基于最大属性熵的 GIDS 报文分类算法[1]

宁卓[1,2] 龚俭[1,2]

（1.东南大学计算机科学与工程系 江苏 南京 210096
2.江苏省计算机网络技术重点实验室江苏 南京 210096）

摘要: 网络带宽的激增对网络入侵检测系统（NIDS）的检测速度提出越来越高的要求。分类算法作为一种有效降低数据包待匹配规则集的方法，其效率对后继检测算法影响重大。本文研究了适用于 GIDS 的经典分类算法 Hicuts 和针对它的修改升级算法 Picuts，针对 Picuts 没有考虑报文域的特征对于分类树的影响的缺点提出了基于最大属性熵的分类树本地优化策略和新的分类树生成算法 MaxFeatureEntropy。并介绍了计算 IDS 规则属性熵的计算方法。最大属性熵从理论上保证了减小决策树高度。采用开源的 snort1.8.3 的规则集作为实验数据，结果表明：当每结点包含规则数阀值等于 6 时，其空间消耗只有 Hicuts 的 10%， Picuts 的 60%，速度上较之 Hicuts 提升了 44.4%，较之 Picuts 提升了 20%。

关键词: 属性熵、报文分类、分类树、GIDS

中图分类号:

Packet Classification Algorithm Based on Maximum Feature Entropy Used in GIDS

Ning Zhuo[1,2] Gong Jian[1,2]

(Department of Computer Science and Technology, Southeast University, Nanjing 210096)

Abstract: The heavy workloads of Gigabit Intrusion Detection System make the packet classification algorithm critical to its performance. However, unfortunately the problem of creating a minimal decision tree that is consistent with a set of data is NP hard. Based the former research[1,2], we proposed an new algorithm MaxFeatureEntropy to perform local optimization by choosing the most discriminating feature which has the most high entropy when creating the rule decision tree in GIDS. The method of evaluating the feature entropy of rules is also discussed. The experiment results show that comparing to Hicuts and Picuts, the performance of MaxFeatureEntropy promotes 44.4% and 20% respectively, and it's memory consumption is 10% of Hicuts, and 60% of Picuts.

Keywords: feature entropy; decision tree; packet classification; GIDS

1 研究背景

当前入侵检测系统（IDS）作为一种不可缺少的安全管理工具越来越为人们所接受。它可以实时提供网络中存在攻击的频率和性质，包括攻击类型、攻击危害、攻击起止时间等等，有利于管理员采取响应措施。随着网络带宽越来越大，网络包到达速率的激增对 IDS 检测速率的要求也越来越高。据 2004 年 Dataquest 统计约有 14%的主干路由已经达到了 OC-768 标准(40 Gbps)，21%边界链路达到 OC-192 标准(10 Gbps)。GIDS（Gigabit Intusion Detection System）要在线线速处理如此高速率的流量，对其检测算法要求很高。显然对每个数据包都逐次就这 1239 条规则进行检测

[1]**收稿日期:**

基金项目: 本文得到国家973计划课题（2003CB314804）；教育部科学技术重点研究项目（105084）；江苏省网络与信息安全重点实验室（BM2003201）资助。

作者简介：宁卓（1975－），女，东南大学博士生，主要研究方向为网络安全检测，Email：zhning@njnet.edu.cn。
龚俭（1957－），男，教授，博士生导师，主要研究方向为网络安全，网络管理和网络体系结构等。

是相当低效的。报文分类算法试图为滥用 IDS 提供一种方法使得对于到来的数据包只要经过最少的比较次数就可以确定数据包激活的待匹配规则集，以提高检测效率。理论上已经证明，规则总数为 N，分类报头域数量为 K 的报文分类算法存在两个极限：时间最优算法的时间复杂度是 $O(\log N)$，空间复杂度是 $O(N^K)$；空间最优算法的时间复杂度是 $O(\log^{k-1} N)$，空间复杂度是 $O(N)$。设 K=4，snort1.8.7 的时间最优算法需要内存 1T，空间最优算法内存访问次数约为 954.2。

这两个值对于现有的计算资源要求都过高，因此在实际算法设计中我们既要考虑到算法的速度问题，又要力求在时间和空间消耗中找到一个折中点。1999Gupta 和 McKeown[1]提出了一种灵活的报文分类算法 Hicuts，提出分层检测的思想，将 rule-to-rule 的检测改为 feature-to-feature。以每个待分类的报头域为一个层次，将报文空间逐层等距分组，生成报文分类决策树。当报文到达时，遍历决策树找到一个与之匹配的存储少量规则的节点，再使用线性查找算法，找到匹配的规则。但是该算法在 NIDS 应用中存在着空间异常膨胀和决策树不平衡问题。[2]对上述 Hicuts 算法的缺点提出了两点改进：1）预编译生成决策树时将覆盖规则上提，不再参与分组，以此抑制由该类规则引起的空间指数膨胀；2）采取非均匀分组的方法，以各规则在属性域上取值范围的边界值作为切分点，从而使分组后每个区间的规则集基数趋于平均。尽管[2]提出了报文分类算法的时间复杂度正比于分类树的高度，但是[2]没有讨论什么样的分类树是最优的，如何构造最优分类树，也没有考虑报文域的特征对于分类树的影响。

