

DOI: 10.3785/j.issn.1008-973X.2020.08.012

软件定义网络中源地址验证绑定表安全

李冬, 鲁喻, 于俊清

(华中科技大学网络与计算中心, 湖北 武汉 430074)

摘要: 为了提高软件定义网络(SDN)中 IPv6 源地址验证(SAVI)绑定表的安全性, 从地址分配机制(AAM)消息验证、绑定项更新和拒绝服务(DoS)攻击防御三方面对绑定表进行保护. 基于 SDN 控制器构建 AAM 消息验证表, 记录交换机端口、MAC 地址、主机 IP 地址等信息; 建立 DHCPv6 和 SLAAC 这 2 种地址配置报文的验证模型, 下发表监听路由器通告(RA)消息, 获取 DHCPv6 request/reply 报文和 NS/NA 报文, 基于 AAM 消息验证表验证报文中的地址信息; 针对网络的动态变化建立绑定信息监听和更新机制, 监听主机离线或者主机 IP 失效事件, 及时更新绑定信息, 保证绑定表和实际网络信息的一致性; 基于 OpenFlow 多级流表建立交换机端口限速表, 防止拒绝服务攻击. 实验结果表明, 本方案能够有效防御多种针对绑定表的伪造 AAM 报文攻击, 及时更新绑定表信息, 提高 AAM 消息的处理效率.

关键词: 软件定义网络; 源地址验证; 绑定表; IPv6 安全; 地址分配

中图分类号: TP 393 **文献标志码:** A **文章编号:** 1008-973X(2020)08-1543-07

Security of source address validation improvement binding table in software defined network

LI Dong, LU Yu, YU Jun-qing

(Network and Computation Center, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Address assignment mechanisms (AAM) packet validation, binding entry updating and denial of service (DoS) attack mitigation were considered to improve the safety of IPv6 source address validation improvement (SAVI) binding table in software defined network (SDN). AAM packet validating table is established in SDN controller to record switch port, MAC address and host address. AAM packet validation procedure for DHCPv6 and SLAAC is built, which sends out flow rules for router advertisement (RA) snooping to collect DHCPv6 request/reply packets and NS/NA packets, and verifies IP address in these packets based on AAM packet validating table. The monitoring and updating mechanism of binding entries is established to adapt to dynamic network, in order to detect events such as host offline and IP failure, update binding table in time and ensure consistency between SAVI binding table and actual network information. The traffic rate limiting table of switch port is set up based on the Openflow muti-level flow table to defend against DoS attack. Experimental results show that the proposed procedure can mitigate various attacks which forge AAM packets to break SAVI binding table, update SAVI binding table records in time and improve AAM packets processing efficiency.

Key words: software defined network; source address validation; binding table; IPv6 security; address assignment

随着世界逐步迈入万物互联时代, IPv6 将逐步取代 IPv4 支撑互联网的发展. 由于 TCP/IP 协议栈的设计缺陷, 基于伪造源地址的分布式拒绝服

务 (distributed denial of service, DDoS) 攻击给互联网带来了难以估量的损失^[1]. 源地址验证技术在接入、域内、域间多个层面过滤伪造源地址报文,

收稿日期: 2019-09-18.

网址: www.zjujournals.com/eng/article/2020/1008-973X/202008012.shtml

基金项目: 国家重点研发计划资助项目(2017YFB0801703); 赛尔网络下一代互联网技术创新资助项目(NGII20170408).

作者简介: 李冬(1979—), 男, 讲师, 从事计算机网络和网络安全研究. orcid.org/0000-0001-6431-8980. E-mail: lidong@hust.edu.cn

