ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.3, March 2006, pp.434−444
DOI: 10.1360/jos170434

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

基于 Bloom Filter 的大规模异常 TCP 连接参数再现方法*

龚 俭[1,2], 彭艳兵[1,2+], 杨 望[1,2], 刘卫江[1,2]

[1](东南大学 计算机科学与工程系,江苏 南京 210096)

[2](江苏省计算机网络重点实验室,江苏 南京 210096)

Reconstructing the Parameter for Massive Abnormal TCP Connections with Bloom Filter

GONG Jian[1,2], PENG Yan-Bing[1,2+], YANG Wang[1,2], LIU Wei-Jiang[1,2]

[1](Department of Computer Science and Technology, Southeast University, Nanjing 210096, China)

[2](Jiangsu Provincial Key Laboratory of Computer Network Technology, Nanjing 210096, China)

+ Corresponding author: Phn: +86-25-83794000 ext 215, E-mail: ybpeng@njnet.edu.cn, http://www.njnet.edu.cn

Gong J, Peng YB, Yang W, Liu WJ. Reconstructing the parameter for massive abnormal TCP connections with Bloom Filter. *Journal of Software*, 2006,17(3):434−444. http://www.jos.org.cn/1000-9825/17/434.htm

Abstract: The large scaled TCP abnormal behavior, such as DDoS, scanning etc., can be detected by some metrics and their experimental values derived by the uniqueness of TCP connections. An algorithm named Bloom Filter Reproduction (BFR) is proposed to reconstruct the original parameters in large scaled TCP abnormal behaviors pithily by enhanced simple hash functions. Without maintaining the TCP information of 96bits' 5-tuple, the BFR algorithm can reconstruct the abnormal parameters such as IP address or their aggregation timely during the detection process. The experiments show that BFR can disclose several abnormal behaviors mixed in network traffic at the same time with high precision and low overhead.

Key words: massive abnormal connections; abnormality intrusion detecting; parameter recovery; Bloom Filter; TCP

摘 要: 提出由 TCP 连接的唯一性导出的 TCP 数量平衡性测度及其经验范围可用于检测 TCP 连接的大规模异常,如 DDoS、扫描等.使用带哈希增强算法的 Bloom Filter Reproduction (BFR)方法对 TCP 连接大规模异常的参数进行快速再现,如 IP 地址、端口的分布等,使得在检测过程中无须维护 TCP 五元组的信息.实验结果表明,该方法能够以较少的资源占用和较高的准确性来揭示网络流量中混杂的多种异常现象.

关键词: 大规模连接异常;异常入侵检测;参数恢复;Bloom Filter;TCP

中图法分类号: TP393 文献标识码: A

对于 TCP 的安全研究一直是热点问题,其中研究最多的大规模 TCP 连接异常是扫描和 DDoS 攻击.大规模扫描往往是蠕虫扩散、入侵和滥用的前奏,而 DDoS 攻击由于难以防范,也是互联网的重要威胁.因此,如何快速

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314804 (国家重点基础研究发展规划(973)); the Key Project of Chinese Ministry of Education of China under Grant No.105084 (教育部科学技术重点研究项目); the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No.BM2003201 (江苏省网络与信息安全重点实验室)

Received 2005-04-21; Accepted 2005-10-08

而准确地发现网络中出现的大规模扫描和 DDoS 攻击,是入侵检测的重要内容[1−5].本文从一个新的角度来构造大规模 TCP 连接异常的检测方法,利用 TCP 报文数量间的约束建立一种可用于大规模 TCP 连接异常检测的测度,并可将其运用于大规模扫描和 DDoS 攻击的检测.

在发生大规模连接异常时,TCP 的一些参数,如受害主机的详细信息、扫描发起方的 IP 等,在进行实时异常检测时,如果不维护五元组的信息往往难以确定,但维护五元组对于主干路由器的资源占用将是难以接受的开销,特别是在主干网络的带宽增长到 40Gbps 以上后,这项工作对主干路由器将是一个巨大的挑战.Bloom Filter[6]是最近在网络研究中比较热门的方法,在网络抽样[7]、还原[8]、流分布估计[9]里有着广泛的应用.使用 Bloom Filter 能够快速鉴别流的信息,并能把 TCP 流的信息维护从 96 比特的五元组空间映射到很短的哈希串所代表的空间,即用多个短标签来代替一个长标签,极大地减少了维护和遍历空间的计算资源开销.但是,如果希望得到原始长标签,如 TCP 五元组的信息,就对哈希算法的可逆性和哈希值的映射空间提出了很高的要求.Bloom Filter 把所有的哈希串映射到同一个哈希空间,将不可避免地导致不同哈希函数间的相互干扰.因此,如何选择合理的 Hash 函数和哈希方法,使得 Hash 后的数组能够反映出原来五元组的信息以及哈希冲突的解决是本文研究的主要内容.

本文第 1 节提出 TCP 报文数量的约束,通过分析异常 TCP 行为对 TCP 报文数量约束的影响,提出几个快速检测 TCP 连接大规模异常的测度,并确定了检测为大规模异常时 TCP 宏观平衡性参数的阈值下限.第 2 节分析 TCP 连接大规模异常的分布特征,提出了利用精心选择的 Bloom Filter 算法确定大规模异常发起者和受害者的 IP 地址分布特征的算法 Bloom Filter Reproduction(BFR),并给出了算法的时间复杂度.第 3 节给出一个利用 TCP 宏观平衡性测度定位 TCP 连接大规模异常的实例,给出了异常发生的时间点,并给出了对异常时间段进行的 Bloom Filter 分析.第 4 节是相关的研究工作.第 5 节是结论并介绍将来的工作.

