利用流量特征的 GIDS 报文分类优化算法

宁 卓[1],孙知信[1,2],龚 俭[3],张维维[3]
(1. 南京邮电大学宽带无线通信与传感网技术教育部重点实验室,江苏南京 210003;
2. 南京大学计算机软件新技术国家重点实验室,江苏南京 210093;3. 东南大学 计算机科学与工程学院,江苏南京 210096)

摘 要: 本文结合流量的动态特征和入侵检测系统规则库的静态特征生成高性能报文分类树,提出了一个新的面向骨干网高速入侵检测的报文分类算法 FlowCopySearch(FCS).改进在于:①从流量的新角度提出了最优分类树定义并引入分类域熵衡量每个分类域对于流量的分类能力;②将传统分类算法中每个报文都必须频繁执行的内存拷贝操作简化为每个流只执行一次内存拷贝操作,克服了报文分类算法的瓶颈.实验结果表明 FCS 更适用于骨干网大流量 trace 的报文分类,较之两种经典分类算法,分类速度提高了 10.1% ~ 45.1%,同时存储消耗降低了 11.1% ~ 36.6%.

关键词: 入侵检测系统; 属性熵; 重尾分布特性
中图分类号: TP393.08 **文献标识码:** A **文章编号:** 0372-2112 (2012) 03-0530-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.03.020

An Improved GIDS Packet Classification Algorithm Using the Characteristic of the Traffic

NING Zhuo[1], SUN Zhi-xin[1,2], GONG Jian[3], ZHANG Wei-wei[3]
(1. *Key Laboratory of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu* 210003, *China*;
2. *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*;
3. *School of Computer Science and Technology, Southeast University, Nanjing, Jiangsu* 210096, *China*)

Abstract: A classification algorithm FlowCopySearch (FCS) is developed that systematically profiles static intrusion signatures and network traffic to generate a high performance and memory-efficient packet classification tree. The improvements are two folds. Firstly, the best classification tree is formally defined and packet feature entropy is proposed to measure how well a packet field can partition the traffic. Secondly, FCS copies a rule set for a flow instead of traditionally copying the rule set for every packet in the flow, so the classifying speed is increased considerably. The experiment results show that in backbone trace FCS is preferred. Compared to the other two classical algorithms, FCS can not only speed up classification by as much as 10.1% ~ 45.1% in speed, but also save memory consumption of 11.1% ~ 36.6% at the same time.

Key words: intrusion detection system; packet feature entropy; heavy hitter

1 引言

滥用入侵检测系统 IDS(Intrusion Detection System)中的报文分类算法是使用多个分类域的高维分类算法.假定 IDS 的规则集 R 含有 N 个规则,每个规则 R_j 包含了若干关于报文头 S 个域以及关于报文负载的 T 个域的某种约束.对于一个到达报文 P,假设考虑报文头的 S 个域和报文负载的 T 个域,则($S+T$)维报文分类算法的任务是在 R 中找到 P 所激活的子规则集 $R_i \sim R_n$.然后报文检测会调用字符串匹配算法在 $R_i \sim R_n$ 中查找特征字符串以确定 P 匹配的所有攻击规则.显然分类算法的优劣直接影响到每个到达报文需要进行的匹配工作量 $R_i \sim R_n$,对 IDS 的性能影响巨大,因此通过提高分类算法的性能来提高 IDS 的性能一直是入侵检测一个重要的研究方向.

解决骨干网 10Gbps 速率以上的 IDS 报文分类问题,涉及到大规则集中的高维分类,是比较困难的.为了论述方便首先介绍用到的符号如下:设 d 为分类维数,n 为规则个数.文献[1]通过分析子空间不相交情况下的多维空间点定位问题给出了分类算法的性能上下界:

收稿日期:2011-06-22;修回日期:2011-12-14
基金项目:国家 863 高技术研究发展计划(No.60973140,No.61170276);江苏省自然科学资金(No.BK2009425);南京邮电大学引进人才科研启动基金(No.NY210081)

当 $d>3$ 时,多维分类时间最优算法的时间复杂度为 $O(\log_2 n)$,空间复杂度达到 $O(n^d)$;空间最优算法中当空间复杂度达到 $O(n)$ 时,则时间复杂度为 $O((\log_2 n)^{d-1})$.因此各种分类算法既要考虑到 IDS 的报文处理速度要求,又要力求在时间和空间消耗中找到一个折中点.

硬件算法扩展性差[2],特别对于 IDS 规则库更新频繁,报文分类维数多(10 个以上)的情况,硬件算法显得成本过高.软件算法最有代表意义的就是 Hicuts[3],它将以往逐条规则(rule-to-rule)的比较改变为逐个分类域(feature-to-feature)的比较,其查找时间复杂度是 $O(d)$,奠定了 IDS 分类树的基本结构.但是 Hicuts 存在着空间异常膨胀和决策树不平衡问题,最坏空间复杂度达到 $O(n^d)$.NI Trie 分类算法[4]具有更强的表述能力,但是带来的缺点是如果规则集 R 存在大量强冲突会大幅降低 NI Trie 的分类速度.P-Hicuts[5]针对 Hicuts 空间复杂度过高的问题提出了非均匀切分和覆盖规则上提两点改进,从而降低了分类树的高度,减小了平均分类复杂度.文献[6]改进了 P-Hicuts 中 TCP 标志域数值化方法,降低了 TCP 标志域的分类节点数目,但是对分类树查找速度没有改进.经过上述若干次改进后 IDS 的报文分类算法基本定型,理论上其平均时间复杂度为 $O(d)$,正比于分类树的深度,各种优化手段保证空间占用远小于其最坏空间复杂度 $O(n^d)$.