能够从体系结构上根本地解决这一问题^[2]。Bi 等^[3]建设 IPv6 实验床,开展源地址验证技术部署工作。源地址验证技术包括接入网、域内和域间验证。在传统接入网中主要通过邻居发现 (neighbor discovery, ND) 报文嗅探和动态主机配置协议 (dynamic host configuration protocol, DHCP) 报文嗅探在以太网交换机上建立绑定关系来实现^[4],包括无状态地址分配^[5]、动态地址分配^[6]和混合地址分配^[7]等多种场景。域间和域内有多种验证方法,例如: Bi 等^[8]提出的基于二层状态机对域间数据报文进行签名验证, Li 等^[9]提出的建立域内分布式协助机制对伪造源地址报文进行过滤。Jia 等^[10]对各种验证方法进行总结,发现在接入网络中实现源地址验证非常重要,目前的主要问题是缺乏灵活性、部署成本较高,而软件定义网络 (software defined network, SDN) 技术能够较好地规避这些问题。

以 Openflow 为代表的 SDN 技术实现了控制逻辑和转发逻辑的解耦^[11],网络的灵活控制和可编程性使得源地址验证较易实现和部署^[12]。Yao 等^[13-20]提出多种 SDN 下源地址验证方法,其中 Yao 等^[13]采用 openflow/NOX 架构的虚拟源地址验证边界 (virtual source address validation edge, VAVE) 验证方案,方案基于源地址验证技术,采用构建过滤规则生成器、计算非法地址空间、计算流表路径等实现 VAVE 的源地址验证; Liu 等^[14]设计实现 SDN 中接入网源地址验证 (source address validation improvement, SAVI) 方案 SDN-SAVI,方案通过监听地址分配机制 (address assignment mechanisms, AAM) 消息构建 <SwitchPort, MAC, IP> 三元组绑定表,并根据绑定表生成对应的验证规则下发到交换机中,以达到主机级源地址验证的目的。另外,还有基于 SDN-SAVI 提出的面向 SDN 动态源地址验证方案^[15],该方案考虑 OpenFlow 交换机的转发性能问题,通过监测主机行为针对性下发验证规则,提高交换机的转发性能; Li 等^[19-20]提出在 IPv6 物联网和传感网中基于 SDN 进行源地址验证的方案。在这些方案中,绑定表都是实现源地址验证技术的关键,但是未考虑如何保障绑定表的安全。

针对目前 SDN 中源地址验证绑定表安全问题,提出绑定表安全保障机制。主要工作如下: 1) 构建 AAM 消息验证表和 RA 消息处理模块,通过验证表记录端口主机信息,包括交换机端口、

MAC 地址、主机所有的 IP 地址以及 IP 地址对应的信息等,通过建立验证表和处理路由通告消息为 AAM 消息验证提供依据。2) 设计并实现 DHCPv6 和 NS/NA 消息的验证和处理模块,依据 AAM 消息验证表和 RA 消息的处理结果并参考地址配置过程,对 DHCPv6 和 NS/NA 消息进行验证。采用先到先服务的策略并使用控制器进行辅助响应的方法对 NS/NA 消息进行处理。3) 设计并实现绑定表的更新模块,考虑网络的动态变化,监听主机离线或者主机 IP 失效事件以及及时更新绑定表。4) 优化绑定表的监听模块,设计三级流表结构,利用 meter 表实现端口限速。5) 通过伪造多种类型的 AAM 消息模拟攻击实验,验证方案的有效性;设计主机离线和 IP 失效的场景,验证更新方法的有效性,并通过对比试验验证监听优化方案的优越性。

1 AAM 消息验证与处理

1.1 AAM 验证表

AAM 验证表是验证 AAM 报文的依据,主要功能是记录网络设备端口和主机对应信息,并根据记录对主机发出的 AAM 消息进行验证。验证表基本结构为: <SwitchPort, MAC, IPList>。其中, SwitchPort 字段是主机直连的交换机端口,包含交换机 datapathId 和端口号; IPList 的基本结构为 <IP, Valid-Lifetime, Binding-Status, Type>, Valid-Lifetime 为 IP 有效时间, Binding-Status 为绑定状态 (0 表示已绑定, 1 表示尚未绑定), Type 为 IP 地址类型,主要地址类型有链路本地地址和全球单播地址。

