·安全技术· 文章编号：1000-3428(2019)10-0176-07 文献标志码：A 中图分类号：TP391

SDN 中交换机控制代理拒绝服务攻击研究

邓书华[a]，卢泽斌[a]，李正发[a]，高协平[a,b]

（湘潭大学 a. 智能计算与信息处理教育部重点实验室；b. 信息工程学院，湖南 湘潭 411105）

摘 要： 针对软件定义网络（SDN）交换机控制代理吞吐量有限的不足，提出一种交换机控制代理拒绝服务攻击方法。为保护 SDN 交换机控制代理资源，设计层次化多阈值的攻击检测方案，通过计算 SDN 交换机及交换机各端口所关联的 Packet-In 消息的速率，并将其与给定的阈值进行比较来检测攻击。实验结果表明，该方法可实时检测交换机控制代理拒绝服务攻击，与同类攻击检测方法相比，不仅能够识别攻击流量，定位攻击发生的位置，而且不需要改变网络架构。

关键词： 软件定义网络；OpenFlow 交换机；拒绝服务攻击；控制代理；网络架构

开放科学（资源服务）标志码（OSID）：

中文引用格式： 邓书华，卢泽斌，李正发，等. SDN 中交换机控制代理拒绝服务攻击研究[J]. 计算机工程，2019，45(10)：176-182.

英文引用格式： DENG Shuhua，LU Zebin，LI Zhengfa，et al. Research on switch control agent denial-of-service attack in SDN[J]. Computer Engineering，2019，45(10)：176-182.

Research on Switch Control Agent Denial-of-Service Attack in SDN

DENG Shuhua[a]，LU Zebin[a]，LI Zhengfa[a]，GAO Xieping[a,b]

（a. The MOE Key Laboratory of Intelligent Computing and Information Processing；
b. College of Information Engineering，Xiangtan University，Xiangtan，Hunan 411105，China）

【Abstract】 Aiming at the shortcomings of limited throughput of Software Defined Networking（SDN）switch control agent，a switch control agent denial-of-service attack method is proposed. To protect the SDN switch control agent resources，a hierarchical multi-threshold attack detection scheme is designed to detect attacks by calculating the rate of Packet-In messages associated with each port of the SDN switch and comparing it with a given threshold. Experimental results show that the algorithm can detect the switch control agent denial-of-service attacks in real time. Compared with the similar attack detection method，the proposed method can not only identify the attack traffic，but also locate the location of the attacks without changing the network architecture.

【Key words】 Software-Defined Networking（SDN）；OpenFlow switch；denial-of-service attack；control agent；network architecture

DOI： 10.19678/j.issn.1000-3428.0052435

0 概述

软件定义网络（Software-Defined Networking，SDN）是一种新兴的网络体系结构，它能够实现控制逻辑与转发功能的分离，并采用集中式的控制方式，向上层应用提供开放的可编程接口。这种设计模式简化了交换机的设计、网络协议的部署和网络策略的更新，因此，得到了广泛的应用[1-3]。谷歌通过在它的数据中心网络中部署 SDN 技术，将链路带宽利用率提高到了近 100%[4]。

尽管 SDN 取得了较大的成功，但是这种体系结构也带来了新的安全威胁，比如 SDN 中的 DoS 攻击[6]、网络拓扑投毒攻击[7]和标识符绑定攻击[8]等。SDN 将网络设备中的控制逻辑分离之后，采用集中式的方式控制整个网络。因此，控制器成为整个网络的大脑，也是最有价值的攻击目标。目前大多数

基金项目： 国家自然科学基金（61771415）；赛尔网络下一代互联网技术创新项目（NGII20160408）。

作者简介： 邓书华（1991—），男，博士研究生，主研方向为软件定义网络安全；卢泽斌，博士研究生；李正发，硕士研究生；高协平，教授、博士生导师。

收稿日期： 2018-08-17 **修回日期：** 2018-10-25 **E-mail：** shuhuadeng@163.com

SDN 安全问题的研究都集中在控制器方面。比如 FloodGuard[6]、AVANT-GUARD[9] 和 FloodDefender[10] 等都是用来防御控制层的 DoS 攻击。但与此同时,SDN 数据平面的安全问题并没有得到广泛关注。