WIND[7]创新地提出利用流量的动态特征来指导分类树构造的方法,其本质是一种分类节点切分的启发式方法.以流量中分类域的特殊属性值能排除的规则个数 m 来衡量是否为此特殊属性值建立单独的分类节点.但它的方法还远未成熟,这体现在①WIND 只是通过实验证明按照当前流量特征构建的分类树,可以将报文分类速度提高至未经优化的 Snort 的 1.3 ~ 1.7 倍,占用存储量比 Snort 分类树少近 15%,而直觉地得出多次出现的特殊属性值分类节点的建立可以加快分类速度的结论,而且其采用 DARPA99 小规模人工合成数据集得出的实验结论是否具有一般性有待商榷.②WIND 根本没有讨论有关自适应的动态更新的问题,比如采用多长时间的流量样本、能利用流量的那些动态特征来提高分类树查找速度构造以及如何更新分类树结构等等关键问题都没有讨论.

受 WIND 启发,本文一改只考虑 IDS 规则库的静态特征的分类角度,对上述 WIND 未尽研究点深入探讨,利用动态流量的特点改进 IDS 静态分类算法,提出了一个新的分类算法 FlowCopySearch(FCS),实验证明较之经典静态分类算法 P_Hicuts[5]和动态分类算法 WIND[7]它能在提高分类速度的同时减低算法运行时内存消耗.

2　动态最优分类树的求解

传统分类算法[3~6]仅仅只考虑 IDS 规则库的静态特征,而完全忽略了访问分类树的主体——网络流量的特性,因而无法利用动态流量特性来优化分类树结构.WIND 虽然用流量特性来确定分类节点的划分,但是它的算法性能无法适应骨干网入侵检测的高要求,且不具备自适应更新能力.本文从流量的角度出发,认为本质上优化分类树的目标并不是像传统分类算法那样,将 IDS 规则库分得越均匀越好,而是将当前流量对分类树的访问分得越均匀越好,即最优分类树的目标是使得当前流量的分类代价最小,并在后继章节讨论了此目标下的两种优化改进措施.

定义 1　流量的分类代价　设 $\boldsymbol{P}=\{p_1,p_2,\cdots,p_i,\cdots p_n\}$ 表示时间片内到达报文集合,则 P 的分类代价 $V(P)$定义为 $V(P)=\sum_{i=1}^{n} w_i|R_i|$,其中 w_i 是报文 p_i 访问分类树的路径长度,$R_i=\{r_m,\cdots,r_n\}$是 p_i 激活的分类规则集,$|R_i|$是 R_i 包含的规则数目.

定义 2　最佳分类树　使得流量的分类代价 $V(P)$ 最小的分类树是当前流量的最佳分类树.

要使得 $V(P)$取最小值,显然希望每个报文的 $w_i|R_i|$都取最小值.按分类树的分类原则,对于某一个分类域的划分采用非均匀切分方法[5],因此以什么样的顺序安排分类域,尽快使得 $|R_i|<$阀值,保证 w_i 取最小值,是最佳分类树的优化关键所在.

3　改进 1——基于分类域熵大小顺序选择分类树的层次划分特征

在上述目标下,对于时间片内的流量 P,分类树划分的原则应该是保证把流量尽量均匀地划分到更多的规则分类结点中去,即把流量分得越均匀越好.据此我们采用信息论的熵理论,提出了分类属性熵的概念来衡量各种分类域对于流量的分类能力.

定义 3　分类域熵　设 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_n\}$ 代表当前流量,每个 x_i 都代表一个报文,X 也称作报文集合.设 $R=\{r_1,r_2,\cdots,r_i,\cdots,r_m\}$是 IDS 的攻击规则集合,$A=\{a_1,a_2,\cdots,a_j,\cdots,a_u\}$代表分类域集合,$R$ 在 a_j 上的非均匀切分形成的分类为 $\mathrm{Cut}_{a_j}=\{C^1_{a_j},\cdots,C^i_{a_j},\cdots,C^{t+1}_{a_j}\}$.则 $\forall x_i,\exists C^j_{a_j}$,使得 $x_i\in C^j_{a_j}$.则分类域 a_j 对于当前流量 X 和规则集分类 Cut_{a_j} 的分类属性熵定义为:$H_X(a_j)=-\sum_{i=1}^{t+1} p(C^i_{a_j})\log_2\frac{1}{p(C^i_{a_j})}$,$p(C^i_{a_j})=\frac{U(C^k_{a_j})}{n}$,其中 $U(C^k_{a_j})$是分类结点 $C^k_{a_j}$被 X 中的所有报文访问的次数,n 是报文总数.

