# Método Heurístico para Rotular Grupos em Sistema de Detecção de Intrusão baseado em Anomalia

**Hermano Pereira**[1], **Edgard Jamhour**[2]

[1]Companhia de Informática do Paraná - CELEPAR
80.530-010 - Curitiba - PR, Brasil

[2]Pontifícia Universidade Católica do Paraná - PUCPR
80.215-901 - Curitiba - PR, Brasil

hermanopereira@celepar.pr.gov.br, jamhour@ppgia.pucpr.br

***Abstract.*** *The intrusion detection systems are part of security suite necessary for environment protection that contains information available on the Internet. Among these systems it is highlighted the unsupervised learning, as they are able to extract environment models without prior knowledge concerning the occurrence of attacks among the collected information. A technique used to create these models is the data clustering, where the resulting clusters are labeled either as normal or as attack (in anomalous case). This paper proposes a heuristic method for labeling clusters, where the false positive rates achieved during experiments were significantly lower compared to the methods described in related work.*

***Resumo.*** *Os sistemas de detecção de intrusão fazem parte de um ferramental de segurança necessário na proteção de ambientes que disponibilizam informações via Internet. Dentre esses sistemas vêm se destacando os de aprendizagem não-supervisionada, pois são capazes de extrair modelos de comportamento do ambiente sem o conhecimento prévio de ataques dentre as informações coletadas. Uma técnica utilizada para criar esses modelos é a de agrupamento de dados, onde os grupos resultantes são rotulados como normais ou, no caso de anomalias, como ataques. Neste trabalho é proposto um método heurístico para rotular grupos que durante os experimentos resultou em taxas de falsos positivos significativamente menores em relação aos métodos de trabalhos relacionados.*

## 1. Introdução

Os Sistemas de Detecção de Intrusão (*Intrusion Detection Systems*), sigla IDS, são sistemas que atuam junto ao sistema operacional ou em uma rede de computadores buscando identificar atividades maliciosas. Em geral, a estratégia de análise do IDS é baseada em mau uso (*misuse-based*) ou baseada em anomalia (*anomaly-based*). Os IDSs baseados em mau uso fazem a detecção de ataques pela representação destes através de padrões, tais como regras ou assinaturas. Já os IDSs baseados em anomalia precisam conhecer antecipadamente o comportamento do ambiente a ser monitorado, para então detectar ataques através do desvio do comportamento ou na ocorrência de anomalias. A principal vantagem do IDS baseado em mau uso está na possibilidade dos ataques serem especificados por especialistas, porém, esses IDSs necessitam que suas bases de padrões sejam atualizadas constantemente. Já os IDSs baseados em anomalias são dependentes do ambiente

que monitoram, necessitam de mais recursos para efetuar os treinamentos e geram muitos falsos positivos. Todavia apresentam vantagens, tais como detectar ataques novos e obter baixos índices de falsos negativos.

Na comunidade científica é possível encontrar diversos trabalhos de IDSs que procuram reduzir os índices de falsos positivos, e uma técnica baseada em anomalia que vêm obtendo bons resultados é a aprendizagem não-supervisionada (*unsupervised learning*) que utiliza agrupamento de dados (*data clustering*). Através dessa técnica os modelos de detecção de intrusão são criados a partir de treinamento utilizando algoritmos de agrupamento sobre uma base de dados não rotulada (ou seja, sem supervisão). Assim os grupos resultantes recebem um rótulo de normalidade ou, no caso de anomalia, um rótulo de ataque. Porém, quando os grupos menores em número de instâncias ou isolados (*outliers*) que são legítimos mas recebem rótulos de ataque, acabam resultando em aumento no número de falsos positivos durante a detecção de intrusão. Com o intuito de fazer uma melhor avaliação desses grupos e reduzir o índice de falsos positivos, este trabalho apresenta um método heurístico para atribuição de rótulos de reputação aos grupos de acordo com a quantidade e a origem das informações. Diferentemente de normalidade ou de ataque, os rótulos atribuídos podem representar uma reputação que varia de péssima à excelente de acordo com uma escala empírica, a qual pode determinar o quanto uma atividade pode ser considerada uma intrusão.

Para testar o método proposto, o IDS foi implementado e testado sobre um conjunto de requisições HTTP (*Hypertext Transfer Protocol*) que foram coletadas de um servidor *web*, o qual disponibilizava alguns sítios para a *Internet* e recebeu diversos ataques. Para a validação do método proposto, outros dois métodos de atribuição de rótulos de trabalhos relacionados também foram implementados e testados sobre o mesmo conjunto de requisições. Ao final, o método proposto obteve na maioria dos testes os melhores resultados.

Além desta seção, os trabalhos relacionados são descritos na seção 2; a metodologia é apresentada na seção 3; o método heurístico proposto é apresentado na seção 4 e os resultados dos experimentos realizados são apresentados na seção 5. Por fim, na seção 6, são feitas as conclusões e sugestões para trabalhos futuros.

## 2. Trabalhos Relacionados

Os principais trabalhos relacionados com esta pesquisa são os de [Portnoy et al. 2001] e de [Zhong et al. 2007], os quais utilizaram algoritmos de agrupamento para fazer o treinamento no modo não-supervisionado. Em seus seus experimentos os grupos resultantes foram avaliados utilizando um método heurístico, e neste trabalho eles foram implementados e comparados com o método proposto.

