ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.9, September 2007, pp.2245−2258
DOI: 10.1360/jos182245

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

DDOS攻击检测和防御模型*

孙知信[1,2+], 姜举良[1,2], 焦 琳[1,2]

[1](南京邮电大学 计算机学院,江苏 南京 210003)

[2](南京邮电大学 计算机技术研究所,江苏 南京 210003)

DDOS Attack Detecting and Defending Model

SUN Zhi-Xin[1,2+], JIANG Ju-Liang[1,2], JIAO Lin[1,2]

[1](College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

[2](Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

+ Corresponding author: Phn: +86-25-85198095, E-mail: sunzx@njupt.edu.cn, http://www.njupt.edu.cn

Sun ZX, Jiang JL, Jiao L. DDOS attack detecting and defending model. *Journal of Software*, 2007,18(9): 2245−2258. http://www.jos.org.cn/1000-9825/18/2245.htm

Abstract: This paper presents the APA-ANTI-DDoS (aggregate-based protocol analysis anti-DDoS) model to detect and defend the DDoS attack. APA-ANTI-DDoS model contains the abnormal traffic aggregate module, the protocol analysis module and the traffic processing module. The abnormal traffic aggregate module classifies the network traffic into normal traffic and the abnormal traffic; the protocol analysis module analyzes the potential features of DDoS attack traffic in the abnormal traffic; the traffic processing module filters the abnormal traffic according to the current features of DDoS attack, and resumes the non-attack traffic with the help of testing the congestion control feature of the traffic. The paper then implements the APA-ANTI-DDoS system. The experimental results show that APA-ANTI-DDoS model can primely detect and defend DDoS attack and resume the non-attack traffic at the time of miscarriage of justice to guarantee the legal communication traffic.

Key words: distributed denial of service attack; congestion control; flood attack; aggregate; abnormal traffic; protocol analysis

摘 要: 提出了基于聚集和协议分析防御分布式拒绝服务攻击(aggregate-based protocol analysis anti-DDoS,简称APA-ANTI-DdoS)模型来检测和防御 DDoS 攻击.APA-ANTI-DDoS 模型包括异常流量聚集、协议分析和流量处理.异常流量聚积把网络流量分为正常流量和异常流量;协议分析寻找异常流量中 DDoS 攻击流量的特征;流量处理则根据当前的 DDoS 攻击流量特征,过滤异常流量并测试当前聚积流量的拥塞控制特性,恢复被误判的流量.随后实现

* Supported by the National Natural Science Foundation of China under Grant No.60572131 (国家自然科学基金); the Key Technologies R&D Program of Jiangsu Province of China under Grant No.BE2007058 (江苏省科技攻关项目); the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China and Nanjing Government (国家教育部和南京市回国人员基金); the Scientific Development Foundation of Government (南京市科技发展计划); the Scientific Research Foundation of NJUPT under Grant No.NY206008 (南京邮电大学攀登计划); the Scientific Research Foundation of ZTE and Huawei Corporation of China (中兴及华为基金)

Received 2005-12-30; Accepted 2006-06-01

了 APA-ANTI-DDoS 系统.实验结果表明,APA-ANTI-DDoS 模型能很好地识别和防御 DDoS 攻击,能在误判时恢复非攻击流量,保证合法的正常网络通信.

关键词: 分布式拒绝服务攻击;拥塞控制;洪流攻击;聚集;异常流量;协议分析

中图法分类号: TP309 **文献标识码:** A

DDoS攻击[1]是一种非常重要的网络异常流量来源,其对网络的破坏力是惊人的,将会造成巨大的损失[2].目前检测和防御DDoS攻击有很多种办法:一些是基于包内容过滤的技术,如Snort中有几十条关于DDoS的包内容过滤规则;一些特别的检测机制[3,4]和几种保护方法[5−7]注重完善目前网络协议,补救其漏洞;还有一些使用各种统计模型[8,9],希望抓住发生DDoS攻击时的网络流量统计特点,以识别和防御DDoS攻击.

本文介绍的基于聚集和协议分析防御分布式拒绝服务攻击(aggregate-based protocol analysis anti-DDoS,简称 APA-ANTI-DdoS)模型吸取异常流量聚积和协议特征分析思想,根据 DDoS 攻击最基本的特征——短时间内网络上有大量相同目的地址的 IP 包,实现检测和防御 DDoS 攻击,同时运用流量拥塞控制思想纠正误判行为,恢复正常的大流量通信.

本文第 1 节介绍相关研究.第 2 节是 DDoS 攻击特性研究.第 3 节是 DDoS 攻击检测和防御模型 APA-ANTI-DDoS.第 4 节是 APA-ANTI-DDoS 系统实现.第 5 节是算法分析.第 6 节是实验分析.第 7 节是结论与展望.

1 相关研究

DDoS检测和防御方法可以分为 3 类:基于协议特征分析的DDoS检测和防御[3,4]、基于聚积的DDoS检测和防御[10−12]以及基于网络流量统计模型的DDoS检测和防御[8,9,13].目前,DDoS检测和防御方法还存在以下几个问题:基于协议特征分析的DDoS检测和防御方法只能用于防御具有明显异常流量协议特征的DDoS攻击类型[3,4],对于许多没有明显协议区别特征的DDoS攻击类型则无效;基于聚积的和基于网络流量统计模型的DDoS检测和防御方法不能区分正常的大流量和DDoS攻击流量,将会导致合法用户流量被误判为攻击流量,误判发生后无法恢复正常的大流量通信.

文献[3,4]提出了对 TCP 的连接状态进行研究,以主动识别 TCP SYN FLOOD 攻击.这种方法只适用于 TCP SYN FLOOD 攻击,而无法检测和防御 UDP FLOOD 和 ICMP FLOOD.文献[14]在网关上防止伪造源地址 DDoS 攻击标准,但其只适用于伪造源地址的 DDoS 攻击.文献[8,9]尝试建立统计异常流量检测模型,检测和防御 DDoS 攻击,但是没有一个很好的处理异常流量的策略,即使检测出当前出现异常网络流量,也不能区分是合法的高速流量,还是非法的 DDoS 攻击流量.文献[15,16]提出了在贴近攻击源端网络防御 DDoS 攻击,把边缘网络(edge network)分为 DDoS source,DDoS victim 或者 normal 类型,对不同的类型实行不同的防御策略,也同样存在着不能区分合法的高速流量和非法的 DDoS 攻击流量的问题.

当发生DDoS时,短时间内网络上会产生大量具有相同目的地址(被攻击目标主机的IP地址)的IP包.本文运用Bloom Filter算法[10,17−20],把 2^{32}个IP地址映射到一个较小的空间来检测DDoS攻击.文献[10]将Bloom Filter算法应用到DDoS的防范工作,阐述了如何检测DDoS攻击,但没有解决不同网络环境中HashTable溢出上限取值大小的问题和HashTable定时更新时的间断性溢出问题:算法中上限取得小,会把太多的合法流量误判为攻击流量;上限取得大,又会把相当一部分攻击流量漏检了.而且不同的网络环境上限值应该是有所区别的,文献[10]没有给出上限值计算方法,实验中只是随意指定了几个常数(5000,7000);HashTable必须按周期t更新,在下一个周期开始时,置所有的域为初始值(比如 0).这样,如果当前HashTable溢出,则不能保证溢出的连续性,在下一个周期开始时又必须累积到上限那么多的包数才能溢出.文献[10]没有提出相应的异常流量分类处理策略和防御策略,也没有涉及对误判行为的处理方法.文献[10]中HashTable(如后文图 4 所示)还存在碰撞问题,这将在第 5.2 节加以详细描述.