分类域熵 $H_X(a_j)$的取值范围在$[0,\log_2(t+1)]$之间.当 $H_X(a_j)$取值为 0 时,说明采用 a_j 分类,当前流量 X 对于分类 $\mathrm{Cut}_{a_j}=\{C^1_{a_j},\cdots,C^i_{a_j},\cdots,C^{t+1}_{a_j}\}$的访问分布达到最大集聚,即所有报文都访问的是同一个分类节点 $C^i_{a_j}$.$H_X(a_j)$取值为 $\log_2(t+1)$时,当前流量 X 的访问分布最发散,X 中的报文访问每个分类节点 $C^i_{a_j}$的取值次数相等 $U(C^1_{a_j})=U(C^2_{a_j})=\cdots=U(C^n_{a_j})$.采用分类域熵我们可以方便地统计采用不同的分类域 a_j 分类,某个时间片中流量对分类 $\mathrm{Cut}_{a_j}=\{C^1_{a_j},\cdots,C^i_{a_j},\cdots,C^{t+1}_{a_j}\}$的不同分类节点的访问分布情况,分布越发散说明该分类域能将当前流量分得越均匀,反之,则说明该分类域根本无法有效划分当前流量.

按照分类域熵的定义,本节考察了目前 Snort 中可用的 14 个分类域的分类能力,它们在实验中的取值范围以及分类能力排序如表 1 所示.

表 1 分类域分类能力排序表

分类能力排序	分类域	缩写	取值范围
1	宿端口	Dstport	4.x
2	源端口	Srcport	3.x
3	宿地址	DstIP	1.x
4	TCP 标志域	TCPFlags	1.x
5	源地址	SrcIP	0.4x ~ 0.8x
6	ICMP 标志	ICMPId	0.2x ~ 0.3x
7	TCP 序列号	TCPSeq	0.2x ~ 0.3x
8	ICMP 代码	ICMPCode	0.2x ~ 0.3x
9	协议	Protocol	0.2x ~ 0.3x
10	Ttl 域	CheckTtl	0.00x
11	TCP 报文净荷长度	TCPLen	0.00x
12	UDP 报文净荷长度	UDPLen	0.00x
13	TCP 确认序列号	TCPAck	0
14	ICMP 序列号	ICMPSeq	0

为了衡量这些分类域分类能力的连续变化趋势,采用 CERNET 江苏省网边界上一条 10Gbps 链路 2009 年 2 月 24 日连续 8 个小时的 netflow 数据作为实验数据,抽样率为 1/256,考察真实流量下每分钟各种分类属性熵的取值分布及变化趋势.这样做的原因有三:①netflow 数据流量大,持续时间长,更能反映分类域熵变化的宏观趋势;②研究表明随机抽样对于熵测度的影响不大[8],采用抽样数据既能保证分类熵测度的准确性又能降低处理复杂度;③时间片取一分钟是因为报文分类树构造的时间级别是分钟级的,因此小于分钟级别的分类域熵变化对优化分类树结构没有意义.但是采用 netflow 数据只能衡量前五个分类域(五元组)的熵变化,因为 netflow 数据不包含其它的分类域信息,只能采用相应时段的 trace 来计算其它分类域熵.由于骨干网 trace 数据量大,只考察了相同 10Gbps 链路上前 5 小时内其它分类域熵每分钟的熵值变化,如图 2 所示.

图 1 和图 2 显示了上述 14 种分类域的连续取值分布.图 1 显示了五元组信息(源/宿地址、源/宿端口和协议)熵值连续 8 小时的每分钟熵值分布图.图 2 显示了其它分类域 5 小时的每分钟熵值分布图.实验结果显示①每分钟每种分类域熵值取值都比较稳定,几乎都在一个固定值附近变化.宿端口和源端口的熵值分别为四点几和三点几左右,DstIp 、SrcIp 和 TCPFlags 熵值在 1 左右,而其它分类域熵值取值都不到 0.5.ICMPSeq 和 TCPAck 两个分类域甚至在 5 小时 trace 中取值始终为 0.②取值偶有突变点产生,而且分类域熵值越大,突变点越多,而那些熵值小于 0.5 的分类域几乎不存在突变点.