Outros trabalhos que também utilizaram agrupamento de dados como técnica baseada em anomalia são: [Eskin et al. 2002], [Guan et al. 2003], [Mahoney et al. 2003] e [Leung and Leckie 2005]; onde os rótulos de ataques foram atribuídos de acordo com os *outliers* encontrados pelos algoritmos de agrupamento. No artigo de [Zhang et al. 2005], os *outliers* foram detectados através de agentes distribuídos e analisados por um IDS central. No trabalho de [Singh et al. 2009], os *outliers* comuns identificados em sistemas diferentes foram considerados ataques. De maneira diferente dos demais, no trabalho de [Petrović 2006] os grupos compactos identificados por índices de validação de agrupa-

mento foram considerados ataques em massa.

Entre os trabalhos que trataram de detecção baseada em anomalia em serviços *web* e HTTP podem ser encontrados os de [Criscione et al. 2009], [Robertson et al. 2010] e [Corona and Giacinto 2010], onde os algoritmos de agrupamento foram utilizados apenas como uma técnica de análise complementar de suas arquiteturas.

## 3. Metodologia

Esta seção apresenta a metodologia que foi aplicada para realizar os experimentos. Um ambiente de testes foi preparado com uma base de 5 milhões de requisições HTTP que foi subdividida em 20 partições (detalhes na Seção 3.1). Para executar os testes, as requisições de uma partição X foram utilizadas durante o treinamento e resultou em modelos de detecção para serem utilizados na inspeção das requisições de uma partição Y, assim ilustra o fluxograma da figura 1. Na fase de treinamento, uma a uma as requisições foram normalizadas para que pudessem ser comparadas uma com as outras (Seção 3.2). Assim um algoritmo de agrupamento de ligação simples (Seção 3.4) foi aplicado e as requisições similares foram agrupadas. Para calcular a distância entre as requisições foi utilizada a medida de similaridade euclidiana que está descrita na Seção 3.3. Após o agrupamento, os grupos resultantes foram avaliados e rotulados de acordo com três métodos heurísticos (Seção 3.5): o método de [Portnoy et al. 2001], o método de [Zhong et al. 2007] e o método proposto neste trabalho. Cada método resultou em modelos que foram utilizados na fase de detecção de intrusão.

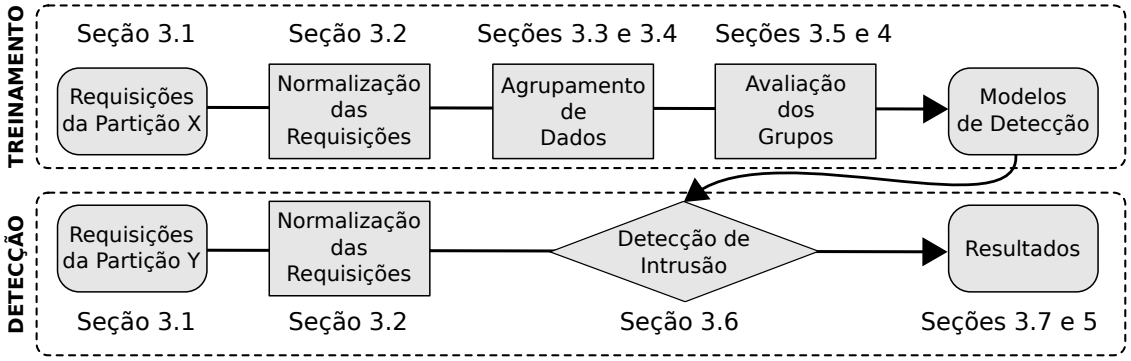


**Figura 1. Fluxograma - sequência aplicada nos testes**

Para a detecção de intrusão em uma partição Y foi utilizado o modelo de detecção da partição X. Por exemplo, o modelo extraído do treinamento sobre a partição P01 foi utilizado para a detecção de ataques na partição P02 seguinte; e o modelo extraído da partição P02 foi utilizado na partição P03 e assim sucessivamente. Exceto no método de detecção descrito por [Zhong et al. 2007], que para a detecção de ataques na partição Y foi utilizado um modelo de detecção da própria partição Y. Então foram testados três métodos de detecção de intrusão (descritos na Seção 3.6), o método de [Portnoy et al. 2001]; o método de [Zhong et al. 2007], e uma variação do método de [Portnoy et al. 2001]. Por fim, cada requisição inspecionada foi detectada como normal ou como ataque e os resultados foram totalizados e apresentados conforme as medidas de comparação (descritas na Seção 3.7): taxa de detecção de ataques, taxa de falsos positivos e índice de medida-F.

### 3.1. Base Rotulada

Nos testes realizados no trabalho de [Portnoy et al. 2001], a base rotulada [KDD 1999] foi subdividida em 10 partes e os experimentos foram realizados sobre quatro dessas partes. No trabalho de [Zhong et al. 2007] foi utilizada a base rotulada [DARPA 1998] (a base [KDD 1999] é uma versão dessa mesma base), onde mais de 100 mil registros foram usados nos testes. Nas duas bases existem aproximadamente 5 milhões de registros de informações coletadas de uma rede que foi utilizada para avaliação de IDSs.

Visto que as bases utilizadas nos trabalhos relacionados datam há mais de 10 anos, nesta pesquisa foi criada uma base própria onde um servidor *web* disponível na *Internet* foi monitorado e seus acessos foram coletados, analisados e rotulados. No total foram coletadas aproximadamente 5 milhões de requisições HTTP entre as 19:00 do dia 16/12/2010 e as 18:00 do dia 28/12/2010 (GMT). Após a análise foram identificados e rotulados mais de mil ataques e aproximadamente 30 mil anomalias. Para aumentar o número de ataques sem gerar um incidente de segurança, foram adicionados a esta base mais de 127 mil ataques gerados por 4 ferramentas de ataques *web* que foram extraídos da base da competição [iCTF 2008]. A tabela 1 apresenta as 20 partições com a quantidade de ataques rotulados, ataques adicionados da base [iCTF 2008], anomalias e as demais requisições rotuladas como normais.

