软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(1):183−207 [doi: 10.13328/j.cnki.jos.005879]

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

软件定义网络中的异常流量检测研究进展*

徐玉华[1,2], 孙知信[1,2]

[1](南京邮电大学 国家邮政局邮政行业技术研发中心(物联网技术),江苏 南京 210003)

[2](宽带无线通信与传感网技术教育部重点实验室(南京邮电大学),江苏 南京 210003)

通讯作者: 孙知信, E-mail: sunzx@njupt.edu.cn

摘 要: 软件定义网络(software defined networking,简称 SDN)是一种新型的网络架构.SDN 将控制层从数据层分离并开放网络接口,以实现网络集中控制并提高网络的可扩展性和编程性.但是 SDN 也面临诸多的网络安全威胁.异常流量检测技术可以保护网络安全,防御恶意流量攻击.对 SDN 异常流量检测进行了全面的研究,归纳了数据平面和控制平面可能遭受到的网络攻击;介绍并分析了位于应用平面、控制平面和中间平台的异常流量检测框架;探讨了异常流量识别机制、负载均衡机制、异常流量追溯机制和异常缓解机制;最后指明 SDN 异常流量检测在未来工作中的研究方向.

关键词: 软件定义网络;网络安全威胁;异常流量检测;异常流量追溯;异常流量缓解

中图法分类号: TP393

中文引用格式: 徐玉华,孙知信.软件定义网络中的异常流量检测研究进展.软件学报,2020,31(1):183−207. http://www.jos.org.cn/1000-9825/5879.htm

英文引用格式: Xu YH, Sun ZX. Research development of abnormal traffic detection in software defined networking. Ruan Jian Xue Bao/Journal of Software, 2020,31(1):183−207 (in Chinese). http://www.jos.org.cn/1000-9825/5879.htm

Research Development of Abnormal Traffic Detection in Software Defined Networking

XU Yu-Hua[1,2], SUN Zhi-Xin[1,2]

[1](Technology Research and Development Center of Postal Industry of State Post Bureau (Technology of Internet of Things), Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

[2](Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education (Nanjing University of Posts and Telecommunications), Nanjing 210003, China)

Abstract: Software defined networking (SDN) is new network architecture. SDN separates control layer from data layer and opens network interfaces to realize centralized network control and improve the scalability and the programmability of the network. But SDN is also facing a lot of network security threats. Abnormal traffic detection technologies can protect the network against malicious traffic attacks. This paper presents a comprehensive survey on the abnormal traffic detection of SDN. The possible network attacks on data plane and control plane are overviewed. Abnormal traffic detection frameworks on application plane, control plane, and intermediate platform are introduced and analyzed. The mechanisms of abnormal traffic identification, load balancing, abnormal traffic traceback, and abnormal traffic mitigation are discussed. The future work direction of SDN abnormal traffic detection is pointed out at the end.

Key words: software defined networking; network security threats; abnormal traffic detection; abnormal traffic traceback; abnormal traffic mitigation

* 基金项目: 国家自然科学基金(61672299, 61972208); 江苏省普通高校研究生科研创新计划

Foundation item: National Natural Science Foundation of China (61672299, 61972208); Postgraduate Research & Practice Innovation Program of Jiangsu Province

收稿时间: 2018-08-01; 修改时间: 2019-05-08; 采用时间: 2019-08-03; jos 在线出版时间: 2019-11-06

CNKI 网络优先出版: 2019-11-06 11:49:11, http://kns.cnki.net/kcms/detail/11.2560.TP.20191106.1148.006.html

在当今信息时代,云计算和大数据的出现,使得计算机网络具有动态性、复杂性、并发性和实时性等特点,这就要求网络管理员能够实施高级策略实现网络的自动配置.而传统网络配置更改复杂,容易导致配置错误[1];此外,Internet 的固化机制阻碍了网络基础设施的创新和演进[2].为了解决传统 TCP/IP 网络结构的诸多难题,斯坦福大学的 Clean Slate 课题组于 2009 年提出了软件定义网络(software defined networking,简称 SDN)的概念[3].SDN 将控制平面与数据平面分离,提高了网络的可编程性、集中控制性,简化了网络的配置操作过程.SDN 能有效降低网络负载,提高网络工作效率,实现网络性能的扩展.

然而,SDN 与传统网络一样面临着网络攻击的威胁[4].网络攻击发生时常常表现为流量异常,所谓异常流量是指网络流量的行为不符合预期的正常行为模式.异常流量的出现,意味着网络中可能存在某些未经授权的信息访问和数据操作[5],例如拒绝服务(denial of service,简称 DoS)攻击使相应的服务器过载、蠕虫和病毒通过网络利用已知漏洞对主机进行特权访问与攻击等.因此,异常流量在网络中的频繁出现,将危害网络的有效性和可靠性.

异常流量检测技术是指通过有效的技术手段识别和过滤网络中的异常流量,是一种保证网络安全的基本方法[6].及时、准确的异常流量识别,能够有效减小恶意攻击对网络以及网络运营业务造成的影响.SDN 的实时异常流量检测能够保障 SDN 网络信息的机密性、完整性和安全性,同时能够进一步推动 SDN 的发展与应用.因此,研究 SDN 中的异常流量检测技术具有重要的理论和应用价值.

SDN 网络架构、工作流程与传统网络有所不同,利用网络漏洞进行网络攻击方面也存在显著差异,因此 SDN 网络异常流量检测方法和框架也具有其独特性.本文在分析了 SDN 架构特点及其安全威胁的基础上,对现有的 SDN 异常流量检测框架进行分析和归纳,并对 SDN 异常流量检测过程中的主要机制及其实现方法进行总结和探讨,希望能够为 SDN 网络的异常流量检测研究提供参考.

本文第 1 节简述和分析 SDN 的关键技术及其主要特点.第 2 节归纳 SDN 网络各层的异常流量安全威胁并分析其主要特征.第 3 节分析并探讨现有的 SDN 异常流量检测框架.第 4 节对 SDN 异常流量检测的主要机制及其实现方法进行归纳和分析.第 5 节提出 SDN 异常流量检测的未来工作方向.第 6 节对全文进行总结.

1 SDN 的关键技术

SDN 的核心思想是:将控制平面从数据平面分离,从而把控制功能从网络设备中移除,使网络设备成为简单的转发单元.SDN 的转发决定是基于流的,流是由分组字段值中的匹配准则及动作指令来定义的.流抽象统一了不同类型的网络设备,包括路由器、交换机、防火墙和中间件的行为,使得网络管理更为灵活.SDN 具有开放性、逻辑集中控制性、可编程性、可扩展性、抽象与虚拟化等特点[7].

1.1 SDN的基本架构

SDN 的基本架构由上至下可以分为应用平面、北向编程接口、控制平面、南向编程接口和数据平面[8],如图 1 所示.

- 应用平面包括各种服务和应用,如云计算、深度数据包解析(deep packet inspection,简称 DPI)、入侵检测系统(intrusion detection system,简称 IDS)\入侵防御系统(intrusion prevention system,简称 IPS)、安全监测、负载均衡等;
- 控制平面主要包含 SDN 的控制器,它负责底层转发设备的统一控制,以及提供上层应用网络的呼叫能力;
- 数据平面主要包含交换机、路由器等网络转发单元[9].转发单元根据由控制平面管理的流表来转发数据分组,它还负责收集网络信息,并向控制平面发送网络拓扑数据.

应用程序和网络管理系统通过控制器提供的北向编程接口向控制器请求服务[10].控制器根据应用平面程序来提供网络拓扑管理、网络服务发现机制和网络路径计算等服务,并建立资源、策略、服务之间的关系模型,对网络状态进行管理,将网络状态存储至相应的数据库.南向编程接口则允许控制器与数据平面的转发单元进行安全通信.OpenFlow 是一种经常用作南向应用程序编程接口(application programming interface,简称 API)的

协议,它定义了一组开放的用于数据转发的命令.这些命令允许路由器基于应用程序发现网络拓扑.

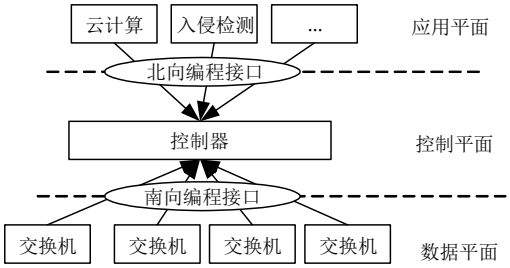

Fig.1 Basic architecture of SDN[8]

图 1 SDN 的基本架构[8]

1.2 OpenFlow技术

OpenFlow 是 SDN 最重要的技术之一.如图 2 所示,在 OpenFlow 架构中[11],OpenFlow 转发单元包含一个或多个流表,而控制器通过 OpenFlow 协议安全地与转发单元通信,实现转发单元流表的更新与删除.其中,流表由流条目组成,每个流条目确定如何处理和转发属于流的分组.流条目包括以下几个部分:(1) 用于匹配相应分组的字段或规则;(2) 用于统计特定流相关数据的计数器,例如,接收的分组的数量、字节的数量和流的持续时间;(3) 用于决定如何处理匹配的数据包的动作指令.

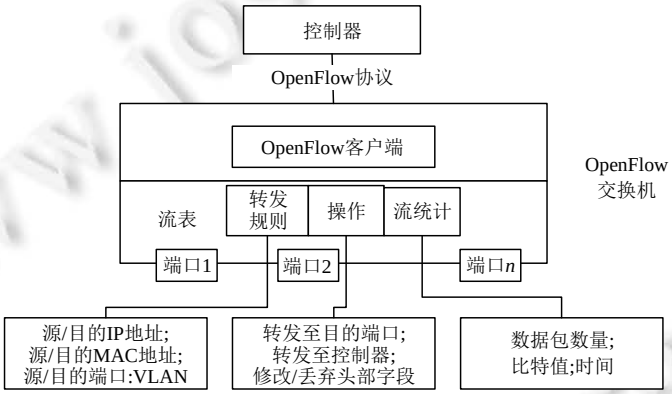

Fig.2 Communication between the controller and the forwarding devices via OpenFlow protocol[11]

图 2 控制器与转发单元之间通过 OpenFlow 协议[11]进行通信

当分组到达转发单元时,分组报头字段被提取并且与流表条目的匹配字段进行匹配[12]:如果找到匹配的条目,转发单元则执行相应的指令;如果流表查找过程没有找到匹配字段,转发单元则执行流表缺失流条目所定义的指令.每个流表必须包含一个流表缺失条目,以处理未匹配情况,例如丢弃分组;在下一个流表上继续匹配过程;或者通过 OpenFlow 信道将分组转发到控制器进行传输路径的确认,再根据下发结果进行转发.

SDN 网络架构和 OpenFlow 技术使控制器能实时掌握全局网络状态、拓扑信息、应用需求,并提供可编程的数据接口,网络管理者可以根据网络状态动态分配网络资源,这为异常流量检测和处理提供了基础[13].

2 SDN 安全性分析及其异常流量特征分析

SDN 通过引入集中控制器即决策点、控制平面和数据平面之间的标准接口(OpenFlow)以及为网络管理应用提供统一的 API,使得网络具有逻辑集中性、可编程性、开放性等特点.而 SDN 这些特点也给 SDN 网络带来了安全问题,例如,基于 OpenFlow 的 SDN 的操作语义降低了在控制和数据平面上安装复杂攻击的障碍[14],允许

任何不匹配的分组发送到控制器来询问转发规则.SDN 网络的异常流量攻击主要体现在数据平面和控制平面,分析数据平面和控制平面的安全性及其可能遭受到的网络攻击,并进一步分析各类攻击所造成的流量的变化,可以为 SDN 网络异常流量检测提供攻击特征依据.

2.1 数据平面安全威胁

数据平面安全威胁主要以攻击交换机和用户个人计算机(personal computer,简称 PC)为主.如图 3 所示,在数据平面,攻击者可以通过恶意用户主机和恶意交换机对合法用户和合法交换机进行攻击.交换机需要与控制器进行必要通信,获取相应的控制服务,因此,数据平面的安全性会影响控制平面度的安全性.尽管 OpenFlow 支持交换机和控制器之间的传输层安全性协议(transport layer security,简称 TLS)认证,但 TLS 本身不能防止受感染的交换机发送数据包.例如,SDN 控制器使用 OpenFlow PACKET_IN 消息传播和构建网络拓扑,当终端主机发送伪造的消息时,该消息将被交换机作为 PACKET_IN 消息中继到控制器,从而中毒其网络拓扑.其次,交换机具有有限的流表[15],恶意的网络攻击可能会造成流表的溢出.此外,交换机直接与用户主机连接,更容易暴露在攻击者面前,因此数据平面安全具有一定的脆弱性.