为了对 AAM 消息进行验证,须获取主机地址配置方式,因而须构建监听流表监听 RA 消息,并对来自主机端口的 RA 消息进行监测。将 RA 消息的 Managed Address Configuration 字段,记为 M 标识,根据 M 标识获取地址配置信息,区分端口主机是采用无状态地址自动配置 (stateless address autoconfiguration, SLAAC) 方式,还是 DHCPv6 地址配置方式,进而进行分类验证。

1.2 DHCPv6 消息的验证与处理

由于缺乏对 DHCPv6 消息的验证,绑定表会绑定伪造 IP 甚至破坏已绑定的表项,因而须对 DHCPv6 相关的消息和报文进行处理。

1) DHCPv6 消息的验证。根据 DHCPv6 消息类

型分别对服务端消息和客户端消息采用不同的验证方法. 在 DHCPv6 服务器接入 SDN 中时, 将 DHCPv6 服务器的直连交换机端口在控制器中设置为 Trust_Port 以验证 DHCPv6 消息, 来自 Trust_Port 的消息是合法的, 否则是伪造的.

对于客户端消息 (Solicite、Request、Confirm、Decline), 根据该消息的上联交换机端口对应的 AAM 验证表项, 验证其源 IP、MAC 是否匹配. 对于 Confirm 和 Decline 消息还须验证其身份联盟 (identity association, IA) 地址选项 IP 是否匹配. 若不匹配, 则该消息是伪造的, 记录异常并丢弃; 若匹配, 转入下一步处理.

2) DHCPv6 消息的处理. 利用控制器处理 Request 消息的具体过程分为 2 步: 1) 控制器收到 Request 消息后解析其 IA 选项提取 IP 地址; 2) 将该 IP 与 Request 消息作为键值对存入 Map 中. 处理 Reply 消息的过程如下: 1) 当控制器收到 Reply 消息时, 解析其 IA 选项 IP 地址; 2) 根据 Reply 和该 IP 对应的 Request 消息构建绑定项. 这种处理机制存在 2 个安全隐患: 当 2 个 Request 消息 IA 选项 IP 地址相同时, 可能会导致绑定错误, 且主机不能对外通信; 若主机 H_1 配置的 IP 地址已经被主机 H_2 占用, 直接绑定的结果会导致主机 H_2 的绑定项被覆盖, 以至于主机 H_2 无法对外通信.

由此可知, 确定 Request 和 Reply 是否为一对请求和响应需要更加精确的计算, 不能根据 Request/Reply 消息直接绑定. 本方案为每个主机端口建立待处理消息集合, 并将客户端消息加入对应端口待处理消息集合. 当收到 Reply 消息时, 控制器根据其目的地址找到对应的交换机端口, 并从该端口待处理消息集合中找到与 Reply 消息的 Transaction ID 一致的消息, 根据响应消息类型进行处理, 处理动作如表 1 所示.

表 1 DHCPv6 报文处理方法

Tab.1 Processing action of DHCPv6 packets

报文类型	处理方法
Request	更新 AAM 验证表, 绑定状态设为 1
Decline	删除 AAM 验证表和绑定表对应项
Confirm	修改 AAM 验证表交换机端口绑定状态设为 1

1.3 NS/NA 消息的验证与处理

无论网络中主机地址配置方式是 SLAAC 还是 DHCPv6, 均须通过重复地址检测 (duplicate ad-

dress detection, DAD) 来判断 IP 地址是否可用, 在 DAD 过程中使用的请求和响应消息分别为 NS、NA 消息.

1) NS/NA 消息的验证. 对 NA 消息的验证主要是对其源 IP 地址、MAC 地址和目标地址进行验证, 根据上发 NA 消息的交换机端口对应的 AAM 验证表项, 检查 NA 消息源 IP 地址和 MAC 地址是否匹配; 同时判断其目标地址是否为组播地址, 若不是, 还须检查其目标地址是否匹配.