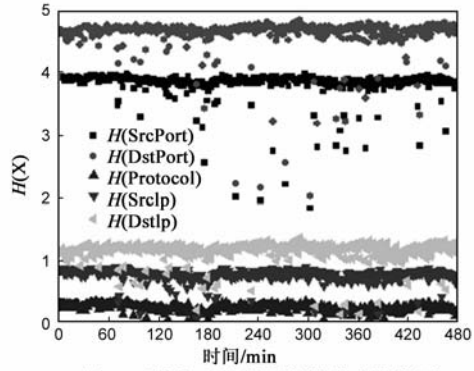

图1 五个主要分类域熵值分布比较图

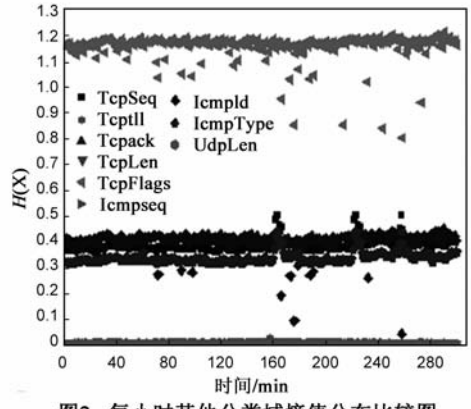

图2 每小时其他分类域熵值分布比较图

总结图 1 和图 2 的结果,得到表 1——分类域熵的取值范围表.实验结果说明①对于流量而言真正能起到分类作用的是取值 0.5 以上的 5 个分类域,它们熵取值大小顺序几乎不随时间变化而变化,说明它们分类能力差别明显.其大小排列为 $H(\mathrm{DstPort})>H(\mathrm{SrcPort})>H(\mathrm{DstIp})>H(\mathrm{TcpFlags})>H(\mathrm{SrcIp})$.其中比较特殊的是 $H(\mathrm{DstIp})$和 $H(\mathrm{TcpFlags})$取值都在 1.1 ~ 1.2 之间,两个分类域取值几乎重合,但是 TcpFlags 突变点较多,因此取 $H(\mathrm{DstIp})>H(\mathrm{TcpFlags})$.②其它分类域的分类能

力都不强. TcpSeq、IcmpType 和 IcmpId 都集聚在 0.3 到 0.4 之间,取值大小交替变化,分类能力差别不大. TtlCheck、TCPLen 和 UDPLen 都集聚在 0.00x 左右,取值曲线几乎完全重叠,分类能力几乎相等.特别地 TCPAck 和 ICMPSeq 的熵值数小时的取值几乎全为 0 的,说明大部分时间实验中所有的流量都只访问了一个分类节点,说明 TCPAck 和 ICMPSeq 几乎不具备分类能力.

在构建分类树时,如果每个分类层次都选取能最大限度划分当前流量的分类域来分类,则可以保证 w_i 的最小性,即达到了 $\sum_{i=1}^{n} w_i | R_i |$ 最小的目标,生成的分类树就是能将当前流量划分得最均匀的最优分类树.

WIND[7]指出由于流量动态变化,此刻最优的分类树也许下一时刻就不是最优的,因此最优分类树存在自适应更新的问题,但并未对自适应更新问题进一步讨论.上述分类熵的取值分布变化说明了骨干网分类树自适应更新周期合理的时间间隔应以天计算.主要有三个原因:①图 1、2 的实验结果证明了对实际流量分类有意义的分类域有 5 个,按分类能力大小排序相对恒定:H(DstPort) > H(SrcPort) > H(DstIp > H(TcpFlags) > H(SrcIp).我们未显示的实验结果表明大多数情况下 24 小时内上述排序基本恒定.②剩下的分类域取值相近,尽管随着流量变化它们的分类域大小顺序会随之变化,但由于它们分类能力差且相差不大,既使保证它们的实时更新,分类树的 $V(P)$差别也不大.③分类树更新的代价较大,即使在 1Gbps 的链路下更新时间只有 1s 都会造成平均 200K 个报文的检测延误.因此兼顾更新效果和代价两方面因素考虑,自适应更新分类树的时间间隔以天为宜.

4　改进 2——利用流长重尾特性优化分类查找算法

上一节的改进措施保障了分类树结构的最优性,要进一步提高分类算法速度,首先必须明确其瓶颈所在.分类算法主要涉及两种操作:①分类树节点查找.一个报文 p_i 在分类树中都依据分类域取值不同会访问一条确定的分类树分支,对应了 w_i 的代价;②分类节点的规则号拷贝.p_i 每访问一个分类节点,就会把分类节点包含的所有规则号作为标签缀在 p_i 后面,当查找结束将 p_i + 规则标签拷贝至公共缓冲区,以供后继检测使用.这个访问代价正比于 $| R_i |$,匹配的规则越多,拷贝需要的代价越大.这两种操作中①只涉及 CPU 消耗,机器指令执行速度快,而②涉及内存的多次拷贝,相较于①,操作②显然是算法速度的瓶颈所在.因此要想提高报文查找的速度,必须减少②的拷贝代价.

上节中对各种分类域熵的实验发现各种分类域的取值都偏小,流量对于各种分类域的访问都呈现出高度汇聚的趋势.比如 DstPort 取值范围为(0 ~ $\log_2$142),而实验中 DstPort 取值为四点几,同 Srcport 取值范围为(0 ~ $\log_2$68),而取值为三点几.说明了即使面向均匀分类流量的目标,采取最大分类域熵的方法构建了最佳分类树,流量对分类树的访问仍然极其不均衡,大多数流量只访问有限的分支.理论分析得知这种不均衡性其实是 IP 流长服从重尾分布的表现.相同的流因为五元组信息(源/宿地址、源/宿端口、协议)一样,因此五层分类过后,流中所有报文都会访问相同的中间节点.再加上除了 TCPFlags 外其他的分类域分类能力极弱,在这些分类域分类的节点中同一个流的所有报文几乎都只访问一个分类节点.只有极少情况下 TCPflags 特殊标志会导致同一流中少量报文访问其他的分类树分支,激活不一样的最终分类树节点,除此之外几乎流中所有报文都将访问分类树的同一分类树终结点.

