一种轻量级的 SYN Flooding 攻击检测方法*)

严 芬[1,2] 王佳佳[2] 陈轶群[2] 殷新春[2] 黄 皓[1]
(南京大学计算机软件新技术国家重点实验室 南京 210093)[1]
(扬州大学信息工程学院计算机科学与工程系 扬州 225009)[2]

摘 要 提出了一种轻量级的源端 DDoS 攻击检测的有效方法。本方法基于 Bloom Filter 技术对数据包信息进行提取,然后使用变化点计算方法进行异常检测,不仅能够检测出 SYN Flooding 攻击的存在,而且能够避免因为正常拥塞引起的误报。重放 DARPA 数据实验表明,算法的检测结果与类似方法相比更精确,使用的计算资源很少。

关键词 DDoS,源端检测,Bloom Filter,变化点检测,SYN Flooding 攻击

Light-weight Detection Method against SYN Flooding Attacks

YAN Fen[1,2] WANG Jia-jia[2] CHEN Yi-qun[2] YIN Xin-chun[2] HUANG Hao[1]
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)[1]
(Department of Computer Science and Engineering, Technology Institute, Yangzhou University, Yangzhou, 225009, China)[2]

Abstract An efficient light-weight method for defending against DDoS attacks at the source-end is designed. We use Bloom Filter to pick up the abstract of packets, and then use change point computation technology to detect abnormity. The method can not only detect the existence of SYN Flooding attacks, but also avoid the false alarm of normal congestion. In experiment environment, DARPA data is replayed and the result shows that our method obtains more accurate detection result with less computation than other similar methods.

Keywords DDoS, Detection at the source-end, Bloom filter, Change point detection, SYN flooding attack

1 引言

DDoS(Distributed Denial of Service)攻击利用足够数量的傀儡机产生数目巨大的攻击数据包,对一个或多个目标实施 DoS 攻击,耗尽受害端的资源,使受害主机丧失提供正常网络服务的能力。最近的一次大规模 DDoS 攻击发生在 2007 年 2 月,全球 13 个根域名服务器中至少有 3 个遭受了黑客的攻击,时间长达 12h。DDoS 攻击已经是当前网络安全最严重的威胁之一,是对网络可用性的挑战。

从 DDoS 的检测位置划分,主要包含目的端检测、中间网络检测和源端检测 3 种方法。目的端 DDoS 检测比较容易,因为目的端的攻击数据流量最大,缺点是如果上游网络链接被阻塞,目的端不管做什么也于事无补。中间网络检测比源端检测具有更好的可实施性和更低的覆盖要求,但也存在边界网关的修改、路由器的负载、缺乏网间合作等问题。而源端检测是最为理想的一种方法,能够在受害端受到攻击之前阻止攻击的发生,将攻击对网络的威胁降到最低。

SYN Flooding 攻击是 DDoS 攻击的一种,又称为半开式连接攻击。这种攻击主要是利用 TCP 连接时的 3 次握手信息造成的。当攻击者恶意地快速、连续送出许多 TCP SYN 包给被攻击端,而没有后续确认包传出时,被攻击端可用的 TCP 连接队列会因为存储太多的正在等待连接的信息而超出其容许量,从而导致暂停服务。目前,DDoS 攻击中约有 90%是 SYN Flooding 攻击[1],所以对 SYN Flooding 攻击的检测是当前 DDoS 攻击研究的重点之一。

在保证检测率的基础上,本文提出了一种源端 DDoS 攻击检测方法。此方法主要采用了两种核心技术:一是采用基于 Bloom Filter 的数据结构,该结构使用静态的固定存储空间和静态存储方法,能够用较少的存储空间存储大量的数据包信息;使用 Hash 函数能够进一步使得数据在存储空间内平均分布。二是使用了 CUSUM 检测方法,它能够快速地反映出数据包特征的变化情况。本文方法不仅具有准确性高、检测率高、消耗的计算资源少的特点,而且能够区分正常的网络拥塞与攻击,并可以在攻击的早期检测出攻击的存在,同时适用于对付攻击工具发出多个具有相同伪造源地址的攻击包来对抗基于伪造源地址计数的攻击检测的情况。

