Fatadist: Uma Ferramenta para Classificação de Ataques baseada em Discriminantes Estatísticos

Victor Pasknel de Alencar Ribeiro[1], Raimir Holanda Filho[2]

Universidade de Fortaleza, Ceará, Brasil.
pasknel@hotmail.com[1], raimir@unifor.br[2]

Abstract. This work proposes an attack traffic identification based on statistical discriminators. Attack identification is of great importance to many areas such as: intrusion detection, security, quality of service and the development of new hardware tools related to security. For the identification of each kind of attack, statistical discriminators were used based on their power of classification. The results obtained through this technique are presented in this work.

Keywords: Traffic Analysis, Statistical Discriminators, Attack Detection.

1 Introdução

Os ataques são uma grave fonte de riscos para as empresas, portanto a detecção e identificação são atividades que devem ocorrer de forma rápida, no intuito de tratá-los e impedi-los.

Sabendo-se que a identificação de ataques, quando os mesmos ocorrem, é uma tarefa difícil, este trabalho apresenta uma proposta para identificar e classificar ataques baseado na análise dos fluxos através da utilização de discriminantes estatísticos e árvore de decisão.

As redes são susceptíveis a vários tipos de ataques (e.g., UDP *flood*, ataque de dicionário, *Port Scan*) e cada um com suas próprias características [1], [2], [3]. Os ataques considerados neste trabalho podem ser divididos em três categorias: Negação de serviço (DoS - *Denial of Service)*, ataques de enumeração (*Enumeration*) e quebra de senha.

Para identificar e separar os ataques, foi utilizado o método de discriminantes estatísticos onde elegemos uma ou mais variáveis que possam identificar e isolar características únicas de um ataque em detrimento dos outros. Estas variáveis são denominadas de discriminantes [4].

Neste trabalho selecionamos as variáveis discriminantes através da análise de gráficos de *boxplot*. A eficiência de cada discriminante escolhido é comprovada através do método de árvore de decisão utilizado para a classificação de ataques.

A seção 2 deste artigo apresenta alguns trabalhos relacionados com detecção de ataques, assim como os ataques utilizados neste trabalho. Na seção 3 é demonstrada a geração dos traços, bem como a topologia da rede utilizada. A metodologia utilizada para a classificação dos ataques é apresentada na seção 4. Na seção 5 são exibidos os

resultados finais e finalmente na seção 6 são demonstradas as principais conclusões e futuros projetos.

2 Classificação de Ataques

Análise de tráfego é um dos temas que tem sido foco de atenção por diversos pesquisadores nos últimos anos. Os trabalhos de identificação de ataques publicados demonstram diferentes métodos para esta tarefa, dentre eles estão a detecção baseada em anomalias, comportamento, e propriedades estatísticas.

Em [5] é apresentado um método de detecção de ataques baseado em anomalias de tráfego. Esta abordagem baseia-se na premissa que ataques provocam desvios no comportamento normal da rede. A partir de tais anomalias, este método é capaz de identificar tanto ataques conhecidos como novos ataques.

Métodos de identificação baseados em comportamento utilizam algoritmos de aprendizado com o auxilio de informações sobre como os ataques se comportam [6].

Uma metodologia para a classificação de classes de tráfego de aplicações da internet utilizando discriminantes estatísticos e análise de agrupamento (*cluster)* é demonstrada em [7].

Estes trabalhos anteriores têm se dedicado à detecção de tráfego de ataque, entretanto este apresenta soluções para a classificação de diferentes tipos de ataques. Neste trabalho é proposta a classificação de tráfegos de ataques a partir da análise das propriedades estatísticas dos fluxos.

Ataques de *flood* têm como propósito enviar uma grande quantidade de dados para um determinando alvo, com o objetivo de sobrecarregar a vítima, impedindo assim que requisições realizadas por clientes legítimos sejam respondidas. Este tipo de ataque pode ser efetivado utilizando diferentes protocolos, dentre eles o HTTP, SMTP e FTP [1].

