Detecção de Ataques DDoS com Gráficos de Controle e Bases de Regras Nebulosas

Anderson Fernandes P. Santos[1], Renato S. Silva[2]
[1] Instituto Militar de Engenharia (IME)
Rio de Janeiro – RJ – Brasil
anderson@ime.eb.br
[2] Laboratório Nacional de Computação Científica (LNCC)
Petrópolis – RJ – Brasil
rssr@lncc.br

Abstract- Distributed denial of service attacks (DDoS) are characterized by a great amount of packets that is sent to a specific target server. This behavior, that characterizes this family of attacks, can be used to determine if some server is under attack or not, mainly in the begging of it. We proposed an algorithm to detect this attack by analyzing the behavior of the time interval between packets through theory of control graphs for non normal process with fuzzy rules. From control graphs we use the localization and dispersion estimators to control the variability of time interval between packets, the selected control variable, in specific periods calculated through fuzzy rules.

Palavras-chave: Negação de Serviço, Detecção por Anomalia, Processos Não Normais, Gráficos de Controle e Sistemas Baseados em Regras Nebulosas.

I. INTRODUÇÃO

Ataques de negação de serviço, *DoS*, foram noticiados pela primeira vez por volta de 1996 quando o PANIX (*Public Access Networks Corporation*) tornou-se indisponível durante uma semana por razões até então desconhecidas. [1]

Nos últimos anos, diversas empresas e entidades já sofreram ataques simples e distribuídos, dentre as quais o Yahoo, Amazon, Dell e o E-bay em 2000 [2], Microsoft e SCO em 2004, [3] e recentemente a Georgia em 2008 [4], os sites Facebook, Twitter [5] e diversos sites governamentais e institucionais dos Estados Unidos e Coréia do Sul em 2009 [6].

Estes ataques são baseados, principalmente, no consumo de recursos disponíveis dos servidores alvos, como o canal de comunicação e memória.

Os ataques de negação de serviço distribuídos são constituídos por vários fluxos de pacotes de fontes distintas [7] cujo objetivo é indisponibilizar um serviço oferecido por um servidor. Diversas estratégias já foram desenvolvidas para evitar estes tipos de ataques. Estas estratégias podem ser classificadas, segundo a taxonomia proposta por Mirkovic et al. [8], em quatro grandes categorias: Reconhecimento de Padrão, Comportamento Anômalo, Híbrido e de Terceira Parte.

Os mecanismos de defesa que utilizam o reconhecimento de padrão fazem a identificação através de características específicas de ataques já conhecidos. Estes mecanismos são utilizados em ferramentas de detecção de intrusão, como o *snort* [9]. As ferramentas desta categoria possuem um alto grau de identificação do ataque, ou seja, uma baixa taxa de falsos positivos, porém necessita de que o ataque seja previamente conhecido. Desta forma, ataques novos dificilmente são identificados, e pequenas variações nos ataques já conhecidos dificultam a sua identificação.

Os mecanismos de defesa classificados como Comportamento Anômalo são aqueles que identificam um ataque a partir da mudança de comportamento de alguma característica da rede [10] ou do tráfego [11], por exemplo. Os mecanismos com esta característica possuem uma significativa taxa de falsos positivos, porém podem identificar ataques ainda não conhecidos.

Os mecanismos híbridos são aqueles que são constituídos pelas duas características anteriores e assim, podem identificar tanto ataques novos quanto antigos com uma significativa taxa de sucesso. Limwiwatkul e Rungsawang propuseram um mecanismo desse tipo em [12].

Os mecanismos de terceira parte são constituídos de outras ferramentas acessórias que são usados para identificar o ataque, como os mecanismos de *traceback* [13] ou marcação de pacotes [14].

Neste artigo é proposto um algoritmo, da família de mecanismos de detecção por comportamento anômalo, que visa detectar ataques de negação de serviço baseados no monitoramento do intervalo de tempo entre os pacotes, medidos em janelas cujos tamanhos são obtidos através de uma base de regras nebulosas.

