成本最小化
	Tomassilli A[73]	延迟最小化
	Cziva R[74]	成本最小化
	Octans[75]	提升吞吐量

络环境的防护需求, 因此, 针对虚拟网络环境所提出的研究方案较少, 主要有以下解决方案: 基于虚拟网络嵌入的拓扑探测防护方案^[15,85]和基于虚拟网络设备安全漏洞提出的二层网络攻击防护方案^[16,86]。为此, 对于拓扑泄露、数据拦截及篡改相关问题的防护方案, 本节不再对其进行重复总结。重点对虚拟网络特有属性导致的数据泄露安全问题所提出的解决方案进行系统的归纳、分析和总结。

在虚拟化网络环境中, 虚拟化技术自身存在的安全漏洞及虚拟网络共享基础设施资源导致的同驻攻击, 是造成虚拟网络内部及虚拟网络之间数据泄露的重要因素。同驻攻击安全防护, 多数从虚拟化安全监控^[23,87-89]、hypervisor 安全保护^[90-94]、侧信道/隐蔽信道攻击防护^[20,95-98]等出发来减少数据泄漏, 该方式虽然有效, 但需要对现有云基础设施进行修改, 导致性能开销和部署成本增加, 并且不能确保针对当前未知侧通道攻击的防护, 为此, 研究者从虚拟网络构建的视角提出解决虚拟网络信息泄露的相关方案。虚拟网络由虚拟化资源层的虚拟节点和虚拟链路组成, 在构建虚拟网络的过程中, 重要的工作是有有效的虚拟资源分配和正确的虚拟网络隔离, 本节从该角度将业界提出的安全方案分为两类: 基于虚拟网络嵌入的防护(4.2.1)和基于虚拟网络隔离强化的防护(4.2.2), 并在表 4 对以上方案进行了对比分析。

4.2.1 基于虚拟网络嵌入的防护

虚拟网络嵌入(Virtual Network Embedding, VNE)综合考虑虚拟节点和虚拟链路的资源请求, 在一个虚拟网络请求到达时, 根据虚拟节点和链路上的相关约束, 将虚拟网络映射到底层网络中特定的物理

节点和链路上, 本质上是一个资源分配的问题。早期虚拟网络嵌入的研究工作大都集中在一般的嵌入问题上^[99-102], 未结合考虑虚拟网络的安全需求, 而网络虚拟化引入了易被攻击者利用的安全漏洞, 若虚拟网络嵌入到没有足够保护的底层资源时则易受攻击。为此, 有学者从虚拟网络嵌入安全的角度提出虚拟网络安全解决方案, 在考虑有效资源分配的同时, 需结合虚拟网络的安全约束, 进行虚拟节点和链接到底层网络的最佳映射, 以此增强虚拟网络的安全性。

文献[103]在进行虚拟资源分配时, 除了考虑节点 CPU 能力、链路带宽消耗等资源约束外, 还结合虚拟网络安全约束, 提出满足三类安全需求的虚拟网络嵌入模型, 允许每个虚拟网络请求都有一组与之相关的安全需求, 旨在从三种不同级别的保密性中(端到端加密、点对点加密和非重叠网络), 利用数据加密技术为虚拟网络通信提供其中一种保密性, 但该方案通过为虚拟网络请求分配独占的链路和节点资源(即不与其他虚拟网络共享物理资源), 这种保护高安全级别数据免受攻击的方式过于极端, 不具实用性。

在此基础上, 文献[104-105]提出抽象分类法对虚拟网络的安全需求和底层基础设施的安全级别进行建模, 引入虚拟网络安全级别的概念, 将虚拟节点或链路的安全需求以安全级别表示, 并分析底层基础设施可提供的相同或更高安全级别的物理资源, 在虚拟网络嵌入中遵循安全级别映射, 确保虚拟节点映射到具有相同或更高安全级别的底层节点, 但该方案提出的安全级别概念过于抽象, 缺乏详细的安全规范定义, 具有局限性。对此, 文献[106]提出在线虚拟网络嵌入的解决方案, 并支持详细的安全规范, 该解决方案结合租户需求考虑虚拟网络安全约束, 支持在多种云(公共云和私有云)中共享资源(节点/链接), 通过建立 SecVNE 模型, 求解为一个混合整数线性规划, 并提出了一种新的策略语言来指定底层网络的特性, 使得虚拟网络规范能够在多云的基础上映射, 实现将虚拟网络映射到考虑所有约束(与节点、边缘和云安全相关约束)的底层网络上, 并寻找最优嵌入, 弥补了上述方案的不足。文献[107]分析了上述安全感知虚拟网络嵌入的相关工作, 从缩短安全感知虚拟网络嵌入过程的运行时间, 对底层网络节点的重要性进行客观评价, 提出了一种混合整数线性规划模型 SA-VNE, 解决安全感知虚拟网络嵌入。该方案利用信息熵的 TPOSIS 方法对底层网络节点的重要性进行排序, 来为每个虚拟节点选

表 4 数据泄露相关防护方案分析

Table 4 Analysis of data leakage related protection schemes					
	类别	相关文献	描述	优点	不足
虚拟网络嵌入	避免拓扑探测	[15][85]	将虚拟网络安全约束与虚拟网络资源分配过程结合, 以对虚拟网络进行有效保护	提高了虚拟网络的安全性	降低了虚拟网络映射的成功率, 对网络服务性能影响较大
	基于安全感知	[103][104][105][106][107]			
	降低隐蔽信道攻击	[108][109]			
	可信虚拟域	[116]			
虚拟网络隔离强化	基于专用内核模块	[117]	在现有虚拟网络隔离机制的基础上提升隔离的安全性, 以避免隔离机制失效导致的数据泄露、拒绝服务攻击	增强了虚拟网络的隔离性能	未考虑虚拟网络隔离和共享(安全通信)之间的平衡
	基于代理软件	[118]			
	基于 SDN 技术	[119][120][121][122]			
	提升虚拟交换机性能	[123][124][125]			
	基于安全沙盒	[126]			
	结合 SR-IOV 技术	[127][128][129][130]			

择最适合的底层网络节点, 并利用最短路径算法执行链路映射。

以上方案仅用于抽象共享底层网络中安全机制的可用性, 还不足以涵盖虚拟节点之间隐蔽通道的攻击风险, 并增加了嵌入具有低隐蔽信道攻击的风险。在共享的底层网络中, 任何隐蔽通道攻击都涉及两个虚拟节点(一个发送一个接收), 两个虚拟节点协同操作, 其他虚拟节点扮演旁观者的角色, 而旁观者的工作量会干扰攻击者两个虚拟节点之间的数据传输^[108], 为此, 文献[109]通过嵌入适当的旁观者, 模拟虚拟节点的工作负载对隐蔽信道传输质量的影响, 提出风险容忍共存约束, 将虚拟网络嵌入作为一个优化问题, 实现降低隐蔽信道攻击的安全嵌入。

综上所述, 以上方案将考虑虚拟网络最佳映射与虚拟网络环境的安全性视为同等重要, 在一定程度上提高了虚拟网络的安全性, 但对虚拟网络安全的防护效果有限, 可将虚拟网络嵌入防护方案与其他防护方案结合使用。

4.2.2 基于虚拟网络隔离强化的防护

在虚拟网络创建中, 除了以上提到的有效虚拟资源分配外, 另一项重要的工作是提供正确的隔离。在同一物理主机内, 网络软件设备(如虚拟交换机)通过 VLAN 和隧道协议(如 VxLAN)管理虚拟网络的内部和外部流量, 提供网络流量和 IP 地址的隔离功能。虚拟网络之间的隔离对于确保每个虚拟网络的完整性必不可少, 尽管虚拟网络隔离机制旨在提供虚拟网络之间的正确隔离性, 但虚拟化技术自身存在的安全漏洞易被恶意实体利用, 导致数据泄露及拒绝服务攻击。文献[110]对现有的虚拟网络隔离机制的细节进行了分析与对比, 强调了它们各自的优缺点, 但作者未对其隔离机制是否面临隔离失效的威胁做出评估。对此, 文献[111-112]提出了对虚拟网络隔离

的审核机制, 并通过实验分析指出现有虚拟网络隔离机制(VLAN、VxLAN、GRE、NVGRE^[113]、FlowVisor^[114]、OpenFlow^[115]等)未把虚拟网络隔离的安全性作为其主要目标, 存在弱点, 无法直接抵御网络底层软件和硬件资源的攻击, 共享的底层网络组件一旦受到攻击, 会使隔离机制失效, 进而为攻击者进行网络嗅探、更改网络配置打开大门。因此, 增强虚拟网络之间的隔离性能, 可以最大限度的防止数据泄露威胁以及恶意租户发起的 DoS 攻击。