1 大规模的连接异常发生时 TCP 报文数量的异常检测

由于对既有 TCP 连接的 Hijack 攻击是个别行为,在较大规模层次上发生有一定难度,因此,这里忽略 Hijack 攻击对 TCP 连接大规模异常行为的影响.由于 TCP 连接的唯一性,TCP 的五元组唯一标识了一个 TCP 流,而 TCP 建立连接和关闭连接的过程是唯一的,因而每个 TCP 连接的握手和关闭报文也是唯一的.为了研究无限时间网络里的 TCP 流,我们需要研究给定时间粒度内 TCP 流的报文数量关系.当合适的时间粒度内完整 TCP 的数量比不完整的 TCP 连接数量大的时候,可以认为该时间粒度内 TCP 的完整性可以得到保证.如果实际网络里 TCP 的连接接近于完整和正常,则其握手报文和关闭报文的数量应该是一致的,且存在着一个约束:

$$\Delta N_{SYN} \geq \Delta N_{SYN+ACK} \tag{1}$$

$$\Delta N_{SYN+ACK} \approx \Delta N_{FIN+ACK}/2 \tag{2}$$

$$\Delta N_{RST} \to 0 \tag{3}$$

ΔN_x 表示某充分大的时间粒度内某种标记为 x 的 TCP 报文数量;式(1)表示连接建立过程的 TCP 数量约束;式(2)表示连接全部过程的 TCP 数量的约束;式(3)表示连接结束过程的 TCP 数量约束.通过分析确定:1min.~10min.的时间粒度,对于这些约束的吻合程度是一个充分大的时间粒度.具体分析将另文描述.

TCP 的 SYN+ACK 报文比 SYN 报文携带了更多的连接信息,更接近于 TCP 流的数量.因此,我们定义 SYN+ACK 报文为 TCP 流的首报文.在分析 TCP 流的大规模异常时,可以以 SYN+ACK 的报文作为单位基准来确定 TCP 的一些异常行为的尺度.在宏观状态下,网络里 TCP 连接可能发生的异常情况可以归纳为如下几种,描述了 TCP 各类报文间的一种平衡约束,我们把它们称为 TCP 报文平衡性测度.

1.1 SYN报文与SYN+ACK报文数量不匹配

SYN 异常的典型标志是 SYN 报文和 SYN+ACK 报文的数量不匹配.例如,在路由循环或者 SYN 扫描没有回应时导致 $\Delta N_{SYN+ACK}/\Delta N_{SYN}$ 远小于 1.这里,定义 $\Delta N_{SYN+ACK}/\Delta N_{SYN}$ 为 TCP 握手效率(TCP shaking efficiency,简称 TSE),即

定义 **1**. $TSE=\Delta N_{SYN+ACK}/\Delta N_{SYN}$.

根据式(1),TSE 的值不可能大于 1.该值越靠近 1,就越接近健康状态.而当 TSE 超出[0.5,2]时,没有 SYN+ACK 报文应答的 SYN 报文数量与应答的 SYN+ACK 报文数量一样多,或者相反,可以看出:存在比作为基准的 SYN+ACK 的规模还大的异常存在.因此,TSE 判别异常的标准是:TSE 在[0.5,2]以外都是大规模异常的.

1.2 RST报文数量异常

RST 报文设计的用途就是响应异常到达报文[10],如主机崩溃或者因其他原因而出现错误的连接,或者用于拒绝非法的数据段、非法连接请求.正常情况下,RST 报文的数量比例应该远比 SYN 要小.

在连接建立以前,TCP 协议栈对异常到达含有 ACK 的非 RST 报文应答 RST_ONLY,不含 ACK 的应答 RST+ACK;在连接建立后,TCP 协议栈对于 TCP 数据传送过程中遇到的异常到来的报文一般也会以 RST+ACK 报文作为应答.在非 RST 扫描时,没有 ACK 标志位的报文会多一些,从而导致 RST+ACK 报文会多一些;在 DDoS 时,可能存在多种报文组合,使得 RST+ACK 报文和 RST_ONLY 报文的比例可能接近.显然,引起 RST 报文出现的情况非常复杂,引起 RST_ONLY 和 RST+ACK 报文的原因也有很大的不同,使得它们的行为也大不相同.对 RST 报文的分析是 TCP 异常行为检测的一个很主要的方面,这也是 AAAR 和 ANAR 测度定义的出发点.

这里,定义$\Delta N_{RST_ONLY}/\Delta N_{SYN+ACK}$为异常 ACK 报文到达率(abnormal ack arrival rate,简称 AAAR):

定义 **2**. $AAAR=\Delta N_{RST\_ONLY}/\Delta N_{SYN+ACK}$.

这里,定义$\Delta N_{RST+ACK}/\Delta N_{SYN+ACK}$为异常非 ACK 报文到达率(abnormal non-ack arrival rate,简称 ANAR).

定义 **3**. $ANAR=\Delta N_{RST+ACK}/\Delta N_{SYN+ACK}$.

理想条件下,AAAR 和 ANAR 都趋向于 0,以$\Delta N_{SYN+ACK}$瞬时值作为上限来衡量网络中 RST 应答的异常变化.AAAR 和 ANAR 的经验稳定范围为[0,0.4],且越小越好,超过这个范围时就肯定存在大规模异常.

由于 RST+ACK 报文和 RST_ONLY 报文常常有一些关联,比如在进行多种 TCP 报文 DDoS 攻击时,这两种报文数量会有一些共同的变化特征,但在扫描的过程中,根据扫描的种类而使得这两种报文的数量有很大的差异,因此建议在进行异常检测时,综合考虑上述指标.