本文以 Bloom Filter 算法为基础,提出了 APA-ANTI-DDoS 模型.本文将在模型的实现系统中改进目的地址聚积的 HashTable 的计数方式,缓解 Hash 碰撞;运用反馈思想实现动态计算溢出上限值;HashTable 阈值减半更

新,解决 HashTable 间断性溢出问题;同时,本文也给出了异常流量分类处理策略和防御策略,误判恢复策略将包含在异常流量处理策略和防御策略中.

2 DDoS 攻击特性研究

DDoS 攻击包括 UDP flood,TCP/SYN flood,ICMP/PING flood,ICMP/SMURF flood 和这些攻击的任意组合.文献[21]详细分析了 DDoS 攻击特性,把 DDoS 攻击分为直接攻击(direct attack)和反射攻击(reflector attack).

对 DDoS 直接攻击和间接攻击进行分析,可以得到以下 DDoS 攻击特征:

A. 攻击流量目的地址过于集中,且无拥塞控制特性.正常的网络应用程序应该是考虑到网络拥塞的,当发现网络通信质量很差时,正常的应用程序应该减少相互通信速率.这种拥塞控制规则可能由传输层实现,如 TCP 拥塞控制策略;也可能由应用层负责实现而传输层没有实现,比如,UDP 协议没有拥塞控制策略.DDoS 攻击流量不会考虑网络拥塞,攻击流量持续不断,出现网络拥塞仍然大量发包.

B. 流向目标机的 TCP/UDP 流量目的端口太多或者目的端口过于集中.用随机端口攻击目标机,出现同时向目标机数千端口发送数据包;用固定端口攻击目标机,出现同时向目标机单一端口发送大量数据包.

C. 当发生 TCP FLOOD/ICMP Flood 时,流向目标机流量包含大量的相同标志位数据包.标志位数据包可能是 TCP SYN 包、TCP RST 包、ICMP ECHO 包、ICMP MASK 包、ICMP TIME 包等.

3 APA-ANTI-DDoS 模型

APA-ANTI-DDoS 模型如图 1 所示,包括异常流量聚集、流量抽样、协议分析和流量处理.异常流量聚集把网络流量分为高度可疑流量、一般可疑流量和正常流量.流量抽样对高度可疑的聚集流量进行抽样,提供聚集流量抽样信息给协议分析继续处理.协议分析处理这些高度可疑的网络流量抽样信息,试图分析出它的 DDoS 攻击特性,给出过滤规则,并交给流量处理.流量处理直接处理可疑的实体流量(包括高度和低度可疑的流量),根据当前有效规则信息决定当前流量是否被过滤或者转入正常处理过程.正常流量则直接转入正常处理过程.流量处理需要反馈信息给协议分析,协议分析也需要反馈信息给异常流量聚集,图 1 中 Feedback 箭头表示反馈信息.

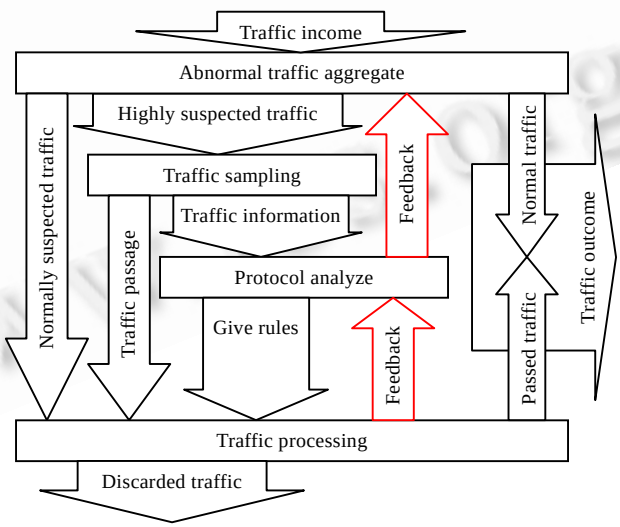

Fig.1 APA-ANTI-DDoS model

图 1 APA-ANTI-DDoS 模型

APA-ANTI-DDoS 模型定义协议分析状态机来统一分析聚集流量中的各个协议流量(包括 ICMP,UDP 和 TCP),如图 2 所示.表 1 定义了状态机事件,表 2 定义了状态机状态.

状态机是聚集流量的状态机.在发生DWOV事件的情况下,如果当前聚集流量状态处于FREE状态,则应该转移到WAITTING状态.在分析不出规则(DDoS特性)的情况下,如果发生NOOV事件或者TOUT事件,则返回FREE状态;在分析出规则或者发生UPOV的情况下,状态机给出协议规则,进入SLOWSTART状态,开始测试目标协议流量的拥塞控制特性,调节目标协议流量的过滤率;在分析不出规则但是又发生UPOV事件的情况下,状态机将给出关于整个聚集流量的过滤规则,进入SLOWSTART状态,开始测试目标流量的拥塞控制特性,调节目标聚集流量的过滤率.

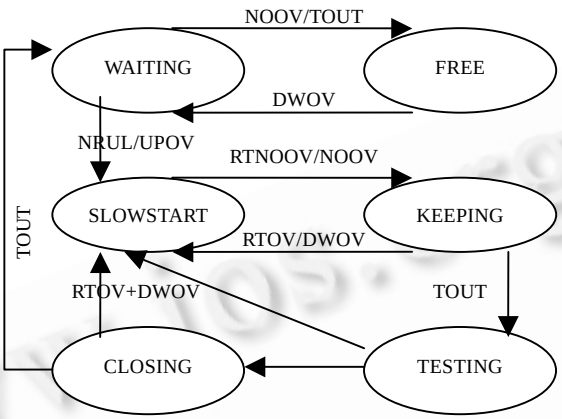

Fig.2 Protocol analysis model state machine

图2 协议分析模型状态机

初始进入 SLOWSTART 状态时,设置一个很小的过滤率 t(0<t<<1,比如取 t=1/16),过滤目标聚集流量的很小一部分.如果过滤导致目标聚集流量迅速减小发生 NOOV 或者 RTNOOV,则进入 KEEPING 状态.如果过滤没有使目标聚集流量减少,则加倍过滤率 t,取 t=Min(2×t,1).SLOWSTART 状态是一个特别的状态,其在增加过滤率 t 之前,需要自上一次增加过滤率 t 的时间延迟一个最小时延 DELTA_DELAY,以保证当前增长的过滤率 t 作用于当前目标流量一段时间.这里体现了第 2 节所述的 DDoS 特性 A.当处于 KEEPING 状态时,保持本聚集流量过滤率(可能是对特定 IP 的 TCP 流量、特定 IP 的 UDP 流量、特定 IP 的 ICMP 流量,也可能是整个 IP 流量),直至出现 TOUT 事件,转移到 Testing 状态;或者出现 DWOV 加上 RTOV 事件回归到 SLOWSTART 状态.这里的意图也很明显:攻击很可能是间断性的,我们无法防御所有的间断性攻击,但可以防御间隔时间较小的间断性攻击流量.TESTING 状态是非过滤状态,但依然保持规则的有效性(如果有的话).当处于 TESTING 状态时,如果发生超时事件,则转移到 WAITTING 状态,规则将自动失效;如果发生 DWOV 加上 RTOV 事件,则立刻转入 SLOWSTART 状态.TESTING 也是为了防止间断性攻击.

