1 模型抽象

$$R_i : C_i \mapsto A_i \tag{1}$$

Ci表示过滤流量包的规则条件。Ai表示满足规则后的行为结果。以防火墙策略为例,字段有协议，源IP：端口，目的IP：端口。具体来讲：C表示d维字段的区间，A表示访问结果允许还是拒绝。

```
R1: deny icmp any 150.160.170.*
R2: deny tcp any:any 150.160.170.*:1024-65535
R3: allow tcp any:any 150.160.170.*:80
R4: deny any any:any any:any
```

图 1: 防火墙

$$\begin{aligned} &C \equiv [S_c1, E_c1][S_c2, E_c2][S_c3, E_c3] \cdots [S_cd, E_cd] \\ &A \in \{accept, deny\} \end{aligned} \tag{2}$$

2 流量聚类

对应的聚类规则：

1 每个规则对应的是一个超矩形，因为每个包的每个字段都是一个区间段

2 相似字段的包属于同一个策略规则

3 规则的匹配顺序基于策略生成架构

采用层次聚类算法对包进行聚类（Agglomerative Clutsering 是一种自底而上的层次聚类方法,它能够根据指定的相似度或距离定义计算出类之间的距离。）

3 聚类之间距离的测量

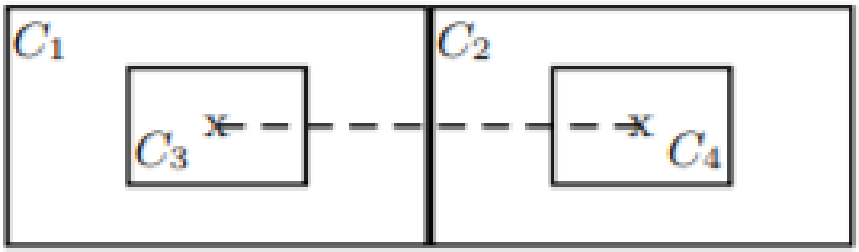

Figure 1: Cluster distance measure.

图 2: 距离

$$D(X,Y) = \sum_{i=1}^{d} d(x_i, y_i)$$

$d(x_i, y_i)$ is calculated as:

City-block.

$$d(x_i, y_i) = |E_{xi} - E_{yi}| + |S_{xi} - S_{yi}|$$

Euclidian.

$$d(x_i, y_i) = \sqrt{(E_{xi} - E_{yi})^2 + (S_{xi} - S_{yi})^2}$$

图 3: 距离公式

4 策略生成的过程

4.1 第一阶段

算法解析：

1 初始化聚类参数

2 将流量包p插入到距离最近的聚类Cm中

3 如果Cm的面积大于 ϕ，则新建另外的一个Cr

4 否则的话，判断Cm的密度是否大于W则进行split

5 最后判断整个C的数量，如果C的数据大于N，则合并距离最小的两个聚类，直到满足聚类数小于N的要求

4.2 第二阶段

算法解析：

1 层次聚类算法中，每一层的聚类数为上一层的一半。对应while算法中的while循环，删除待合并的Ci和Cj，增加合并后的聚类。

2 ρ 表示合并的时候的action比率。

Algorithm 1 Clustering (P: packet sequence, ϕ, λ, ω, N)

$C \leftarrow \epsilon$ {Initialize cluster set}
for all $p \in P$ **do**
  Find cluster C_w where $D(p, C_w)$ is minimal
  $C'_w \leftarrow Insert(p, C_w)$
  if $Size(C'_w) > \phi$ **then**
    $C_\gamma \leftarrow Create(p)$
    $C \leftarrow C_\gamma$
  else
    $C_w \leftarrow C'_w$
    If $Size(C_w) > \omega$ **then** $Split(C_w)$
  end if
  if $|C| > N$ **then**
    Find i, j s.t $D(C_i, C_j)$ is minimal
    $Merge(C_i, C_j)$
  end if
end for

图 4: 第一阶段算法

Algorithm 2 Build Hierarchy (C: cluster set, δ, ρ)

R $\leftarrow \epsilon$ {Initialize rule set}
for all $l < \delta$ **do**
  R $\leftarrow$ C {Add current clusters to rule set}
  $n = N/2$
  while |C| $> n$ **do**
    Find i, j s.t $D(C_i, C_j)$ is minimal
    C $\leftarrow$ C $- C_i$ {Remove clusters}
    C $\leftarrow$ C $- C_j$
    C $\leftarrow Merge(C_i, C_j, \rho)$ {Merge and add to current level}
  end while
  $l \leftarrow l + 1$
end for
Finalize Policy

图 5: 第二阶段算法

5 实验和评估

分析了流量包中的协议，然后从协议分布、具体规则生成进行了实验验证。