软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(12):3104−3119 [doi: 10.13328/j.cnki.jos.004942]

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

一种防御 DDoS 攻击的软件定义安全网络机制*

王秀磊[1], 陈 鸣[1], 邢长友[1], 孙 志[1,2], 吴泉峰[1]

[1](解放军理工大学 指挥信息系统学院,江苏 南京 210007)

[2](海军航空工程学院青岛校区 信息管理中心,山东 青岛 266041)

通讯作者: 王秀磊, E-mail: xiuleiwang1988@126.com

摘 要: 软件定义网络的出现为防御 DDoS 攻击提供了新的思路.首先,从网络体系结构角度建模分析了 DDoS 攻击所需的 3 个必要条件:连通性、隐蔽性与攻击性;然后,从破坏或限制这些必要条件的角度出发,提出了一种能够对抗 DDoS 攻击的软件定义安全网络机制 SDSNM(software defined security networking mechanism).该机制主要在边缘 SDN 网络实现,同时继承了核心 IP 网络体系架构,具有增量部署特性.利用云计算与 Chord 技术设计实现了原型系统,基于原型系统的测量结果表明,SDSNM 具有很好的扩展性和可用性.

关键词: 网络安全;DDoS;软件定义网络;OpenFlow;网络体系结构

中图法分类号: TP393

中文引用格式: 王秀磊,陈鸣,邢长友,孙志,吴泉峰.一种防御 DDoS 攻击的软件定义安全网络机制.软件学报,2016,27(12):3104−3119. http://www.jos.org.cn/1000-9825/4942.htm

英文引用格式: Wang XL, Chen M, Xing CY, Sun Z, Wu QF. Software Defined Security Networking Mechanism to Defend Against DDoS Attacks. Ruan Jian Xue Bao/Journal of Software, 2016,27(12):3104−3119 (in Chinese). http://www.jos.org.cn/1000-9825/4942.htm

Software Defined Security Networking Mechanism Against DDoS Attacks

WANG Xiu-Lei[1], CHEN Ming[1], XING Chang-You[1], SUN Zhi[1,2], WU Quan-Feng[1]

[1](College of Command Information System, PLA University of Science & Technology, Nanjing 210007, China)

[2](Information Management Center, Naval Aeronautical Engineering Institute Qingdao Branch, Qingdao 266041, China)

Abstract: The emerging software defined networking (SDN) offers a new way to rethink the defense of DDoS attacks. In this paper the DDoS attacks are first modeled and analyzed from the perspective of network architecture, and the necessary conditions of DDoS attacks such as connectivity, concealment and aggressivity are presented. Then for breaking or limiting these necessary conditions, a software defined security networking mechanism (SDSNM) against DDoS attacks is proposed. The security mechanism is implemented in the edge SDN networks while inheriting the core infrastructure of IP network. Cloud computing and Chord technology are also employed to solve the expansibility and consistency problems. The experiments demonstrate that SDSNM is feasible and incrementally deployable.

Key words: network security; DDoS; software defined networking; OpenFlow; network architecture

近年来,分布式拒绝(distributed denial of service,简称 DDoS)攻击的发生频率、危害性与设计复杂度不断增加,已经成为当今因特网安全的最主要威胁之一[1−3].DDoS 攻击者通过俘获大量傀儡机形成僵尸网络,向目标主

* 基金项目: 国家重点基础研究计划(973)(2012CB315806); 国家自然科学基金(61379149, 61402521, 61103225); 江苏省自然科学基金(BK20140070, BK20140068); 江苏省未来网络科技计划(BY2013095-1-06)

Foundation item: National Program on Key Basic Research Project (973) (2012CB315806); National Natural Science Foundation of China (61379149, 61402521, 61103225); Natural Science Foundation of Jiangsu (BK20140070, BK20140068); Jiangsu Future Network Innovation Institute Research Project on Future Networks (BY2013095-1-06)

收稿时间: 2015-03-12; 修改时间: 2015-08-11; 采用时间: 2015-11-09; jos 在线出版时间: 2016-01-16

CNKI 网络优先出版: 2016-01-18 13:51:03, http://www.cnki.net/kcms/detail/11.2560.TP.20160118.1351.006.html

机发送大量资源请求,消耗网络的计算、存储或带宽等资源,阻止目标主机为合法用户提供正常服务.随着网络规模的不断扩大以及网络端系统性能的不断提高,DDoS 攻击峰值流量不断被刷新.例如:欧洲反垃圾邮件公司 Spambaus 网站 2013 年遭受到史上最大流量 DDoS 攻击,攻击流量峰值高达 300Gbps;2014 年,一种利用网络时间协议(NTP)进行反射放大的 DDoS 攻击,攻击流量创纪录达到 400Gbps[3].

DDoS 攻击易于实施、危害性大、隐蔽性强且难以溯源的特点使其难以防御,并在因特网中日益泛滥.尽管目前在 DDoS 攻击防御方面进行了大量研究,但依然没有找到有效的方案,DDoS 攻击防御和控制仍是网络安全领域重要的研究课题[4].本文的贡献在于:首先,从网络体系架构角度对 DDoS 攻击进行建模,抽象出 DDoS 攻击的必要条件;然后,从破坏或限制 DDoS 攻击的必要条件出发,提出一种软件定义安全网络机制(software defined security network mechanism,简称 SDSNM),并分析了该机制的性质;最后,设计了一种能够在现有 IP 网络上增量部署的实现方案,原型系统验证了 SDSNM 的可行性与可用性.

本文第 1 节介绍 SDN 在 DDoS 攻击防御领域内的相关研究.第 2 节建模分析 DDoS 攻击的必要条件.第 3 节研究为对抗 DDoS 攻击的安全网络机制应具备的性质.第 4 节详细描述 SDSNM 的框架和实现方案.第 5 节利用原型系统验证 SDSNM.最后对全文进行总结和展望.

1 相关工作

TCP/IP 设计之初只关注了连通性与鲁棒性,缺乏对安全性的考虑[5].在端到端原则[6]的指导设计下,核心网络仅仅负责分组快速转发,而将各种复杂控制和应用放在了端系统实现.这种核心简单、边缘复杂的体系结构保证了网络健壮性并使其得以迅速普及壮大,但是基于目的地址尽力而为的转发行为也为各种安全威胁提供了条件.DDoS 攻击正是利用 TCP/IP 的设计缺陷实现对其进行攻击.目前,DDoS 攻击的研究主要集中在检测方法方面[7],但是由于因特网本身的开放性、分布式控制以及潜在的源地址欺骗行为[8],即便在端系统能够准确地检测到 DDoS 攻击的发生,由于无法保证攻击分组源地址的真实性、分布式路由协议导致路由的多样性,无法确定攻击者位置,依然难以从根本上阻断 DDoS 攻击.Shenker 等人在文献[9]指出:基于密码学的方法已经可以很好地解决网络安全通信的鉴别、机密性和完整性问题[10,11],但是可用性依然无有效方案.文献[7,12]指出,其根本原因在于当前的网络本身不具备网络攻击检测和处理能力.因此,相关研究[12−14]试图通过修改现有网络体系结构,达到提高网络可用性的目的.在可问责互联网协议(accountable internet protocol,简称 AIP)中[12],每个通信实体都利用其公钥哈希值作为身份标识符,通过基于身份标识符的自认证机制确保每个接入主机身份可信,同时,阻断协议(shutoff protocol)可以对攻击主机进行定位,通知其网卡对攻击流量进行阻断.可问责与隐私保护互联网协议(accountable and private internet protocol,简称 APIP)[13]改进了 AIP,通过引入可靠性代理保证端主机分组的可靠性;端主机上报异常后,代理能够定位攻击主机并阻断其异常行为.清华大学吴建平教授等人提出的源地址认证体系结构 SAVA(source address validation architecture)[14,15]是一种层次化体系结构,通过对分组的源地址进行认证,很好地解决了 IPv4/IPv6 网络中的源地址伪造问题.对当前网络体系结构进行改造,必须在设计之初就加入对可用性的考虑,使网络体系结构本身内嵌对 DDoS 等可用性攻击的防御机制,能够从根本上抵御可用性威胁.虽然上述体系结构可以抵御网络可用性威胁,但是这些设计在大规模部署上面临着很大困难,它们都是一种革命性的设计,要求对当前网络结构进行改造,无法与现有的网络基础设施兼容.因此,研究新的体系结构和机制必须考虑增量部署需求.同时,上述研究也表明:设计下一代网络,从根本上消除 DDoS 攻击的条件,已经成为当前 DDoS 防御研究的趋势.

