基于嗅探技术的字段操纵攻击研究

徐建峰[①②] 张方韬[①] 徐 震*[①] 王利明[①]

[①](中国科学院信息工程研究所 北京 100093)

[②](中国科学院大学网络空间安全学院 北京 100049)

摘 要：软件定义网络(SDN)为网络基础设施提供灵活性、可管理性以及可编程性的同时，引入了诸多新型的攻击向量。该文介绍了攻击者针对OpenFlow关键字段发起的恶意操纵攻击，并设计了3种基于数据包转发时延的嗅探技术以保证字段操纵攻击在真实SDN网络中的可实施性。实验结果表明，字段操纵攻击严重消耗了SDN网络资源，进而导致合法用户之间的通信性能明显降低。

关键词：软件定义网络；OpenFlow；字段操纵攻击；嗅探

中图分类号：TN918; TP393 **文献标识码**：A **文章编号**：1009-5896(2020)10-2342-08

DOI: 10.11999/JEIT191047

Field Manipulation Attacks Based on Sniffing Techniques

XU Jianfeng[①②] ZHANG Fangtao[①] XU Zhen[①] WANG Liming[①]

[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*)

Abstract: The flexibility, manageability, and programmability brought by Software-Defined Networking (SDN), however come at the cost of new attack vectors. Malicious manipulation attacks against the key fields in OpenFlow is proposed, and three sniffing technologies based on forwarding delay to ensure the feasibility of manipulation attacks are designed. The experimental results show that the field manipulation attacks consume SDN resources greatly, leading to a significant decrease in the communication performance between legitimate users.

Key words: Software-Defined Networking (SDN); OpenFlow; Field manipulation attack; Sniffing

1 引言

软件定义网络(Software-Defined Networking, SDN)通过将控制平面与数据平面解耦，提供了一种实现“可编程网络”的新方式。控制平面通过南向协议(如OpenFlow[1])与数据平面交互，并通过软件实现复杂的网络功能。由于全局视图和集中管控的优势，SDN自提出便获得了学术界和工业界的广泛关注，诸多应用场景也被相继提出，如数据中心[2]、云[3]、IoT[4]以及网络安全[5–7]等。

SDN为网络基础设施提供了灵活性、可编程性以及可管理性，同时也引入诸多攻击向量，如Packet-In泛洪攻击、恶意交换机以及虚假拓扑等[8–10]，这些问题导致网络性能下降，甚至破坏其可用性。本文提出了SDN中存在的一种新型攻击，即字段操纵攻击，它使攻击者能够利用OpenFlow字段的正常功能来恶意消耗网络资源。为了证明该攻击的可行性，本文实现了针对匹配、超时以及计时器字段的攻击实例。匹配操纵攻击实例中，攻击者通过伪造首部敏感字段，触发大量Packet-In消息进入控制器，急剧增加控制器计算资源的占用。超时操纵攻击实例中，攻击者根据流规则超时向网络中注入低速攻击流量，使流表长时间被无效规则占据，导致合法规则无法安装。计时器操纵攻击实例中，攻击者依据SDN应用对计时器的处理逻辑向网络注入攻击流量，频繁触发控制器发送控制消息至交换机，从而降低交换机的转发性能。

上述攻击实例中，攻击流量都是根据SDN敏感信息进行构造，因此本文设计了3种嗅探技术以获得相应的敏感信息。对于匹配操纵攻击，基于控制变量准则的嗅探技术依次改变探测包首部的匹配字段，并通过观察其传输时延来确定敏感字段。对于超时操纵攻击，基于二分法的嗅探技术以“二分时

收稿日期：2019-12-30；改回日期：2020-07-23；网络出版：2020-07-28

*通信作者： 徐震 xuzhen@iie.ac.cn

基金项目：北京市科技计划项目(Z181100002718003)

Foundation Item: Beijing Municipal Science and Technology Project (Z181100002718003)

间序列”发送探测包以快速获得流规则超时设置。对于计数器操纵攻击，基于流量模型的嗅探技术通过向网络中注入不同统计特征的流量来推测控制平面对计数器的处理逻辑。近些年，SDN嗅探技术被广泛研究。Cao等人[11]利用嗅探发现数据流量和控制流量的共享链路，进而利用低速TCP流量发起CrossPath攻击。Shin等人[12]通过观察数据包的往返时延识别SDN网络。Cao等人[13]提出了使用低速流量溢出流表的LOFT攻击，不同于超时操纵攻击中的指数嗅探方式，LOFT攻击采用了线性嗅探方式。

本文在基于Open vSwitch和Ryu控制器的实验平台上证明了嗅探技术的可行性，并评估了字段操纵攻击实例对网络的影响。实验结果表明字段操纵攻击加剧了控制器计算资源、交换机计算资源以及流表资源的消耗，明显增加了合法主机的通信时延。

