# 基于字节熵矢量加权指纹的二进制协议识别*

黄笑言，陈性元，祝　宁，唐慧林
（解放军信息工程大学，郑州 450004）

**摘　要**：协议识别在入侵检测等网络安全领域具有广泛应用。根据二进制协议特点，提出了一种基于字节熵矢量加权指纹的协议识别方法。对不同协议类型的网络流，利用字节熵矢量描述其报文格式属性，基于局部加权的思想对其进行聚类，得到类簇中心及各类簇字节熵的权重分配，构建协议的字节熵矢量加权指纹，通过指纹的距离度量及距离阈值设定对协议进行识别。实验表明，该方法对常见二进制协议的识别召回率达到 94% 以上，并可以发现训练集中未出现的协议。

**关键词**：协议识别；二进制协议；格式属性；局部加权聚类
**中图分类号**：TP309.2　　**文献标志码**：A　　**文章编号**：1001-3695(2015)02-0493-05
doi:10.3969/j.issn.1001-3695.2015.02.038

## Binary protocol identification based on weighted byte entropy vector

HUANG Xiao-yan, CHEN Xing-yuan, ZHU Ning, TANG Hui-lin
(*Information Engineering University of PLA*, *Zhengzhou* 450004, *China*)

**Abstract**: Protocol identification was applied in the field of network security such as intrusion detection. According to the characteristic of binary protocol, this paper proposed a protocol identification method based on weighted byte entropy vector fingerprint. It firstly extracted the byte entropy vector which represented the attribute of the message format from the flow and built the weighted byte entropy vector fingerprint by using locally weighted K-means algorithm. Experiments show that the accuracy of the method for identifying the common binary protocols reach more than 94%, and can find the unknown protocols in the training sets.

**Key words**: protocol identification; binary protocol; format attribute; local weighted cluster

## 0　引言

协议识别对入侵检测系统、协议安全性分析、网络安全监控等方面具有重要应用价值，对其进行深入研究能够有效降低系统面临的安全风险，有助于提供信息系统抵御恶意攻击的能力，具有十分重要的意义。由于网络中大量非标准端口，动态端口和复用端口的使用增加导致早期的基于端口的协议识别技术的准确率大大降低。因此很多研究者开始研究新的协议识别方法。目前的方法分为两类[1]，一类是基于报文格式特征的方法，另一类是基于流量统计特征的方法。

基于流量统计特征的方法运用流的持续时间、包数目等流级别的特征[2~4]和包大小、方向及间隔时间等包级别的特征[5~7]区分不同的协议流量。流级别的特征需要检测整条流的信息，且划分粒度较粗，只能区分应用类型（如 P2P 或非 P2P 流量），无法做到精细的协议类型的识别[8]。包级别的特征可以实现对协议类型的识别，但是要严格依赖包的到达顺序[9]，因此不适于基于 UDP 传输的协议识别。

基于报文格式特征的方法通过检测传输层包负载内容，根据协议规范中的报文格式区分不同的协议，具有细粒度分类协议的能力，并且不区分 UDP 和 TCP 流量。这种方法最大的问题是格式特征的构建。传统的方法通过人工方法从规范中提取特征，但是由于很多协议规范的私有性及不完整性使得人工方法失效，因此有人开始研究自动构建格式特征的方法。

刘兴彬等人[10]采用 Apriori 频繁项挖掘算法获取协议格式规范中规定的关键字作为格式特征，但是这种方法需要不断维护和更新协议的关键字库，因此缺乏灵活性。Haffner 和 Ma 等人[11,12]利用流负载的前 $N$ 个字节值，使用统计模型和机器学习的方法构建协议的内容统计特征。由于二进制协议和文本协议的字节取值特点不同，2009 年 Khakpour 等人[13]通过计算一个数据包负载的信息熵，实现对文本、二进制和加密流量的区分，但无法识别精确的协议类型。赵咏等人[1]针对文本类协议内容的语义特点，在语义空间上对流量进行相似性比较并聚类，自动发现文本类协议特征。

目前的方法大多根据协议在格式规范中规定的常量字节值进行识别，因此在识别某些常量字节值较少的二进制协议时具有很大局限性。2010 年，Finamore 等人[14]利用卡方统计模型，基于比特分组在网络流中取值变化训练 SVM 分类器，对训练集中出现的二进制协议进行区分，但是这种方法并未构建协议的识别特征，以至于无法发现训练集中未出现的协议。