惠普实验室的一个小组最早从可信虚拟域(Trusted Virtual Domain, TVD)^[116]的定义入手, 提出安全的网络虚拟化框架来加强虚拟网络隔离安全, 该框架结合了现有的网络技术(如以太网封装、VLAN 标记、VPN)和安全策略, 通过使用可信虚拟域自动实例化和部署适当的安全机制来实现网络虚拟化的安全目标, 每个 TVD 代表一个独立的域, 利用 TVD 提供访问控制和可靠网络隔离, 保证虚拟网络信息的机密性和完整性。SilverLine 系统^[117]使用专用的内核模块提供隔离, 并添加到虚拟机监视器的特权客户操作系统中, 将该隔离与客户操作系统的修改结合起来, 实现隔离强化。

以上方案可以防止跨虚拟机攻击、配置错误或侧通道攻击造成的数据泄漏, 但均需要对管理程序进行侵入式修改, 且不易扩展, 而 BlueShield^[118]体系结构避免了对管理程序进行修改, 采用类似 VLAN 的隔离特性, 但不依赖于 hypervisor, 通过 BlueShield 代理实现虚拟网络的高度防篡改隔离。但该隔离机制中代理软件一旦被破坏, 攻击者可能会获得所在主机服务器的特权访问, 导致恶意 DoS 攻击。

随着软件定义网络(Software Defined Network, SDN)的出现, 为虚拟网络隔离强化提出了新的解决

思路, 可利用 SDN 特定的流转发规则结合部署的安全策略实现对虚拟网络的隔离强化^[119-120]。基于 SDN 技术的隔离机制, 具有细粒度流量控制的优势, 但 SDN 架构本身存在的安全问题可能会导致隔离失效。对此, 研究者尝试针对网络虚拟化架构平台中的安全漏洞提出解决方案, 确保虚拟网络隔离安全。文献[121]提出对 FlowVisor 的修改, 使网络管理员可以定义允许每个控制器在其流中使用哪些操作, 来限制每个虚拟网络控制器可以使用哪种类型的动作, 从而抑制恶意攻击破坏隔离而操纵其他虚拟网络的流量。对于 OpenFlow 协议^[60]因缺乏强制的安全机制, 易受中间人攻击, 文献[122]采用切片的形式化语义, 确保切片上的数据包处理独立于其他切片, 切片编译为 OpenFlow 交换机的算法, 由编译器处理与实现隔离相关的所有繁琐细节, 确保切片的隔离性。

除了以上方案之外, 通过增强虚拟交换机安全^[123-125]、虚拟机隔离安全^[126]以及结合 SR-IOV 技术来提升虚拟网络的隔离性能, 也是强化虚拟网络隔离的主要解决思路, 如 Vagabond^[127]系统和 HYVI^[128]体系结构, 基于 SR-IOV 技术, 允许对多个虚拟网络执行隔离策略, 以此提高虚拟网络隔离性。但该隔离机制未对虚拟网络中的流量进行封装, 易受恶意租户攻击。为此, 文献[129-130]将 SR-IOV 技术与现有虚拟网络隔离机制结合, 提出涵盖数据路径隔离、软件资源隔离和硬件资源隔离的三种隔离策略的隔离方案, 并引入了租户网络域(TND)的概念, 基于网络功能虚拟化(NFV)和硬件辅助虚拟化(HAV)技术来增强虚拟网络的隔离性能, 使虚拟网络隔离防护更全面、安全。

虽然虚拟网络之间的正确隔离是确保虚拟网络完整性的重要属性, 但在许多情况下, 虚拟网络之间可能还需要彼此共享服务, 完全的隔离通常是不希望的, 对此, 在虚拟网络隔离解决方案中, 一方面需要在提供严格隔离的同时还能让虚拟网络之间实现可控的安全通信; 另一方面, 虚拟网络隔离还需要考虑灵活性, 隔离边界能随攻击及时做出调整, 以防止恶意攻击利用虚拟网络扩大攻击范围, 可基于微隔离, 实现对虚拟网络细粒度、灵活的隔离管理。

4.3 “东西”向流量监控方案

“东西”向流量安全是虚拟网络的特有属性, 物理服务器内部虚拟机之间的通信流量存在监控盲点(见 3.3 节)。要解决对内部流量的可控可见, 首先可基于流量细粒度划分和分布式协同监控, 实现对流量的深度监控。其次可根据安全需求, 实现对流量的

动态控制, 使其经过相应的安全防护设备。为此, 本小节依次从深度流量监控(4.3.1)和流量动态控制(4.3.2)两方面对现有解决方案进行讨论。

4.3.1 深度流量监控

网络流量监控是防御攻击的有效手段, 但由于虚拟资源的分散性、虚拟机的动态迁移、以及内部流量的不可见等特性, 使得传统流量监控方式无法应对云内虚拟网络流量的管控^[131]。虽然有研究者从监控覆盖范围^[132-133]及网络动态变化的角度^[134]对传统流量监控方案进行了优化, 但仍无法适用于虚拟网络。特别是对于许多重要的安全应用(如入侵检测^[135]、大流量识别^[136]、异常检测^[137]、DDoS 攻击检测^[138])均依赖于细粒度的流量监控, 而传统网络中基于分组采样的监控工具 NetFlow^[139]和 sFlow^[140]虽然为不同的监控任务提供了通用支持, 但以给定的概率选择分组易低估小数据流, 难以满足应用程序对不同流量执行不同监控的需求, 并且在实现细粒度流量监控时会消耗大量资源(网络带宽、CPU、和内存)。

在虚拟网络环境中, 结合 SDN 技术可解决传统网络安全防护在虚拟化环境中无法避免的安全问题^[141], 为实现虚拟网络细粒度流量监控提供了解决思路, SDN 控制器具有全局网络实时的流量信息, 更易对网络状态和流量进行监控和管理, 并可对网络泛洪或网络异常高效检测。特别是基于 SDN 技术能通过 OpenFlow 交换机流量计数器获得端到端流量, 为此, 受 OpenFlow 的启发, 文献[142]提出了通用的流量测量框架 OpenSketch, OpenSketch 可根据不同检测任务的精度需求, 实现细粒度的流量测量。该架构将测量控制与数据层分离, 在数据层中, OpenSketch 提供了一个三阶段流水线: Hash 运算、流分类和流统计, 控制层可根据不同任务的准确性需求对三个阶段进行灵活组合, 该方案在实现细粒度流量测量的过程中开销小, 且准确性高, 但 OpenSketch 在部署时需要升级所有网络节点, 成本过高。

对此, 文献[143]提出一个细粒度监控系统 OpenNetMon, 该系统作为 POX 控制器的内部模块, 以自适应的速率轮询交换机获取流量统计信息, 持续的对所监控网络的吞吐率、延迟和丢包率进行测量, 并为安全应用程序(如 IDS、IPS 等)提供流量统计信息。SDN-Mon^[144]对流量监控框架进行了修改, 使用多张表, 将流量监控功能从控制器中分离出来, 从而允许控制器根据应用程序的要求定义任意的监控匹配字段集, 以更细粒度更灵活地监控流量。然而, SDN-Mon 采用 Lagopus 软件交换机, 该方案仅适用

于该交换机。文献[145]提出一种端到端的流量恢复模型,该模型基于分形插值、三次样条插值和加权几何平均算法,实现从粗粒度采样的流量数据中获取细粒度准确的流量估计。

除了需要细粒度的流量测量外,在虚拟网络中采用传统的交换机(或路由器)独立监控流量的方式,不仅会消耗大量网络资源,还存在有些流量交换机无法覆盖的问题,为此,文献[146-148]提出了分布式协作监控方案,将对目标流量的监测任务分配到整个网络,以减少交换机上存储的规则和数据包处理开销。cSamp^[146]使用数据包的五元组哈希值在交换机之间分配采样负载,要求每个交换机维护一个辅助表以支持流分布,每个数据包进出路由器(或边缘交换机)都携带一个条目,在实际云环境中,这意味着数百万个表条目,查找该表会导致每个数据包

额外的处理开销,在实际中不可用。DCM 分布式协同监测系统^[147],通过SDN控制器在数据平面上部署两级 Bloom 过滤器,第一级过滤掉不需要监测的流量,第二级对第一级中存放的需监测的流量进行检查。这种方式节省了交换机的处理资源,但 Bloom 过滤器的查找需要进行内存访问和哈希操作,处理开销较大。为此,文献[148]提出一种轻量级流量分配方案,该方案将每个交换机上的内存开销最小化为单个采样概率值,将流量分配描述为数据包被前路由路径上的交换机记录和未被记录的优化问题,分别通过边缘交换机与边缘或核心交换机实现协作的流量监测。