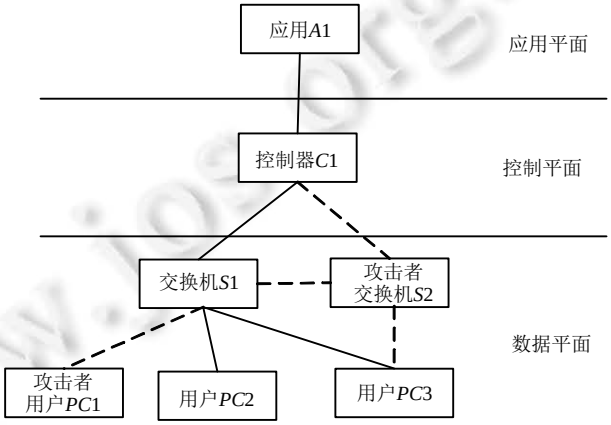

Fig.3 Abnormal flow attacks on the data plane

图 3 数据平面的异常流量攻击

数据平面可能受到的网络攻击具体如下.

(1) SDN 扫描

攻击原理:SDN 扫描[16]攻击根据匹配的流和非匹配流响应时间的不同来扫描匹配字段,确定网络是否使用 SDN/OpenFlow 交换机,并根据扫描到的匹配字段进一步对控制器和交换机进行攻击,而不需要高性能或高容量装置.SDN 扫描器向目标交换机重复发送同一头域内不同字段值的数据包,并记录和比较每个数据包的响应时间.达到规定次数后,SDN 扫描器更改头域,再重复此操作.

SDN 扫描的流量特征:因为 SDN 扫描主要是扫描头域,并根据每段头域扫描的相应时间计算误差,因此,网络上在某几个时间段内会有一定的流量从同一主机转向交换机,且这几个时间段相隔时间较为相同.

(2) ARP 攻击

攻击原理:地址解析协议(address resolution protocol,简称 ARP)攻击[17]一般可分为 ARP 泛洪攻击和 ARP 欺骗攻击.SDN 数据平面的 ARP 泛洪攻击是指攻击者通过向交换机管理的局部网络发送大量的 ARP 报文,使得交换机和合法用户无法正常通信.SDN 数据平面的 ARP 欺骗是指攻击者伪造一系列错误的 MAC(media access control address)地址信息,并将错误的 MAC 地址映射到合法网际协议地址(Internet protocol address,简称 IP),然后按照一定的频率向数据平面交换机发送错误的 ARP 报文,使交换机无法保存正确的 MAC 地址信息,进而无法正常工作.另一方面,攻击者也可以通过嗅探工具将自己的 MAC 地址与合法的 IP 地址映射,然后劫持合法用户流量数据,致使该用户被屏蔽出网络.

ARP 攻击的流量特征:为了使交换机无法与合法用户进行通信,网络上可能出现的大量 ARP 的请求和响应报文.另一方面,攻击者为了劫持合法用户的数据,则可能短时间内向交换机发送一定数量的 ARP 报文.

(3) 网络病毒攻击

攻击原理:网络病毒攻击的目的一般是攻击网络上尽可能多的计算机,SDN 的客户端 PC 机较易成为病毒攻击的主要对象.以蠕虫攻击[18]为例,蠕虫攻击的工作方式是攻击者随机产生 IP 地址并进行地址探测,如果该地址的主机存在该 IP 地址,则扫描该主机是否存在漏洞;如果存在漏洞,则在该主机上复制病毒进行传染攻击.

网络病毒攻击的流量特征:因为攻击者的计算机需要产生大量的 IP 地址进行探测,因此在一定时间内,攻击者的计算机会产生大量的网络流量,并且这些网络流量中有很多空的或不可达的数据包.而被攻击的计算机或交换机的同一个端口,则在某一时间段内会接收到来自同一 IP 地址计算机的大量的流量.

(4) 交换机劫持

攻击原理:为了防止广播风暴并节省能源,OpenFlow 控制器提供了生成树服务[19].当发生任何拓扑更新时,会触发生成树服务以阻止这些冗余端口.然而,这种能力可以被攻击者利用来发起拒绝服务攻击,通过将伪链路注入到现有拓扑中,攻击者可以借用生成树服务来“杀死”正常的交换机端口.在交换机劫持过程中,攻击者先利用中间人攻击[20,21]窃取原有交换机的注册证书、MAC 地址和 IP 地址等信息,然后利用上述方法使原有交换机无法正常工作,之后将自己的 MAC 地址与被攻击的交换机的 IP 地址映射在一起,向所属控制器进行重新注册和认证,最后与客户主机进行重新连接,从而取代原有的交换机.攻击者就可以劫持交换机以窃取网络信息或造成该交换机所属的局域网络瘫痪.

交换机劫持的流量特征:网络上有大量的流量涌入原有的交换机造成该交换机无法输出流量,而该交换机所属网络与控制器之间却有流量的正常传输.

2.2 控制平面安全威胁

控制平面安全威胁,主要是指以网络中控制器为目标的网络攻击[4],如图 4 所示,恶意应用、恶意控制器、恶意交换机和恶意用户都可以对控制器造成安全威胁.大多数 SDN 控制器在设计和开发之初,主要关注的是网络资源的调度和控制问题而未将控制器的安全问题作为核心研究内容[22].控制平面的控制器是中心化决策实体,控制器管理多个交换机,从拓扑结构角度而言,控制器容易成为网络瓶颈.此外,控制器负责所属网络的资源管理和服务管理,是网络的处理中心,控制平面的安全性对数据平面有直接影响,因此,攻击者如果能掌控 SDN 网络控制器,就可以轻易地在网络中进行恶意行为.

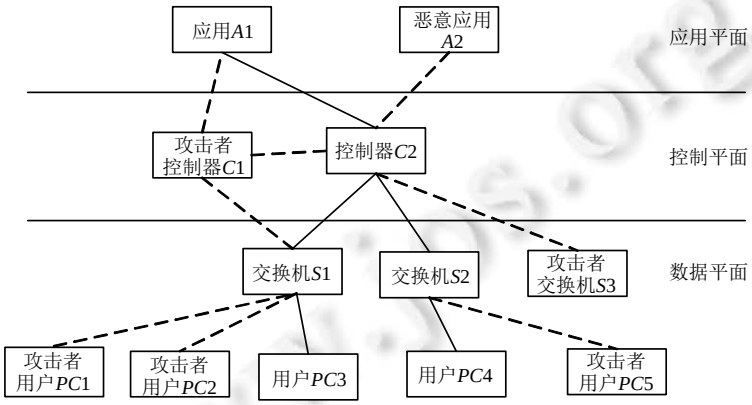

Fig.4 Abnormal flow attacks on the control plane

图 4 控制平面的异常流量攻击

控制平面可能受到的网络攻击具体如下.

(1) 面向控制器的 DoS 攻击

攻击原理:面向控制器的 DoS 攻击是最严重的威胁,因为它会影响整个网络性能,造成网络延迟和合法数据包的丢弃,甚至可以导致整个网络被禁用或瘫痪.攻击者向控制器发起各种不同的 DoS 拒绝服务攻击[23],其中主要包括两种:第 1 种由交换机引起的 DoS 攻击,因为交换机需要定时向控制器发送信息,向控制器报告自己的状态,或询问相关流规则,因此,攻击者可以利用多个交换机同时向控制器发送各类信息,从而导致控制器对 Packet_In 事件响应非常缓慢,并发送 Packet_Out 消息的速度也随之变慢;另一种则是由用户主机引起的 DoS 攻击,攻击者可以不断地发送具有随机性的 IP 分组将控制器置于非响应状态[24],例如控制大量新入网的僵尸 PC 机向同一控制器管理的不同交换机发送数据,因为交换机内没有匹配的流转发规则,则需要向控制器发送大量询问信息.这种攻击的目的是生成大量的数据包并发送到控制器[25],控制器则需为每个数据包生成新的流规则,最终导致控制器一直处于繁忙状态而无法响应正常用户的请求.

DoS 攻击流特征:在攻击期间,会有海量流量转向被攻击的控制器,且在一段时间后,控制器所负载的流量将达到饱和状态.

(2) 控制器劫持

攻击原理:控制器劫持攻击[26]不同于数据平面的交换机劫持,攻击者使用应用平面恶意程序或 DoS 攻击将原控制器屏蔽出原网络,并将伪造的控制器迁移至目标网络.攻击者首先向原来控制器的指定代理进行 TCP 连接,通知代理服务器原控制器已故障进行新控制器的迁移,代理服务器在“伪控制器”上安装相关策略并进行相关参数配置之后,再将交换机迁移至“伪控制器”.这样就形成了控制器劫持攻击,攻击者就可以通过“伪控制器”在交换机的流表中创建条目来控制整个网络[27],而不易被网络管理者所发现.

控制器劫持流量特征:原有的合法控制器中无流量传输,而其所管理的各个交换机在网络中却有流量输送.

(3) 恶意应用攻击

攻击原理:SDN 使第三方应用能够被集成到网络体系结构[28]中,攻击者可以设计开发恶意的应用软件,通过北向接口操纵控制器[20,24],使之成为恶意控制器,恶意控制器可以直接破坏网络;或者直接通过南向接口篡改交换机的流转发规则,导致流转发至特定节点从而窃取流信息.

恶意软件攻击流量特征:恶意应用攻击控制器时,其流量特征必须根据恶意应用程序的具体功能才能判断.例如:如果攻击者利用恶意应用程序窃听信息,那么网络流量就会从特定交换机转向特定的用户主机.如果攻击者使用恶意应用通过控制器攻击交换机,那么网络上就可能会有大量的流量从各个交换机转向特定的交换机.

3 SDN 异常流量检测框架

通过上一节的分析,传统网络的解决方案不能直接适用于 SDN,因为传统的防御方案假设交换机是智能的,而控制平面和数据平面的分离迫使 SDN 交换机成为只具备简单功能的转发实体,且安全防御功能尚不是 SDN 架构中的内置功能,所以安全性已成为 SDN 网络的一个重要问题.而可扩展的、灵活的 SDN 异常流量检测系统可以防御 SDN 网络所存在的安全威胁,该系统需要在执行特定的流量分析时,对分析结果进行及时反应.这就要求部署合理的系统框架,并配置相关的系统组件实现系统功能.SDN 异常流量检测框架应该具有以下特点.

(1) 灵活性——灵活的组件配置完成各类异常检测的任务,例如,主动反应定义路径的实例化;在流表中部署特定或通用流规则;以及管理流参数,例如,超时时间和数据速率;
(2) 准确性——针对不同的网络分析情况进行不同的流量处理,保障网络安全;
(3) 功能完整性——部署不同的功能组件,实现异常流量检测过程中的不同功能,如异常识别、异常分析、异常缓解等;
(4) 高效性——高效性包括两个方面:一方面是指系统的响应时间应足够快;另一方面是指性能开销应足够小,而不应影响其他网络应用程序.

根据框架部署结构和实现原理的不同,可以将 SDN 异常流量检测框架分为应用平面框架、控制平面框架和中间平台框架.表 1 列举了现有的主要 SDN 异常流量检测框架,将其进行了分类,并列举了各个框架的主要技术及其优缺点.