**Tabela 1. Particionamento do Conjunto de Requisições**

| Partição | Ataques | Adicionados | Anomalias | Normais | Total |
|---|---|---|---|---|---|
| P00 | 4 | 0 | 318 | 249678 | 250000 |
| P01 | 29 | 0 | 410 | 249561 | 250000 |
| P02 | 5 | 6570[a] | 1124 | 242301 | 250000 |
| P03 | 51 | 2690[b] | 2647 | 244612 | 250000 |
| P04 | 50 | 0 | 1685 | 248265 | 250000 |
| P05 | 8 | 0 | 868 | 249124 | 250000 |
| P06 | 17 | 0 | 1237 | 248746 | 250000 |
| P07 | 95 | 0 | 1001 | 248904 | 250000 |
| P08 | 21 | 0 | 770 | 249209 | 250000 |
| P09 | 91 | 85570[c] | 891 | 163448 | 250000 |
| P10 | 10 | 27228[c] | 510 | 222252 | 250000 |
| P11 | 49 | 0 | 1365 | 248586 | 250000 |
| P12 | 83 | 0 | 1272 | 248645 | 250000 |
| P13 | 111 | 3888[d] | 2311 | 243690 | 250000 |
| P14 | 14 | 0 | 1301 | 248685 | 250000 |
| P15 | 246 | 0 | 2611 | 247143 | 250000 |
| P16 | 39 | 0 | 2707 | 247254 | 250000 |
| P17 | 129 | 0 | 3669 | 246202 | 250000 |
| P18 | 157 | 0 | 2192 | 247651 | 250000 |
| P19 | 39 | 1398[a] | 847 | 247716 | 250000 |
| **Total** | **1248** | **127344** | **29736** | **4841672** | **5000000** |

Os ataques adicionados da base [iCTF 2008] são das ferramentas:
[a] - [Nessus 2011], [b] - [Acunetix 2011], [c] - [DirBuster 2011] e [d] - [Nikto 2011].

Os ataques [iCTF 2008] que foram adicionados nesta base foram gerados por fer-

ramentas de varredura por vulnerabilidade, as quais tentaram encontrar vulnerabilidades conhecidas em servidores e sítios *web*. A ferramenta [Nessus 2011] realiza ataques para diversos protocolos e aplicações, mas nesta base foram adicionados apenas os ataques que tentaram identificar vulnerabilidades de servidores HTTP e de sítios *web*. Já os ataques adicionados da ferramenta [Nikto 2011] procuravam identificar vulnerabilidades em sítios *web*; e de maneira um pouco mais elaborada a ferramenta [Acunetix 2011] tentava extrair informações de sítios *web* incluindo ataques através do método POST. Apesar dos ataques adicionados da ferramenta [DirBuster 2011] serem massivos, são apenas ataques que tentaram buscar por páginas ou arquivos que deveriam estar ocultos ou protegidos mas que poderiam revelar informações do servidor/sítio atacado. E dentre os ataques que realmente ocorreram ao servidor *web* foram identificados diversos tipos: tentativas de injeção de código SQL, inclusão remota de arquivos, travessia de caminho, busca forçada, abuso de *proxy* e até mesmo ataques de *malwares*.

### 3.2. Normalização dos Dados

As bases utilizadas por [Portnoy et al. 2001] e [Zhong et al. 2007] levaram em conta 41 atributos extraídos de sessões TCP. [Portnoy et al. 2001] normalizou os dados de cada atributo subtraindo o valor da média e dividindo pelo desvio padrão. Já em [Zhong et al. 2007] foram utilizadas 105 dimensões de características das instâncias coletadas da base e foram normalizadas para valores entre zero e um (0,1).

Diferentemente desses trabalhos, a base utilizada nesta pesquisa contém requisições HTTP. Sendo assim foram utilizados apenas 7 campos de cada requisição: *UPath* - caminho do recurso na URL; *UQuery* - consulta ao recurso na URL; *Host* - domínio ou endereço IP do requisitado; *User-agent* - agente navegador do usuário; *Cookie* - dados de sessão do usuário; *Referer* - URL de referência, e *Content* - conteúdo do corpo HTTP. Esses campos foram escolhidos por terem relação com a maioria dos ataques realizados via HTTP.

Como se trata de informações do tipo texto e que estão ofuscadas, o conteúdo de cada campo foi normalizado apenas contabilizando a quantidade de caracteres alfabéticos (a-z e A-Z), caracteres numéricos (0-9) e caracteres não-alfanuméricos. Por exemplo, o texto "Mozilla/5.0" resultaria em 7 para caracteres alfabéticos, 2 para numéricos e 2 para não-alfanuméricos. Por fim, cada requisição teve seus 7 campos normalizados onde cada campo ficou com 3 dimensões (alfabéticos, numéricos e não-alfanuméricos).

### 3.3. Medida de Similaridade

No trabalho de [Portnoy et al. 2001] a medida de similaridade utilizada foi a distância euclidiana, e no trabalho de [Zhong et al. 2007] as medidas eram utilizadas de acordo com o algoritmo de agrupamento, sendo a distância euclidiana ou de Mahalanobis.

