一种基于Multistage Bloom Filter的入侵检测流量筛选算法

刘贺[1]，龚俭[1,2]，杨望[1,2]

（1.计算机科学与工程学院，东南大学，南京，211189；2.江苏省计算机网络技术重点实验室，东南大学，南京，211189）

摘要：目前入侵检测系统的性能是高速网络环境中入侵检测的一个瓶颈。存在诸多的方法如改进硬件、采用分布式系统等来提高入侵检测系统的性能。其中一个重要的方法是给入侵检测系统加入流量筛选模块。本文在前人的基础上，对其流量筛选算法进行分析并给予改进，改进之后的算法较原来的算法有更好的性能和功能，更适合作为高速网络环境中入侵检测系统的流量筛选算法。

关键词：入侵检测；Multistage Bloom Filter；流量筛选算法

An Intrusion Detection System Traffic Filter Algorithm Based on Multistage Bloom Filter

Liu He[1]，Gong Jian[1,2]，Yang Wang[1,2]

(1.School of Computer Science & Engineering, Southeast University, Nanjing, 211189;

2. Jiangsu Provincial Key Laboratory of Computer Network Technology, Southeast University, Nanjing, 211189)

Abstract: At present, performance of Intrusion Detection System (IDS) is the bottleneck of high-speed network's intrusion detection. There are many ways to improve its performance such as improving hardware, using distributed systems, etc. Among these, an important way is to add a traffic filter module to IDS. Based on a previous study, this paper improves its traffic filter algorithm which shows better performance and functionality than the original one. Thus, the improved algorithm is more suitable for high-speed network's intrusion detection.

Keywords: Intrusion Detection; Multistage Bloom Filter; Traffic Filter Algorithm

1 引言

近年来网络入侵事件频繁发生，如分布式拒绝服务攻击（DDoS）、蠕虫（Worms）传播、端口扫描（Port Scan）等，有的甚至造成了巨大的经济损失。随着互联网逐渐渗透到人类社会各个方面，设计准确、快速的网络入侵检测方法已经成为网络研究中的重要课题。但是入侵检测系统的性能已经是高速网络环境中进行入侵检测的瓶颈。目前使用的最广泛的开源 Snort 在 2.4GHz 双核 CPU、2G 内存配置下的极限处理能力约30~40kpps，在实际网络流量下处理能力<200Mbps，Bro因其技术架构的复杂性其极限处理能力甚至不到8kpps，实际网络流量下处理能力小于60Mbps[1]。高速网络环境中的入侵检测还存在诸多的问题。

为了实现高速网络环境下的入侵检测，可以从多个方面实施，比如改进硬件设施，采用分布式系统等，但是目前很少有人从流量筛选角度来考虑。华东北地区网络中心为了改进其入侵检测系统MONSTER，使之能够接入万兆以太网中进行实时检测，采用流量筛选的方法，给入侵检测系统加入一个流量筛选子系统，提前过滤掉一部分对入侵检测无用的流量，减轻入侵检测系统的负担。参考文献[1]中提出了一些适用于高速网络环境中的流量筛选算法。本文继承前者，对它其中的部分算法进

作者简介：刘贺，（1987-），男，硕士研究生，E-mail: hliu@njnet.edu.cn；龚俭，（1957-），男，教授，博导，E-mail: jgong@njnet.edu.cn；杨望，（1979-）男，讲师，E-mail: wyang@njnet.edu.cn.

行分析，并给于改进。针对的原有算法主要是其流抽样保持算法FSH（Flow Sampling & Hold）和带反馈指导的流长流量选择方法FFS（FeedBack FlowLength Sampling）[1]。

本文第2节介绍FSH算法和FFS算法，并分析其存在的问题；第3节针对FSH和FFS算法存在的问题，给予改进，并详细介绍了改进之后的算法；第4节对改进的算法的冲突和时间空间复杂度进行了分析，最后经过实验验证改进过后的算法确实能够比原有算法更加的优越；第5节对全文进行总结。