Table 1 Comparison of frameworks of SDN abnormal traffic detection

表 1 SDN 异常流量检测框架对比

框架名称	所属层次	主要技术	优点	缺点
SPHINX[29]	控制平面	SPHINX 动态学习新的网络行为,并当检测到对网络控制平面行为的可疑更改时发出警报	SPHINX 能够以低性能开销实时检测 SDN 中的攻击,并且不需要更改控制器进行部署	无法识别瞬态攻击,不适用于大规模的 SDN 环境
PANDA[30]	控制平面	通过网络监控模块、网络入侵检测模块和网络入侵响应模块实现异常流量检测的基础功能	能满足网络动态检测的基础需求,可扩展性高	功能性较为基础,不能满足复杂的异常流量识别需求
SDN-MON[31]	控制平面	将监控信息与现有转发表相分离,允许控制器根据应用要求定义任意一组监控匹配字段,以便灵活地监控流量	监控模块可以更加灵活地为各种网络管理应用程序提供服务	只提供了一种流量监控功能,不具备异常流量识别和防御功能
FLOWGUARD[32]	应用平面	检查网络流路径空间,并为多种网络情况设计多种检测策略,自动和实时地执行异常流量攻击的解决方案	框架功能性强,集成度高,提供 SDN 网络可视化、优化和迁移	因为框架的集成度高,因此系统策略更新困难度大,需要重新设计和部署检测框架
RAD[33]	应用平面	流量分析器根据攻击签名和网络监测情况识别异常流量;流量管理器测量网络利用率和延迟,以便在异常流量情况下确定多维路由和负载平衡的最佳路径;规则管理器为选定的最佳路径生成并安装流规则	能实时监控和管理网络,并在出现网络异常时提供相应的攻击防御方法	主要利用现有的网络流量分析工具,因此网络流量检测功能相对较为简单,不能对复杂的网络情况进行分析与处理
SDN-ecosystem[34]	应用平面	通过流量收集模块提取网络监控数据,通过检测模块确定收集的流量是否为异常流量,缓解模块通过丢包、修改流条目来缓解异常流量,报告模块存储检测到的网络攻击及其流量特征	在对异常流量进行处理之后,能够生成已识别攻击的报告,以验证和审核系统的处理结果,并实现 SDN 网络的自动化管理	异常流量检测模块使用正常流量特征配置文件进行异常识别,其算法复杂度较大,易造成 CPU 过载
ATLANTIC[35]	应用平面	结合信息熵的偏差和一系列机器学习算法来分类识别流量,并对不同级别的异常流量使用不同的缓解处理方法	提供了精细化的异常流量识别方法,提高了异常识别的精确度	不具备异常追溯功能,没有考虑网络的负载均衡
Streamon[36]	控制平面与应用平面之间的中间平台	流量分析应用程序(状态跟踪、特性分析、异常条件等)的编程逻辑与基本原语(计数、匹配、事件生成等)相分离,并通过部署自定义状态、相关状态转换、监视操作和触发条件来支持多阶段的实时跟踪和异常检测	该框架可用高级语言编写与平台无关的便携式监控设备的在线分析任务,而不需要通过低级语言来访问监控设备内部以编程其监控逻辑	不能长时间监测网络流量,且不提供复杂计算、统计分析和深度数据包解析的基本组件
FRESCO[37]	控制平面与应用平面之间的中间平台	提供了一个 Click-inspired[38] 编程框架,能够共享、组合许多不同的安全检测和缓解模块	所需的代码量较小,简化了安全功能的开发,具有重定向、镜像和隔离等安全功能	没有考虑 SDN 异常检测过程中的数据平面和控制平面的负载均衡
PayLess[39]	控制平面与应用平面之间的中间平台	使用自适应算法收集流信息,并为不同聚合级别的流信息的收集和统计提供了灵活的 RESTful API	使用自适应算法提供高度准确的信息,而不会产生显著的网络开销	不具备恶意流量识别和分析的基本组件
Athena[40]	控制平面与应用平面之间的中间平台	南向组件用于监视网络行为包括控制平面,从 SDN 控制消息中提取特征,实现实时检测算法,并调用缓解机制;北向组件提供高级 API,以使应用程序能够执行异常检测任务;分布式数据库集群提供网络特征访问;由分布式并行计算集群运行 Athena 应用程序	Athena 采用分布式数据库和集群计算平台,可以在网络中大规模部署分布式检测算法,实现 SDN 的分布式异常流量检测	由于网络特征存储在分布式数据库中,且应用程序运行在分布式集群中,会造成较大的网络通信开销,甚至导致网络通信堵塞

3.1 位于控制平面的SDN异常流量检测框架

如图 5 所示,位于控制平面的 SDN 异常流量检测框架作为控制器的一个内置功能软件部署在控制器中.因为检测框架位于控制器内,则与控制器其他功能模块协调工作,尤其可以利用控制器网络全局拓扑和流量统计信息来提高异常流量的检测效率.

OpenFlow 比传统通信协议简单得多,且所有智能集中在控制器中,网络更新则是可观察的,这就显著地简化了 SDN 内控制消息的分析.因此,在控制器中部署一个小的异常流量检测功能相对于全面的异常流量检测框架,更易于检测控制消息的模式变化.

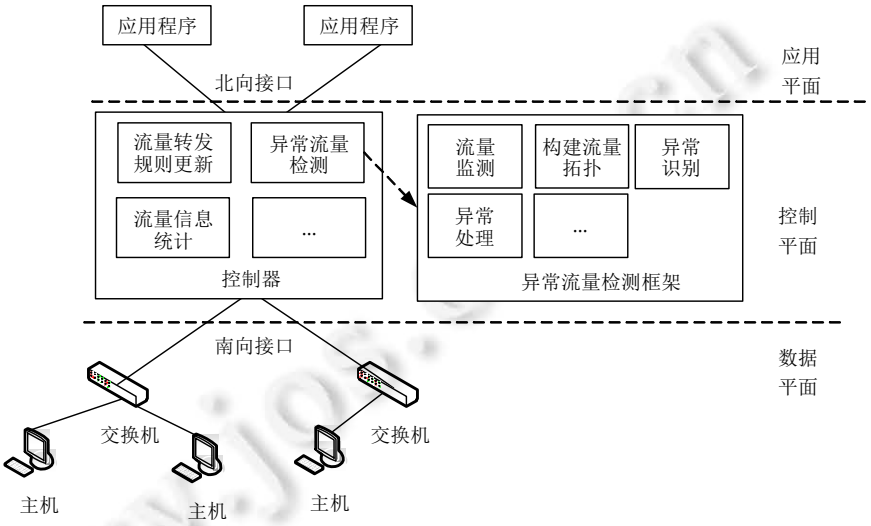

Fig.5 SDN abnormal traffic detection framework on control plane

图 5 位于控制平面的 SDN 异常流量检测框架

在控制平面,SPHINX[29]必须拦截每个网络更新以检测异常行为;分析 OpenFlow 控制消息,以递增地构建和更新与网络中的每个流相对应的流图;然后,它通过识别与每个流图相关联的网络拓扑或数据平面的元数据中的变化来检测网络攻击.SPHINX 还提供轻量级的策略引擎,管理员可以通过指定表达式来执行各类策略来检测网络中的安全威胁.而 PANDA[30]则通过网络监控模块利用控制器的 RESTful API 来发送交换机命令,聚合交换机流量并进行统计和预处理;网络异常检测模块提供可以实现各类轻量级入侵检测方法的灵活接口;网络入侵响应模块则基于标准策略来定义响应动作,自动执行适当的补救策略.如图 6 所示,SDN-Mon[31]由控制器侧模块和交换机侧模块组成.

- 控制器侧模块具有 SDN-Mon 监控 API,该 API 在 SDN 控制器平台上运行,以支持控制器中相关的监控应用程序;
- 交换机侧模块由 3 个组件组成:SDN-Mon 本地控制应用程序、SDN-Mon 监控 API、SDN-Mon 监控数据库.这些组件可以共同实现交换机中的监控功能.

由此,SDN-Mon 可以通过控制器应用程序定义的监控字段来实现细粒度的网络监控.

由于 SDN 异常流量检测框架作为控制器的一个功能模块,只能执行异常流量检测过程中的基础功能,无法进行较为复杂的 SDN 异常流量检测计算,以避免将流量数据输出到控制器核心区.因为核心区无法处理规模过于庞大的流量数据,或提供具有严格的延迟要求的流量处理措施.

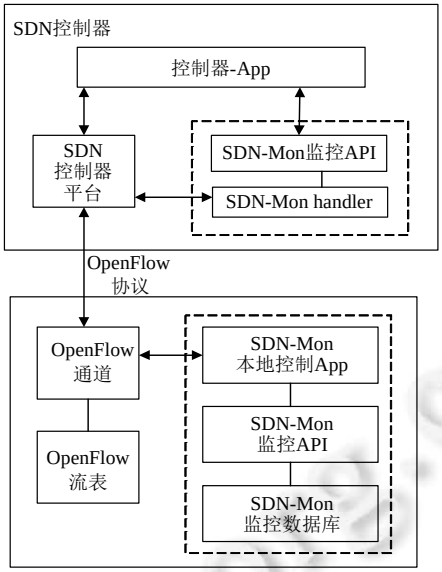

Fig.6 Abnormal traffic detection framework of SDN-Mon[31]

图 6 SDN-Mon 异常流量检测框架[31]

3.2 位于应用平面的SDN异常流量检测框架

如图 7 所示,位于应用平面的检测框架作为 SDN 的一个应用程序框架,通过北向接口与控制器通信,以进行异常流量的检测.位于应用平面的检测框架具有集成度高、功能性强的异常流量检测管理功能,网络管理员只需在异常流量检测应用中简单地设置参数或部署相关算法,就可以自动化地进行流量的识别分类和相关处理.该类系统能够在执行特定的流量分析(例如,追踪事件和多级攻击)时对变化的分析需求进行及时反应,并且能处理多种成分的流量特征、异常及攻击行为和网络行为误判等多种情况.

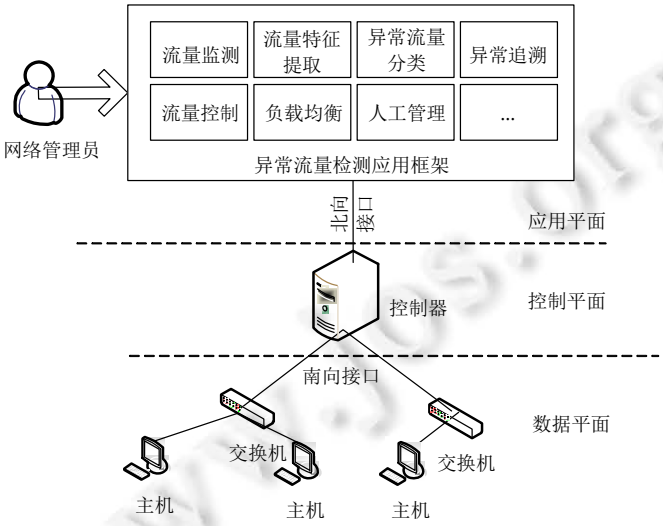

Fig.7 SDN abnormal traffic detection framework on application plane

图 7 位于应用平面的 SDN 异常流量检测框架

在应用平面,FLOWGUARD 综合框架[32]提供了一个系统性的网络异常保护方案,主要包含异常检测模块

和异常决策模块:异常检测模块通过测试流路径空间来检测异常流量,并能够检查流表,追踪流路径;异常决策模块实现自动和实时地解决各类网络异常.FLOWGUARD 还集成了多种工具包,用于支持 SDN 的可视化、优化、迁移和集成.RAD 框架[33]提供了 SDN 异常流量的实时监控和处理功能,它由 3 个主要模块组成:流量分析器、流量管理器和规则管理器.流量分析器使用 sFlow-RT(一种实时网络测量工具)和 Snort IDS(一种基于签名的入侵检测系统)来监控流量,当检测到异常时,流量管理器根据网络状态生成新的路由路径,规则管理器则为新的路由路径的数据平面生成流规则.SDN-ecosystem 框架[34]能够对 SDN 网络进行自动化管理,它通过多个流量特征来分析正常的网络的状态,并生成正常配置文件用于识别异常流量模式,并根据识别的异常选择缓解策略;它不仅有数据收集、异常检测和缓解模块,还具有报告生成模块,在对异常流量进行处理之后,生成已识别攻击的报告,以验证和审核系统的处理结果.而 ATLANTIC 框架[35]能够实现各类异常流量检测算法,如图 8 所示,主要可以分为统计层和分类层:在统计层,网络驱动模块定时与控制器通信获取流量数据,然后传输给特征选择模块来记录流量特征参数,流量统计模块收集和处理来自网络驱动模块和特征选择模块的信息并传输至分类层的相应的模块;在分类层,流量监测模块主要部署信息熵和信息统计等轻量级异常流量检测算法,流量分类模块部署机器学习等重量级异常流量检测算法,流量管理模块执行流量异常缓解措施.因此,ATLANTIC 框架实现了轻量级和重量级算法相结合的异常流量检测.

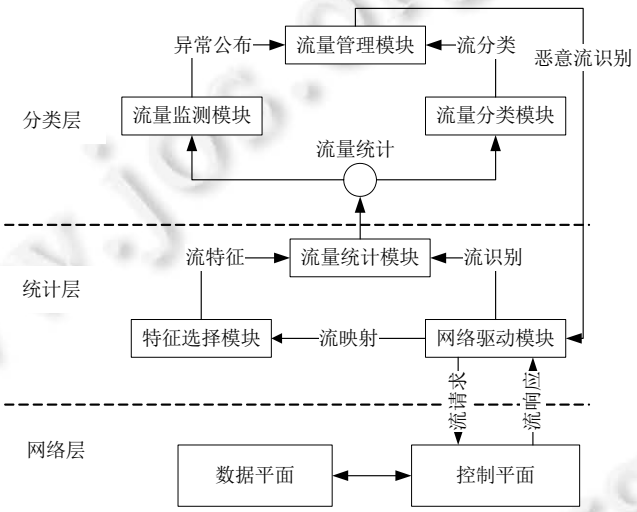

Fig.8 Abnormal traffic detection framework of ATLANTIC[35]

图 8 ATLANTIC 异常流量监测框架[35]

虽然应用平面的异常流量检测框架的功能性强,但与此同时,高集成性导致框架不具备灵活性和可编程性.当异常流量检测的策略和功能发生变化时,系统框架需要重新设计和部署.

3.3 作为中间平台的SDN异常流量检测框架

如图 9 所示:SDN 异常流量检测框架还可以被设计成位于应用平面和控制平面的中间平台,该平台支持开发人员对 SDN 安全应用进行二次开发,并将异常流量检测的基础功能抽象为逻辑组件,然后通过北向接口来调用控制器中的流量信息统计、流量转发、规则更新等流管理功能模块,并为各类异常流量检测应用提供专用的 API 来组合和调用异常流量检测功能组件.用户则可以在应用平面根据特定的需求编写相关代码,实现所需功能.因此,作为中间平台的 SDN 异常流量检测框架具有可扩展性、精确性、灵活性、可编程性等特点,简化了 SDN 网络安全服务的开发和部署.