以上深度流量监控方案可用于虚拟网络安全应用和其他流量工程中,以增强对虚拟网络流量的管控,从而提升虚拟网络的安全性。表 5 对以上方案进行了对比分析。

表 5 深度流量监控相关方案分析

Table 5 Analysis of deep traffic monitoring related schemes

相关方案	研究侧重点	测量类型	原理	不足
OpenSketch [142]	细粒度	主动	设计了可动态配置的三阶段流水线,可提供通用的细粒度流量测量	架构可用性受限,且成本较高
OpenNetMon[143]	细粒度	主动	自适应轮询交换机,对网络的吞吐率、丢包和延迟率进行测量	对于高速网络,流量监控粒度受限,未考虑安全监控
SDN-mon [144]	细粒度	主动	监控与转发表解耦,使控制器能在交换机中插入更多带有通配符的通用转发规则	需要检查每个新的流条目,以确定监测流
Jiang D [145]	细粒度	被动	从采样的粗粒度流量中获取细粒度网络流量	算法较复杂,难以准确获取细粒度流量
cSamp [146]	分布式	被动	利用哈希值在交换机之间分配流量	实际网络中难以部署
DCM [147]	分布式	主动	使用两级过滤方法监测流量	流量分配的处理开销较大
Xu H [148]	分布式	被动	设计了协作式流量监控方案	缺乏网络性能监控

4.3.2 流量动态控制

为了使虚拟网络内部流量经过部署在服务器外部的硬件安全设备或服务器内部的虚拟化安全设备实现检测(详见 4.1 节),均需要对流量进行动态控制。在此,分别对传统流量控制方案和软件定义的流量控制方案进行讨论。

(1) 传统流量动态控制

一些安全厂商结合自身优势提出了流量控制方案。对宿主机内同一虚拟网络的虚拟机之间需要监控的流量,VMware 公司提出通过 hypervisor API 接口的流量控制方案,牵引流量到同宿主机内部署的安全设备虚拟机进行分析(图 5 所示),但该方式受虚拟化系统 API 开放程度的限制。

厂商方案中最易实现的是通过对宿主机内虚拟交换机配置端口镜像策略实现流量控制(图 6 所示),如 VMware^[149]基于 VMware vSphere 平台虚拟交换机(vSphere distributed switch, VDS)的端口镜像技术,

将所需检测的虚拟机流量通过交换机端口牵引到同宿主机内的安全设备虚拟机或引出到外部物理交换机旁挂的安全设备进行分析,这种方案需要对虚拟交换机进行配置,随“东西”向流量的增加,会影响虚拟交换机的性能。

此外, Gigamon 和 Fortinet 公司分别推出了基于代理的流量牵引方案(图 7 所示),Gigamon^[150]通过在宿主机内部署代理虚拟机(GigaVUE-VM)来复制虚

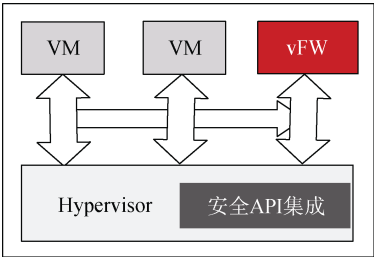


图 5 基于 API 接口引流

Figure 5 Traffic steering based on API interface

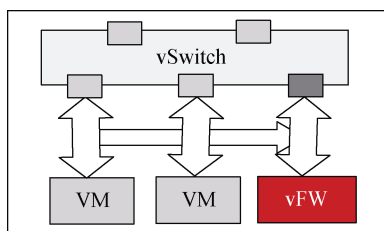


图 6 基于端口镜像引流

Figure 6 Traffic steering based on port mirroring

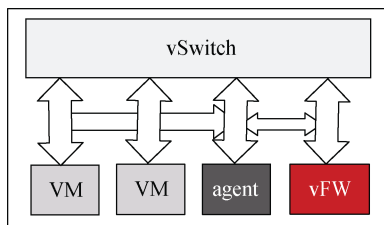


图 7 基于代理引流

Figure 7 Traffic steering based on agent

拟机中的流量进行安全检测,或者引流到安全设备进行检测,这种方式需要对虚拟网络的相关配置进行修改,且可移植性较差。Fortinet^[151]通过在每个虚拟机内安装代理,由代理对数据包修改或标记完成流量转发控制,该方案需安装在每个虚拟机中,涉及的安装规模较大,且需要获取客户虚拟机的 root 权限。Veryx 和 IXIA 基于 vTAP 也分别提出了 vTAP 插件^[152]、基于内核的 vTAP^[153]流控制方式等。

可见,以上传统流量控制方案存在局限性,随着 SDN 技术在云环境的部署应用,可基于 SDN 实现流量控制。

(2) 软件定义的流量动态控制

SDN 控制器的全局网络可视性及可编程性,对虚拟网络中需要检测的“东西”向流量,可基于 SDN 控制器对交换机下发流表项,将流量牵引到安全设备。基于 SDN 的方案不需要考虑安全设备的部署位置,且相比其他流量控制方式具有全局、灵活性等特点。云杉网络、VMware、绿盟等厂商分别推出了基于 SDN 的流量牵引方案,如 DeepFlow^[154]、VMware NSX^[155]和 NCSS^[156],以上流量控制方案由于基于各厂商的专有性,对接实现易产生依赖,且用户在维护和可伸缩性方面存在困难。为此,近年学术界重点基于 SDN 技术提出虚拟网络流量动态控制方案。

SDN 作为网络功能虚拟化(Network function virtualization, NFV)实施安全策略的重要补充,可利用 SDN 的集中可控性,基于控制器在网络单元中安装细粒度转发规则,引导可疑流量到由安全虚拟功能组成的服务功能链来实施安全策略。CloudWatcher^[157]

网络安全防护架构方案,基于 SDN 技术控制流的路由路径,对符合流规则的流量强制转发到相应的安全设备所在的网络节点进行检测,提出了 4 种不同的路由算法以优化流量通过特定安全设备的路由路径,并对虚拟机迁移提供持续安全的监视,但随着网络中流量的增加,该方案的性能会下降。SIMPLE 的策略执行层^[158]利用 SDN 进行流量控制,流量转发基于标签和交换机之间的隧道,并通过使用紧凑的转发表来减少转发状态。

在牵引流量经过特定的服务功能链(或中间盒序列)时,存在中间盒修改数据包头或数据包内容的情况,会导致后续中间盒对流量执行安全策略时出现错误判断。虽然 SIMPLE^[158]提出了采用载荷相似度匹配的方法,对流量进行溯源,但这种解决方案会消耗较多的计算资源。为此,FlowTags^[159]对中间盒进行扩展,提出标签法,通过对被修改的流量打标签,以提供与流量相关联的上下文信息,使得后续中间盒可以根据标签来确定要执行的安全策略,但该方案需要修改每个中间盒带来了额外开销。

文献[160]采用开源代码工具设计流量控制方案,基于 L2 的 OpenFlow 规则进行流量控制,采用广域网基础设施连接管理器(WICM)作为流量管理的组件,并使用 VLAN ID 标记用户流量,由 Netfloc 在重写 MAC 地址的基础上引导流量通过服务功能链。文献[161]指出对牵引特定流量到服务功能链存在环路的情况,OpenFlow 协议无法做出准确的流量转发决策,虽然 SIMPLE^[158]给出了解决该问题的方案,但没有具体的实现细节。对此,作者提出在交换机中安装转发规则之前,检测交换机物理序列是否存在循环链路的算法,并在确定存在循环链路时决定由哪些交换机负责添加 VLAN Tag,通过 VLAN Tag 中的 VLAN ID 来标注每个包的处理状态以实现准确引导流量通过服务功能链,但以上方案依赖 VLAN ID 标识用户流量,对用户数量有所限制。

以上控制流量经过服务功能链的方案,依赖于控制器下发的转发规则,一方面控制器过载易成为性能瓶颈,另一方面未考虑流量牵引到服务功能链时,对服务功能链动态变化的响应。Zave P^[162]认为基于会话层协议控制流量经过服务功能链的方法,可很好的应对服务功能链的动态变化。对此,作者对 TCP 进一步扩展提出 Dysco 会话协议,通过对虚拟安全功能增加 Dysco 代理守护进程,在服务功能链动态调整时,重新调度锚节点之间的流量。该方案实现相比以上方案不受特定网络环境的限制,但协议本身会增加协议栈处理时延开销。

文献[163-164]在动态控制流量的同时考虑了流量通过服务功能链的性能。OpenNF^[163]作为一种基于SDN 控制器设计的服务功能控制框架,通过一组API 导出和导入虚拟安全功能的状态,可对虚拟安全功能状态和网络转发状态实现协调控制,当虚拟安全功能状态更新时,在服务功能链之间实现流量的重新分配,确保流量经过中间盒的准确性。Slick^[164]将流量牵引和服务功能链部署结合考虑,流量牵引不基于预先部署的固定中间盒,可根据特定的流量协调中间盒的放置数量及位置,考虑提升网络带宽利用率的同时,确保流量牵引到服务功能链时良好的端到端性能。

文献[165-167]结合网络性能,提出对流量控制的路由优化方案。文献[165]提出在满足网络资源(吞吐量、中间盒负载、交换机内存等)及路由(特定服务功能链)约束的前提下,基于多点到点的节点树