SDN 是一种 Clean Slate 的网络体系结构,其控制平面与数据平面分离的设计思想,为解决传统网络安全问题提供了新的思路.文献[16]利用 SDN 控制器能够获取数据平面全局信息的特性,防御基于僵尸网络的 DDoS 攻击.论文通过在控制器运行检测算法实现对网络 DDoS 攻击行为的发现,基于对网络重要服务器信息(例如 Web server,DNS server 等)的掌握,通过阻塞和重定向攻击流实现对服务器的保护.文献[17]考虑到目前源地址验证标准(SAVI)的不足,通过在 NOX 上开发应用程序来获取全局网络视图,从而实现对设备的认证.文献[18]研究了 SDN 与云计算结合环境下的 DDoS 攻击防御机制,作者指出:将 SDN 与云计算结合,一方面 SDN 中心化控

制与网络虚拟化功能能够解决云计算环境下 DDoS 防御半径延伸和网络拓扑动态变化的问题,同时,云计算的弹性资源配置性质可以扩展 SDN 控制平面计算与通信瓶颈,二者互补为解决企业网 DDoS 防御提供了新的机遇.但是当前,这些方案仅仅局限于单一 SDN 局域网,例如校园网、数据中心以及原型网络等,大规模部署缺乏实验环境.SDN 作为下一代网络候选研究方案之一,必然要研究其在大规模网络中的应用.目前,已经有相关研究考虑 SDN 在广域网中的应用,例如 Google 的 SDN 项目 B4[19],该项目使用 SDN 改造了 Google 数据中心广域网,极大地提高了 WAN 网的带宽利用率.Onix[20]已经提出一套面向较大规模真实网络的部署方案,并提出新的实现技术,为 SDN 的大规模部署提供了技术指导.基于 OpenFlow 的未来互联网测平台已经在世界各国建立起来,Internet 2[21]已完成 100G 接入链路的 SDN 网络建设,部署范围包括全美 30 多个节点.另外,我国目前也已经建成了 CERNET2,CNGI(China next generation Internet)[22]等未来网络实验平台,这些平台的建设都为 SDN 创新建设提供了良好的测试与测量平台.在推进基于 SDN 的下一代互联网体系结构建设过程中,必须从开始就加入对安全的考虑,尤其要考虑解决困扰当前因特网可用性的 DDoS 攻击等威胁.目前,SDN 的研究集中在数据中心和园区网络,但是未来在广域网中进行部署也已经引起 ONF 的关注.如何在 SDN 增量部署的过程中,利用其中心化控制的优势保证未来互联网免受 DDoS 攻击困扰,具有重要的研究意义.据我们所了解,目前还没有相关的研究.

SDN 控制器是网络的控制和管理核心,其性能间接影响数据平面的通信性能,控制平面的扩展性是 SDN 大规模应用必须解决问题.目前,已有相关工作讨论 SDN 控制平面的扩展问题.Onix 通过引入分布式哈希表(distributed hash table,简称 DHT),在不同的控制器之间实现信息的共享.文献[23]第 1 次将云计算技术引入到控制器的分组处理和存储中,通过扩展东西向接口实现控制平面性能的扩展.文献[24]分析了未来 SDN、云计算与大数据技术之间的相互促进和影响,指出:SDN 灵活调度、全局控制的特性,能够为云计算和大数据在资源管控和调度上提供良好的平台;同时,云计算弹性的资源分配和集中式数据处理功能,为 SDN 扩展控制平面处理能力提供了条件,大数据技术为 SDN 应用的海量数据处理、数据挖掘提供支持,使得 SDN 控制平面的处理性能瓶颈得以环节.目前,SDN 在云计算安全方面已经有了很多应用[18,25,26],但是利用云计算技术实现对 SDN 控制平面扩展的研究还处于探索阶段,并已经引起了足够的重视.

1.1 DDoS攻击的必要条件

定义 1(网络(network)*N*). $N=\{H,SD,L,RM\}$,$H=\{h_1,h_2,h_3,\ldots\}$表示主机集合,交换设备集合 $SD=\{sd_1,sd_2,sd_3,\ldots\}$,$L=\{l_1,l_2,l_3,\ldots\}$表示链路集合,路由机制 *RM*(routing mechanism)具有为源主机 h_{src} 和目的主机 h_{dst} 确定通信路径 $p_i\langle h_{src\text{-}},h_{dst}\rangle$的功能:

$$connection(h_{src},h_{dst})\rightarrow p_i\langle h_{src\text{-}},h_{dst}\rangle=[h_{src},l_i^1,sd_i^1,l_i^2,sd_i^2,\ldots,h_{dst}],$$

其中,$l_i^1,l_i^2,\ldots\in L,sd_i^1,sd_i^2,\ldots\in SD$.

定义 2(傀儡机(dummy host)集合 *DH*). $DH=\{dh_1,dh_2,dh_3,\ldots\}$表示攻击者所捕获的具有安全漏洞的主机集合.$dh_i$ 可以按照指定的攻击速率 *r*,在某设定时刻 *t* 向目标主机 th_i 发起攻击 $attack(th_i,r,t)$.

定义 3(目标主机(target host)集合 *TH*). $TH=\{th_1,th_2,th_3,\ldots\}$为被 DDoS 攻击的目标主机集合.当 *DH* 被唤醒时,DDoS 攻击使 th_i 的计算和存储资源出现严重短缺,网络可用性严重下降.

定义 4(攻击者(attacker)*A*). DDoS 攻击组织者和发起者,通常具备 3 种基本功能:*scan*(*N*),扫描网络发现潜在脆弱主机集合 *H*′;*install*(*daemon*,*H*′),安装攻击守护程序形成傀儡机;*set_attack*(*DH*,*r*,*t*,*th*),向傀儡机发送攻击命令,其中,*r* 为攻击速率,*t* 为设定的攻击开始时间.

DDoS 攻击的一般过程可描述为:

1) 确定傀儡机集合.*A* 基于 *connection*(*A*,*N*)扫描目标网络,输出脆弱主机集合:$scan(N)\rightarrow H'$;*A* 向集合 *H*′中的主机安装攻击守护程序 *daemon*,形成傀儡机集合:$install(daemon,H')\rightarrow DH=\{dh_1,dh_2,dh_3,\ldots\}$;
2) 唤醒 *DH*.在时间区间$[t,t+\Delta t]$内,向 *DH* 中所有主机发送攻击特定目标主机 th_i 指令:

$$set\_attack(DH,r,t,th_i);$$

3) 发起攻击.*DH* 发起 *attack*(r,t,th_i),攻击时通常采用假冒源地址.

从 DDoS 攻击工作流程可以看到,形成 DDoS 攻击必须具备 3 个基本条件.

1) 连通性条件:a) *A* 发起 DDoS 攻击之前必须获取一定数量的脆弱主机,以形成僵尸网络(*botnet*)(假定这些主机操作系统存在安全漏洞),这要求 *A* 与脆弱主机之间必须存在通信链路 $p\langle A,DH\rangle$,致使脆弱主机成为 *dh*;b) *DH* 能够向 *th* 发起攻击,*dh* 与 *th* 之间必须存在通信链路 $p\langle DH,th\rangle$;
2) 隐蔽性条件:a) *A* 经过 *dh* 中转对 *th* 进行了间接攻击,隐藏自己;b) *dh* 采用假冒源地址来隐藏自己;c) *A* 发现并控制 *dh*,*A* 启动 *dh* 攻击和 *dh* 向 *th* 请求资源这 3 个过程在 TCP/IP 网络结构下没有任何记录;
3) 攻击性条件.DDoS 攻击通过 *DH* 向 *th* 发送大量资源请求以形成规模攻击效应.*th* 受害程度与 DDoS 攻击请求资源数量以及持续时间呈正相关.

如果能够限制上述条件中的至少一个,则 DDoS 攻击就难以在 *N* 中实施.

1) 如果不满足连通性,即 $connection(A,DH)\rightarrow p_i\langle A_{c\rightarrow},DH\rangle=\varnothing$,*A* 将无法完成 $scan(N)\rightarrow H'$,$install(daemon,H')$,*DH* 成员无法实施 $set_attack(DH,r,t,th_i)$和 $attack(r,t,th_i)$无法形成 DDoS 攻击;
2) 如果不满足隐蔽性,就容易定位 *A* 和 *DH*,使得网络管理者能够及时确定并屏蔽 *A* 与 *DH*,有效终止 DDoS 攻击;
3) 如果不满足攻击性,DDoS 无法有效消耗网络资源,即 $attack(\tilde{r},t,th_i)$($\tilde{r}$ 是一种受控的发送速率),形不成有效攻击力.