本文第 2 节介绍 SYN Flooding 攻击检测的研究现状;第 3 节分析 SYN Flooding 攻击的特征及其与正常网络拥塞的区别;第 4 节介绍使用 Bloom Filter 提取攻击特征、判断数据包是否具有伪造的源地址的方法;第 5 节使用 CUSUM 算法对提取的特征进行检测,发现由于攻击引起的变化;第 6 节给出了实验结果及性能分析,并与同类工作进行了比较;最后对全文进行总结并展望未来的工作。

2 研究现状

鉴于 SYN Flooding 攻击的普遍存在性,目前对这种 DDoS 攻击检测方法的研究很多。文献[2-4]提出了数据包特征匹配的方法,将 3 次握手中第 1 次握手的数据包的特征与

*)基金项目:国家 863 高技术研究发展基金(2003AA142010)资助课题,国家自然科学基金(60473093)资助课题和江苏省高技术研究计划项目(BG2004030)资助课题。严 芬 博士研究生,主要研究方向为网络与信息安全;王佳佳 硕士研究生,主要研究方向为信息安全;陈轶群 硕士研究生,主要研究方向为信息安全;殷新春 教授,主要研究方向为密码学与信息安全;黄 皓 教授,博士生导师,主要研究方向为计算机信息系统安全、网络与信息安全。

第2次握手的数据包或者第3次握手的数据包特征进行匹配，利用异常检测的方法来发现攻击，但该方法不能区分正常网络拥塞情况与DDoS攻击。文献[5]提出了使用代理服务器代替服务器与客户端进行3次握手连接，但SYN Flooding攻击发生时，半连接的数目非常多，因而对代理服务器的性能要求很高。A Zuquete[6]和J. Lemon[7]都通过改善TCP协议的性能来缓解和消除攻击，通过缩短SYN Timeout、修改SYN Cookie和增加SYN Cache等方法来减小3次握手连接需要的时间和增大能够维持的连接数量，但该方法对大量的、快速半连接的出现仍然无能为力。文献[8]将当前网络中SYN数据包的到达率与正常模型比较，判断是否发生了攻击，该方法同样未能区分正常网络拥塞与DDoS攻击。L. Feinstein等人[9]对源地址进行熵运算，根据源地址的变化范围是否明显增大或减小判断是否发生了攻击，该方法不仅未能区分正常网络的拥塞情况与DDoS攻击，而且也不能检测出具有相同伪造源地址的数据包攻击。

3 SYN Flooding攻击特征分析

为了及时检测出攻击，需要了解SYN Flooding攻击发生时与正常状态下网络行为的区别。正常情况下，如果主机A向主机B发送一个SYN包请求建立TCP连接，那么主机B接到该请求后会向主机A发送一个SYN+ACK包，最后主机A再向主机B发送一个ACK包，此时3次握手连接成功，可以进行数据传输了。当攻击发生时，网络中将会出现大量的SYN包，受害主机B的连接队列资源很快被耗尽，从而不能及时发送SYN+ACK包。就算B发送了SYN+ACK包，由于主机A假冒了主机C的源地址，A和C都不可能回送ACK包。此时，网络中SYN包的数量远远大于ACK包的数量(如图1所示)。但是，当正常情况下网络出现拥塞时，TCP通信也不畅通，同样可能出现SYN包数量大于ACK包数量或者SYN包到达速率发生变化的情况。因此，仅仅根据SYN包和SYN+ACK包、ACK包的对称关系来判断是否发生攻击，或者仅仅依靠某段时间内SYN包的到达率来判断攻击发生是不够的。

