**DDOS防御系统的设计**

**一．背景：**

拒绝服务攻击（DDoS）一直是互联网的最严重威胁之一．即使在当前云计算环境下，DDoS仍然是一种主要攻击手段．DDoS利用多个主机向目标（victim）发送大量攻击报文导致目标不能提供正常的服务．一般来说，DDOS采用伪造源IP地址进行攻击．伪造源IP地址可以防止跟踪攻击源，利用了当前IP协议没有检查源IP地址真实性的缺陷．伪造源IP地址有三种方式，即随机伪造、．DDoS子网内伪造和固定伪造的攻击特性是大量突发攻击报文，使目标端来不及处理达到攻击目的．DDoS易于实施而难以防御，如何有效地防御DDoS攻击得到了持续的研究，并提出了多种方案．这些方案大致可分为两部分，即入侵检测和攻击过滤．发生攻击以后，如何从正常报文流中分辨出攻击报文并丢弃是防御DDoS攻击的重要步骤．根据防御DDoS的地方不同，可以分为攻击源端防御、路由器防御和目标端防御等三种方式，考虑到源端防御和路由器防御的复杂性和实施困难，在目标段防御成为一种便于实施的方法，也得到越来越多的重视和研究．为了在目标段防御DDoS攻击，研究人员提出了多种方法．鉴于DDoS攻击时突然出现大量攻击报文的特点，在发生攻击时，通过各种特征的真实流量轮廓对比正常流量轮廓，利用贝叶斯算法或者漏桶打分来决定是否攻击报文．这些方案使用了DDoS大量突发的特点，但是对于低流量的攻击无能为力，即使是伪造IP地址的攻击也不具有很好的过滤效果，同时这些方案的算法统计了IP报文的多种特征，加重了处理的负担．利用数据流相关性分辨DDoS攻击和正常的突发报文在近年来得到了大量关注，但这种方法无法实现很高的精度，大多用于定性分析．为了提高精度，根据DDoS伪造源IP地址的特点，一些研究利用布隆过滤器对DDoS报文进行过滤．但是这些方法只能对随机伪造地址的DDoS具有较好的防御作用．

本文综合了各种防御DDoS方案的优点，利用BF技术，设计了基于源地址信息的DDoS防御系统．本系统部署在目标端，容易实施．其工作原理是，在目标端正常工作期间，建立一个跳数和源IP地址有关的正常流量统计表；在DDoS攻击期间，如果是一个伪造地址的攻击报文，由于其IP没有出现或者出现的IP与其跳数不符合统计规律而被过滤；如果是一个在子网内伪造或者没有伪造地址的攻击报文，虽然其跳数和源IP地址符合映射关系，但是由于攻击报文的大量突发性，它将根据超出正常的统计流量特征而过滤掉．同HCF相比，为了减少存储空间的需求，本文采用了改进的计数布隆过滤器来保存原地址信息的统计值．本系统并不需要路由器和源端协助，在实际使用过程中，针对某个服务器配置这样一个防御系统是很方便的，因此具有实用性．

1. **过滤方案设计**

**2.1方案分析**

根据统计，一个服务器所面临的访问客户具有相对稳定性．也就是说，对一个web网站访问的合法地址大多数在以前出现过，一个服务器在碰到突发拥塞时，其82.9%的IP地址以前出现过；而受到CodeＲed攻击时，仅有0.6－14%的IP地址出现过．基于这个特点，在一个服务器正常工作时，对其访问的IP地址进行统计；而在攻击发生时，依据统计的合法IP地址轮廓可以大致判断攻击报文并过滤之．但是现在攻击者已经学得更聪明了，他们可以用一些IP地址正常访问服务器来，然后用这些IP地址进行攻击，以逃避用上述防御方法进行攻击过滤．因此在方法设计上，还需要采用特征轮廓来进行判别过滤．

通常对于一个特定的服务器来说，访问这个服务器的真正客户报文的跳数分布具有一定的统计规律．攻击发生时，多个攻击僵尸（zombies）送到服务器的报文相关跳数统计将大幅度提高．注意我们这里使用跳数，而不直接使用IP首部的TTL域，因为不同的主机有不同的TTL初始值，但是可以根据接收到的IP报文计算出其经过的跳数．当发现某个跳数对应的统计超过正常的统计规律时，就知道在这个跳数所对应的距离处具有攻击源．所以，在设计的算法中，让每个服务器依据跳数建立一个自己客户的IP地址统计空间，便于实现前述策略的过滤处理．