在对 NS 消息进行验证前, 根据其携带信息以及来源将其分为 4 类: 1) 无须验证的 NS 消息, 包含指定了源地址的 NS 消息, 即 NS 消息 IPv6 分组的源地址不是空地址, 以及来自非边缘交换机端口的 NS 消息; 2) 链路本地 NS 消息, 其源地址为空地址, 且目标地址是链路本地地址; 3) DHCPv6 端口 NS 消息, 其源地址为空地址, 目标地址不是链路本地地址, 并且上发 NS 消息的交换机端口属于 DHCPv6 端口; 4) SLAAC 端口的 NS 消息, 其源地址为空地址, 目标地址不是链路本地地址, 且上发 NS 消息的交换机端口属于 SLAAC 集合. 其中链路本地 NS 消息、DHCPv6 集合 NS 消息和 SLAAC 集合 NS 消息都属于 DAD 消息, 须进行验证.

如图 1 所示为本地链路地址验证流程, 交换机端口收到本地链路 NS 消息, 表示有新主机上线. 此时找到 AAM 验证表中该端口的原主机, 并对其发起邻居不可达检测 (neighbor unreachable detection, NUD) 探测检查原主机是否在线, 若原主机在线, 说明该 NS 消息为伪造消息, 否则为合法消息.

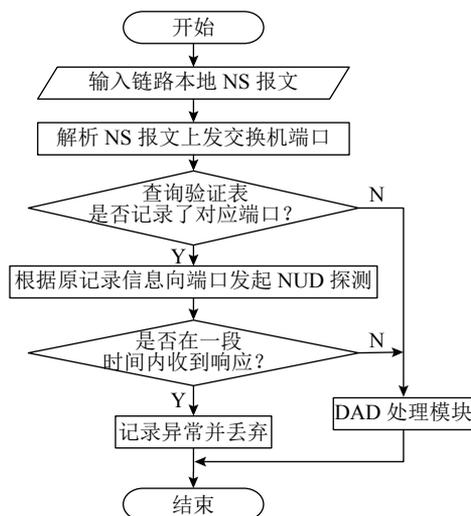


图 1 本地链路地址验证流程

Fig.1 Validation procedure of local link address

DHCPv6 端口的 NS 消息是主机通过 DHCPv6 地址配置方式得到 IP 地址后, 在该 IP 地址进行解析时发出的消息, 因此根据 AAM 验证表中该端口是否有绑定状态为 1 的 IP 与 NS 目标地址相同来判断消息是否合法。

SLAAC 端口的 NS 消息是由 SLAAC 地址配置的主机发出的 DAD 请求, 考虑到在实际网络中, 临时地址的有效时间通常大于 RA 消息的重传时间, 因此对于同一个 RA 消息前缀, 允许只有一个临时地址与之对应. SLAAC 端口 NS 消息中合法的消息须满足以下 3 个条件: 1) 该端口已申请本地链路地址; 2) 在端口绑定前缀中存在与该消息目标地址前缀匹配的信息; 3) 在 AAM 验证表中该前缀对应交换机端口的 IP 地址个数小于 2 个。

2 绑定表更新与监听优化

2.1 绑定表更新

考虑到网络的动态变化, 会发生 IP 地址失效和主机离线事件, 在事件发生时须及时发现并对对应绑定项进行解绑. IP 地址类型主要有本地链路地址、SLAAC 配置地址和 DHCPv6 配置地址。

本地链路地址永久有效, SLAAC 配置的地址分为临时地址和非临时地址, 都在收到 RA 消息后更新其有效时间, 其中实际非临时地址有效时间不大于 RA 指定的有效时间. 考虑到实际网络非临时地址有效时间大于 RA 重传时间, 此处将其有效时间设置为 RA 指定的有效时间是可靠的. DHCPv6 配置地址有效时间则由 Request、Rebind、Renew 和对应的 Reply 消息决定. 更新模块监听和解析对应消息, 在 AAM 验证表中设置 IP 地址有效时间并周期检查, 当发现 IP 地址失效时即对该 IP 进行解绑。

如图 2 所示为绑定表绑定记录更新流程, 主要是监听交换机主机接入端口状态变化, 并对存活主机发起 NUD 探测来检测主机是否存活, 对不在网络中的主机进行解绑。