2 FSH 算法和 FFS 算法

2.1 FSH算法介绍

在本文中，流的定义是指具有相同五元组（源IP地址、目的IP地址、源端口、目的端口及协议类型）的报文的集合。流的长度是指流中包含的报文个数。

DDoS、Port Scan、Network Scan和Worms这几种攻击的流量特征主要表现为短时间内连接建立数目多，都具有某种聚类特征。即这些恶意攻击发起的流多，但是很可能每个流的报文数目很小。此时采用测度为流数目的Sampling & Hold方法即可保证包含流数目多的攻击者/被攻击者/服务的被抽样概率足够大，这种抽样方法称为流抽样保持算法FSH（Flow Sampling & Hold）。其大致思想是，若某个报文被抽样，则与其同属一个流的报文全部被抽样。算法如下：

① 根据报文的五元组计算出他的HASH值，判断该报文所属的流是否已抽样。若已抽样，更新流的信息，否则进行下步。

② 以某个概率抽样该报文，如果该报文被抽样，则在HASH表中新建一项，否则进行后面的处理。

2.2 FFS算法介绍

带反馈指导的流长选择算法（FFS）主要是针对小规模攻击，思想是高抽样比短流，低抽样长流，超长流不抽样。其算法如下：

① 每个到达报文通过多级Bloom Filter以计算报文所属流的长度。

② 抽样比取取决于流长特征。采用随机报文抽样方法对短流以高抽样比*pshort*抽取报文，对长流以低抽样比*plong*抽取报文，对超长流不再抽样。流长小于等于10的报文认为是短流，大于1000的流认为是超长流，否则属于长流。

③ 每个时间片结束都统计计算*pshort*和*plong*。作为下一时间片的抽样率。

④ 反馈指导策略。对已经发现攻击的流打标记，以100%抽样率抽取已经发现前导攻击的攻击流。

其中，Bloom Filter是一种常用的基于HASH的匹配方法，它可以使用较少的存储空间来判断对象是否存在，主要用于数据库应用中，随着网络测量中数据量的飞速增长和有限的计算空间形成强烈对比，该技术在网络中被逐步引入并得到广泛的应用。

Multistage Bloom Filter是Bloom Filter的一种变形。它采用多级过滤器。每级过滤器有多个表项组成，每个表项包含一个计数器，开始时计数器清零。当一个数据分组到来时，通过HASH函数对流关键字做HASH计算，将其映射到一个表项，并更新对应表项的计数器。同一个流的所有数据包都会映射到同一个表项。当一个数据包到达时，它通过HASH函数会映射到各级过滤器的某个表项，这些表项的最小值作为这个数据包所在的流的长度。如果最小值是0，说明这个报文是其所属流的第一个报文，同时这个流是一个新流。如下图：

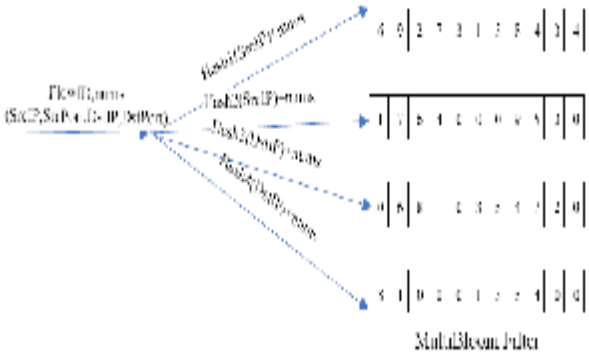

图 1　Multistage Bloom Fiter

2.3 两种算法的分析

多点流是在一段时间内与多个 IP 地址通信的节点。使用 FSH 算法能够保证在一个节点同时有多个流的情况下能够被抽样的概率足够的大。利用

FSH 算法还减少了内存的使用率，避免对每个流都在内存中都建立一个表项。但是 FSH 算法也有它的副作用，FSH 算法会抽中大量的长流，因为这些长流含有的报文数目比较多，那么这些长流被抽中的概率就要比短流高。大量的视频流、多媒体流等长流将会被送入入侵检测系统，对入侵检测系统而言，这却是一个无用的负担。使用 FSH 算法抽样的流，可能不是一个完整的流，而是前面若干个报文被丢弃的流。一个完整的流是一个完整的语义信息，能够更好的有助于入侵检测系统检测这个流是否含有攻击。

FFS 算法能够保证短流被抽中的概率比较大。但是 FFS 算法与 FSH 算法组合在一起，就会有某种功能上的冲突。当被 FFS 算法判定为超长流不再抽样的时候，这个流中的一个报文却被 FSH 算法抽中，然后这个流后继报文还是要被选择。