1.3 FIN报文数量异常

这里,定义$\Delta N_{FIN+ACK}/2\Delta N_{SYN+ACK}$为链接关闭率(connection closing rate,简称 CCR),即

定义 **4**. $CCR=\Delta N_{FIN+ACK}/2\Delta N_{SYN+ACK}$.

根据式(3),FIN+ACK 的取值应该是 2 倍于 SYN 报文.也就是说,CCR 的取值应该在 1 附近.实验模拟的经验取值范围为[0.5,2],超过这个范围时存在大规模异常.

上述 TCP 报文平衡性测度给出了一个途径,能让我们确定 TCP 连接大规模异常发生的时间.但仅知道时间还不够,我们需要知道大规模异常的分布特点,便于我们采取相应的措施.

2 大规模 TCP 连接异常分布的特性与异常参数再现算法

2.1 大规模TCP连接的分布式特征

基于 TCP 的大规模扫描和 DDoS 有不同的分布式特征,在宏观层次上很容易区分.而掌握了这些特征,才能够找到有效的预防和阻止方法.下面对基于 TCP 的大规模扫描和 DDoS 的特征进行详细分析.

首先,扫描的发起者为了收到返回的报文,一般会采用真实的 IP 地址,会导致某类型 TCP 报文的源集中在少数一个 IP 或者多个 IP(分布式扫描或者多个扫描时间同时发生);为了避开防火墙的阻拦,扫描者一般会使用常用的端口以绕过简单防火墙的阻拦;对于主机扫描的特征是宿 IP 的分布比较均匀,而不服从重尾分布,源端口的分布也是分散的.

对于 DDoS 攻击,典型的特征是宿 IP 地址集中在受害机器上,端口也集中在常用的服务上;而源 IP 和源端口会比较均匀.另外,由于主机监听端口时,响应 SYN+ACK 报文的源 IP 是该主机,而如果大量的 SYN+ACK 报文涌向主机的监听端口,则可以肯定是一种 DoS 攻击.对于多 TCP 报文类型的 DDoS 攻击,各种报文的比例会比较接近,因此会对整体 TCP 报文的比例产生影响.

另外,DDoS 受害主机在它有能力进行应答时,会根据协议产生 RST 报文对异常报文进行回应.该应答的特征是源 IP 和源端口比较集中,而宿 IP 和宿端口则比较均匀,与大规模扫描有点相似.为了进行区分,可以察看 RST 是否有其他类型 TCP 报文与之对应,对应报文的五元组特征应该与 RST 报文同时出现.

所以,综上所述,对地址和端口简单地归纳和分析就可以得出源宿的分布规律,从而判断出异常类型,并且可以初步推断异常发生时的基本参数,如受害者的地址、端口或者异常发起者源 IP 和端口.但是,TCP 五元组共 96 比特,如果维护每个 TCP 流的五元组的信息,其内存开销或者算法的计算开销是难以忍受的,需要充分利用 Bloom Filter 的多个短标签表达长标签的能力.

Bloom Filter 是信息压缩的一种有效手段,在数据库[11,12]、分布式文件系统[13]、串匹配[14]、路由查找[15]等很多领域广泛使用,在网络测量和分析领域里也有报道[7−9].但是,Bloom Filter 的 Hash 函数很难还原出原来主机的特征,除非对 Hash 函数进行仔细选择,使得哈希函数能够直接携带原始信息.有了精心挑选的函数,我们就可以提出一种简易可行的异常参数再现算法,再现 IP 地址分布,过程如下:

1. 对 TCP 五元组进行 Counting Bloom Filter 变换;
2. 使用第 1 节定义的 TCP 相关测度的异常参数阈值确定异常发生的时间;
3. 如果发生了异常,对哈希存储空间按照 Bloom Filter 函数的计数器进行排序;
4. 对 Top10 进行综合分析,根据上述分布特征断定和验证异常的类型,再现异常的参数.

2.2 Bloom Filter的引入

Bloom Filter 算法使用多个短的哈希串来再现一个长字符串所代表的空间,其具体的工作原理可以参见文献[6,16],这里不再赘述.下面对 Bloom Filter 算法的 Hash 函数如何保持原来主机的信息进行论证.

为了能够直接反映原始五元组的信息,哈希函数的选择是尽可能简单地进行空间映射变换.我们可以考虑非常简单的比特位映射,即直接把 96 比特的五元组字符串按照比特划分为多个段,还可以按照源、宿把 Hash 函数分为两组.为了提高确定地址信息的精度,可以选择一些 Hash 函数,使得它们的比特位部分重迭,这样,在还原 IP 信息时可以利用这些重叠的比特位使得精度更高.为这些 Hash 函数和 Hash 数组准备独立的存储空间也是降低冲突、减少相互干扰的重要手段.在表 1 中,Hash 函数就是对原始比特串截取相应的短比特串,这样的 Hash 函数非常简单,也容易还原成原始的比特串.这样的 Hash 函数是非均匀的,但是 Bloom Filter 的各种特性仍然成立,只是其错误肯定率(FPR)有了变化,这里不再详细讨论.

Table 1 The Hash functions for bloom filter deployed in this paper

表 1 本文 Bloom Filter 所使用的 Hash 函数

Source Hash group	Computing steps	Destination Hash group	Computing steps
SIPH	Get high order 16 bits from source IP	DIPH	Get high order 16 bits from destination IP
SIPL	Get low order 16 bits from source IP	DIPL	Get low order 16 bits from destination IP
SIPM	Get middle 16 bits from source IP	DIPM	Get middle 16 bits from destination IP
SPORT	Get source port	DPORT	Get destination port

表 1 中 S 表示源,D 表示宿;H 表示取 IP 地址的高 16 比特,M 表示取 IP 地址的中间 16 比特,L 表示取 IP 地址的低 16 比特;PORT 表示端口.使用上述 Hash 函数对 SYN_ONLY,SYN+ACK,RST_ONLY,RST+ACK,FIN_ONLY,FIN+ACK 共 6 类 TCP 报文分别使用上述 Hash 函数.在利用第 1 节的方法检测到异常发生时,分析相应测度所涉及的 TCP 报文类型的源宿分布特征,即可快速定位异常的类型和参数.图 1 解释了本文使用 Counting Bloom Filter 进行信息还原的过程.