既然同一个流中几乎全部报文都会访问分类树中相同的路径,需要拷贝的规则集标签也一样,据此可以优化规则标签的拷贝策略.对同一个流 F_i 中的所有报文只拷贝一次规则标签,而不是像迄今为止所有的报文分类算法那样为每个报文多次拷贝重复的相同规则标签.如此可以大幅降低系统的内存拷贝次数和时间,克服查找瓶颈,提高报文查找速度.新的分类树查找算法 FCS 描述如下.

FCS 使用 MultiBloom Filter[9,10]实时计算流长,其优点是计算速度快、联合多个 hash 进行检验计算精度高,是一类广泛应用的实时流长统计方法.为了保证算法精度我们采取三级 MultiBloom Filter 计算流长,其技术细节不再赘述.本算法中用五元组定义一个流,为了保证查找速度,长流表采用 hash 表 + 链表结构存贮长流信息.

分类报文查找算法 FlowCopySearch

I:报文 P_i;

O:指针 p_rules,p_rules 是指向公共缓存中存储报文 P_i + 标签集合 R_i 的一个位置指针.

算法:

① F = hash(P_i[SrcIP, DstIP, SrcPort, DstPort, Protocol]);

②Lengthof(F) = MultiBloom(P_i);//通过 MultiBloom Filter 实时计算 P_i 所属流 F 的流长 Lengthof(F);

②if (Lengthof(F) < 长流阀值) { //P_i 属于短流

　R_i = Old_P-Hicuts(P_i, root);

　p_rules = Copy(Q, P_i + R_i);//当查找至分类终结点时将 P_i + 标签集合 R_i 拷贝入短流公共缓存区 Q 中;

　return(p_rules);

```
}
Else{
    If(Lengthof(F) = = 长流阀值){
    Ri = Old_P-Hicuts(Pi, root);
    Li = FCScopy(L, F, Pi + Ri);// Pi + 标签集合 Ri
    拷贝入长流公共缓存区 L 中,其 FlowID 为 F,位置
    指针为 Li;
    //Pi 属于长流 F,为 F 在长流 hash 表中创建长流
记录:
    longFlow[hash(F)].FlowID = F;
    longFlow[hash(F)].p_middleNode = 分类树中间节
点指针;
    longFlow[hash(F)].p_finalNode = 分类树终节点指
针;
    longFlow[hash(F)].p_rules = 规则标签在公共内存
中的指针 Li;
    p_rules = Li;
    return(p_rules);
    }
    Else{//Pi 是长流 F 的后继报文
    //从中间结点往下查找 Pi 的分类树终结点 final_
pointer;
    final_pointer = SearchFromFlowMidNode(Pi, longFlow
[hash(F)].p_middleNode);
    If (final_pointer = = longFlow[hash(F)].p_finaln-
ode){
    //无需再次拷贝规则标签,Pi 报文对应的规则标
签已经由其前面的报文拷贝入流共享内存缓冲池中
    p_rules = longFlow[hash(F)].p_rules;
    }
    Else{//产生回溯,Pi 和流中的其他报文访问的终
结点不一样
    Ri = Old_P-Hicuts(Pi, longFlow[hash(F)].p_mid-
dleNode)//按 P-Hicuts 的传统搜索方法从 Pi 所属的流
长流 F 中的中间节点记录 longFlow[hash(F)].p_node
开始查找分类树,将遍历的每个分类树中节点的规则
号集合拷贝至 Pi 的规则标签 Ri 中,
    p_rules = Copy(Q, Ri)//当查找至分类终结点 Pi +
标签集合 Ri 拷贝入报文公共缓存中;
    return(p_rules);
    }
  }
}
```

5 FlowCopySearch 的算法复杂度分析

空间复杂度 算法的空间占用主要来自四方面:①MultiBloom Filter;②分类树空间占用;③长流表的空间占用.④共享内存的使用.对于 1Gbps 链路,按平均每个报文 500 个字节计算,一分钟时间片下最大活动报文数 $k \approx 1.2 \times 10^7$;最大活动流数 n 的数量级为 10^6,每一个 Filter 数组的 hash counter 数目为 b;为了保证流长统计精度,$n \ll b$,因此取 24 位 hash 函数,三级 MultiBloom Filter 空间占用为 $3 \times 2^{24} = 48$MB;②的分类树空间占用约为 50MB 左右.③的空间占用为 1 级 Bloom Filter 的空间占用 16MB.由于骨干网流量下通常流长 > 10 的长流只占不到 1/5,实验中长流的数量级为 10^4,每个长流记录占用 32B,因此长流记录平均占用 320KB.④改进后共享内存空间划分为长流报文内存空间和短流报文内存空间,长流报文内存空间以 FlowID 作为索引,为每个流存储一个规则标签,但是为了保证查找速度使用 hash 表存储方式,因此引入了一个 Bloomfilter 的空间占用 16MB,改进后空间复杂度为 $O(n)$ + 16MB.而短流报文内存空间仍旧维持原来的每个报文一个规则标签,空间复杂度为 $O(k)$.因此 FCS 大幅减少了共享内存空间的占用,弥补了它消耗大量内存在判断流长和长流信息存储消耗中的不足.