**收稿日期**：2014-01-15；**修回日期**：2014-03-10　　**基金项目**：国家“973”计划资助项目(2011CB311801)；河南省科技创新人才计划资助项目(114200510001)

**作者简介**：黄笑言(1989-)，女，福建福州人，硕士研究生，主要研究方向为信息安全、协议逆向(15138903016@sina.cn)；陈性元(1963-)，男，安徽无为人，教授，博士，主要研究方向为信息安全、分布式操作系统；祝宁(1981-)，男，辽宁抚顺人，讲师，博士，主要研究方向为网络对抗；唐慧林(1980-)，男，安徽枞阳人，讲师，硕士，主要研究方向为信息安全.

本文面向二进制协议,提出了一种基于字节熵矢量加权指纹(weighted byte entropy vector,WBEV)的协议识别方法。该方法利用字节在网络流中的取值变化,为每类协议构建识别特征,可以解决对于常量字节值较少的协议其识别特征构建困难的问题。具体思想是首先从网络流中抽取字节熵矢量作为其格式属性,然后采用局部加权的思想对网络流进行聚类。构建的识别特征不仅包含字节熵矢量统计描述,还包含每维字节熵在识别该协议时的贡献程度,使得贡献程度大的字节熵对协议识别的影响增大。

## 1 基于 WBEV 指纹的识别基本思想

### 1.1 问题分析

协议报文由若干个协议字段组成。根据其在网络流中的取值变化情况,协议字段通常分为三种类型:a)常量值,即字段在网络流中具有固定的值,如协议版本、协议标志和会话 cookie 字段;b)有限值,即字段在网络流中具有有限的几个值,如协议的行为控制字段;c)随机值,即字段在网络流中随机取值,如由加密或压缩算法产生的随机值。协议具有特定的格式规范,规定了报文的字段划分及字段类型,其报文在网络流中具有不同的格式属性,但在缺少格式规范的条件下,对协议字段进行直接划分是一件困难的事[15]。由于二进制协议字段通常以字节为最小单位进行组织[16],且在报文中具有固定位置,因此对于二进制协议报文,字节在网络流中的取值特性反映了字段的取值特性。本文利用字节的信息熵描述字节在网络流中的取值特性,进而描述报文的格式属性,对不同协议类型进行识别。

### 1.2 相关概念

#### 1.2.1 字节熵矢量

**定义 1** 字节熵。考虑一个双向网络流中的 $L$ 个数据包,$x_i$ 为位于第 $i$ 个偏移的字节,其取值是 0 ~ 255,则第 $i$ 个字节熵为 $H(x_i) = -\sum_{n=1}^{255}\frac{a_n}{L}\log\frac{a_n}{L}$。式中:$a_n$ 表示 $x_i$ 在 $L$ 个数据包中取值为 $n$ 的频数,$\sum_{n=1}^{255}a_n = L$。

根据信息熵的概念,熵为随机变量不确定性的度量,它反映了随机变量的分布特性,熵越大,不确定性越大。本文提出的字节熵概念将报文中每个位置的字节看做是网络流中 $L$ 个数据包的随机变量。若字节在网络流中取固定的值,则其不确定性最小,其熵值为 0;若字节在网络流中随机取值,则其不确定性大,最大熵值为 $\log L$;若字节在网络流中取有限的几个值,则其熵值位于前两种熵值的中间。协议报文的前几个字节为协议的消息头部,具有特定的格式规范,同一位置的字节所属字段的取值类型不同,相应的字节熵也会有所差异。由此可见,报文前几个字节的字节熵可以反映协议报文的格式特性,进而作为识别协议的依据。本文利用前几个字节的字节熵度量协议报文格式属性,并据此构建协议的识别指纹。

**定义 2** 字节熵矢量。考虑包负载前 $P$ 个字节熵,由此得到一个网络流的 $P$ 维字节熵矢量 $H_{1\times p} = \{H(x_1), H(x_2), \cdots, H(x_p)\}$。

#### 1.2.2 字节熵矢量加权指纹

以网络流的字节熵矢量作为报文的格式属性,通过聚类将网络流划分成类簇,同一类簇的网络流属于同一协议,每个类簇中心是该类簇中字节熵矢量的均值。由于协议可能具有不同的模式,因此同协议的网络流可能会被划分成多个类簇,即一个协议对应一个类簇集合。