2.2 基于端口限速的监听优化

当恶意主机发送大量 AAM 消息的时候, 网络和控制平面负载都会增大, 导致控制器拒绝服务. 针对这个问题, 目前的解决方案是利用 meter 表对交换机进行限速, 然而实际网络中每个交换机所连主机个数并不相同, 恶意大流量 AAM 消息

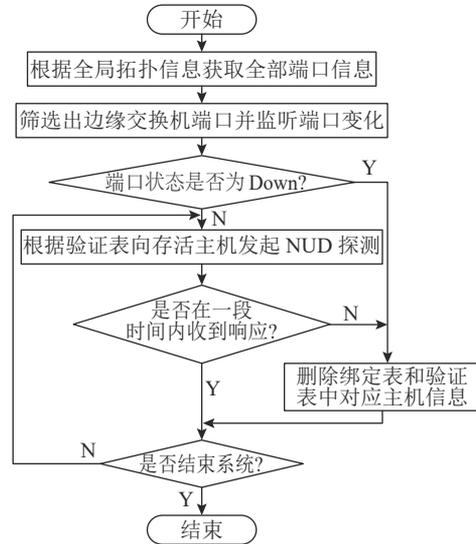


图 2 绑定表项更新流程

Fig.2 Procedure of binding entries updating

会耗尽带宽资源导致正常主机无法正常访问网络。

为了实现限速, 同时解决拒绝服务问题, 本方案设计端口流表, 基于多级流表结构实现对每个主机的限速, 解决限速导致的拒绝服务问题. 例如, 在开放接入端口 p_1 、 p_2 、 p_3 的交换机中, 匹配主机端口 p_1 、 p_2 、 p_3 的数据包分别由不同的 meter 表进行限速, 其他数据包由 table_miss 直接交给控制器处理, 无须进行限速。

3 实验结果与分析

为了验证绑定表安全防御机制的有效性, 实验采用 Floodlight 作为控制器、基于 Mininet 和 Openvswitch 搭建 SDN 安全攻击模拟平台, 并开发脚本程序, 部署相关实验, 所构建的实验网络拓扑如图 3 所示. 在实验中由交换机 s_7 发送 RA 消息, 当网络采用 DHCPv6 地址配置方式时, 主机 H_1 作为服务器。

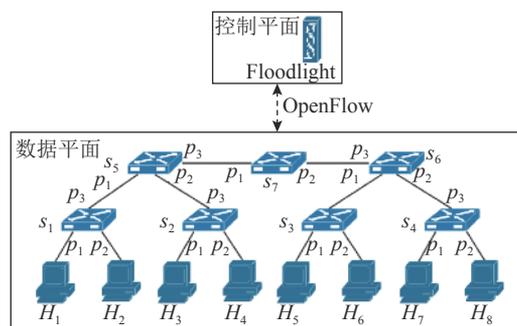


图 3 绑定表安全实验测试平台

Fig.3 Platform of binding table security testing

3.1 AAM 消息验证分析

为了评估验证方案检测伪造 AAM 消息的效果,实验分别设置 SLAAC 地址配置和 DHCPv6 地址配置、主机使用临时地址和不使用临时地址的场景.在不同环境中,伪造不同的 AAM 消息,检查验证方案是否有效以及是否会对绑定表产生影响.在实验中主机 H_2 作为攻击者,发出各种伪造 AAM 消息,伪造消息类型如表 2 所示.如表 3 所示为不同条件下不同伪造消息攻击实验的结果.检测结果表示“验证方案能否检测出伪造消息”/“对绑定表是否有影响”.须说明:1)对于消息 2,伪造的 Request 消息在 4 种情况下都不能检测出来,但是其对应的 Reply 能被检测出来,因此不会绑定伪造 IP.2)在 SLAAC 地址配置中,当主机不使用临时地址时,不能检测出消息 7、8.对于消息 7,绑定表会对其进行绑定,而当该主机再次伪造类似消息时会被检测到,即该伪造消息只有