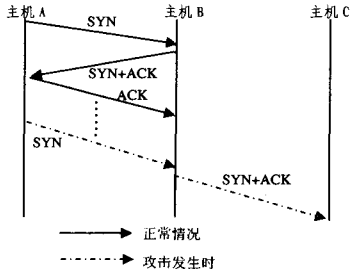

图1 正常与非正常三次握手连接示意图

DDoS攻击往往采取伪造攻击包源地址的方法。攻击发生与正常情况下网络拥塞的一个主要区别就在于网络中是否出现了大量的伪造源地址。攻击发生时会出现大量以前从未出现过的源地址，而网络在正常拥塞的情况下出现的源地址大部分都是以前出现过的[10]。另外，DDoS攻击一般借助工具产生，这些工具伪造攻击包的源IP地址，而且一个伪造的源IP被多次使用，即向目标发出多个源IP相同的攻击包。但是，现有的根据伪造地址检测攻击的方法以被伪造的IP出现的个数为检测依据。虽然当大量的攻击包出现时，也能检测到攻击。但显然不能做到快速有效地检测。因此，为了区分正常网络中的拥塞和DDoS，同时考虑对重复出现的攻击包的处理，尽早地检测到攻击，我们根据网络中是否多次出现了大量新的、伪造的源IP地址来判断是否发生了SYN Flooding攻击，而不仅仅根据伪造的新IP地址的个数。

4 基于Bloom Filter的信息提取

Bloom Filter最早诞生于1970年，原本是依次测试一系列信息是否属于给定的信息集以确定其成员资格，1980年开始用于降低不同文件对磁盘访问的速度以及其它方面，现在被扩展应用于DDoS攻击检测中。在Bloom Filter结构中，一个二维向量表由k级向量组成，每一级向量对应于一个Hash函数，并且包含m位向量。每一位向量包含一个计数器$C_{i,j}$ ($1\leqslant i\leqslant k, 1\leqslant j\leqslant m$)，如果被击中则加1。由于存在Hash冲突，该结构存在一定程度的误差，这种误差可以通过调整Hash函数和m值来减小。使用Bloom Filter能够节约存储空间，也便于对源地址信息进行下一步处理，使用Hash函数能够使得数据在向量表中的分布更分散。我们的算法基于Bloom Filter数据结构，并对其做了一定的改进，用一个Hash函数对应于两级向量。

DDoS攻击发生时，会出现大量源地址被伪造的数据包，即源IP中出现大量的、以前从未出现过的新地址。这些包的出现势必使与网络地址相关的统计特性发生变化。基于这种特性，通过合适的算法定能检测出攻击的存在。首先，说明如何提取被伪造的源IP出现的次数。使用Hash函数将数据包的源IP映射到Bloom Filter结构中，每一个源IP对应于Bloom Filter中的一个计数器。为了确定一个源IP地址是否为新出现的被伪造的地址，我们改进了Bloom Filter结构，用一个Hash函数对应于两级向量T_A和T_S(如图2所示)。改进后的结构能够准确、方便地统计伪造源地址出现的次数。Hash函数用于计算数据包源IP地址对应的向量计数器C_j的位置j ($1\leqslant j\leqslant m$)。T_A和T_S均为一维m位向量，计数器的初始值均为0。T_A用来保存经过路由器的数据包的ACK标志位的信息，其值只可能为0或1。若T_A中的某计数器值为0，则说明与该计数器相对应的源地址尚未成功建立连接；若T_A中的某计数器值为1，则说明与该计数器相对应的源地址已经成功建立了连接。T_S用来保存最近一段时间经过路由器的、源地址疑似被伪造的数据包的数量的信息。若T_S中的某计数器为0，则说明与该计数器对应的源IP还没有出现伪造包的情况；若T_S中的某计数器值为非0，则说明与其对应的源IP地址的数据包尚未被确认，该IP有可能是被伪造的。