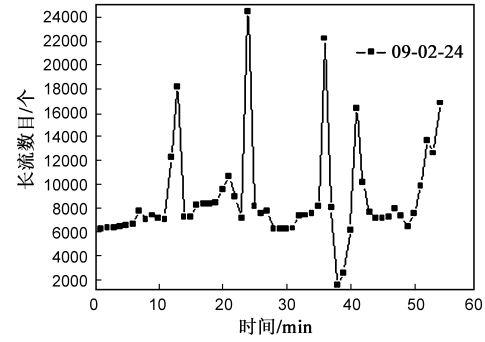

图3 骨干网每分钟长流数目的分布统计图

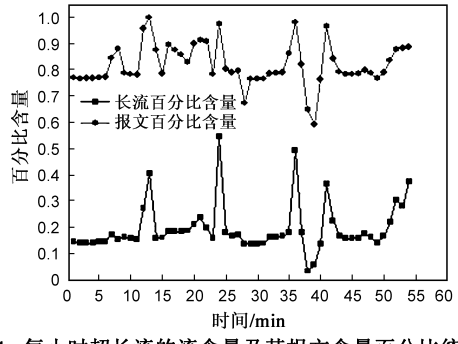

图4 每小时超长流的流含量及其报文含量百分比统计图

时间复杂度 采用 MultiBloom Filter 实时计算流长复杂度为 $O(1)$.设分类树深度为 d,对于短流报文其查找没有任何改进,其最坏查找复杂度仍旧维持 $O(d)$;改进后对于大部分超长流报文的最坏查找复杂度为 $O(d)$-5;而且节省了每个分类树节点的内存拷贝操作时间大大缩短了查找时间.对于极少部分的超长流报

文产生的回溯情况,产生的多余代价就是多访问了几层分类节点,且由于这些访问不需要拷贝规则号,实际上只浪费了少量的 CPU 指令时间,对性能产生负面影响较小.

算法误差分析　算法误差由 Bloom Filter 引入.误差存在于流长统计.MultiBloom Filter 的一大优点是错误否定率为 0,即不可能漏判长流,或则将长流判为短流.其错误肯定率由于采用了独立的 Bloom Filter 空间,也会随着 Filter 级数的增加指数级减小.采用 3 级 Multi-Bloom Filter,错误肯定率 False Positive Rate(FPR) = $(10^6/2^{24})^3 \approx 2.5 \times 10^{-4}$.即平均 10000 个流中有 2、3 个短流会被错判为长流.

算法有效性分析　算法主要的改进在于节省了长流报文的内存拷贝次数和时间.因此改进的效果取决于流量中长流的数量以及长流报文在总报文数中的百分比含量.长流的数量越少,长流表信息维护的开销越小;长流包含的报文越多,算法节省了内存拷贝次数和时间越多,分类算法的速度提高越明显.而流长的重尾特性[11]保证算法的有效性.流长的重尾特性是指短流数量虽多,但是承担的实际网络负载却很小;长流数量虽然少却担负了大部分的网络实际负载.同文献[11]我们将长流阀值取为 10,图 3 显示了华东北地区网络中心 2.5Gbps 链路上 09-02-24 日 13:00 ~ 14:00 一小时 trace 每分钟长流数目的分布.如图所示每分钟长流数目变化很大($1500 \sim 2.5 \times 10^5$),大多数情况下每分钟约有 $6 \times 10^3 \sim 8 \times 10^3$ 条左右长流.所以长流 hash 表发生冲突的平均概率为 $8 \times 10^3/2^{24} \approx 0.048\%$,保证了长流报文查找长流 hash 表的复杂度为 $O(1)$.图 4 显示了图 3 的 trace 中长流的流百分比含量和报文百分比含量对比图.如图所示,长流的流百分比含量在变化范围在 3% ~54% 之间变化,但是绝大多数时间片长流含量在 10% ~ 20% 之间,而其所含报文的报文百分比含量在 50% ~ 99% 之间.实验结果验证了少量的长流产生的报文担负绝大多数的网络实际负载,平均每分钟 FCS 只需要处理 $< 10^4$ 个长流流信息,就可以提高平均 80% × 10^6 个报文的分类处理速度.

6　实验讨论

实验机采用两台 Intel Xeon dual 3.06GHz,2GB 的服务器.一台作为流量发生器,一台作为报文分类机.分别采用两种报文数据源作为模拟流量输入,一种是 DARPA1999 第四周的公开测试数据集,另一种是华东北地区 10Gbps 骨干网链路上 1/4 的负载均衡 trace,相当于一条 2.5Gbps 链路,实际流量均值约为 1.2Gbps.尽管 DARPA 和 trace 的速率不一样,流量发生器会以 150Kpps 的固定速率播放模拟流量,模拟平均 600Mbps 的流量并比较了此时三种报文分类算法 FCS、P-Hicuts 和 WIND 的报文分类速度.采用处理每 10K 报文所用去的平均分类时间来衡量分类速度,所有实验结果采用 5 次实验平均值.实验采用 Snort 最新的官方规则库 3.1.5,包含了 2579 条规则,分类域采用 2.2 节所提到的 14 个分类域.