表 2 AAM 伪造消息类型

Tab.2 Types of forged AAM messages

消息编号	伪造消息类型	描述
1	Request/Reply	伪造 MAC 地址
2	Request/Reply	伪造 IA 选项 IP 地址
3	Decline	伪造 IA 选项 IP 地址为其他主机 IP
4	NS	伪造 MAC 地址
5	NS	目标地址为本地链路地址
6	NS	目标地址为全球单播地址且前缀不匹配
7	NS	目标地址为前缀匹配的 全球单播地址且未被使用
8	NS	目标地址为其他主机 IP 地址

表 3 AAM 报文伪造攻击实验测试结果

Tab.3 Experimental results of forged AAM packets attack

编号	SLAAC		DHCPv6	
	有临时地址	无临时地址	有临时地址	无临时地址
1	能/无	能/无	能/无	能/无
2	能/无	能/无	能/无	能/无
3	能/无	能/无	能/无	能/无
4	能/无	能/无	能/无	能/无
5	能/无	能/无	能/无	能/无
6	能/无	能/无	能/无	能/无
7	能/无	不能/有	能/无	能/无
8	能/无	不能/无	能/无	能/无

在第 1 次未被检测到.对于消息 8,虽然该消息未被检测到,但是通过 DAD 模块的处理,绑定表不会对其进行绑定.3)实验结果表明,该方法能较好地保护绑定表.

3.2 绑定表更新分析

实验环境采用无状态地址配置,路由通告消息前缀信息 Valid-Lifetime 设置为 120 s, Preferred-Lifetime 设置为 80 s,如图 4 所示为主机 H_1 、 H_2 接入网络产生的绑定表.可以看出,绑定表能够正常绑定本地链路地址、临时地址和永久地址.

为了测试绑定表更新机制,在 mininet 中将主机 H_2 和交换机 s_1 断开,即让主机 H_2 离线,结果得到如图 5 所示的输出日志,说明本方案中绑定表能够监测到网络的动态变化,发现主机离线事件.

RA 消息设置的有效时间小于重发 RA 消息时间,所以主机 H_1 的全球单播地址会在 120 s 之后失效,最终 H_1 、 H_2 的绑定表项只剩 H_1 的链路本地地址存在,如图 6 所示,绑定表能够及时进行更新.

```

=====绑定表=====
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:2001:db1::200:ff:fe00:1
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:2001:db1::fd98:cd40:d6fa:4106
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:fe80::700:ff:fe00:1
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:2001:db1::200:ff:fe00:2
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:2001:db1::88d7:e53f:3a62:52c3
SwitchPort: [s1, 2]; MAC:00:00:00:00:00:02; IP:fe80::700:ff:fe00:2
    
```

图 4 绑定表原始记录

Fig.4 Original records in binding table

```

=====更新日志=====
发现主机离线, SwitchPort: s1, 2
对应主机MAC: 00:00:00:00:00:02
    
```

图 5 绑定表消息更新日志

Fig.5 Log of binding entries updating

```

=====绑定表=====
SwitchPort: [s1, 1]; MAC:00:00:00:00:00:01; IP:fe80::200:ff:fe00:1
    
```

图 6 绑定表更新后记录

Fig.6 Updated binding entries

3.3 消息监听对比分析

为了对比在相同发包条件下,无限速方案、交换机限速方案和端口限速方案对控制器负载的影响,实验基于如图 3 所示的实验环境,分别设定主机 H_1 、 H_2 以 150、20 pks/s 的速率发送 DAD 请求,相关分析如下.如图 7 所示为在无限速时 H_1 、 H_2 每秒发出的 DAD 请求和控制器每秒收到的 Packet_In 包数 n .图中, t 为时间.可以看出,主机 H_1 、 H_2 发出的 AAM 消息基本全部上传到控制器.如图 8 所示为对交换机限速时的消息数量统计,速度限制为 50 pkt/s.可以看出,交换机 s_1 每秒发