A partir do tamanho de janela inferido, através da base de regras nebulosas, os valores de intervalos de tempos são calculados e analisados. Caso estes intervalos excedam os limites impostos pelos gráficos de controle, são emitidos alertas de ataques de negação de serviço. Caso contrário, é considerado que não há ataques acontecendo.

Este artigo está composto por sete seções. Na Seção II é descrito o *trace* usado neste trabalho. Na Seção III é descrita a teoria de gráficos de controle para processos não normais. Na Seção IV é descrita a teoria de sistemas nebulosos, que fundamenta o uso de base de regras nebulosas. A Seção V apresenta o algoritmo proposto que foi chamado de *Algoritmo de Detecção de Anomalias com Janelas Adaptativas*. Na Seção VI são descritos os experimentos e resultados obtidos e a Seção VII conclui este artigo.

II. *TRACES* UTILIZADOS

A avaliação do algoritmo proposto foi baseada em um *trace* de pacotes gerado a partir de simulações. Optou-se pela utilização do *trace* do DARPA [15], um estudo realizado pelos Laboratórios de Pesquisa do MIT (*Lincoln Laboratory*) e da Força Aérea Norte Americana (*Air Force Research Laboratory*) sobre ataques cibernéticos à redes de computadores. Neste estudo foi modelada uma rede de computadores de um quartel da Força Aérea Norte Americana e posteriormente foram realizadas simulações. A partir destas simulações, foram gerados *traces*.

Foram realizadas simulações nos anos de 1998, 1999 e 2000. De cada simulação foram gerados *traces* de cinco semanas, onde cada semana é composta de cinco dias (segunda à sexta). As primeiras três semanas são utilizadas para treinamento, onde a primeira e terceira semanas são livres de ataques. A segunda semana possui ataques conhecidos e identificados. As demais semanas são usadas para testes/validação dos algoritmos e possuem ataques. As informações apresentadas nestes *traces* foram coletados através da ferramenta *tcpdump* [16].

Para o presente estudo foi utilizada como variável de controle o intervalo de tempo entre os pacotes, a partir do *trace* realizado no ano de 1999. Esta escolha se fundamenta no fato de que ataques de negação de serviço (distribuídos ou não), quando têm como meta exaurir o canal de comunicação, emitem uma quantidade elevada de pacotes para o servidor alvo. Assim, o intervalo de tempo entre os pacotes, durante o ataque, diminui sensivelmente.

Fizeram parte dos dados utilizados os intervalos de tempo entre pacotes das semanas indistintamente, ou seja, as três semanas iniciais para treinamento e as demais para testes foram tratadas da mesma maneira. Isso se deve ao fato de que, em condições normais, não é possível dizer se a rede está sob ataque de negação de serviço ou, por exemplo, pelo excesso de requisições a determinados serviços (*flash crowds*).

III. GRÁFICOS DE CONTROLE PARA PROCESSOS NÃO NORMAIS

O conceito de gráficos de controle foi introduzido inicialmente pelo Dr W. Shewhart na década de 20 no processo de fabricação dos telefones, no Laboratório Bell [17]. Até os dias de hoje, os gráficos de controle são usados para determinar se um processo está controlado ou não. A idéia básica dos gráficos de controle é que todo processo possui alguma variabilidade, e para isto há duas causas básicas: uma chamada de causa comum e presente em todos os processos que existem. A segunda é chamada de especial e é provocada por outros eventos. Destas duas causas, a primeira é intrínseca ao sistema e a segunda é externa ao sistema e pode ser identificada e corrigida [18].

Entretanto, estes gráficos são indicados para serem usados, preferencialmente, quando os dados são normalmente distribuídos e esta situação nem sempre é encontrada em casos reais, como Pyzdek [19] demonstrou para o processo de galvanização.

Desta forma, o gráfico de controle proposto por Duclos é uma escolha natural para o tratamento de dados que não são normalmente distribuídos. Duclos propôs a construção de um novo gráfico de controle indicado para ser utilizado quando o processo é não normal [20, 21]. Neste caso são utilizados como estimador de localização a média, e como estimador de dispersão o desvio padrão. Desta forma, o gráfico de controle proposto por Duclos torna-se uma generalização do gráfico de controle de Shewhart.