在 SDN 网络中,控制功能从转发设备中分离,交换机只需负责完成数据包的转发以及与控制平面的通信。交换机中的所有控制逻辑都集中在控制代理组件中,控制代理则运行在交换机 CPU 硬件上。此外,原来需要许多硬件支持的机制,如交换机端口安全、路由器中的路由协议等都在控制器中实现。这种方式极大地简化了 SDN 交换机的设计,并降低了对交换机硬件配置的要求。因此,多数交换机都使用了性能一般的 ASIC 芯片来实现数据包的转发。此外,SDN 交换机还使用了低端的 CPU 来完成与控制平面的通信。比如 Pica8 的 P-3297 交换机使用的 ASIC 芯片是 Triumph2,CPU 是 P2020。

SDN 交换机的这种设计方式在数据平面和转发平面分离之后是合理的,不仅节约了成本也能够满足网络和用户的需求,但同时也带来了潜在的安全威胁。当交换机中没有对应的流表项匹配网络中的数据包时,由交换机控制代理将这些不匹配的数据包封装成 Packet-In 消息发送到控制器请求处理。同时,控制代理通过解析控制器下发的 Flow-Mod 和 Packet-Out 消息来完成流表的安装和数据包的转发。在这种情况下,攻击者可以通过构造大量的不匹配数据包消耗交换机控制代理资源。一旦交换机控制代理过载,将显著降低交换机的转发性能,影响网络的正常运行。此外,由于控制代理运行在交换机 CPU 之上,而目前的硬件交换机普遍使用了低端的 CPU,降低了对交换机控制代理发起攻击的门槛。

针对以上问题,本文设计一种交换机控制代理拒绝服务攻击,并提出一种攻击检测算法。通过硬件实验环境,分析攻击方法的可行性,并在 Floodlight 控制器中对算法的有效性进行了验证。

1　相关工作

目前,国内外学者对 SDN 体系结构的安全问题研究已经取得了一些成果。其中,数据平面的安全研究主要集中在主机、交换机和链路这些组件上。文献[7]提出数据平面的主机位置劫持攻击和链路虚构攻击,并在此基础上实现了流量窃听、拒绝服务攻击和中间人攻击,为了检测这些攻击并保护数据平面资源,设计并实现了 TopoGuard 系统。文献[12]提出一种交换机流表容量和使用率推理攻击模型,通过推理结果发起流表溢出攻击。文献[13]提出一种通过控制器中可疑应用程序和非法数据包发起流表溢出攻击的方法,并设计一种动态的基于时间驱动的流表溢出攻击缓解策略。文献[14-15]研究 SDN 数据平面侧信道攻击,并提出一些认证和授权策略来避免侧信道攻击。

尽管这些解决方案都能在一定程度上提高 SDN 数据平面的安全性,但是交换机控制代理这一关键元素的安全性研究并没有得到关注。在现有的研究中,文献[11]提出控制代理的性能瓶颈问题。通过研究发现,交换机控制代理的吞吐量非常有限,当交换机在处理高负载流量、突发流量或者 DoS 攻击流量时,将会导致交换机处理能力显著下降,为此,提出 Scotch 架构,即利用软件交换机控制代理吞吐量高的特点,使用叠加(Overlay)网络的方法,将硬件交换机的负载转移到软件交换机上以增强交换机控制代理的处理能力。Scotch 虽然能够在一定程度上扩展交换机控制代理的处理能力,但是需要额外技术手段(如在交换机之间建立隧道)的支持。此外,Scotch 没有考虑交换机控制代理的安全性,即这种方案无法检测针对交换机控制代理的拒绝服务攻击,也无力应对交换机控制代理 DoS 攻击。

本文的主要贡献如下:

1)提出针对交换机控制代理的 DoS 攻击,并在物理实验环境下证明了攻击的可行性,分析了攻击所产生的影响。

2)给出一种层次化多阈值的交换机控制代理 DoS 攻击检测算法,通过实验证了该算法的有效性。

3)通过 2 种缓解交换机控制代理 DoS 攻击的防御策略,以应对不同程度的控制代理 DoS 攻击。

2　交换机控制代理拒绝服务攻击

2.1　SDN 交换机原理

SDN 中有 2 种类型的 OpenFlow 交换机,即硬件交换机和软件交换机,这 2 种交换机的设计原理遵循 OpenFlow 协议。

硬件 OpenFlow 交换机的主要应用场景是数据中心网络、骨干网和校园网,是构建 SDN 物理网络的核心组件。硬件 OpenFlow 交换机的架构如图 1 所示,主要组件包括流表和 OpenFlow 控制代理。OpenFlow 交换机在处理数据包时,首先将数据包的指定字段与流表中的表项进行匹配。一旦匹配成功,数据包按照流表项中的相关指令进行转发,这个过程需要交换机中的 ASIC 芯片支持。当交换机中不存在与数据包匹配的表项时,数据包的转发需要请求控制器进行处理。此时,OpenFlow 控制代理将不匹配的数据包封装成 Packet-In 消息并转发到控制器。同时,控制代理通过解析控制器下发的消息,实现对交换机的控制。OpenFlow 控制代理的主要组件是 ovs-vswitchd 进程,运行在交换机 CPU 上,所有不匹配数据包的处理过程都通过该进程进行处理。因此,OpenFlow 交换机对不匹配数据包的处理能力主要取决于控制代理的吞吐量。此外,由于控制代理运行在交换机 CPU 上,因此交换机 CPU 的处理能力很大程度上决定了控制代理的吞吐量。

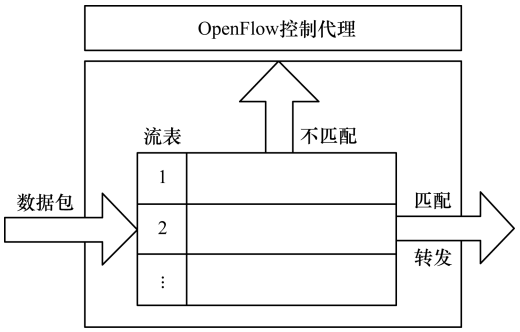

图 1 OpenFlow 交换机架构

软件 OpenFlow 交换机主要部署在云计算和 hypervisor 平台中,是构建 SDN 虚拟网络的核心组件。在实际应用场景中,使用最广泛的软件交换机是 Open vSwitch,Open vSwitch 架构如图 2 所示[4]。Open vSwitch 的主要组件包括运行在内核中的数据通道和流表以及运行在用户空间的 ovs-vswitchd 进程和 ovs-db 数据库、ovs-ofctl 等管理工具。数据通道从物理网卡或者虚拟网卡处获取网络中的数据包,并根据流表信息实现对数据包的转发。ovs-vswitchd 进程根据 ovs-db 中的信息确定数据包转发路径,实现对不匹配数据包的转发。因此,软件交换机对不匹配数据包的处理能力主要取决于运行 Open vSwitch 交换机的虚拟机或主机的配置。在通常情况下,Open vSwitch 交换机控制代理的吞吐量高于硬件交换机控制代理的吞吐量。但是,Open vSwitch 交换机控制代理的吞吐量是有限的,在交换机控制代理 DoS 攻击下,也会成为网络的性能瓶颈。

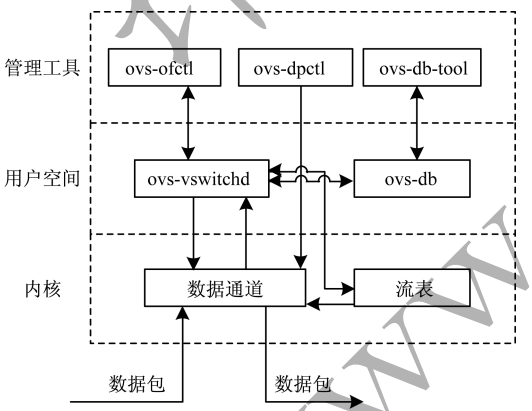

图 2 Open vSwitch 架构

2.2 攻击模型

针对交换机控制代理吞吐量有限的不足,本文提出一种控制代理拒绝服务攻击模型。如图 3 所示,主机 A、B 和攻击者控制的主机连接在 SDN 硬件交换机上,SDN 控制器运行在服务器上并使用 OpenFlow 协议与交换机进行通信。假设攻击者可以通过恶意软件控制 SDN 网络中的一个或者多个主机,并且拥有该主机对数据包的操作系统级读写权限。这种假设与文献[6-7]一致,也是合理的。