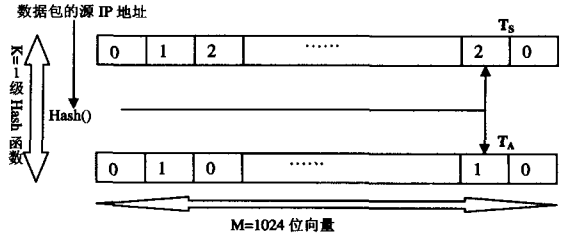

图2 算法中使用的Bloom Filter结构

当一个数据包到来时，基于Bloom Filter的信息提取算法描述如下。

步骤1 提取数据包的源IP地址，进行Hash运算并找出与之对应的计数器C_j ($1\leqslant j\leqslant m$)；

步骤2　提取该数据包的ACK标志位的信息，计算并重置 T_A 中计数器 C_j 的值：

• 若该数据包的ACK标志位为0，且 T_A 中 C_j 原本为0，则 T_A 中 C_j 仍为0。这说明此源IP还没有成功建立过连接，可能是合法的新IP，也可能是伪造的新IP；

• 若该数据包的ACK标志位为0，且 T_A 中 C_j 原本为1，则 T_A 中 C_j 仍为1。这说明该源地址曾经成功建立了3次握手连接，不属于新的源地址；

• 若该数据包的ACK标志位为1，则 T_A 中 C_j 置1，此时可能有两种情况：

① 此数据包可能是在建立3次握手连接中的第3步，此时3次握手连接建立成功，说明该地址是合法的新源IP；

② 此数据包可能是数据传输过程中的包，由于TCP/IP协议规定传输数据之前必须先建立连接，所以，3次握手连接一定已经成功建立了，该地址不是新的源IP。

步骤3　计算并重置 T_S 中计数器 C_j 的值：

• 如果 T_A 中的计数器 C_j 为1，则 T_S 中 C_j 置0，因为 T_A 中 C_j 为1说明对应的IP地址是合法的，而我们只需要统计新的可疑的源地址出现的次数，已经确定的正确的源地址不予考虑；

• 如果 T_A 中 C_j 为0，则 T_S 中 C_j 加上 a（$a>0$，用于标识某伪造源IP地址出现了一次）。因为 T_A 中 C_j 为0说明对应的IP地址尚未成功建立3次握手连接，可能是伪造的源地址，需要进行统计。

在正常情况下，3次握手是能够成功的。因此，T_S 中 m 个计数器基本都为0（除非Hash函数引起冲突，出现了脏计数器）。即使在正常网络发生拥塞时，3次握手也不能完全成功，但是并不会出现伪造的源地址。因为拥塞发生前很多地址都应该出现过，并已成功建立过连接，这些地址对应的 T_A 中的向量计数器为1。因此，T_S 中的 m 个计数器仍然基本都为0。当SYN Flooding攻击发生时，网络中会出现大量以前从未出现的源地址，这些地址对应的 T_A 中 C_j 为0，而 T_S 中向量计数器会出现很多非0值，即发生了异常。又由于网络中不可避免地偶尔会发生错误，因此我们将 T_S 周期性地重置，每隔时间 t，T_S 中的向量计数器按 $C_t=(1-a)C_{t-1}(a>0)$ 进行重置。由于 T_A 中的计数器用于记录源IP地址的连接历史信息，这些历史信息需要保留下来用作后续的判断，所以 T_A 不参与重置。

我们的检测方法属于源端检测方法。在一个有限的源端局域网范围内，IP地址的变化范围是有限的，而攻击发生时伪造的源IP的范围会远远大于局域网内IP地址的变化范围。因此，伪造的源地址被误认为是正确的源地址的概率是很小的。以一个C类网络为例，一个伪造的源地址被误认为是正确的源地址的概率为：$P=\frac{1}{n^k}$，设 $n=256$，$k=3$，则 $P\approx 5.96*10^{-8}$，这样的误报率是很低的。正确的源地址互相冲突或者伪造的源地址互相冲突都是可以接受的。因为，正确的源地址由于ACK标志位均为1，所以出现的次数不进行累计。而伪造源地址的包由于ACK标志位均为0，所以这些伪造地址出现的次数要进行累计。由于目前存在的一些DDoS攻击工具（如TFN等），某个伪造的源IP地址会重复出现在多条攻击数据包中，此时一个伪造的源地址对应多个攻击数据包。我们需要结合伪造的地址以及大量的数据包，快速而有效地检测攻击发生，并且算法需要避免针对不同攻击工具产生的检测结果的误差。我们统计单位时间内伪造的源地址出现的总次数，而非总个数，从而将相同源IP地址的多个伪造攻击包看成是多个攻击。