图 1 为每个 Hash 函数准备独立的存储空间消除了针对 Hash 函数间的相互干扰造成的内部冲突.图 2 中,$h_k(x)=y_k(k=1,2,3)$,图中的 $h_k^{-1}(y_k)$表示 y_k 的逆映射所代表的集合.在 y_k 所在空间的占用率比较小的情况下,可以多个短标签以很低的误差确定长标签 x 的存在[16].关键在于如何确定这些 y_k 是对应的同一个 x.

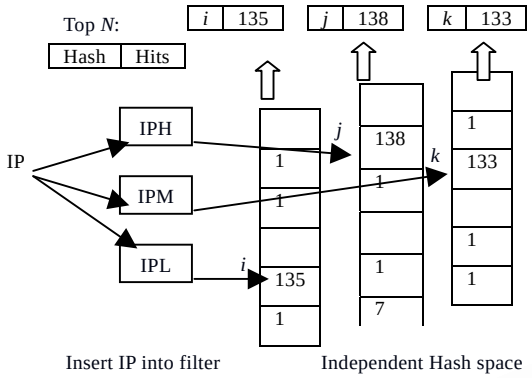

Fig.1 The storage and Hash enhanced bloom filter

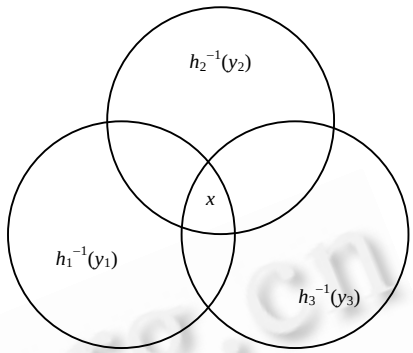

Fig.2 The effect of multi-Hash h_k (k=1,2,3)

图 1 Counting Bloom Filter 的存储和 Hash 函数改进 图 2 多个 Hash 函数 h_k(k=1,2,3)确定原始长标签 x

2.3 还原源串的过程

根据 Eddie Kohler 等人观察到的 IP 地址结构[17],活跃 IP 的分布是非常不均匀的重尾分布,相邻网段或者 IP 的活跃度的差异更大.因而,IPH,IPM 和 IPL 的 Top10 也是重尾分布的.如果多个 IP 的活跃度相差很小,可能是一种非正常的行为,可以推断是某种分布式的行为.因此,即使多个独立的长标签对应一个短标签也没有问题,因为它们的活跃度可能相差很大而不影响我们的分析结论,同时,也可以根据不同哈希函数对应的短标签的重叠来进行纠正.

如果结合表 1 中哈希函数间重迭的比特,我们可以用多个短标签 y_k 来代替长标签.同时,可以根据这些短标签唯一确定我们所关注的长标签所代表的对象 x,这也是 Bloom Filter Reproduction 算法的核心.由于我们选择的 Hash 算法能够逆向推导出原来长标签所代表的比特,因而这个方法能够确定原来的长标签的值.

对于本文的 Bloom Filter 方法,有两个约束:

约束 **1**. 一个长标签会分别在每个短标签里的映射有且仅有一次出现.

约束 **2**. IPH 里的低 8 位的值可能对应于多个 IPM 里高 8 位出现的值.同理,IPM 里的低 8 位的值可能对应于多个 IPL 里高 8 位出现的值.

约束 1 是一个重要约束,可以从 Hash 函数的单值性导出;约束 2 可以根据 IP 的聚类特性和约束 1 导出.

比如,IP 地址 A.B.C.D 在某个地方出现了 1 000 次,则约束 1 对于 IPH,IPM 和 IPL 而言,A,B,C,D 都得出现 1 000 次,如果 A.B.C.D 此时是活跃的,则 IPH,IPM 和 IPL 中相应的条目也应该是活跃的;约束 2 表明,由于存在多个长标签对应一个短标签的冲突,对于具有聚类特征的 IP 地址而言,如果不能找到与 A.B 相当活跃度的 B.C(即活跃度比 A.B 低很多),则表示 A.B 是一个活跃前缀,也就是活跃 IP 的聚类特征.

再举一个例子:表 2 列出了 WIDE[18]项目里的一个 Trace 文件——20050107 最初 5min.里 TCP 的 SYN_ONLY 报文的源 IP 各种 Hash 函数值,共 144 759 个有 SYN_ONLY 标记的 TCP 报文.该 Trace 的平均有效带宽为 17.99Mbps,链路带宽为 100Mbps[19].Hash 串的值我们按 8 比特分成两段,采用 *IP* 地址类似的表示方法.