Um ataque de varredura de portas (*Port Scan*) refere-se à técnica de constatar seqüencialmente, via uma requisição de conexão TCP ou de simples datagramas UDP, números de portas em uma máquina que possam ser aproveitadas em futuros ataques através de possíveis brechas de segurança sob as mesmas [8].

Neste trabalho são consideradas duas categorias de ataques de quebra de senha. O primeiro (*Dictionary attack*), tem como objetivo a quebra de senhas de um mecanismo de autenticação utilizando um conjunto predeterminado de palavras. O segundo (*Brute Force*), realiza o ataque mesclando letras e dígitos de forma aleatória até conseguir a adivinhação da senha [9].

3 Geração dos Traços

Para a realização de ataques e a coleta de pacotes, primeiramente é necessário criar um cenário de rede onde é possível efetivar estas importantes tarefas.

Todo o processo de geração, análise e classificação dos ataques, foi realizado de maneira controlada, em laboratório. A Figura 1 demonstra o ambiente utilizado para a geração do tráfego de ataque.

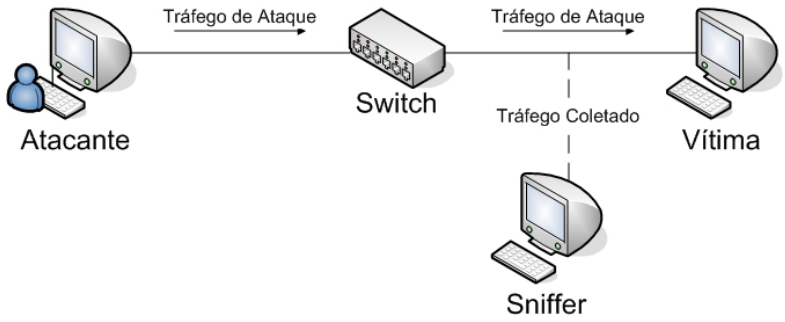

Fig. 1. Topologia da Rede.

Neste cenário foram utilizados três computadores: o atacante, a vítima, e o *sniffer*. O atacante possui o papel de enviar tráfegos de ataques contra os serviços, como FTP, SMTP e HTTP, de uma determinada vítima. Para a realização de ataques DoS e *Port Scan* foi utilizado o programa DoS v5.5, enquanto os ataques *Dictionary* e *Brute Force* foram gerados pelos programas *Hydra* e *Brutus* respectivamente.

Para a coleta de pacotes é necessária a utilização de um analisador de protocolos (*Sniffer*), neste trabalho foi utilizado o software Wireshark para a realização desta tarefa.

Durante a geração foram realizadas sessões de um único ataque, para uma melhor análise de cada tipo, e por fim duas sessões contendo todos os ataques combinados, os quais serão usados para validar o método de classificação descrito na seção 4.3.

Neste trabalho é utilizado o princípio de fluxos de dados, o qual pode ser definido como um número de pacotes trafegando entre dois dispositivos, através de um protocolo em comum e um par de portas específicas [4].

Os pacotes coletados durante um tráfego de dados foram filtrados por um programa de autoria própria, escrito em *Perl*. Com o objetivo da criação de fluxos de dados a partir das informações fornecidas apenas pelos cabeçalhos dos pacotes obtidos.

Os traços são formados pelos fluxos do protocolo TCP e armazenam informações tais como: como tamanho do pacote e flags TCP. Estes dados foram obtidos através dos cabeçalhos dos pacotes coletados.

Uma atividade de classificação manual dos fluxos é realizada durante esta etapa do trabalho. Esta tarefa consiste em realizar uma identificação manual para cada tipo de ataque analisado. Uma variável identificadora, a qual representa o fluxo de ataque em questão, é adicionada em cada arquivo de traço gerado.