针对上面存在的问题，提出一种新的设计方案，该方案除了能够完成找出多点流和高抽样短流、低抽样长流、不抽样超长流的功能外，还要能够避免其存在的问题。

3 改进算法

新的算法结合 FSH 和 FFS 两种算法，将其进行修改。修改后的算法根据 Multistage Bloom Filter 来判断一个流是否是新流，如果是新流，则对其实行抽样，如果被抽中，则将其加入到流表中。后续到达的属于这个流的报文将全部被选择。如果不是新流，就按照 FFS 算法的思想（高抽样短流、低抽样长流和不抽样超长流）来抽样。这样就避免了原有 FSH 算法和 FFS 算法的相互冲突，并且被在流表中的流是一个完整的流。同时还可以设定，当流表中的流长度大于 1000 后不再选择其中的报文。

具体新的算法如下：

① 首先判断这个报文所属的流是否已被抽中，如果被抽中就选择这个报文。否则转（2）。

② 采用 Multistage Filter 技术来实时计算这个报文所属流的长度，如果有一个表项为 0，说明这个流是新流，转（3）；否则转（4）

③ 以事先确定的一个概率来抽样这个流，如果流被抽中，则就在流 HASH 表中新建一个表项。

④ 如果流的长度小于 10 就以 *pshort* 的概率抽样这个报文，如果大于 1000 就丢弃这个报文，否则就已 *plong* 的概率抽样这个报文。

算法的流程图如下：

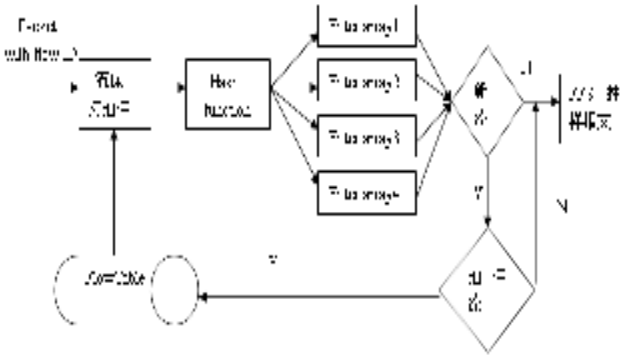

图 2 改进算法流程图

伪代码如下：

```
1.  Initialize
2.  FilterArra y_i ← 0(i=1,2,3,4)
3.  FlowTable ← 0
4.  each incoming packet pkt
5.  if(pkt.flowID in FlowTable)
6.      抽样这个报文，并更新流的信息
7.  else
8.      hash_value _i=H_i(pkt.flowID);(i=1,2,3,4)
9.      flowLength=min( hash_value _i);(i=1,2,3,4)
10.     if(flowLength==0)
11.         以pf（事先已定好的流抽样概率）的概率抽样这个流，如果抽中在FlowTable中新加
    一项，否则 FilterArra y_i[hash_value _i]加1。
12.       else
13.           实施FFS 抽样，同时FilterArra y_i[hash_value _i]加1
14.       endif
15. endif
```

图 3 改进算法伪代码图

4 算法分析与验证

4.1 Multistage Bloom Filter冲突分析

在 Multistage Bloom Filter 中，可能有许多流会被映射到同一计数器，这会引起误正。如小的流被映射到包含大流的计数器；新流被映射到已存在流的计数器中。下面结合实际例子进行误正分析：假设是 OC48（2.5Gbps）的信道，假设报文平均长度为 500B，则信道中一分种内报文个数约等于

$3.75*10^7$，此时流的数量级一般在10^6左右。假设每级 Bloom Filter 采用 24 位的 HASH 函数，则平均一个计数器会被$10^6/2^{24}=0.0625$个不同的流命中，采用 4 级 Bloom Filter 的误判率要小于$0.0625^4\approx10^5$。

4.2 算法时间空间复杂度分析

假设输入有n个报文，算法先判断报文所属的流是否已被抽样，处理复杂度为O（n），接着采用四级Bloom Filter统计流长，每个报文计算4个HASH函数，取最小值为流长，流长判断的复杂度为O(4n)，如果是新流（流长为0），以流抽样率抽样这个流；如果流长小于10；以*Pshort*抽样；如果流长小于1000，以*Plong*抽样，时间复杂度都是O（n）。所以总的时间复杂度是O（n），与原来分离的两个算法时间复杂度相当，可以在线实时处理。