Inicialmente foi definido o conceito de janela de observação como sendo o intervalo de tempo sob o qual será realizado o cálculo da variável de controle selecionada (intervalo de tempo entre os pacotes da rede).

Tendo por base estes gráficos de controle, a cada janela de observação i e $i+1$, os estimadores de localização e dispersão, através do método de Duclos, são calculados e comparados.

Os limites de controle de cada janela de observação são calculados a partir dos valores dos estimadores. No caso do i-ésimo intervalo, estes limites são dados por uma vizinhança de centro no valor do estimador de localização e raio igual ao triplo do desvio padrão do mesmo intervalo, ou seja,

$$i = [\mu_i - 3.\sigma_i, \mu_i + 3.\sigma_i] \quad (1)$$

O fator triplo advém do algoritmo de Duclos [20] e foi escolhido devido à comprovação da eficiência prática em [22]. Deve-se salientar a coincidência ocorrida entre este valor e o clássico obtido por Shewhart na teoria clássica [17].

A identificação do ataque é realizada através da comparação dos estimadores em janelas de observação vizinhas *i* e *i+1*. Se o estimador de localização do intervalo *i+1* for maior que o limite superior de controle *i*, considerar-se-á que há mudança de comportamento e, conseqüentemente, um ataque está sendo finalizado (vide Eq. 2).

$$\mu_{i+1} > \mu_i + 3.\sigma_i \quad (2)$$

De forma similar, se o valor do estimador de localização do intervalo *i+1* for menor, que o limite inferior de controle *i*, considerar-se-á que houve mudança de comportamento e que um ataque está sendo iniciado (vide Eq. 3).

$$\mu_i < \mu_{i-1} - 3.\sigma_{i-1}, com\ \mu_i > 0 \quad (3)$$

IV. Sistemas Nebulosos

Algumas vezes a representação do conhecimento não pode ser realizada através de variáveis quantitativas, principalmente quando se está buscando uma forma de representar o conhecimento humano. Em resposta a esse problema de imprecisão da informação que surgiu a teoria dos conjuntos nebulosos [23]. A partir da aplicação desta teoria em sistemas baseados em conhecimento surge a lógica nebulosa, que é utilizada para o desenvolvimento de sistemas que objetivam o controle de processos [24].

Uma vez que a base de conhecimento está fundamentada em variáveis lingüísticas, ou seja, através do conhecimento representado de forma imprecisa, torna-se necessário o uso de um sistema que possa converter os valores numéricos em variáveis lingüísticas e vice-versa. Estes sistemas, conhecidos como baseados em regras nebulosas possuem a estrutura geral mostrada na Figura 1.

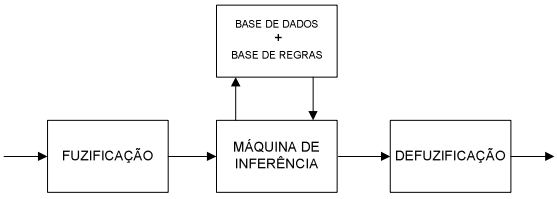

Figura 1: Sistema baseado em regras;

Na Figura 1, o termo fuzificação está relacionado com a conversão de termos numéricos em termos nebulosos. Assim, são formuladas funções de pertinência para cada variável, que englobem totalmente o domínio dos valores escalares encontrados na entrada do sistema. Estas funções são responsáveis por representar o conhecimento humano. As funções mais comuns encontradas são a triangular, trapezoidal e *singleton*.

Com a informação transformada em variáveis lingüísticas, a máquina de inferência é acionada e a base de conhecimento é consultada. Nesta base, as regras são formadas por sentenças do tipo *se-então*, e a partir do acionamento das regras de produção, uma resposta é gerada pelo sistema. Estes resultados podem ser do tipo lingüístico ou funcional.