StreaMon[36]是一个支持各类流量监控需求的执行平台,StreaMon 处理引擎架构包含 4 个模块层次:用于解析数据包的事件层、更新流量存储的数据结构的度量层、计算流量特征值的特征层、触发特定处理操作的判

决层.这 4 个模块层次又组成了两个子系统:向程序员提供逻辑组件的逻辑子系统和提供快速计算和存储的测量子系统.它将编程逻辑与基本原语(计数、匹配、事件生成等)相分离,用高级语言就可实现与平台无关的便携式监控设备的在线分析任务.

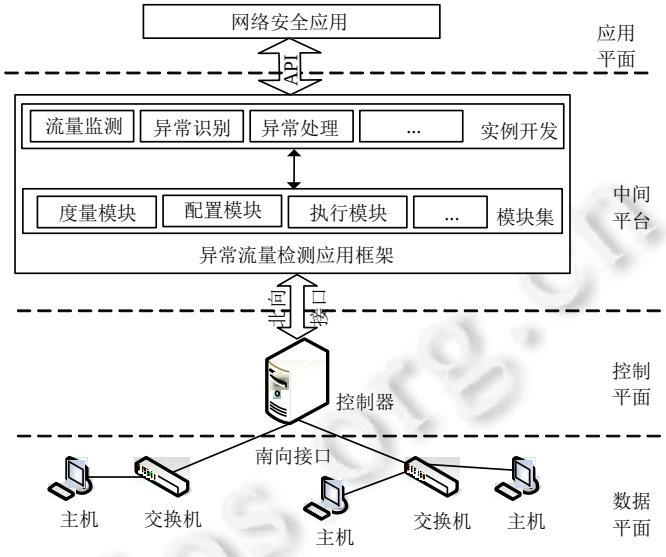

Fig.9 SDN abnormal traffic detection framework on intermediate platform

图 9 位于中间平台的 SDN 异常流量检测框架

FRESCO 框架[37]如图 10 所示,包括:应用层平台提供一个解释器和支持可组合的应用程序开发的应用接口;安全内核强制执行来自开发的安全应用程序的策略动作,内核组件集成到 NOX(一个开源的 SDN 控制器).FRESCO 的应用层平台使用 NOX python 实现,提供两个关键的开发功能:开发环境和资源控制器(访问及统计网络流事件).FRESCO 提供了一个 Click-inspired[38]编程框架,它的代码量较小,简化了安全功能的开发,具有重定向、镜像和隔离等安全防御功能.

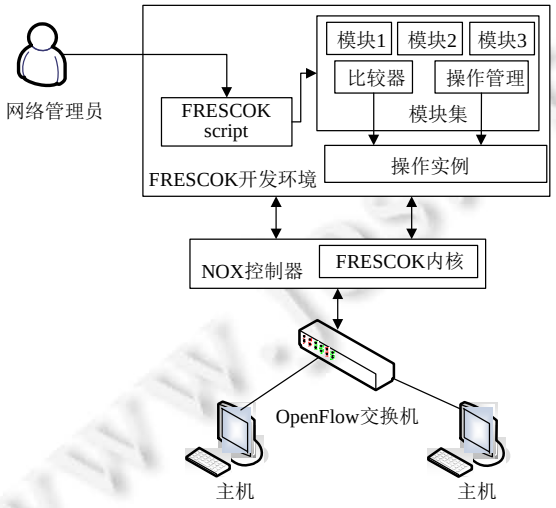

Fig.10 Abnormal traffic detection framework of FRESCO[37]

图 10 FRESCO 异常流量监测框架[37]

PayLess[39]为不同聚合级别的流统计信息收集提供灵活的 RESTful API.它使用自适应的轮询信息收集算法,可实时提供高度准确的信息,而不会产生大量网络开销.它的主要模块包括:

(1) 请求解释器:负责将应用程序表示的高级基元转换为流级基元;

(2) 调度程序:根据从应用程序收到的请求确定要轮询的统计信息类型,再由调度算法确定轮询时间戳;

(3) 交换机选择器:确定要轮询的交换机集,以便在计划时间戳中获取所需的统计信息;

(4) 聚合器和数据存储:负责从所选交换机收集和存储原始数据.

Athena[40]提供了一组组件,以构成友好的开发环境.Athena 导出高级 API,从而允许开发人员以与底层 SDN 无关的方式创建异常检测应用程序.Athena 提供控制平面和数据平面功能抽象,实现快速原型设计并最大限度地降低部署成本.如图 11 所示,Athena 框架由 3 类元素组成:南向组件、分布式数据库集群和计算集群以及北向组件.

- 南向组件用于监视网络行为,包括控制平面,从 SDN 控制消息中提取特征,实现实时检测算法,并调用缓解机制;
- 北向组件提供高级 API,以使应用程序能够执行异常检测任务;
- 第 3 类元素包括提供网络特征访问的分布式数据库集群和用于运行 Athena 应用程序的分布式并行计算集群.

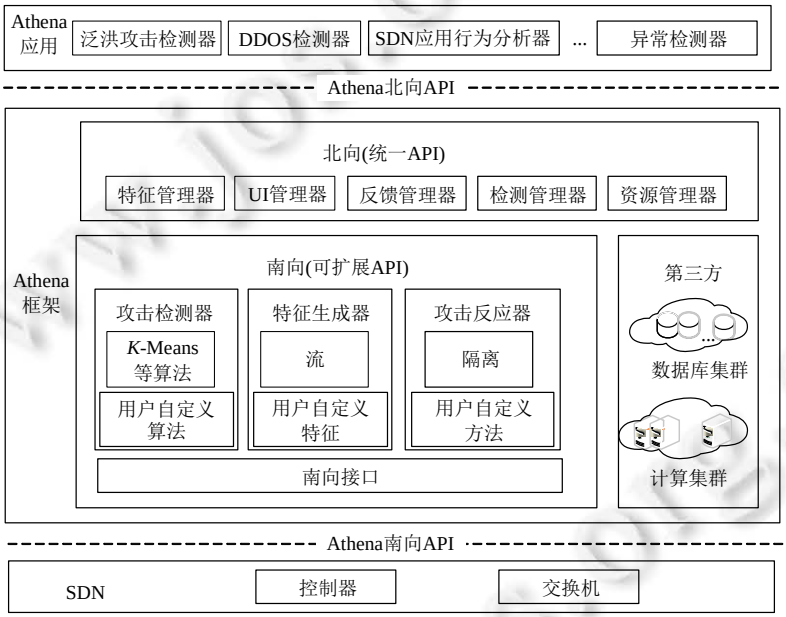

Fig.11 Abnormal traffic detection framework of Athena[40]

图 11 Athena 异常流量监测框架[40]

因为作为中间平台的异常流量检测框架需要网络管理人员通过编写程序调用其相关组件才能实现异常流量的识别和分析,所以提高了 SDN 管理的复杂度,同时对网络管理员自身提出较高的要求.此外,部分中间平台框架不提供复杂计算、统计分析和深度数据包解析(deep packet inspection)的基本组件,这些需要通过软/硬件结合来解决.

4 SDN 异常流量检测的主要机制

SDN 异常流量检测机制是指异常流量检测过程中所使用的主要功能,主要包括异常流量识别机制、负载均

衡机制、异常流量追溯机制和异常流量缓解机制.

4.1 异常流量识别机制

异常流量识别机制一般是指通过各类算法来对流量特征进行提取,并根据提取出的流量特征来对流量进行分类,从而识别异常的流量.同时,根据异常流量的表征来推断网络可能遭受的攻击,并针对不同的攻击采取相应的安全防御措施.因此,在控制平面和数据平面中的异常流量监测和识别,是 SDN 网络安全的关键功能.

(1) 轻量级异常流量识别算法

轻量级异常流量识别算法是指算法复杂度较低、算法效率较高的异常流量识别算法,但是该类算法不具备学习性,可识别的异常流量种类较少,功能性较低.轻量级异常流量识别主要包括基于参数统计的识别算法、基于信息熵的识别算法和基于标签统计的识别算法.

基于参数统计的异常流量识别算法是指在流量检测系统中设置与网络流量特征相关的参数,并收集一段时间内正常流量的参数数据,进行处理后作为统计阈值;在流量检测过程中,对特征参数进行观察与统计,如果超过设定的阈值,则被认定为异常流量.例如,在 Poster 方案[41]中,监测 SDN 网络链接数据、服务延迟数据和吞吐量等参数,并将监测到的数据与预设定的阈值相比较,如果超出阈值,则被认为是异常流量,并将异常流量警报发送至调度程序.何亨等人[42]将网络数据流中数据分组的 IP 头部和 TCP 头部中一些属性值作为统计参数,当特定属性值的出现频率超过预定的阈值时,则判定有异常流量的发生.Gkountis 等人[43]则在 SDN 的控制器中部署一张统计表来统计每个 IP 地址数据流中的数据包数量和每个流中数据包的大小,并计算统计值的中间值,将中间值与阈值比较,从而确定 SDN 中是否有分布式拒绝服务(distributed denial of service,简称 DDoS)攻击.

信息熵是信息论的重要概念,它可以度量网络数据相关联度的不确定性或随机性[44].很多研究表明的不确定性或随机性的网络流量具有自相似、长相关和重尾分布等分布特征,信息熵值正好可以描述这些特征,因此能够通过分析流量信息熵特征的变化来检测流量异常[45,46].信息熵已成为网络异常流量检测的重要手段之一.在 SDN 网络中,由于转发单元主要负责数据流的转发,不具备复杂计算的能力,而控制器负责网络控制具有较强的计算能力,所以很多研究中,异常流量检测需要从交换机收集流表,并在控制器中进行异常检测.但在大规模网络中,流表的收集过程又会对交换机和控制器之间的通信造成过载.使用信息熵识别算法可以解决这一困难,Mousavi 等人[47]在控制器中实施基于目的 IP 地址的熵变化的异常流量识别算法可以检测早期阶段的 DDoS 攻击,此方法能够在攻击流量的前 500 个数据包内检测 DDoS 攻击.Wang 等人[48]在 SDN 的边界网络中运行的基于熵的轻量级 DDoS 洪泛攻击检测模型实现了分布式 SDN 中的快速异常检测,并减少了控制器在流收集过程中的过载情况.

基于标签统计的异常流量识别算法是指在转发单元的流条目中增加流标签,通过流标签来识别和统计出入本网络的不同数据流,从而对异常流量进行识别.基于标签统计的异常流量识别算法可以实现不同聚合级别的流统计,以提高异常流量的检测精度.PathMon 方案[49]将流和路径信息编码为标签,以提供相应的链路到链路的相关性和特定流路径统计的灵活性,用于进一步的异常检测或流量工程任务.Wang 等人[50]扩展了流入口计数器并增加了入口标签,提出了一种基于 OpenFlow 的确定性入口分组标签的分布式攻击检测模型.该模型可以通过入口标签进行 IP 追溯,并在攻击源处过滤恶意流量,以较低的误报率实现较高的检测精度.FADE 方案[51]中使用标签统计机制来识别 SDN 的转发异常,它利用流标签以识别不同的流,并使用专用流规则来收集最小流集合的统计信息,通过分析统计信息以识别转发异常.

(2) 重量级异常流量识别算法

重量级异常流量识别算法是指算法复杂度大、算法精度高、识别的攻击类型广、功能性强的异常流量识别算法,其中,主要包括机器学习、数据挖掘等算法.该类算法可以通过训练数据集来自动构建模型,并根据流量特征对流量进行分类识别,根据历史数据对网络攻击进行检测[52].但是该类算法效率较低,当从高速网络的大量流量和噪声数据中准确地识别异常模式时,算法实现速度较慢[53],往往要求检测设备具有较高的计算能力.表 2 列出了应用在 SDN 中的基于机器学习的主要异常流量识别算法的优缺点和相应的文献.