Neste trabalho foi utilizada a equação 1 para calcular a distância (d) entre uma requisição *a* e uma requisição *b*. Assim cada campo de uma requisição foi comparado com o campo correspondente da outra requisição usando a medida de distância euclidiana. Assim as 3 dimensões ($m$=3) de quantidade de caracteres descritas na seção 3.2 serviram como base na comparação entre esses campos. Ao final, a distância (d) entre as requisições foi o resultado do somatório da distância euclidiana entre os sete campos ($n$=7) dessas requisições.

$$\mathrm{d}(a,b) = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} (a_{ij} - b_{ij})^2} \quad (1)$$

### 3.4. Agrupamento de Dados

No trabalho de [Zhong et al. 2007] o objetivo era testar a eficiência de diferentes algoritmos de agrupamento na detecção de intrusão, e para isso os autores testaram diversos algoritmos baseados em centróides. Já neste trabalho o objetivo foi fazer uma comparação entre os métodos utilizados para atribuição de rótulos, por isso foi utilizado apenas um algoritmo de agrupamento de ligação simples (*single-linkage*), similar ao utilizado no trabalho de [Portnoy et al. 2001]. O algoritmo consiste nos seguintes passos:

- Iniciar um conjunto de grupos vazios;
- Associar cada requisição com seu grupo mais próximo desde que não ultrapasse o limite de distância *W* pré-definido. O limite de distância *W* é a distância (d) máxima que as requisições podem estar de seu grupo. Se não houver um grupo correspondente, a requisição atual será um novo grupo;
- Repetir o passo anterior para reorganizar as requisições em seus grupos mais próximos.

### 3.5. Avaliação dos Grupos

Após realizar os agrupamentos, os grupos resultantes foram avaliados e rotulados para serem utilizados como modelo de detecção de intrusão. O método heurístico proposto por [Portnoy et al. 2001], que é apresentado como "*labeling clusters*", consiste nos seguintes passos:

- Ordenar os grupos por quantidade de instâncias;
- Selecionar um percentual *N* dos grupos maiores em número de instâncias e rotular como normais;
- Rotular os grupos restantes como ataques.

O método heurístico proposto por [Zhong et al. 2007], que é apresentado como "*self-labeling*", é realizado com os seguintes passos:

- Selecionar o grupo com o maior número de instâncias, rotular como normal e definir como o centróide $\mu_0$;
- Ordenar todos os grupos de acordo com sua distância ao centróide $\mu_0$, e fazer o mesmo procedimento com todas as instâncias;
- Selecionar um percentual $\eta$ de instâncias e rotular como normal;
- Rotular as demais instâncias como ataques.

[Zhong et al. 2007] define $\eta$ como percentual de instâncias normais, mas em outra parte de seu trabalho o mesmo parâmetro também é definido como percentual de instâncias que são ataques.

O método heurístico proposto neste trabalho realiza uma avaliação mais detalhada visando reduzir o índice de falsos positivos, e os passos são os seguintes:

- Calcular um índice de popularidade para cada grupo;

- Encontrar *hosts* que tornaram grupos populares e atribuir uma confiabilidade;
- Calcular um índice de confiabilidade para cada grupo de acordo com os *hosts* que o acessaram;
- Calcular um índice de reputação dada a soma ponderada do índice de popularidade com o índice de confiabilidade;
- Atribuir um rótulo ao grupo, relacionando o índice de reputação com uma escala empírica: excelente, ótima, boa, regular, ruim ou péssima.

Detalhes do método proposto são apresentados na seção 4.

### 3.6. Detecção de Intrusão

Após obter os grupos rotulados, os mesmos foram utilizados na tomada de decisão durante a detecção de intrusão. No trabalho de [Portnoy et al. 2001] os ataques foram detectados da seguinte maneira (método referenciado como D1):

- Extrair um modelo de grupos rotulados de uma partição X da base;
- Inspecionar cada instância da partição Y comparando com os grupos rotulados da partição X (sendo a partição Y subsequente à partição X);
- Após comparar com todos os grupos, cada instância da partição Y receberá o mesmo rótulo do grupo X mais próximo.

No trabalho de [Zhong et al. 2007] o método de detecção utilizado foi o seguinte (método referenciado como D2):

- Realizar a atribuição de rótulos nas instâncias da partição Y da base somente após ordenar cada instância de acordo com sua distância em relação ao centróide $\mu_0$;
- Instâncias na partição Y rotuladas como ataques dado um percentual $\eta$ serão consideradas como tal.

Neste trabalho também foi testada a detecção de intrusão levando em conta o limite de distância *W* utilizado durante o agrupamento (método referenciado como D3):

- Extrair um modelo de grupos rotulados de uma partição X da base;
- Inspecionar cada instância da partição Y comparando com os grupos rotulados da partição X (sendo a partição Y subsequente à partição X);
- Após comparar com todos os grupos, cada instância da partição Y receberá o mesmo rótulo do grupo X mais próximo desde que o limite de distância *W* não seja extrapolado. Caso não haja correspondência com grupo algum, a instância será considerada um ataque ou, no caso do método proposto, receberá uma reputação péssima.

### 3.7. Medidas de Comparação

Após realizar todos os testes de detecção de intrusão os resultados foram contabilizados como verdadeiros positivos (*VP*), falsos positivos (*FP*), falsos negativos (*FN*) e verdadeiros negativos (*VN*). Para simplificar os cálculos, as anomalias rotuladas que foram detectadas ou não, tiveram seus resultados ignorados. Os resultados foram calculados utilizando as seguintes fórmulas: taxa de detecção de ataques (TDA) ou revisão (inglês: *recall*), taxa de falsos positivos (TFP), precisão (inglês: *precision*) e a medida-F (inglês: *F-measure*). Detalhes sobre essas medidas podem ser encontradas em [Fawcett 2003].

## 4. Método Heurístico Proposto

Este método foi criado a partir de observações experimentais durante a verificação de ataques na base de requisições, já foi implementado e descrito anteriormente na dissertação de [Pereira 2011]. Durante essa análise foi possível observar que as informações das requisições que eram mais comuns (ou **populares**) tinham uma chance menor de serem consideradas ataques, e que aquelas que não eram comuns podiam ser consideradas **confiáveis** se o *host* de origem também fosse confiável. Assim as informações poderiam ser agrupadas, e através de cálculos empíricos determinar um índice de popularidade e um índice de confiabilidade para cada grupo. Ao final, esses índices foram combinados para calcular um índice de reputação, e o processo todo resultou em um método heurístico para atribuir rótulos de **reputação** para os grupos. Este método foi subdividido em quatro etapas que são apresentadas nesta seção.

