

**UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE INFORMÁTICA**

**Metodologia para avaliação de sistemas de detecção de
intrusão**

Leonardo Lemes Fagundes

**Prof. Luciano Paschoal Gaspari
Orientador**

*Monografia submetida como
requisito parcial para a obtenção do
título de Bacharel em Informática.*

São Leopoldo, novembro de 2002

Resumo

A disseminação dos sistemas de detecção de intrusão, Intrusion Detection Systems (IDSs), resultaram no aumento da necessidade de metodologias que auxiliem no processo de escolha dessas ferramentas. Para tal, diversas abordagens têm sido desenvolvidas, mas (a) requerem a realização de procedimentos complexos que consomem muito tempo para serem realizados e (b) não possuem nenhuma forma de sistematização. Este artigo propõe uma metodologia alternativa para avaliar IDSs em relação aos seguintes aspectos: capacidade de detecção, taxas de falsos positivos geradas e escalabilidade dos sistemas avaliados. Essa metodologia foi desenvolvida tendo em mente o uso em qualquer organização e propõe uma seqüência de procedimentos sistemáticos que podem ser executados em um curto período de tempo. Ao contrário de outras metodologias, o conhecimento sobre a implementação dos IDSs não é necessário.

Abstract

The dissemination of Intrusion Detection Systems (IDSs) has resulted in increase of the need for methodologies that help in the process of choosing such systems. To do that several approaches have been developed, but (a) they require the accomplishment of complex procedures that take too much time to be executed and (b) do not provide any systematic way of executing them. This paper proposes an alternative methodology to evaluate IDSs regarding the following aspects: detection capability, false positive rate and scalability. This methodology was developed having in mind its usage in any organization and proposes a sequence of systematic procedures that can be executed in a short time period. Besides, it does not require the knowledge on the implementation of the IDSs evaluated (as opposed to most available approaches).

Lista de abreviaturas

CPU	<i>Central Process Unit</i>
DNS	<i>Domain Name server</i>
DoS	<i>Deny of Service</i>
FTP	<i>File Transfer Protocol</i>
GPL	<i>General Public License</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IDS	<i>Intrusion Detection System</i>
IDSs	<i>Intrusion Detection Systems</i>
IP	<i>Internet Protocol</i>
IRC	<i>Internet Relay Chat</i>
Mbps	<i>Mega bits por segundo</i>
MIT	<i>Massachusetts Institute of Technology</i>
MTU	<i>Maximum Transfer Unit</i>
RFC	<i>Request For Comments</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagrama Protocol</i>

Sumário

1 <u>Introdução</u>	12
2 <u>Metodologias para avaliação de IDSs</u>	14
2.1 <u>Puketza et al. 1997</u>	14
2.1.1 <u>Ataques concorrentes</u>	14
2.1.2 <u>Avaliação</u>	15
2.2 <u>Lippmann et al. 2000</u>	16
2.2.1 <u>Método para classificação dos ataques</u>	16
2.2.2 <u>Tipos de tráfego</u>	18
2.2.3 <u>Ambiente de teste</u>	19
2.3 <u>Alessandri, 2000</u>	20
2.3.1 <u>Metodologia</u>	20
2.3.2 <u>Avaliação do IDS</u>	21
2.4 <u>Barber, 2001</u>	21
2.5 <u>Síntese das propostas de metodologias existentes</u>	22
3 <u>A metodologia de avaliação proposta</u>	26
3.1 <u>Seleção dos Ataques</u>	26
3.1.1 <u>Ataques propostos</u>	26
3.1.2 <u>Descrição técnica do ataque</u>	33
3.2 <u>Seleção de ferramentas</u>	37
3.3 <u>Ferramentas de ataque</u>	38
3.4 <u>Geração do tráfego do cenário de avaliação</u>	39
3.4.1 <u>Coleta do tráfego de ataque</u>	39
3.4.2 <u>Geração do tráfego de fundo</u>	41
3.5 <u>Montagem do ambiente de avaliação</u>	42
3.6 <u>Análises dos IDSs</u>	43
3.6.1 <u>Capacidade de detecção</u>	43
3.6.2 <u>Escalabilidade</u>	44
3.6.3 <u>Taxa de falsos positivos</u>	47
4 <u>Estudo de caso</u>	49