(MultiPoint-To-Point Trees, MPTPT)路由算法,实现控制流量到相应中间盒的路由优化。文献[166]进一步提出在满足服务质量(QoS)要求的同时,实现控制流量到服务功能链的离线路由(基于流量整合算法)和在线路由(基于原始-对偶改进算法)的优化方案。VNFP-TS^[167]提出对流量控制的优化以提升网络性能和降低资源消耗,该方案将中间盒布局和流量控制优化的问题作为一个二进制整数规划(Binary Integer Programming, BIP)模型,提出一种基于动态规划的成本优化算法,以最小化网络总成本为目标,来求解中间盒的最优流量控制。

上述流量动态控制的防护方案可归结为 3 类:第 1 类,基于转发规则或会话协议控制流量经过安全设备的实现;第 2 类,流量牵引与中间盒的协调控制。第 3 类,流量经过中间盒的最优控制方案。表 6 对以上方案进行了对比分析。

表 6 流量动态控制相关方案分析
Table 6 Analysis of traffic dynamic control related schemes

相关文献	优点	不足
CloudWatcher[157]	提出的策略脚本语言易于用户控制流量通过特定的网络节点	对指定的不能相互通信的安全设备,算法无法生成路由路径
SIMPLE[158]	考虑了中间盒的负载均衡	流量溯源易消耗计算资源
FlowTags[159]	避免了数据包上下文信息的丢失	需对中间盒进行修改,带来额外开销。
Trajkovska I[160]	考虑了数据路径冗余	基于 VLAN ID 标识流量,规模受限
He Q[161]	避免了路由存在环路时流量的错误转发	基于 VLAN ID 标识流量,规模受限
Dysco[162]	流量控制不受特定网络环境的限制	会话协议易受攻击
OpenNF[163]	确保了流量经过服务功能链时的状态一致性	为适应 OpenNF 原语,需要修改网络功能(NF)
Slick[164]	提升了网络带宽的利用率	随中间盒数量的增加,流量控制算法运行时 间过长
Walid A[165][166]	结合考虑了网络性能	计算成本较高
VNFP-TS[167]	有效降低了网络成本	计算成本较高

5 研究展望

通过上述分析可知,目前虚拟网络防护已有不少解决方案,在包括虚拟防火墙、安全服务动态部署、虚拟网络嵌入安全、虚拟网络隔离强化、深度流量监控、流量动态控制等领域均取得了不错的防护效果,但目前的防护方案并不能完全满足虚拟网络防护需求,虽然利用软件定义网络技术可以较好地解决传统安全防护技术无法解决的安全问题,但软件定义网络架构本身存在的安全问题易被攻击者利用实施攻击,因此,对虚拟网络的防护还需进一步深入研究,以建立一套安全、高效、智能的虚拟网络防御体系架构。结合虚拟网络的研究趋势,笔者认为未来虚拟网络安全防护可重点关注以下两方面:

第一、构建动态防御技术体系。可根据虚拟网络动态特性有机结合多种安全机制,从虚拟网络边界防护、虚拟网络内部流量可控可见到底层虚拟化资源(虚拟节点和虚拟链路)防护三个方面,实现协同、高效和灵活的安全防护。着重研究安全服务按需动态部署、边界访问控制、深度流量监控及虚拟网络隔离强化,构建多层次、可靠的虚拟网络动态防御体系框架。

第二、提升软件定义网络架构自身的安全性。利用软件定义网络技术解决虚拟网络安全问题,不可避免因软件定义网络自身存在的安全漏洞被利用而导致攻击,如 OpenFlow 协议未强制 TLS 协议的双向认证技术、控制器劫持、流表溢出及篡改、以及软件定义架构中数据平面与控制平面分离导致南北

向接口存在安全隐患等, 因此, 还需要融合软件定义网络的自身安全机制, 从提升软件定义网络应用安全、北向应用接入安全、控制器安全、南向设备接入安全到底层转发设备的安全, 以强化软件定义网络自身的安全性。

笔者目前重点研究了虚拟网络动态防御体系的构建, 将 SDN、NFV 技术与云平台相结合, 通过对开源的 SDN 控制器平台进行开发, 提出一种基于软件定义边界的动态防御框架, 该框架如图 8 所示, 整个框架包括相互关联的 3 层: 安全管理层、安全控制

层和安全数据层。

(1) 安全管理层

安全管理层实现对网络策略、流量检测、安全服务编排及动态接入的统一管理。通过对云虚拟网络变化的感知, 随虚拟网络安全防护需求生成安全策略、编排安全服务并动态接入, 以实现安全运维的自动化。安全管理层将安全控制层获取的全局网络设备流统计信息用于流检测分析, 并基于虚拟机迁移和安全服务链反馈的监测结果及时进行网络策略的管控(分析、更新与下发)。

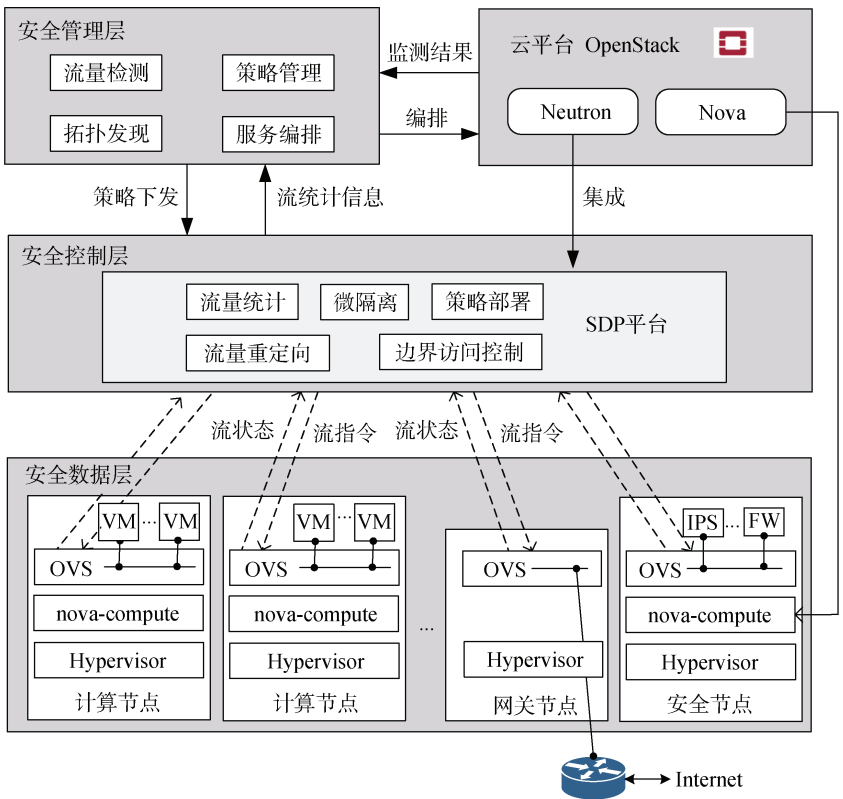


图 8 SDP 动态防御框架
Figure 8 SDP dynamic defense framework

(2) 安全控制层

安全控制层作为整个动态防御体系的核心, 集成了一套虚拟网络防护服务模块, 包括对流信息的统计服务、流量的重定向服务、虚拟网络隔离边界动态调整的“微隔离”服务、安全策略在指定位置正确部署的服务等。安全控制层的工作过程如下: SDP 控制器从安全数据层的网络设备中获取流信息, 进行统计后交给上层安全管理层, 当安全管理层的流量检测模块发现异常流量并发出安全警告时, SDP 控制器接收安全管理层发来的流量牵引指令, 将需要进一步检测的数据流经流量重定向服务计算出的重定向路径, 牵引到安全管理层编排的服务链, 完

成进一步监测, 若发现攻击行为, 则通过“微隔离”服务和安全策略部署服务, 将安全管理层发来的安全策略正确部署到安全数据层的隔离控制点, 以调整虚拟网络隔离边界, 缩小攻击范围。同时, 由虚拟边界访问控制服务决定任意虚拟机之间的访问控制权限, 建立基于零信任概念的边界访问控制, 以防止因虚拟机迁移、攻击者由“跳板”绕过安全防护设备而导致的虚拟边界隔离失效。

(3) 安全数据层

安全数据层重点提升所分配虚拟网络资源的安全性及虚拟网络的隔离性。首先, 确保底层物理资源安全性的同时对其划分安全级别, 在对虚拟网络分

配资源时, 结合考虑底层共享资源的安全性, 将虚拟网络映射到相同或更高安全级别的物理资源中, 避免与攻击者同驻, 同时降低拓扑探测风险; 其次, 基于 SDP 控制器实现对虚拟网络的隔离强化, 降低数据泄露风险。