另外,取值特性稳定的字节能够更好地帮助识别该类协议,即每个维度的字节熵在识别协议时的贡献程度存在差异。由于协议格式存在差异,因此就字节熵的识别贡献程度而言,不同协议可能具有不同的字节熵组合。本文在构建协议指纹时对字节熵矢量进行加权,通过权重衡量不同维度的字节对识别该协议的贡献程度,形成每类协议指纹的特定描述。

**定义 3** 类簇指纹。某类簇的类簇指纹 $\varphi = \{u, w\}$。其中:$u = [u_1, u_2, \cdots, u_p]$ 表示类簇中心,即字节熵矢量均值,$w = [w_1, w_2, \cdots, w_p]$ 表示各维度字节熵的权重。

**定义 4** 字节熵矢量加权指纹。某协议的字节熵矢量加权指纹 $\vartheta = \{\varphi_1, \varphi_2, \cdots, \varphi_l\}$。其中 $\{\varphi_1, \varphi_2, \cdots, \varphi_l\}$ 表示该协议对应的类簇指纹集合。

### 1.3 基于 WBEV 指纹的协议识别流程

基于 WBEV 指纹的协议识别方法分为两个阶段,即指纹构建阶段和协议识别阶段。在指纹构建阶段,首先抽取网络流的字节熵矢量作为格式属性;然后对网络流进行聚类,得到每个类簇中心及各个维度字节熵的权重,作为类簇指纹;最后将类簇与协议类型映射,得到每个协议所对应的类簇集合,继而得到协议指纹。在协议识别阶段,对于待识别的网络流,首先抽取它的字节熵矢量,通过计算其与各协议指纹的距离,判断它的协议类型。图 1 表示基于 WBEV 指纹的识别流程。

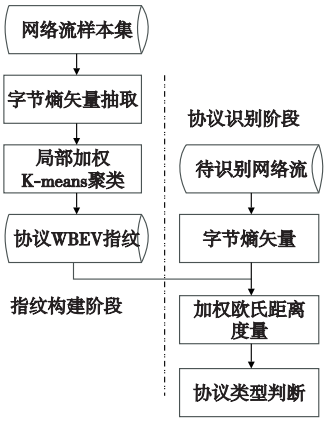


**图1 基于WBEV指纹的识别流程**

## 2 基于 WBEV 指纹的协议识别方法

### 2.1 传统 K-means 聚类

传统 K-means 算法的目标是在样本集 $\{x_1, x_2, \cdots, x_n\}$ 上找到 $K$ 个划分,使得所有样本与对应的类簇中心的距离平方和最小。算法结束后返回一个类簇中心矩阵 $C = [u_1, u_2, \cdots, u_K]^{\mathrm{T}}$,其中 $u_i$ 为第 $i$ 个类簇的中心。

传统 K-means 算法易于描述,收敛速度快,且适于处理大规模数据,但是这种聚类算法采用欧氏距离度量样本与类簇中心的距离,$\| x_j - \mu_i \| = \sqrt{\sum_{m=1}^{p}(x_{jm} - u_{im})^2}$ 没有考虑各个属性维度对不同类簇的识别权重问题。

### 2.2 基于局部加权 K-means 聚类的类簇指纹构建

为克服 K-means 聚类的不足,本文融合局部加权 K-means 聚类的思想[17],提出基于局部加权 K-means 聚类的类簇指纹

构建算法。根据类簇样本在各维字节熵上的分布,选取分布散度较小的维度赋予较大权值。

$w_{im}$表示第 $i$ 个类簇中第 $m$ 个字节熵的权重,$\boldsymbol{w}_i = [w_{i1}, w_{i2}, \cdots, w_{ip}]$为第 $i$ 个类簇中字节熵的权矢量。$\boldsymbol{w}_i$ 满足 $\sum_{m=1}^{p} w_{im} = 1, \forall 1 \leqslant i \leqslant K$。

本文利用 Domeniconi 等人[17]提出的权值求解算法求得权值 $w_{im} = \frac{e^{-Y_{im}/h}}{\sum_{t=1}^{p} e^{-Y_{it}/h}}$。式中:$Y_{im} = \frac{1}{|C_i|}\sum_{x_j \in C_i}(x_{jm} - u_{im}{}^2)$;参数 $h$ 为大于 0 的某个常数,控制类簇中散度对维度权重的影响情况。$h$ 较小时散度对权重分配的影响较大;$h = 0$ 时,将所有权重赋予散度最小的维度;$h = \infty$ 时,不考虑散度对权重的影响,即将权重平均赋予每个维度。本文依据文献[17]对参数 $h$ 取值进行分析的实验结果将 $h$ 设定为 1/3。