<u>4.1</u>	<u>Capacidade de detecção</u>	49
<u>4.2</u>	<u>Escalabilidade</u>	50
<u>4.2.1</u>	<u>Análise de escalabilidade do <i>Snort</i></u>	50
<u>4.2.2</u>	<u>Comparação dos resultados da análise de escalabilidade</u>	56
<u>4.3</u>	<u>Taxas de falsos positivos</u>	60
<u>4.3.1</u>	<u>Análise das taxas de falsos positivos gerados do <i>Snort</i></u>	60
<u>4.3.2</u>	<u>Análise das taxas de falsos positivos gerados do <i>Firestorm</i></u>	61
<u>4.3.3</u>	<u>Comparação dos resultados da análise de taxas de falsos positivos</u>	62
<u>5</u>	<u>Considerações finais</u>	64
<u>5.1</u>	<u>Seleção de ataques</u>	64
<u>5.2</u>	<u>Capacidade de detecção</u>	64
<u>5.3</u>	<u>Análise de escalabilidade</u>	65
<u>5.4</u>	<u>Análise das taxas de falsos positivos</u>	65
<u>5.5</u>	<u>Possibilidade de expansão da metodologia</u>	65
<u>5.6</u>	<u>Trabalhos futuros</u>	65

Lista de figuras

<u>Figura 1 - Ataque concorrente [Puketza et al. 1997]</u>	15
<u>Figura 2 - Níveis de privilégios [Kendall, 1999]</u>	17
<u>Figura 3 - Ambiente de teste usado na avaliação descrita em [Lippmann et al. 2000]</u>	19
<u>Figura 4 - Avaliação do IDS [Alessandri, 2000]</u>	21
<u>Figura 5 - Exemplo de evasão da letra “A” [Ptacek e Newsham, 1998]</u>	27
<u>Figura 6 - Exemplo de inserção da letra “X” [Ptacek e Newsham, 1998]</u>	27
<u>Figura 7 - Evolução das ferramentas de ataque</u>	37
<u>Figura 8 - Ambiente de rede para geração do tráfego dos cenários de teste</u>	40
<u>Figura 9 - Seqüência de atividades a serem realizadas para coletar o tráfego de ataque</u>	41
<u>Figura 10 - Ambiente de rede para o cenário de avaliação</u>	42
<u>Figura 11 - Seqüência de atividades a serem realizadas para analisar capacidade de detecção dos IDSs</u>	43
<u>Figura 12 - Seqüência de reprodução dos tráfegos utilizados na análise de escalabilidade</u>	44
<u>Figura 13 - Exemplo dos resultados da análise de escalabilidade em relação aos ataques de negação de serviços</u>	1
<u>Figura 14 - Exemplo de comparação dos resultados da análise de escalabilidade em relação aos ataques de negação de serviços</u>	46
<u>Figura 15 - Exemplo de comparação entre as taxas de falsos positivos geradas pelos IDSs avaliados</u>	48
<u>Figura 16 - Análise de escalabilidade do <i>Snort</i> em relação aos ataques de inserção</u>	51
<u>Figura 17 - Análise de escalabilidade do <i>Snort</i> em relação aos ataques de evasão</u>	52
<u>Figura 18 - Análise de escalabilidade do <i>Snort</i> em relação aos ataques de varredura de portas</u>	52
<u>Figura 19 - Análise de escalabilidade do <i>Snort</i> em relação aos ataques de negação de serviço</u>	53
<u>Figura 20 - Análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de inserção</u>	54
<u>Figura 21 - Análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de evasão</u>	55
<u>Figura 22 - Análise de escalabilidade do <i>Firestormt</i> em relação aos ataques de varredura de portas</u>	55
<u>Figura 23 - Análise de escalabilidade do <i>Firestormt</i> em relação aos ataques de negação de serviço</u>	56

<u>Figura 24 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de evasão</u>	57
<u>Figura 25 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de inserção</u>	58
<u>Figura 26 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de varredura de portas</u>	59
<u>Figura 27 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de negação de serviço</u>	59
<u>Figura 28 - Comparação entre os resultados obtidos na análise das taxas de falsos positivos</u>	63