但是，一个服务器要创建一个表来保存如此信息代价很大，需要巨量的内存空间．因此，BF技术逐渐广泛应用到DDoS攻击过滤方法中．计数布隆过滤器（CBF）是BF的变种，可以对集合内的元素进行计数统计，因此适合用来建立访问频率统计轮廓，实现对没有伪造IP地址或者部分伪造地址的DDoS攻击过滤．

**2.2方案设计**

本算法的基本思想是在没有攻击发生时，对合法访问服务器的源IP地址访问频率进行统计，并与跳数关联，达到更好的过滤效果．在攻击发生时，对访问服务器的源IP地址进行打分，如果超过阀值，则认为是攻击报文而过滤之．目前，大多数系统仍然采用IPv4，源IP地址是一个32位的数据．对所有的IP地址进行完整的统计将耗费巨量内存，因此大多数文献采用了BF技术，减少对内存消耗．本算法借鉴并改进了当前对源IP地址的哈希算法．特别是关联了跳数之后，我们只对特定跳数内的源IP地址进行访问频率统计．

在本算法中，我们把IP地址分为n片（n=1，2，3…32），每片是m位，且m≥32/n．每片对应N个位m置，则N=2m．在BF进行hash算法时，直接将IP对应片的值映射到合适的位置，简化了hash运算过程，可实现高速统计和过滤．当m＞32/n时，表示各个片对应的IP分片具有重叠，这会增强各个片的相关性，减少差错率，但是会要求更多的内存．选择合适的n和m值，是效率和内存消耗的一个折衷．

为了在服务器端统计其合法客户的IP地址信息，本算法需要构造一个统计信息表．根据研究，服务器可以利用收到报文的TTL来计算其客户所经历的跳数值HC，且0≤HC≤31．根据前述的算法，并结合大多数文献的做法，我们取n=4、m=9来设计信息表，并在信息表中对各个跳数值的源IP地址的各片进行统计，如图1所示．

在图1中，我们把一个IP分为4片，每一片有9位，其对应IP地址的位分布如表1所示．

实际上，各个分片对应的位模式的差异对某些情况下的过滤表现有些差异，可以根据不同服务器受到的攻击模式的不同而进行调整．

表 1 算法中的分片与位映射

Table 1 Algorithm of fragmentation and a mapping

片名字 映射位

S1 0 ～ 9

S2 8 ～ 16

S3 16 ～ 24

S4 23 ～ 31

为了更好地进行统计，每个统计区域是4个字节，分为正常统计区和异常统计区．当服务器正常工作时，每来一次客户的访问，本信息表通过分析IP首部的源IP地址和TTL，将对应跳数m的IP地址对应各个片的位置分别加1，表示在正常统计区，记为xj，i ．．其中j表示分片的序号，而i表示片的值，其范围是0 ≤ i ≤ 511( 29 － 1)

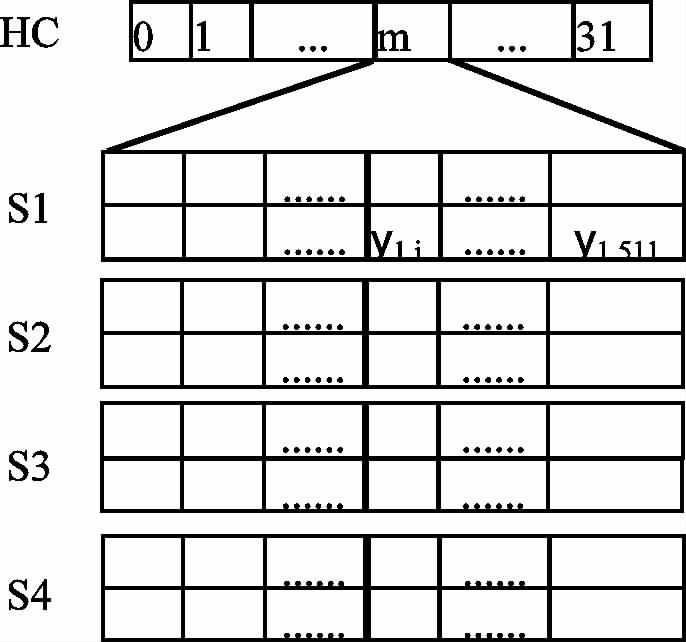


图 1 信息表的结构示意图

Fig． 1 Information table structure diagram

当攻击发生时，对每个新来的报文停止在正常统计区进行统计，并记录下正常统计区进行的统计时间tx．在攻击时间收到的报文在完全按照以前的计算，在异常统计区进行统计，记为yj，i，并计算此刻的统计时间ty．