目前,因特网 DDoS 攻击易于实施而难以防范的特点,与 TCP/IP 体系结构的设计缺陷密切相关.

1) IP 网络中,任意两点之间默认连通的状态保证了 DDoS 攻击所需的连通性条件:首先,由于 IP 地址语义过载,一旦得知某主机的 IP 地址就知道其所在位置;其次,分组以无连接方式通信,路由器基于路由选择协议为到达的分组计算出通向目的地的路径.因此在通常状况下,IP 网络能够满足连通性条件,这为 *A* 的攻击行为提供了支持;
2) IP 网络转发行为的弱相关性使攻击者的行为具有隐蔽性,进而难以溯源.IP 网络的隐蔽性体现在:简单高效的体系结构,使得流接入和转发没有保留任何信息,因而流之间的关联性难以追溯,通常是通过人为地在网络中增加安全中间盒设备来增强流的相关性,从而改善弱溯源性;
3) IP 网络中,资源的统计复用特性使得大量的攻击者行为具有显著攻击性.通常,*botnet* 成员越多,它们发送的资源请求越多,消耗的网络资源份额就越大,攻击效果就越明显.

因此,只有设法弥补 TCP/IP 网络体系架构存在的缺陷,才能有效地从根源上解决 DDoS 攻击行为.

2 防御 DDoS 攻击的网络安全机制性质

本节从破坏或限制 DDoS 攻击的必要条件出发,探讨防御 DDoS 攻击的网络安全架构所应当具备的性质,为设计新型网络安全机制提供依据.

性质 1. 网络任意两点之间的默认状态应从连通变为不连通.

任何网络都要为通信过程提供连通性支持,因此连通性条件必须保持,但是要改变网络两点间状态默认为不连通的话,则易于在通信前增加访问控制机制,易于留下通信踪迹记录.尽管 DDoS 攻击是由一系列通信操作构成,但默认不连通易于破坏隐蔽性条件.本性质能够使通信过程变成一种面向连接的显式认证过程.

IP 网络不具备性质 1.如果要使新型机制具备该性质,就要解决如何扩展面向连接机制以及该连接机制与 IP 无连接机制之间的映射、互连互通等问题.

性质 2. 网络流状态应从弱相关变为强相关.

从破坏隐蔽性条件出发,增强网络流之间的相关性,才能使 DDoS 的隐蔽性条件无法成立,当通过溯源能够方便地还原一系列操作的轨迹时,就会使 DDoS 攻击者面对网络审计的威慑,认真考虑自己行为所导致的后果.

IP 网络本身无溯源能力,即使加上某些安全设备也只能做到弱溯源[27,28].在具有性质 1 的网络中,易于增加流准入控制和流日志功能,从而使网络具备性质 2.在大规模网络中,只要该网络安全机制解决好海量流信息的

存储和查找问题,该机制就有了扩展性.

性质 3. 网络资源分配机制应从平等竞争变为预约监管.

IP 网络的分组统计复用特性,使得发送分组越多则占用网络资源就越多,这为 DDoS 攻击恶意消耗网络资源提供了便利.从限制攻击性角度出发,如果网络安全架构能够限制和监管资源的无度使用,就能遏制 DDoS 攻击性因素.

要具备性质 3,在流接入控制时,应当为流分配资源使用额度并进行实施监管,防止滥用网络资源的情况.

性质 4. 应当继承因特网现有的软硬件资源.

因特网技术的任何改进,都应当以增量部署的方式进行,能够继承因特网庞大基础设施和软件应用,否则将可能被束之高阁.设计这种安全网络机制极具挑战性.

3 防御 DDoS 攻击的软件定义安全网络机制设计

3.1 软件定义的安全网络机制SDSNM

基于上述分析,本文提出了一种能够防范 DDoS 攻击的软件定义网络安全机制 SDSNM,图 1 描述了该机制逻辑框架.该机制设计要点包括:

1) 保持 IP 核心网络体系结构和端系统操作系统不变,使安全网络机制具有性质 4;
2) 在 IP 网络的边缘部分采用软件定义网络(software defined networking,简称 SDN)[29]技术(如 OpenFlow[30])完成下列功能:(1) 任何两点间的通信都要经过控制器认可,使性质 1 成立;(2) 记录通信过程并能根据要求进行查询分析,使性质 2 成立;(3) 对每条流采用流桶机制控制其使用的资源,使性质 3 成立;
3) 在控制平面设置 Chord[31]环,以协同不同的控制器,使 SDSNM 能够用于广域网环境中;
4) 开发应用程序,提供如用户注册、接入控制、流信息存储和回溯等功能;设置云平台[32],提供海量数据存储和查询等功能.

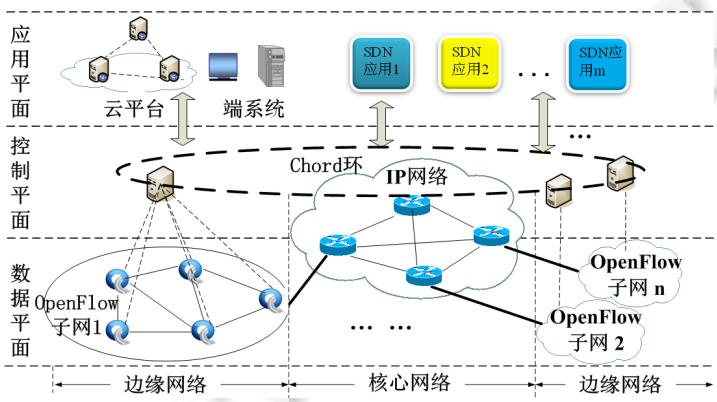


Fig.1 Logic framework of SDSNM

图 1 防范 DDoS 攻击的 SDSNM 逻辑架构

SDSNM 覆盖核心网与边缘网两个部分:核心网由 IPv4 路由器和通信链路等组成;边缘网由众多 SDN 子网组成,每个 SDN 子网由 OpenFlow 交换机连接的控制器、用户主机和服务器等组成.Chord 环扩展控制平面东西向接口,运行在 IP 网络之上,能够向各子网控制器提供统一的网络状态、控制智能和控制策略.控制器可将流信息存储到云平台中,亦可请求云平台提供流信息查询等服务.

网络应用(也包括 DDoS 攻击)的流量均来自边缘网,而在 SDSNM 机制中,一个流通常是由 3 个有序段组成:源边缘网的 OpenFlow 流段、核心网的 IP 流段和目的边缘网的 OpenFlow 流段.尽管 IP 流段是无连接的,但边

缘网络的 OpenFlow 流是面向连接的,且完全受控,这就为满足性质 1~性质 3 奠定了基础.

SDSNM 中任意两台位于不同的 OpenFlow 子网中的主机 h_{src} 与 h_{dst} 的通信过程如图 2 所示.假定 h_{src} 和 h_{dst} 均具有全局性 IP 地址.