Table 2 An instance from published trace (some data for top 10 ignored)

表 **2** 一个实际的例子(略去了部分 Top 10 的数据)

SIPH		SIPM		SIPL	
Location	Hits	Location	Hits	Location	Hits
3.81	57 297	81.132	57 297	132.197	57 297
4.101	56 888	101.144	56 886	144.252	56 884
213.110	15 620	110.133	15 620	133.36	15 623
129.255	2 701	84.83	1 640	83.198	1 639

根据约束 1,我们可以确定几个最活跃的 IP 地址分别是 3.81.132.197,4.101.144.252 和 213.110.133.36.而对

于前缀 129.255,在 SIPM 和 SIPL 里没有活跃 IP 与之对应,可能是因为这个网段的活跃 IP 活动比较分散使得我们 Top10 没有能够包含到.因而,该前缀对应的 IP 地址是活跃且分散的,即该前缀就是该 IP 地址的聚类特征.由于 IPH,IPM 和 IPL 的活跃度服从重尾分布,Top *N* 的分析即可保证我们得到的聚类信息是原始比特串的主成分.下面描述从 Top *N* 里还原原始比特串的分布特征信息.

2.4 数据挖掘算法

根据约束 1、约束 2 和 IP 地址的聚类特征可以推导出一个判断 IP 分布特征的算法,这个算法在满足异常条件时,能够返回占主导成分的活跃前缀包含的 IP 地址的聚类特征,或者返回最活跃的主机的 IP 地址.每个 Top *N* 由两个值构成:一个保存 Hash 串的值;一个保存该 Hash 函串的命中次数 hits.对原始 Hash 串数组的 hits 进行排序时,需要带着该 Hash 数组的下标,即该 Hash 串的值.它携带了原始 IP 的部分信息,使得我们能够还原原来的 IP 地址.

```
struct Top for Top N of IPH/IPM/IPL
    hash: 16bit, divided into to part high8bit and low8bit;
    hits: 32bit;
end struct

get total_hit for arrived packets;
get hash from BloomFilter;
get Top N by hits for IPH, IPM and IPL with original hash value;
for each TOP10 in IPH
    if IPH.hits>total_hit/N
        get IPH.hash.low8bit;
        check it in IPM.hash.high8bit;
        if sum (IPM.hits with the same 8bit in IPM.hash.low8bit)<total_hit/N
            return IPH is an active prefix;
        else
            for each IPM.high8bit with the same 8bit do
            get IPM.hash.low8bit
            check it in IPL.hash.high8bit
            if sum (IPL.hits with the same 8bit in IPM.hash.low8bit)<total_hit/N;
                return merg_as_prefix (IPH,IPM) is an active prefix;
            else
                return merge_to_IP (IPH,IPM,IPL) and IPL.hits/total_hit;
            fi;
            endfor;
        fi;
    fi;
endfor
```

将这个算法分别用于源 IP 地址和宿 IP 地址的哈希函数集,即可获得此时网络里某种 TCP 报文的活跃情况.端口信息也可以按照上面重叠比特的方法来确定.但是,因为 IP 地址比端口更重要,为了节约资源,我们没有对端口进行类似的计算.但是,端口的 Top10 分析往往能够给我们一些佐证,也可以使用简单的方法归纳出来,不在这里给出.我们把本节提出的参数再现算法称为 BFR(bloom filter representation).

对 BFR 算法的结果综合分析,得到各种 TCP 报文的活动规律,结合第 2.1 节的分析即可判断出是否发生了

大规模异常,如扫描和 DoS 攻击.

2.5 算法的复杂性分析

维护 32 比特空间和维护 16 比特空间的代价差别是显而易见的,因而维护 16 比特空间是首选.本文使用的 Bloom Filter 算法使用的 16 比特 Hash 函数,对 Hash 空间的访问开销为 $O(1)$;算法的主要计算开销在于排序找出 Top N,Top N 的计算方法有很多快捷的方法来实现[20,21].16 比特串表示的空间大小为 $M=2^{16}$,算法的空间复杂度为 $O(M+N)$,计算复杂度为 $O(M+N\log N)$.由于 N(本文取 $N=10$)比 M 小很多,可以认为算法的时间和空间复杂度为 $O(M)$.

3 基于 BFR 的 TCP 连接大规模异常检测

3.1 基本方法

结合第 1 节的 TCP 连接大规模异常的时间判断法则和第 2 节的 BFR 算法可以方便地检测 DoS/DDoS 攻击、大规模扫描、蠕虫的扩散等基于 TCP 报文的大规模异常行为.

按照时间粒度对网络里各种 TCP 报文的数量进行统计,计算出相应的测度,与经验范围比较即可得到是否存在着 TCP 连接的大规模异常.同时,为各种 TCP 报文维护一个相应的 Bloom Filter 数据结构.在检测到大规模异常后,对相应报文的 Bloom Filer 数据结构进行 BFR 分析,得到各种不同的分布式特征,综合分析即可得到大规模异常的类型是 DdoS,还是扫描,还是多种大规模异常混合.

对于不同的 TCP 连接大规模异常,它的行为总是异于正常的 TCP 交互.比如:基于 SYN-SYN_ACK 报文交互的 DDoS 攻击与 SYN-SYN_ACK-RST 报文交互的 DDoS 攻击相比,RST 的聚类特征与 SYN 报文相似,而 SYN 报文与 SYN_ACK 的聚类特征刚好相反;对于基于多报文的带有链路消耗攻击特征的 DDoS 攻击,各种异常的报文的聚类特征都类似;基于 SYN-SYN_ACK-RST 的 DDoS 攻击与基于 SYN-SYN_ACK 的大规模扫描的源和宿 IP 地址的分布刚好特征相反.

上述分布式特征即可作为进一步保护和预防措施中的客体.

3.2 实例:用TCP宏观平衡性测度的异常阈值对异常发生的时间定位

为了验证上述提出的异常参数确定算法的有效性,本文使用了 NLANR 的 Trace[22].该 Trace 采集于 2002 年 5 月 19 日~24 日,在 Bell 实验室的网络边界上;报文格式为 Dag3.2E,链路带宽 9Mbps,本地主机 400 人,服务器在外面,双向对称路由,平均有效带宽为 1.66Mbps.本文进行报文统计的时间粒度Δt 为 10min.,观察时间从 19 日 16:00 开始,取一天的数据.