2　OpenFlow流规则字段

OpenFlow已被成功应用于诸多商业部署[14]。目前OpenFlow协议已经极大扩展了流规则所包含的字段属性。如图1所示，当前流规则字段可被划分为7种类型[15]。匹配和优先级共同标识流规则，交换机按照所有匹配规则中优先级最高的规则处理数据包，处理动作由指令携带。计数器记录流量的统计信息。超时控制流规则的生存周期，其中空闲超时表示如果匹配流规则的数据包没有在小于该字段的时间内到达，则删除该规则，而强制超时意味着在流规则安装后确定的时间内被强制删除。Cookie可以实现流规则过滤功能。标志用于选择流规则的管理方式，如通过设置该字段使交换机在删除规则时自动向控制器报告。

3　字段操纵攻击

字段操纵攻击利用OpenFlow字段的原始功能实现消耗网络资源的目的。由于字段类型的多样性，该攻击存在多种实施方式。本节首先介绍攻击模型，然后针对匹配、超时和计时器字段分别描述攻击实例。

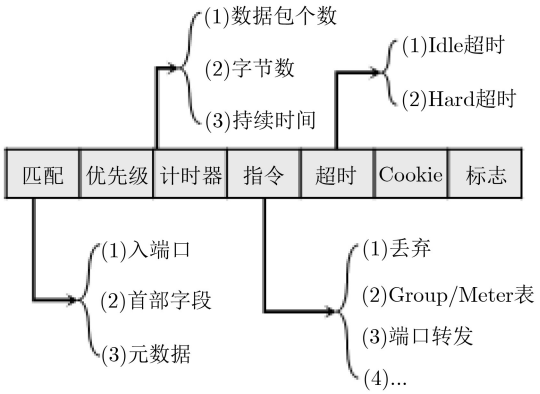


图 1 OpenFlow协议中的标准流规则

3.1　攻击模型

攻击模型将网络系统组件与破坏特定功能的攻击者能力联系起来。为了充分分析攻击模型，该小节分别阐述了系统模型、威胁模型以及攻击者能力模型。

系统模型描述了SDN网络结构，其中包括控制平面、数据平面以及两平面之间的控制连接，记作$\boldsymbol{G}=\{\boldsymbol{C},\boldsymbol{D},\boldsymbol{N}\}$。控制平面由一个或者多个控制器组成，记作$\boldsymbol{C}=\{c_1,c_2,\cdots,c_n\}$。数据平面由SDN交换机、终端主机以及设备链路组成，记作$\boldsymbol{D}=\{\boldsymbol{S},\boldsymbol{H},\boldsymbol{L}\}$，其中交换机集合记为$\boldsymbol{S}=\{s_1,s_2,\cdots,s_m\}$，主机集合记为$\boldsymbol{H}=\{h_1,h_2,\cdots,h_k\}$，$\boldsymbol{L}$标识设备间存在的通信链路，记作$\boldsymbol{L}\subseteq(\boldsymbol{S}\cup\boldsymbol{H})\times(\boldsymbol{S}\cup\boldsymbol{H})$。控制平面与数据平面之间的连接是“多对多”且被保护的(如TLS安全连接)，即1个交换机同时连接多个控制器，1个控制器与其管理域内的多个交换机通信，因此$\boldsymbol{N}\subseteq\boldsymbol{C}\times\boldsymbol{S}=\{(x,y)|x\in\boldsymbol{C},y\in\boldsymbol{S}\}$。

威胁模型描述攻击所利用的脆弱性信息，而字段操纵攻击所利用的脆弱性在于SDN组件资源的有限性。具体地，1个控制器实例负责管理多个交换机，操纵攻击通过耗尽控制器计算资源来造成其管理域内的网络瘫痪；SDN交换机的计算与流表资源十分匮乏，攻击通过消耗交换机资源影响合法流量的转发。

攻击者能力模型描述了攻击者发起攻击应具备的基本能力，该模型描述的能力强弱决定了攻击实施的难易。在字段操纵攻击中，攻击者无需入侵交换机或控制器，无需篡改控制消息，更无需具备操纵网络的特权，只要能够控制终端主机发送数据包即可。由此可见，攻击者只需获得终端主机的控制权便能够实现字段操纵攻击。下面，本文将通过3种攻击实例来阐述字段操纵攻击的原理与流程。

3.2　匹配操纵攻击实例

匹配操纵攻击是一种更加复杂的Packet-In泛洪攻击[8]。该攻击通过伪造首部敏感字段来构造攻击流量，而不是普通Packet-In泛洪攻击中的伪造任意字段，从而以更加高效的方式消耗网络资源。其中，敏感字段意味着该字段的变化会触发控制平面产生新流规则。