5　基于变化点计算的SYN Flooding检测方法

变化点检测源于统计学，用于判定被观察的序列是不是保持一致。如果不是，则确定发生变化的时刻，其中的CUSUM算法在参数模型已知的情况下是渐进最优的，且适合于检测变化比较小的序列。

基于Bloom Filter结构得到统计序列之后，需要对其进行检测，以判断是否发生了攻击。本文取 T_S 中各计数器值的累加结果 $\sum_{j=1}^{1024}C_j$ 作检测序列，每隔时间间隔 t，得到 T_S 中向量计数器新的值。由此，可以得到一个根据时间间隔 t 变化的序列 $\{\Delta_i, i=1,2,3,\cdots\}$，其中 $\Delta_i=\sum_{j=1}^{1024}C_j$。正常情况下，$\{\Delta_i, i=1,2,3,\cdots\}$ 是一个接近于零的序列，而发生攻击时 Δ_i 会大大增加。假设正常情况下序列的均值 $E(\Delta_i)=\alpha$，设当 $t=n$ 时发生攻击，攻击进行时 $E(\Delta_i)=\alpha+h$。整个过程可以表示为 $\Delta_i=\alpha+\xi_i I(i<n)+(h+\eta_i)I(i\geqslant n)$。其中，$\{\xi_i, i=1,2,3\cdots\}$ 和 $\{\eta_i, i=1,2,3\cdots\}$ 是均值为0的序列。$I(H)$ 是这样的函数：当条件 H 成立时 $I(H)=1$，当条件 H 不成立时 $I(H)=0$。

网络中数据包的变化情况很复杂，很难用固定的模型描述。无参数CUSUM算法不含参数，并且与模型无关，计算量也很小，使用时不会给路由器带来很大负担。因此，我们采用无参数的CUSUM算法对统计序列进行变化点检测。无参数CUSUM算法要求检测的序列在正常情况下具有负的均值，变化发生后具有正的均值。因此，我们构造一个新的序列 $\{Z_i, i=1,2,3,\cdots\}$，其中 $Z_i=\Delta_i-\beta$，$\alpha<\beta<\alpha+h$。参数 β 是常数，用来帮助产生均值为负的随机序列 $\{Z_i, i=1,2,3,\cdots\}$。这样，在正常情况下 $\{Z_i, i=1,2,3,\cdots\}$ 均为负值。当攻击发生时，由于出现大量伪造的源地址，T_S 中各计数器值的累加结果将迅速增大并且超过 β，则 Z_i 会迅速变大并且为正值，符合CUSUM计算的条件。

为便于计算，降低在线检测的开销，我们使用无参数CUSUM算法的递归版本：

$$y_i=(y_{i-1}+Z_i)^+$$

$$y_0=0$$

其中，x^+ 表示如果 $x>0$，$x^+=x$，否则 $x^+=0$。当攻击发生时，y_i 将以很快的速度增大。

$$d_N(y_i)=\begin{cases}0, & y_i\leqslant N\\ 1, & y_i>N\end{cases}$$

其中，N 是检测的阈值，$d_N(y_i)=0$ 表示正常，而一旦发现 $d_N(y_i)=1$，则说明发生了SYN Flooding攻击。

通过选择最佳参数 β 和 N 可以实现低误报率和短检测时间。β 用来将 $\{\Delta_i, i=1,2,3,\cdots\}$ 偏移到 $\{Z_i, i=1,2,3,\cdots\}$，在正常运转中它具有一个负的平均值。选择的 β 越大，在 $\{Z_i, i=1,2,3,\cdots\}$ 中出现正值的可能性就越小，因此测试统计 y_i 累积到一个较大的值来发现攻击的可能性越小。N 是 y_i 的攻击门限，N 越大，误报率越低，但检测时间越长。