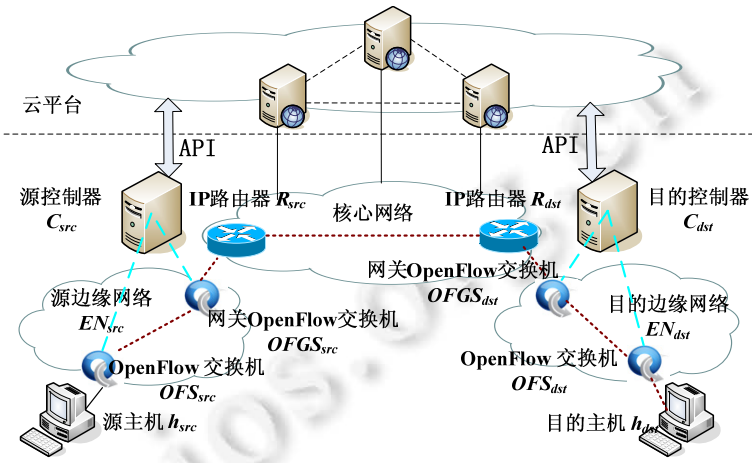


Fig.2 Communication process between two hosts in different edge network

图 2 两台位于不同边缘网络中主机的通信场景

1) 来自主机 h_{src} 的新流 $flow$=($smac$,$dmac$,sip,dip,sp,dp)由交换机 OFS_{src} 端口 s_port 进入子网 EN_{src}.设 OFS_{src} 的标识为 s_id,$smac$/sip/sp 分别表示 h_{src} 的 MAC/IP 地址以及源端口,$dmac$/dip/dp 分别表示目的主机 h_{dst} 的 MAC/IP 和目的端口.由于任何主机在接入网络之前需要到 Chord 注册其 MAC/IP 地址、接入交换设备 ID 和接入端口,因此,$dmac$/dip 可以通过 Chord 进行查询获得.此时,OFS_{src} 没有匹配项,则发送 PACKET_IN 报文至控制器 C_{src};

2) C_{src} 执行第 3.3 节中的算法 1“SDSNM-accessControl”以查验该流的合法性:如果合法,则继续执行步骤 3);否则,C_{src} 拒绝其接入,并将被拒绝流的相关信息写入到云平台中;

3) C_{src} 在 EN_{src} 中建立 OFS_{src} 到网关交换机 $OFGS_{src}$ 的路径,将转发路径流表项下发至相应的交换设备流表;C_{src} 将流接入信息写入云平台;

4) 流分组到达 EN_{src} 网关路由器 R_{src},核心网络利用传统的 IP 路由选择协议,将该流分组送达主机 h_{dst} 所在子网网关路由器 R_{dst},并由它转发至目的边缘网络 EN_{dst} 网关交换机 $OFGS_{dst}$;

5) $OFGS_{dst}$ 在转发表中没有查询到匹配流表项,则将发送 PACKET_IN 报文发送至 C_{dst},若 C_{dst} 查询判定流分组合法后,建立从 $OFGS_{dst}$ 到 OFS_{dst} 的路径,将相关流表项下发至交换设备流表.至此,后继流分组就可以沿此路径从 h_{src} 到达 h_{dst};

6) 若 C_{src} 或 C_{dst} 直接或间接感知可能存在 DDoS 攻击,即可调用第 3.3 节中的算法 2“SDSNM-traceback”分析相关流,以确定攻击源.

由此可见,SDSNM 具有以下特点:

(1) 强制接入控制.任何通信都包含两侧 OpenFlow 子网中面向连接、需要认证的过程,这既保证了端到端通信的连通性,也使通信增加了无法绕开的强制安全措施.由于通过注册,控制器知道来自各端口分组的标识,无法实施 IP 地址等信息欺骗;

(2) 限制攻击性条件.两侧 OpenFlow 子网提供的流量控制机制,可以限制 DDoS 的攻击性条件;

(3) 破坏隐蔽性条件.基于云计算的对接入控制过程的日志功能,能够为破坏 DDoS 攻击的隐蔽性条件带来便利;

(4) 具有良好的继承性和扩展性.接入控制、流信息审计(存储、查询和统计)等复杂功能位于网络边缘,

减少了对核心网络的干预,可以完全继承现有核心网络基础设施.采用云计算等技术可以实现对海量信息的处理和存储,使系统具有很好的扩展性.

3.2 关键功能模块设计

如图 3 所示,SDSNM 的功能模块主要位于应用平面:一部分在控制器中运行,另一部分在云计算平台中运行.Chord 环位于控制平面,用以屏蔽应用平面功能模块的局域性,使功能模块及其相关策略能为 SDSNM 共享.各边缘网络底层的物理设备主要由通信主机和 OpenFlow 交换机构成,SDSNM 利用标准的南向接口(OpenFlow 协议)获取底层网络的拓扑、硬件资源状态等信息,同时可以控制各底层网络设备行为.

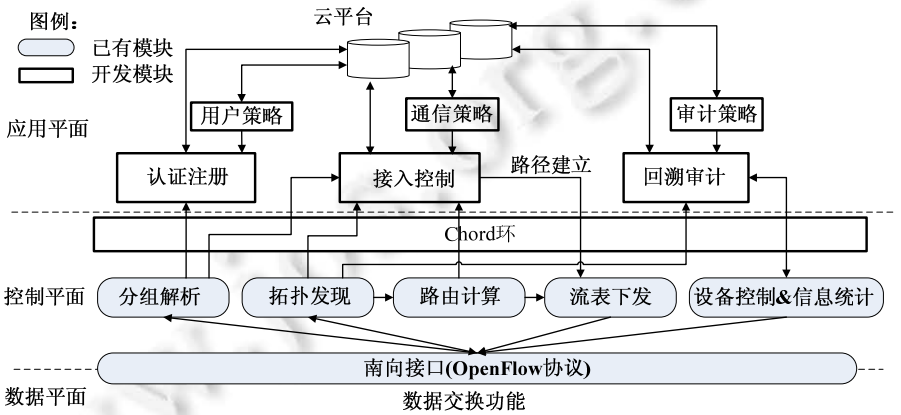


Fig.3 Modules of SDSNM

图 3 SDSNM 基本功能模块

SDSNM 主要功能模块包括认证注册、接入控制和回溯审计及其各自对应的策略管理模块,它们均位于应用平面.

- 认证注册及策略管理模块

SDSNM 要求通信实体首次通信前必须先注册,认证注册模块可以通过某种实名制措施来确认通信实体身份[17,33].通过认证的合法通信实体必须将其 ID、所使用的 IP 地址、所接入的交换设备 ID、接入端口号和所关联的控制器以及能够使用的服务等信息存入应用平面,并将认证注册信息写入云平台.用户策略模块则为整个系统提供定制的认证注册相关规则,如注册实体的认证方式,注册实体能够通信的对象、能够使用的应用类型和服务质量等.在注册认证成功后实体将得到一个接入口令,用于后继接入控制.

- 接入控制及通信策略模块

SDSNM 要求通信实体每次进行网络通信时必须经过接入控制.接入控制模块对于 ID 和接入口令正确的实体放行通信.该模块调用底层路由模块计算通信路径,并向相关交换机下发流表.该模块能够周期性地将流相关信息写入云平台.第 3.3 节中的 SDSNM-accessControl 算法描述了该模块的功能.通信策略模块则为整个系统提供定制的接入控制相关规则,例如何时在何种情况下允许该类实体接入、接入路径的限制、通信对象的限制和接入口令的更换条件等.接入控制是所有安全通信的第 1 步,无论是在传统因特网还是在 SDSNM 机制中,都是保证系统安全运行的基础,系统采用宽松的接入控制,能够减少端系统认证开销,加速通信进程,但是相应地增加了网络检测和策略定制的开销,要求网络本身设计更加复杂的安全审计和回溯机制;严格的接入控制能够相应地减少网络中安全威胁的进入,从通信第 1 步就对安全因素进行了过滤,但是严格接入控制也降低了用户的体验.目前,基于 SDN 进行接入控制的研究正在开展,文献[34,35]所讨论的基于 SDN 所改进的动态接入控制机制可以在 SDSNM 中进行部署应用.接入控制及通信策略定制模块功能的实现是开放性问题,如何设计一种高效、分级且支持移动性的接入控制机制[36],将在系统下一步研究过程中进行深入研究.

- 回溯审计及审计策略模块

动态灵活的全局控制策略定制以及能够获取网络全局资源信息视图的能力,是 SDN 区别于传统网络的优势.在宽松的接入控制条件下,SDSNM 允许系统以主动检测方式或者被动触发方式对 DDoS 攻击进行检测.由于 SDSNM 继承了 SDN 的工作模式,因此我们可以利用 SDN 能够在流建立阶段获取分组信息的能力进行主动 DDoS 检测,如文献[16,18]所设计的机制都可以在本系统中进行应用.被动的触发方式则由端系统直接向控制器报告攻击的发生,例如文献[7]所应用的方案.回溯审计模块能够利用底层设备控制接口实时监控网络底层设备资源使用信息,主动检测并发现 DDoS 攻击或得到某实体受到 DDoS 攻击的报告;然后,调用云平台的分析功能确定攻击源.该模块提供限制可疑接入设备端口或者将攻击流量重定向到安全中间设备进行深度检测的功能,达到防范恶意攻击的目的.第 3.3 节中的 SDSNM-traceback 算法描述了该模块的功能.审计策略模块为整个系统提供定制的回溯审计相关规则,如审计的方式、审计的参数和审计的时段等.