Esta atividade será necessária para verificar a taxa de acerto do método de classificação, assim como os valores de falso negativo e falso positivo, durante a etapa de validação. Uma árvore de decisão será criada a partir da combinação dos diversos fluxos de ataque coletados. Após a geração da árvore modelo, as variáveis identificadoras serão utilizadas para reconhecer cada fluxo de ataque, em um determinado nó folha da árvore.

Tendo em vista que o TCP é um protocolo baseado em estado, ou seja, possui um controle de estabelecimento e término de comunicação, não temos problema em determinar a formação dos fluxos que formam os traces baseados neste protocolo, ao contrário de traços do protocolo UDP.

Para a formação dos traços de ataques baseados no protocolo TCP consideramos o total de 2000 fluxos para cada um dos traços de ataque.

4 Metodologia

A metodologia proposta consiste na execução de três fases: padronização dos traços, seleção dos discriminantes e treinamento.

4.1 Padronização dos Traces

Todos os traços de ataque utilizados são formados por linhas onde cada uma representa um fluxo de ataque. Cada fluxo contém informações que serão exploradas pelos discriminantes candidatos para que se possa eleger um ou mais para a classificação de cada *trace* de ataque. A padronização ocorre durante ao uso de gráficos de *boxplot* e com a eliminação de *outliers.*

Gráficos boxplot são utilizados neste trabalho com o objetivo de analisar e comparar conjuntos de dados (variáveis estatísticas) de diferentes classes de ataque. A partir destes diagramas, poderemos visualizar a distribuição dos dados e a presença de *outliers*.

O gráfico boxplot caracteriza a mediana, primeiro e terceiro quartis na distribuição acumulada dos valores que estão sendo analisados pela discriminante candidata em 50%, 25% e 75% respectivamente.

Os *outliers* são valores que se mostram a mais do que um múltiplo de 1.5 e 3.0 de N dos valores interquartílicos, acima ou abaixo dos percentuais 75% e 25% respectivamente. Onde N é o resultado da diferença entre o primeiro e terceiro quartil.

Os valores das informações analisadas que estiverem acima do múltiplo de 3.0 que estejam abaixo do primeiro e acima do terceiro quartil serão considerados desvios do comportamento do ataque e os fluxos que contém tais informações serão descartados. Estes são denominados de *outliers* extremos.

A retirada dos *outliers* extremos garante a padronização do comportamento de todos os fluxos dos traces de ataque para que os mesmos possam ser analisados e classificados por cada discriminante candidato.

4.2 Seleção dos Discriminantes

Consideramos a escolha dos discriminantes que serão utilizadas na análise dos ataques como o principal passo para a classificação dos mesmos. Foram eleitos discriminantes candidatos para analisar o valor de discriminancia das variáveis contidas em cada um dos fluxos dos trace de ataque. Dessa forma os gráficos de *boxplot* podem ser construídos e analisados. As variáveis candidatas são apresentadas na Tabela 1.

Tabela 1. Discriminantes Candidatos.

Variáveis discriminantes candidatos
Total de pacotes de cliente/servidor
Total de pacotes de servidor/cliente
Total de pacotes c/s e s/c
Total de bytes de cliente/servidor
Total de bytes de servidor/cliente
Total de bytes c/s e s/c
Total de pacotes ACK de cliente/servidor
Total de pacotes ACK de servidor/cliente
Total de pacotes por tempo de cliente/servidor
Total de pacotes por tempo de servidor/cliente

Ao analisar os gráficos de *boxplot* com os valores das variáveis de todos os tipos de ataques, procurou-se identificar através daquela variável em questão, se há conjunto de quartis os quais possuam valores diferenciados de todos os outros que estejam no gráfico. Para a classificação dos seis ataques relacionados desse trabalho, foram usados os quatro melhores discriminantes como visto na Tabela 2.

Tabela 2. Discriminantes Escolhidos.