Table 2 Comparison of abnormal traffic identifications in SDN based on machine learning

表 2 基于机器学习的 SDN 异常流量识别主要研究对比

算法名称	优点	缺点	主要文献
神经网络	能够从有限、嘈杂及不完整的数据计算出最优解,而不需要专家知识,它可以找到未知或新的网络攻击	训练过程缓慢,不适合实时检测,且容易造成过度拟合	[54−56]
贝叶斯	对变量之间的关联性概率关系进行编码,该算法能够结合先前的数据和知识对网络攻击进行预测	对于流量的连续特征难以处理,如果先验知识错误,可能造成完全错误的流量分类	[57−59]
支持向量机(SVM)	具有良好的小样本学习能力和较高的决策率以及对输入数据维度的不敏感性	需要较长的数据训练时间,大多数 SVM 算法使用的二进制分类器,不能给出关于检测到的攻击类型的附加信息	[60−62]
模糊算法	通过求解近似值来代替精确值,从而降低算法的复杂度,该算法对于探针攻击和端口扫描较为有效	规则的动态更新较为困难,无法精确分类和识别部分网络攻击	[63,64]
最邻近规则分类(*K*NN)	通过周围有限的邻近的样本来确定所属类别,因此能够对流量数量和时间等参数进行简单、有效的分类,从而检测出异常流量,此外,通过对参数 *K* 的选择具备丢弃噪音数据的健壮性	当训练样本分布不均匀时,异常检测结果会有较大误差;算法复杂度高,需要计算待检测数据与全体已知样本之间的距离,因此单纯使用 *K*NN 算法时不适用于异常流量实时检测	[67,68]

神经网络算法可以较精确地检测异常流量或网络入侵攻击.Damian 等人[54]提出了基于人工神经网络的入侵检测系统,特征生成器从细粒度数据流中提取流量统计信息并进行处理形成流量特征,传输至分类器,由分类器检测数据平面中的恶意活动.控制器监控特征生成器和分类器的操作,并对流量分类的结果进行可视化.自组织特征映射(self-organizing feature map,简称 SOM)和学习矢量量化(learning vector quantization,简称 LVQ)等多个神经网络算法在该系统中进行了测试.而 Tuan 等人[55]构建了基于深度神经网络模型的入侵检测系统,该模型中使用 NSLKDD 数据集进行训练,并通过 SDN 中获得的网络链接时间长度、网络协议类型、从源地址到目的地址的数据字节数、从目的地址到源地址的数据字节数、两秒内同一主机的连接数、两秒内同一服务的连接数这 6 个基本特征来进行 SDN 异常流量检测.此外,他们还将门控递归单元-递归神经网络算法(gated recurrent unit-recurrent neural network)[56]引入至深度数据包检测系统,从而进一步提高了异常流量的检测率.

贝叶斯算法可以预测网络攻击,从而可以定义 SDN 控制器的安全规则,防止恶意用户访问网络.

Saurav 等人[57]使用历史数据来训练基于贝叶算法的模型,通过训练模型来识别潜在易受攻击的主机,并且根据 SDN 控制器中的数据计算网络安全规则.Osama 等人[58]则使用贝叶斯算法来检测 SDN 中的僵尸网络,他们将贝叶斯推理引擎作为严格的、自适应的数学方法来建模僵尸网络生命周期事件之间的因果关系,从而计算每个主机的感染可能性.Huan 等人[59]使用控制器收集 ARP 数据包以建立全局 MAC-IP 映射信息的知识,然后使用贝叶斯算法来检测 ARP 攻击和 DDoS 攻击.

支持向量机(support vector machine,简称 SVM)的泛化能力较强,能够实现较小的分类误差且效率相对较高.Kokila[60]和 Wang[61]等人均使用 SVM 进行异常流量检测.

① 数据训练阶段:执行数据预处理和签名分析.预处理包括:将符号特征转换为数值;归一化标度;特征数据被缩放到落入范围[−1,1];攻击类型被识别为两个类别之一,0 为正常,1 为攻击.在签名分析中,从原始特征中选择不同的特征子集,以评估合格子集;

② 检测阶段:使用基于行为分析的模式来识别和分类 SDN 网络的入侵攻击.

为了进一步提高 SVM 分类的精确性,Li 等人[62]认为影响 SVM 分类效果的两个参数是惩罚参数和核函数参数,并通过遗传算法 GA 来寻找惩罚参数和全局最优解参数,然后使用具有最优参数的 SVM 来检测 SDN 的 DDoS 攻击分组.

基于模糊算法使用模糊数学隶属度理论,利用综合评价法将定性评价转化为定量评价,即,使用模糊综合评价方法来对由许多因素限制的事物或对象进行总体评价.该算法具有结果清晰、规则性强的特点,适合解决网络攻击检测中各种不确定性问题.Qiao 等人[63]通过模糊算法对 DDoS 攻击程度进行了综合评价.Dotcenko 等人[64]则将 TRW-CB 算法[65]、速率限制算法[66]与模糊算法相结合,从而识别恶意扫描攻击.其中,TRW-CB 算法是基于规律:对于良性主机,连接尝试成功的概率应该比恶意主机高得多;速率限制算法则是基于规律:在端口扫

描或病毒传播的过程中,恶意主机试图在短时间内连接到各种主机.

最邻近规则(*k*-nearest neighbor,简称 *K*NN)分类算法是通过测量不同特征值之间的距离进行分类.该算法简单有效,但是当训练样本分布不均匀时,异常检测结果会有较大误差,且复杂度较高.Latah 等人[67]在 SDN 环境中比较了神经网络、贝叶斯、SVM 和 *K*NN 等多个基于机器学习的异常流量检测算法,通过实验观察到,*K*NN 算法具有最高的准确性和较大的时间开销.Peng 等人[68]将置信机器(transductive confidence machines,简称 TCM)算法与 *K*NN 算法相结合,提出一种双概率值的 TCM-KNN 算法,实现 SDN 异常流量监测,同时提高了异常流量的检测效率与精度.

4.2 负载均衡机制

异常流量检测过程中,交换机定时地给监视器发送流量信息,这些信息一方面会引起网络过载;另一方面,由于SDN逻辑集中特性,使得控制器或检测器在异常流量检测过程中的工作负载过大而易形成SDN瓶颈.这些将导致网络性能的降低,甚至合法数据分组的丢失.网络监测过程中,抽样和聚合方法常用来避免基础设施的过载,然而这两种方法精确度不高,可能影响异常流量检测的准确性.因此,需要基于负载均衡的 SDN 异常流量检测机制来平衡网络负载和网络异常检测的精确性.表 3 所示为目前 SDN 异常流量检测过程中的负载均衡类别和所涉及的文献.

Table 3 Classification of load balancing mechanisms

表 3 负载均衡机制分类

均衡方法	主要均衡范围	作用	主要文献
自适应流收集法	SDN 负载均衡	减轻整个 SDN 的网络负载,其中包括控制器的工作量、交换机发送的数据量和网络链路传输的数据量	[69−71]
网络切割分配法	控制器负载均衡	主要减轻单个控制器的工作量,避免控制器成为网络瓶颈	[72,73]

SDN 异常流量检测过程中的负载均衡机制主要有两类:自适应流收集法和网络切割分配法.自适应控制包含两个方面.

1) 根据分组头部字段控制空间聚合,根据监测情况对现有聚合规则加以放大或缩小;

2) 调整交换机的信息报告时间,根据监测情况延长或缩短交换机的信息报告时间.

自适应流收集方法减少了网络内发送数据的数量,可以解决 SDN 异常流量检测过程中,整个网络的负载均衡问题.网络切割分配法主要是将所检测的网络进行划分并分配给多个控制器或检测器,由于该方法并没有减少发送数据的数量,主要是减轻了单个控制器或检测器的负担.

对于第 1 类方法,Zhang 等人[69]使用 OpenWatch 来监视不同的流量组和更改每个报告的聚合持续时间.当有异常迹象时,OpenWatch 则缩短监测时间间隔,生成具有较短持续时间的记录,并利用线性预测公式计算聚合数据流的预测值,然后与阈值范围相比较:如果预测值小于范围最低值,则设置对应于该聚合的扩展标志;如果值高于范围最高值,则设置缩小标志.然后,根据设置的标志进行自适应聚合操作.Garg 等人对 Zhang 的自适应算法进行了改进[70],不需要设置标志,直接根据比较值来进行聚合操作,以降低算法的复杂性.但是,当合法用户在网络内发送大量数据流并引起数据流计数的较大波动时,该算法将此视为异常流量,并且引起警报,因此, Garg 等人又提出计算动态阈值范围来进行数据流聚合规则的动态更新[71].

对于第 2 类方法,Hark 等人[72]提出了基于协同流量矩阵的分布式协作控制平台 DISTTM,减轻了单个控制器的监控负担.DISTTM 由安装在多个控制器上的协作模块组成,它们之间交换消息以计算控制器上的流量矩阵.分布式多控制器的具体协作机制是:当某个控制器的监控流量达到阈值时,协调器就会收集所在网络的所有控制器的监控信息,根据所收集的信息来计算和分配每个控制器的监控范围.Abaid 等人[73]则提出了MalwareMonitor负载均衡异常流量检测机制,如图12所示.该机制在控制器上部署负载均衡模块,该模块将网络逻辑分割成“切片”,“切片”是指具有特定特性(例如,特定端口、协议或源/目的地址)的所有业务;然后在数据平面中创建流规则,并把来自每个“切片”的所有分组的副本转发到相应的检测器节点;检测器对接收到分组进行深度数据包解析,并将检测结果发送至决策模块;决策模块则向数据平面传送流规则以阻止网络攻击.

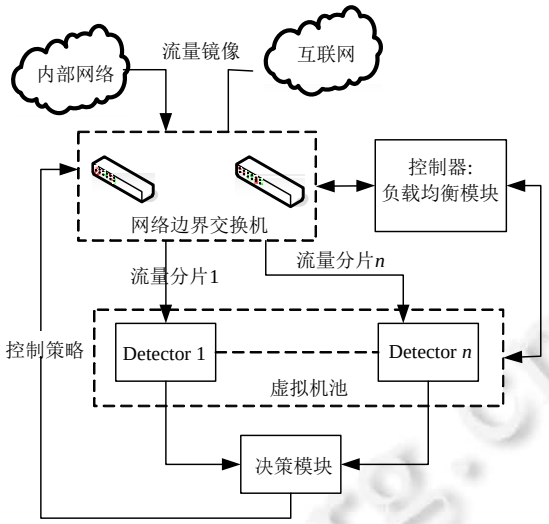

Fig.12 Architecture of MalwareMonitor[73]

图 12 MalwareMonitor 架构[73]

4.3 异常流量追溯机制

SDN 异常流量检测过程中,通过特殊的方式跟踪异常流量的传输路径,搜寻流量的真实来源,不仅可以减轻网络攻击,还可以找到攻击者的确切位置,从源头上阻止网络攻击的进行.表 4 列举了 SDN 异常流量追溯机制的主要研究内容及其优缺点.

Table 4 Comparison of SDN abnormal flow traceback mechanisms

表 4 SDN 异常流量追溯机制的主要研究对比

文献	主要技术	优点	缺点
[74]	基于上下文的编码技术	使用分组的头部较小空间,对流量本身影响较小	在溯源异常流量过程中,相关的交换机会产生巨量的流条目
[75]	CherryPick 路径选择,路径 ID 标识和路径重建	交换机溯源规则可扩展性强;相关的交换机在溯源过程中产生流条目数量相对较小	当发生数据包丢失时,由于 CherryPick 选择性地挑选路径,无法精确识别分组丢弃的位置
[76]	基于有向图模型的 OpenFlow 技术	无需查询 IP 地址就可以溯源攻击流量的所有路径	需要对大量的报头字段和流条目的字段进行匹配
[77]	反向策略,NetCore 技术	可以跨自治域进行流量溯源,溯源方法具有较强的扩展性	在控制器端需要进行较复杂的计算
[78]	标记技术,全局网络拓扑计算	能有效抵御 DDoS 攻击,算法复杂度较低	溯源的精确度较低,易将合法的大数据量传输当作非法攻击
[79]	升级网络设备增加时钟同步和本地流记录日志,并聚合本地日志形成全局日志	能够定量地重现数据平面的数据流传输过程和重现控制平面的控制事件交互过程	需要对 SDN 网络设备进行升级改造,成本较高

如图 13 所示为异常流量追溯一般模型.目前,在 SDN 中异常流量追溯的研究主要集中在通过交换机根据控制的标记规则对可疑流量进行标记和统计,并将可疑的流量信息传输至控制器;控制器对异常流量的网络拓扑信息进行编辑,通过触发机制来进行流量溯源.

例如,Zhang 等人[74]提出了 PathLetTracer 流量溯源机制,其中,控制器对所有的数据包转发路径进行编码,称为“上下文编码”;然后,交换机将该编码作为流量的 ID 标识增加到流条目中,由用户查询作为触发机制;之后,控制器对编码进行译码,根据网络拓扑对流量进行溯源.Tammana 等人[75]针对前者交换机流条目过多的缺点提出了 CherryPick 流量溯源机制,对于每个分组,交换机具有简单的链路选择规则(可转化为 OpenFlow 机制),如果分组匹配到其中一个规则,就选择相应的入口链路,并将相应的 ID 标识嵌入到分组中,然后由控制器根据 ID 标

识重建网络路径进行溯源.