2 问题的提出和解决

求解最优分类算法实质上就是求解最佳分类树。我们给出问题的形式化定义如下：

定义一 分类树的势 一棵分类树的势定义为：$\sum_{i=1}^{n} w_i * |R_i|$，其中 w_i 是根接点到第 i 个结点的路径长度，R_i 表示第 i 个结点中存储的规则子集，$|R_i|$表示规则子集的基数，n 是所有结点的数目。即分类树的势是其所有结点的规则子集的基数与其到根结点的路径长度乘积的和。

定义二 一棵分类树称为最佳的当且仅当其分类树的势是所有分类树中最小的。

不幸的是对一个确定数据集求解最佳分类树的问题显然是一个 NP 难题。而我们试图构造出一种次优的分类树满足 GIDS 的检测要求。我们的思想是在每一步分类中进行本地最优化选择，每一步都选择数据集中检测代价最小的特征对规则集分类，并且这个特征使得对当前数据集的划分个数最大，分得最均匀。分类过程循环迭代直至待分类的报文域为空或分类树结点包含的规则子集大小已经小于阀值。这个过程中如何找出这样的特征是关键所在。

2.1 规则属性研究

入侵规则库由若干规则组成，每一条规则代表了一种攻击。每条规则分别由若干属性特征 $f_1, f_2, \ldots f_n$ 组成，每个特征的取值范围为 $V_1, V_2, \ldots V_n$。按特征的检测代价不同大致有两类特征：一类称为标识特征，一类称为负载特征。研究 snort 规则库可知前者都为报头域属性，多为整数型、地址型（IPV4 或 IPV6）和位操作，而负载特征多为字符串型。显然后者的检测代价高于前者。Snort 中存在的报文域有几十种：协议类型 Type、源地址 Saddr、宿地址 Daddr、源端口 sport、宿端口 dport、flow、Msg、content、reference、classtype、sid、rev、depth 等等，除了前五个特征（经典五元组）作为分类特征外，我们发现一些协议选项特征也是极好的分类特征，如 6）ICMP code；7）ICMP type；8）TCP flags。此外还有 9）流方向；10）flow。我们的算法中将上述 10 个特征作为待分类报文域。但是 snort 中报头域特征数量为 2~3 个的规则占总数的 88%，我们仍然采用 Picuts 的覆盖规则上提法和非均匀分组克服此现象带来的空间膨胀问题。

2.2 量化属性熵

为了构造出一棵优化的分类树，我们借助熵理论的成果。设 X 是一个数据集，$X=\{aaa,bbb,...\}$是数据分类集合，在数据集 X 中每个数据项 x 都属于一个类，即 $x\in C_x$，则 X 相对于这 C_x 个分类的熵定义为：$H(X)=\sum_{x\in C_x}P(x)\log\frac{1}{p(x)}$，其中 $P(x)$ 是 x 在 X 中的出现概率。当类的分布不均衡时，也就是当数据比较单一时，熵的值就较小，相反当类分布比较均匀时，也就是当数据类型较杂时，熵的值就较大。在入侵检测中 X 特指整个规则库，每一个规则都独立地代表了一个攻击，自成一类。且每一个攻击的出现都是等概率的。所以规则库的熵为：

$$H(X)=\sum_{i=1}^{n}-\frac{1}{n}\log_2(\frac{1}{n})=-\log_2(\frac{1}{n})=\log_2(n) \qquad (1)$$

其中 n 是 X 中总规则数。同样我们用下式衡量待分类报文属性 A 对规则集 X 的信息量，称为属性熵。以此刻画报文属性对分类的贡献，可见属性熵越大，分类效果越好。

$$Gain(X,A)=H(X)-\sum_{v\in Values(A)}\frac{|X_v|}{|X|}H(X_v)=\log_2(n)-\sum_{v\in Val(A)}\frac{n_v}{n}\log_2(n_v) \qquad (2)$$

其中 $Values(A)$ 是规则集中A 的一组可能的值，X_v 是当属性A的值为v时X的一个子集。|Xv|和|X|分别代表规则子集 X_v 和规则集X的集合大小，分别等于n_v和n。按式（2）我们可以找出一个特征顺序，它对规则集的信息量由大到小排列，熵越大说明类的分布越平均，完全满足GIDS的要求。

3 报文分类算法

报文分类算法包括用于生成分类树的生成算法和用于检测的搜索算法，在此我们仅提供改进了的生成算法 MaxFeatureEntropy，搜索算法同[2]。其中 Feature 表示分类特征集合，nodes 表示待分类结点集合，R（node）表示结点 node 包含的规则子集。