6 结束语

云虚拟网络既面临传统网络固有的攻击威胁, 还引入了伴随云计算技术所产生的新的安全威胁, 其安全问题受到工业界和学术界的广泛关注, 近年已取得了一些研究成果, 但仍需研究面向云计算原生特征的体系化安全解决方案。因此, 本文提出了基于软件定义边界的动态防御框架。

下一步我们将在基于软件定义边界动态防御框架的实现基础上, 重点围绕软件定义安全策略的动态生成机制开展研究, 最终实现安全、高效、智能的云虚拟网络动态防御。

致谢 感谢各位老师和同门对本文工作提出的指导建议, 同时感谢评审专家和编辑部老师给予的宝贵修改建议。

参考文献

- [1] Chowdhury N M M K, Boutaba R. A Survey of Network Virtualization[J]. *Computer Networks*, 2010, 54(5): 862-876.
- [2] Wang G H, Eugene Ng T S. The Impact of Virtualization on Network Performance of Amazon EC2 Data Center[C]. *2010 Proceedings IEEE INFOCOM*, 2010: 1-9.
- [3] AlZain M A, Soh B, Pardede E. A Survey on Data Security Issues in Cloud Computing: From Single to Multi-Clouds[J]. *Journal of Software*, 2013, 8(5): 1068-1078.
- [4] Ristenpart T, Tromer E, Shacham H, et al. Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds[C]. *The 16th ACM Conference on Computer and Communications Security*, 2009: 199-212.
- [5] Chen S, Wang R, Wang X F, et al. Side-Channel Leaks in Web Applications: A Reality Today, a Challenge tomorrow[C]. *2010 IEEE Symposium on Security and Privacy*, 2010: 191-206.
- [6] Zhang T W, Zhang Y Q, Lee R B, et al. DoS Attacks on Your Memory in Cloud[C]. *The 2017 ACM on Asia Conference on Computer and Communications Security*, 2017: 253-265.
- [7] Zargar S T, Joshi J, Tipper D. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks[J]. *IEEE Communications Surveys & Tutorials*, 2013, 15(4): 2046-2069.
- [8] Somani G, Gaur M S, Sanghi D, et al. DDoS Attacks in Cloud Computing: Issues, Taxonomy, and Future Directions[J]. *Computer Communications*, 2017, 107: 30-48.
- [9] Carapinha J, Feil P, Weissmann P, et al. Network virtualization opportunities and challenges for operators[C]. *Future Internet Symposium*, 2010: 138-147.
- [10] Anderson T, Peterson L, Shenker S, et al. Overcoming the Internet Impasse through Virtualization[J]. *Computer*, 2005, 38(4): 34-41.
- [11] Chen Z, Dong W Y, Li H, et al. Collaborative Network Security in Multi-Tenant Data Center for Cloud Computing[J]. *Tsinghua Science and Technology*, 2014, 19(1): 82-94.
- [12] Priebe C, Muthukumaran D, O'Keeffe D, et al. CloudSafetyNet[C]. *The 6th Edition of the ACM Workshop on Cloud Computing Security*, 2014: 117-128.
- [13] Prakash C, Lee J, Turner Y, et al. Pga[C]. *The 2015 ACM Conference on Special Interest Group on Data Communication*, 2015: 29-42.
- [14] Zhou W X, Croft J, Liu B Z, et al. Automatically Correcting Networks with NEAT[C]. *The 15th USENIX Conference on Networked Systems Design and Implementation*, 2018: 595-608.
- [15] Pignolet Y A, Schmid S, Tredan G. Adversarial VNet Embeddings: A Threat for ISPs?[C]. *2013 Proceedings IEEE INFOCOM*, 2013: 415-419.
- [16] Grover J, Shikha, Sharma M. Cloud Computing and Its Security Issues-A Review[C]. *Fifth International Conference on Computing, Communications and Networking Technologies*, 2014: 1-5.
- [17] Liberman Garcia A. The evolution of the Cloud: the work, progress and outlook of cloud infrastructure[D]. Massachusetts Institute of Technology, 2015.
- [18] Ren K, Wang C, Wang Q. Security Challenges for the Public Cloud[J]. *IEEE Internet Computing*, 2012, 16(1): 69-73.
- [19] Yarom Y and Falkner K. Flush+reload: A high resolution, low noise, L3 cache side-channel attack[C]. *In 23rd USENIX Security Symposium*, 2014: 719-732.
- [20] Zhang Y Q, Juels A, Reiter M K, et al. Cross-VM Side Channels and Their Use to Extract Private Keys[C]. *The 2012 ACM Conference on Computer and Communications Security*, 2012: 305-316.
- [21] Wu H Q, Ding Y, Winer C, et al. Network Security for Virtual Machine in Cloud Computing[C]. *5th International Conference on Computer Sciences and Convergence Information Technology*, 2010: 18-21.
- [22] Shetty J, M R A, Shobha G. A Survey on Techniques of Secure Live Migration of Virtual Machine[J]. *International Journal of Computer Applications*, 2012, 39(12): 34-39.
- [23] Fernandes N C, Duarte O C M B. XNetMon: A Network Monitor for Securing Virtual Networks[C]. *2011 IEEE International Conference on Communications*, 2011: 1-5.
- [24] Oliveira R R, Marcon D S, Bays L R, et al. No More Backups: Toward Efficient Embedding of Survivable Virtual Networks[C]. *2013 IEEE International Conference on Communications*, 2013: 2128-2132.
- [25] Yeow W L, Westphal C, Kozat U, et al. Designing and Embedding

- Reliable Virtual Infrastructures[C]. *The Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, 2010: 33-40.
- [26] Gupta B B, Badve O P. Taxonomy of DoS and DDoS Attacks and Desirable Defense Mechanism in a Cloud Computing Environment[J]. *Neural Computing and Applications*, 2017, 28(12): 3655-3682.
- [27] Dong S, Abbas K, Jain R. A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments[J]. *IEEE Access*, 2019, 7: 80813-80828.
- [28] IEEE 802.1Qxx. Data center bridging. <http://www.ieee802.org/1/pages/802.1az.html>. May. 2014.
- [29] Secure virtual network configuration for virtual machine (vm) protection. <http://dx.doi.org/10.6028/NIST.SP.800-125B>. Mar. 2016.
- [30] vShield 5.0. https://www.vmware.com/content/dam/digitalmarketing/vmware/zh-cn/pdf/vshield_50_quickstart-PG-CN.pdf. Spt. 2011.
- [31] Yu S, Doss R, Zhou W L, et al. A General Cloud Firewall Framework with Dynamic Resource Allocation[C]. *2013 IEEE International Conference on Communications*, 2013: 1941-1945.
- [32] Liu M, Dou W C, Yu S, et al. A Clusterized Firewall Framework for Cloud Computing[C]. *2014 IEEE International Conference on Communications*, 2014: 3788-3793.
- [33] Salah K, Callyam P, Boutaba R. Analytical Model for Elastic Scaling of Cloud-Based Firewalls[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(1): 136-146.
- [34] Dezhabad N, Sharifian S. Learning-Based Dynamic Scalable Load-Balanced Firewall as a Service in Network Function-Virtualized Cloud Computing Environments[J]. *The Journal of Supercomputing*, 2018, 74(7): 3329-3358.
- [35] Security challenges in sdn (software-defined networks). <https://www.sdncentral.com/security-challenges-sdn-software-defined-networks>. Oct. 2014.
- [36] Suh M, Park S H, Lee B, et al. Building Firewall over the Software-Defined Network Controller[C]. *16th International Conference on Advanced Communication Technology*, 2014: 744-748.
- [37] Pena J G V, Yu W E. Development of a Distributed Firewall Using Software Defined Networking Technology[C]. *2014 4th IEEE International Conference on Information Science and Technology*, 2014: 449-452.
- [38] Tran T V, Ahn H. A Network Topology-Aware Selectively Distributed Firewall Control in SDN[C]. *2015 International Conference on Information and Communication Technology Convergence*, 2015: 89-94.
- [39] Porras P, Shin S, Yegneswaran V, et al. A Security Enforcement Kernel for OpenFlow Networks[C]. *The First Workshop on Hot Topics in Software Defined Networks*, 2012: 121-126.
- [40] Khurshid A, Zou X, Zhou W, et al. Veriflow: Verifying network-wide invariants in real time[C]. *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation*, 2013: 15-27.
- [41] Kazemian P, Chang M, Zeng H Y, et al. Real Time Network Policy Checking Using Header Space Analysis[C]. *The 10th USENIX Conference on Networked Systems Design and Implementation*, 2013: 99-112.
- [42] Wang J, Wang Y, Hu H, et al. Towards a security-enhanced firewall application for openflow networks[M]. Springer, 2013: 92-103.
- [43] Kazemian P, Varghese G, McKeown N. Header space analysis: Static checking for networks[C]. *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation*. 2012: 113-126.
- [44] Hu H X, Han W, Ahn G J, et al. FLOWGUARD: Building Robust Firewalls for Software-Defined Networks[C]. *The Third Workshop on Hot Topics in Software Defined Networking*, 2014: 97-102.
- [45] Wang J, Wang J, Jiao H Y, et al. A Method of OpenFlow-Based Real-Time Conflict Detection and Resolution for SDN Access Control Policies[J]. *Chinese Journal of Computers*, 2015, 38(4): 872-883.
(王鹏, 王江, 焦虹阳, 等. 一种基于 OpenFlow 的 SDN 访问控制策略实时冲突检测与解决方法[J]. *计算机学报*, 2015, 38(4): 872-883.)
- [46] Gao S, Li Z C, Xiao B, et al. Security Threats in the Data Plane of Software-Defined Networks[J]. *IEEE Network*, 2018, 32(4): 108-113.
- [47] Zerkane S, Espes D, Le Parc P, et al. Software Defined Networking Reactive Stateful Firewall[C]. *ICT Systems Security and Privacy Protection*, 2016: 119-132.
- [48] Tran T V, Ahn H. Challenges of and Solution to the Control Load of Stateful Firewall in Software Defined Networks[J]. *Computer Standards & Interfaces*, 2017, 54: 293-304.
- [49] Nife F, Kotulski Z. Multi-Level Stateful Firewall Mechanism for Software Defined Networks[C]. *Computer Networks*, 2017: 271-286.
- [50] Chowdhary A, Huang D J, Alshamrani A, et al. SDFW: SDN-Based Stateful Distributed Firewall[EB/OL]. 2018: 1811.00634. <https://arxiv.org/abs/1811.00634v1>.
- [51] Caprolu M, Raponi S, Di Pietro R. FORTRESS: An Efficient and Distributed Firewall for Stateful Data Plane SDN[J]. *Security and Communication Networks*, 2019, 2019: 6874592.
- [52] Cheng Q M, Wu C M, Zhou H F, et al. Guarding the Perimeter of Cloud-Based Enterprise Networks: An Intelligent SDN Firewall[C]. *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems*, 2018: 897-902.
- [53] Gao S, Li Z C, Yao Y, et al. Software-Defined Firewall: Enabling Malware Traffic Detection and Programmable Security Control[C]. *The 2018 on Asia Conference on Computer and Communications Security*, 2018: 413-424.
- [54] Ward R, Beyer B. Beyondcorp: a new approach to enterprise security[J]. *The magazine of USENIX & SAGE*, 2014, 39(6): 6-11.
- [55] Osborn B, McWilliams J, Beyer B, et al. BeyondCorp: Design to deployment at Google[J]. *The magazine of USENIX & SAGE*, 2016,