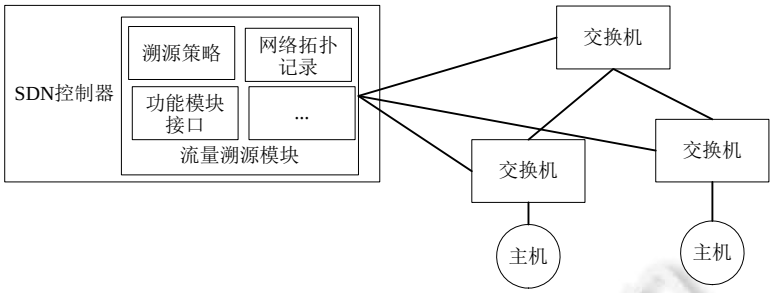

Fig.13 General model of abnormal flow traceback

图 13 异常流量追溯的一般模型

对于网络攻击的自动溯源,Francois 等人[76]则利用控制器掌握全局拓扑的优点,根据网络有向图,利用流量特征(即源/目的 MAC 地址、源/目的 IP 地址、源/目的端口号和协议类型),在发现攻击的交换机上执行路径溯源.而 Zhang 等人[77]主要使用反向策略进行攻击流量追溯:控制器计算攻击流量的转发策略及其反向策略;反向策略将被迭代地应用于定位的分组及其相关联的分组集合,直到反向策略发现被丢弃的分组或者到达网络边界处的入口链路为止.Cui 等人[78]针对 DDoS 攻击,利用控制器记录和计算每个交换机的流条目总数量和恶意流条目的比例,来确定并标记恶意交换机;使用全局网络拓扑、攻击目的地和标记的恶意交换机的组合来计算攻击流量通过的网络路径.翁溪等人[79]则通过全局日志对数据进行溯源,首先需要对网络设备进行升级,增加时钟同步和本地日志记录功能,然后对分布式的本地日志进行预处理去除冗余信息,并对本地日志进行聚合形成按时间戳排序的全局日志;通过全局日志可以定量地重现数据平面的数据流传输过程,从而对异常流量进行溯源.

4.4 异常流量缓解机制

SDN 异常流量缓解机制是指在 SDN 发现异常流量之后,网络设备自动地或者网络管理员人为地对异常流量采取措施,防止异常流量进一步破坏网络,维护网络的正常运行.异常流量缓解是异常流量检测过程中的重要步骤,是阻止网络攻击的最后防线.SDN 异常流量缓解机制的主要技术及其优缺点见表 5.

Table 5 Classification of SDN abnormal flow mitigation mechanisms

表 5 SDN 异常流量缓解机制的主要技术分类

异常流量缓解技术	优点	缺点	主要文献
丢弃攻击流量	操作简单,效率高	对合法业务流影响较大,易造成合法业务流的丢失	[80−83]
基于流聚合的弹性缓解方法	分离攻击流和合法业务流,采用弹性机制缓解异常流量的攻击	算法较为复杂,设备开销较大	[84,85]
转发至流过滤器	最大限度地降低对合法业务流的影响	会对网络通信造成额外开销	[86,87]

直接丢弃攻击流量是异常缓解机制最常用的手段,因为该方法网络设置简便,执行效率高.

例如:在方案 FlowTrApp[80]中,如果源地址尝试发送的流的数量超过合法计数器的阈值,则数据流将被丢弃以阻止转发;Giotis[81,82]和 Carvalho[83]等人则在 OpenFlow 交换机的流表中插入流条目(或修改现有的)以阻止恶意流量,流条目中附带特定动作(转发、丢弃和修改字段操作),转发操作附加到每个良性流条目,而将丢弃操作附加到恶意流条目.然而,流量识别大多数情况下不能达到百分之百的精确,识别出的恶意流量可能存在假阳性,即部分合法业务流被误识别为攻击流量.因此,直接丢弃识别出的“恶意流量”对合法业务流的影响较大,易造成合法业务流的丢失,而使部分合法业务无法进行.

基于流聚合的弹性缓解方法不直接丢弃全部的异常流量,而是通过流聚合来分离攻击流和合法业务流,并采用弹性机制来缓解异常流量对网络的攻击.例如,SDN-Guard 方案[84]确保合法流以最小的往返时间转发,而恶意流量则分配较高的延时以防止网络堵塞和服务器崩溃,然后根据流规则中条目的共享属性将恶意流聚合到

相同的链路,以防交换机流表溢出.Kalliola 等人[85]则通过流聚合规则分离出明显的恶意流量并加入至黑名单,根据黑名单丢弃恶意流量;对于具有潜在威胁的流量则被分配到副本服务器,然后通过调用副本服务器资源来进一步缓解攻击.基于流聚合的弹性缓解方法需要对流量执行聚合算法,并对不同级别的流量执行不同的操作,因此会对交换机带来较大的计算开销,执行效率也相对较低.

为了减小异常缓解机制对交换机转发数据包的影响,有些研究提出通过流过滤器来缓解异常流量对网络的影响,即:把识别出的异常流量转发至流过滤器,流过滤器根据检测情况对异常流量采取相应的措施.例如:Sahay 等人[86]对异常流量进行标记,并将其转发到流过滤器,流过滤器根据标记信息采取相应的策略对异常流量进行处理(比如改变其头域),然后再转发至网络,以消除异常流量对网络的威胁.针对控制器的 DDoS 攻击,在基于时间的缓解方案[87]中,控制器将目的地址无效或未知的数据包转发到流过滤器,当流过滤器检测到无效数据包的数量在一定时间内显著增加时,流过滤器通知控制器更新交换机规则,将异常分组直接转发到流过滤器,然后,流过滤器执行 DDoS 攻击的聚类时间模式,以防止下一次 DDoS 攻击.虽然异常流量处理开销从交换机或控制器转移至流过滤器,但是因为流过滤器需要与交换机和控制器频繁通信,也会给网络造成额外的负担.

对于跨域的分布式网络攻击,单个 SDN 网络异常流量缓解技术并不能完全抵御恶意流量的攻击,因为相互连接的 SDN 网络之间都有可能存在恶意流量的传输.因此,Giotis 等人[88]提出了基于 SDN 控制器的合作攻击缓解机制(cooperative attack mitigation mechanism,简称 CAMM).在该机制中,相邻 SDN 域通过 SDNi(SDN interface)协议交换信息,一旦某个 SDN 网络检测到攻击,就会向其相邻域发出通知,请求他们对攻击流进行过滤直到攻击停止,并向上游传播信息,从而尽力达到服务攻击的源 SDN 域.

此外,即使攻击流被成功阻止,由攻击流生成的大量恶意流条目仍存在于交换机中.因此,在异常缓解之后需要删除恶意流条目,释放交换机的存储资源.此时要借助溯源机制找到恶意流的传输路径并删除相关流条目.

5 未来研究方向

虽然现在已有各个研究团队对 SDN 异常流量检测做了大量的研究工作,提出了很多有效的系统框架和管理机制,但网络攻击呈现出多样化、复杂化、规模化等特点,安全问题已成为制约 SDN 发展的关键因素.因此,未来研究工作应该包括以下几项.

(1) 构建功能完整的 SDN 异常流量检测框架.

在未来的工作中,应该在研究现有框架的基础上建立具有灵活性、准确性、功能完整性和高效性等特点的 SDN 异常流量检测框架.因此,需要在控制层细化和抽象 SDN 控制器的功能模块,使其进一步支持网络流量的检测;在应用层建立功能完整并支持多种检测算法的异常流量检测应用,该应用能够对各类异常流量进行自动化处理,同时支持与网络管理员的交互操作,因为管理员可能会更新参数或替换某些组件功能,以提高检测的准确性;需要对检测功能模块进行抽象并提供高级 API,使得网络管理人员能够针对复杂的网络情况进行二次开发,对已有框架功能进行补充和改进;该框架应包含异常流量特征集数据库,并支持动态更新功能,一旦系统检测并处理新类型的网络攻击时,能够对数据库文件进行更新.

(2) 提高 SDN 异常流量检测效率.

SDN 异常流量检测效率包括两方面:时间效率和准确率.现有的 SDN 异常流量检测机制主要是将传统的检测方法部署到 SDN 中,然而由于 SDN 与传统网络架构特征有所不同,这对检测效率会产生影响.因此,针对 SDN 的网络特征,在流量监测阶段需要权衡监测的精确度与网络通信效率,在流量检测阶段需要权衡检测算法的复杂度与误报率.此外,还需要进一步降低现有检测算法中的预处理、特征训练和异常识别等过程中的算法复杂度,以提高 SDN 异常流量的检测效率.

(3) 研究在云计算环境中的 SDN 异常流量检测所面临的挑战.

云计算实现了网络资源的灵活共享、用户业务的快速开发与部署,并且能够根据业务需求进行自动部署、弹性伸缩和故障隔离等.因为云计算中广泛的网络访问,使得云计算网络容易受到异常流量的攻击,尤其是 DDoS 攻击.而传统的流量检测技术因为缺乏灵活性和可编程性,导致云计算网络中异常流量检测的精确度和

效率较低,且开发成本高昂.而基于 SDN 的异常流量检测技术提供了新的动态网络架构,将传统的云网络骨干网转变为功能丰富的服务平台.然而由于 SDN 本身逻辑集中管理性和交换机资源有限性等特点,云环境中使用 SDN 进行异常流量检测也面临着诸多挑战.因此需要结合云计算网络和 SDN 的网络特征,研究在云计算环境中的 SDN 异常流量检测所面临的挑战,其中包括:在动态的云计算环境中如何使用 OpenFlow 技术进行流量监测;如何使用 SDN 实现云计算环境中的应用层攻击流量的检测;如何实现在云计算环境中的 SDN 异常流量检测过程中的网络数据的隐私保护;在云计算环境中如何防止控制器成为网络瓶颈等.在研究这些挑战的基础上,再进一步研究相应的安全措施,从而提高 SDN 异常流量检测技术在云计算环境中的安全性、可靠性与稳定性.

(4) 研究自适应的 SDN 异常流量检测机制.

入侵者往往会调整其网络攻击以逃避现有的检测方案,流量异常的性质也会随着时间的推移而不断变化,因此,研究自适应的 SDN 异常流量检测机制也就成为重要课题.首先,需要研究自适应的抽样算法,该算法能够根据 SDN 网络流量变化情况自适应地调整抽样频率,在确保抽样精确度的同时降低网络开销;其次,需要研究自适应的异常流量识别算法,该算法需要将轻量级识别算法和重量级识别算法层次化地、有机地结合在一起,例如通过轻量级识别算法来识别流量异常与否,再通过重量级识别算法为异常流量进行分类,其中,需要为每类异常流量开发适当且快速的特征选择方法,并改进现有的分类器,使其能够高效、准确地对异常流量进行分类;最后,根据识别出的网络攻击类型及其网络状况,选择恰当的异常流量处理方案.

(5) 研究跨域分布式的 SDN 异常流量检测机制.

现有的研究工作主要集中在单个控制器在局域网中的异常流量检测,但对于规模较大的跨域网络攻击,单个控制器难以溯源,无法从源头进行控制.为了有效防御跨域网络攻击并减轻单个控制器的工作负载,提高 SDN 的可扩展性,跨域分布式 SDN 异常流量检测机制具有重要的研究意义.未来工作中,需要将云计算技术与 SDN 异常流量检测技术相结合,将异常流量检测框架和异常流量特征数据库部署在云端,将流量监测功能模块部署在各控制器中,通过云端与控制器的实时通信来收集各局域网的网络信息(其中,包括流量信息和网络拓扑信息等),并需要在云端开发异常流量并行检测算法.此外,还需要进一步研究云端服务器如何与控制器有效通信,在收集有效信息的同时,尽可能减少通信开销,从而避免网络过载.

(6) 研究大数据环境中的数据中心网络的 SDN 异常流量检测方法.

随着信息技术的发展,数据数量不断增加,网络规模不断扩大,网络性能要求不断提高,这就需要研究大数据环境中的数据中心网络的 SDN 异常流量检测方法.首先,在研究软件定义网络的通信协议和通信架构的基础上对 SDN 协议进行扩展,改进 SDN 转发规则,使其能够支持海量数据在 SDN 中的转发和网络拓扑管理;其次,在研究 OpenFlow 流表特性的基础上,实现数据中心网络中并行化的流量特征提取,建立特征参数总集,并对特征总集进行降维优化得到最优参数集;此外,在研究各类异常流量识别算法的基础上,提出能够支持并行计算的异常流量检测算法,并实现异常流量检测过程中的负载均衡,防止在数据中心网络中进行异常流量检测而导致网络堵塞.

6 总 结

随着互联网、云计算和大数据等信息技术的发展,软件定义网络架构的开发和部署也不断深入.SDN 在提高网络可扩展性、灵活性、可编程性的同时也面临诸多的网络安全威胁,SDN 异常流量检测技术可以防御恶意流量攻击,保障网络安全.本文在分析数据平面和控制平面的网络攻击及其导致的流量变化的基础上,对现有的 SDN 异常流量检测技术进行了有益探索,首先分析了位于应用平面、控制平面和中间平台这 3 类 SDN 异常流量检测框架的主要技术及其优缺点,总结了异常流量识别机制、负载均衡机制、异常流量追溯机制、异常缓解机制的主要实现方法.目前,SDN 异常流量检测技术的研究还处在起步阶段,例如:现有的异常流量的框架的研究主要集中在单个 SDN 网络,对于跨域大型 SDN 网络的研究相对较少;识别算法的应用相对比较单一,无法识别动态变化的网络攻击.所以面对不断扩大的网络规模和复杂多样的网络攻击,需要建立功能完整的 SDN 异常流量检测框架,提高检测效率,分析云计算环境中 SDN 异常流量检测所面临的挑战,并进一步研究自适应、跨域

分布式以及大数据环境中的 SDN 异常流量检测机制.