输入：snort 的规则集合
输出：优化的分类树 DecisionTree
初始化 *Feature* = 2.1 节介绍的 10 个特征；
新建结点数组 *nodes，nodes[0]=input；nodesNum = 1*；
DecisionTree 指向 nodes；
Feature 按式（*2*）求出的属性熵的值从大至小排列；
i=0;
while(Feature 不空 *or R(worknode)*<阀值*)*
{
MaxFeature=Feature[i]；
Feature = Feature – MaxFeature；
For(j=0;j++;j< nodesNum) {
worknode = nodes[j];
计算结点 *workNode* 在 *MaxFeature* 特征上的覆盖规则集合 *Ronr*(workNode)，保存到结点 *child* 中。计算 *R(workNode)*与 *Ronr*(workNode)的差集 *R'*(workNode)；
清空集合 *P*，取 *R'(workNode)*各规则在报头域 *d* 上的取值范围的边界值插入到 *P* 中。完

成后 *P={e[1],. .e[i]. ., e[nc'+1]}, e[i]* 表示 *P* 中的点，*nc'+1* 为 *P* 中点的个数；
以 *P* 作为 *d* 的切分点集，切分后形成连续区间集 $L=\{S_1,..,S_{nc'}\}$，其中 $S_j=[e[j], e[j+1]]$，*(1≤j≤nc')*；
创建 *nc'*个 *workNode* 的子节点 *(1≤t≤nc')*，为每个计算其 $R(C_t)$；
为 *workNode* 和其孩子结点 *child* 和每个 C_t 建立父子关联；
}
nodes<- C_t; nodeNum = nc' +1;
i++;
}
Return DecisionTree;

4 试验

理论上分类树的高度小于等于报文属性域的个数 D，分类树结点的最大分组数与规则总数 N 成正比。表 1 是在 snort1.8.7 规则集上使用三种分类算法（Hicuts、Picuts、MaxFeatureEntropy）生成的分类树时空性能比较。理论上三种算法的最坏时间复杂度都是 O(D)，空间最坏复杂度为 $O(N^D)$。但是在实际中可以看到 MaxFeatureEntropy 和 Picuts 由于采用了覆盖规则上提和非均匀切分技术较之 Hicuts 性能有很大提升。较之 Hicuts，MaxFeatureEntropy 的空间消耗约占 Hicuts 的 10%，由于分类速度正比于分类树高度，由试验数据可知分类树最大高度由 9 降低至 5，(9-5)/9=44.4%。分类速度几乎提高一半。较之随机挑选属性域作为分类标准的 Picuts，MaxFeatureEntropy 的优化作用也很明显。如表 1 所示当一个结点所允许包含的规则数不多于 4 时，其空间消耗降低了(21.4-13.1)/ 21.4=38.7%，分类树高度降低了 1，分类速度提升了 1/6=16.7%；当规则数阀值为 6 时空间消耗提升了(19.7-12.34)/19.7=37.5%，分类树高度降低 1，分类速度提升了 1/5=20%。

表 1　三种分类算法的性能比较结果

阀值	空间消耗（Hicuts）	空间消耗（Picuts）	空间消耗（MaxFeatureEntropy）	分类树高（Hicuts）	分类树高（Picuts）	分类树高（MaxFeatureEntropy）
4	192.2 (Mbytes)	21.4 (Mbytes)	13.1 (Mbytes)	3-9	4-6	2-5
6	139.5 (Mbytes)	19.7 (Mbytes)	12.3 (Mbytes)	3-8	3-5	2-4

5 总结与展望

本文提出了基于最大属性熵的分类算法。采用本地优化策略，分类的每一步都采用具有最大属性熵的特征来划分，试验结果表明生成的分类树的性能较之 Hicuts 和 Picuts 有较大的提升。这个方法最显著的特点在于不要求特征是条件独立的，因此可以相对任意地加入对最终分类有用的特征,而不用顾及它们之间的相互影响。不足之处在于属性熵的求取过程中认为各规则等概率出现，而在实践中我们发现有一小部分攻击频繁出现，大部分攻击很少出现，即规则命中率并非是等概率的，且规则频繁集随时间变化而变化。下一代的分类方法应该基于非等概率的属性熵，规则库也不再是静态的。可以以规则的命中率、规则近期活跃程度等因素的加权平均为考量改造静态规则库为动态规则库，开发与其动态调整策略一致的报文分类算法是我们后期的工作。

参考文献

[1]. GUPTA P, MCKEOWN N. Packet Classification On Multiple Fields[A]. Proc. ACM SIGCOMM, Computer Communication Review[C].Cambridge, MA, USA: 1999. 147-160.

[2]. P-Hitcuts: 基于 HiCuts 的 GIDS 报文分类算法，龚俭，魏薇，周鹏，哈尔滨大学学报。（已录用）