4 APA-ANTI-DDoS 系统

4.1 流量抽样实现

文献[22]给出了 3 种采样模型,如图 3 所示.

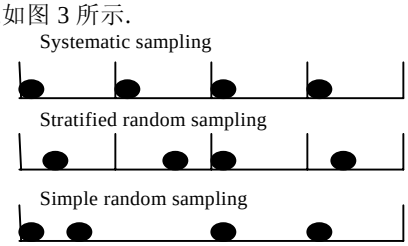

Fig.3 Sampling model[20]

图 3 采样模型[20]

用进入包计数触发采样:固定采样每隔 N 个包,采集 1 个;分层固定采样每 N 个包中随机采样 1 个;随机采样则随机地采集包,其平均采集概率为每 N 个包采集 1 个.本文的流量抽样模型不同于文献[23,24]中的网络流量抽样模型,文献[23,24]是对整个网络流量的抽样,要求抽样流量能够反映网络流量的统计特征.由于本文的流量抽样模型是对高度怀疑流量的抽样,其抽样流量只需反映协议分析需要的 DDoS 攻击特征——数量特性.本文的流量抽样模型选取分层固定采样方式.

4.2 异常流量聚集实现

如图 4 所示,APA-ANTI-DDoS系统设置一种由 4 个独立Hash函数组成的 4×256 的表结构HashTable来跟踪时间段t内通过路由器到达不同目的地址的IP包数,实现异常流量聚集.设IP地址为$a.b.c.d$,*Hash*1 是对a域的一一映射,*Hash*2 是对b域的一一映射,*Hash*3 是对c域的一一映射,*Hash*4 是对d域的一一映射.当一个IP包进入路由器后,其目的地址被 4 个独立的Hash函数分别映射到各自不同的 256 个域中的某一个,此时,保存在被映射域中的变量a_{ij}($1\le i\le 4,1\le j\le 256$)中最小的(可能是 1 个或者多个)加 1.如果一个包的到达使当前包目的IP地址的 4 个映射域都达到(或超过)上限,则说明到达这个包的目的地址的IP包频率已经非常高了.

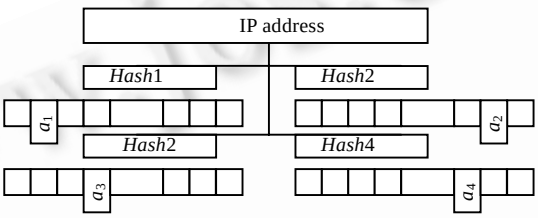

Fig.4 Abnormal traffic aggregate HashTable

图 4 异常流量聚集 HashTable

HashTable需要定时更新.HashTable更新包括重置HashTable域值和动态计算溢出上限:重置HashTable域值不是置 0,而是对所有域执行$a_{ij}=a_{ij}/2$;动态计算溢出上限见算法 1.

算法 1.

//定义参数并初始化a_{ij},如图 2 所示,*back*为协议分析反馈值,在第 4.3.1 节算法 2 中详细叙述

float N=5,l=1,ρ=0.5

//*UpLimit* 和 *DownLimit* 算法

$$l_{ij}=\begin{cases}0, & a_{ij}=0\\ 1, & a_{ij}!=0\end{cases},\ TempLimit=\mathrm{Min}\left(\frac{\sum_{j=1}^{m}a_{1j}}{\sum_{j=1}^{m}l_{1j}},\frac{\sum_{j=1}^{m}a_{2j}}{\sum_{j=1}^{m}l_{2j}},\frac{\sum_{j=1}^{m}a_{3j}}{\sum_{j=1}^{m}l_{3j}},\frac{\sum_{j=1}^{m}a_{4j}}{\sum_{j=1}^{m}l_{4j}}\right)$$

IF (*DownLimit*==0)

DownLimit=*TempLimit*×*l*,*UpLimit*=*TempLimit*×*N*

ELSE

L=*l*+*back*,*back*=0

DownLimit=*DownLimit*×(1−ρ)+*TempLimit*×ρ×*l*,*UpLimit*=*DownLimit*×*N*

ENDIF

HashTable 设定了两个上限值:*UpLimit* 和 *DownLimit*.设置两个上限的理由是,HashTable 总是要有一定的下限溢出流量,启动后续的算法 2 设置 *back*,调节当前溢出上、下限值,又要保证对高聚集流量的快速反应.初始上限值 *UpLimit*=*N*×*TempLimit*,*DownLimit*=*TempLimit*.以后,*Uplimit* 和 *DownLimit* 由 *back* 值、历史 *l* 值、历史 *Uplimit*、*DownLimit* 值和当前 *TempLimit* 值共同决定.*TempLimit* 是刚刚过去的一个周期非 0 域值统计平均值,它去掉了域值为 0 的域的影响.其中,*back* 是由协议分析反馈的,是对当前溢出上限值的修正值.

HashTable聚集结果分为无溢出、半溢出、下限溢出和上限溢出.无溢出则正常转发流量;下限溢出和上限溢出需要对包信息进行采样,再由规则执行继续处理包;半溢出则直接由规则执行继续处理包.对于下限溢出和上限溢出,同时计算其聚集溢出流速IP *OVERRATE*=Min($a_{1\cdot m1}$,$a_{2\cdot m2}$,$a_{3\cdot m3}$,$a_{4\cdot m4}$)/(当前时间——HashTable上次更新时间),$a_{1\cdot m1}$,$a_{2\cdot m2}$,$a_{3\cdot m3}$,$a_{4\cdot m4}$是当前溢出对应HashTable的 4 个域.IP OVERRATE攻击协议分析使用.

4.3 协议分析

协议分析对异常流量聚集出的每个聚集IP流量进行分析,同时,协议分析还需要反馈信息给异常流量聚集,以实现动态调整HashTable的溢出上限.表 1 定义的事件在APA-ANTI-DDoS系统中有如下对应关系:UPOV对应HashTable上限溢出;DWOV对应HashTable下限溢出;NOOV对应HashTable无溢出;RTNOOV对应本协议分析结构处理的IP流速/IP *OVER RATE*<=NORMAL;RTOV对应本协议分析结构处理的IP流速/IP *OVER RATE*>=*HIGH*.*NORMAL* 和 *HIGH* 都是比例常数(0<*NORMAL*,*HIGH*<1).

4.3.1 协议分析反馈信息——*Back* 调整

图 2 定义的协议分析状态机只是定义了单一的某个具体IP流量聚集的状态分析过程.协议分析同一时间内处理的IP流量聚集个数应该是有限制的,设最大处理聚集个数为MAX_NUM.对于整个协议分析来说,还有可能出现下面两种情况:(1) 出现少于MIN_NUM个IP流量聚集,此时会导致HashTable的溢出上、下限长时间不能得到更新;(2) 出现多于MAX_NUM个IP流量聚集,会导致超出协议分析的处理极限.为此,我们需要运用算法 1 中所述的 *back* 来调整HashTable的溢出上、下限.*Back* 调整算法见算法 2.