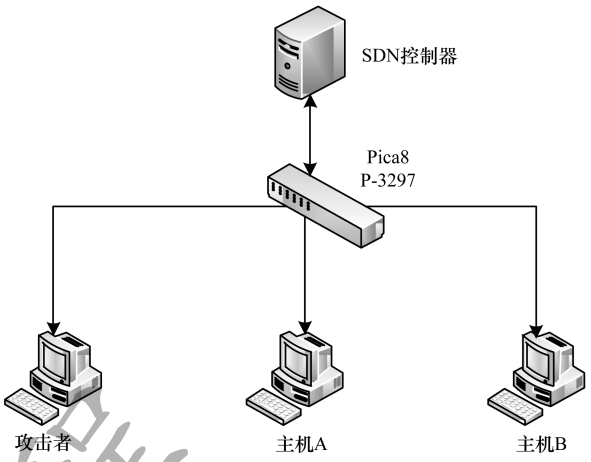

图 3 交换机 DoS 攻击模型

攻击者通过控制网络中的主机,在短时间内构造大量交换机无法匹配的数据包发送到交换机中。交换机在处理这些数据包时,首先通过控制代理将这些数据包封装起来,以 Packet-In 消息的形式发送到 SDN 控制器请求处理。控制器通过解析 Packet-In 消息,并根据自身运行的网络策略和路由算法,确定数据包的转发路径。通常,在控制器中没有攻击者所构造的这些数据包的目的地址信息。因此,控制器将向交换机控制代理发送一个 Packet-Out 消息,显示的命令交换机将该数据包从除入口以外其他所有端口洪泛。同时,控制器还会下发一个 Flow-Mod 消息到控制代理中,将与该数据包相关的流表项安装到交换机中。因此,攻击者在短时间内注入的这些数据包会产生大量的 Packet-In、Packet-Out 和 Flow-Mod 消息,这些消息的处理会给交换机控制代理带来极大的负载开销。由于终端主机的 CPU 和运行控制器的服务器 CPU 处理能力都远远超过交换机 CPU 的处理能力,因此这种硬件资源的不对称性使得运行在交换机 CPU 上的控制代理很快满负载运行。一旦交换机控制代理过载,当主机 A 中的合法流量需要通过交换机请求控制器进行处理时,交换机控制代理无法将该请求送到控制器,从而导致拒绝服务攻击。

在上述攻击模型中,实现交换机控制代理 DoS 攻击的关键在于,将构造交换机无法匹配的数据包与正常的网络流量一起发送到交换机中。本文通过给数据包的目的 IP 地址和端口随机赋值,将这些数据包与正常流量混合之后发送到交换机中来发起攻击。此外,本文通过控制同一交换机下的多个主机对边缘交换机发起攻击,也可以控制多个交换机下的主机对核心层的交换机发起攻击。由于 SDN 数据平面与控制平面通信的主要瓶颈在于交换机 CPU,且攻击流量能够成倍地增加控制代理的负载,因此这种攻击方式能够以很小的代价达到攻击的目的。与针对 SDN 控制器和流表的 DoS 攻击方式相比,这种攻击方式不仅攻击代价小,而且简单、高效。

2.3　实例分析

为证明交换机控制代理拒绝服务攻击的可行性,并测试控制代理的吞吐量,本文在物理网络环境下进行了一系列的实验。网络中的主机使用的是常规的台式电脑。交换机的型号是 Pica8 P-3297,控制器使用的是 Floodlight1.2,运行在一台机架式服务器上。服务器的基本配置信息包括:Ubuntu 16.04 操作系统,2 颗 Intel Xeon E5-2650 v2 CPU,主频为 2.6 GHz,32 GB 运行内存。在实验中,交换机和控制器之间使用 OpenFlow 1.3 协议进行通信。实验过程中使用 packETH 来构造各种类型的数据包。packETH 是一个图形界面和命令行结合的数据包构造工具,本文可以在攻击者控制的主机上运行 packETH,随机地给数据包各个字段赋值。随后,根据实验需求灵活地设置数据包的发送速率。此外,本文使用命令行工具 Top 来测量交换机的负载状况。Top 所测量进程的 CPU 占用率是指该进程运行所占用的 CPU 时钟周期数在总的 CPU 时钟周期数中所占的比率。在对称多处理(Symmetrical Multi-Processing,SMP)环境中,进程 CPU 的占用率是运行该进程的所有 CPU 核心上的占用率的和。当 CPU 过载时,进程的 CPU 占用率可能超过 100%。

本文共进行了 5 组实验,每组实验的攻击速率从 100 package/s 递增到 500 package/s,其他条件都保持不变。为了比较交换机在正常工作和攻击状态下的负载情况,本文在启动测量的 10 s 之后以设定的速率发起攻击,攻击持续 60 s。同时,为了观测控制代理 DoS 攻击对网络的持续影响,本文还测量了攻击结束后 60 s 时间内的交换机 CPU 使用率。根据交换机的架构,本文控制代理的主要组件是 ovs-vswitchd 进程。所以,ovs-vswitchd 进程的 CPU 使用率能够直接反映交换机控制代理的负载状况。因此,本文将 ovs-vswitchd 进程的 CPU 使用率提取出来,其结果如图 4 所示。