引入权重后,样本与类簇中心的欧氏距离扩展为加权欧氏距离:

$$\| x_j - \mu_i \| = \sqrt{\sum_{m=1}^{p} w_{im}(x_{jm} - u_{im})^2}$$

并且需要在 K-means 聚类迭代步骤中增加权重的计算。下面给出了基于局部加权 K-means 聚类的类簇指纹构建详细流程。

输入:$N$ 行 $P$ 维字节熵矢量矩阵$[h_1, h_2, \cdots, h_N]^{\mathrm{T}}$,类簇个数 $k$ 和常量 $h$。

输出:类簇集合$\{C_1, C_2, \cdots, C_K\}$、类簇中心矩阵 $\boldsymbol{u}$、类簇权重矩阵 $\boldsymbol{W}$。

a)利用初始点选择算法选择 $k$ 个中心点$\{u_1, u_2, \cdots, u_K\}$;

b)初始化各个类簇的权重矢量矩阵 $\boldsymbol{w}_{K\times P} = \frac{1}{D}$;

c)将所有网络流划入与其最接近的聚类点中,对于每一个字节熵矢量 $h_j$,选择$\sqrt{\sum_{m=1}^{P} w_{im}(h_{jm} - u_{im})^2}$最小的 $C_i$;

d)计算新的权重:

$$Y_{im} = \frac{1}{|C_i|}\sum_{x_j \in C_i}(h_{jm} - u_{im})^2, w_{im} = \frac{e^{-Y_{im}/h}}{\sum_{t=1}^{P} e^{-Y_{it}/h}};$$

e)重新划分字节熵矢量集合,并计算每个类簇新的中心点;

f)重复步骤 c)~e),迭代进行;

g)经过 $T$ 次迭代后,如果继续迭代每个集合不再发生变化,则聚类结束;得到类簇集合、类簇中心矩阵及类簇权重矩阵。

### 2.3　类簇与协议类型的映射

将网络流划分为类簇后,为了通过类簇指纹获得相应的协议指纹,需要将类簇与协议类型进行映射,即明确每个类簇所属的协议类型。

本文利用简单的启发式规则将类簇与协议类型进行映射。假设 $N_{iv}$是在类簇 $C_i$ 中协议类型 $v$ 的数目,$N_i$ 是类簇 $C_i$ 的数目。$R_{iv} = \frac{N_{iv}}{N_i}$表示类簇 $C_i$ 中协议类型为 $v$ 的样本数目所占比例。类簇 $C_i$ 与协议类型的映射函数为 $\theta(C_i) = \{v \mid \max_v R_{iv}\}$。

注意到构建出的协议指纹虽是类簇指纹的集合,但在识别过程中,为方便描述,本文假设每个协议只与一个类簇相对应。

### 2.4　基于 WBEV 指纹的协议识别规则

#### 2.4.1　基于指纹的距离度量

由于构建出的协议指纹包含类簇中心及为各维度字节熵分配的权重,因此计算待识别网络流与协议指纹的距离,即计算它们之间的加权欧氏距离。

设某协议类型 $C_i$ 的指纹 $\varphi_i$ 为$\{u_i, \boldsymbol{w}_i\}$,对于待识别的网络流 $f$,其字节熵矢量为 $\boldsymbol{h} = [h_1, h_2, \cdots, h_p]$,则 $f$ 与 $\varphi_i$ 的距离 $d(f, \varphi_i) = \sqrt{\sum_{m=1}^{p} w_{im}(h_m - u_{im})^2}$。

#### 2.4.2　距离阈值

在识别阶段,直观的想法是分别计算待识别网络流与各个协议指纹的距离,用距离最近的协议类型对其进行标志,但是这会导致无法识别训练集中未出现的协议,因此本文设定距离阈值,以判定该样本是否属于该协议。

判断是否属于该协议即判断是否属于该协议对应的类簇,一个类簇内的距离分布由$\{\mu, \sigma^2\}$描述。其中:$\mu$ 为类簇内距离的期望,即样本与中心点的平均距离;$\sigma^2$ 为类簇内距离的方差,描述类簇内距离的离散程度。