Os modelos lingüísticos são caracterizados pelo fato de que os resultados obtidos em cada regra são termos nebulosos pertencentes a um conjunto fixo de termos. No caso dos modelos funcionais, para cada regra é obtido um valor para uma variável de controle. O valor final é resultado da média ponderada dos valores obtidos levando-se em consideração o grau de compatibilidade com a premissa da regra.

A fase final corresponde em transformar os valores obtidos a partir da máquina de inferência em valores escalares. Este processo é chamado de defuzificação e é responsável por quantizar o resultado obtido pelo processo de inferência.

O valor do tamanho da janela de observação está diretamente relacionada com a qualidade dos resultados obtidos, uma vez que os cálculos dos estimadores de localização e dispersão estão relacionados com a combinação de todos os valores em análise, neste trabalho é o caso do intervalo de tempo entre os pacotes. Uma janela de observação muito grande tende a esconder comportamentos ocorridos na janela. Uma janela muito pequena tende a discretizar excessivamente o trabalho em estudo, além de aumentar significativamente o processamento.

Assim foram elaboradas funções de pertinência para o dimensionamento da janela de observação, considerando como variáveis de entrada os estimadores de localização e dispersão, exibidos na Figura 2 e 3, respectivamente, que serão usados na etapa de fuzificação.

Na defuzificação foi usada a função de pertinência, Figura 4, para o valor da janela de observação calculada pelas regras de inferência da base de dados.

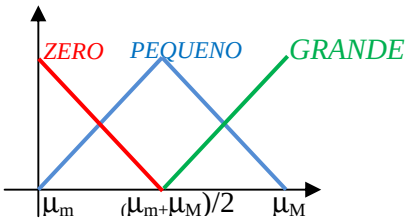

Figura 2: Função de pertinência para o estimador de localização.

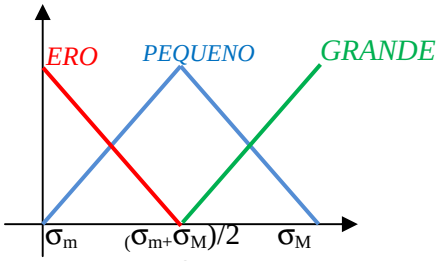

Figura 3: Função de pertinência para o estimador de dispersão.

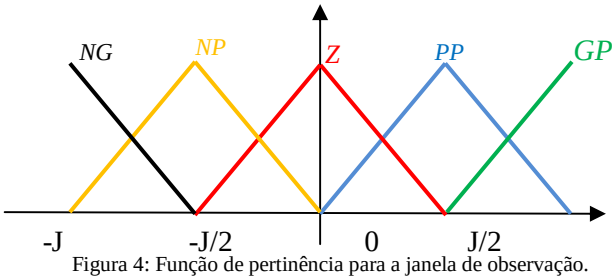

Figura 4: Função de pertinência para a janela de observação.

A base de regras, específica para este problema, é definida a partir das variáveis lingüísticas representativas dos estimadores. Assim, foram utilizados os seguintes conjuntos:

Regra 1) Se os estimadores de localização e dispersão são próximos de zero, então considerar um grande aumento no valor da janela de observação, uma vez que entende-se que neste caso não há problema acontecendo e que poder-se-ia aumentar o respectivo tamanho;

Regra 2) Se o estimador de localização é próximo de zero e o de dispersão é pequeno ou o estimador de localização é pequeno e o de dispersão é próximo de zero, então considerar um pequeno aumento no valor da janela de observação. Neste caso, entende-se que há alguma variabilidade no processo, porém esta variabilidade não representa nenhum descontrole no processo;

Regra 3) Se os estimadores de localização e dispersão são muito grandes, então considerar uma significativa diminuição no valor da janela. Neste caso, percebe-se que os valores estão muito grandes e poderão "esconder" o fenômeno, portanto sendo necessário reduzir o respectivo tamanho;

Regra 4) Se o estimador de localização é muito grande e o de dispersão é pequeno ou próximo de zero, ou se o de localização é pequeno ou próximo de zero e o dispersão é muito grande, então considerar uma pequena diminuição no valor da janela. Neste caso são dois fenômenos distintos, porém resultando em um mesmo fenômeno: o de um grande intervalo para a análise. Neste caso, é necessário diminuir o tamanho para melhorar a análise;

Regra 5) Se os estimadores de localização e dispersão forem pequenos, então considerar um pequeno aumento no valor da janela. Neste caso, pequenos tamanhos de janela tornam o algoritmo ineficiente.

