一个检测超流的早期淘汰算法

程 光, 强士卿
(东南大学计算机科学与工程学院, 江苏省计算机网络技术重点实验室, 江苏 南京 210096)

摘要: 超流是网络中具有大量不同宿 IP/源 IP链接的源 IP/宿 IP. 实时检测出高速网络(OC48, OC192等)中的超流对网络安全具有非常重要的意义. 近年来已经有论文提出了不同超流检测算法, 但是这些算法都没有考虑控制测量过程中超流缓存空间的大小. 论文在超流检测过程中增加了早期淘汰功能, 将超流缓存中链接数较少的记录提前淘汰, 以便腾出空间用于存储新检测到的 IP记录, 实现对超流缓存资源的控制. 论文最后还使用实际网络日志数据对算法进行验证, 实验表明: 早期淘汰算法可以使系统以更少的内存空间和测量资源检测超流信息.

关键词: 早期淘汰算法; 超流; 流量测量
中图分类号: Q 393 文献标识码: A 文章编号: 0438-0479(2007)S2-0202-03

检测出链接大量宿 IP(源 IP)的源 IP(宿 IP)的问题属于超流问题, 超流问题在网络安全中具有重要的应用价值. (1) 检测网络蠕虫: 一个蠕虫主机向大量的宿地址发送探测流量, 这台蠕虫主机可以看成是一个超流主机. (2) 检测 DDoS/DoS: DDoS/DoS攻击过程中, 大量的主机/IP地址向一个目标地址发送大量的流量, 这个被攻击的目标主机也可以看成一个超流主机. (3) 检测端口扫描攻击: 一个主机为了发现易攻击的对象存在, 向不同的 IP地址和不同的端口发起大量链接, 这个源 IP也可以被看成是超流 IP. (4) 超流问题还可以用在 P2P分布式网络中, 主机对中可能会产生大量的链接, 超流 IP被认为是热点 IP, 通过实时检测超流 IP有助于进行负载均衡以提高网络的效率.

检测一个 IP是否是超流的最直观方法是使用哈希表直接记录所有的流信息, 如 Snort[1]和 FlowScan[2]都是使用这种方法, 它的缺点是需要消耗大量的内存空间. Venkataraman[3]提出两种基于流抽样的超流 IP检测技术, 这种方法的本质和直接使用哈希表存储所有流信息没有区别, 也需要消耗大量的内存空间用于存储抽样流. Qi[4]提出使用抽样技术和数据流方式相结合的两种检测聚合流大小的方法.

由于链接数很少的 IP数量很大, 各类超流检测算法虽然能够过滤掉部分链接数少的 IP, 但是还是有大量链接数少的 IP没有被过滤, 这些 IP记录占用了大量超流缓存空间. 超流检测算法在测量过程中只是考虑对超流缓存空间进行增加或者更新记录内容, 而对已经在超流缓存中的记录不进行删除, 这样导致超流缓存空间无法得到控制. 本论文在超流检测过程中增加了早期淘汰功能, 将链接数较少的 IP记录提前淘汰, 以便腾出空间用于存储新检测到的 IP记录. 论文的研究重点是在超流检测过程中增加了早期淘汰算法, 使得系统可以使用更少的内存空间和测量资源检测超流信息.

1 早期淘汰算法

图 1是一个带有早期淘汰功能的超流检测算法, 其中的超流检测算法可以是目前已有一种算法, 本文在此基础上增加早期淘汰功能, 以实现对超流缓存空间中的记录进行实时检测, 淘汰部分不属于超流的 IP记录. 链接数少的 IP数量很大, 所以超流检测算法虽然能够过滤部分非超流 IP记录, 但是还是有大量的非超流被记录, 占用了大量的超流缓存空间, 因此在测量过程中超流缓存空间中链接数较少的记录提前淘汰可以腾出空间用于存储新到达的记录.