算法 2.

```
//define the global parameters
float back=0, float back_hold=0, bool reverse=false
//back Algorithm
IF (state 1==true||state 2==true)
    IF (back_hold)
        IF (state 1==true && back_hold<0||state 2==true && back_hold>0)
            IF (!reverse)
                back_hold×=2
            END IF//(IF (!reverse))
            back=back_hold
        ELSE
            back_hold=back_hold/−2,back=back_hold,reverse=true
        ENDIF//(IF state 1==true && back_hold<0||state 2==true && back_hold>0)
    ELSE
        IF (state 1)
            back=back_hold=−0.1
        ELSE
            back=back_hold=0.1
        ENDIF//IF (state 1)
    ENDIF//IF (back_hold)
ELSE
    back_hold=0, reverse=false
ENDIF//IF (state 1==true||state 2==true)
```

4.3.2 协议分析结构

如图 5 所示,用TYPE_FIELD结构统计当前聚集IP流量的标志包统计特性.定义了 4 个域:*iRateH*,历史记录流量速率;*iRecH*,前一记数值;*iRecN*,当前记数值;*Htime*,历史记录时间.*iRecN* 域对当前聚集流量的标志数据包进行计数,统计其聚集IP标志包流速.TYPE_FIELD结构最多每隔ALPHA_SPAN秒就更新一次.更新时,*iRateH* 更新为当前的平均流速 *iRateH*=(*iRecN*−*iRecH*)/(当前时间−*Htime*);*Htime* 域更新为当前时间;*iRecH* 更新为 *iRecN* 的当前计数值;*iRecN* 则保持不变.

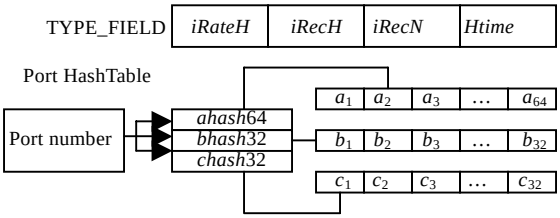

Fig.5 TYPE_FIELD and port HashTable

图 5 TYPE_FIELD 和端口散列表结构

端口散列表类似于异常流量聚集中的 HashTable,其工作方式与第 2.2 节所述类似.但是协议分析是用来计算协议端口(TCP/UDP)流量的聚集程度和分散程度的.端口散列表 Hash 函数设置如下:*ahash*64 是对端口第 0~5bit 的一一映射;*bhash*32 是对端口第 6~10bit 的一一映射;*chash*32 是对端口第 11~15bit 的一一映射.分析方法见算法 3.

算法 3.

$$l_{a_j}=\begin{cases}0, & a_j=0\\ 1, & a_j!=0\end{cases}(1\le j\le 64),\quad l_{b_j}=\begin{cases}0, & b_j=0\\ 1, & b_j!=0\end{cases}(1\le j\le 32),\quad l_{c_j}=\begin{cases}0, & c_j=0\\ 1, & c_j!=0\end{cases}(1\le j\le 32)$$

端口离散度:
$$Discrete=\frac{\sum_{j=1}^{64}l_{a_j}+\sum_{j=1}^{32}l_{b_j}+\sum_{j=1}^{32}l_{c_j}}{64+32+32}$$

端口集中度:
$$Aggregate=\frac{\mathrm{Min}(\mathrm{Max}(a_i),\mathrm{Max}(b_j),\mathrm{Max}(c_l))}{\sum_{j=1}^{32}c_j}$$

算法 3 对聚集 IP 流量的 TCP/UDP 端口特性进行统计.*Discrete* 代表的是所有端口散列表的非 0 域占所有域的比例;*Aggregate* 代表的是端口散列表中聚集的最大值占所有当前 IP 流量的比例.聚集的最大值不是简单的所有端口散列表各个域的最大值,而是各单个散列函数对应域最大值中的最小值.很明显,各单个散列函数对应域最大值中的最小值更接近实际的最大端口聚集流量值.当 *Discrete* 接近 1 时,表示端口过于分散,此时很可能是随机端口攻击;当 *Aggregate* 接近 1 时,表示端口过于集中,此时若是攻击必然是固定端口攻击.

端口散列表还设置了 6 个域用于统计当前最大流量端口:*aIndex*,*a*Max,*bIndex*,*b*Max,*cIndex*,*c*Max.每次端口散列表更新置 0 时,这 6 个域也置 0.当对某个端口流量进行映射时,设映射域分别为 *a*[*a_i*],*b*[*b_i*],*c*[*c_i*].此时,如果 *a*[*a_i*]>*a*[*aIndex*],那么 *aIndex*=*a_i*,*a*Max=*a*[*a_i*];同样,如果 *b*[*b_i*]>*b*[*bIndex*],那么 *bIndex*=*b_i*,*b*Max=*b*[*b_i*];如果 *c*[*c_i*]>*c*[*cIndex*],那么 *cIndex*=*c_i*,*c*Max=*c*[*c_i*].由此,*aIndex*,*bIndex*,*cIndex* 将始终保存各自 Hash 映射的最大映射域值.由此,在怀疑是否固定端口攻击时,可以根据 *aIndex*,*bIndex* 和 *cIndex* 给出相应的端口值.最终给出的协议分析结构如图 6 所示.

协议分析结构是以聚集 IP 地址为标识的,包含 ICMP 分析块、UDP 分析块和 TCP 分析块.IP 是分析项的 IP 标识;Rule Index 是对应分析项给出的规则索引;IP OVER RATE 是异常流量聚集计算出的溢出流量的流速.协议分析结构每个协议分析块都包含 state 域,用于协议状态机,记录当前聚集流量所处的状态.对于 ICMP 协议,当发生 DDoS 攻击时,聚集 IP 流量的相应攻击 TYPE 类型包必然占据主要地位,为此,协议分析设置一个 TYPE_FIELD 结构数组 icmp_flag[],长度为 MAX_TYPE_VALUE+1.一般常用的协议 TYPE 最大值为 *MASKRP*=18,最小值为 *ECHORP*=0,即 MAX_TYPE_VALUE 可以取 18,数组长度为 19.ICMP 还设置了一个 TYPE_FIELD 类型的 icmp rate 域,统计聚集 IP 流量的 ICMP 类型包流速.

同样,对于 TCP 协议,发生 DDoS 攻击时,当是带标志位攻击,如 SYN Flood 和 RST Flood 时,类似于 ICMP 攻击,必然聚集 IP 流量的相应攻击标志类型包必然占据主要地位,为此,设置一个 TYPE_FIELD 结构数组

tcp_flag[],长度为 *FLAG_COUNT*=6.TCP 设置了一个 TYPE_FIELD 类型的 tcp rate 域,计算当前的聚集 IP 流量的 TCP 标志包流速.对于 UDP/TCP 攻击,还需要设置端口散列表(tcp_table/udp_table)以分析其流量端口特性.

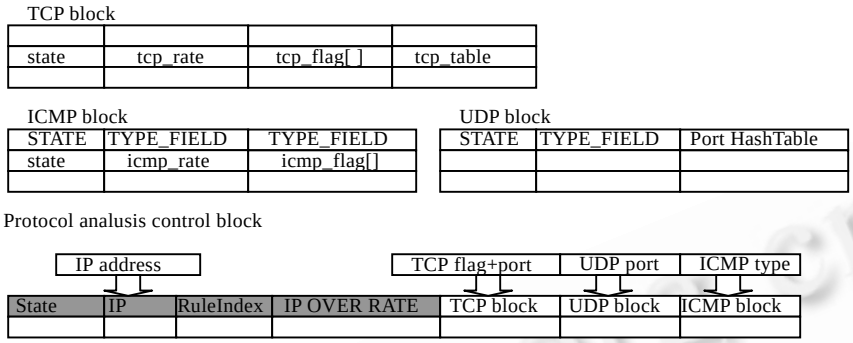

Fig.6 Protocol analysis control block

图 6 协议分析控制结构

4.3.3 过滤规则的产生

设定比例常数 e(0<e<1).对于 ICMP 控制块,当某个 TYPE 类型 *m_type* 的包占据相当部分 ICMP 流量时,

$$\frac{(icmp\_flag[m\_type].iRecN - icmp\_flag[m\_type].iRecH)}{当前时间 - icmp\_flag[m\_type].HTime} \geq IP\ OVER\ RATE \times e .$$

怀疑是 ICMP FLOOD 攻击,给出 ICMP 的 *m_type* 类型过滤规则.