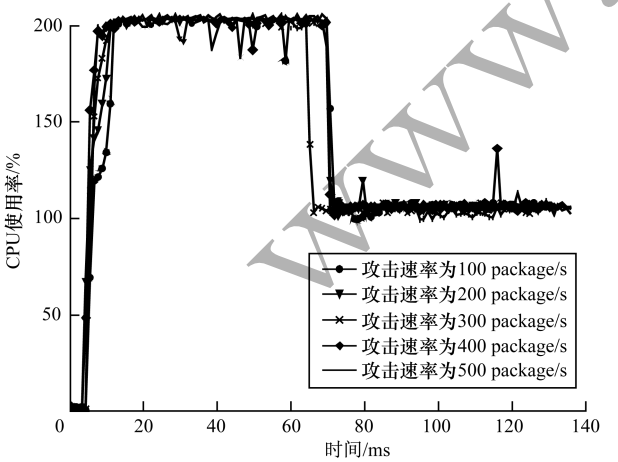

图 4　不同攻击速率下 ovs-vswitchd 进程的 CPU 使用率

根据图 4 可以得出以下结论:

1)在攻击发起之前,即网络处于空闲状态时,交换机控制代理主要处理的是 LLDP 数据包,ovs-vswitchd 进程的 CPU 占用率保持在 7% 左右的水平。

2)当攻击发起时,ovs-vswitchd 进程的 CPU 占用率呈现跳跃式增长,并始终保持在 200% 左右。由于实验中使用的交换机 CPU 是双核,因此在攻击过程中,交换机控制代理满负载运行。

3)在攻击结束后的 60 s 内,交换机控制代理仍然保持在 100% 左右。这表明交换机控制代理在这段时间中还需要继续处理在交换机中累积的 Packet-In 请求消息。

为说明交换机控制代理 DoS 攻击给网络通信带来的影响,本文测量了主机 A 与主机 B 在不同状况下通信的往返时延,实验结果如图 5 所示。在空闲状态下,主机 A、B 之间的往返时延始终保持在 1 ms 左右。随着攻击速率的增加,主机 A、B 之间的往返时延也逐渐增加。当攻击速率达到500 package/s 时,主机 A、B 之间出现了 60% 的丢包率。此时,交换机控制代理已经无法正常处理 A、B 之间通信的数据包。在攻击结束之后的一段时间内,主机 A、B 之间的通信恢复正常,往返时延与空闲状态几乎相同。实验结果表明,当交换机控制代理满负载运行时,交换机无法及时处理网络中的流量请求。

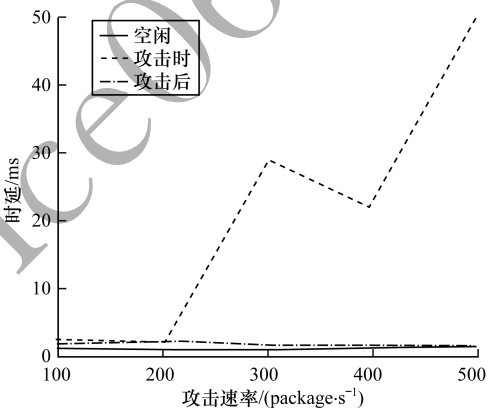

图 5　不同攻击速率下的往返时延

根据上文实验结果可知,攻击者可以通过控制网络中的主机对交换机控制代理发起拒绝服务攻击。这种攻击将会导致以下后果:交换机 CPU 持续工作在高负载状态下,会缩短它的使用寿命;来自被攻击交换机下的正常网络请求无法及时得到响应。

3　层次化多阈值攻击检测算法

针对 SDN 基础设施的 DoS 攻击是目前 SDN 安全研究的热点。在现有的研究中,FloodGuard[6] 和 FloodDefender[10] 提出根据 Packet-In 消息的实时速率和基础设施的资源利用率(控制器 CPU、内存、交换机缓存)来检测 SDN 中数据平面-控制平面饱和攻击,但是没有给出具体的算法和阈值的设定方法,也无法定位攻击发生的位置。FloodShield[16] 使用源

地址验证方法检测数据平面伪造的数据包,无法应对目的地址伪造的流量。MinDoS[17]使用综合评价因子对所有交换机进行攻击评估,但是在应对突发流量情况时,可能会产生错误的评估结果。因此,这些方法都无法快速检测并定位交换机控制代理 DoS 攻击。