云平台可采用现有成熟、廉价的云计算技术[32,37,38]如 Hadoop,为支持 SDSNM 提供存储注册实体、各类策略等静态或半静态信息,存储海量的流注册、接入和流通信信息,提供快速的信息插入、删除以及 DDoS 审计查询、溯源操作等功能.尽管 Chord 环不是 SDSNM 的组成部分,但 Chord 环中存储了通信中的实体 ID、当前位置、相互协作关系等信息,并以高效数据结构允许实体的加入、退出或修改信息,以及帮助这些实体之间高效、迅速地定位.

3.3 关键算法描述

SDSNM 有两种关键算法:一是用户接入控制算法,另一个是攻击回溯和限制算法.下面进行具体描述.

(1) 接入控制算法

接入控制模块基于系统中的认证注册信息,将 IP 原本默认连通的通信模式改造为默认需要认证的通信模式,在保证连通性条件的同时,破坏了 DDoS 的隐蔽性条件和攻击性条件.

首先,控制器通过流建立请求 PACKET_IN 报文,通过 *extract*(·)解析流相关信息(如 *s_id*,*s_port*,*sip*,*dip*,*smac*,*dmac*,*dp*,*sp*);其次,通过这些字段利用 *match_with_policy*(·)查询匹配系统中用户注册信息、通信策略,来判定流的合法性,基于网络拓扑信息在相关交换机上为正常通信流建立路径,并在云平台上登记流信息.

以下给出了 SDSNM 接入控制算法.

算法 1. SDSNM-accessControl.

Input: *PACKET_IN message*;

Output: *path* //流路径.

```
1:  timestamp←time(·)
2:  s_id←PACKET_IN[s_id]
3:  s_port←PACKET_IN[port]
4:  flow[smac,dmac,sip,dip,sp,dp]←extract(PACKET_IN)
5:  result←match_with_policy(flow)
6:  if result is False:
7:      drop(s_id,s_port,flow)
8:      insert_to_log(timestamp,flow)
9:      return NULL
10: else:
11:     d_id, d_port←get_location(dip)
12:     path←compute_path(s_id,s_port,d_id,d_port,topo)
13:     insert_to_log(timestamp,flow)
14:     return path
```

IP 地址与主机位置的映射关系在主机注册时绑定并存储在 Chord 中.*compute_path*(·)利用源主机位置、目

的主机位置以及网络拓扑信息计算通信路径.*d_id* 和 *d_port* 分别表示目标主机的接入交换设备的标识和接入端口.该算法主要的时间开销来自匹配规则,其时间复杂度为 $O(n)$,n 为策略的数量.

(2) 攻击回溯算法

由于所有流的接入、摘要信息均存储于云平中台,一旦检测到 DDoS 攻击,云平台可以调用攻击回溯算法来确定 *botnet* 成员集合和攻击者.

图 4 给出了 DDoS 攻击的时序图.假定从 t_0 时刻开始,在第 1 阶段$[t_0,t_1]$,攻击者 *A* 主动与 *botnet* 成员通信,向 *botnet* 成员下达攻击指令;在第 2 阶段$[t_1,t_{detect}]$,*botnet* 成员向 *th* 大量发送报文.假定在 t_{detect} 时刻,检测到 DDoS 攻击并调用攻击回溯算法.

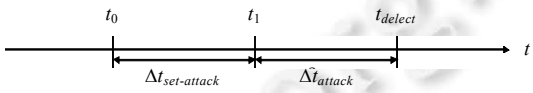


Fig.4 Time sequence diagram of DDoS

图 4 DDoS 攻击的时序图

攻击回溯过程可视为上述过程的逆过程.

假设在时刻 t_{detect} 检测到攻击,回溯算法通过公式(1)查找可能的僵尸主机集合 *DHSet*:

$$DHSet=\{\log_i[sip]|\log_i[dip]==th\_ip \text{ and } flow\_volume_i>\Sigma,\forall \log_i\in \log Set_{attack},i=1,2,\ldots|\log Set_{attack}|\} \quad (1)$$

其中,$\log Set_{attack}$ 是$[t_{detect}\tilde{\Delta}t_{attack},t_{detect}]$内所有流记录的集合,对于任意的 $\log_i\in\log Set_{attack}$,$\log_i[sip]$表示 $\log_i$ 源 IP 地址,$\log_i[dip]$表示 $\log_i$ 目的地址.*th_ip* 表示 *th* 的 IP 地址,$flow\_volume_i$ 表示从 $\log_i[sip]$到 $\log_i[dip]$单向流的流量.如果 $flow\_volume_i$ 超过了预设的流量阈值Σ,同时,其目的地址为 *th*,都应该被加入到 *DHSet*.这里,目标主机的 IP 地址由被攻击主机主动向控制器上报.阈值Σ可以通过统计的方式进行优化设定[28,30].

假设在 *DHSet* 中存在 *m* 个僵尸主机,同时假设 *DHSet* 的大小为 *n*.设正常通信强度下,任意两点间通信的概率为 *p*,因此,若$[t_{detect}\tilde{\Delta}t_{attack}\tilde{\Delta}t_{set_attack},t_{detect}]$内没有发生 DDoS 攻击,则平均通信次数为 *np*;如果发生了 DDoS 攻击,由于所有的僵尸主机必然与攻击主机进行了至少一次通信,因此平均的通信次数为 $m+(n-m)p$,二者之差为 $(1-p)m$.由于 $p<<1$,$m\approx n$,因此在$[t_{detect}\tilde{\Delta}t_{attack}\tilde{\Delta}t_{set_attack},t_{detect}]$的流记录中,与 *DHSet* 通信次数最多的主机最有可能是攻击者.通过 *DHSet* 和$[t_{detect}\tilde{\Delta}t_{attack}\tilde{\Delta}t_{set_attack},t_{detect}]$内的流记录 $\log Set_{set\_attack}$,按照公式(2)可以确定真正的 DDoS 攻击者:

$$\begin{cases} f(i,j)=\begin{cases}1, \text{ if } \log_j[dip]\in DHSet \text{ and } \log_j[sip]==\log_i[sip]\\ 0, \text{ other}\end{cases}\\ count[\log_i[sip]]=\sum_{j=1}^{|\log Set_{set\_attack}|} f(i,j),\log_i\in\log Set_{set\_attack}\\ AttackerSet=\{\log_i[sip]\,|\,count[\log_i[sip]]=\max\}\end{cases} \quad (2)$$

以下给出了 SDSNM 攻击回溯算法.

算法 2. SDSNM-traceback.

Input: t_{detect}, Δt_{attack}, Σ, *th_ip*;

Output: *DHSet*, *AttackerSet*.

1: $\log Set_{attack}\leftarrow get\_\log(t_{detect}\tilde{\Delta}t_{attack}\tilde{\Delta}t_{set\_attack},t_{detect},th\_ip)$

2: **for** i=1, $i\Leftarrow|\log Set_{attack}|$:

3: **if** $\log_i[dip]==th\_ip$:

4: $flow\_volume_i\leftarrow statistics(\log_i[sip],\log_i[dip])$

5: **if** $\log_i[sip]\notin DHSet$ **and** $flow\_volume_i\geqslant\Sigma$:
6: $DHSet\leftarrow add(\log_i[sip])$
7: $AttackerSet\leftarrow\varnothing$, i=1, $count\leftarrow\varnothing$, $ip\_set\leftarrow\varnothing$
8: **while** $Attacker$ **is** $\varnothing$:
9: $\Delta t_{set\_attack}=(2^{i-1})\times\Delta t_{attack}$
10: $\log Set_{set\_attack}\leftarrow get\_\log(t_{detect}\tilde{\Delta}t_{attack}\tilde{\Delta}t_{set\_attack})$
11: **for** j=1, $j\Leftarrow|\log Set_{set\_attack}|$
12: **if** $\log_i[sip]\notin ip\_set$:
13: $ip\_set\leftarrow add(\log_i[sip])$
14: **for** k=1, $k\leqslant|\log Set_{set\_attack}|$:
15: **if** $\log_k[sip]==\log_j[sip]$ **and** $\log_k[dip]\in DHSet$:
16: $count[\log_j[sip]]$++
17: **for** $ip\in ip\_set$:
18: **if** $count[ip]$ **is the maximum**:
19: $AttackerSet\leftarrow ip$
20: **return** $AttackerSet$, $DHSet$

关于Δt_{attack} 的取值,目前已经有成熟的技术[39−41]进行测量和估算,本文不再详述.取$\Delta t_{set_attack}=2^i\Delta t_{attack}$($i$=0, 1,…),其中,$i$ 为取值的次数,如果确定攻击者,则算法终止;否则,继续调整Δt_{set_attack}.