对于 TCP 流量,当某个 FLAG 类型 *m_flag* 的包占据相当部分 TCP 流量时,

$$\frac{(icmp\_flag[m\_type].iRecN - icmp\_flag[m\_type].iRecH)}{当前时间 - icmp\_flag[m\_type].HTime} \geq IP\ OVER\ RATE \times e .$$

怀疑是 TCP FLOOD 攻击,给出 TCP 的 *m_type* 类型过滤规则.如果不是 TCP 固定 Flag 攻击,当 *TCP RATE*>=(*e*×*IP OVERRATE*),端口集中度 *Aggregate*>=*e*1 给出 TCP 固定端口(*tcp_table.aIndex*,*tcp_table.bIndex*,*tcp_table.cIndex* 对应的端口值)过滤规则.如果也不是 TCP 固定端口攻击,当 *TCP RATE*>=(*e*×*IP OVERRATE*),端口离散度 *Discrete*>=*e*2 时,怀疑是 TCP FLOOD 攻击,给出 TCP 随机端口过滤规则.

对 UDP 流量,当 *UDP RATE*>=(*e*×*IP OVERRATE*),端口集中度 *Aggregate*>=*e*1 给出 UDP 固定端口(*tcp_table.aIndex*,*tcp_table.bIndex*,*tcp_table.cIndex* 对应的端口值)过滤规则.如果不是 UDP 固定端口攻击,当 *UDP RATE*>=(*e*×*IP OVERRATE*),端口离散度 *Discrete*>=*e*2 时,怀疑是 UDP FLOOD 攻击,给出 UDP 随机端口过滤规则.

还存在一种特殊的过滤状态,此时,HashTable 处于上限溢出状态,而协议分析依然无法聚集出正常的规则.此时,协议分析会给出一个特殊的过滤规则,对整个聚集 IP 流量进行过滤.

最终给出的过滤规则如图 7 所示.IP 域代表过滤的目的地址.RuleMode 域指明给出的是 tcp 规则、udp 规则、icmp 规则,还是不区分协议的 IP 规则.TCP Mode 域是指在 tcp 规则有效时,给出的 tcp 流量过滤信息.首先指明是 flag,或者 port,或者 all 过滤.当是 flag 过滤时,flag 域给出过滤的 flag 标志;当是 port 过滤时,port 域给出过滤的 port 信息.UDP Mode 首先指明是 port 过滤,还是 all 过滤.如果是 port 过滤,则给出过滤的 port 信息.ICMP Mode 首先指明是 type 过滤,还是 all 过滤.如果是 type 过滤,则给出过滤的 type 信息.

4.3.4 规则执行

过滤规则按照有效的规则过滤特定聚集流量.每个协议规则定义 FILTER_FILED 过滤域,执行过滤算法 4.如果进入包 IP 地址匹配某个过滤规则的 IP,且包协议(IP/TCP/UDP/ICMP)对应过滤规则有效,则执行过滤算法.首先对进入的 IP 包计数,在 FilterMODE 为非 0 时,如果计数值 FilterCODE 位与上 Filter MASK 的值与 Filter MASK 不相等,则为通过流量,否则为过滤掉的流量;在 FilterMODE=0 时,如果 Filter CODE 位与 Filter MASK 的值和 Filter MASK 相等,那么为通过流量,否则为过滤掉的流量.

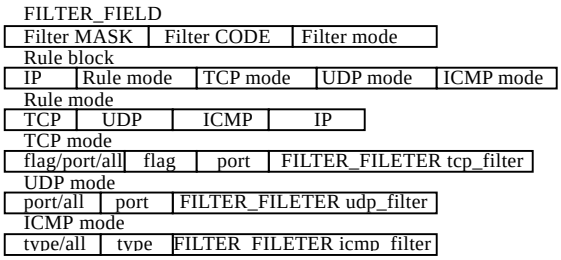

Fig.7 Rule block

图 7 过滤规则结构

算法 4.

```
Filter CODE++
IF (Filter MODE⊗((Filter MASK & CODE)==Filter MASK))
    //the passed traffic
ELSE
    //the filtered traffic
ENDIF
```

FILTER_FILED 和算法 4 实现了协议分析的 Backoff 算法对过滤率 t 的动态修改.FILTER_FILED 结构中的 Filter CODE 简单地对 IP 包计数,Filter MASK 和 FilterMODE 由相应的协议控制块设置和维护.可以实现的过滤率 t 与 FilterMASK 和 FilterMODE 的关系如下:

$$t=\begin{cases}1/k, & FilterMODE!=0,\quad k=FilterMASK+1,\quad \text{interger}\\ (k-1)/k, & FilterMODE==0,\quad k=FilterMASK+1,\quad \text{interger}\end{cases}.$$

5 算法分析

5.1 Hash映射表分析

Hash 映射表每条独立的 Hash 映射都会发生不同 IP 地址被映射到同一个域的情况.设 Hash 映射表是 $k\times m$ 结构,经过 k 个独立 Hash 函数映射到相同 k 个域的 IP 地址概率期望值为 $E(k\times m)$,则

$$E(k\times m)=\frac{2^{32}}{m^k}.$$

APA-ANTI-DDoS模型选择 4×256hash映射表结构,于是$E(4\times256)=2^{32}/256^4=1$,即概率上我们实现了IP地址的一一映射.HashTable映射表的映射空间为$Space=4\times256\times4=2^{12}$BYTE.如果采用对逐个IP记录其包流量的方法,则理论上需要记录所有IP的值,其映射空间为$Space_OLD=2^{32}\times4$(索引IP地址长度)×4(计数参数长度)=2^{36}BYTE>>$Space$.HashTable映射是一种时间、空间消耗相对都很小的映射算法.这保证了Hash映射表可以用很小的空间代价和时间代价实现对整个 32bit域的IP地址的聚集.

5.2 HashTable映射碰撞分析

文献[10]中,HashTable(如图 4 所示)还存在碰撞问题.比如对于目的IP流量 10.10.138.200 和 127.9.156.33 分别映射时,第 2 个独立映射Hash2 将映射到同一个域$a_{2\ 10}$.同理,每个独立映射的Hash函数都会带来冗余映射,如果冗余足够大,则将导致HashTable冗余溢出.我们对文献[10]中HashTable的映射计数方式作了改进:当一个IP包进入路由器后,其目的地址被 4 个独立的Hash函数分别映射到各自不同的 256 个域中的某一个,此时,保存在被映射域中的变量$a_{ij}(1\le i\le4,1\le j\le256)$中最小的(可能是 1 个或者多个)加 1.文献[10]中是对所有的被映射域$a_{ij}(1\le i\le4,1\le j\le256)$加 1,而APA-ANTI-DDoS系统只对其中最小的一个或者多个映射域加 1.