- 41(1): 28-34.
- [56] Spear B, Cittadini L, Saltonstall M. BeyondCorp: The Access Proxy[J]. *The magazine of USENIX & SAGE*, 2016,41(4): 28-33.
- [57] Beske C M C, Peck J, Saltonstall M. Migrating to BeyondCorp: maintaining productivity while improving security[J]. *The magazine of USENIX & SAGE*, 2017, 42(2): 49-55.
- [58] Skyport Systems and The Zero Trust DC. <http://movingpackets.net/2016/03/31/skyport-systems-and-the-zero-trust-dc>. March. 2016.
- [59] Check Point Software-Defined Protection. <https://www.checkpoint.com/press/2014/check-point-introduces-software-defined-protection-security-architecture/>. Feb. 2014.
- [60] Cloud Security Alliance. <https://cloudsecurityalliance.org/research/working-groups/software-defined-perimeter/>. July. 2020.
- [61] NSX Micro-segmentation. <https://networkinfern.net/micro-segmentation-and-nsx>. Dec. 2014.
- [62] Catbird 6.0: Private Cloud Security. <https://docplayer.net/2298462-Catbird-6-0-private-cloud-security.html>. July. 2014.
- [63] Shin S W, Porras P, Yegneswara V, et al. Fresco: Modular composable security services for software-defined networks[C]. *20th Annual Network & Distributed System Security Symposium*, 2013:1-16.
- [64] Shin S, Yegneswaran V, Porras P, et al. AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks[C]. *The 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS'13*, 2013: 413-424.
- [65] Wang Z J, Tao D, Lin Z W. Dynamic Virtualization Security Service Construction Strategy for Software Defined Networks[C]. *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks*, 2016: 139-144.
- [66] Scheid E J, Machado C C, dos Santos R L, et al. Policy-Based Dynamic Service Chaining in Network Functions Virtualization[C]. *2016 IEEE Symposium on Computers and Communication*, 2016: 340-345.
- [67] Prakash C, Lee J, Turner Y, et al. Pga[C]. *The 2015 ACM Conference on Special Interest Group on Data Communication*, 2015: 29-42.
- [68] Basile C, Valenza F, Lioy A, et al. Adding Support for Automatic Enforcement of Security Policies in NFV Networks[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(2): 707-720.
- [69] Hao Z, Lin Z W, Li R, et al. A SDN/NFV Security Protection Architecture with a Function Composition Algorithm Based on Trie[C]. *The 2nd International Conference on Computer Science and Application Engineering*, 2018: 1-8.
- [70] Liu Y C, Lu Y, Qiao W X, et al. A Dynamic Composition Mechanism of Security Service Chaining Oriented to SDN/NFV-Enabled Networks[J]. *IEEE Access*, 2018, 6: 53918-53929.
- [71] Chemodanov D, Callyam P, Esposito F. A near Optimal Reliable Composition Approach for Geo-Distributed Latency-Sensitive Service Chains[C]. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019: 1792-1800.
- [72] Luizelli M C, Raz D, Sa'ar Y. Optimizing NFV Chain Deployment through Minimizing the Cost of Virtual Switching[C]. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018: 2150-2158.
- [73] Tomassilli A, Giroire F, Huin N, et al. Provably Efficient Algorithms for Placement of Service Function Chains with Ordering Constraints[C]. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018: 774-782.
- [74] Cziva R, Anagnostopoulos C, Pezaros D P. Dynamic, Latency-Optimal vNF Placement at the Network Edge[C]. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018: 693-701.
- [75] Zheng Z L, Bi J, Yu H, et al. Octans: Optimal Placement of Service Function Chains in Many-Core Systems[C]. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019: 307-315.
- [76] Trassare S T, Beverly R, Alderson D. A Technique for Network Topology Deception[C]. *MILCOM 2013 - 2013 IEEE Military Communications Conference*, 2013: 1795-1800.
- [77] Jafarian J H, Al-Shaer E, Duan Q. Adversary-Aware IP Address Randomization for Proactive Agility Against Sophisticated Attackers[C]. *2015 IEEE Conference on Computer Communications*, 2015: 738-746.
- [78] Singh J, Dhariwal S, Kumar R. A Detailed Survey of ARP Poisoning Detection and Mitigation Techniques[J]. *International Journal of Computer Technology and Applications*, 2017, 9(41): 131-137.
- [79] Tripathi N, Mehtre B M. Analysis of Various ARP Poisoning Mitigation Techniques: A Comparison[C]. *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies*, 2014: 125-132.
- [80] Anu P, Vimala S. A Survey on Sniffing Attacks on Computer Networks[C]. *2017 International Conference on Intelligent Computing and Control*, 2017: 1-5.
- [81] Achleitner S, La Porta T, McDaniel P, et al. Cyber Deception: Virtual Networks to Defend Insider Reconnaissance[C]. *The 8th ACM CCS International Workshop on Managing Insider Security Threats*, 2016: 57-68.
- [82] Piskozub M, Spolaor R, Conti M, et al. On the Resilience of Network-Based Moving Target Defense Techniques Against Host Profiling Attacks[C]. *The 6th ACM Workshop on Moving Target Defense*, 2019: 1-12.
- [83] Kang H S, Son J H, Hong C S. Defense Technique Against Spoofing Attacks Using Reliable ARP Table in Cloud Computing Environment[C]. *2015 17th Asia-Pacific Network Operations and Management Symposium*, 2015: 592-595.
- [84] Al Sukkar G, Saifan R, Khwaldeh S, et al. Address Resolution Protocol (ARP): Spoofing Attack and Proposed Defense[J]. *Communications and Network*, 2016, 8(3): 118-130.
- [85] Wang Y, Chau P, Chen F Y. Towards a Secured Network Virtualization[J]. *Computer Networks*, 2016, 104: 55-65.
- [86] Modi C, Patel D, Borisaniya B, et al. A Survey on Security Issues and Solutions at Different Layers of Cloud Computing[J]. *The*