具体早期淘汰算法思路如下: 将测量时间区间 T 分成 h 等份子时间区间, 设在第 i 个子时间区间中, 如果超流缓存记录在第 i 个子时间区间中没有新的流到达, 或者已到达链接的速率小于定义的阀值 y, 则该记录被淘汰.

假设定义链接数大于 F 的 IP为超流, 一个超流中的链接到达次序服从随机分布, 即每个链接在每个子

收稿日期: 2007-07-28
基金项目: 江苏省基础研究计划(自然科学基金)项目(BK2006092), 国家教育部科学技术研究项目(105084)和东南大学优秀青年教师教学科研资助计划(4009001018)资助
作者简介: 程光, 男, 博士, 副教授.
Email: gcheng@njnet.edu.cn

时间区间内到达的概率相等. 因此对于某一个子时间区间内, 链接数为 F 的 IP 在第 i 个子时间区间内没有新的链接到达的概率为: p (第 i 子时间区间的链接数 $= 0) = (1 - 1/h)^F$. 设定义的平均链接数阀值是 v, 当前 i 个子时间区间内到达链接的平均速率小于或等于 v 时, 即在前 i 个子时间区间内到达的链接小于或等于 $v^* i$, 且当前的时间区间又没有新链接到达, 则将这个 IP 记录淘汰.

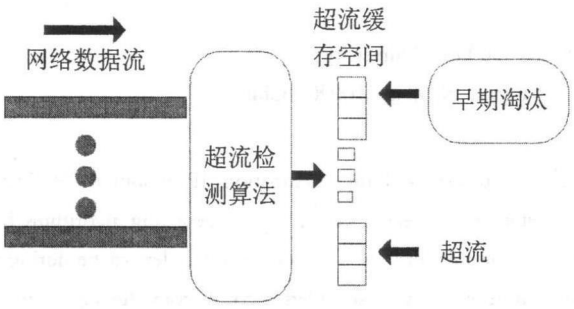

图 1　带早期淘汰算法的超流检测图

下面将 h 等份的时间间隔分成两个部分, 前 i 个时间间隔为一部分, 后 $h - i$ 个时间间隔为另一部分, 任意链接落入第一部分的概率为 i/h, 概率分布为二项分布, 落入第一部分链接数的数学期望为: $E(\hat{s}) = F \cdot i/h$, 其方差为: $\mathrm{Var}(\hat{s}) = F \cdot i/h \cdot (1 - i/h) = F \cdot i \cdot (h - i)/h^2$. 由于 F 取值较大, 因此该分布可以近似为正态分布. 将分布归一化处理, 其正态分布的均值为公式 (1).

$$x_i' = \frac{x - E(\hat{s})}{\sqrt{\mathrm{Var}(\hat{s})}} = \frac{v \cdot i \cdot h - F \cdot i}{\sqrt{F \cdot i \cdot (h - i)}} \qquad (1)$$

前 i 个子时间区间内到达链接的平均速率小于 v 的概率为 $\Phi(x)$, 结合第 $i+1$ 个子时间区间链接到达数的概率 p_{i+1}^0, 如果设定 $\Phi(x) \leqslant 0.01$, 则 $x \leqslant -2.33$, $\frac{v \cdot i \cdot h - F \cdot i}{\sqrt{F \cdot i \cdot (h - i)}} \leqslant -2.33$ 根据过滤规则, 超流可能在第 $i+1$ 个子时间区间被早期淘汰的概率 $p_i = \Phi(x) \cdot p_{i+1}^0 = \Phi(x) \cdot (1 - 1/h)^F$, 因此其在测量时间内可能被淘汰的概率为公式 (2).

$$P_E = 1 - \prod_{i=1}^{h} (1 - p_i) \qquad (2)$$

举例: 检测链接数的阀值为 $F = 100$, 区间数量为 $h = 10\%\ F = 10$, 淘汰阀值为 $20\%\ F = 20$, 平均流速阀值为 2, 则 $x \leqslant -3.35$, $\in [1, 10]$, 因此 $\Phi(x) < 0.0004$, $\in [1, 10]$, $p_i = \Phi(x_i') \cdot (1 - 1/h)^F = 0.0004 \cdot (1 - 1/10)^{100} = 1.06 * 10^{-8}$, 则第 $i+1$ 个子时间区间内被淘汰的概率最大不超过: $p_E = 1 - (1 - 1.06 \times 10^{-8})^{10} = 1.06 \times 10^{-7}$.