比如,当前目的地址为 10.10.138.200,分别映射到$a_{1\ 10}$,$a_{2\ 10}$,$a_{3\ 138}$,$a_{4\ 200}$这 4 个域.设$a_{1\ 10}$=400,$a_{2\ 10}$=500,$a_{3\ 138}$=300,$a_{4\ 200}$=200.APA-ANTI-DDoS系统中的HashTable将只对$a_{4\ 200}$加 1.因为由 4 个映射域中最小值$a_{4\ 200}$=200 可知,从开始映射到当前时刻通过的本目的地址 10.10.138.200 包个数不超过 200,其他 3 个映射域多出的值肯定是HashTable映射碰撞的结果,因此,我们也就没有必要再把另外 3 个域也加 1,虽然其他 3 个域加 1 对当前的目的地址个数不会造成影响,但必然会对下列地址造成影响:10.*.*.*,*.10.*.*,*.*.138.*.由此,HashTable只对每次映射域中的最小几个计数,缓解了HashTable映射的碰撞问题.

为了根本解决 HashTable 的碰撞问题,第 4.3.3 节中判断是否为攻击时,具体考察其真实流速占据溢出流速(IP OVER RATE)的比例.如果真实流速与溢出流速达到一个比较大的比例,则认为是攻击,同时可以据此得出攻击特性,再根据特性给出过滤规则.因此,APA-ANTI-DDoS 系统只是把 HashTable 溢出流量作为异常流量,最终需要对异常流量继续分析,最后对可疑流量根据过滤规则过滤,识别和防御 DDoS 攻击.即 PA-ANTI-DDoS 系统对异常流量的后续分析可以避免文献[10]中 HashTable 碰撞带来的假性 HashTable 溢出问题.

5.3 Hash映射表下限动态逼近算法

文献[10]中只是假定了一个固定不变的溢出上限值,结果或者溢出上限值偏高,致使更多的攻击流量被漏检;或者溢出上限值偏低,致使更多的合法通信流量被误检.算法1和算法3,具体阐述了Hash映射表的下限计算.其具有自适应性功能,能够根据实际网络状态和本地可使用的资源,使 *DownLimit* 逐渐逼近当前适当的 HashTable 过滤下限.图 8 是算法 3 的一个实例,当前 *T*0 时刻的下限值为 *L*0,合适的下限范围为(*L*0+3.5*S*,*L*0+4.5*S*).在时间(*T*0,*T*0+3*t*)内,*Back* 算法使 *DownLimit* 以指数(指数 2,基数 *S*)增长逼近目标范围(*L*0+3.5*S*,*L*0+4.5*S*);在(*T*0+3*t*,*T*0+5*t*)时间内以线性(−2*S*)增长逼近目标范围(*L*0+3.5*S*,*L*0+4.5*S*);在(*T*0+5*t*,*T*0+6*t*)时间内以线性(*S*)增长逼近目标范围(*L*0+3.5*S*,*L*0+4.5*S*).*DownLimit* 的顺序依次是 *L*0,*L*0+*S*,*L*0+3*S*,*L*0+7*S*,*L*0+5*S*,*L*0+3*S*,*L*0+4*S*.

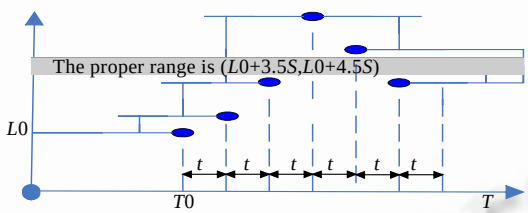

Fig.8 HashTable self-adaptive limitation analysis

图 8 HashTable 自适应性上下限分析

5.4 Hash映射表间断性溢出问题

文献[10]中,HashTable 映射表的间断性溢出如图 9 所示.由于每次更新置所有的计数域为 0,每次更新之后总需要有一段时间累积,HashTable 才能再次溢出.

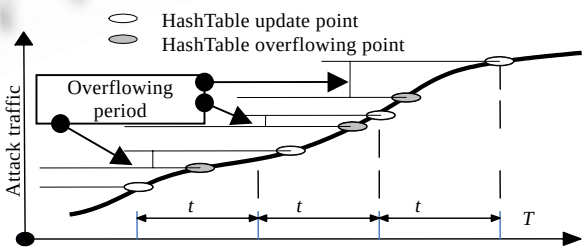

Fig.9 HashTable OverFlowing analysis

图 9 HashTable 间断性溢出

本文中的异常流量聚集在每次更新时,置计数域值为当前的一半,同时对 HashTable 半溢出流量执行过滤,

那么,当前溢出的聚集流量在下一个周期要么溢出,要么半溢出.第 4.2 节中,溢出和半溢出流量都将经过规则过滤,这样就避免了文献[10]中的间断性溢出而导致的过滤的不彻底性.设聚集流量平均到达速度为 v,HashTable 更新周期为 t,那么其平均周期流量总和 S 为 $S=\sum_{i=0}^{n}\left(\frac{1}{2}\right)^{n}vt$,在 n 趋于无穷大时,S 是收敛的,S 收敛于 $2vt$.

5.5 DDoS攻击行为分析

APA-ANTI-DDoS 模型协议分析状态机(如图 2 所示)的 SLOWSTART 状态和 KEEPING 状态测试目标聚积流量的拥塞控制特性,首先设定一个很小的过滤率 t,如果很小的过滤没有使相应的聚集流量减少,则加倍过滤率 t,取 t=Min(2×t,1).DDoS 攻击流量不遵循正常的网络应用逻辑,其在网络出现堵塞时,依然大量占据网络带宽.所以在 DDoS 攻击期间,t 将一直保持增长直至达到 100%.图 10 的上两幅图是 DDoS 攻击流量分析示意图,实线标识攻击流量流速分析,虚线代表过滤流量流速分析.其中分为 3 个时期:初始期、磨合期和稳定期.初始期攻击流速急速增长而过滤流速为 0(即还未被确认为攻击流量),磨合期过滤流速逐渐增长,最终达到稳定期,过滤流速基本上等于攻击流速.

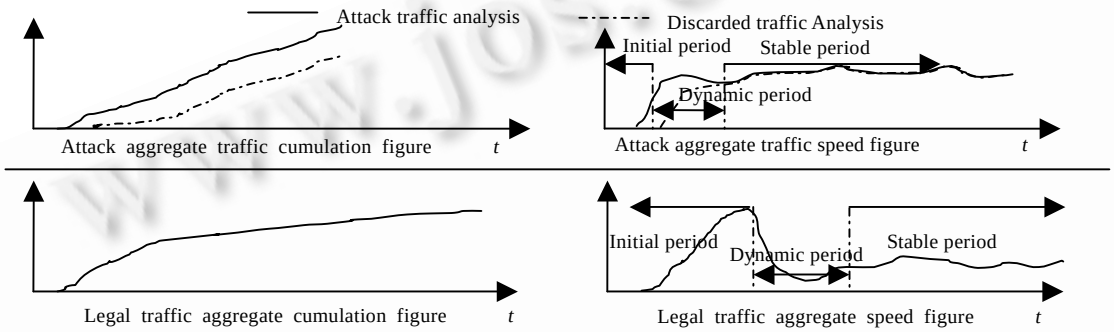

Fig.10 Aggregate traffic analysis

图 10 流量流速分析图

5.6 误判纠正行为分析

文献[8,9,15,16]不可避免地带来误判问题,但没有给出如何恢复被误判的合法用户的正常通信流量.正常的大流量一开始可能被误判为 DDoS 攻击流量,但是,随着相应的流量被过滤,其认为发生了网络拥塞,必须逐渐下调其通信速率,直至其流速减少到不足以引起 HashTable 溢出,从而恢复正常的通信,最终结果是其最大传输速率受到限制.图 10 的下面两幅图是遵循拥塞控制思想的合法大流量数据传输被误判过滤,然后流量恢复分析的示意图,其中分为 3 个时期:初始期、磨合期和稳定期.初始期流速急速增长,磨合期由于误判行为导致流量流速上下波动,最终,流量流速达到稳定期,其流速沿着平均速率上下波动.