该算法的时间复杂度为 $O(n)$,其中,n 为流记录信息数目.

4 原型系统实验

4.1 实验环境

为了验证 SDSNM 的可行性,并对其性能进行评估,我们在实验室环境下构建了如图 5 所示的原型系统.该原型系统由 4 个 OpenFlow 子网和 1 个核心 IP 网络组成,每个 OpenFlow 子网包括多台用户主机、2 台 OpenFlow 交换机[30](由具有 5 端口千兆以太网卡的 Linux PC 运行 Stanford OpenFlow 1.0.0 软件实现)、1 台 POX[42]控制器和 1 台 Open Chord[43]结点.

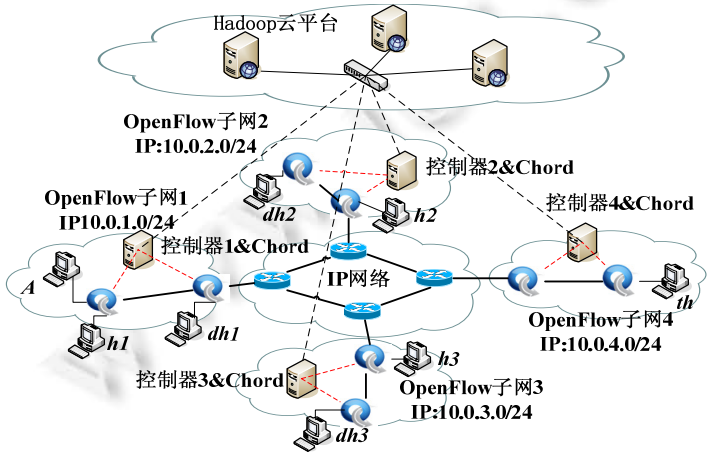


Fig.5 Prototype of SDSNM

图 5 SDSNM 原型系统

tfn2k[44]是典型的 DDoS 工具,可以实现多种 DDoS 攻击,实验将其作为 DDoS 攻击工具.为使评价更为客观,实验增加了背景流量(基于 Lincoln 实验室 DARPA2000[45]数据集),通过 tcpreplay[46]对记录进行了修改并重放.实验的其他设置包括:

1) 在 OpenFlow 子网 1 中设置并运行 tfn2k 的攻击主机 A,同时,在 OpenFlow 子网 1~子网 3 中事先人为设置傀儡机 dh_1,dh_2 和 dh_3,并分别植入了守护进程(这使攻击更易进行);一旦这些傀儡机接收到 A 的指令,便向目标主机 th 发起攻击;
2) 在 h_1~h_4 主机上,使用 tcpreplay 以指定的速率向 th 发送背景流量,其符合均值为 1000packets/s 的均匀分布;
3) 各控制器从云平台 Hadoop 下载 SDSNM 相关功能模块和策略,并通过 Chord 环共享网络状态信息和主机注册信息;
4) 在目标主机 th 运行抓包程序俘获分组并进行分析.

验证 SDSNM 基本功能,本节设计了两组实验:一组通信策略基于“白名单”,一组基于“黑名单”.

4.2 “白名单”实验

该实验采用了最严格的通信策略,即,不允许除“白名单”外所有实体的接入及通信,其目的是检验 SDSNM 是否能够防止 DDoS 攻击的发生.

- 实验 1 过程

(1) 从 0s~1080 s 运行背景流量;
(2) 360s~720s 攻击主机发起 UDP Flood DDoS 攻击持续 360s;
(3) 设置正常网络与目标主机会话速率为 200 次/s,会话基于 UDP,一次会话平均为 5 个 UDP 报文;
(4) DDoS UDP Flood 会话速率为 2000 次/s,一次会话长度为 1 个 UDP 报文.

- 实验结果及分析

由图 6 给出的统计结果可见:

- 在没有采用 SDSNM 情况下,在 0s~360s 期间,th 接收分组约为 1000packets/s;而在 360s 后,急剧增加至 2950packets/s,并持续到 720s;
- 采用了 SDSNM 接入控制算法加上“白名单”策略,在整个通信过程中,th 接收分组均保持在 1000packets/s 左右.

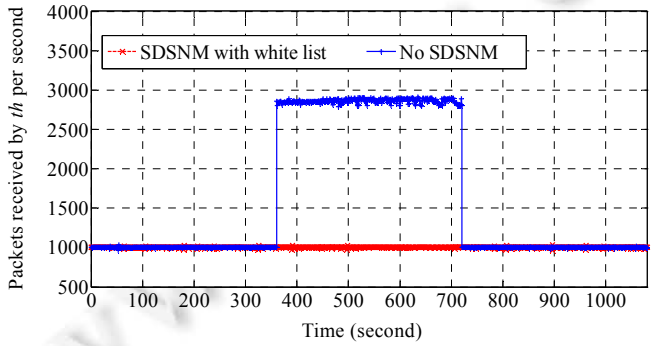


Fig.6 Packets received by th per second

图 6 th 接收分组的数量

实验过程中,通过在控制器上统计单位时间 PACKET_IN 事件数目发现:在缺少 SDSNM 的情况下,控制器和交换机会受到 DDoS 攻击.因为 UDP 攻击报文频繁地变换源地址和端口号,OpenFlow 无法匹配新的 UDP 报文,会向控制器转发大量 PACKET_IN 报文.采用了 SDSNM 后,由于网络中主机通信都必须通过认证注册保证真实性,同时,“白名单”通信策略只允许认证注册的主机通信,并限制异常通信端口,因此,SDSNM 机制同时可以

保护控制器和交换机免受 DDoS 攻击.如图 7 所示:采用 SDSNM 机制并对异常通信端口采取丢包策略后,交换机流表匹配成功/查询次数比值在 0s~1 080s 内始终保持较高的比率.

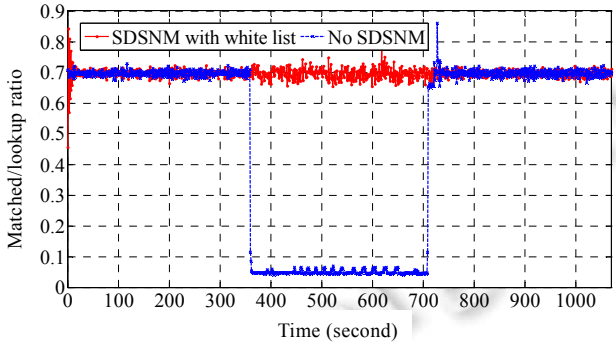


Fig.7 Packet matching ratio of *th* access switch

图 7 *th* 接入交换机流表匹配成功次数与查找次数比值

SDSNM“白名单”机制由接入控制算法实现.实验结果表明,通信规则的数量和存储方式会影响通信性能.图 8 显示了如果将通信规则存储在 Hadoop 云平台,通信规则的不同量级对流建立时延几乎没有影响,有良好的扩展性;而将通信规则存储在普通文件系统中时,扩展性则较差.

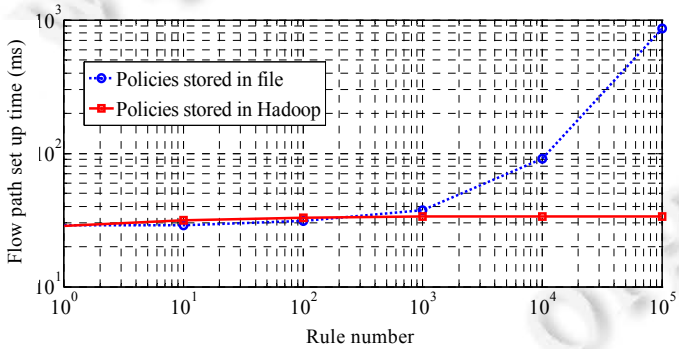


Fig.8 Data path building time in different policy number and storage ways

图 8 通信规则的数量和存储方式对流路径建立时延的影响

- 实验小结

采用严格的接入控制策略,SDSNM 具有理想的防范 DDoS 攻击的效果,即使在事先人为设置 *botnet* 的情况下,也不可能发生 DDoS 攻击;结合通信策略,可以防止对控制器和相关交换机的 DDoS 攻击,用 Hadoop 处理通信规则系统具有良好的扩展性.