在攻击阶段，对每个到来的报文，对其各个分片进行打分．每个分片利用其攻击统计轮廓和正常统计轮廓的比值作为打分基础．对于第j片，其分值fj可由公式（1）计算．

*yji / ty*

*fj =* ． ( 1)

*xji / tx*

根据攻击的特点，如果是随机伪造地址，某个分片j的xj，i如果为0，计算分值fj趋于无穷大;对于伪造固定IP地址，此IP地址对应的分值fj将不断增加，最终导致fj＞＞1;对于子网IP地址伪造，情况比较复杂，一般f1、f2甚至f3应该比较大，而f4可能因为伪造数字分散而比较小．因此我们利用公式(2)计算最终得分

4

*f =* ∑*aj*. *fj* ． ( 2)

*j =* 1

其中，aj 为分值因子，说明不同分片的分值权重，以反应不同分值的权值．一般地，取a1 =0.35，a2 =0.35，a3 =0.2，a4 =0.1．当f大于一个阀值时，本算法认为此IP报文是一个攻击报文阀值的选取与所在服务器IP地址相关的经验值．如果选择较小，将增加false positive;否则将增加false negative．

# **3. 过滤算法分析与设计**

**3.1 内存要求**

与一般的使用布隆过滤器的DDoS过滤算法相比，本算法使用了跳数来隔离统计处理，在同等情况下，增加了内存的消耗下面详细分析本方案需要的内存大小

如果某个特殊的跳数值存在合法的源 IP 地址访问，都将建立一个大小为 y 的空间单位根据前面的分析，每个单位需要保存攻击统计值和正常统计轮 廓，统计值的范围和选取的统计时间单位大小有关．对于一般服务器，每个统计值由4字节的整数表示足够了因此需要存储空间为

mem( m) = m. y. ( 4 + 4) = 8m. 4. 29 = 16m ( KB) ． ( 3)

其中 m 表示该服务器可能具有的跳数值的个数一般跳数值在0～31共32个值以内．根据观察，大多数情况下，一个服务器主机收到的合法报文的跳 数值个数在 10 至 20 之间． 考虑到不同服务器具有的m个数不一致，我们对每个跳数需要的空间采用动态申请，这样避免了不必要的内存开销． 例如在 m 为16时，需要的内存开销大概为256KB．此外还要增加数百字节的跳数统计开销同HCF的内存开销相比，如果其采用静态申请固定空间，本算法要求内存小很 多；如果 HCF 采用动态树形机构，在IP地址相当多 时，二者内 存要求相当，但 HCF需要更多的查询负载．。而与其他方法相比，本算法通过改进哈希算法，并没有增加总的内存消耗。

**3.2 过滤器效率**

DDoS过滤器的效率表现在false negative和false positive的大小，其中false negative 表示攻击报文误认为正常报文的概率，而false positive表示将正常报文误认为攻击报文的概率。

针对本方案，假设某个跳数下合法的源IP地址集合为T，元素个数为 t其4个分片的集合分别为T1 、T2 、T3 、T4．包含元素个数分别为t1 ，t2 ，… t4 ．当一个随机伪造IP地址的攻击报文到来的时候，其IP地址的各分片分别落于正常报文的概率为t1/T1 ，t2/T2， … t4/T4 ．在随机伪造地址的情况下，攻击报文误认为正常报文的概率(false negative) 为:

4 4

Pfn = ∏(ti / Ti) = (∏ti) / 232 ( 4)

*i =* 1 *i =* 1

由式(4)可知，Pfn 和合法报文的源地址各个分片的分布之积成比。由于本算法中根据跳数将不同的源IP分为若干区域，根据IP地址的地域特性，大大降低了某个跳数内的地址部分分布积，相对于其它类似方案，更加有效地降低了Pfn的最大值。

子网内随机伪造IP地址的情况变得复杂。设攻击报文的前面几个分片采用固定的子网． 随着攻击的进行，伪造子网部分的 IP 地址部分将因为统计值的累积增加而获得很高的分值，最终因为总分值超过阀值而被认为是攻击报文。如果正常报文的前面k部分与攻击报文的这部分有重叠，将会可能误认为攻击报文。假设攻击报文采用的子网部分涉及的元素个数分别为 q1 ，q2 ，… qk，当 k＞1时，误认为攻击报文的概率(false positive)大概为:

*k*

Pfp≈ ∏ ( qi / ti) ． ( 5)

*i =* 1

由(5)容易看出，Pfp 和攻击报文子网部分分布积与合法报文相应部分分布积之比成正比其(false negative)总是保持很低的水平．因为随着攻击的进行，k个固定子网部分的攻击分值不断提高，其被认为正常报文的概率越来越低。

根据前面的分析，本方法的误判率和合法源IP地址各个分片的有效元素分布 t1 ，t2 ，… t4 关系很大． 而 t1 ，t2 ，… t4 与 t 的关系为:

4 4

max (ti) ≤ t ≤ ∏ ti ( 6)

*i =* 1 *i =* 1

由式(6)可知，随着n的增加，在 t1 ，t2 ，… t4 一定时，t可在一个相当大的范围内变化。因此对本方案影响最大的是合法IP地址在各个部分的分布，而不是合法源IP地址的个数本事。

**3. 3 适应性分析**

一个设计良好的 DDoS 过滤机制，需 要良好的适 应性． 网络情况在不断变化，如果在不同的环境下需 要不同的配置才能发挥最好的效果，那是很难适应实 际需要的． 下面我们分析本算法的适应性．

首先，虽然网络在一定时间内会保持相对稳定， 但是长时间内网络拓扑将发生一定变化，这将影响部 分正常客户的跳数值． 此外，在不同的时候，服务器将有不同的客户统计轮廓． 为了自动反应这种变化，我们利用公式( 7) 不断刷新图 1 中的正常统计轮廓值:

xij = xij － old .α/ tx． ( 7)

这种刷新在一个固定时间间隔内进行式中α是一个相关系数0＜α<1，根据时间间隔和服务器的统 计变化规律而定。当一个源IP地址的跳数发生变化时或者不访问服务器时，原先跳数值下的相关 aij将减小甚至为 0，实现了自适应的变化。攻击发生时，xij 停止刷新，进入打分过滤操作。攻击停止时，xij 置为0，重新开始统计和刷新操作。

其次，本方案对攻击者攻击方式的变化具有很好的适应性。前面我们分析了攻击者随机伪造地址和子 网内伪造地址的适应性。如果攻击者伪造 TTL 值让我们得到错误的跳数，本方案也是能够很好适应的。因为即使攻击者能够改变跳数值，但是它很难知道那个跳数值下的IP地址各部分的统计轮廓，一样会根据前述的原理而过滤掉。

**3. 4 过滤算法**

根据前述，本方案有关算法总结如下． 在正常情况下没有攻击时:

①从接收报文 TTL 域计算跳数 m。

②如果此跳数下还没有统计流量数据，则申请 16KB 内存; 如果攻击阶段刚刚结束，将其中的各部分各个统计量xji、yij 的值置0。

③根据源 IP 地址各部分的值，进行统计。

④如果到达刷新时间，根据公式( 7) 进行刷新。

一旦认为攻击发生时:

①从接收报文的 TTL 域计算跳数 m。

②如果跳数 m 下没有历史统计量，作为攻击报文丢弃。

③取出源 IP 地址，在当前跳数下对其各个部分的当前统计值进行累加。

④根据公式(1)(2)计算源IP地址的分值f，如 果 f≥δ则为攻击报文，丢弃否则允许通过。．

1. **总结**

本文基于信息源统计的思想，提出了一种新的过滤DDoS攻击报文的方案。该方案的基本思想是每个服务器的客户信息源具有一定的统计分布，而攻击者很难准确知道这种分布来调整其攻击方案以逃避防御。相对于其他方案，该方案利用计数布隆过滤器对收到报文的跳数和源IP地址进行统计，有效地减少了内存的要求。与以前的工作相比，本文建议的方案对各种伪造报文攻击都能提供良好的过滤效果，此外，本方案独立工作于目标端，不需要路由器的协作，容易实施。

未来还可以在两个方面对本方案进行优化。首先是进一步减少内存要求。实际运行中发现，很多跳数下的统计信息仅仅占用极少量的内存（不到100字节），但却分配了16KB的空间。其次，打分模型可以进一步改进，最好能够动态适应，提供更细的粒度，进一步降低各种情况下的false negative 和false positive，提高算法的性能。