### 4.1. Primeira Etapa: Índice de Popularidade dos Grupos

O objetivo de calcular o índice de popularidade *p* é determinar o quanto um grupo está equilibrado em relação a diversidade de *hosts* que o acessaram. Isso significa que quanto melhor for a distribuição dos acessos realizados pelos *hosts* melhor será o índice de popularidade. A linha de corte *l* é um parâmetro definido antes de calcular *p* com o intuito de evitar que *hosts* que realizaram muitos acessos colaborem com o aumento de *p* de um grupo. O valor de *p* deve ficar entre 0 (zero) e 1 (um). Assim o quanto mais o valor de *l* for próximo de zero, mais *hosts* serão necessários para tornar um grupo popular.

O algoritmo 1 apresenta uma proposta simples para que a linha de corte (*l*) possa funcionar conforme o que foi proposto. O valor de *i* identifica o *host* e o valor de *m* identifica a quantidade total de acessos efetuados ao grupo, portanto, o valor de $m_i$ representa o total de acessos do *host i* dentro do grupo. Já a variável $q_i$ representa o percentual de acessos efetuados pelo *host i* dentro do grupo. A variável *z* é booleana e serve para manter o laço de repetição enquanto houver algum valor de $q_i$ zerado, isso significa que o índice de popularidade só pode ser calculado se na última iteração nenhum *host* ultrapassar o percentual estabelecido na linha de corte. Outra estrutura de repetição faz com que todos os acessos realizados pelos *hosts* de 1 até *n* sejam analisados. Ao final do algoritmo, o índice de popularidade (*p*) é calculado de acordo com os valores obtidos de *q*, e está detalhado na equação 2.

$$p = \frac{1}{n}\sum_{i=1}^{n} a \text{ onde } \begin{cases} a = 0, & \text{se } q_i = 0 \\ a = 1, & \text{se } q_i > 0 \end{cases} \tag{2}$$

### 4.2. Segunda Etapa: Confiabilidade dos *Hosts*

Nesta etapa os *hosts* que popularizaram os grupos deverão receber um grau de confiança, o qual será utilizado como base para calcular a confiabilidade dos grupos. Para isso utiliza-se a equação 3, onde para cada *host* (*i*) é feito o somatório ($v_i$) de todos os índices de popularidade ($p_j$) dos grupos que esse *host* acessou. A tupla (*i*,*j*) apresentada na equação 3 representa uma instância de acesso do *host i* ao grupo *j*, e a variável *m* é a quantidade total de grupos no agrupamento.

**Algoritmo 1** Cálculo do índice de popularidade

```
q_i = ∅
z = true
while z do
  z = false
  for i = 1; i ≤ n; i = i + 1 do
    if q_i > 0 or q_i = ∅ then
      q_i = m_i / m
    end if
    if q_i > l then
      q_i = 0
      m = m - m_i
      z = true
    end if
  end for
end while
Calcular p dado q (equação 2)
```

$$v_i = \sum_{j=1}^{m} a \text{ onde } \begin{cases} a = 0, & \text{se } \nexists \text{ (i,j)} \\ a = p_j, & \text{se } \exists \text{ (i,j) e } q_{ij} > 0 \end{cases} \tag{3}$$

Então a confiabilidade dos *hosts* ($w_i$) deverá ser maximizada calculando as três equações: 4, 5 e 6. Nestas equações a variável $u$ representa o somatório de todos os índices de popularidade ($p$). Na equação 5 a variável $g$ representa o índice do *host* mais confiável, e na equação 6 é calculada para todos os *hosts* a sua confiabilidade de acordo com os valores obtidos de $g$ e de $u$.

$$u = \sum_{j=1}^{m} p_j \tag{4}$$

$$g = max\left(\frac{v_i}{u}\right) \tag{5}$$

$$w_i = \frac{v_i}{g \cdot u} \tag{6}$$

### 4.3. Terceira Etapa: Índice de Confiabilidade dos Grupos

Após a identificação da confiabilidade de cada *host* é possível calcular o índice de confiabilidade ($c$) de cada grupo. Para calcular este índice, basta somar a confiabilidade $w_i$ dos *hosts* e dividir pela quantidade $h$ de *hosts* que participaram desse grupo. Esse cálculo pode ser visualizado na equação 7.

$$c = \frac{1}{h} \sum_{i=1}^{h} w_i \tag{7}$$

### 4.4. Quarta Etapa: Índice de Reputação dos Grupos

Uma vez que se obtém os índices de popularidade e de confiabilidade dos grupos, calcula-se o índice de reputação ($r$) estipulando os fatores de ponderação (equação 8), desde que a soma de $y_p$ e $y_c$ seja igual a 4.

$$r = y_p \cdot p + y_c \cdot c \tag{8}$$

A ideia em aplicar uma escala de 0 (zero) a 4 (quatro) para ponderação está em dar ênfase a um índice mais do que ao outro. Como é possível observar nas etapas anteriores, é mais custoso ao índice de confiabilidade atingir valores próximos de 1 (um). Consequentemente o resultado do índice de reputação ($r$) também ficará na escala de zero (0) a quatro (4) e uma sugestão é utilizar esta escala empírica para definir uma reputação: péssima (0,0 a 0,1), ruim (0,1 a 1,0), regular (1,0 a 1,5), boa (1,5 a 2,0), ótima (2,0 a 3,0) e excelente (3,0 a 4,0). Ao final, os grupos receberão seus rótulos e poderão ser utilizados como modelo na detecção de intrusão.