Discriminantes
Total de bytes c/s e s/c
Total de pacotes de servidor para cliente
Total de pacotes ACK de servidor para cliente
Total de bytes de cliente para servidor

A Figura 2 mostra o gráfico *boxplot* da classificação do ataque de SMTP *flood*. Essa classificação é possível visto que os valores da área quartílica do ataque de SMTP *flood* não se sobrepõe a nenhum valor dos outros ataques.

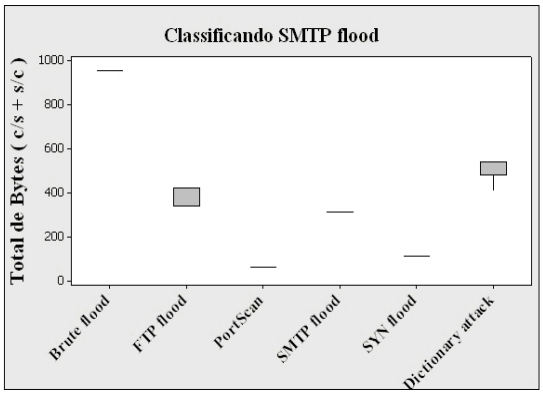

Fig. 2. Classificação do SMTP flood.

A variável discriminante “Total de Bytes c/s e s/c”, onde se contabiliza a soma dos bytes enviados entre cliente e servidor e vice-versa em cada fluxo, classifica os ataques de SMTP *flood* e de *port scan*. Esta mesma variável realiza a classificação do ataque de *Port Scan*. É oportuno salientar que esse gráfico de *boxplot* já não traz os fluxos do ataque de SMTP *flood* (Figura 3).

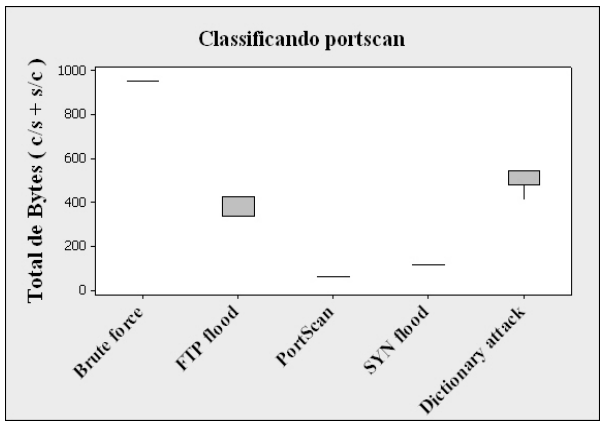

Fig. 3. Classificação do port scan.

A variável discriminante “Total de pacotes servidor/cliente”, onde contabilizamos a soma dos pacotes enviados do servidor para o cliente, classifica o ataque de SYN *flood*, como podemos observar na Figura 4.

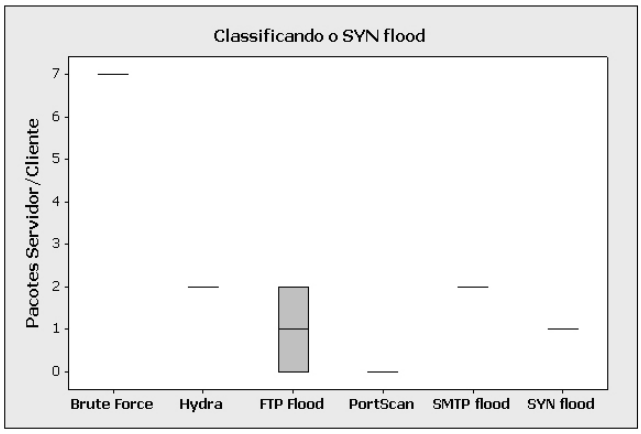

Fig. 4. Classificação do SYN flood.