图 3 列出了该 Trace 的 TCP 主要控制报文随时间变化的曲线.从图 3 中可以看出,SYN_ONLY 报文在 19 日 19:30(A 点)和 20 日 1:00~2:00(B 点)存在两个尖峰,其中后面一个尖峰还伴有 rst_ONLY 报文的小尖峰.A 尖峰表明存在 SYN_ONLY 报文异常,但是其他报文曲线没有响应,表明可能是一次扫描;B 存在 RST_ONLY 报文伴随,可能是一次 DDoS 攻击,当然可能是两种扫描,需要对此进行进一步的探索.B 尖峰的中间还带有一个尖峰 C,可能是其他类型的异常,是与 B 不同的行为模式.下面使用上述定义的测度进行分析.

图 4 列出了 TSE,CCR 和 RST 报文 AAAR,ANAR 的时间分布曲线,根据定义可以看出,TSE 的负峰 A,B,C 和 E 对应着 4 个异常的时间点,与图 3 中异常发生的时间点吻合.RST 报文 AAAR,ANAR 的时间分布曲线 RST_ONLY 和 RST+ACK 的行为具有较大的区别,根据定义得到的异常时间点 B 和 E 也与图 3 里异常时间点一一对应.

定位了异常发生的时间,下面只需对异常发生的 5 个点 A,B,C,D,E 的各种报文尖峰进行详细的异常类型和参数分析.

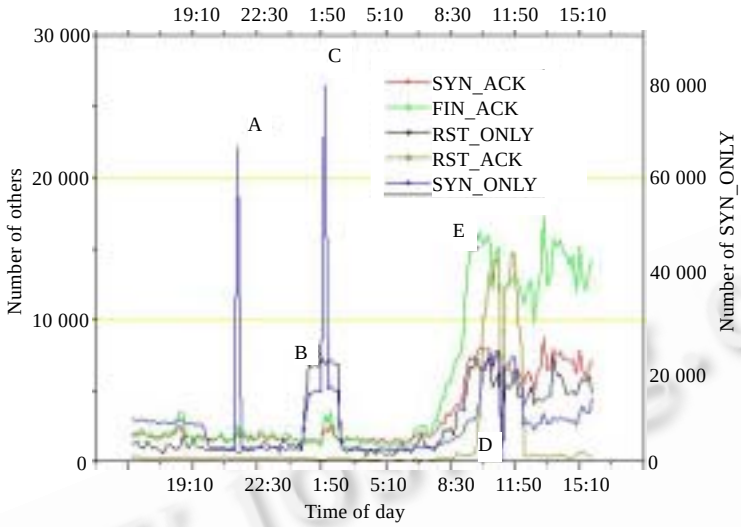

Fig.3 The time curves of TCP packets

图 3 TCP 各类报文的时间分布曲线

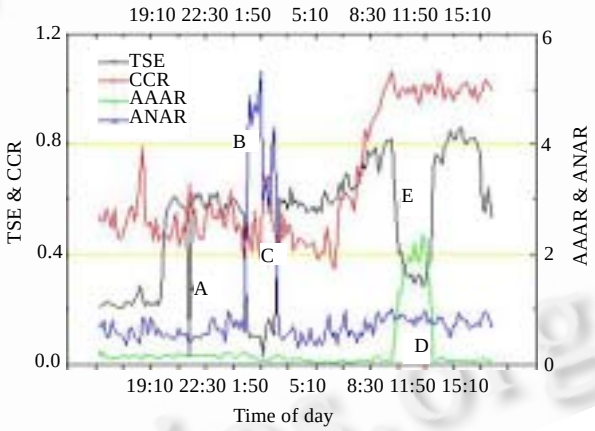

Fig.4 The time curves of TSE,CCR,AAAR and ANAR

图 4 TSE,CCR,AAAR 和 ANAR 的时间分布曲线

3.3 实例:用BFR对异常的参数和类型进行分析

上节定位了异常发生的时间点,下面需要对该时间点的TCP报文进行BFR分析.我们可以发现,对于多报文的扫描和DDoS攻击,本文提供的Bloom Filter方法对不同类型的TCP报文得到了高度一致性的结论.为了避免篇幅过长,这里只对上述5点的异常行为进行分析,只给出A点的Top 10表格(见表3).

3.3.1 A 点

根据 BFR 算法,10.2.74.207 从 21 端口向外发送了大量的 SYN_ONLY 报文,占了 SYN_ONLY 总报文数的 95.5%.目标地址则为 10.2 网段,分布均匀;目标端口集中为 21.可以看出,这个 IP 地址对外作了一次扫描.SYN_ACK 报文只有 1 926 个,且只有极其少量的主机 IP 响应上述 10.2.74.207 的扫描.

3.3.2 B 点

10.2.117.165 又发出了大量的 SYN_ONLY 报文,源端口和目标地址分散,目标端口为 111.这些又是扫描的特征.其伴随的 RST_ONLY 报文仍然是 10.2.117.165 又发出了大量的 RST_ONLY 报文,源端口和目标地址分散,

目标端口为 111,即该主机同时使用两种扫描方法进行扫描.

3.3.3 C 点

主机 10.2.74.207 从 21 端口向 21 端口发 SYN 报文,宿 IP 分散.10.2.117.165 的多报文扫描也同时存在,这里略去其 RST_ONLY 报文的表格.D 点和 E 点的 Top10 表格省略,下面只列出分析的结果.