Lista de tabelas

<u>Tabela 1 - Avaliação do NSM: valores retornados [Puketza et al. 1997]</u>	16
<u>Tabela 2 - Tipos de ações possíveis [Korba, 2000]</u>	18
<u>Tabela 3 - Características das principais metodologias para avaliação de IDSs</u>	24
<u>Tabela 4 - Descrição técnica do cenário inicial de ataques proposto</u>	34
<u>Tabela 5 - Seleção do cenário de avaliação</u>	37
<u>Tabela 6 - Ferramentas para reproduzir os ataques selecionados</u>	38
<u>Tabela 7 - Exemplo de resultados da análise de capacidade de detecção</u>	44
<u>Tabela 8 - Exemplo de resultados da análise de escalabilidade</u>	45
<u>Tabela 9 - Exemplo dos resultados da análise de escalabilidade em relação aos ataques de negação de serviço</u>	1
<u>Tabela 10 - Exemplo dos resultados da análise das taxas de falsos positivos em relação aos ataques de negação de serviço</u>	47
<u>Tabela 11 - Exemplo dos resultados da análise das taxas de falsos positivos obtidos por dois IDSs</u>	48
<u>Tabela 12 - Resultados obtidos na análise da capacidade de detecção</u>	50
<u>Tabela 13 - Resultados obtidos na análise de escalabilidade do <i>Snort</i> em relação aos ataques de inserção</u>	51
<u>Tabela 14 - Resultados obtidos na análise de escalabilidade do <i>Snort</i> em relação aos ataques de evasão</u>	51
<u>Tabela 15 - Resultados obtidos na análise de escalabilidade do <i>Snort</i> em relação aos ataques de varredura de portas</u>	52
<u>Tabela 16 - Resultados obtidos na análise de escalabilidade do <i>Snort</i> em relação aos ataques de negação de serviços</u>	53
<u>Tabela 17 - Resultados obtidos na análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de inserção</u>	53
<u>Tabela 18 - Resultados obtidos na análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de inserção</u>	54
<u>Tabela 19 - Resultados obtidos na análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de varredura de portas</u>	55
<u>Tabela 20 - Resultados obtidos na análise de escalabilidade do <i>Firestorm</i> em relação aos ataques de negação de serviços</u>	56
<u>Tabela 21 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de evasão</u>	56
<u>Tabela 22 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de inserção</u>	57

<u>Tabela 23 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de varredura de portas</u>	58
<u>Tabela 24 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de negação de serviço</u>	59
<u>Tabela 25 - Snort: percentual de falsos positivos em relação aos ataques evasão</u>	60
<u>Tabela 26 - Snort: percentual de falsos positivos em relação aos ataques inserção</u>	60
<u>Tabela 27 - Snort: percentual de falsos positivos em relação aos ataques de varredura de portas</u>	60
<u>Tabela 28 - Snort: percentual de falsos positivos em relação aos ataques negação de serviço</u>	61
<u>Tabela 29 - Firestorm: percentual de falsos positivos em relação aos ataques evasão</u> .	61
<u>Tabela 30 - Firestorm: percentual de falsos positivos em relação aos ataques inserção</u>	61
<u>Tabela 31 - Firestorm: percentual de falsos positivos em relação aos ataques de varredura de portas</u>	62
<u>Tabela 32 - Firestorm: percentual de falsos positivos em relação aos ataques de negação de serviço</u>	62
<u>Tabela 33 - Comparação entre os resultados obtidos na análise das taxas de falsos positivos</u>	62

Agradecimentos

Ter a quem agradecer é uma graça divina, pois isto significa que ao longo dessa caminhada não estive sozinho e que comigo estiveram todos aqueles que de alguma forma desejaram esta vitória.

Deus se faz presente em todos os momentos da minha vida, todas as grandes conquistas que eu obtive foram graças a ele e, portanto, agradeço a Deus hoje e sempre pelas vitórias a mim concedidas.

Ao falar em Deus, em formação religiosa, lembro em especial dos meus avós a quem devo toda a minha iniciação religiosa. A dona Sunilda e o seu Pedro sempre pediram a Deus para que os obstáculos a minha frente, não fossem maiores do que a minha disposição ou força.

Aos demais integrantes da minha família (tio Marcos, tia Rose, tia Solanes, a prima Roberta, a Paolinha), pessoas que sempre acreditaram que eu iria conseguir e que com certeza entenderam a minha ausência ao longo dos últimos meses e durante vários outros períodos dessa jornada.

Os meus colegas do Instituto de Informática eu agradeço imensamente. No entanto, duas pessoas em especial eu agradecerei para sempre, a Madelaine e o Daniel Ferreira, pois eles acompanharam no dia a dia todas as minhas angústias. O Daniel inclusive teve seus ouvidos sacrificados com as minhas canções desesperadas.

Os colegas e amigos com os quais eu formei a maioria dos grupos de trabalho, esses também merecem a minha gratidão, são eles: Leandro Franco, Luciano Grangeiro (“O paqueto”) e Vinícius Costa de Souza (“O gordo careca”). Todos grandes amigos que eu conquistei durante esse período e com os quais eu aprendi muito.