A variável discriminante “pacotes ACK s/c”, onde contabilizamos o total de pacotes ACK enviados do servidor para o cliente, classifica o ataque *Brute Force*. Os valores dessa variável discriminante, no que diz respeito a este ataque, permanecem

sempre o mesmo por todo o ataque. Podemos observar o *boxplot* de classificação na Figura 5.

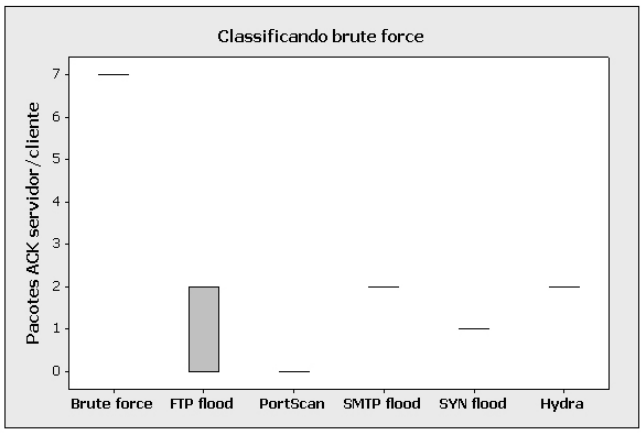

Fig. 5. Classificação do brute force.

A variável “discriminante bytes c/s”, onde contabilizamos a quantidade de bytes enviados do cliente para o servidor, classifica o ataque *Dictionary*. Podemos verificar o *boxplot* de classificação deste ataque na Figura 6.

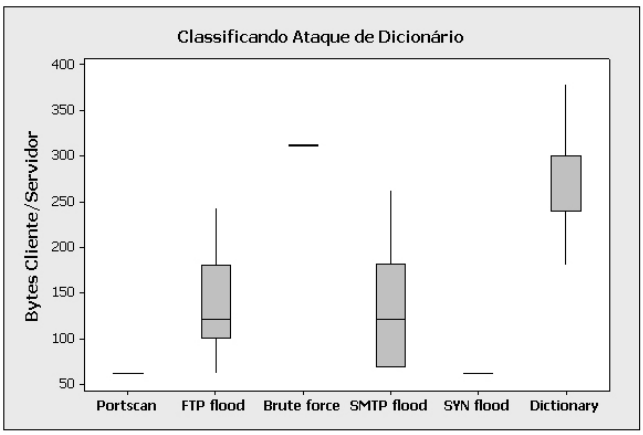

Fig. 6. Classificação do Dictionary.

4.3 Treinamento

Árvores de decisões realizam classificação de dados através de variáveis de entrada pré-estabelecidas. Esse método faz uso da estratégia de dividir para conquistar: um conjunto de dados complexos é quebrado em conjunto de dados menores e a técnica de quebrá-los em conjuntos menores é aplicada recursivamente [10].

Neste trabalho, todos os dados coletados são representados por valores numéricos e por esta causa existe a necessidade da utilização de um algoritmo de árvore de decisão que seja capaz de trabalhar com tais valores. O M5P [11] foi selecionado como o algoritmo de aprendizado. Este método é utilizado para se trabalhar tanto com valores contínuos como discretos. Sua eficiência é demonstrada em [12].

O primeiro passo para a criação de uma árvore de decisão é selecionar o algoritmo de aprendizado, no caso o M5P, e a partir do valor de entrada escolhido, este algoritmo cria uma árvore modelo aonde em cada folha são relacionados os valores instanciados com a variável de entrada escolhida. Este passo é denominado de fase de treinamento.

Os discriminantes selecionados (Tabela 2) foram utilizados individualmente como parâmetro de entrada para o algoritmo M5P, tendo sido geradas, portanto, 4 árvores modelos diferentes. As demais variáveis (Tabela 1) são utilizadas na definição das regras de produção da árvore modelo. Dentre as 4 opções de árvores geradas, a variável “Total de Bytes c/s e s/c” demonstrou ser a melhor escolha como parâmetro de entrada. Os valores das regras de produção utilizadas pela árvore modelo são definidos pelo algoritmo M5P, tendo como base a variável de entrada em questão.