3.2.1　基于控制变量准则的嗅探技术

本小节提出一种基于控制变量准则的嗅探技术来帮助攻击者确定哪些首部字段会触发控制器下发流规则。对于多因素的问题，可以通过每一次只改变其中的某一个因素同时控制其余因素不变来研究被改变的因素对事物的影响。在嗅探技术的控制准

则中，自变量为所有的首部字段类型(如MAC地址字段与IP地址字段等)，因变量为控制器是否下发流规则。因此，当检测某一首部字段是否为敏感字段时，只需要控制探测包中的其它首部字段不变，仅改变待研究字段。假设修改后的字段会导致流规则下发，则说明该字段为敏感字段，反之为非敏感字段。攻击者可以根据探测包的往返时延(Round-Trip Time, RTT)来判断控制器是否下发了流规则。具体地，当探测包无法与流规则匹配时，交换机会通过Packet-In事件请求控制器下发流规则以指导交换机转发报文，该方式导致转发时延远高于交换机直接转发报文的时延。

按照上述准则，嗅探过程如下：攻击者每一次仅修改原始报文中1个首部字段来构造探测报文，假如探测报文会触发Packet-In事件，说明该报文不能与原始报文产生的流表项匹配，进而证明所修改字段为敏感字段，反之判定不是敏感字段。最新OpenFlow规范[15]仅支持42个首部字段，因此攻击者通过不超过42次试验便能确定敏感字段。图2展示了窃取2层转发应用敏感字段的结果。当保持其它首部字段不变仅修改IP地址时，探测报文RTT没有发生明显变化，说明报文与之前产生的流表项匹配，因此断定IP字段不是敏感字段，而仅修改MAC地址时，探测报文RTT会发生明显变化，这说明MAC字段是敏感字段。

3.2.2 攻击过程

如图3所示，在嗅探阶段，攻击者利用基于控制变量准则的嗅探技术窃取到控制平面的敏感字段。在触发阶段，攻击者通过随机伪造敏感字段来构造攻击流量，并将其通过傀儡主机注入到SDN中，进而触发大量的Packet-In事件精准地消耗控制器计算资源。

3.3 超时操纵攻击实例

SDN交换机通常使用三态内容寻址存储器(Ternary Content Addressable Memory, TCAM)保存流规则，以实现数据包的线速处理，但是由于高成本和高功耗，TCAM只允许保存数千条流规则[16]。因此OpenFlow引入了超时字段来控制流规则的生存周期。超时操纵攻击是指攻击者根据空闲超时配置周期性地注入低速攻击流，使得恶意规则长期占据交换机流表资源，进而导致合法规则无法安装。

3.3.1 二分法嗅探技术

基于二分法的嗅探技术帮助攻击者推测出流规则的超时设置，其探测过程如表1所示。$\{p_1, p_2, \cdots, p_n\}$表示属于同一数据流的报文，其中$p_i$表示第$i$次探测所发送的数据包。当攻击者向网络中注入$p_1$时，由于当前交换机中没有对应的转发规则，$p_1$会触发流规则安装过程，因此$p_1$会获得较高的转发时延。等待一个足够长的时间$t$后(保证$p_1$所安装的流规则由于超时被剔除)，攻击者再次向网络中注入p_2，结果又得到了较高的转发时延，由此可以得到流规则的超时位于区间$[0, t]$之内。接着等待$t/2$时间后，攻击者向网络中注入p_3，如果p_3仍然得到了较高的转发时延，可以推断流规则超时位于区间$[0, t/2]$之内；如果p_3引入了较低的转发时延，则说明在p_3转发时，交换机仍保存转发该数据包所需的规则，故推断出流规则超时位于区间$[t/2, t]$之内。然后攻击者等待$t/4$或者$3t/4$后向网络中注入p_4，并继续上述比较，将超时范围进一步缩小。若干轮探测后，攻击者便可以获得足够精确的超时区间。

3.3.2 攻击过程

攻击者首先通过二分法嗅探技术获得流规则超时区间，然后以区间中的最小值为发送间隔，周期性地向网络中注入低速攻击报文。这些报文使得流规则在即将到期前重置空闲超时，从而延长了规则生存时间。当流表完全被无效规则占据时，合法规则便无法正常安装。