References:

[1] Kim H, Benson T, Akella A, Feamster N. The evolution of network configuration: A tale of two campuses. In: Proc. of the 2011 ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2011. 499−514. [doi: 10.1145/2068816.2068863]

[2] Kreutz D, Ramos FM, Verissimo P, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. Proc. of the IEEE, 2015,103(1):14−76. [doi: 10.1109/JPROC.2014.2371999]

[3] Zhang CK, Cui Y, Tang HY, Wu J. State-of-the-art survey on software-defined networking (SDN). Ruan Jian Xue Bao/Journal of Software, 2015,26(1):62−81 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4701.htm [doi: 10.13328/j.cnki.jos.004701]

[4] Scott-Hayward S, O'Callaghan G, Sezer S. SDN security: A survey. In: Proc. of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS). Piscataway: IEEE, 2013. 1−7. [doi: 10.1109/SDN4FNS.2013.6702553]

[5] Bhuyan MH, Bhattacharyya DK, Kalita JK. Network anomaly detection: Methods, systems and tools. IEEE Communications Surveys & Tutorials, 2013,16(1):303−336. [doi: 10.1109/SURV.2013.052213.00046]

[6] Marnerides AK, Schaeffer-Filho A, Mauthe A. Traffic anomaly diagnosis in Internet backbone networks: A survey. Computer Networks, 2014,73:224−243. [doi: 10.1016/j.comnet.2014.08.007]

[7] Nadeau TD, Gray K. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. Sebastopol: O'Reilly Media, Inc., 2013. 117−156. [doi: 9781449342302]

[8] Kim H, Feamster N. Improving network management with software defined networking. IEEE Communications Magazine, 2013, 51(2):114−119. [doi: 10.1109/MCOM.2013.6461195]

[9] Tri HTN, Kim K. Assessing the impact of resource attack in software defined network. In: Proc. of the 2015 Int'l Conf. on Information Networking. Piscataway: IEEE, 2015. 420−425. [doi: 10.1109/ICOIN.2015.7057934]

[10] Bian S, Zhang P, Yan Z. A survey on software-defined networking security. In: Proc. of the 9th EAI Int'l Conf. on Mobile Multimedia Communications. Brussels, 2016. 190−198.

[11] Nunes BAA, Mendonca M, Nguyen XM, Obraczka K, Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials, 2014,16(3):1617−1634. [doi: 10.1109/SURV.2014.012214.00180]

[12] Francois J, Dolberg L, Festor O, Engel T. Network security through software defined networking: A survey. In: Proc. of the 2014 Conf. on Principles, Systems and Applications of IP Telecommunications (IPTComm). New York: ACM Press, 2014. 1−8. [doi: 10.1145/2670386.2670390]

[13] Zhou TQ, Cai ZP, Xia J, Xu M. Traffic engineering for software defined networks. Ruan Jian Xue Bao/Journal of Software, 2016, 27(2):394−417 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4935.htm [doi: 10.13328/j.cnki.jos.004935]

[14] Kalkan K, Gur G, Alagoz F. Defense mechanisms against DDoS attacks in SDN environment. IEEE Communications Magazine, 2017,55(9):175−179. [doi: 10.1109/MCOM.2017.1600970]

[15] Garg G, Garg R. Review on architecture and security issues in SDN. Int'l Journal of Innovative Research in Computer and Communication Engineering, 2014,2(11):6519−6524.

[16] Shin S, Gu G. Attacking software-defined networks: A first feasibility study. In: Proc. of the 2nd ACM SIGCOMM Workshop on Hot topics in Software Defined Networking. New York: ACM Press, 2013. 165−166. [doi: 10.1145/2491185.2491220]

[17] Alsmadi I, Xu D. Security of software defined networks: A survey. Computers & Security, 2015,53:79−108. [doi: 10.1016/j.cose.2015.05.006]

[18] Lin C, Wu C, Huang M, Wen Z, Cheng Q. Adaptive ip mutation: A proactive approach for defending against worm propagation. In: Proc. of the 35th IEEE Symp. on Reliable Distributed Systems Workshops (SRDSW). Piscataway: IEEE, 2016. 61−66. [doi: 10.1109/SRDSW.2016.21]

[19] Hong S, Xu L, Wang H, Gu G. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In: Proc. of the Network and Distributed System Security Symp. Internet Society, 2015. 1−15. [doi: 10.14722/ndss.2015.23283]

[20] Scott-Hayward S, Natarajan S, Sezer S. A survey of security in software defined networks. IEEE Communications Surveys & Tutorials, 2016,18(1):623−654. [doi: 10.1109/COMST.2015.2453114]

[21] Zhang K, Qiu X. CMD: A convincing mechanism for MITM detection in SDN. In: Proc. of the 2018 IEEE Int'l Conf. on Consumer Electronics (ICCE). Piscataway: IEEE, 2018. 1−6 [doi: 10.1109/ICCE.2018.8326334]

[22] Wang MM, Liu JW, Chen J, Mao J, Mao KF. Software defined networking: Security model, threats and mechanism. Ruan Jian Xue Bao/Journal of Software, 2016,27(4):969−992 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5020.htm [doi: 10.13328/j.cnki.jos.005020]

[23] Khairi MHH, Ariffin SHS, Latiff NMA, Abdullah AS, Hassan MK. A review of anomaly detection techniques and distributed denial of service (DDoS) on software defined network (SDN). Engineering, Technology & Applied Science Research, 2018,8(2): 2724−2730.

[24] Ahmad I, Namal S, Ylianttila M, Gurtov A. Security in software defined networks: A survey. IEEE Communications Surveys & Tutorials, 2015,17(4):2317−2346. [doi: 10.1109/COMST.2015.2474118]

[25] Kloti R, Kotronis V, Smith P. OpenFlow: A security analysis. In: Proc. of the 21st IEEE Int'l Conf. on Network Protocols (ICNP), Vol.13. Piscataway: IEEE, 2013. 1−6. [doi: 10.1109/ICNP.2013.6733671]

[26] Feghali A, Kilany R, Chamoun M. SDN security problems and solutions analysis. In: Proc. of the 2015 Int'l Conf. on Protocol Engineering (ICPE) and Int'l Conf. on New Technologies of Distributed Systems (NTDS). Piscataway: IEEE, 2015. 1−5. [doi: 10.1109/NOTERE.2015.7293514]

[27] Prasad AS, Koll D, Fu X. On the security of software-defined networks. In: Proc. of the 4th European Workshop on Software Defined Networks. Piscataway: IEEE, 2015. 105−106. [doi: 10.1109/EWSDN.2015.70]

[28] Scott-Hayward S, Kane C, Sezer S. Operationcheckpoint: SDN application control. In: Proc. of the 22nd IEEE Int'l Conf. on Network Protocols. Piscataway: IEEE, 2014. 618−623. [doi: 10.1109/ICNP.2014.98]

[29] Dhawan M, Poddar R, Mahajan K, Mann V. SPHINX: Detecting security attacks in software-defined networks. In: Proc. of the 2015 Network and Distributed System Security (NDSS) Symp. Reston: Internet Society, 2015. 1−15. [doi: 10.14722/ndss.2015.23064]

[30] Granby BR, Askwith B, Marnerides AK. SDN-PANDA: Software-defined network platform for anomaly detection applications. In: Proc. of the 23rd IEEE Int'l Conf. on Network Protocols (ICNP). Piscataway: IEEE, 2015. 463−466. [doi: 10.1109/ICNP.2015.58]

[31] Phan XT, Fukuda K. SDN-Mon: Fine-grained traffic monitoring framework in software-defined networks. Journal of Information Processing, 2017,25:182−190. [doi: 10.2197/ipsjjip.25.182]

[32] Hu H, Han W, Ahn GJ, Zhao Z. Flowguard: Building robust firewalls for software-defined networks. In: Proc. of the 3rd Workshop on Hot topics in Software Defined Networking. New York: ACM Press, 2014. 97−102. [doi: 10.1145/2620728.2620749]

[33] Kim M, Park Y, Kotalwar R. Robust and agile system against fault and anomaly traffic in software defined networks. Applied Sciences, 2017,7(3):266. [doi: 10.3390/app7030266]

[34] Carvalho LF, Abrao T, de Souza Mendes L, Proenca Jr ML. An ecosystem for anomaly detection and mitigation in software-defined networking. Expert Systems with Applications, 2018,104:121−133. [doi: 10.1016/j.eswa.2018.03.027]

[35] Da Silva AS, Wickboldt JA, Granville LZ, Schaeffer-Filho A. Atlantic: A framework for anomaly traffic detection, classification, and mitigation in SDN. In: Proc. of the 2016 IEEE/IFIP Network Operations and Management Symp. (NOMS 2016). Piscataway: IEEE, 2016. 27−35. [doi: 10.1109/NOMS.2016.7502793]

[36] Bonola M, Bianchi G, Picierro G, Pontarelli S, Monaci M. StreaMon: A data-plane programming abstraction for software-defined stream monitoring. IEEE Trans. on Dependable and Secure Computing, 2015,14(6):664−678. [doi: 10.1109/TDSC.2015.2499747]

[37] Shin S, Porras P, Yegneswara V, Fong M, Gu G, Tyson M. Fresco: Modular composable security services for software-defined networks. In: Proc. of the 20th Annual Network & Distributed System Security Symp. Reston: Internet Society, 2013. 1−16.

[38] Kohler E, Morris R, Chen B, Jannotti J, Kaashoek MF. The Click modular router. ACM Trans. on Computer Systems, 2000,18(3): 263−297. [doi: 10.1145/354871.354874]

[39] Chowdhury SR, Bari MF, Ahmed R, Boutaba R. Payless: A low cost network monitoring framework for software defined networks. In: Proc. of the 2014 IEEE Network Operations and Management Symp. (NOMS). Piscataway: IEEE, 2014. 1−9. [doi: 10.1109/NOMS.2014.6838227]

[40] Lee S, Kim J, Shin S, Porras P, Yegneswaran V. Athena: A framework for scalable anomaly detection in software-defined networks. In: Proc. of the 47th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Piscataway: IEEE, 2017. 249−260. [doi: 10.1109/DSN.2017.42]

[41] Lu Z, Chen F, Wu J, Cheng G. Poster: A secure control plane with dynamic multi-NOS for SDN. In: Proc. of the Network and Distributed System Security Symp. Reston: Internet Society, 2017.

[42] He H, Hu Y, Zheng LH, Xue ZY. Efficient DDoS attack detection and prevention scheme based on SDN in cloud environment. Tong Xin Xue Bao/Journal on Communications, 2018,39(4):139−151 (in Chinese with English abstract). http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2018068 [doi: 10.11959/j.issn.1000-436x.2018068]

[43] Gkountis C, Taha M, Lloret J, Kambourakis G. Lightweight algorithm for protecting SDN controller against DDoS attacks. In: Proc. of the 10th IFIP Wireless and Mobile Networking Conf. (WMNC). Piscataway: IEEE, 2017. 1−6. [doi: 10.1109/WMNC.2017.8248858]

[44] Zhu Z, Wang L, Xia L, Cao S. Flow performance analysis of software defined networking based on cross entropy. In: Proc. of the 8th Int'l Congress on Image and Signal Processing (CISP 2015). Piscataway: IEEE, 2015. 1556−1560. [doi: 10.1109/CISP.2015.7408132]

[45] Navaz ASS, Sangeetha V, Prabhadevi C. Entropy based anomaly detection system to prevent DDoS attacks in cloud. Int'l Journal of Computer Applications, 2013,62(15):42−47. [doi: 10.5120/10160-5084]

[46] Mu XK, Wang JS, Xue YF, Huang W. Abnormal network traffic detection approach based on alive entropy. Tong Xin Xue Bao/Journal on Communications, 2013,34(2):51−57 (in Chinese with English abstract). http://www.infocomm-journal.com/txxb/CN/10.3969/j.issn.1000-436x.2013.Z2.011 [doi: 10.3969/j.issn.1000-436x.2013.z2.011]

[47] Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. In: Proc. of the 2015 Int'l Conf. on Computing, Networking and Communications. Piscataway: IEEE, 2015. 77−81. [doi: 10.1109/ICCNC.2015.7069319]

[48] Wang R, Jia Z, Ju L. An entropy-based distributed DDoS detection mechanism in software-defined networking. In: Proc. of the 2015 IEEE Trustcom/BigDataSE/ISPA. Piscataway: IEEE, 2015. 310−317. [doi: 10.1109/Trustcom.2015.389]

[49] Wang MH, Wu SY, Yen LH, Tseng CC. PathMon: Path-specific traffic monitoring in OpenFlow-enabled networks. In: Proc. of the 8th Int'l Conf. on Ubiquitous and Future Networks (ICUFN). Piscataway: IEEE, 2016. 775−780. [doi: 10.1109/ICUFN.2016.7537143]