本文基于类簇内的距离均值和方差为每个类簇设定一个距离阈值,小于这个阈值的被认为属于这个类簇,同时也属于这个类簇对应的协议,反之则不属于这个类簇对应的协议。

$$\text{if } d(f, \varphi_i) > \mu + T \times \sigma : f \notin C_i$$

式中:$T$ 为正值参数,控制方差对距离阈值的影响程度。

#### 2.4.3　识别规则

对于待识别网络流,分别计算其与每个协议指纹的距离。对于每个指纹,首先判断距离是否超出设定的阈值范围,如果所有距离均超出设定的阈值,则认为是未知协议;否则选择距离最小的指纹,它所标志的协议类型就是该网络流的协议类型。其识别规则如下:

for each $\varphi_i$: if $d(f, \varphi_i) > \mu + T * \sigma$
$f$ as unknown
if $f$ is not as unknown;
$C(f) = \arg\min d(f, \varphi_i)$

## 3　实验与分析

### 3.1　数据集的获取

本文对五种常用二进制协议进行参数调整和测试,分别是 IKE、Skype、OICQ、eDonkey 和 NetBios-NS 协议。数据集由三部分组成,第一部分为实验室局域网环境 FortiGate 设备产生的 IKE 流量,第二部分为公开数据集 Tstat 的 Skype 的 UDP 流量[18],第三部分为广域网流量,包括 OICQ、edonkey、NetBios-NS 和其他二进制协议等。本文将实验数据集分为两组,一组为用于指纹构建的训练集,一组为用于协议识别的测试集。

训练集中只包括要构建的五种二进制协议,测试集中除了以上五种协议外,还包括训练集中未出现的二进制协议,以测试识别方法对未知协议的发现能力。在对上述训练集进行预处理过程中,首先基于五元组进行分流,经分流后产生二进制协议网络流 5 421 条。另外,在指纹构建阶段中,为了度量聚类纯度以确定某些参数,本文使用 Wireshark 软件对流量进行标注。

### 3.2　指纹构建阶段

#### 3.2.1　聚类质量度量指标

1)归一化互信息值 NMI(normarlized mutual information)

令 $Y$ 为样本集中协议类型的变量,$C$ 为样本集中类簇标志的变量,则 $C$ 与 $Y$ 的互信息为 $MI(C, Y) = H(Y) - H(Y|C)$。式中:$H(Y)$为协议类型的熵,描述样本集中协议类型的分布,$H(Y|C)$为协议类型的条件熵。$H(Y|C)$表示已知类簇标志后,样本空间的协议类型分布。计算公式如下:

$$H(Y|C) = -\sum_{c_i} P(c_i) \sum_{y_v} P(y_v|c_i) \log P(y_v|c_i)$$

NMI$(C, Y)$表示归一化的互信息,计算公式如下:

$$\mathrm{NMI}(C,Y)=\frac{H(Y)-H(Y|C)}{\sqrt{H(C)}\sqrt{H(Y)}}$$

归一化的互信息取值为 0 ~ 1 之间。由 NMI 的公式可以看出,NMI 用于度量类簇的纯度,即形成的类簇数目越少,每个类簇中样本的协议类型单一,NMI 值就越大,NMI 取最大值 1 时表示协议类型与类标志形成一一映射。

2)DB(davies bouldin) index 准则

DB index 准则同时使用了类间距离和类内离散度,其准则基本内容如下:

$S_i=\frac{1}{|C_i|}\sum_{x\in C_i}d(x,c)$,表示类簇内平均离散度;

$d(c_i,c_j)$,表示两个类簇中心 $c_i$ 与 $c_j$ 的距离;

$\mathrm{DB}=\frac{1}{k}\sum_{i=1}^{k}R_i$,其中 $k$ 为分类数目,$R_i=\max_{j=1,\cdots,K,j\neq i}\frac{S_i+S_j}{d(c_i,c_j)}$。

DB 指数用于度量类簇的形状。当类内离散度小而类间距离大时 DB 指数越小,说明类簇内部紧凑,类簇之间分散,聚类效果越好。由于 NMI 反映类簇的纯度,即判断是否可以将网络流集合划分成不同协议的类簇,可见它是衡量聚类结果的基础指标,而 DB 指数用于度量聚类的形状,其作用是进一步度量聚类质量的优劣。因此,本文在参数确定时,利用 NMI 指标进行参数调整;而在聚类质量分析时,同时利用两种指标进行聚类质量的度量。