## 5. Experimentos

Nesta seção são apresentados os resultados dos experimentos onde o treinamento foi realizado com agrupamentos variando o valor de $W$ para 80, 100 e 120. O valor de $W$ para 120 foi utilizado pois resultou em melhores índices de validação de agrupamento durante os testes preliminares. Já os valores de $W$ para 80 e 100 foram selecionados pois resultaram em número de grupos aproximados aos utilizados no trabalho de [Zhong et al. 2007]: 100 e 200 grupos. Após executar os agrupamentos os valores de $W$ para 80, 100 e 120 geraram em média, respectivamente, 222,7, 125,5 e 77,2 grupos.

Para a execução dos métodos heurísticos de atribuição de rótulos, os valores utilizados para $N$ foram 0,02, 0,07 e 0,15, que são os mesmos utilizados em [Portnoy et al. 2001]. No trabalho de [Zhong et al. 2007] o valor utilizado para $\eta$ foi de 0,115 para representar o percentual de ataques na base em que foi testado, mas nestes experimentos além desse percentual também foram testados os valores de 0,0575 e de 0,23, que são respectivamente a metade e o dobro. No método proposto, os valores selecionados para $l$ foram 0,05, 0,1 e 0,2 que respectivamente representam, por exemplo, o mínimo de 20, 10 e 5 *hosts* necessários para tornar um grupo popular. Além disso, no método proposto, foram ponderados os valores de $y_p$ = 1 e $y_c$ = 3, considerando a confiabilidade do grupo mais importante do que a popularidade.

Após aplicar o método heurístico proposto, os grupos rotulados com uma reputação ruim (entre 0,1 e 1,0) ou péssima (menor que 0,1) foram considerados ataques durante a detecção de intrusão. Também foram realizados testes com cada um dos métodos de detecção de intrusão: [Portnoy et al. 2001] referenciado como D1; [Zhong et al. 2007] referenciado como D2, e o método que leva em conta o valor de $W$, referenciado como D3.

No total foram realizados 1539 testes utilizando 3 agrupamentos ($W$=80, $W$=100 e $W$=120); com 3 métodos de avaliações de grupos ([Portnoy et al. 2001], [Zhong et al. 2007] e o Proposto); com 3 variações de parâmetros ($N$=0,02, $N$=0,07 e $N$=0,15; $\eta$=0,0575, $\eta$=0,115 e $\eta$=0,23; $l$=0,05, $l$=0,1 e $l$=0,2); juntamente com 3 métodos de detecção (D1, D2 e D3) aplicados nas 19 partições da base de requisições.

Cada linha da tabela 2 apresenta os melhores resultados obtidos durantes os experimentos de um método heurístico específico. As partições nas quais os resultados foram relevantes para discussão são apresentados nesta tabela, os demais resultados das outras partições foram suprimidos.

**Tabela 2. Melhores resultados obtidos**

| P | Método | VP | FP | FN | TDA | TFP | F |
|---|---|---|---|---|---|---|---|
| P02 | Proposto($W$=100;$l$=0,1;D3) | 3579 | 1212 | 2996 | 0,5443 | 0,0050 | 0,6297 |
| | Portnoy($W$=100;$N$=0,07;D3) | 6565 | 39064 | 10 | 0,9985 | 0,1612 | 0,2515 |
| | Zhong($W$=80;$\eta$=0,0575;D2) | 2226 | 13709 | 4349 | 0,3386 | 0,0566 | 0,1978 |
| P03 | Proposto($W$=100;$l$=0,2;D1) | 2648 | 129 | 93 | 0,9661 | 0,0005 | 0,9598 |
| | Zhong($W$=120;$\eta$=0,0575;D1) | 2551 | 7690 | 190 | 0,9307 | 0,0314 | 0,3930 |
| | Portnoy($W$=120;$N$=0,15;D1) | 2557 | 9418 | 184 | 0,9329 | 0,0385 | 0,3475 |
| P04 | Proposto($W$=120;$l$=0,2;D3) | 3 | 27 | 47 | 0,0600 | 0,0001 | 0,0750 |
| * | *Proposto($W$=100;$l$=0,1;D2)* | *28* | *2699* | *22* | *0,5600* | *0,0109* | *0,0202* |
| | Portnoy($W$=80;$N$=0,15;D3) | 41 | 16650 | 9 | 0,8200 | 0,0671 | 0,0050 |
| | Zhong($W$=120;$\eta$=0,23;D2) | 46 | 25174 | 4 | 0,9200 | 0,1014 | 0,0036 |
| P09 | Portnoy($W$=120;$N$=0,07;D1) | 85067 | 20083 | 594 | 0,9931 | 0,1229 | 0,8916 |
| * | *Proposto($W$=40;$l$=0,05;D3)* | *43550* | *27422* | *42111* | *0,5083* | *0,1677* | *0,3584* |
| | Proposto($W$=100;$l$=0,05;D3) | 92 | 1891 | 85569 | 0,0011 | 0,0116 | 0,0021 |
| | Zhong($W$=80;$\eta$=0,0575;D2) | 35 | 7499 | 85626 | 0,0004 | 0,0459 | 0,0007 |
| * | *Proposto($W$=40;$l$=0,2;D3)* | *26582* | *11452* | *656* | *0,9759* | *0,0515* | *0,8144* |
| P10 | Portnoy($W$=100;$N$=0,02;D3) | 27238 | 100182 | 0 | 1,0000 | 0,4508 | 0,3523 |
| | Proposto($W$=80;$l$=0,2;D3) | 13 | 1121 | 27225 | 0,0005 | 0,0050 | 0,0010 |
| | Zhong($W$=120;$\eta$=0,23;D3) | 16 | 57059 | 27222 | 0,0006 | 0,2567 | 0,0004 |
| P13 | Proposto($W$=120;$l$=0,05;D1) | 3768 | 2951 | 231 | 0,9422 | 0,0121 | 0,7031 |
| | Portnoy($W$=120;$N$=0,15;D2) | 3775 | 14729 | 224 | 0,9440 | 0,0604 | 0,3355 |
| | Zhong($W$=120;$\eta$=0,115;D2) | 3770 | 16219 | 229 | 0,9427 | 0,0666 | 0,3143 |
| P14 | Proposto($W$=120;$l$=0,2;D1) | 7 | 52 | 7 | 0,5000 | 0,0002 | 0,1917 |
| * | *Proposto($W$=100;$l$=0,1;D3)* | *14* | *1702* | *0* | *1,0000* | *0,0068* | *0,0163* |
| | Zhong($W$=100;$\eta$=0,0575;D1) | 9 | 4127 | 5 | 0,6429 | 0,0166 | 0,0044 |
| | Portnoy($W$=120;$N$=0,15;D1) | 12 | 11495 | 2 | 0,8571 | 0,0462 | 0,0020 |
| P19 | Zhong($W$=120;$\eta$=0,0575;D1) | 1437 | 11635 | 0 | 1,0000 | 0,0470 | 0,1980 |
| * | *Proposto($W$=40;$l$=0,2;D3)* | *739* | *10516* | *698* | *0,5142* | *0,0424* | *0,1164* |
| | Proposto($W$=80;$l$=0,2;D3) | 90 | 910 | 1347 | 0,0626 | 0,0037 | 0,0738 |
| | Portnoy($W$=100;$N$=0,07;D1) | 1437 | 43794 | 0 | 1,0000 | 0,1768 | 0,0616 |