3.4 计时器操纵攻击实例

SDN应用利用Statistics Query/Reply控制消

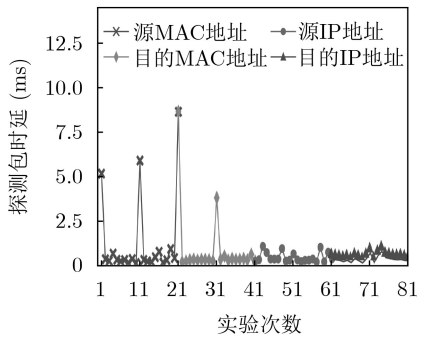


图 2 Ryu控制器2层转发应用的嗅探结果

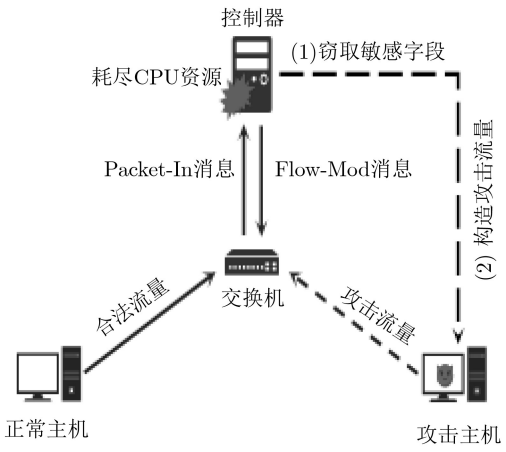


图 3 匹配操纵攻击流程图

表 1　基于二分法的嗅探技术

初始化：探测包序列$\{p_1, p_2, \cdots, p_n\}$；最小超时初始设置为0；最大超时初始设置为$t$(保证$t$时间后规则被剔除)；
(1) 注入p_1数据包；
(2) 循环，对于探测包序列$\{p_1, p_2, \cdots, p_n\}$中的每一个数据包$p_i$:
(3)　设置等待时延为(最小超时+最大超时)/2；
(4)　等待时延过后，注入p_i数据包，并获得p_i数据包的往返时延；
(5)　如果往返时延较大，说明p_i数据包再次触发了流规则安装过程，则：
(6)　更新最大超时为(最小超时+最大超时)/2；
(7)　否则，说明p_i数据包没有触发了流规则安装过程，然后：
(8)　更新最小超时为(最小超时+最大超时)/2；
(9) 当全部探测包发送完毕，返回得到的最小超时和最大超时；

息来实现优化、监控以及保护网络等目标[17]，其工作流程一般为：(1) 获得交换机中的计数器字段；(2) 根据计数器信息下发新的网络策略。在计时器操纵攻击中，攻击者向网络中注入特定统计特征的攻击流量，故意触发计时器处理逻辑，从而导致控制器频繁下发流规则，消耗交换机计算资源。

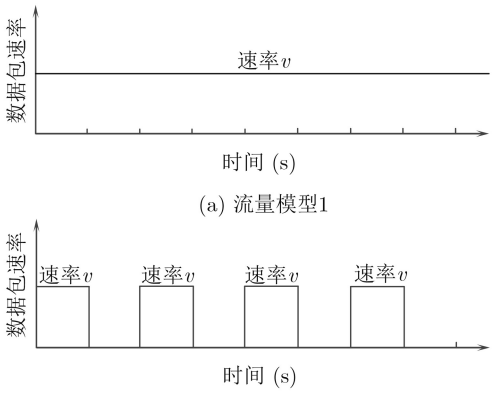
3.4.1　基于流量模型的嗅探技术

基于流量模型的嗅探技术帮助攻击者获得SDN应用对计时器字段的处理逻辑，该技术依赖于不同控制消息对交换机计算资源的消耗不同。目前OpenFlow主要包含3种下行控制消息：Flow-Mod消息，Statistics Query消息以及Packet-Out消息，其中Flow-Mod的计算开销最大，Statistics Query次之，Packet-Out最小[16]，因此交换机在处理这些控制消息时转发数据包的性能不同。

SDN应用定期使用Statistics Query/Reply消息来获得计时器信息，每一个查询消息都会导致探测包转发时延增加，据此攻击者可以获得查询统计信息的周期。此外在获得统计信息后，假如计时器满足了流规则的产生条件，控制器会继续向交换机发送Flow-Mod消息，此时会再一次造成探测包转发时延的升高。攻击者通过向网络中注入不同统计特征的背景流量(交换机中已存在对应的转发规则)来推测SDN应用对计时器字段的处理逻辑。图4展示了两种不同模型的流量模板，第1种流量具有稳定的数据包速率v，数据包大小p，第2种流量中数据包速率分布类似于一个跳跃函数。假如攻击者通过向网络中注入这两种类型的流量测量到的探测数据包时延如图5所示，则攻击者可以推测该应用是根据数据包个数来下发流规则。通过进一步改变探测包的发送速率，攻击者可以得到应用对计时器字段的判断阈值。