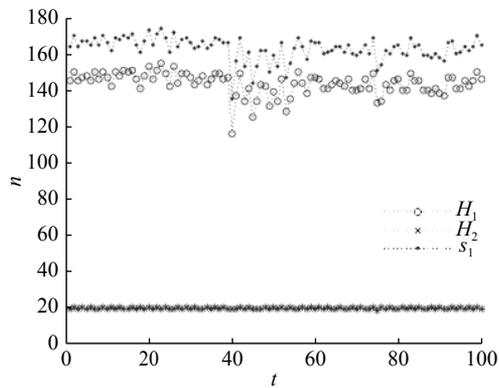


图 7 无限速方案测试结果

Fig.7 Results of scheme with no traffic rate limit

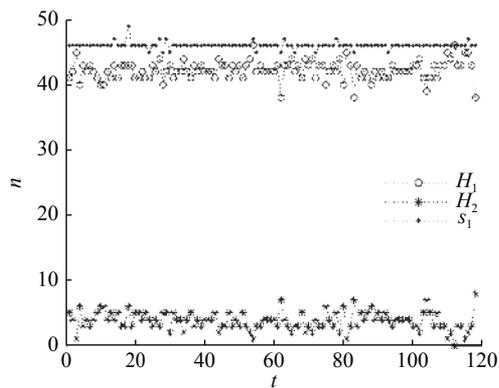


图 8 交换机限速方案测试结果

Fig.8 Results of scheme with traffic rate limit on switch

出的 Packet_In 包数被限制在每秒 50 个以下, 其中 H_1 触发的 Packet_In 约为每秒 43 个, H_2 触发的 Packet_In 约为每秒 6 个, 交换机限速对主机 H_1 、 H_2 都有影响。如图 9 所示为对交换机端口限速时的消息数量统计结果, 每个端口速度限制为 25 pkt/s。可以看出, 交换机 s_1 每秒上发的 Packet_In 包数低于 45 个。来自主机 H_1 的数据包数量约为每秒 24 个; 来自主机 H_2 的约为每秒 20 个。若主机 H_1 为恶意主机, 在无限速时 H_1 发出的大流量

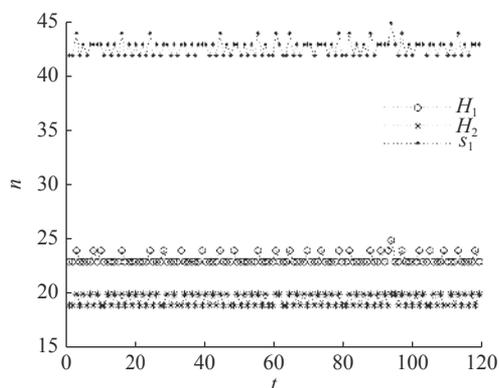


图 9 交换机端口限速方案测试结果

Fig.9 Results of scheme with traffic rate limit on switch port

AAM 消息会增加交换机和控制器之间的链路负载, 也会增加控制器处理负载; 在交换机限速时, H_1 发出的大流量 AAM 消息会侵占限速通道, 导致 H_2 的合法 AAM 消息发生丢包; 三级流表端口限速方案能缓解恶意主机 H_1 发起的 DoS 攻击, 保障正常主机 H_2 的通信通道。

4 结 语

研究 SDN 中 IPv6 源地址验证绑定表的安全问题, 设计实现绑定表安全保障机制。通过构建 AAM 报文验证表验证 AAM 消息; 监听主机在线状态和 IP 地址有效时间, 在发现主机离线和 IP 失效时及时更新绑定表。基于三级流表结构设计端口流表, 利用 meter 表实现对主机端口限速, 保障交换机与控制器的通信安全且不会产生漏绑。

下一步将从以下 3 个方面进行优化: 在安全可靠的前提下动态验证 NS 消息, 提升报文验证效率; 通过控制器采集端口数据, 初步检验主机是否在线以减少控制器主动探测发包的数量; 结合网络拓扑发现机制解决攻击者自身隐藏问题。

参考文献 (References):