- Journal of Supercomputing*, 2013, 63(2): 561-592.
- [87] Criswell J, Dautenhahn N, Adve V. Virtual Ghost[J]. *ACM SIGPLAN Notices*, 2014, 49(4): 81-96.
 - [88] Seo J, Lee B, Kim S, et al. SGX-Shield: Enabling Address Space Layout Randomization for SGX Programs[C]. *Proceedings 2017 Network and Distributed System Security Symposium*, 2017: 478-487.
 - [89] Jia L N, Zhu M, Tu B B. T-VM: Trusted Virtual Machine Inspection in Cloud Environments[C]. *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2017: 478-487.
 - [90] Li C X, Raghunathan A, Jha N K. Secure Virtual Machine Execution under an Untrusted Management OS[C]. *2010 IEEE 3rd International Conference on Cloud Computing*, 2010: 172-179.
 - [91] Wang X G, Qi Y, Dai Y H, et al. TrustOSV: Building Trustworthy Executing Environment with Commodity Hardware for a Safe Cloud[J]. *Journal of Computers*, 2014, 9(10): 2303-2314.
 - [92] Wu C, Wang Z, Jiang X. Taming hosted hypervisors with (mostly) deprivileged execution[C]. *Network and Distributed System Security Symposium*, 2013: 1-15.
 - [93] Deng L, Liu P, Xu J, et al. Dancing with Wolves: Towards Practical Event-Driven VMM Monitoring[J]. *ACM SIGPLAN Notices*, 2017, 52(7): 83-96.
 - [94] Cho Y, Kwon D, Yi H, et al. Dynamic Virtual Address Range Adjustment for Intra-Level Privilege Separation on ARM[C]. *Proceedings 2017 Network and Distributed System Security Symposium*, 2017: 1-15.
 - [95] Pattuk E, Kantarcioglu M, Lin Z Q, et al. Preventing Cryptographic Key Leakage in Cloud Virtual Machines[C]. *The 23rd USENIX conference on Security Symposium*, 2014: 703-718.
 - [96] Oren Y, Kemerlis V P, Sethumadhavan S, et al. The Spy in the Sandbox: Practical Cache Attacks in JavaScript and Their Implications[C]. *The 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015: 1406-1418.
 - [97] Chen S C, Liu F F, Mi Z Y, et al. Leveraging Hardware Transactional Memory for Cache Side-Channel Defenses[C]. *The 2018 on Asia Conference on Computer and Communications Security*, 2018: 601-608.
 - [98] Yu X, Xiao Y, Cameron K, et al. Comparative Measurement of Cache Configurations' Impacts on Cache Timing Side-Channel Attacks[C]. *12th {USENIX} Workshop on Cyber Security Experimentation and Test*, 2019:1-9.
 - [99] Cheng X, Su S, Zhang Z B, et al. Virtual Network Embedding through Topology-Aware Node Ranking[J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2): 38-47.
 - [100] Dietrich D, Rizk A, Papadimitriou P. AutoEmbed[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(4): 465-466.
 - [101] Esposito F, Matta I. A Decomposition-Based Architecture for Distributed Virtual Network Embedding[C]. *The 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing*, 2014: 53-58.
 - [102] Fischer A, Botero J F, Beck M T, et al. Virtual Network Embedding: A Survey[J]. *IEEE Communications Surveys & Tutorials*, 2013, 15(4): 1888-1906.
 - [103] Bays L R, Oliveira R R, Buriol L S, et al. Security-Aware Optimal Resource Allocation for Virtual Network Embedding[C]. *2012 8th International Conference on Network and Service Management and 2012 Workshop on Systems Virtualization Management*, 2012: 378-384.
 - [104] Xing C Q, Lan J L, Hu Y X. Virtual Network with Security Guarantee Embedding Algorithms[J]. *Journal of Computers*, 2013, 8(11): 2782-2788.
 - [105] Liu S H, Cai Z P, Xu H, et al. Security-Aware Virtual Network Embedding[C]. *2014 IEEE International Conference on Communications*, 2014: 834-840.
 - [106] Alaluna M, Ferrolho L, Figueira J R, et al. Secure Multi-Cloud Virtual Network Embedding[EB/OL]. 2017: 1703.01313. <https://arxiv.org/abs/1703.01313v3>.
 - [107] Zhang P Y, Li H S, Ni Y J, et al. Security Aware Virtual Network Embedding Algorithm Using Information Entropy TOPSIS[J]. *Journal of Network and Systems Management*, 2020, 28(1): 35-57.
 - [108] Zhang R, Su X J, Wang J P, et al. On Mitigating the Risk of Cross-VM Covert Channels in a Public Cloud[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2015, 26(8): 2327-2339.
 - [109] Wang Z M, Wu J X, Guo Z H, et al. Secure Virtual Network Embedding to Mitigate the Risk of Covert Channel Attacks[C]. *2016 IEEE Conference on Computer Communications Workshops*, 2016: 144-145.
 - [110] Del Piccolo V, Amamou A, Haddadou K, et al. A Survey of Network Isolation Solutions for Multi-Tenant Data Centers[J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(4): 2787-2821.
 - [111] Madi T, Jarraya Y, Alimohammadifar A, et al. ISOTOP: Auditing Virtual Networks Isolation across Cloud Layers in OpenStack[J]. *ACM Transactions on Privacy and Security*, 2018, 22(1): 1-35.
 - [112] Nasser A, Khamis H M, Duncan I M M. Investigation of Virtual Network Isolation security in Cloud computing: data leakage issues[J]. *Computer Science*, 2016, 1(5): 1-4.
 - [113] NVGRE: Network virtualization using generic routing encapsulation. <https://www.rfc-editor.org/info/rfc7637>. Sept. 2015.
 - [114] Sherwood R, Gibb G, Yap K K, et al. Flowvisor: A network virtualization layer[J]. *OpenFlow Switch Consortium, Tech. Rep*, 2009, 1(10): 132-146.
 - [115] Klöti R, Kotronis V, Smith P. OpenFlow: A Security Analysis[C]. *2013 21st IEEE International Conference on Network Protocols*, 2013: 1-6.
 - [116] Cabuk S, Dalton C I, Ramasamy H, et al. Towards Automated Provisioning of Secure Virtualized Networks[C]. *The 14th ACM Conference on Computer and Communications Security*, 2007: 235-245.
 - [117] Mundada Y, Ramachandran A, Feamster N. SilverLine: Data and Network Isolation for Cloud Services[C]. *HotCloud*. 2011: 1-6.