3.4.2　攻击过程

攻击者利用基于流量模型的嗅探技术获得计时器的采集周期以及处理逻辑，然后在每一个周期内向交换机发送精心设计的攻击流量，使流规则中的计时器字段能够触发新规则的产生，从而导致每一个采集周期内都有大量的流规则下发至交换机。



(a) 流量模型1

(b) 流量模型2

图 4 两种类型的流量特征

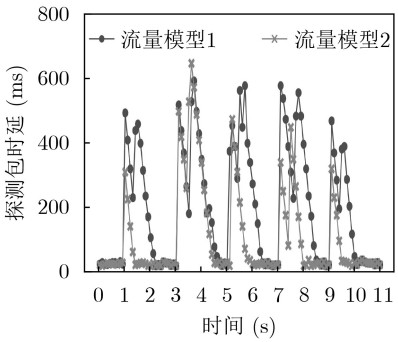


图 5 基于流量模型的嗅探技术

4　实验评估

4.1　实验环境

实验平台由2台Ubuntu 16.04 LTS主机(Intel Xeon E5-1603 2.8 GHz CPU, 8 GB RAM)构成，其中一台主机作为控制平面安装了Ryu控制器，另一台主机使用Mininet搭建底层网络。实验拓扑如图6所示，1个恶意主机和4个合法主机共同连接到

Open vSwitch上，同时交换机与控制器建立OpenFlow安全连接。

4.2 匹配操纵攻击实例

本实验评估了匹配操纵攻击对控制器计算资源的影响，实验环境如下：控制器部署2层转发应用，流规则超时为3 s，正常主机1以3000 pps的速率向网络中注入背景流量。攻击者使用嗅探技术得到了控制平面的敏感字段是MAC地址(嗅探结果以及描述见图2)，因此通过伪造该字段产生攻击流量。

(1) 对控制器CPU资源的影响：图7所示，当网络中仅存在背景流量时，控制器CPU占用率大约为15%(场景1)。在此基础上，分别注入相同速率的合法流量和攻击流量来测量正常场景和攻击场景下CPU消耗情况。具体地，10 s时主机2以1000 pps的速率注入合法流量，CPU占用率增加至25%(场景2)；当攻击主机以1000 pps的速率注入攻击流量时，CPU占用率增加至85%(场景3)。

(2) 对转发时延的影响：为了检测攻击对用户转发性能的影响，测量了上述场景下主机3和4之间的RTT变化。图8展示了当网络中仅存在背景流量时，主机之间的平均RTT为6.77 ms，而在场景2和场景3中，该时延分别上升至10.32 ms和142.50 ms，并且在攻击发生时，时延抖动明显增加。

(3) 结果分析：随着注入流量的增多，交换机发送更多的Packet-In消息至控制器，因此控制器计算资源的消耗会随着网络流量的增多而升高。特别地，在匹配操作攻击中，几乎每一个攻击报文产生一个Packet-In消息，因此该攻击最大限度地消耗控制器计算资源，进而影响着合法用户的通信时延。

4.3 超时操纵攻击实例

本实验评估了超时操纵攻击对交换机存储资源的影响，实验环境如下：控制器部署2层转发应用，流规则空闲超时设置为3 s，强制超时为0，背景流量以3000 pps的速率注入网络中，交换机最多可安装3000条流规则。攻击者利用二分法嗅探技术在对数级时间复杂度内得到超时设置，然后每秒向网络中注入300条新流，并对每一数据流每隔2.5 s发送1个数据包以刷新规则超时。

(1) 对交换机存储资源的影响：实验测量了在有无攻击发生时交换机中流表项的数量。图9展示了正常网络状况下，流表中大约存在250条流规则。第5 s攻击实施后，流规则数量以大约300条/s的速率上升，并在15 s时将流表完全填充。由于攻击者每秒向网络注入300条新流，并且恶意规则会长期保持在流表中，因此流规则数量的上升速率大约为300条/s。

(2) 对转发时延的影响：超时操纵攻击会使交换机流表发生溢出，进而会对交换机连接的主机造成通信影响。如图10所示，在正常的网络环境中，主机3和4之间的平均往返时延为6.59 ms，而当攻