**P**: partição da base que teve as requisições inspecionadas;
**Método**: método heurístico aplicado, agrupamento, parâmetros e método de detecção;
**VP**: verdadeiros positivos, a quantidade de ataques detectados;
**FP**: falsos positivos, a quantidade de alertas falsos de ataques;
**FN**: falsos negativos, a quantidade de ataques que não foram detectados;
**TDA**: apresenta a taxa de detecção de ataques;
**TFP**: apresenta a taxa de falsos positivos;
**F**: apresenta o índice de medida-F que foi alcançado.

A medida-F foi utilizada para identificar o método que obteve melhor resultado em cada particionamento. Essa medida faz uma média harmônica entre a precisão e a re-

visão, considerando mais importante os testes que obtiveram boas taxas de detecção mas também que geraram poucos falsos positivos. Assim com os experimentos realizados sobre as 19 partições, o método heurístico proposto obteve melhores resultados de medida-F em 16 partições e na maioria dos testes as taxas de falsos positivos não ultrapassaram de 1% conforme pode ser visualizado no gráfico da figura 2.

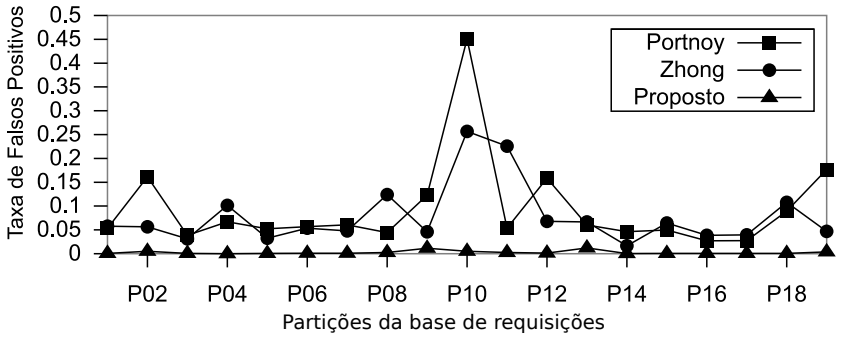


**Figura 2. Gráfico - Taxas de Falsos Positivos**

Discussão sobre os resultados obtidos com o método de [Zhong et al. 2007]:

- Ao investigar o motivo pelo qual [Zhong et al. 2007] não obteve bons resultados, foi possível observar que dentro da base haviam concentrações de grupos com quantidade considerável de requisições normais, mas que estavam distante do centróide principal e acabaram resultando no aumento de alertas de falsos positivos.
- Uma exceção foi o treinamento realizado na partição P18, onde o modelo obtido com esse método resultou em melhor detecção na partição P19, a qual continha diversos ataques da ferramenta Nessus.

Discussão sobre os resultados obtidos com o método de [Portnoy et al. 2001]:

- Na maioria das partições o método heurístico de [Portnoy et al. 2001] resultou nas melhores taxas de detecção de ataques. Porém esse bom resultado foi penalizado pela grande quantidade de falsos positivos.
- O melhor resultado de [Portnoy et al. 2001] foi na partição P09 com treinamento realizado na partição P08, onde o número de falsos positivos não foi significante em comparação ao número de ataques da ferramenta DirBuster que foram detectados.

Discussão sobre os resultados obtidos com o método proposto:

- Na maioria das partições o método proposto obteve melhores resultados nas taxas de falsos positivos (conforme o gráfico na figura 2), porém suas taxas de detecção de ataques não foram tão significativas quanto as de [Portnoy et al. 2001]. Isso ocorre devido ao cálculo de medida-F, pois se o número de falsos positivos for tolerado é possível aproximar das melhores taxas de detecção. Para comprovar, nas partições P04 e P14 as linhas cujas as primeiras colunas estão marcadas com um asterisco (*) estão os testes que obtiveram melhores taxas de detecção.
- Nas partições P09 e P10 este método obteve péssimos resultados, isso se dá aos ataques realizados pela ferramenta DirBuster que durante o agrupamento gerou

grupos densos e próximos aos grupos considerados como normais (uma situação parecida ocorreu com a partição P19). Para melhorar a detecção nessa situação um teste adicional foi realizado, onde o limite de distância $W$=40 foi utilizado com o intuito de criar mais grupos. O resultado pode ser visualizado nas linhas destacadas por um asterisco (*) nas partições P09, P10 e P19.