6　实验结果及分析

我们采用DARPA于2000年提供的攻击场景测试集LLDOS1.0[11]进行检测。m 值的选取依赖于局域网IP地址变化的范围。m 值太小则使用Bloom Filter提取信息时冲突变大，m 值太大则浪费存储空间。实验中，每一级的向量组均为 $m=1024$ 位，采样间隔为200个数据包。正常情况下，Δ_i 随时间没有特别明显的变化。如图3所示，正常情况时的最

大值为5，因此取$\beta=5$。在检测过程中，参数a分别被设置为1，2和3。图4给出了检测结果。从结果可以看出，当攻击发生时，y_i的值迅速增大，明显区别于正常情况。

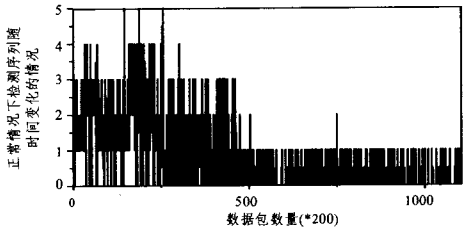

图3 正常情况下Δ_i随时间的变化

文献[3]直接计算未成功匹配的数据包的数量，再用CUSUM算法检测攻击。该方法没有对数据包的信息进行提取，只是简单地从数量上进行操作，一方面阈值很难确定，另一方面加上正常情况下网络中也会有不能成功建立连接的情况，会导致检测结果不准确。图5是利用DARPA数据集验证该方法的结果。从中可以看出，该方法的检测率和误报率很难同时达到最优。图6表明本文算法的检测率和误报率要优于此方法，通过实验得到本文算法的检测率为93.89%，误报率为0.43%。

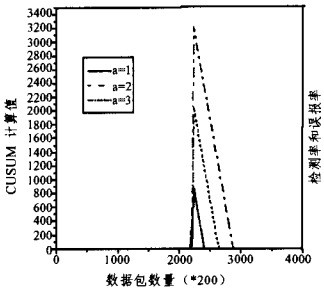

图4 本文算法检测结果

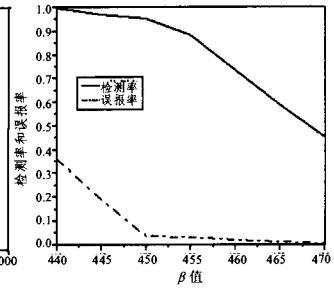

图5 文献[3]方法在不同β值下的检测率和误报率

文献[4]也是使用Hash函数将数据包的源地址映射到Bloom Filter结构中。其方法是：当第一次握手的数据包到达时，将该包的源地址添加到存储空间中。当第3次握手的数据包到达时，再将该地址删去，然后用CUSUM算法对未成功建立连接的数据包的数量进行运算。该算法使用了一个4级Bloom Filter向量组，即使用4个不同的Hash函数存储数据包的摘要，添加和删除过程各计算一遍。算法的计算和存储需要较大的时间和空间开销。另外，该算法不保留合法的历史源IP地址信息，因而无法区分正常的网络拥塞和真正的DDoS攻击。从对数据包源地址的存储过程明显看出，本文算法的时间和空间复杂度要比文献[4]低。此外，本文算法在检测过程中不会将正常拥塞出现的不完整连接看成是伪造源IP的DDoS攻击，误报率低。算法的误差仅在攻击数据包与正常数据包出现冲突时才可能产生。而攻击包与正常包会在两种情况下冲突：情况1，同一个被伪造的IP源地址被多次使用，则伪造的包被多次累加，计算结果完全正确；情况2，伪造的源IP与合法的源IP经过Hash函数计算后，击中同一个位置。此时，若正常的IP先出现，伪造的IP后出现，则不影响结果；若伪造的IP先出现，而正常的IP后出现，则会产生误判。而根据前文分析可知，这种误判的可能性是很小的。因此，本文算法累加伪造源IP出现的总次数，即使Hash运算可能存在冲突，对算法的影响也很小。图6也说明了本文方法在实验中具有较好的检测效果和很低的误报率。