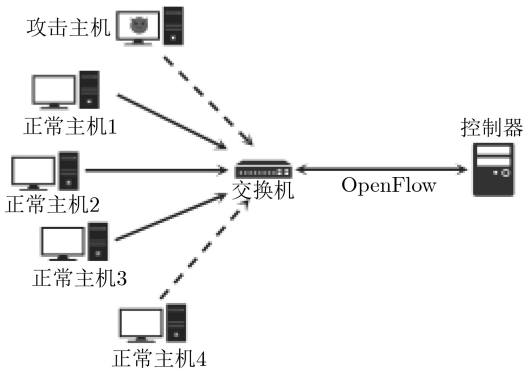


图 6 实验拓扑

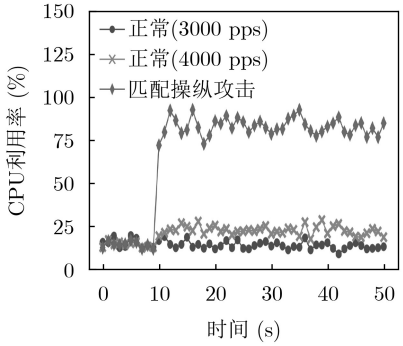


图 7 匹配操纵攻击对控制器CPU的影响

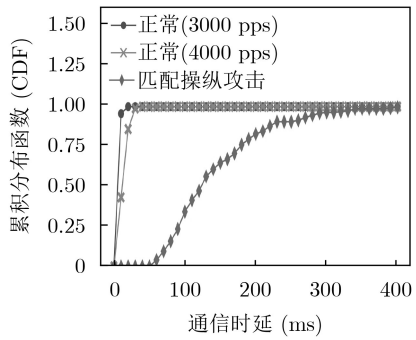


图 8 匹配操纵攻击对合法用户时延的影响

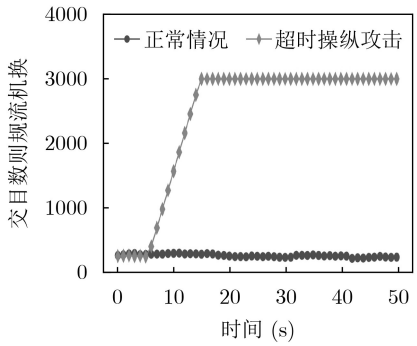


图 9 超时操纵攻击对交换机流表资源的影响

击发生后，由于交换机已无法正常安装流表项，导致主机通信时延急剧上升至约215.51 ms。

(3) 结果分析：空闲超时的特征为，在其失效之前，每一个匹配的数据包都会重新刷新流规则的生存时间，因此超时字段操纵攻击能够使无效流规则长期占据流表资源，迫使交换机无法安装正常流规则，进而影响合法主机的通信(控制器仅可以通过Packet-out消息指导数据包转发)。

4.4 计时器操纵攻击实例

本实验评估了计时器操纵攻击对交换机计算资源的影响，实验环境如下：控制器部署流量监控应用，信息采集周期设置为2 s，包个数计数器大于20时重新计算流规则，背景流量以1000 pps的速率注入网络。攻击者利用基于流量模型的嗅探技术获得计时器采集周期和处理逻辑，然后据此信息，以3000 pps的速率向网络中注入攻击流量(共300条数据流，对每条流以10 pps的速率发送数据包)。

(1) 对交换机计算资源的影响：图11描述了当网络中仅存在背景流量时，交换机CPU平均利用率大约是25%。在10 s时发起计时器操纵攻击后，由于交换机中记录的流量统计信息满足监控应用产生流规则的条件，控制器会频繁向交换机下发流规则，进而交换机CPU利用率急速增加至大约110%。

(2) 对转发时延的影响：通过测量正常主机3与4之间的通信往返时延变化来展示计时器操纵攻击对用户转发行为的影响。图12展示了当网络中仅存在背景流量时，主机之间的平均往返时延为6.83 ms，而在攻击发生后，该平均时延上升至65.47 ms，并且抖动明显增加。

(3) 结果分析：根据攻击流量模式，攻击者向网络注入的300条流量会在监控应用信息采集时触发新规则的产生条件，因此控制平面会以大约300条/s的速率将恶意规则下发给交换机，造成交换机计算资源的严重消耗，进而降低了转发合法报文的效率。