Outra pessoa que contribuiu muito para esta conquista foi meu orientador, Luciano Paschoal, um professor de fato e extremamente dedicado.

A minha querida irmã, Tatiane Lemes, que além de torcer muito por mim soube entender que o computador não é só um equipamento para trocar *e-mails* com o namorado que está longe, mas também um recurso fundamental para desenvolver este trabalho.

Agradeço de forma especial a minha querida mãe, Tania Lemes, um exemplo de pessoa honesta, trabalhadora e com muita disposição para lutar e conquistar os seus objetivos, sejam quais forem as adversidades. Ela esteve comigo em todos os momentos, inclusive quando não estávamos próximos fisicamente.

Não posso deixar de agradecer a uma pessoa que amo muito e que junto comigo compartilhou todas as emoções provocadas por um trabalho de conclusão, a minha noiva Karine Neves. A *K* soube estar comigo e entender os meus momentos de ausência.

Finalmente, gostaria de dizer o meu muito obrigado e desejar que Deus abençoe a todos os colegas, amigos e familiares que estiveram torcendo por mim ao longo de toda a graduação.

1 Introdução

Com o uso em grande escala da Internet é verificado um conseqüente aumento nos tipos e na quantidade de ataques, de maneira que todos os sistemas estão sujeitos a inúmeras ameaças, sejam elas internas, externas, acidentais ou maliciosas [Campello, 2001]. Através da exploração dos diferentes tipos de vulnerabilidades como falhas de configuração, falhas de implementação e uso indevido de recursos disponíveis, surge um universo de ataques possíveis. Exemplos desse universo de ataques incluem desde varreduras de portas, negação de serviços, seqüestro de conexões até ataques mais sofisticados, tais como negação de serviço distribuído, inserção e evasão [Ptacek e Newsham, 1998], [Durst et al. 1999], [Mutaf, 1999]. Com o objetivo de minimizar as chances de um intruso obter sucesso em suas atividades, diversos mecanismos de proteção são utilizados. Entre esses mecanismos surgem a criptografia, a certificação digital, a infra-estrutura de chaves públicas, os *firewalls*, os protocolos de autenticação e, ainda, os sistemas de detecção de intrusão.

Sistemas de detecção de intrusão, *Intrusion Detection Systems* (IDSs), representam uma importante técnica de monitoramento, cuja principal função é detectar ações maliciosas, tais como tentativas de ataques e obtenção de informações [Amoroso, 1999], [Proctor, 2001]. Os IDSs são classificados, tradicionalmente, segundo o escopo de monitoramento utilizado: rede ou *host*. Os sistemas de detecção de intrusão baseados em rede capturam e analisam pacotes de rede, realizando a busca por ataques direcionados a determinados serviços e estações existentes nesse ambiente. Através desse tipo de IDS as informações que trafegam em um *backbone* e nos diversos segmentos de rede de uma organização podem ser monitoradas sem interferir no desempenho da rede ou das estações conectadas à mesma. Já os IDSs baseados em *host* monitoram as atividades do sistema através de dados coletados nas próprias estações, possibilitando detectar atividades não autorizadas que estejam sendo realizadas por usuários da estação.

Atualmente, após aproximadamente mais de uma década de pesquisas, existem disponíveis no mercado diversos sistemas de detecção de intrusão. Entre esses IDSs destacam-se o Snort [Roesch, 1999], o Bro [Paxson, 1999], o NFR (Network Flight Recorder) [Nfr, 2001], o Dragon [Enterasys, 2002] e o RealSecure [Iss, 1999], todos baseados em rede, embora alguns desses sistemas possuam também módulos para monitoramento de estações. Segundo [Puketza et al. 1997], [Lippmann et al. 1999] e [Barber, 2001], com a popularização dos IDSs surge a necessidade por ferramentas e metodologias para avaliação e teste dos mesmos, pois ao analisar o comportamento de um conjunto de IDSs é possível determinar, com maior exatidão, qual ferramenta melhor se adapta aos diferentes ambientes de rede existentes. Isso se faz necessário porque o volume e as características do tráfego de uma rede variam em função das diferentes instituições, sendo provável que uma determinada ferramenta de detecção altere a sua forma de atuação em função de tais características.