- [1] SATIN A, BERNARDI P. Impact of distributed denial-of-service attack on advanced metering infrastructure [J]. *Wireless Personal Communications*, 2015, 83(3): 2211–2223.
- [2] WU J P, BI J, MARCELO B, et al. Source address validation improvement framework [EB/OL]. (2013-10-01). <https://tools.ietf.org/html/rfc7039>.
- [3] BI J, YAO G, WU J P. An IPv6 source address validation testbed and prototype [J]. *Journal of Networks*, 2009, 4(2): 100–107.
- [4] HU J L, WU Y S. Source address validation based ethernet switches for IPv6 network [C]// *IEEE International Conference on Computer Science and Automation Engineering*. Zhangjiajie: IEEE, 2012: 84–87.
- [5] BI J, YAO G, BAKER F, et al. SAVI solution for stateless address [EB/OL]. (2010-04-18). <https://tools.ietf.org/html/draft-bi-savi-stateless-00.pdf>.
- [6] BI J, WU J P, YAO G, et al. Source address validation improvement (SAVI) solution for DHCP [EB/OL]. (2015-05-01). <https://tools.ietf.org/html/rfc7513>.
- [7] BI J, YAO G, HALPERN J, et al. Source address validation improvement for mixed address assignment methods scenario [EB/OL]. (2017-02-01). <https://tools.ietf.org/html/rfc8074>.

- [8] BI J, LIU B Y, WU J P, et al. Preventing IP source address spoofing: a two-level, state machine-based method [J]. **Tsinghua Science and Technology**, 2009, 14(4): 413–422.
- [9] LI J, BI J, WU J P. Towards a cooperative mechanism based distributed source address filtering [C]// **22nd International Conference on Computer Communications and Networks**. Nassau: IEEE, 2013: 1–7.
- [10] JIA Y H, REN G, LIU Y, et al. Review of internet inter-domain IP source address validation technology [J]. **Journal of Software**, 2018, 29(1): 176–195.
- [11] NICK M, TOM A, HAIR B, et al. OpenFlow: enabling innovation in campus networks [J]. **ACM SIGCOMM Computer Communication Review**, 2008, 38(2): 69–74.
- [12] CHEN G L, HU G W, JIANG Y, et al. SAVSH: IP source address validation for SDN hybrid networks [C]// **IEEE Symposium on Computers and Communication**. Messina: IEEE, 2016: 409–414.
- [13] YAO G, BI J, XIAO P Y. Source address validation solution with OpenFlow/NOX architecture [C]// **19th IEEE International Conference on Network Protocols**. Vancouver: IEEE, 2011: 7–12.
- [14] LIU B Y, BI J, ZHOU Y. Source address validation in software defined networks [C]// **16th ACM SIGCOMM Conference**. Florianópolis: ACM, 2016: 595–596.
- [15] 周启钊, 于俊清, 李冬. SDN 环境下源地址动态验证方法研究 [J]. **通信学报**, 2018, 39(增 1): 235–243.
ZHOU Qi-zhao, YU Jun-qing, LI dong. Dynamic source address validation in software defined network [J]. **Journal of Communications**, 2018, 39(Suppl. 1): 235–243.
- [16] LI C L, WU Q, LI H W, et al. SDN-Ti: a general solution based on sdn to attacker traceback and identification in IPv6 networks [C]// **IEEE International Conference on Communications**. Shanghai: IEEE, 2019: 1550–3607.
- [17] ZHANG C Q, HU G W, CHEN G L, et al. Towards a SDN-based integrated architecture for mitigating IP spoofing attack [J]. **IEEE Access**, 2017(6): 22764–22777.
- [18] YAN Z H, DENG G S, WU J Y. SAVI-based IPv6 source address validation implementation of the access network [C]// **International Conference on Computer Science and Service System**. Nanjing: IEEE, 2011: 2530–2533.
- [19] LI X, NIU J W. A robust ECC based provable secure authentication protocol with privacy protection for industrial internet of things [J]. **IEEE Transactions on Industrial Informatics**, 2017, 14(8): 3599–3609.
- [20] LI X, NIU J W, KUMARI S, et al. A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments [J]. **Journal of Network and Computer Applications**, 2018, 103: 194–204.