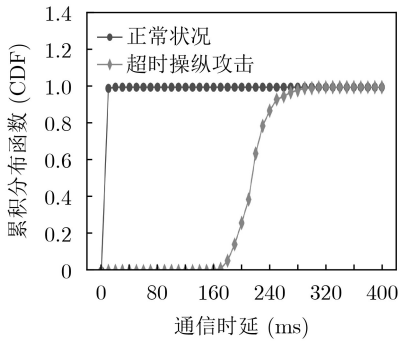


图 10 超时操纵攻击对合法用户通信时延的影响

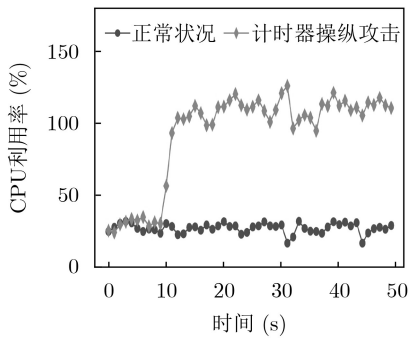


图 11 计时器操纵攻击对计时器操纵攻击的影响

5 可能的缓解机制

(1) 抑制攻击消息：匹配或计时器操纵攻击频繁地触发某些SDN应用正常逻辑的执行，因此短时间内会有大量的恶意控制消息分发至这些应用。控制器可以根据控制信息分发速率或流规则生成速率来判断攻击的目标应用，并为进入到该应用的控制消息设置低优先级。相反地，对于进入到其它应用的控制消息，设置为高优先级。不同优先级的控制消息放置到不同发送速率的缓存队列以降低处理恶意控制消息的吞吐量，并保证合法控制消息的处理时延，因此缓解网络资源消耗，类似防御思想被广泛应用在SDN中[11,16]。对于超时操纵攻击，通过观察图9发现，在流表溢出之前流规则的数量持续增加，并且增加速度可能很慢，这些特征可以帮助防御者捕获攻击。一旦检测到攻击，控制器便可以剔除可疑规则(即始终在流表中但每秒转发很少数量数据包的规则)。此外，由于攻击者定期生成数据包以刷新规则超时，可疑规则的数据包转发速率可能会显示出周期性模式，这一特征可以帮助防御者进一步定位恶意规则。同时，防御者可以利用Meter表[15]来限制具有攻击特征的数据流进入网络，从而缓解超时操纵攻击。

(2) 动态配置规则超时：根据调查发现[13]，同一控制器上的大部分应用将流规则超时设置为固定值。我们建议在安装新规则或重新安装规则时，应

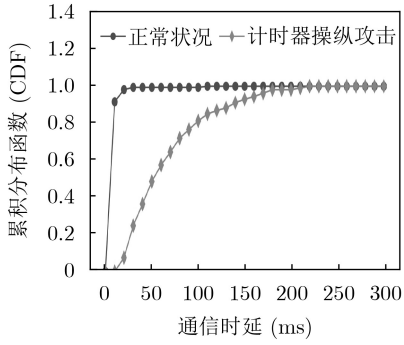


图 12 计时器操纵攻击对合法用户时延的影响

用为规则设置不同的超时以干扰超时信息嗅探过程，如Zhu等人[18]提出了一种根据流量统计特性为规则设置超时值的智能超时控制机制。因此，攻击者无法轻易推断出控制器设置的超时。在这种情况下，由于缺少准确的超时信息，超时操纵攻击难以实施。此外，控制器可以灵活地使用强制超时，禁止规则长期驻留在流表中，从而缓解超时操纵攻击。

(3) 动态调整RTT：防御者可以通过干扰RTT测量来阻止嗅探过程[13,16]，进而防止操纵攻击的发生。具体地，控制器在传输新流的前几个数据包时加入人为的随机抖动，如控制器接收到Packet-In消息后不会立即为新流产生规则(而是连续收到同一个流触发的多个Packet-In消息后才产生规则)，同时故意等待随机时延后才将数据包返回给交换机。因此，攻击者无法准确推断出控制器是否为探测包安装新规则。该机制潜在的缺点是，对良性流量会产生额外的转发延迟，并要求控制器处理更多的Packet-In消息。

6 结束语

本文提出了一种针对OpenFlow网络的新型攻击，即字段操纵攻击，它利用特定形式的攻击流量有效地消耗网络资源。该文实现了3种字段操纵攻击实例，并提出了3种嗅探技术以获得攻击所必需的敏感信息。实验结果证明了字段操纵攻击严重消耗了控制器计算资源、交换机计算资源以及流表资源，进而明显降低了合法用户的通信效率。我们希望这项工作能够引起研究人员对SDN安全更多的关注，尤其是在创新网络应用时SDN架构自身存在的安全威胁。

参 考 文 献

[1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, *et al*. OpenFlow: Enabling innovation in campus networks[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69–74. doi: 10.1145/1355734.1355746.

[2] ZENG Yue, GUO Songtao, and LIU Guiyan. Comprehensive link sharing avoidance and switch aggregation for software-defined data center networks[J]. *Future Generation Computer Systems*, 2019, 91: 25–36. doi: 10.1016/j.future.2018.08.034.

[3] WANG Haopei, SRIVASTAVA A, XU Lei, *et al*. Bring your own controller: Enabling tenant-defined SDN apps in IaaS clouds[C]. IEEE Conference on Computer Communications, Atlanta, USA, 2017: 1–9. doi: 10.1109/INFOCOM.2017.8057137.