Diversas abordagens para avaliação de sistemas de detecção de intrusão têm surgido. As mais significativas são as propostas em: [Puketza et al. 1997], [Lippmann et al. 1998], [Lippmann et al. 1999], [Durst et al. 1999], [Alessandri, 2000] e [Barber, 2001]. Entretanto essas metodologias possuem diversas limitações, entre elas: não

possuem uma forma sistematizada para a execução dos procedimentos previstos, são constituídas de uma série de atividades exaustivas que são realizadas durante semanas, exigem que os usuários da metodologia possuam conhecimentos específicos tais como: a implementação dos IDSs (o que não é possível no caso de IDSs proprietários) ou, ainda, o desenvolvimento de *scripts* em determinadas linguagens. Além disso, essas metodologias são pouco documentadas e em alguns casos os resultados obtidos, através de determinados testes, são questionáveis devido à forma com que tais experimentos foram conduzidos.

Esta monografia apresenta uma abordagem alternativa para avaliação de IDSs que constitui-se em uma metodologia factível fora do ambiente acadêmico, que possui um conjunto de procedimentos sistematizados, realizáveis em um curto espaço de tempo e que não exige o conhecimento prévio das ferramentas de detecção a serem avaliadas. A metodologia proposta, ao contrário de outras abordagens, avalia as potencialidades dos IDSs e não a implementação ou a base de assinaturas dos mesmos. Os cenários de testes são compostos por uma quantidade pequena, mas significativa de ataques, que são facilmente reproduzíveis num ambiente de rede local. Os testes previstos nessa metodologia avaliam as seguintes características: capacidade de detecção, taxas de falsos positivos e escalabilidade dos sistemas avaliados.

O trabalho está organizado da seguinte forma: o capítulo 2 apresenta uma síntese das principais publicações relacionadas ao tema em questão. O capítulo 3 descreve a metodologia proposta. Os resultados obtidos na fase de testes são apresentados no capítulo 4. O capítulo 5 finaliza esse trabalho com algumas considerações finais e apresenta as perspectivas para trabalhos futuros.

2 Metodologias para avaliação de IDSs

Esse capítulo apresenta uma síntese das principais abordagens desenvolvidas até o momento para avaliação de sistemas de detecção de intrusão. Ao término dessa síntese é traçado um paralelo entre as principais características que compõem essas abordagens.

2.1 Puketza et al. 1997

Os primeiros trabalhos realizados com o objetivo de desenvolver metodologias para avaliação de sistemas de detecção de intrusão surgiram no final dos anos 90. Em 1997 foi publicada a primeira proposta de metodologia para avaliação de IDSs, [Puketza et al. 1997], desenvolvida na Universidade da Califórnia. Essa proposta consiste, primeiramente, na seleção de cenários de teste. Esses cenários são reproduzidos através de *scripts* que simulam tanto ataques quanto atividades consideradas normais. A melhor forma de escolher um cenário de teste, segundo o autor, é basear-se na política de segurança da empresa, pois é ela que define o que é e o que não é uma intrusão. A segunda fase dessa proposta corresponde à realização dos procedimentos de teste, que estão divididos nas seguintes avaliações: (a) identificação de intrusão, experimentos que verificam a habilidade do IDS de detectar duas formas distintas de ataques: concorrentes e seqüenciais (b) uso de recursos, que correspondem a testes que avaliam a quantidade de recursos computacionais (carga da CPU, memória principal e espaço em disco) consumidos pelo IDS e (c) testes de saturação, cujo objetivo é verificar o comportamento desses sistemas quando submetidos a situações que visam esgotar diferentes recursos (exemplo: aumento do número de processos concorrentes em execução na estação em que o sistema estiver instalado).

2.1.1 Ataques concorrentes

No intuito de evitar a detecção de um determinado ataque o intruso pode distribuir suas atividades de forma concorrente [Puketza et al. 1997]. A figura 1 ilustra o que a metodologia em questão considera um ataque concorrente¹.

Como se pode observar o ataque em questão é uma tentativa de quebra de senhas, no qual três intrusos atuam de forma a somar esforços para realizar o ataque. Essa figura lista as atividades que correspondem a cada um dos intrusos: o intruso identificado pela letra “A” deve copiar a ferramenta *cracker program* de uma estação remota e aguardar pelo término das demais atividades realizadas pelos outros intrusos. Quando isso ocorrer, deve excluir tal ferramenta; o intruso “B” tem como principal atividade compilar e executar a ferramenta de ataque. Já o intruso “C” deve verificar a saída gerada por essa ferramenta (*cracker program*) e tentar realizar uma conexão, com a senha que foi descoberta, ao servidor alvo. As setas na figura 1 indicam a ordem em que as atividades devem ser executadas.