在 SDN 中,控制器在正常网络状况下接收到来自于 OpenFlow 交换机的 Packet-In 消息保持在一个较低的速率,并且比较随机。但是,在交换机 DoS 攻击状况下,被攻击交换机会频繁地向控制器发送 Packet-In 消息,消耗控制代理的资源。因此,交换机在被攻击状态下 Packet-In 消息的速率相对较高,并且在一段时间内都保持在一个较高的水平上。此外,当网络中突发流量产生时,来自于交换机的 Packet-In 消息的速率也会较高,但是通常持续时间较短。为了快速检测交换机控制代理 DoS 攻击,定位攻击发生的位置,根据 SDN 网络中的这些特征,本文提出了层次化多阈值攻击检测(Hierarchical Multi-threshold Detection,HMTD)算法。与上文提到的方法相比,HMTD 算法的主要创新是通过使用两阶段的阈值,既能快速响应网络状态的突变,也能快速地定位攻击可能发生的位置。

算法 1 层次化多阈值攻击检测算法

输入 Packet-In 消息 M,控制器接收的 Packet-In 消息的数量 *Count*,控制器接收的 Packet-In 消息的速率 *CurrentRate*,由交换机端口发送的数据包所导致的 Packet-In 消息的速率 *SwitchRate*,控制器接收 Packet-In 消息的阈值 δ_1,交换机端口数据包导致的 Packet-In 消息阈值 δ_2

输出 数据包操作指令 C(Command)

```
1. Count = 0, CurrentRate = 0,  SwitchRate = 0
2. for each M do
3. Count + +
4. if Count %  N  = = 0 then
5. CurrentRate  =  Compute_Controller_Packet-In_Rate( )
6. if CurrentRate  >  δ1 then
7. SwitchRate  =  Compute_SwitchPort_Packet-In_Rate( )
8. if SwitchRate  >  δ2 then
9. Logger. info("Warn!!!")
10. return Command. stop
11. end if
12. end if
13. else
14. return Command. continue
15. end if
16. end for
```

算法 1 描述了交换机 DoS 攻击检测的原理,其基本步骤如下:

步骤 1 初始化 Packet-In 消息计数器数量 *Count*、控制器端的 Packet-In 消息速率 *CurrentRate* 和交换机各端口上报的 Packet-In 消息的速率 *SwitchRate*(第 1 行)。

步骤 2 当控制器接收到 Packet-In 消息时,首先更新计数器 *Count* 的值(第 2 行、第 3 行)。控制器每收到 N 个 Packet-In 消息时,计算控制器端的 Packet-In 消息速率 *CurrentRate*(第 4 行、第 5 行)。

步骤 3 当 *CurrentRate* 的值超过给定阈值 δ_1 时,计算每个交换机各端口上报的 Packet-In 消息的速率 *SwitchRate*(第 6 行、第 7 行)。

步骤 4 如果 *SwitchRate* 的值超过了给定的阈值 δ_2,控制器将在终端发出消息,告知网络管理员可能发生了交换机控制代理 DoS 攻击(第 7 行~第 11 行)。

需要注意的是,N、δ_1 和 δ_2 的取值决定了攻击检测的准确度。本文根据网络的规模和流量请求等信息来设定 N 的值,使计算的结果能够更加准确地反映网络中的负载变化。δ_1 的值应该根据网络中交换机的数量、交换机控制代理的吞吐量和控制器的处理能力来设置。如果 δ_1 的值取得太大,一些低速的 DoS 攻击不能被检测。如果 δ_1 的值太小,也会影响算法的执行效率。同样地,δ_2 的值应该根据交换机端口的数量、交换机控制代理的吞吐量和网络中流请求的分布情况来设置。

本文给出一个 N、δ_1 和 δ_2 取值的参考方法。假设网络中有 n 个 OpenFlow 交换机,其中第 i 个交换机 s_i 的控制代理的吞吐量为 T_{si},端口数量为 m,控制器的吞吐量为 C。本文可以根据网络中的历史流量数据求得过去一段时间中第 i 个交换机的平均 Packet-In 数量 f_{s_i}。本文假设控制代理或者控制器的吞吐量超过 80% 时,需要对 Packet-In 进行检测。因此,N、δ_1 和 δ_2 的取值可以分别根据式(1)~式(3)来确定。在实际网络应用的过程中,本文还可以根据网络运行的状况,设计一些优化算法来动态地调整 N、δ_1 和 δ_2 的值。

$$N = \sum_{i=1}^{n} f_{s_i} \tag{1}$$

$$\delta_1 = \min\{0.8\sum_{i=1}^{n} T_{s_i}, 0.8C\} \tag{2}$$

$$\delta_2 = \min\left(\frac{\sum_{i=1}^{n} f_{s_i}}{nm}, \frac{0.8\sum_{i=1}^{n} T_{s_i}}{nm}\right) \tag{3}$$