3.3.4 D 点

所有的 TCP 报文总数加起来不到 500 个,可能是链路故障,也可能是路由器被 DDoS 攻击复位了.

3.3.5 E 点

SYN_ONLY 报文:10.0.0.161 对 10.2.176.19 的 2000 端口发送了 13 554 个报文进行 DoS 攻击;
对 10.0.0.38,10.0.0.3 的 80 端口进行的访问.

SYN_ACK 报文:10.0.0.38 和 10.0.0.3 的响应.

FIN_ACK 报文: SIP:10.0.0.3,10.0.0.38,10.0.0.161,10.2.191.137 端口 80;
DIP:10.0.0.3,10.0.0.38,10.0.0.161,10.2.191.137 端口 80.

RST_ACK 报文:10.2.176.19 的 2000 端口对 10.0.0.161 的 DoS 攻击的自然回应.

Table 3 The Top 10 in bloom filter's storage of abnormal SYN_ONLY packet at point A, total_hit 66 388

表 3 对 A 点的异常报文 SYN_ONLY,Bloom Filter 里 Top 10 存储的内容,其 total_hit 为 66 388

SIPH	Hits	SIPM	Hits	SIPL	Hits	SPORT	Hits
10.2	63 600	2.74	63 497	74.207	63 381	21	63 381
10.0	2 665	0.0	2322	0.38	476	22 189	9
10.1	123	0.1	53	0.3	460	22 307	9
-		2.107	18	0.28	419	22 328	9
-		0.4	12	0.83	412	22 393	9
-		0.16	10	0.232	270	22 516	9
-		2.72	10	0.203	140	22 881	9
-		0.48	9	1.76	51	22 962	9
-		0.47	8	0.3	39	22 981	9
-		0.3	7	0.22	27	23 047	9
DIPH	Hits	DIPM	Hits	DIPL	Hits	DPORT	Hits
10.0	26 106	0.0	1511	0.203	480	21	63 381
10.1	25 762	0.186	436	0.28	476	80	1 505
10.2	14 520	2.93	259	186.21	333	711	1 241
-		2.77	256	0.27	286	25	110
-		2.8	256	0.13	87	113	51
-		2.82	256	1.44	65	135	22
-		2.83	256	13.105	45	6 346	22
-		2.84	256	74.195	34	179	18
-		2.85	256	1.51	33	443	11
-		2.86	256	46.212	30	139	8

对上述情况进行综合可以得知,这个时刻的情况比较复杂,同时存在扫描、DoS 攻击和正常连接.

通过对 BFR 的输出进行综合分析发现:这个算法在再现异常发生时的 IP 地址参数时是高效的,不同的 TCP 类型报文间的结果相互应证.

4 相关的研究

Bloom Filter 算法最早在 1970 年提出[6],用于压缩比特串空间搜索的开销.后来,各种改进和性能评估都是针对其资源开销来进行的,然后其在各种领域被广泛使用.在网络流的研究热潮里[16],使用 Bloom Filter 算法确定流的数量的工作最卓越的就是在 IMC2003 上获得最佳学生论文奖的 Kummar 等人的工作[23].他们使用一个比特位来代表实际存在的流,以此估计网络里实际的流数目.这样做的确节约了很多计算资源,但这种方法却损失了很多流的信息.

2004 年,Schweller R 等人在 IMC2004 会议上提出一种可反向追溯的方法[24],基于 Sketch 算法.Sketch 的工作原理和 Counting Bloom Filter 很像,与 Counting Bloom Filter 最大的区别在于,在 hash 的过程中引入了一个符号函数,使得计数器可以减少,Counting Bloom Filter 的计数器只增加不减少.这种可反向追溯的方法基于复杂

的分类过程,且哈希函数非常复杂,整个还原过程不直观.

上述方法的局限性在于,它们并没有一个简单的还原原始信息的机制,在发生冲突的地方,不能用不同的哈希函数相互纠正、相互补充的功能.本文充分利用不同的 Hash 函数的相互补充和印证来消除外部冲突,获得完美的信息再现效果,给每个哈希数组独立的存储空间,以少量的空间消耗消除了 Hash 函数在共享空间上的内部冲突.另外,精心选择的简单映射也使得信息再现的工作很容易.

对于利用 SYN/FIN/RST 报文对来进行异常检测的方法早已经有研究报道,Wang HN 等人[3]使用 SYN/FIN(RST)的复杂统计模型来检测 SYN Flooding 攻击,使用 SYN-Dog[4]来监视叶结点路由器的工作状况.但这个方法没有考虑聚类,使得维护五元组的资源消耗很大.Yuichi O.等人提出通过分析 TCP 的 SYN 报文的统计信息来检测 DDoS 攻击[5],但没有从系统的角度来阐述为什么可以这样做、是否能做得更好.本文利用 TCP 连接的唯一性产生的 TCP 连接报文数量的约束来指导发现异常行为,利用特别选择 Hash 函数的 Bloom Filter 算法来快速确定大规模 TCP 连接异常发生时的参数,如受害主机和异常行为的发起者的 IP,使得快速检测大规模异常和实时防御成为可能.如果把相应的 Bloom Filter 数据结构存储下来,可以作为主干流量的一个摘要.

5 结论与将来的工作

本文提出一种在不维护五元组的时候还原大规模 TCP 连接异常的发起者和受害者的快捷方法——BFR.它使用简单的 Hash 函数,不仅保证了信息再现的过程的简洁性,也保证了整个算法高的资源使用效率、低的计算复杂度.它利用 Hash 函数中重叠的比特位和给每个 Hash 函数独立的储存空间来化解原始信息间的外部冲突和 Hash 函数间的内部冲突,以少量的资源开销,提高了 BFR 算法的精度;它利用 TCP 报文类型数量的自然约束快速定位异常发生的时间,结合 BFR 算法,能够快速检测高速信道上的 TCP 连接的大规模异常行为及相关的类型和参数.