- [118] Barjatiya S, Saripalli P. BlueShield: A Layer 2 Appliance for Enhanced Isolation and Security Hardening among Multi-Tenant Cloud Workloads[C]. *2012 IEEE Fifth International Conference on Utility and Cloud Computing*, 2012: 195-198.
- [119] Moraes H, Nunes R V, Guedes D. DCPortalsNg: Efficient isolation of tenant networks in virtualized datacenters[C]. *Proc.13th ICN*, 2014: 241-246.
- [120] Fekih Ahmed M, Talhi C, Pourzandi M, et al. A Software-Defined Scalable and Autonomous Architecture for Multi-Tenancy[C]. *2014 IEEE International Conference on Cloud Engineering*, 2014: 568-573.
- [121] Costa V T, Costa L H M K. Vulnerabilities and Solutions for Isolation in FlowVisor-Based Virtual Network Environments[J]. *Journal of Internet Services and Applications*, 2015, 6: 18.
- [122] Gutz S, Story A, Schlesinger C, et al. Splendid Isolation: A Slice Abstraction for Software-Defined Networks[C]. *The First Workshop on Hot Topics in Software Defined Networks*, 2012: 79-84.
- [123] Thimmaraju K, Shastry B, Fiebig T, et al. The vAMP Attack: Taking Control of Cloud Systems via the Unified Packet Parser[C]. *The 2017 on Cloud Computing Security Workshop*, 2017: 11-15.
- [124] Thimmaraju K, Shastry B, Fiebig T, et al. Taking Control of SDN-Based Cloud Systems via the Data Plane[C]. *The Symposium on SDN Research*, 2018: 1-15.
- [125] Thimmaraju K, Rétvári G, Schmid S. Virtual Network Isolation: Are we there Yet?[C]. *The 2018 Workshop on Security in Softwarized Networks: Prospects and Challenges*, 2018: 1-7.
- [126] Gao X P, Wang S M, Chen X Q. VNSS: A Network Security Sandbox for Virtual Computing Environment[C]. *2010 IEEE Youth Conference on Information, Computing and Telecommunications*, 2010: 395-398.
- [127] Dey K, Mishra D, Kulkarni P. Vagabond: Dynamic Network Endpoint Reconfiguration in Virtualized Environments[C]. *The ACM Symposium on Cloud Computing*, 2014: 1-13.
- [128] Dong Y Z, Zhang X T, Dai J Q, et al. HYVI: A HYbrid Virtualization Solution Balancing Performance and Manageability[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2014, 25(9): 2332-2341.
- [129] Medeiros B, Simplicio M A Jr, Andrade E R. Multi-Tenant Isolation of What?: Building a Secure Tenant Isolation Architecture for Cloud Networks[C]. *The ACM Symposium on Cloud Computing*, 2018: 518-518.
- [130] Medeiros B, Simplicio M A, Andrade E R. Designing and Assessing Multi-Tenant Isolation Strategies for Cloud Networks[C]. *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops*, 2019: 214-221.
- [131] Wang X, Liu Z, Li J, et al. Tualatin: Towards Network Security Service Provision in Cloud Datacenters[C]. *2014 23rd International Conference on Computer Communication and Networks*, 2014: 1-8.
- [132] Yang M, Li Y, Jin D P, et al. Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey[J]. *Mobile Networks and Applications*, 2015, 20(1): 4-18.
- [133] Zang H, Nucci A. Traffic Monitor Deployment in IP Networks[J]. *Computer Networks*, 2009, 53(14): 2491-2501.
- [134] Jackson A W, Milliken W, Santivanez C A, et al. A Topological Analysis of Monitor Placement[C]. *Sixth IEEE International Symposium on Network Computing and Applications*, 2007: 169-178.
- [135] Xing T Y, Xiong Z Y, Huang D J, et al. SDNIPS: Enabling Software-Defined Networking Based Intrusion Prevention System in Clouds[C]. *10th International Conference on Network and Service Management and Workshop*, 2014: 308-311.
- [136] Xu H L, Huang H, Chen S G, et al. Scalable Software-Defined Networking through Hybrid Switching[C]. *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017: 1-9.
- [137] Lakhina A, Crovella M, Diot C. Diagnosing Network-Wide Traffic Anomalies[J]. *ACM SIGCOMM Computer Communication Review*, 2004, 34(4): 219-230.
- [138] Yan Q, Yu F R, Gong Q X, et al. Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges[J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(1): 602-622.
- [139] Cisco systems netflow services export version 9.<https://www.hjpat/doc/rfc/rfc3954.html>. Oct.2004.
- [140] Wang M, Li B, Li Z. SFlow: Towards Resource-Efficient and Agile Service Federation in Service Overlay Networks[C]. *24th International Conference on Distributed Computing Systems*, 2004: 628-635.
- [141] Shin S, Xu L, Hong S, et al. Enhancing Network Security through Software Defined Networking (SDN)[C]. *2016 25th International Conference on Computer Communication and Networks*, 2016: 1-9.
- [142] Yu M L, Jose L, Miao R, et al. Software Defined Traffic Measurement with OpenSketch[C]. *The 10th USENIX Conference on Networked Systems Design and Implementation*, 2013: 29-42.
- [143] van Adrichem N L M, Doerr C, Kuipers F A. OpenNetMon: Network Monitoring in OpenFlow Software-Defined Networks[C]. *2014 IEEE Network Operations and Management Symposium*, 2014: 1-8.
- [144] Phan X T, Fukuda K. SDN-Mon: Fine-Grained Traffic Monitoring Framework in Software-Defined Networks[J]. *Journal of Information Processing*, 2017, 25: 182-190.
- [145] Jiang D D, Huo L W, Li Y. Fine-Granularity Inference and Estimations to Network Traffic for SDN[J]. *PLoS ONE*, 2018, 13(5): e0194302.
- [146] Sekar V, Reiter M K, Willinger W, et al. Csamp[C]. *The 5th USENIX Symposium on Networked Systems Design and Implementation*, 2008: 233-246.
- [147] Yu Y, Qian C, Li X. Distributed and Collaborative Traffic Monitoring in Software Defined Networks[C]. *The Third Workshop on Hot Topics in Software Defined Networking*, 2014: 85-90.

- [148] Xu H L, Chen S G, Ma Q P, et al. Lightweight Flow Distribution for Collaborative Traffic Measurement in Software Defined Networks[C]. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019: 1108-1116.
- [149] VMware-vSphere 6.7. <https://www.vmware.com/jp/education-services/certification/vsphere-6-7-foundation-exam.html>.Feb.2019.
- [150] GigaVUE-VM datasheet. <https://www.gigamon.com/content/dam/resource-library/english/data-sheet/ds-gigavue-vm-virtual-mach-ine.pdf>.Apr.2016.
- [151] Fortinet. <https://www.fortinet.com/content/dam/fortinet/assets/ebook/eb-forticasb-cloud-security-use-cases.pdf>. Mar.2019.
- [152] Veryx vTAP datasheet. https://www.veryxtech.com/wp-content/uploads/2018/04/Datasheet-Veryx-vTAP_05.pdf.Apr.2018.
- [153] Ixia Panthom vTAP with tapflow filtering. <https://www.viavisolutions.com/ptbr/literature/ixia-phantom-vtap-tapflow-filtering-datasheet-en.pdf>.Feb.2016.
- [154] DeepFlow. <https://www.yunshan.net/solutions/npb.html>.Sep. 2020.
- [155] VMware NSX. <https://www.vmware.com/products/nsx-cloud.html>. May.2020.
- [156] NSFOCUS NCSS. https://www.nsfocus.com.cn/html/2019/198_0926/27.html.Nov.2019.
- [157] CloudWatcher[C]. *The 2012 20th IEEE International Conference on Network Protocols*, 2012: 1-6.
- [158] Qazi Z A, Tu C C, Chiang L, et al. SIMPLE-Fying Middlebox Policy Enforcement Using SDN[C]. *The ACM SIGCOMM 2013 conference on SIGCOMM*, 2013: 27-38.
- [159] Fayaz S K, Chiang L, Sekar V, et al. Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions Using FlowTags[C]. *Symposium on Networked Systems Design and Implementation*, 2014: 543-546.
- [160] Trajkovska I, Kourtis M A, Sakkas C, et al. SDN-Based Service Function Chaining Mechanism and Service Prototype Implementation in NFV Scenario[J]. *Computer Standards & Interfaces*, 2017, 54: 247-265.
- [161] He Q C, Wang Y, Li W J, et al. Traffic Steering of Middlebox Policy Chain Based on SDN[C]. *2017 IFIP/IEEE Symposium on Integrated Network and Service Management*, 2017: 754-759.
- [162] Zave P, Ferreira R A, Zou X K, et al. Dynamic Service Chaining with Dysco[C]. *The Conference of the ACM Special Interest Group on Data Communication*, 2017: 57-70.
- [163] Gember-Jacobson A, Viswanathan R, Prakash C, et al. OpenNF: Enabling Innovation in Network Function Control[J]. *ACM SIGCOMM Computer Communication Review*, 2014, 44(4): 163-174.
- [164] Anwer B, Benson T, Feamster N, et al. Programming Slick Network Functions[C]. *The 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, 2015: 1-13.
- [165] Gushchin A, Walid A, Tang A, et al. Scalable Routing in SDN-Enabled Networks with Consolidated Middleboxes[C]. *The 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, 2015: 55-60.
- [166] Guo L Q, Pang J, Walid A. Dynamic Service Function Chaining in SDN-Enabled Networks with Middleboxes[C]. *2016 IEEE 24th International Conference on Network Protocols*, 2016: 1-10.
- [167] Liu Y, Pei J N, Hong P L, et al. Cost-Efficient Virtual Network Function Placement and Traffic Steering[C]. *ICC 2019 - 2019 IEEE International Conference on Communications*, 2019: 1-6.



涂碧波 于 2009 年在中国科学院计算技术研究所计算机系统结构专业获得博士学位。现任中国科学院信息工程研究所研究员、博士生导师、中国科学院大学网络空间安全学院岗位教授。研究领域为数据中心前沿技术与安全体系,包括操作系统安全、软件定义边界和信息保护等。Email: tubibo@iie.ac.cn



孙瑞娜 于 2011 年在新疆大学计算机应用技术专业获得硕士学位。现在中国科学院信息工程研究所攻读博士学位。研究领域为网络安全。研究兴趣包括: 云数据中心网络安全、软件定义网络。Email: sunruina@iie.ac.cn



游瑞邦 于 2020 年在中国科学院信息工程研究所计算机系统结构专业获得博士学位。现任中国科学院信息工程研究所高级工程师。研究领域为计算机与网络体系结构。研究兴趣包括: 系统安全、云数据中心网络安全。Email: youruibang@iie.ac.cn



程杰 于 2016 年在燕山大学电子信息工程专业获得学士学位。现在中国科学院信息工程研究所攻读博士学位。研究领域为可信计算, 云计算安全。研究兴趣包括: 云数据中心网络安全。Email: chengjie@iie.ac.cn



陶小结 于 2017 年在海南大学通信工程专业获得学士学位。现在中国科学院信息工程研究所攻读博士学位。研究领域为云计算安全。研究兴趣包括: 同驻攻击、跨虚拟机侧信道攻击。Email: taoxiaojie@iie.ac.cn



张坤 于 2013 年在北京航空航天大学计算机技术专业获得硕士学位。现在中国科学院信息工程研究所攻读博士学位。研究领域为操作系统及虚拟化安全。研究兴趣包括: 计算机系统结构、操作系统安全和虚拟化安全。Email: zhangkun@iie.ac.cn