4 实验结果与分析

为验证 HMTD 算法的有效性,本文在 Floodlight 控制器中实现了 HMTD 算法,并在物理交换机环境进行了测试,实验环境和硬件参数与 2.2 节一致。根据拓扑结构和交换机控制代理的吞吐量,本文将 N 的值设置为 500,δ_1 的值设置为 400,δ_2 的值设置为 50。即控制器每收到 500 个 Packet-In 消息计算一次

Packet-In 消息到达控制器的速率,当 *CurrentRate* 值超过 400 时,本文判断需要对网络中的 Packet-In 消息进行进一步的分析来确认是否发生了 DoS 攻击。此时,根据算法 1 中的描述计算 SwitchRate 的值。如果 SwitchRate 的值大于 50,本文认为该交换机端口产生了针对交换机 CPU 和控制代理的拒绝服务攻击。

在实验中,本文依然通过控制与交换机直连的主机,使用 packETH 构建数据包发送到交换机中,数据包发送速率设置为 400 package/s。控制器在启动时启用 switchdefense 模块,实验结果如图 6 所示。当交换机数据包发送速率为 400 package/s 时,在控制器一端计算的 *CurrentRate* 会超过 400。因为网络中除了来自于主机流量的 Packet-In 请求之外,同时还有其他的 Packet-In 请求消息,比如包含了 LLDP 数据包的 Packet-In 消息。根据算法 1,会进一步计算交换机每个端口所产生的 Packet-In 请求。由于攻击产生的数据包都是从同一个交换机端口发送到网络中,因此该端口的 *SwitchRate* 也会超过 50。所以,算法 1 会通过控制器终端发出警告信息,告知用户网络中可能发生了交换拒绝服务攻击,攻击流量可能来源于交换机 < 00:00:00:00:00:00:00:01 > 的 1 号端口。如果一段时间内 switchdefense 持续上报该端口的警告信息,则认为该交换机端口发生了 DoS 攻击。实验结果表明,HMTD 算法能够及时地发现可疑的 DoS 攻击并准确地定位到攻击的位置。

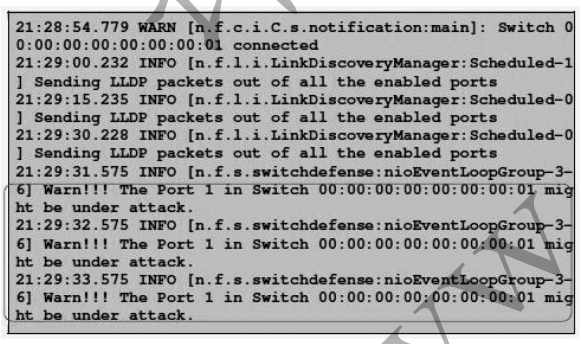

图 6　交换机 DoS 攻击检测结果

当网络管理员接收到警告信息时,本文提供2 种方法来应对这种攻击。第 1 种方法是网络管理员可以通过下发流表的方式,将该端口的流量定向到入侵检测系统进行深度分析,以进一步确定是否发生该类 DoS 攻击。如果确认是攻击流量,则将该交换机端口隔离,阻止流量进入网络中。使用这种方法,能够对交换机 DoS 攻击行为进行准确判断,但是耗时较长,也无法及时缓解针对交换机控制代理的 DoS 攻击。第 2 种方法是网络管理员在持续接收到同一个或者几个交换机端口的警告信息时,通过下发流规则的方式在交换机一端直接丢弃随后一段时间内的网络流量。使用这种方式,能够保证网络中其他位置的正常用户网络流量请求及时得到响应。但是,可能导致可疑端口所连接的正常用户的流量请求无法得到响应。因此,本文可以根据交换机控制代理的负载情况在不同的阶段部署这 2 种策略来应对交换机控制代理 DoS 攻击,即交换机控制代理还没有达到饱和状态时,使用第 1 种攻击防御策略,否则,网络管理员可以部署第 2 种策略来对抗这种攻击。

已有的方案在处理 SDN 基础设施 DoS 攻击时,都是通过将 Packet-In 消息进行缓存,再使用不同的调度策略处理 Packet-In 消息,以达到缓解 DoS 攻击的目的。HMTD 算法通过统计 Packet-In 消息的数量并计算其速率来实现对 DoS 攻击的检测。从理论上讲,Packet-In 消息速率的计算所耗费的时间要远小于 Packet-In 消息缓存与调度的时间。因此,该算法能够快速、实时地响应网络状态的突变。在攻击负载很大的情况下,能够达到更好的效果。