[50] Wang R, Zhang Z, Ju L, Jia Z. A novel OpenFlow-based DDoS flooding attack detection and response mechanism in software-defined networking. Int'l Journal of Information Security and Privacy, 2015,9(3):21−40. [doi: 10.4018/IJISP.2015070102]

[51] Pang C, Jiang Y, Li Q. FADE: Detecting forwarding anomaly in software-defined networks. In: Proc. of the 2016 IEEE Int'l Conf. on Communications (ICC). Piscataway: IEEE, 2016. 2021−2026. [doi: 10.1109/ICC.2016.7510990]

[52] Ashraf J, Latif S. Handling intrusion and DDoS attacks in software defined networks using machine learning techniques. In: Proc. of the 2014 National Software Engineering Conf. Piscataway: IEEE, 2014. 55−60. [doi: 10.1109/NSEC.2014.6998241]

[53] Zheng L, Zou P, Jia Y, Han W. Traffic anomaly detection and containment using filter-ary-sketch. Procedia Engineering, 2012,29:4297−4306. [doi: 10.1016/j.proeng.2012.01.661]

[54] Jankowski D, Amanowicz M. On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. Int'l Journal of Electronics and Telecommunications, 2016,62(3):247−252. [doi: 10.1515/eletel-2016-0033]

[55] Tang TA, Mhamdi L, Mclernon D, Zaidi SAR, Ghogho M. Deep learning approach for network intrusion detection in software defined networking. In: Proc. of the 2016 Int'l Conf. on Wireless Networks and Mobile Communications (WINCOM). Piscataway: IEEE, 2016. 258−263. [doi: 10.1109/WINCOM.2016.7777224]

[56] Tang TA, Mhamdi L, Mclernon D, Zaidi SAR, Ghogho M. Deep recurrent neural network for intrusion detection in sdn-based networks. In: Proc. of the 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft). Piscataway: IEEE, 2018. 202−206. [doi: 10.1109/NETSOFT.2018.8460090]

[57] Nanda S, Zafari F, Decusatis C, Wedaa E, Yang B. Predicting network attack patterns in SDN using machine learning approach. In: Proc. of the 2016 IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN). Piscataway: IEEE, 2016. 167−172. [doi: 10.1109/NFV-SDN.2016.7919493]

[58] Haq O, Abaid Z, Bhatti N, Ahmed Z, Syed A. SDN-Inspired, real-time botnet detection and flow-blocking at ISP and enterprise-level. In: Proc. of the 2015 IEEE Int'l Conf. on Communications. Piscataway: IEEE, 2015. 5278−5283. [doi: 10.1109/ICC.2015.7249162]

[59] Ma H, Ding H, Yang Y, Mi Z, Zhang M. SDN-based ARP attack detection for cloud centers. In: Proc. of the 12th IEEE Int'l Conf. on Ubiquitous Intelligence and Computing and the 12th IEEE Int'l Conf. on Autonomic and Trusted Computing and the 15th IEEE Int'l Conf. on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom). Piscataway: IEEE, 2015. 1049−1054. [doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.195]

[60] Kokila RT, Selvi ST, Govindarajan K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In: Proc. of the 6th Int'l Conf. on Advanced Computing. Piscataway: IEEE, 2014. 205−210. [doi: 10.1109/ICoAC.2014.7229711]

[61] Wang P, Lin HC, Lin WH, Chao KM, Lo CC. An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In: Proc. of the 13th Int'l Conf. on e-Business Engineering. Piscataway: IEEE, 2016. 56−63. [doi: 10.1109/ICEBE.2016.020]

[62] Li X, Yuan D, Hu H, Ran J, Li S. DDoS detection in SDN switches using support vector machine classifier. In: Proc. of the 2015 Joint Int'l Mechanical, Electronic and Information Technology Conf. (JIMET 2015). Paris: Atlantis Press, 2015. 344−348. [doi: 10.2991/jimet-15.2015.63]

[63] Yan Q, Gong Q, Deng F. Detection of DDoS attacks against wireless SDN controllers based on the fuzzy synthetic evaluation decision-making model. Adhoc & Sensor Wireless Networks, 2016,33:275−299.

[64] Dotcenko S, Vladyko A, Letenko I. A fuzzy logic-based information security management for software-defined networks. In: Proc. of the 16th Int'l Conf. on Advanced Communication Technology. Piscataway: IEEE, 2014. 167−171. [doi: 10.1109/ICACT.2014.6778942]

[65] Schechter SE, Jung J, Berger AW. Fast detection of scanning worm infections. In: Proc. of the Int'l Workshop on Recent Advances in Intrusion Detection. Berlin: Springer-Verlag, 2004. 59−81.

[66] Williamson MM. Throttling viruses: Restricting propagation to defeat malicious mobile code. In: Proc. of the 18th Annual Computer Security Applications Conf. Piscataway: IEEE, 2002. 61−68. [doi: 10.1109/CSAC.2002.1176279]

[67] Latah M, Toker L. Towards an efficient anomaly-based intrusion detection for software-defined networks. IET Networks, 2018,7(6): 453−459. [doi: 10.1049/iet-net.2018.5080]

[68] Peng H, Sun Z, Zhao X, Tan S, Sun Z. A detection method for anomaly flow in software defined network. IEEE Access, 2018,6: 27809−27817. [doi: 10.1109/ACCESS.2018.2839684]

[69] Zhang Y. An adaptive flow counting method for anomaly detection in SDN. In: Proc. of the 9th ACM Conf. on Emerging Networking Experiments and Technologies. New York: ACM Press, 2013. 25−30. [doi: 10.1145/2535372.2535411]

[70] Garg G, Garg R. Detecting anomalies efficiently in SDN using adaptive mechanism. In: Proc. of the 5th Int'l Conf. on Advanced Computing & Communication Technologies. Piscataway: IEEE, 2015. 367−370. [doi: 10.1109/ACCT.2015.98]

[71] Garg G, Garg R. Accurate anomaly detection using adaptive monitoring and fast switching in SDN. Int'l Journal of Information Technology and Computer Science (IJITCS), 2015,7(11):34−42. [doi: 10.5815/ijitcs.2015.11.05]

[72] Hark R, Stingl D, Richerzhagen N, Nahrstedt K, Steinmetz R. Disttm: Collaborative traffic matrix estimation in distributed SDN control planes. In: Proc. of the 2016 IFIP Networking and Workshops. Piscataway: IEEE, 2016. 82–90. [doi: 10.1109/IFIPNetworking.2016.7497233]

[73] Abaid Z, Rezvani M, Jha S. MalwareMonitor: An SDN-based framework for securing large networks. In: Proc. of the 2014 CoNEXT on Student Workshop. New York: ACM Press, 2014. 40−42. [doi: 10.1145/2680821.2680829]

[74] Zhang H, Lumezanu C, Rhee J, Arora N, Xu Q, Jiang G. Enabling layer 2 pathlet tracing through context encoding in software-defined networking. In: Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking. New York: ACM Press, 2014. 169−174. [doi: 10.1145/2620728.2620742]

[75] Tammana P, Agarwal R, Lee M. Cherrypick: Tracing packet trajectory in software-defined datacenter networks. In: Proc. of the 1st ACM SIGCOMM Symp. on Software Defined Networking Research. New York: ACM Press, 2015. 1−7. [doi: 10.1145/2774993.2775066]

[76] Francois J, Festor O. Anomaly traceback using software defined networking. In: Proc. of the 2014 IEEE Int'l Workshop on Information Forensics and Security (WIFS). Piscataway: IEEE, 2014. 203−208. [doi: 10.1109/WIFS.2014.7084328]

[77] Zhang H, Reich J, Rexford J. Packet traceback for software-defined networks. Technical Report, Princeton: Princeton University, 2015. ftp://ftp.cs. princeton.edu/reports/2015/978.pdf

[78] Cui Y, Yan L, Li S, Xing H, Pan W, Zhu J, Zheng X. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. Journal of Network and Computer Applications, 2016,68:65−79. [doi: 10.1016/j.jnca.2016.04.005]

[79] Weng X, Chen M, Zhang GM, Xu B, Xing CY. Design and implementation of a network measurement and analysis system in OpenFlow networks. Tong Xin Xue Bao/Journal on Communications, 2015,36(3):81−88 (in Chinese with English abstract). http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2015061 [doi: 10.11959/j.issn.1000-436x.2015061]

[80] Buragohain C, Medhi N. FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers. In: Proc. of the 3rd Int'l Conf. on Signal Processing and Integrated Networks. Piscataway: IEEE, 2016. 519−524. [doi: 10.1109/SPIN.2016.7566750]

[81] Giotis K, Androulidakis G, Maglaris V. Leveraging SDN for efficient anomaly detection and mitigation on legacy networks. In: Proc. of the 3rd European Workshop on Software Defined Networks. Piscataway: IEEE, 2014. 85−90. [doi: 10.1109/EWSDN.2014.24]

[82] Giotis K, Argyropoulos C, Androulidakis G, Kalogeras D, Maglaris V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Computer Networks, 2014,62:122−136. [doi: 10.1016/j.bjp.2013.10.014]

[83] Carvalho LF, Fernandes G, Rodrigues JJ, Mendes LS, Proença ML. A novel anomaly detection system to assist network management in SDN environment. In: Proc. of the 2017 IEEE Int'l Conf. on Communications. Piscataway: IEEE, 2017. 1−6. [doi: 10.1109/ICC.2017.7997214]

[84] Dridi L, Zhani MF. SDN-guard: DoS attacks mitigation in SDN networks. In: Proc. of the 5th IEEE Int'l Conf. on Cloud Networking (Cloudnet). Piscataway: IEEE, 2016. 212−217. [doi: 10.1109/CloudNet.2016.9]

[85] Kalliola A, Lee K, Lee H, Aura T. Flooding DDoS mitigation and traffic management with software defined networking. In: Proc. of the 4th IEEE Int'l Conf. on Cloud Networking (CloudNet). Piscataway: IEEE, 2015. 248−254. [doi: 10.1109/CloudNet.2015.7335317]

[86] Sahay R, Blanc G, Zhang Z, Debar H. Towards autonomic DDoS mitigation using software defined networking. In: Proc. of the 2015 NDSS Workshop on Security of Emerging Networking Technologies. Reston: Internet society, 2015. 1−7. [doi: 10.14722/sent.2015.23004]

[87] Dharma NIG, Muthohar MF, Prayuda JDA, Priagung K, Choi D. Time-based DDoS detection and mitigation for SDN controller. In: Proc. of the 17th Asia-Pacific Network Operations and Management Symp. (APNOMS). Piscataway: IEEE, 2015. 550−553. [doi: 10.1109/APNOMS.2015.7275389]

[88] Giotis K, Apostolaki M, Maglaris V. A reputation-based collaborative schema for the mitigation of distributed attacks in SDN domains. In: Proc. of the 2016 IEEE/IFIP Network Operations and Management Symp. Piscataway: IEEE, 2016. 495−501. [doi : 10.1109/NOMS.2016.7502849]

附中文参考文献:

[3] 张朝昆,崔勇,唐澍嗣,吴建平.软件定义网络(SDN)研究进展.软件学报,2015,26(1):62−81. http://www.jos.org.cn/1000-9825/4701.htm [doi: 10.13328/j.cnki.jos.004701]

[13] 周桐庆,蔡志平,夏竟,徐明.基于软件定义网络的流量工程.软件学报,2016,27(2):394−417. http://www.jos.org.cn/1000-9825/4935.htm [doi: 10.13328/j.cnki.jos.004935]

[22] 王蒙蒙,刘建伟,陈杰,毛剑,毛可飞.软件定义网络:安全模型,机制及研究进展.软件学报,2016,27(4):969−992. http://www.jos.org.cn/1000-9825/5020.htm [doi: 10.13328/j.cnki.jos.005020]

[42] 何亨,胡艳,郑良汉,薛正元.云环境中基于 SDN 的高效 DDoS 攻击检测与防御方案.通信学报,2018,39(4):139−151. http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2018068 [doi: 10.11959/j.issn.1000-436x.2018068]

[46] 穆祥昆,王劲松,薛羽丰,黄玮.基于活跃熵的网络异常流量检测方法.通信学报,2013,34(2):51–57. http://www.infocomm-journal.com/txxb/CN/10.3969/j.issn.1000-436x.2013.Z2.011 [doi: 10.3969/j.issn.1000-436x.2013.z2.011]

[79] 翁溪,陈鸣,张国敏,许博,邢长友.OpenFlow 网络测量分析系统的设计实现.通信学报,2015,36(3):81–88. http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2015061 [doi: 10.11959/j.issn.1000-436x.2015061]

徐玉华(1989－),女,江苏常州人,博士生,主要研究领域为软件定义网络,网络安全理论与技术,网络流量识别与控制.

孙知信(1964－),男,博士,教授,博士生导师,主要研究领域为网络通信的理论与技术,计算机网络及安全.