A Figura 7 demonstra a árvore modelo criada com a utilização do algoritmo M5P. Árvore treinada com o primeiro grupo de dados obtidos a partir da coleta de pacotes.

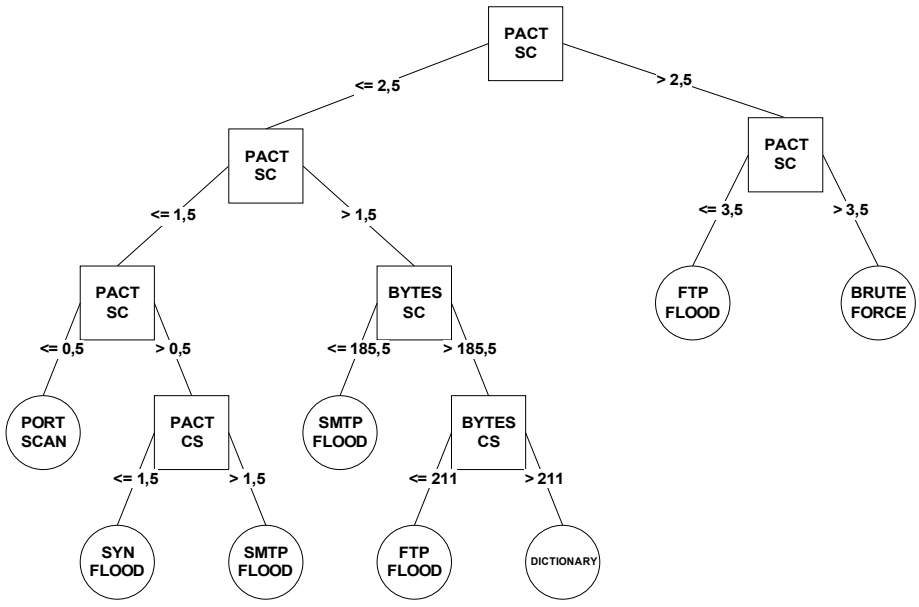

Fig. 7. Árvore Modelo.

Durante o segundo passo, o algoritmo recebe novos fluxos de entrada, porém ainda utilizando a árvore modelo criada durante a fase de treinamento. Neste passo analisamos a eficiência da árvore criada utilizando as variáveis discriminantes selecionadas na seção anterior utilizando novas instancias de dados. Este passo é denominado de fase de validação.

5 Resultados

Após a fase de treinamento, utilizamos o segundo grupo de fluxos de ataque obtidos a partir da coleta de pacotes para a validação da árvore modelo. A importância do processo de classificação manual dos fluxos é notada durante a etapa de validação. A partir da variável identificadora, podemos contabilizar a quantidade total de cada tipo de fluxo de ataque, ao verificarmos a árvore modelo. A Tabela 3 demonstra os valores de taxa de acerto, assim como os valores de falso positivo e falso negativo obtido.

Tabela 3. Classificação dos Ataques

	Taxa de Acerto	*Falso Positivo*	*Falso Negativo*
Port Scan	100%	-	-
SYN flood	91,9%	-	8,1%
FTP flood	99,2%	-	0,8%
Dictionary	98,2%	-	1,8%
Brute Force	98,2%	1,8%	-
SMTP flood	91,1%	8,9%	-

As taxas de acerto variam de 91,1% a 100%. O ataque *port scan* obteve 100% de classificação. Já os ataques SMTP *flood* e *Brute Force* conseguiram 91,1% e 98,2% de classificação respectivamente. O SYN *flood* adquiriu 91,9% de classificação e o FTP *flood* obteve 99,2% de classificação. O ataque de dicionário alcançou 98,2% de classificação. A taxa média de acerto da classificação dos ataques situa-se em 96,4%.