5　结束语

本文通过分析 SDN 交换机架构,提出一种针对交换机控制代理的 DoS 攻击,并设计一种层次化多阈值的攻击检测算法实时检测该攻击。同时,给出 2 种不同的策略应对不同负载程度下的交换机控制代理 DoS 攻击。下一步将设计动态的阈值选取方法,以准确检测控制代理 DoS 攻击并应对网络的动态变化。

参考文献

[1] MCKEOWN N,ANDERSON T,BALAKRISHNAN H,et al. OpenFlow:enabling Innovation in campus networks[J]. ACM SIGCOMM Computer Communications Review,2008,38(2):69-74.

[2] GAO Xing, GU Zhongshu, KAYAALPP M, et al. ContainerLeaks:emerging security threats of information leakages in container clouds[C]//Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks. Washington D. C., USA: IEEE Press, 2017:215-136.

[3] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed control platform for large-scale production networks [C]//Proceedings of the 7th USENIX Symposium on Networked Systems and Design and Implementation. Washington D. C., USA: IEEE Press, 2010:251-264.

[4] JAIN S,KUMAR S,MANDAL S,et al. B4:experience with a globally-deployed software defined wan [J]. ACM SIGCOMM Computer Communication Review, 2013,43(3):3-14.

[5] PFAFF B, PETTIT J, KOPONENT, et al. The design and implementation of open vSwitch[C]//Proceedings of the 12th USENIX Symposium on Networked Systems and Design and Implementation. New York, USA: ACM Press, 2015: 117-130.

[6] WANG Haopei, XU Lei, GU Guofei, et al. FloodGuard: a DoS attack prevention extension in software-defined networks[C]//Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks. Washington D. C., USA: IEEE Press, 2015: 239-250.

[7] HONG Sunming, XU Lei, WANG Haopei, et al. Poisoning network visibility in software-defined networks: new attacks and countermeasures[C]// Proceedings of Internet Society Network and Distributed System Security Symposium. San Diego, USA: [s. n.], 2015: 123-132.

[8] JERO S, KOCH W, SKOWYRA R, et al. Identifier binding attacks and defenses in software-defined networks[C]//Proceedings of the 26th USENIX Security Symposium. New York, USA: ACM Press, 2017: 485-496.

[9] SHIN S, YEGNESWARAN V, PORRAS P, et al. Avant-guard: scalable and vigilant switch flow management in software-defined networks[C]//Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: ACM Press, 2013: 413-424.

[10] GANG Shang, PENG Zhe, XIAO Bin, et al. Flooddefender: protecting data and control plane resources under SDN-aimed dos attacks[C]// Proceedings of INFOCOM IEEE Conference on Computer Communications. Washington D. C., USA: IEEE Press, 2017: 1-9.

[11] WANG An, GUO Yang, HAO Fang, et al. Scotch: elastically scaling up SDN control-plane using vswitch based overlay[C]//Proceedings of the 10th ACM International Conference on Emerging Networking Experiments and Technologies. New York, USA: ACM Press, 2014: 403-414.

[12] LENG Junyuan, ZHOU Yadong, ZHANG Junjie, et al. An inference attack model for flow table capacity and usage: exploiting the vulnerability of flow table overflow in software-defined network[J]. Water Air and Soil Pollution, 2015, 85(3): 1413-1418.

[13] QIAN Ying, YOU Wangqing, QIAN Kai. OpenFlow flow table overflow attacks and countermeasures[C]// Proceedings of European Conference on Networks and Communications. Washington D. C., USA: IEEE Press, 2016: 205-209.

[14] SONCHACK J, DUBEY A, AVIV A J, et al. Timing-based reconnaissance and defense in software-defined networks[C]//Proceedings of the 32nd Annual Conference on Computer Security Applications. New York, USA: ACM Press, 2016: 89-100.

[15] AKHUNZADA A, AHMED E, GANI A, et al. Securing software defined networks: taxonomy, requirements, and open issues[J]. IEEE Communications Magazine, 2015, 53(4): 36-44.

[16] 张梦豪, 毕军, 白家松, 等. 针对 SDN 基础设施的拒绝服务攻击防护框架[J]. 东南大学学报(自然科学版), 2017, 47(增刊): 1-8.

[17] 王涛, 陈鸿昶, 程国振. 基于网络资源管理技术的 SDN DoS 攻击动态防御机制[J]. 计算机研究与发展, 2017, 54(10): 2356-2368.

编辑 索书志