### 3.2.2 参数确定

在指纹构建阶段,需要确定三个参数,即数据包个数 $L$、类簇数目 $K$、字节熵矢量维数 $P$。

首先选择合适的数据包个数 $L$。$L$ 的选择既要确保计算的熵值能够反映字节在网络流中的取值特性,又要受实际网络流数据包数目的制约。一方面,文献[15]使用卡方统计的方法度量数据包比特分组在网络流中取值的随机性,通过实验分析出 $L>10$,就可以反映出比特分组在网络流中的取值特性,但是 $L$ 越大会使得这种特性反映得更准确;另一方面,本文通过对数据集进行观察,发现网络流中数据包数目通常超过 40,因此本文将 $L$ 设定为 40。

然后重点分析字节熵适量维数 $P$ 和类簇数目 $K$ 对聚类结果的影响。图 2 给出了在不同的 $K$ 值及字节熵矢量维度的条件下聚类的 NMI 指标。由于训练集中有五种类型的协议,因此类簇数目从 5 开始逐渐递增。从图中可以看出,虽然样本集中协议类型有五种,但是 $K=5$ 时,互信息值不是很高,经过分析得出 Skype 协议有两种运行模式,即 E2O 和 E2E,因此这会导致两种不同的类簇划分。$K$ 继续增加后,互信息会有小范围内的下降,这是由于虽然 $K$ 值增加,每个类簇的纯度不受影响,但是更细小的类簇划分会导致一个协议类型形成更多类簇映射,而类簇数目越小会影响识别时的效率,因此本文确定 $K=6$。

从图 2 中还可以分析字节熵矢量维度对 NMI 的影响,可以看出维度增加到 10 时,NMI 值有一个显著的增加,随后 NMI 值会随维数增多出现下降的情况。这是由于更多维的字节熵不属于协议的消息头部,在网络流中的取值特性不规律,无法作为识别协议的依据。因此本文选择字节熵矢量维数 $P=10$。

### 3.2.3 聚类质量分析

为了说明基于局部加权 K-means 聚类的指纹构建的优势,本文在相同的条件下运行传统 K-means 聚类算法,构建协议的字节熵矢量指纹,利用 NMI 和 DB 指数两类聚类质量指标对两种方法的聚类质量进行比较。由上文可知,NMI 用于度量类簇的纯度,因此本文利用这两个聚类质量指标从两个方面比较 K-means 与局部加权 K-means 的聚类质量。

表 1 表示取字节熵矢量维度 $P=10$,在不同的 $K$ 值情况下,K-means 与局部加权 K-means 的 NMI 和 DB 指数比较。由表中可以看出,对于 NMI 值两者差别不大,说明 10 维的字节熵可以将不同的协议划分成不同的类簇,但是局部加权 K-means 的 DB 指数显著比 K-means 的 DB 指数小,说明引入加权后的聚类形成的类簇内更紧凑,类簇之间更分散,进一步说明字节熵矢量加权指纹能够使协议样本与相应协议指纹的距离更近,而与其他指纹距离更远。

表 1 类簇质量比较($P=10$)

| 类簇数目 | K-means | | 局部加权 K-means | |
|---|---|---|---|---|
| | DB | NMI | DB | NMI |
| $K=5$ | 0.072 | 0.644 | 0.635 | 0.671 |
| $K=6$ | 0.083 | 0.813 | 0.318 | 0.813 |
| $K=7$ | 0.108 | 0.804 | 0.381 | 0.804 |
| $K=8$ | 0.137 | 0.786 | 0.377 | 0.786 |

## 3.3 协议识别阶段

### 3.3.1 识别性能指标

本文采用如下性能指标对识别器的识别能力进行评估。设测试集的样本数为 $N$,对于协议类型 $M$,$TP$ 表示协议 $M$ 被正确识别为 $M$ 的样本数,$FN$ 表示协议 $M$ 被错误识别为非 $M$ 的样本数,$FP$ 表示非 $M$ 却被错误识别为 $M$ 的样本数,则

$$\text{召回率 } R_{TP}=\frac{TP}{TP+FN},\text{误识别率 } R_{FP}=\frac{FP}{TP+FP}$$

召回率越高,误识别率越低,说明相应的识别效果越好。

### 3.3.2 参数确定

距离阈值中参数 $T$ 值的确定与训练样本集中类簇的距离分布有关,如果距离服从正态分布,则可以利用正态分布理论中的 3sigma 准则,确定 $T$ 值为 3[19]。本文参照文献[19]的方法,利用 Kurt 值检验判断距离是否服从正态分布。