[4] SAHAY R, MENG Weizhi, ESTAY D A S, *et al*. CyberShip-IoT: A dynamic and adaptive SDN-based security policy enforcement framework for ships[J]. *Future Generation Computer Systems*, 2019, 100: 736–750. doi: 10.1016/j.future.2019.05.049.

[5] ZHENG Jing, LI Qi, GU Guofei, *et al*. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(7): 1838–1853. doi: 10.1109/TIFS.2018.2805600.

[6] 姚琳元, 董平, 张宏科. 基于对象特征的软件定义网络分布式拒绝服务攻击检测方法[J]. 电子与信息学报, 2017, 39(2): 381–388. doi: 10.11999/JEIT160370.
YAO Linyuan, DONG Ping, and ZHANG Hongke. Distributed denial of service attack detection based on object character in software defined network[J]. *Journal of Electronics & Information Technology*, 2017, 39(2): 381–388. doi: 10.11999/JEIT160370.

[7] 武泽慧, 魏强, 任开磊, 等. 基于OpenFlow交换机洗牌的DDoS攻击动态防御方法[J]. 电子与信息学报, 2017, 39(2): 397–404. doi: 10.11999/JEIT160449.
WU Zehui, WEI Qiang, REN Kailei, *et al*. Dynamic defense for DDoS attack using OpenFlow-based switch shuffling approach[J]. *Journal of Electronics & Information Technology*, 2017, 39(2): 397–404. doi: 10.11999/JEIT160449.

[8] DENG Shuhua, GAO Xing, LU Zebin, *et al*. DoS vulnerabilities and mitigation strategies in software-defined networks[J]. *Journal of Network and Computer Applications*, 2019, 125: 209–219. doi: 10.1016/j.jnca.2018.10.011.

[9] SKOWYRA R, XU Lei, GU Guofei, *et al*. Effective topology tampering attacks and defenses in software-defined networks[C]. The 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Luxembourg City, 2018: 374–385. doi: 10.1109/dsn.2018.00047.

[10] LI Qi, ZOU Xiaoyue, HUANG Qun, *et al*. Dynamic packet forwarding verification in SDN[J]. *IEEE Transactions on Dependable and Secure Computing*, 2019, 16(6): 915–929. doi: 10.1109/TDSC.2018.2810880.

[11] CAO Jiahao, LI Qi, XIE Renjie, *et al*. The crosspath attack: Disrupting the SDN control channel via shared links[C]. The 28th USENIX Conference on Security Symposium, Berkeley, USA, 2019: 19–36.

[12] SHIN S and GU Guofei. Attacking software-defined networks: A first feasibility study[C]. The 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, China, 2013: 165–166. doi: 10.1145/2491185.2491220.

[13] CAO Jiahao, XU Mingwei, LI Qi, *et al*. Disrupting sdn via the data plane: A low-rate flow table overflow attack[C].

The 13th International Conference on Security and Privacy in Communication Networks, Niagara Falls, Canada, 2017: 356–376. doi: 10.1007/978-3-319-78813-5_18.

[14] JAIN S, KUMAR A, MANDAL S, *et al*. B4: Experience with a globally-deployed software defined wan[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(4): 3–14. doi: 10.1145/2486001.2486019.

[15] Open Networking Foundation. OpenFlow switch specification 1.5. 1[EB/OL]. https://www.opennetworking.org/software-defined-standards/specifications/, 2019.

[16] ZHANG Mengtao, LI Guanyu, XU Lei, *et al*. Control plane reflection attacks in SDNs: New attacks and countermeasures[C]. The 21st International Symposium on Research in Attacks, Intrusions, and Defenses, Heraklion, Greece, 2018: 161–183. doi: 10.1007/978-3-030-00470-5_8.

[17] XU Hongli, YU Zhuolong, QIAN Chen, *et al*. Minimizing flow statistics collection cost of SDN using wildcard requests[C]. IEEE Conference on Computer Communications, Atlanta, USA, 2017: 1–9. doi: 10.1109/INFOCOM.2017.8056992.

[18] ZHU Huikang, FAN Hongbo, LUO Xuan, *et al*. Intelligent timeout master: Dynamic timeout for SDN-based data centers[C]. The 13th International Symposium on Integrated Network Management, Ottawa, Canada, 2015: 734–737. doi: 10.1109/INM.2015.7140363.

徐建峰：男，1995年生，博士生，研究方向为软件定义网络与网络系统安全.

张方韬：男，1982年生，博士生，研究方向为软件定义网络与网络系统安全.

徐　震：男，1976年生，正高级工程师，研究方向为网络系统安全与边缘计算.

王利明：男，1978年生，正高级工程师，研究方向为网络系统安全与大数据安全分析.

责任编辑：余　蓉