## 6. Conclusão

Este trabalho apresentou uma proposta de método heurístico para atribuição de rótulos em grupos que resultou em modelos de detecção de intrusão para um IDS baseado em anomalia. Dois trabalhos relacionados foram implementados e seus resultados foram comparados com os resultados obtidos com o método proposto. Das 19 partes testadas de uma base de requisições *web*, o método proposto obteve melhores resultados em 16 delas. Na maioria dos testes as taxas de detecção de ataques não foram superiores aos dos outros métodos, por outro lado, em diversos testes a taxa de falsos positivos não ultrapassou de 1%. Isso demonstra que este método é promissor, pois o baixo número de alertas falsos permite que um analista de segurança inspecione os resultados de maneira eficiente, mesmo sabendo que se trata de um IDS que foi treinado sem supervisão.

Para trabalhos futuros, tanto as implementações como a base de requisições foram disponibilizadas para o público em [Celepar-Dataset 2011], e poderão ser utilizadas para fins de pesquisa. Como complemento deste trabalho também poderão ser feitos testes do IDS no modo de aprendizagem contínua (*active learning*). Além disso, o método heurístico proposto também precisará de mais estudos, pois o índice de confiabilidade dos *hosts* é calculado de acordo com o *host* que é considerado o mais confiável. Esse cálculo foi suficiente para obter bons resultados neste trabalho, mas dependendo do ambiente do treinamento poderá não ser o ideal para se obter os melhores índices.

## Referências

Acunetix (2011). Acunetix Web Security Scanner (`http://www.acunetix.com/` - acesso em 10/07/2011).

Celepar-Dataset (2011). Celepar - Dataset with web attacks for intrusion detection research (`http://ids.celepar.pr.gov.br/dataset`).

Corona, I. and Giacinto, G. (2010). Detection of Server-side Web Attacks. *Journal of Machine Learning Research - Proceedings Track*, 11:160–166.

Criscione, C., Salvaneschi, G., Maggi, F., and Zanero, S. (2009). Integrated Detection of Attacks Against Browsers, Web Applications and Databases. In *Proceedings of the 2009 European Conference on Computer Network Defense*, EC2ND '09, pages 37–45, Washington, DC, USA. IEEE Computer Society.

DARPA (1998). 1998 DARPA Intrusion Detection Evaluation Data Set (`http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html` - acesso em 10/07/2011).

DirBuster (2011). OWASP DirBuster Project (`https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project` - acesso em 10/07/2011).

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. In *Applications of Data Mining in Computer Security*.

Fawcett, T. (2003). ROC Graphs: Notes and Practical Considerations for Researchers. Tech Report HPL-2003-4, HP Laboratories. Available: http://www.purl.org/NET/tfawcett/papers/ROC101.pdf.

Guan, Y., Ghorbani, A. A., and Belacel, N. (2003). Y-means: A Clustering Method for Intrusion Detection. In *Proceedings of Canadian Conference on Electrical and Computer Engineering*, pages 1083–1086.

iCTF (2008). The 2008 iCTF Data (http://ictf.cs.ucsb.edu/data/ictf2008/ - acesso em 10/07/2011).

KDD (1999). KDD Cup 1999 databases (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html - acesso em 10/07/2011).

Leung, K. and Leckie, C. (2005). Unsupervised Anomaly Detection in Network Intrusion Detection using Clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38*, ACSC '05, pages 333–342, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Mahoney, M. V., Chan, P. K., and Arshad, M. H. (2003). A Machine Learning Approach to Anomaly Detection. Technical Report CS-2003-06, Department of Computer Science, Florida Institute of Technology, Melbourne, FL.

Nessus (2011). Nessus Vulnerability Scanner (http://www.nessus.org/ - acesso em 10/07/2011).

Nikto (2011). Nikto Open Source web server scanner (http://www.cirt.net/nikto2 - acesso em 10/07/2011).

Pereira, H. (2011). Sistema de Detecção de Intrusão para Serviços Web baseado em Anomalias. Master's thesis, PUCPR, Curitiba - PR.

Petrović, S. (2006). A Comparison Between the Silhouette Index and the Davies-Bouldin Index in Labelling IDS Clusters. *Proceedings of the 11th Nordic Workshop of Secure IT*, pages 53–64.

Portnoy, L., Eskin, E., and Stolfo, S. (2001). Intrusion Detection with Unlabeled Data Using clustering. In *Proceedings of ACM Workshop on Data Mining Applied to Security*.

Robertson, W., Maggi, F., Kruegel, C., and Vigna, G. (2010). Effective Anomaly Detection with Scarce Training Data. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA.

Singh, G., Masseglia, F., Fiot, C., Marascu, A., and Poncelet, P. (2009). Mining Common Outliers for Intrusion Detection. In *EGC (best of volume)*, pages 217–234.

Zhang, Y.-F., Xiong, Z.-Y., and Wang, X.-Q. (2005). Distributed Intrusion Detection based on Clustering. *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, 4:2379–2383 Vol. 4.

Zhong, S., Khoshgoftaar, T., and Seliya, N. (2007). Clustering-based Network Intrusion Detection. *International Journal of Reliability, Quality and Safety Engineering*, 14(2):169–187.