$$\mathrm{Kurt}=\frac{E(X-EX)^4}{(DX)^2}$$

若 Kurt = 3,则统计量 $X$ 服从正态分布。表 2 表示各个类簇中距离的 Kurt 值。

表 2 各个类簇距离的 Kurt 值

| 类簇标志 | Kurt 值 | 类簇标志 | Kurt 值 |
|---|---|---|---|
| cluster 1 | 4.22 | cluster 4 | 4.24 |
| cluster 2 | 4.62 | cluster 5 | 3.12 |
| cluster 3 | 2.89 | cluster 6 | 5.76 |

从表 2 可以看出,第 3 个和第 5 个类簇距离接近服从正态分布,但是其他类簇的 Kurt 值与 3 均有差距,说明本文实验中采用的样本数据没有明显的正态分布特征,因此无法将 $T$ 值直接确定为 3。本文通过衡量识别器的性能确定合适的 $T$ 值。

如上文所述,识别性能高的识别器具有较高的召回率和较低的误识别率。由于 $T$ 值同时影响协议的召回率和误识别率,并且在一定程度上两者是相互矛盾的:$T$ 值增大会增大识别器的召回率,提高识别器性能;但过大的 $T$ 值同时也会增大识别器的误识别率,降低识别器性能。

本文利用召回率与误识别率的差值来衡量识别器的性能,选取使两者差值达到最大的 $T$ 值。$T$ 值确定公式如下:

$$T = \{T \mid \max(R_{TP} - R_{FP})\}$$

当召回率 $R_{TP}=1$,误识别率 $R_{FP}=0$ 时,其差值达到最大,即识别器性能达到最大。图 3 表示不同的 $T$ 值条件下识别器的性能变化。可以看出,$T$ 值设定得过大或过小都会使召回率与误识别率的差值下降,即识别器性能下降。表 3 为选取使召回率与误识别率的差值达到最大的 $T$ 值结果。

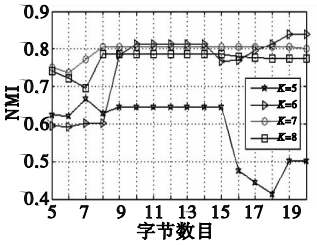


图2　字节维数及类簇数目对NMI的影响

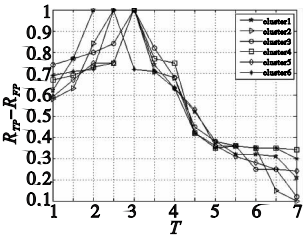


图3　T值对识别性能的影响

表 3　$T$ 值结果

| 类簇标志 | 协议类型 | $T$ | 类簇标志 | 协议类型 | $T$ |
|---|---|---|---|---|---|
| cluster 1 | IKE | 2 | cluster 4 | OICQ | 2.5 |
| cluster 2 | Skype | 2 | cluster 5 | eDonkey | 3 |
| cluster 3 | Skype | 3 | cluster 6 | NBNS | 2 |

### 3.3.3　识别性能分析

由于设定了距离阈值,因此识别器在识别出训练集中已有协议的同时,还能区分出训练集中未出现的协议。表 4 表示对于相同的测试集,分别采用 K-means 和本文方法获得的召回率和误识别率。可以看出,本文方法在误识别率上的性能大大优于 K-means 算法,这是由于本文采用局部加权 K-means 算法为每个协议构建字节熵矢量加权指纹,使得网络流与协议指纹之间的相似度更小,这会降低协议的误识别率,同时能够更加准确地识别出训练集中未出现的协议类型。

表 4　识别性能比较

| 协议类型 | K-means | | 本文方法 | |
|---|---|---|---|---|
| | $R_{TP}$/% | $R_{FP}$/% | $R_{TP}$/% | $R_{FP}$/% |
| IKE | 100 | 1.3 | 100 | 0.2 |
| OICQ | 90.1 | 11.5 | 94.2 | 5.4 |
| Skype | 100 | 8.9 | 100 | 3.1 |
| Net-Bios | 99.89 | 1.1 | 99.89 | 0.0 |
| eDonkey | 99.2 | 4.2 | 99.2 | 1.2 |
| others | 83.2 | 3.2 | 90.5 | 2.7 |