V. Algoritmo de Detecção de Anomalias com Janelas Adaptativas

O algoritmo de detecção de anomalias com janelas adaptativas é mostrado na Figura 5 e a seguir é apresentado uma descrição dos seus passos.

O primeiro passo do algoritmo é a seleção de uma janela inicial J_0 de observação (Passo 1). Este valor inicial serve de referência para os cálculos que serão realizados posteriormente.

A partir de J_0 são calculados os valores dos estimadores de localização e dispersão (Passos 2 e 3). Nesta primeira passagem, não são realizadas comparações (Passo 4) e verificações de mudança de comportamento (Passo 5) uma vez que não há intervalos anteriores. A partir da base de regras nebulosas, para esta primeira janela a ser calculada J_0, o primeiro valor de janela de observação é calculado (Passos 8, 9, 10 e 11): J_1.

A partir da segunda janela de observação o algoritmo executa todas as ações pertinentes. Assim com o valor da *i*-ésima janela de observação pelo algoritmo de Duclos os estimadores de localização e dispersão são calculados (Passo 2). Estes estimadores são armazenados (Passo 3) e é realizada a comparação destes estimadores com a janela de observação anterior *i-1*. Se os valores da *i*-ésima janela estiverem fora dos limites esperados (equações 2 e 3) serão lançados alertas de ataques (final e início de ataque, respectivamente) (Passos 5, 6).

Com os valores dos estimadores de localização e dispersão calculados estes serão traduzidos para variáveis lingüísticas (fuzificação – Passo 8) e, a partir da máquina de

inferência (Passos 9 e 10) as bases de regras nebulosas serão acionadas, conforme o caso e um novo valor de tamanho de janela de observação será fornecido, ainda como variável linguística. A partir deste, um valor escalar é fornecido (defuzificação – Passo 11), terminando o laço de cálculo da janela de observação do intervalo *i+1*-ésimo.

Algoritmo de Detecção de Anomalias com Janelas Adaptativas

DeteccaoAnomaliasJanelasAdaptativas
1. J_0=CriarPrimeiraJanela()
2. [μ, σ]=Duclos(J_{i+1})
3. Armazene estimadores
4. Realize a comparação em *i* e *i+1*
5. Se tem mudanças então
6. Notifique o usuário
7. Senão Se for o fim das análises então
8. Fuzificação
9. Máquina de inferência
10. Faça a busca na Base de Conhecimento
11. Defuzificação
12. Fim-se
13. Fim-se
14. Se não for o final do trace então
15. Retorne ao Passo 2
16. Fim-se

Figura 5: Algoritmo de Detecção de Anomalias com Janelas Adaptativas.

VI. Experimentos e Resultados

A segunda, quarta e quinta semanas de ataques do *trace* utilizado, foram submetidos ao algoritmo. O objetivo deste experimento era detectar apenas os ataques *apache2*, *mailbomb*, *smurf* e *udpstorm*.

Outros ataques puderam ser detectados, devido a resultados indiretos ocasionados destes. É o caso dos demais ataques DoS existentes na base, exceto *teardrop* e *dosnuke*, que usam brechas na implementação do TCP/IP para proferirem seus ataques.

Os ataques que foram alvos nesta pesquisa obtiveram uma margem significativa de identificações positivas, vide tabela 1, exceto com o ataque *udpstorm*, cuja única instância não pode ser identificada.

Os demais ataques presentes no *trace* utilizado puderam ser identificados, na grande maioria com mais de 50% das suas ocorrências, vide tabela 2, dos quais os ataques *warezmaster* e *selfping* tiveram 100% das suas ocorrências identificadas.