对合法大流量通信误判纠正行为的前提是该大流量通信应用程序遵循流量拥塞控制原则,如果通信程序和所使用的传输层协议都不提供流量拥塞控制机制,那么 APA-ANTI-DDoS 模型将不保证其通信流量被误判后的恢复问题.理论上,将 100%地恢复遵循流量拥塞控制原则的大流量通信,如果其被误判为攻击流量.我们在后续实验分析中验证了这一理论分析的正确性.

6 实验分析

我们在某校园网内进行 DDoS 攻击实验,受害主机和攻击者网络带宽都是 100M,其中,攻击者模拟多个主机攻击受害主机,APA-ANTI-DDoS 系统布置在路由器上.攻击者使用 TFN2K 攻击受害主机(我们对 TFN2K 做了一些修改,增大了攻击流量带宽).实验中,攻击流量的包长度对攻击效果影响很大,长度越大,攻击效果越明显.对于 ICMP FLOOD,TCP FLOOD 和 UDP FLOOD 攻击,分别进行不同包长度的 DDoS 攻击,时间持续 2 分钟左

右.同时,我们还做了实验验证,正常的非攻击大流量聚集被误判为攻击时只是暂时性地中断其通信,被过滤的正常流量会很快恢复,但是恢复的正常流量会被限流.

图 11 显示 ICMP ping 攻击(ECHO REQUEST)时路由器记录的网络流量和流速图样,横轴表示时间,单位是秒.左图纵轴表示流量,单位个,右图纵轴表示流速,单位个/秒.图 11 攻击的 ping 包是长度固定的小数据包,图 12 攻击的 ping 包是长度固定的大数据包.从图中我们可以看出,在攻击的开始阶段,攻击数据流呈现出直线上升的趋势,导致 HashTable 溢出,启动协议分析,最终给出规则.开始过滤的流量比例很低,逐渐提高过滤比例,直至达到黄色标记位处,$t \geq 15/16$.完全过滤目标流量.图中的虚线表示攻击数据流,实线表示过滤数据流.

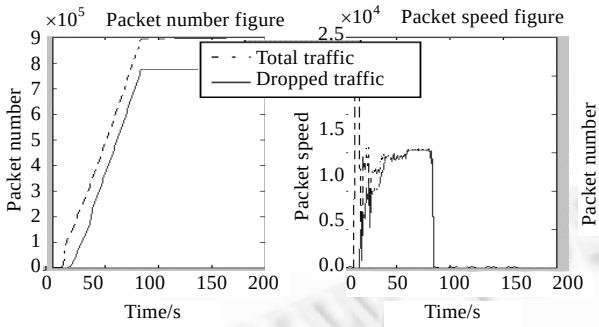

Fig.11 ICMP ping attack (small packet)

图 11 ICMP ping 小包攻击

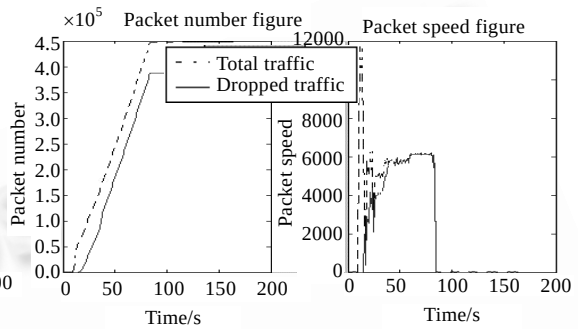

Fig.12 ICMP ping attack (big packet)

图 12 ICMP ping 大包攻击

图 13 显示的是 udp 攻击,长度固定为 40 字节的小数据包,通过攻击流量带宽达到峰值 4.5M 左右.图 14 显示的攻击是 tcp 攻击,攻击包长度是不固定的.它的攻击特征是 flag 标记固定,目的端口随机,源端口随机.其处理过程类似于图 11、图 12 中 ICMP FLOOD 的处理过程.

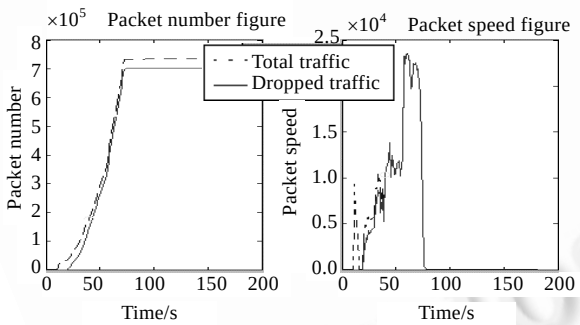

Fig.13 UDP flood attack (small packet)

图 13 UDP 洪流攻击(小包)

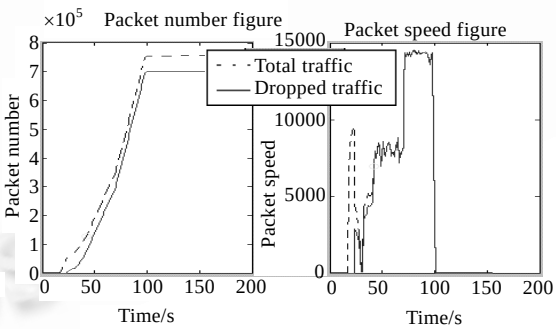

Fig.14 TCP flood attack (have the same flag)

图 14 TCP 洪流攻击(Flag 标记固定)

图 15 显示的是正常的非攻击大流量 IP 聚集,由于流速太大,促使 HashTable 上限溢出而最终导致该聚集流量被限流.开始时流速很大,后面因为 HashTable 溢出,而按一定比例过滤该聚集流量,且逐渐提高过滤流量比例.经过一定的时延,该聚集流量迅速减小,然后逐渐增加,直至达到基本稳定的聚集流速,这个聚集流速没有导致 HashTable 上限溢出.HashTable 经历了从下限溢出到上限溢出,再到下限溢出直至无溢出的过程.协议控制块则依次经历状态为 FREE→WAITTING→SLOWSTAET→KEEPING→SLOWSTART→KEEPING→TESTING→CLOSING→FREE.图 15 很好地验证了图 10 描述的理论流量恢复图样.APA-ANTI-DDoS 运用流量拥塞控制思想成功地恢复了合法大流量通信.图 15 是 FTP 文件下载,下载主机网络带宽 100M.当不加载 APA-ANTI-DDoS 系统时,可达最大下载速率 8~10M/s,平均包速达到 6000 个/s;当加载 APA-ANTI-DDoS 系统后,稳定期速率可达到 2M/s,其平均包速达到 1500 个/s.