## 4　结束语

基于格式特征的协议识别方法由于其较高的准确率在协议识别中得到了广泛的应用,但是如何构建协议的格式特征一直是亟待解决的难题。本文面向二进制协议,提出字节熵矢量加权指纹的概念。首先利用信息熵度量字节在网络流的取值特性,以字节熵矢量描述网络流的报文格式属性;然后基于局部加权 K-means 聚类的指纹构建算法为协议构建字节熵矢量加权形式的指纹,并据此对协议进行识别。实验表明,该方法对测试的几种二进制协议拥有超过 94% 的准确率和低于 6% 的误识别率,表现出较好的识别性能。

**参考文献:**

[1] 赵咏,姚秋林,张志斌,等. TPCAD:一种文本类多协议特征自动发现方法[J]. 通信学报,2009,30(10):29-35.

[2] ZUEV D, MOORE A W. Traffic classification using statistical approach[C]//Proc of the 6th International Workshop on Passive and Active Network Measurement. Berlin: Springer-Verlag, 2005: 321-324.

[3] DAS R, EACHEMPATI S, MISHRA A K, *et al*. Design and evaluation of a hierarchical on-chip interconnect for next-generation CMPs [C]//Proc of the 15th International Conference on High-performance Computer Architecture. Washington DC: IEEE Computer Society, 2009:175-186.

[4] ESTE A, GRINGOLI F, SALGARELLI L. Support vector machines for TCP traffic classification [J]. Computer Networks, 2009, 53 (14):2476-2490.

[5] CROTTI M, DUSI M, GRINGOLI F, *et al*. Traffic classification through simple statistical fingerprinting[C]//Proc of ACM SIGCOMM Computer Communication Review. New York: ACM Press, 2007:5-16.

[6] BERNAILLE L, TEIXEIRA R, AKODKENOU L, *et al*. Traffic classification on the fly[J]. ACM SIGCOMM Computer Communication Review,2006,36(2):23-26.

[7] YAGI S, WAIZUMI Y, TSUNODA H, *et al*. A reliable network identification method based on transition pattern of payload length [C]//Proc of Global Telecommunications Conference. 2008:1-5.

[8] NGUYEN T, ARMITAGE G. A survey of techniques for internet traffic classification using machine learning[C]//Proc of Communications Survey Tutorials. [S. l. ]:IEEE Press,2008:56-76.

[9] 杨哲,李领治,纪其进,等. 基于最短划分距离的网络流量决策树分类方法[J]. 通信学报,2012,33(8):91-102.

[10] 刘兴彬,杨建华,谢高岗,等. 基于 Apriori 算法的流量识别特征自动提取方法[J]. 通信学报,2008,12(6):51-59.

[11] HAFFNER P, SEN S, SPATSCHECK O, *et al*. ACAS: automated construction of application signatures[C]//Proc of the 1st Annual ACM SIGCOMM Workshop on Mining Network Data. 2005.

[12] MA J, LEVCHENKO K, KREIBICH C, *et al*. Unexpected means of protocol inference[C]//Proc of the 6th ACM SIGCOMM Conference on Internet Measurement. 2006.

[13] KHAKPOUR A R, LIU A X. High-speed flow nature identification [C]//Proc of the 29th IEEE International Conference on Distributed Computing Systems. 2009:510-517.

[14] FINAMORE A, MELLIA M, MEO M. KISS: stochastic packet inspection classifier for UDP traffic[J]. IEEE/ACM Trans on Networking,2010,18(5):1505-1515.

[15] 黎敏,余顺争. 抗噪的未知应用层协议报文格式最佳分段方法[J]. 软件学报,2013,24(3):604-617.

[16] CUI Wei-dong, KANNAN J, WANG H J. Discoverer: automatic protocol reverse engineering from network traces[C]//Proc of the 16th USENIX Security Symposium. Boston, MA: USENIX Association, 2007.

[17] DOMENICONI C, GUNOPULOS D, MA Sheng, *et al*. Locally adaptive metrics for clustering high dimensional data[J]. Data Minning and Knowledge Discovery,2009,14(1):63-97.

[18] Skype testbed traces[EB/OL]. http://tstat.tlc.polito.it/traces-skype.shtml.

[19] DUAN Jiang-jiao, ZENG Jian-ping, ZHANG Dong-zhan. A method for determination on HMM distance threshold[C]//Proc of the 6th International Conference on Fuzzy Systems and Knowledge Discovery. 2009:387-391.