4.3 “黑名单”实验

该实验采用了宽松的通信策略,即,允许除了“黑名单”外的所有实体接入及通信.其目的是检验 SDSNM 机制在 DDoS 攻击已经发生的情况下,通过攻击回溯算法确定攻击源的有效性.

- 实验 2 过程

(1) t_0=0s~371s 网络为正常通信;

(2) 在 t_1=371s 时刻,*A* 向 *botnet* 主机下达攻击指令;

(3) 在 t_2=372s 时刻(Δt_{set_attack}=1s),预设的 *botnet* 开始对目标主机进行 UDP Flood DDoS 攻击并持续 20s;

(4) 在 t_2 后的某个时刻 t_{detect}=375s 触发检测(Δt_{attack}=3s),调用攻击回溯算法来确定攻击源;

(5) 设置 60 台虚拟端主机,其中 1 台 *th*、1 台 *A* 和 50 台僵尸主机;

(6) 所有背景会话都基于 UDP,其报文长度为 60Bytes,正常情况下,主机间一次会话的平均速率为 $flow\_rate_{normal}$=300packets/s,DDoS 攻击主机的平均攻击速率为 $flow\_rate_{attack}$=1000packets/s,设置流量阈值Σ=0.5×$flow\_rate_{attack}$.

- 实验结果及分析

在实验 1 中,假定攻击回溯算法的 3 个重要参数 t_{detect},Δt_{attack} 和Δt_{set_attack} 均为已知,在 t_{detect}=375s 时刻,控制器通过回溯算法分析通信日志,回溯攻击主机.实验过程中,根据网络背景会话速率,测试了:(1) 确定疑似 *botnet* 主机集合 *DHSet* 的时间 t_{dh_Set};(2) 确定攻击者候选集合 *AttackerSet* 的时间 t_{attack_Set}.多次实验结果表明,SDSNM 能够准确找到所有 50 台傀儡机和攻击主机 *A*.t_{dh_Set} 和 t_{attack_Set} 如图 9 所示.

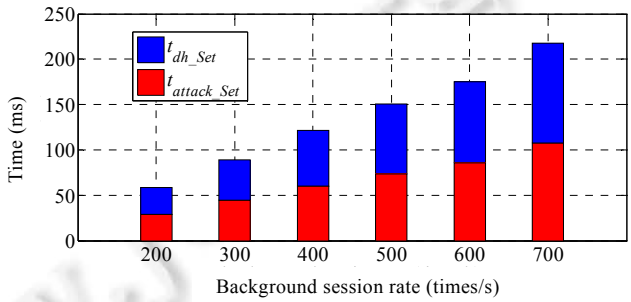


Fig.9 t_{dh_Set} and t_{attack_Set} in different background session rate

图 9 不同网络会话话速率下 t_{dh_Set} 和 t_{attack_Set} 测量值

从图 9 可以看到:随着网络背景会话速率的增加,云平台回溯出 *botnet* 成员和攻击者的时间呈线性增加.这是因为网络背景会话速率越大,在Δt_{attack} 或Δt_{set_attack} 时段内,插入云平台的流记录数量就越多,回溯算法所需查找的记录数量也就越多,导致 t_{dh_Set} 和 t_{attack_Set} 相应增加.

- 实验 3 过程

在实际工作中,我们可以得知 $t_{detect\neg}$时刻,但无法知道Δt_{attack} 和Δt_{set_attack},故本实验研究如何确定Δt_{attack} 和 Δt_{set_attack} 大致的取值范围.本实验基于实验 2,故 $t_{detect\neg}$=375s.根据经验数据,让Δt_{attack} 取一个定值 1s.即:利用 $t_{detect\neg}$前面 1s 期间的记录,基于条件 1 确定 *botnet* 中成员.Δt_{attack} 为 1s 的依据是:当 DDoS 攻击发生后,该区间记录的信息最全、最新.尽管无法确保能够分析出所有 *botnet* 中成员,但足以追溯出大部分 *botnet* 成员.

实验结果见表 1:实验中,多次扩大Δt_{attack} 范围,对于确定 *botnet* 成员有少量影响,而对于确定攻击者几乎没有影响;多次实验Δt_{set_attack},都在首次赋值就准确确定攻击者.

Table 1 Impact of Δt_{attack} to the size of *DHSet* and the iteration of Δt_{set_attack}

表 1 Δt_{attack} 对 *DHSet* 和Δt_{set_attack} 迭代次数 *i* 的影响

	1s	1.5s	2s	2.5s	3s	3.5s	4s	4.5s	5s	5.5s
|*DHSet*|	16	25	28	32	35	36	42	45	47	48
i 取值	1	1	1	1	1	1	1	1	1	1

- 实验小结

在宽松通信规则的情况下,攻击回溯算法能够快速准确地定位攻击者和 *botnet* 成员.实验结果显示:设置适当的Δt_{attack}(如 1s)和Δt_{set_attack} (如 $2^i\Delta t_{attack}$)时间,将用 50ms~200ms 左右的时间(与背景流量大小有关),迅速地确定 *botnet* 成员和攻击者.

5 总 结

尽管已经进行了大量的研究,目前仍然无法有效防范 DDoS 攻击.本文指出了 IP 网络体系结构的设计缺陷,

是导致这种现象的根本原因.本文建模推导了DDoS攻击的连通性、隐蔽性和攻击性必要条件,再从破坏或限制这些必要条件的角度,推演出能够对抗DDoS攻击的软件定义安全网络机制SDSNM应当具有的性质,进而构建了一种SDSNM的原型系统.实验和分析表明:这种新型安全网络体系结构在严格接入情况下使DDoS攻击无法进行;在宽松接入情况下,能够迅速确定 *botnet* 成员和攻击者.快速高效的DDoS攻击检测机制,是加速SDSNM进行DDoS攻击回溯的基础.SDSNM的DDoS检测机制是一个开放性的问题,目前,基于控制器与基于端系统的检测方法都能够在SDSNM中进行部署,但是SDSNM更加精细的接入控制策略以及DDoS攻击检测机制将是下一步研究的重点.新技术的引入不可避免地带来新的安全隐患,SDN 和云平台自身的安全性保护也是未来SDSNM在实际网络环境下进行部署必须进一步研究的问题.

References:

[1] Wang A, Mohaisen A, Chang W, Chen SQ. Delving into Internet DDoS attacks by botnets: characterization and analysis. In: Proc. of the 45th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks (DSN). Rio de Janeiro, 2015. 379−390. [doi: 10.1109/DSN.2015.47]

[2] Popovskyy V, Skibin V. Entropy methods for DDoS attacks detection in telecommunication systems. In: Proc. of the 1st Int'l Scientific-Practical Conf. on Problems of Infocommunication Science and Technology. Kharkov, 2014. 182−185. [doi: 10.1109/INFOCOMMST.2014.6992345]

[3] 2013-2014 DDoS threat landscape report. http://lp.incapsula.com/rs/incapsulainc/images/2013-14_ddos_threat_landscape.pdf

[4] Yau DKY, Lui JCS, Liang F, Yam Y. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. IEEE/ACM Trans. on Networking, 2005,13(1):39−42. [doi: 10.1109/TNET.2004.842221]

[5] Leiner BM, Cerf VG, Clark DD, Kahn RE, Kleinrock L, Lynch DC, Postel J, Roberts LG, Wolff S. A brief history of the Internet. ACM Sigcomm Computer Communication Review, 2009, 39(5):22−31. [doi: 10.1145/1629607.1629613]

[6] Blumenthal MS, Clark DD. Rethinking the design of the Internet: The end-to-end argument vs. the brave new world. ACM Trans. on Internet Technology, 2001,1(1):70−109. [doi: 10.1145/383034.383037]

[7] Zhang YZ, Xiao J, Yun XC, Wang FY. DDoS attacks detection and control mechanisms. Ruan Jian Xue Bao/Journal of Software, 2012,23(8):2258−2072 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4237.htm [doi: 10.3724/SP.J.1001.2012.04237]

[8] Xie GG, Zhang YJ, Li ZY, Sun Y, Xie YK, Li ZC, Liu YJ. A survey on future Internet architecture. Chinese Journal of Computers, 2012,35(6):1109−1119 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2012.0l109]

[9] Koponen T, Shenker S, Balakrishnan H, Feamster N, Rexford J, Arianfar S, Kuptsov D. Architecting for innovation. ACM Sigcomm Computer Communication Review, 2011,41(3):24−36. [doi: 10.1145/2002250.2002256]

[10] Walfish M, Balakrishnan H, Shenker S. Untangling the Web from DNS. In: Proc. of the 1st Conf. on Symp. on Networked Systems Design and Implementation. 2004. 225−238.