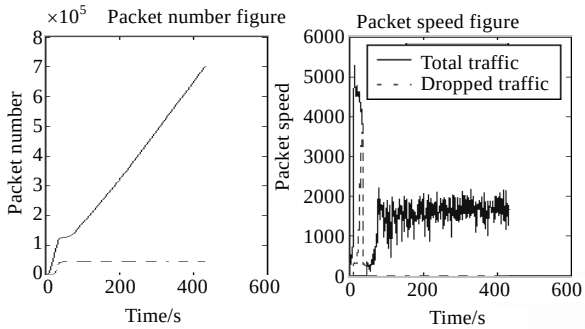

Fig.15 Legal traffic aggregate

图 15 合法大流量聚集

7 结论与展望

本文提出了一种基于聚集和协议分析的防御 DDoS 攻击的包过滤模型 APA-ANTI-DDoS,分别从异常流量聚集、流量拥塞特性分析和聚集流量协议分析 3 方面入手,检测和防御 DDoS 攻击.模型首先使用 HashTable 聚集可疑目的 IP 流量,然后运用协议分析验证各个可疑的聚集 IP 流量的协议特征,最后测试特征流量的拥塞特性,过滤无拥塞控制特性的特征流量.

实验表明,APA-ANTI-DDoS 解决了 HashTable 的上限值动态选取问题、HashTable 间断性溢出问题和 HashTable 碰撞带来的假性溢出问题;同时,APA-ANTI-DDoS 运用测试聚集流量的拥塞控制特性,成功地解决了检测误判行为发生后合法流量的恢复问题,以保证合法的正常通信行为.

作为本文的后续,我们将继续研究 APA-ANTI-DDoS 模型在分布式环境和多目标 DDoS 攻击中的应用.检测和防御 DDoS 攻击应该是一种网络群体性行为,网络节点应该通过合作共同应对 DDoS 攻击.

References:

[1] Meyer L, Penzhorn WT. Denial of service and distributed denial of service-today and tomorrow. In: Proc. of the IEEE 7th AFRICON Conf. Vol.2, 2004. 959−964.

[2] Chen Y, Hwang K, Kwok YW. Filtering of shrew DDoS attacks in frequency domain. In: Proc. of the IEEE Conf. on Local Computer Networks, 30th Anniversary. 2005. 786−793.

[3] Wang HN, Zhang DL, Shin KG. Detecting SYN flooding attacks. In: Proc. of the 21st Annual Joint Conf. of the IEEE Computer and Communications Societies. Vol.3, 2002. 1530−1539.

[4] Xiao B, Chen W, He YX, Sha EHM. An active detecting method against SYN flooding attack. In: Proc. of the 11th IEEE Int'l Conf. on Parallel and Distributed Systems. Vol.1, 2005. 709−715.

[5] Park KH, Lee HJ. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In: Proc. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies. Vol.1, 2001. 338−347.

[6] Savage S, Wetherall D, Karlin AR, Anderson T. Practical network support for IP traceback. ACM SIGCOMM Computer Communication Review, 2000,30(4):295−306.

[7] Snoeren AC, Partridge C, Sancheq LA, Jones CE, Tchakountio F, Kent ST, Strayer WT. Hash-Based IP traceback. ACM SIGCOMM Computer Communication Review, 2001,31(4):3−14.

[8] Li J, Manikopoulos C. Early statistical anomaly intrusion detection of DOS attacks using MIB traffic parameters. In: Proc. of the IEEE Systems, Man and Cybernetics Society, Information Assurance Workshop. 2003. 53−59.

[9] Kim YW, Lau WC, Chuah MC, Chao HJ. Packetscore: Statistical-Based overload control against distributed denial-of-service attacks. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies. Vol.4, 2004. 2594−2604.

[10] Chan EYK, Chan HW, Chan KM, Chan VPS, Chanson ST, Cheung MMH, Chong CF, Chow KP, Hui AKT, Hui LCK, Lam LCK, Lau WC, Pun KKH, Tsang AYF, Tsang WW, Tso SCW, Yeung DY, Yu KY. IDR: An intrusion detection router for defending against distributed denial-of-service (DDoS) attacks. In: Proc. of the 7th Int'l Symp. on Parallel Architectures, Algorithms and Networks. IEEE, 2004. 581−586.

[11] Jin C, Wang HN, Shin KG. Hop-Count filtering: An effective defense against spoofed DDoS traffic. In: Proc. of the 10th ACM Conf. on Computer and Communications Security. 2003. 30−41.

[12] Sun ZX, Tang YW, Zhang W, Gong J, Wang RC. A router anomaly traffic filter algorithm based on character aggregation. Journal of Software, 2006,17(2):295−304 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/295.htm

[13] Sun ZX, Tang YW, Cheng Y. Router anomaly traffic detection based on modified-CUSUM algorithms. Journal of Software, 2005,16(12):2117−2123 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/2117.htm

[14] Ferguson P. Request for network ingress filtering. RFC 2827, 2000. http://rfc.net/rfc2827.html

[15] Mirkovic J, Reiher P. D-WARD: A source-end defense against flooding denial-of-service attacks. IEEE Trans. on Dependable and Secure Computing, 2005,2(3):216−232.

[16] Siaterlis C, Maglaris V. Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics. In: Proc. of the 10th IEEE Symp. on Computers and Communications. 2005. 469−475.

[17] Bloom B. Space/Time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970,13(7):422−426.

[18] Gremillion LL. Designing a bloom filter for differential file access. Communications of the ACM, 1982,25(9):600−604.

[19] Mullin JK. A second look at bloom filters. Communications of the ACM, 1983,26(8):570−571.

[20] Fan L, Cao P, Almeida J, Broder AZ. Summary cache: A scalable wide-area web cache sharing protocol. IEEE/ACM Trans. on Networking, 2000,8(3):281−293.

[21] Chang RKC. Defending against flooding-based distributed denial-of-service attacks: A tutorial. IEEE Communications Magazine, 2002,40(10):42−51.

[22] Claffy K, Polyzos G, Braum H. Application of sampling methodologies to network traffic characterization. Computer Communication Review, 1993,23(4):194−203.

[23] Cheng G, Gong J, Ding W. Distributed sampling measurement model in a high speed network based on statistical analysis. Chinese Journal of Computers, 2003,26(10):1266−1273 (in Chinese with English abstract).

[24] Drobisz J, Christensen KJ. Adaptive sampling methods to determine network traffic statistics including the Hurst parameter. In: Proc. of the 23rd Annual Conf. on Local Computer Networks. 1998. 238−247.

附中文参考文献:

[12] 孙知信,唐益慰,张伟,宫婧,王汝传.基于特征聚类的路由器异常流量过滤算法.软件学报,2006,17(2):295−304. http://www.jos.org.cn/1000-9825/17/295.htm

[13] 孙知信,唐益慰,程媛.基于改进 CUSUM 算法的路由器异常流量检测.软件学报,2005,16(12):2117−2123. http://www.jos.org.cn/1000-9825/16/2117.htm

[23] 程光,龚俭,丁伟.基于统计分析的高速网络分布式抽样测量模型.计算机学报,2003,26(10):1266−1273.

孙知信(1964－),男,江苏南京人,博士,教授,主要研究领域为计算机网络与安全,计算机仿真,软件工程.

焦琳(1982－),女,硕士,主要研究领域为计算机网络与安全.

姜举良(1981－),男,硕士,主要研究领域为计算机网络与安全.