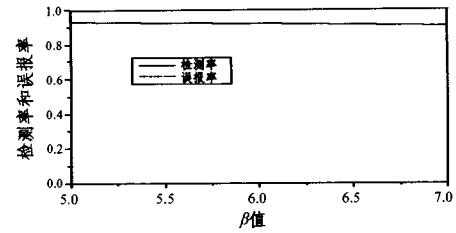

图6 本文方法在不同β值下的检测率和误报率

结束语 在检测DDoS攻击的过程中，源端检测一直是最理想的方法，能够在攻击数据流到达受害端之前检测出攻击，能最大程度地降低攻击数据流对网络的危害。但是源端检测的缺点是攻击数据流量很小，不易检测。而本文提出了一种有效的、轻量级的SYN Flooding攻击检测方法，可以在只出现少量伪造数据包的情况下准确发现攻击，适用于源端检测，也为监控攻击数据包的来源以及通知受害端进行防御赢得了宝贵的时间。另外，在攻击流量较小时进行检测及监控，对网络的性能也不会有太大的影响。由于正常的网络拥塞和DDoS攻击具有很多共同的特点，如何快速准确的区分攻击和正常的网络拥塞一直是个难题。而本文算法不会对正常的网络拥塞产生误报，具有一定的现实意义。

现有的DDoS攻击检测算法一般都是在攻击已经发生的情况下才能检测成功，而此时的攻击流量或多或少都已经对目标主机或目标网络造成了危害。并且绝大多数时候，即使在检测成功的情况下，仍然无法区分正常数据流和攻击数据流。因此，结合对DDoS攻击检测的研究，还需要进一步做好如何在攻击尚未成熟前进行正确预警的研究工作，以及做好在攻击被检测出来以后，如何准确地过滤攻击包，阻止后续攻击的研究工作。

参考文献

[1] Moore D, Volker G, Savage S. Inferring Internet Denial-of-Service Activity[C]// Proceeding of the 2001 USENIX Security Symposium. Aug. 2001:9-22

[2] Chan E, Chan H, Chan K, et al. IDR: An intrusion detection router for defending against distributed denial of-service (DDoS) attacks// Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks 2004 (ISPAN'04). Hong Kong, 2004:581-586

[3] Wang Haining, Zhang Danlu, Shin Kang G. SYN-dog: Sniffing SYN Flooding Sources[C]// Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02). 2002:471-478

[4] 陈伟，何炎祥，彭文灵. 一种轻量级的拒绝服务攻击检测方法[J]. 计算机学报，2006，29(8)：1392-1400

[5] Ohsita Yuichi, Ata S, Murata M. Deployable Overlay Network for Defense Against Distributed SYN Flood Attacks[C]. Computer Communications and Networks, 2005:407-412

[6] Zuquete A. Improving the functionality of SYN Cookies// Proceedings of 6th IFIP Communications and Multimedia Security Conference. 2002:57-77

[7] Lemon J. Resisting SYN flooding DoS attacks with a SYN cache// Proceedings of USENIX BSDCon'2002. Feb. 2002:89-98

[8] Ohsita Y, Ata S, Murata M. Detecting distributed Denial-of-Service attacks by analyzing TCP SYN packets statistically// Proceedings of IEEE Globecom. 2004, 4:2043-2049

[9] Feinstein L, Schnackenberg D, Balupari R, et al. Statistical Approaches to DDoS Attack Detection and Response. IEEE, 2003:303-314

[10] Jung Jaeyeon, Krishnamurthy B, Rabinovich M. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. Honolulu, Hawaii, USA, 2002

[11] http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.htm