利用网络上的某种异常,如扫描和 DoS,DDoS 攻击引发的 TCP 某个标志报文数量的大幅改变,本文的异常检测模型和 BFR 方法可以迅速检测到异常并准确再现异常的受害者/发起者的 IP.对实际网络的详细分析表明,BFR 算法对于揭示异常行为的类型和参数都非常有帮助,并能够在多种类型的异常,如多种扫描和 DoS 攻击同时发生的时候仍能够准确无误.

将来的工作是借助 BFR 算法通过模拟来完善我们的异常检测模型,使其成为一个完备的理论体系,这还需要付出更多的努力.由于异常发生时刻检测的方法可以不区分方向,而把检测异常参数的任务交由 BFR 完成,有可能把这些指标和参数再现方法应用到路由器群或者网络云的边界上.对所有进出网络云的 TCP 流量进行类似的宏观 Tomography 分析,减少流量工程和不对称路由对测量和异常检测的影响,对于 ISP 监控运营中的网络将会提供有力的支持.如果把相应的 Bloom Filter 数据结构存储下来,可以作为主干流量的一个摘要.

References:

[1] Feinstein L, Schnackenberg D, Balupari R, Kindred D. DDoS tolerant networks. In: Proc. of the DARPA Information Survivability Conf. and Exposition. 2003. 73−75.

[2] Chen SG, Chow R. A new perspective in defending against DDoS. In: Proc. of the 10th IEEE Int'l Workshop on Future Trends of Distributed Computing Systems (FTDCS). 2004. 186−190.

[3] Wang HN, Zhang DL, Kang GS. Detecting SYN flooding attacks. Proc. of the INFOCOM 2002. IEEE, 2002,3(23−27):1530−1539.

[4] Wang HN, Zhang DL, Kang GS. SYN-dog: Sniffing SYN flooding sources. In: Proc. of the 22nd Int'l Conf. on Distributed Computing Systems (ICDCS 2002). 2002. 421−428.

[5] Ohsita Y, Ata S, Murata M. Detecting distributed denial-of-service attacks by analyzing TCP SYN packets statistically. In: IEEE GlobeCom. IEEE Communications Society, 2004. 2043−2049.

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

[7] Kumar K, Xu J, Jia W, Spatschek O, Li L. Space-Code bloom filter for efficient per-flow traffic measurement. In: Proc. of the INFOCOM 2004. Vol 3, New York: ACM Press, 2004. 1762−1773.

[8] Nicolas H, Darryl V. Inverting sampled traffic. In: Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. 2003. 222−233.

[9] Kumar A, Sung M, Xu J, Wang J. Data streaming algorithms for efficient and accurate estimation of flow size distribution. In: ACM Sigmetrics. New York: ACM Press, 2004. 177−188.

[10] Postel J. Transmission control protocol, RFC793. Internet Society, 1981.

[11] Koloniari K, Pitoura E. Bloom filters for hierarchical data. In: Proc. of the 5th Int'l Workshop on Distributed Data and Structures (WDAS). 2003.

[12] Bernard C, Joe K, Ronitt R, Ayellet T. The bloomier filter: An efficient data structure for static support lookup tables. In: Proc. of the 15th Annual ACM-SIAM Symp. on Discrete Algorithms Table of Contents. Philadelphia: Society for Industrial and Applied Mathematics, 2004. 30−39.

[13] Little MC, Speirs NA, Shrivastava SK. Using bloom filters to speed-up name lookup in distributed systems. The Computer Journal, 2002,45(6):645−652.

[14] Chin-Chen C, Tian-Fu L, Jyh-Jong L. Partition search filter and its performance analysis. Journal of Systems and Software, 1999, 47(1):35−43.

[15] Sarang D, Praveen K, David ET. Longest prefix matching using bloom filters. In: Proc. of the Conf. on SIGCOMM. New York: ACM Press, 2003. 201−212.

[16] Andrei B, Michael M. Network applications of bloom filters: A survey. Internet Mathematics, 2003,1(4):485−509.

[17] Kohler E, Li JY, Paxson V, Shenker S. Observed structure of addresses in IP traffic. In: Internet Measurement Workshop 2002. New York: ACM Press, 2002. 253−266.

[18] http://tracer.csl.sony.co.jp/mawi/

[19] http://tracer.csl.sony.co.jp/mawi/samplepoint-B/20050107/200501070000.html

[20] Angiulli F, Pizzuti C. Outlier mining in large high-dimensional data sets. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(2):203−215.

[21] Wen J, Anthony KHT, Jiawei H. Mining top-*n* local outliers in large databases. In: ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco, New York: ACM Press, 2001. 293−298.

[22] http://pma.nlanr.net/Traces/long/bell1.html

[23] Kumar A, Jun X, Li L, Jia W. Space-Code bloom filter for efficient traffic flow measurement. In: Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2003. 167−172.

[24] Schweller R, Gupta A, Parsons E, Chen Y. Reversible sketches for efficient and accurate change detection over network data streams. In: Proc. of the ACM SIGCOMM Internet Measurement Conf. (IMC). New York: ACM Press, 2004. 207−212.

龚俭(1957－),男,上海人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络行为学,网络安全.

杨望(1979－),男,博士生,主要研究领域为入侵检测系统的评估.

彭艳兵(1975－),男,博士生,主要研究领域为网络行为学,网络安全.

刘卫江(1969－),男,博士,教授,主要研究领域为网络抽样.