O ataque *Port Scan* foi o único ataque a não apresentar valores de falso positivo e falso negativo. O SYN *flood* teve 8,1% de seus fluxos classificados como SMTP *flood*. O FTP *flood* apresentou 0,8% de fluxos reconhecidos como SMTP *flood.* O *Dictionary* teve 1,8% de fluxos classificados como *Brute Force.*

Apenas dois ataques apresentaram valores de falso positivo. O ataque *Brute Force* obteve 1,8% de fluxos de *Dictionary* reconhecidos como seus, enquanto SMTP *flood* obteve 8,9% de uma combinação de fluxos procedidos de SYN *flood* e FTP *flood*.

6 Conclusão

A técnica de classificação por discriminantes estatísticos, auxiliada pelo uso de árvores de decisão, mostrou que é possível classificar os ataques com a utilização de poucos discriminantes. Este resultado foi obtido com a análise dos discriminantes através de gráficos de *boxplot* e árvores de decisão. Concluímos que os ataques referidos neste trabalho podem ser classificados por um pequeno conjunto de variáveis discriminantes.

De momento nosso trabalho utiliza uma abordagem *offline* para a identificação de ataques, relatando ou não a ocorrência de ataques. Atualmente, estudos estão sendo realizados para implementar um modelo de análise em tempo real.

Apesar de a técnica proposta ter sido aplicada para 6 ataques, ela se apresenta perfeitamente extensível para outros ataques. Pretendemos adicionar novos ataques, buscar novos discriminantes, validar utilizando conjuntos de dados mais conhecidos e simular cenários mais realistas no ambiente de coleta de fluxos, para verificar a ocorrência de mudanças no comportamento e as variáveis em cada ataque.

References

1. Specht, S.M., "Distibuted Denial of Service: Taxonomies of attacks, tools and countermeasures", International conference on parallel and distributed computing systems, PP. 543-550, September 2004.
2. Schuba, C.L., Krsul, I.V., Kuhn, M.G., "Analysis of a Deniel of Service Attack on TCP", IEEE Computer Society, Washington, DC, USA, 1997.
3. Lee, C.B., Roedel, C., Silenok, E., "Detection and Characterization of Port Scan attacks", Department of computer Science & Engineering University of California, San Diego.
4. Moore, A.W., Zuev, D., Crogan, M., "Discriminators for use in flow-based classification", In passive & Measurement Workshop 2003 (PAM2005), August 2005.
5. Barford, P., Kline, J., Plonka, D., Ron, A., "A signal analysis of network traffic anomalies", Internet Measurement Workshop 2002.
6. Brutlag, J., "Aberrant behavior detection in timeseries for network monitoring", USENIX LISA 2000.
7. Holanda Filho, R., Maia, J.E.B., Carmo, M.F.F., Paulino, G., "An Internet Traffic Classification Methodology based on Statistical Discriminators", In: IEEE/IFIP Network Operations & Management Symposium, 2008, Salvador, Bahia. Anais do NOMS 2008, 2008.
8. Kurose, J., Ross K., Redes de computadores e a Internet: Uma abordagem top-down, Pearson Addison Wesley, 2006.
9. Pinkas, B., Sander, T., "Securing Passwords against dictionary attack", ACM conference on computer and communications security, pp. 161-170, 2002.
10. Kirkwood, C. W., "Decision Tree primer", Department of Supply Chain Management Arizona State University Tempe, AZ 85287-4706
11. Wang, Y., Witten, I, H., "Induction of model trees for predicting continuous classes". Poster papers of the 9th European Conference on Machine Learning, 1997.
12. Kalganova, T., "Towards the development of a Problem Solver for the Monitoring and Control of Instrumentation in a Grid Enviroment", School of Engineering and Design Brunel University, 2006.