图 5 显示了三种分类算法的处理速度比较图.它们分别是 P_Hicuts、WIND 和 FCS.以每处理 10K 个报文消耗的时间作为算法速度的衡量.如图所示,首先,无论使用 DARPA 数据集还是骨干网 trace,FCS 的报文分类速度都优于其它两种报文分类算法.相较于 P_Hicuts,FCS 的分类速度提高最大的一天是 09-02-25,速度提升幅度为(16698-9361)/16698 = 43.9%,最小发生在 09-02-24,速度提升幅度为(16784-11772)/16784 = 29.7%.相较于 WIND,最大速度提升幅度为(16742-9197)/ 16742 = 45.1%,最小为(15439-13872)/ 15439 = 10.1%.其次,FCS 对于 trace 的报文分类优化作用显然优于它对于 DARPA99 数据集的作用.FCS 对 DARPA99 的处理速度是每 10K 个报文需要 13872us,而对于 trace 平均 10K 报文只需要 10000us 左右.这是因为 DARPA 的公开测试数据集是 1999 年的人工合成数据,因为模拟大规模攻击其短流含量较之正常情况普遍高很多,长流含量偏低,长流报文含量也低,因此不能充分体现 FCS 的优化作用.FCS 对骨干网 trace 的优化效果明显也说明了改进 2 策略的有效性.再次,WIND 对于 DARPA 数据集的分类速度明显快于其对于骨干网 trace 的分类速度.这是因为 DARPA 数据集数据都集中在一个 C 类网络中,数据量小,所以当 WIND[7] 阀值设置为 5,即为所有能排除 5 个规则以上的分类域特殊值建立独立的分类节点,此时 WIND 的处理速度快于 P_Hicuts,慢于 FCS,分类速度为每 10K 报文 15439us.但是骨干网 trace 数据量大,如果仍保持阀值设置为 5 将导致 WIND 建立的分类树空间占用过大,骨干网情况下必须扩大 WIND 阀值,出于分类树空间不能过大考虑,阀值至少为 500,此时其分类速度随着阀值的增加下降很快.分类速度几乎和 P_Hicuts 没有什么区别,有时甚至慢于 P_Hicuts,所以 WIND 适合局域网小规模报文分类,并不适合大流量情况.综上,三种算法中 P_Hicuts 的表现最为稳定,分类速度始终维持在 1.6×10^4 数量级上,差别很小,而 FCS 和 WIND 则随流量不同分类速度变化较大.大体上可以认为 FCS 最快,P_Hicuts 次之,WIND 最慢.

图 6 显示了采用 trace 和 DARPA 数据集两种测试数据下三种报文分类算法的运行时空间占用比较图.如图所示,骨干网 trace 下 FCS 和 P_Hicuts 的平均内存占用分别为 104.37MB 和 117.52MB,FCS 较之 P_Hicuts

节约了 11.1%的内存.尽管 FCS 将共享内存的占用复杂度由 $O(k)$降到了 $O(n)$,其中 k 为活动报文数,n 为活动流数.但是为了保证查找速度,FCS 大量采用了 hash 存储结构,所以其总内存占用总数并不比 P_Hicuts 少多少.FCS 的另一个缺点在于当流量中短流含量高时(如 DARPA 数据集),FCS 的内存占用直线升高,甚至比 P_Hicuts 高了 8.7%.在 DARPA 小流量中 WIND 的内存占用是所有情况中最小的(82MB),但是对于骨干网大流量情况 WIND 会因为特殊值分类节点过多导致空间占用迅速增大,即使阀值设置增大到 200,其内存空间占用仍然达到所有情况中最大值 143MB.

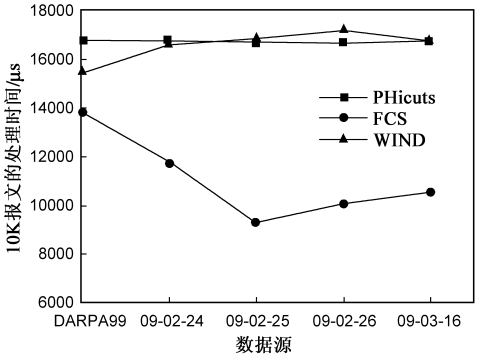

图5 三种报文分类算法的分类速度比较图

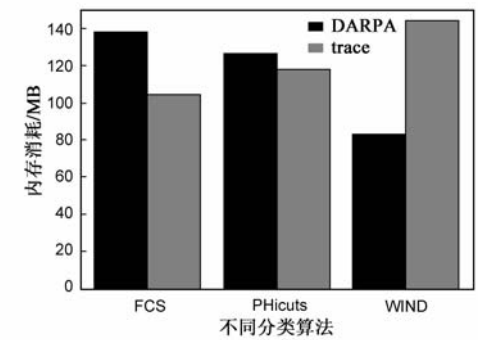

图6 三种报文分类算法的内存占用比较图

综上,无论使用人工合成数据还是网络真实流量作为数据源,FCS 的报文处理速度都是最快的,较于不考虑动态流量优化的静态分类算法 P_Hicuts 速度提升了 29%~43%,相较于为特殊属性值构建分类节点的 WIND, FCS 速度提升了 10.1%~45.7%.但是采用 DARPA 公开数据集的实验结果表明由于其人造攻击数据集中短流含量高,抑制了 FCS 的优化作用,FCS 较之两种算法的速度提高幅度分别降至 17.3%和 10.1%,且 FCS 内存占用直线上升.说明 FCS 算法的缺点在于性能受流量中短流含量影响较大,短流越多,FCS 占用内存越多,速度提升越小.在骨干网 IDS 更注重报文处理速度的需求下,综合性能来说 FCS 仍然是三种算法中最优的.P_Hicuts 的性能最为稳定,WIND 适用于局域网小流量下的报文分类,它在骨干网 trace 下的优化能力不佳也说明了光靠改进分类树结构来提高报文分类速度的方法能力有限.