2　实验分析

论文中我们采用的 NLANR 公布的两组实验数据: IPLS-CLEV 和 IPLS-ATLA 进行测试[5], 表 1 列出 8 种类型数据测试的超流缓存空间进行比较, 其中 "IP 数量" 是数据中 IP 地址的数量, "> = 100 超流" 是指链接数超过 100 的 IP 数量, "> = 100 超流缓存空间" 是不带早期淘汰的超流检测算法为了检测大于 100 的超流需要的超流缓存空间数量, "> = 100 早期淘汰" 是带早期淘汰的超流检测算法为了检测大于 100 的超流需要的超流缓存空间数量. 实验表明增加早期淘汰算法的超流检测算法能够比不使用早期淘汰功能的检测算法节省超流缓存空间 2~4 倍.

表 1　超流缓存空间大小比较

数据	IP 数量	> = 100 超流	> = 100 超流缓存空间	> = 100 早期淘汰
ATLA-0-Source	8432	281	3000	1231
ATLA-0-Dest	33760	103	6463	1041
ATLA-1-Source	11747	104	3193	920
ATLA-1-Dest	47232	190	7096	969
CLEV-0-Source	26160	644	8162	3037
CLEV-0-Dest	76526	359	13357	2969
CLEV-1-Source	18050	236	5427	1556
CLEV-1-Dest	104333	143	14024	1457

3　结　论

检测超流在网络安全中具有重要的应用价值, 传统的超流检测算法没有考虑早期淘汰功能, 因此无法控制超流缓存空间大小. 论文在超流检测算法中增加了一种早期淘汰功能, 在测量过程中对不属于超流的部分记录进行实时淘汰, 以节省和控制超流缓存空间. 论文使用 NLANR 流量日志对算法的性能进行了验证.

参考文献:

[1] Roesch M. Snort- lightweight intrusion detection for network[C]// Proc USENIX Systems Administration Conference 1999.

[2] Plonka D. Flowscan: a network traffic flow reporting and visualization tool[C]// USENIX LISA. 2000.

[3] Venkataraman S, Song D, Gibbons P, et al. New streaming

algorithms for fast detection of superspreaders[C] // In Proc NDSS. 2005.
[4] Zhao Qi, Abhishek Kumar, Xu Jun. Joint data streaming and sampling techniques for detection of super sources and destinations[C] // IMC 2005. 2005: 77-90.
[5] NLANR PMA. Special Traces Archive[EB/OL]. http://pma.nlanr.net/Special/.

An Early Removal Algorithm for Detecting Superspreaders

CHENG Guang, QIANG Shi-qing

(College of Computer Science & Engineering, Southeast University,
Key Laboratory of Computer Network Technology in Jiangsu, Nanjing 210096, China)

Abstract A superspreader is a source IP or destination IP which connects a large number of distinct destination IP or source IP. Detecting superspreaders is very significant to manage and monitor the high-speed network. In recent years, several detecting algorithms have studies how to solve the problem at high-speed link. However they can't control the memory resources of superspreaders cache during the measurement. In this paper, we increase an early removal algorithm to remove some non-superspreaders records from the superspreaders cache so that some free memory space can store new detected IP records and control the superspreaders cache size. Finally, we use NLANR traffic traces to verify the early removal algorithm, and its result shows that the algorithm can save and control the measurement resource.

Key words early removal algorithm; superspreaders; traffic measurement