流表存储的是流的信息，一个流结构体包含开始时间、结束时间、五元组、流长和一个处理冲突的指针,总共需要36B。假设流抽样率为1/20，那么流表中流的数量级在10^5左右，所以流表总的存储容量约为4M。四个Bloom Filter数组约占$2^{24}*8=128$M。则整个算法所需要的空间大约为132M。一般服务器的内存都在4G以上，所以可行，并且新的算法与原有的两个算法耗费的内存相当。

4.3 实验结果及分析

数据一：在江苏省网边界路由器抓取500000报文，流个数为36881，平均流长约为13.5。原来两个算法和改进算法使用相同的参数测试数据：流抽样率为1/30，*pshort*=0.1，*plong*=0.05。测试数据如下表：

表1 测试数据一

算法	报文数	流数
原有两种算法	364759	3688
改进算法	13732	1212

原有两种算法实际的流抽样率是$3688/36881\approx1/10$，364759个报文被选中，占总报文的72.9%，平均流长为99。与测试的参数相比，这些数据说明，原有FSH和FFS算法在测试中选中了大量的长流，并且流抽样率误差很大。而新算法的实际流抽样率是$1212/36881\approx1/30$，抽中的报文数只有13732，占总报文的2.7%，平均的流的长度为约为11，相比较而言，抽中的短流比较多。

数据二：江苏省网边界路由器报文数10^7个，流总数为3351339，平均流长为3。原有算法和改进算法还是采用相同的参数：流抽样率为1/30，*pshort*=0.3，*plong*=0.05，经原有算法和改进算法测试后数据如下表：

表2 测试数据二

算法	报文数	流数
原有两种算法	7240328	132257
改进算法	286079	110839

原有两种算法的实际流抽样率约为 3.94%，选中的报文占总报文的 72.4%，平均流长约为 57。改进算法的实际流抽样率约为 3.301%，选中的报文占总报文的 2.86%，平均流长约为 2.6。

经过两组数据，对比原有两种算法与改进算法，可以得出：改进算法克服了原有算法会抽中大量长流的问题，减少选中的报文；并且从流抽样率中可以发现，改进算法的实际流抽样率与理论流抽样率很接近，这也说明了采用 Multistage Bloom Filter 的冲突很小，适合被算法的所使用。总的来说，改进这后的算法要比原有算法更适合作为高速网络环境中入侵检测的流量筛选算法。

5 结论

由于目前入侵检测系统的性能使其不足以在高速网络环境中部署，所以很多研究从硬件、分布式系统的角度考虑来加快入侵检测系统的性能。参考文献[1]提出的流量筛选方法是一个非常好的思路，剔除掉对入侵检测无用的流量，在减轻入侵检测系统负担的同时还能够保证检测的精度。本文在对其流量筛选算法研究的基础上，进行改进，改进后的算法，在完成相同功能的条件下，筛选结果更加有

利于入侵检测。该算法借助于 Multistage Bloom Filter 完成流抽样和高抽样短流、低抽样长流、超长流不抽样的功能，克服了原有 FSH 算法会抽样大量长流的副作用，也避免了原有 FSH 算法和 FFS 算法的冲突，流抽样率也更加准确。总得来说，改进的算法更适合作为高速网络环境下入侵检测系统的流量筛选算法。

参考文献

[1] 宁卓. 大规模网络中基于流量特征的入侵检测性能改进[D]. 东南大学: 华东北地区网络中心, 2010.

[2] 杨家海 吴建平 安常青. 互联网络测量理论与应用[M]. 北京: 人民邮电出版社, 2009.

[3] 周明中 龚俭 丁伟等. 基于 MGCBF 算法的长流信息统计[J]. 东南大学学报, 2006, 6(3):472-476

[4] 王洪波 程时端 林宇. 高速网络超连结主机检测中的流抽样算法研究[J]. 电子学报，2008, 36(4):809-818

[5] 程光 龚俭 丁伟等. 面向 IP 流测量的哈希算法研究[J]. 软件学报, 2005, 16(5):652-658