7 总结

本文充分利用了访问分类树的主体——动态流量的特性来改进以往完全静态的分类算法.改进分为两个部分:其一,优先使用分类能力强的分类域作为分类树的分类层次测度,构造出能将流量划分得最均匀的最佳分类树.其二,利用流长的重尾特征将每个报文都不得不进行的规则集拷贝缩减到对属于同一个流的大部分报文只要进行一次规则集拷贝,克服了报文分类速度瓶颈.实验结果证明 FCS 在骨干网大流量 trace 下表现优异,较之其它两种经典分类算法能在提高分类速度的同时降低内存消耗.

参考文献

[1] 田立勤,林闯.报文分类技术的研究及其应用[J].计算机研究与发展,2003,40(6):765-775.
TIAN Li-qin, LIN Chuang. Study and application of packet classification[J]. Journal of Computer Research and Development, 2003, 40(6): 765-775. (in Chinese)

[2] LAKSHMINARAYANAN K, RANGARAJAN A, VENKARACHARY S. Algorithms for advanced packet classification with ternary CAMs[A]. SIGCOMM'05[C]. Philadelphia, USA: ACM Press, 2005. 193-204.

[3] GUPTA P, MCKEOWN N. Packet classification on multiple fields[A]. ACM SIGCOMM'99[C]. Canbridge: ACM Press, 1998. 147-160.

[4] 陆晟,龚俭.一种新的高维报文分类算法——无相交树算法.计算机学报,2003,26(11):1505-1509.
SHENG Lu, GONG Jian. A multi-dimension packet classification algorithm: nonIntersection trie[J]. Journal of Computers, 2003, 26(11): 1505-1509. (in Chinese)

[5] 龚俭,魏薇,周鹏.适用 GIDS 报文分类的 P-Hicuts 算法[J].哈尔滨工业大学学报,2008,40(3):448-452.
GONG Jian, WEI Wei, ZHOU Peng. P-Hicuts algorithm for GIDS packet classification[J]. Journal of Harbin Institute of Technology, 2008, 40(3): 448-452. (in Chinese)

[6] 马亚鸣,龚俭,杨望.高速报文分类算法 P-Hicuts 的改进[A].第十届海峡两岸网络技术交流会[C].南京:东南大学出版社,2009.351-357.
MA Ya-ming, GONG Jian, YANG Wang. Improved implementation of fast packet classification algorithm P-Hicuts[A]. Strait Network Technology Exchange Symposium09[C]. Nanjing: East South University Press, 2009. 22-28. (in Chinese)

[7] SINHA S, JAHANIAN F, PATEL M J. WIND: Workload-aware intrusion detection[A]. Recent Advances in Intrusion

Detection 2006[C]. Hamburg, Germany: Springer Press, 2006. 290 – 310.

[8] BRAUCKHOFF D, TELLENBACH B, WAGNER A. Impact of packet sampling on anomaly detection metrics [A]. SIGCOMM'06 [C]. Brazil: ACM Press, 2006. 159 – 164.

[9] ESTAN C, VARGHESE G. New directions in traffic measurement and accounting [A]. SIGCOMM' 2002 [C]. Karlsruhe, Germany: ACM Press, 2003. 270 – 313.

[10] 谢鲲,赵姣姣,张大方,毕夏安. 基于计数布鲁姆过滤器的快速多维包分类算法[J]. 电子学报, 2010, 38(5): 1046 – 1052.
XIE Kun, ZHAO Jiao-jiao, ZHANG Da-fang, BI Xia-an. A fast multi-dimensional packet classification algorithm using counting Bloom filter[J]. Acta Electronica Sinica, 2010, 38(5): 1046 – 1052. (in Chinese)

[11] 周明中. 大规模网络 IP 流行为特性及其测量算法研究[D]. 南京: 东南大学, 2006.
ZHOU Ming-zhou. Study of Large-scale Network IP Flows Behavior Characteristics and Measurement Algorithms[D]. Nanjing: School of Computer Science and Technology, 2006. (in Chinese)

作者简介

宁　卓　女, 1975 年 10 月出生于贵州省贵阳市, 博士. 现为南京邮电大学物联网学院讲师, 主要研究领域为网络安全、网络行为学.
E-mail: ningz@njupt.edu.cn

孙知信　男, 1964 年 9 月出生于安徽省宣城市, 博士. 现为南京邮电大学物联网学院教授, 博士生导师. 研究方向为: 计算机网络与安全、多媒体通信, 移动互联网等.
E-mail: sunzx@njupt.edu.cn