[11] Wendlandt D, Avramopoulos I, Andersen D, Rexford J. Don't secure routing protocols, secure data delivery. In: Proc. of the 5th ACM Workshop on Hot Topics in Networks. 2006.

[12] Andersen DG, Balakrishnan H, Feamster N, Koponen T, Moon D, Shenker S. Accountable internet protocol (AIP). ACM Sigcomm Computer Communication Review, 2008,38(4):339−350. [doi: 10.1145/1402946.1402997]

[13] Naylor D, Mukerjee MK, Steenkiste P. Balancing accountability and privacy in the network. ACM Sigcomm Computer Communication Review, 2014,44(4):75−86. [doi: 10.1145/2740070.2626306]

[14] Wu JP, Bi J, Li X, Ren G, Xu K, Williams M. A source address validation architecture (SAVA) testbed and deployment experience. RFC 5210, 2008.

[15] Wu JP, Cui Y, Li X, Xu M, Metz C. 4over6 transit solution using IP encapsulation and MP-BGP extensions. RFC 5747, 2010.

[16] Lim S, Ha J, Kim H, Kim Y, Yang S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In: Proc. of the 6th Int'l Conf. on Ubiquitous and Future Networks. 2014. 63−68. [doi: 10.1109/ICUFN.2014.6876752]

[17] Yao G, Bi J, Xiao PY. Source addresses validation solution with OpenFlow/Nox architecture. In: Proc. of the 19th IEEE Int'l Conf. on Network Protocols (ICNP). Vancouver: IEEE Press, 2011. 7−12. [doi: 10.1109/ICNP.2011.6089085]

[18] Wang B, Zheng Y, Lou WJ. DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks, 2015,81:308−319. [doi: 10.1016/j.comnet.2015.02.026]

[19] Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou J, Zhu M, Zolla J. B4: Experience with a globally-deployed software define WAN. ACM Sigcomm Computer Communication Review, 2013,43(4):3−14. [doi: 10.1145/2534169.2486019]

[20] Koponen T, Casado M, Gude N, Stribling J, Poutievski L, Zhu M, Ramanathan R, Iwata Y, Inove H, Hama T. Shenker S. Onix: A distributed control platform for large-scale production networks. In: Proc. of the 9th USENIX Conf. on Operating System Design and Implementation (OSDI). Vancouver: USENIX, 2010. 351−364.

[21] Internet 2. 2012. http://www.internet2.edu

[22] CNGI. http://www.cngi.cn/

[23] Handigol N, Heller B, Jeyakumar V, Mazieres D, McKeown N. I know what your packet did last hop: Using packet histories to troubleshoot networks. In: Proc. of the 11th USENIX Conf. on Networked System Design and Implementation. 2014. 71−85.

[24] Qadir J, Ahad N, Mushtaq E, Bila M. SDNs, clouds and big data: New opportunities. In: Proc. of the 12th IEEE Int'l Conf. on Frontiers of Information Technology (FIT). 2014. 28−33. [doi: 10.1109/FIT.2014.14]

[25] Wanderson PJ, Daniel AS, Rafael T, Sousa J. Analysis of SDN contributions for cloud computing security. In: Proc. of the IEEE/ACM 7th Int'l Conf. on Utility and Cloud Computing. 2014. 922−927. [doi: 10.1109/UCC.2014.150]

[26] Yan Q, Yu FR. Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Communication Magazine, 2015,53(4):52−59. [doi: 10.1109/MCOM.2015.7081075]

[27] Burch H, Cheswick B. Tracing anonymous packets to their approximate source. In: Proc. of the 14th USENIX Conf. on System Administration. 2000. 878−886.

[28] Song DX, Perrig A. Advanced and authenticated marking schemes for IP traceback. In: Proc. of the IEEE INFOCOM. Anchorage, 2001. 878−886. [doi: 10.1109/INFCOM.2001.916279]

[29] Open networking foundation (ONF). 2012. http://www.opennetworking.org

[30] Mckeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: Enabling innovation in campus networks. ACM Sigcomm Computer Communication Review, 2008,38(2):69−74. [doi: 10.1145/1355734.1355746]

[31] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer looks up service for Internet applications. ACM Sigcomm Computer Communication Review, 2001,31(4):149−160. [doi: 10.1145/964723.383071]

[32] Hadoop. http://hadoop.apache.org

[33] Hu GW, Xu K, Wu JP, Cui Y, Shi F. A general framework of source addresses validation and traceback for IPv4/IPv6 transition scenarios. IEEE Network, 2013,27(6):66−73. [doi: 10.1109/MNET.2013.6678929]

[34] Nayak AK, Reimers A, Feamster N, Clark R. Resonance: Dynamic access control for enterprise networks. In: Proc. of the 1st ACM Workshop on Research on Enterprise Networking. 2009. 11−18. [doi: 10.1145/1592681.1592684]

[35] Matias J, Garay J, Mendiola A, Toledo N, Jacob E. FlowNAC: Flow-based network access control. In: Proc. of the 2014 3rd European Workshop on Software-Defined Networks. 2014. 79−84. [doi: 10.1109/EWSDN.2014.39]

[36] Duan XY, Wang XB. Authentication handover and privacy protection in 5G HetNets using software-defined networking. IEEE Communication Magazine, 2015,53(4):28−35. [doi: 10.1109/MCOM.2015.7081072]

[37] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. In: Proc. of the 6th Symp. on Operating System Design and Implementation (OSDI 2004). 2004,51(1):107−113. [doi: 10.1145/1327452.1327492]

[38] Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE. Bigtable: A distributed storage system for structured data. ACM Trans. on Computer Systems, 2008,26(2):205−218. [doi: 10.1145/1365815.1365816]

[39] Tartakovsky AG, Polunchenko AS, Sokolov G. Efficient computer network anomaly detection by change point detection methods. IEEE Journal of Selected Topics in Signal Processing, 2013,7(1):4−11. [doi: 10.1109/JSTSP.2012.2233713]

[40] Peng T, Leckie C, Ramamohanarao K. Proactively defecting distributed denial of service attacks using source IP address monitoring. Networking, 2004,3042:771−782. [doi: 10.1007/978-3-540-24693-0_63]

[41] Tartakovsky AG, Rozovskii BL, Blažek RB, Kim H. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. IEEE Trans. on Signal Processing, 2006,54(9):3372−3382. [doi: 10.1109/TSP.2006.879308]

[42] POX. http://www.noxrepo.org/pox/about-pox/

[43] Kaffille S, Loesing K. Open Chord version 1.0.4 User's Manual. Distributed and Mobile System Group, Otto-Friedirich-Universitat. Bamberg, 2007.

[44] Tribe flood network 2000 (TFN2K). https://en.wikipedia.org/wiki/Tribe_Flood_Network

[45] MIT Lincoln Laboratory. DARPA intrusion detection scenario specific datasets 2000. http://www.ll.mit.edt/IST/

[46] Replay tools. http://tcpreplay.synfin.net/

附中文参考文献:

[7] 张永铮,肖军,云晓春,王风宇.DDoS 攻击检测和控制方法.软件学报,2012,23(8):2258−2072. http://www.jos.org.cn/1000-9825/4237.htm [doi: 10.3724/SP.J.1001.2012.04237]

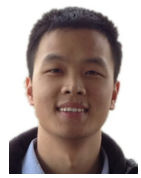
[8] 谢高岗,张玉军,李振宇,孙毅,谢应科,李忠诚,刘韵洁.未来网络体系结构研究综述.计算机学报,2012,35(6):1110−1119. [doi: 10.3724/SP.J.1016.2012.0l109]

王秀磊(1988－),男,山东邹城人,博士生,主要研究领域为网络安全,软件定义网络,未来网络体系结构.

陈鸣(1956－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络测量,网络性能分析与建模,分布式系统,未来网络.

邢长友(1982－),男,博士,副教授,CCF 会员,主要研究领域为网络与分布式计算,未来网络,网络流媒体.

孙志(1973－),男,博士生,工程师,主要研究领域为网络测量,网络性能分析与建模,负载均衡.



吴泉峰(1990－),男,硕士生,主要研究领域为未来网络,网络安全.