Tabela 1: Ataques alvos neste estudo.

Ataque	*Identificação*		*Falso Negativo*	
	Qtde	%	Qtde	%
Apache2	2	66%	1	33%
Mailbomb	2	33%	4	66%
Smurf	3	60%	2	40%
Udpstorm	0	0%	1	100%

Tabela 2: Ataques que não foram foco do estudo.

Ataque	*Identificação*		*Falso Negativo*	
	Qtde	%	Qtde	%
back	3	50%	3	50%
pod	3	50%	3	50%
land	2	50%	2	50%
crashiis	1	9%	10	91%
synflood	1	17%	5	83%
processtable	1	25%	3	75%
arppoison	2	40%	3	60%
dosnuke	0	0%	4	80%
syslogd	3	75%	1	25%
selfping	2	100%	0	0%
tcpreset	2	66%	1	33%
teardrop	0	0%	2	100%
warezmaster	4	100%	0	0%

VII. Conclusões

Embora o assunto de ataques de negação de serviços distribuídos não seja novo, o mesmo ainda não está resolvido, conforme tem sido noticiado [5, 6]. Estes ataques têm sofrido alguma modernização durante os anos, porém o seu *modus operandi* continua exatamente o mesmo desde quando foram usados pela primeira vez (ataques que consomem recursos).

Muitas pesquisas foram desenvolvidas propondo soluções e ainda não há uma definitiva para o assunto. Assim esta área de pesquisa ainda apresenta assuntos que podem ser explorados buscando soluções que permitam evitar que serviços essenciais não se tornem indisponíveis por ataques.

O algoritmo proposto neste artigo demonstrou, pelos resultados exibidos na seção VI, ser adequado para a identificação de ataques de negação de serviço, distribuídos ou não, que buscam exaurir o canal de comunicação. Outro fato a ser destacado é a total ausência de informação do ataque, como por exemplo o protocolo utilizado. Isto

permite que novos ataques possam ser identificados com este algoritmo.

Além disso, o fato de não ser preciso período prévio sem ataques permite um grau de aplicabilidade maior que as demais técnicas existentes, que normalmente necessitam de períodos prévios de conhecimento sem ataques existentes.

Entretanto, por identificar o início ou final do ataque sem identificar exatamente qual ataque esteja ocorrendo, permite que sejam realizados trabalhos futuros no sentido de agregar outras técnicas que permitam determinar a natureza do ataque e propor contramedidas que permitam manter a disponibilidade do serviço que esteja ocorrendo. Além disso, a partir da análise de resultados, vide seção VI, percebe-se que é possível melhorar este algoritmo pela adoção de critérios bayesianos de informação (BIC) na seleção dos períodos de anomalia.

REFERÊNCIAS

[1] Managing the Threat of Denial-of-Service Attacks - CERT® Coordinator Center - http://www.cert.org/archive/ pdf/Managing DoS.pdf

[2] Paul, B. – "DDoS: Internet Weapons of Mass Destruction" – Network Computing – http://networkcomputing.com/ 1201/1201f1c1.html, Jan 8TH , 2001

[3] Keizer, G. K. – "Mydoom DoS Attack On Microsoft Falters Gregg Keizer", CRN - http://www.crn.com/news/security/ 18831398/mydoom-dos-attack-on-microsoft-falters.htm;

[4] Melikishvili, A. "The Cyber Dimension of Russia's Attack on Georgia" - Eurasia Daily Monitor Volume: 5 Issue: 175, September 12, 2008

[5] Van Buskirk, E. – "Denial-of-Service Attack Knocks Twitter Offline"- http://www.wired.com/epicenter/2009/08/ twitter-apparently-down/

[6] Benson, P., Greene, R. A. – "U.S. government sites among those hit by cyberattack" - CNN - http://edition.cnn.com/ 2009/ TECH/ 07/08/government.hacking/index.html

[7] Mirkovic, J., Prier, G., Reiher, P., 2002, "Attacking DDoS at the Source", 10th IEEE International Conference on Network Protocols, Paris, France, 12-15 Nov.

[8] Mirkovic, J., Martin, J.and Re.iher, P., "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," *ACM Sigcomm Computer Comm. Rev.*, vol. 34, no. 2, 2004, pp. 39–53.

[9] SNORT – http://www.snort.org

[10] Siaterlis, C., and Maglaris, V., "Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics", *10th IEEE Symposium on Computer and Communications. ISCC2005*.

[11] Gil,T.M. and Poletto, M., "MULTOPS: a data-structure for bandwidth attack detection", In *Proceddings of 10th Usenix Security Symposium*, August 2001.

[12] Limwiwatkul, L. and Rungsawang, A. "Distributed Denial of Service Detection using TCP/IP Header and Traffic Measurement Analysis". *International Symposium on Communications and Information Technologies 2004 (ISCTI2004)* Sapporo, Japan, October 26-29, 2004.

[13] Wong, T. Y., Law, K. T., Lui, J. C. S. and Wong, M. H., "An Efficient Distributed Algorithm to Identify and Traceback DDoS Traffic", The Computer Journal 2006 49(4):418-442.

[14] Savage, S., Wetherall, D., Karlin, A., Anderson, T., "Pratical Network Support For IP Traceback" – *ACM Special Interest Group on Data Communication 2000 – Stockholm – Sweeden*.August 28, September 1 – 2000.

[15] DARPA - Defense Advanced Research Projects Agency - http://www.ll.mit.edu/IST/ideval/index.html.

[16] TCPDUMP - http://www.tcpdump.org.

[17] Montgomery, D. C., Runger, G. C., *"Applied Statistics and Probability for Engineers"*. John Wiley & Sons - 1999.

[18] ENGINEERING STATISTICS HANDBOOK – http://www.itl.nist.gov

[19] Pyzdek, T. – "Non-Normal Distributions in the Real World" - *Quality Engineering*: "Why Normal Distributions Aren't [All That Normal]," 1995, 7(4), pgs. 769-777

[20] Duclos, E. and Pillet, M. "Contribution à la Maîtrise Statistique des Precédés, Cas des Procédés Non Normaux". PhD, Université de Savoie. 1997.

[21] Duclos, E., Pillet, M. and Avrillon, L. "The L-Chart for Non-Normal Processes". Quality Technology & Quantitative Management. Vol 2, No 1, pp. 77-90, ICQAQM 2005.

[22] Santos, A.F.P., Silva, R.S., "Detecting Bandwidth DDoS Attacks with Control Charts", *15th IEEE International Conference on Networks*, Adelaide, Austrália, 19-21 Nov 2007.

[23] Zadeh, L.A. – "Fuzzy Sets, Information and Control" – Fuzzy Sets 8:338-353, 1965.

[24] Sandri, S., Correa, C., "Lógica Nebulosa" - V Escola de Redes Neurais, ITA, São José dos Campos, SP, Brasil, 19 Jul 1999.

[25] Intrusion detection attacks database http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html

VIII. DESCRIÇÃO DOS ATAQUES

Tabela 3: Descrição Dos Ataques [25]

Ataque	Descrição
Apache2	Ataque de negação de serviço realizado contra servidores web apache onde um cliente envia uma requisição com diversos cabeçalhos HTTP. Se o servidor receber muitas destas requisições ele irá tornar-se lento, e eventualmente poderá indisponibilizá-lo.
Smurf	Os atacantes enviam um pacote de requisição ICMP echo direcionado para o endereço de broadcast de várias redes com o endereço do servidor alvo. Todos os equipamentos destas redes responderão enviando um pacote resposta ICMP para o servidor alvo.
Mailbomb	O atacante envia muitas mensagens para o servidor, sobrecarregando a fila de correio do servidor e causando, provavelmente, falha no sistema.
UdpStorm	Provoca congestionamento na rede e lentidão. Quando uma conexão é estabelecida entre dois serviços UDP, estes dois produzem uma saida em uma taxa muito alta que poderá provocar um ataque de negação de serviço.