Para cada um dos intrusos existentes em uma sessão de ataque, há um determinado *script* que simula suas atividades. A fim de permitir o desenvolvimento desses *scripts*, a plataforma desenvolvida possui (a) comandos básicos para simular conexões, realizar

¹ Na abordagem em questão ataques concorrentes correspondem a atividades intrusivas que são realizadas a partir de mais de uma estação.

autenticação, acessar servidores de transferência de arquivos e executar comandos do sistema operacional, (b) mecanismos de sincronização responsáveis por monitorar a seqüência em que atividades de uma sessão devem ser realizadas, (c) mecanismos de comunicação, capazes de controlar o recebimento e envio de dados entre os *scripts* de uma mesma sessão, e (d) recursos para armazenar e reproduzir *scripts*.

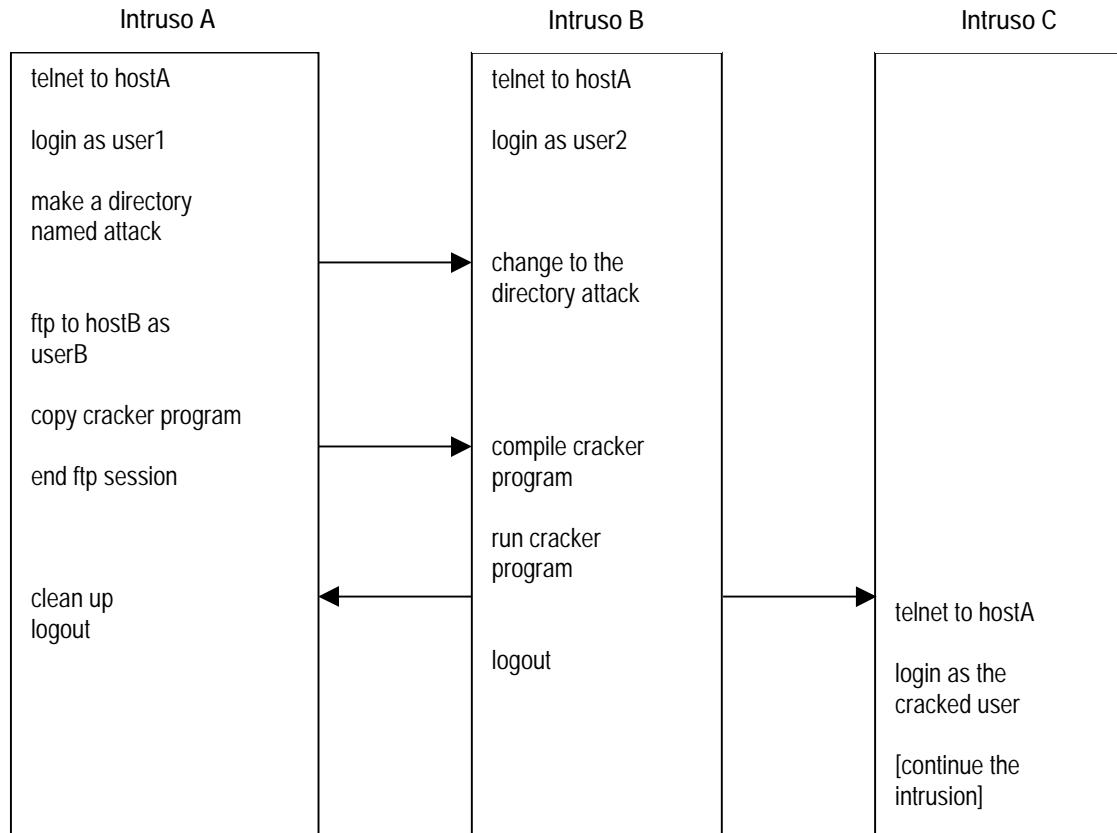


Figura 1 - Ataque concorrente [Puketza et al. 1997]

2.1.2 Avaliação

Nos testes realizados, um único IDS conhecido por *Network Security Monitor* (NSM), foi submetido aos experimentos previstos pela plataforma de software desenvolvida por [Puketza et al. 1997]. O NSM é um sistema de detecção de intrusão baseado em rede que, a exemplo de outras ferramentas, detecta ataques através da análise de assinaturas. No entanto, esse IDS possui uma característica incomum aos demais: atribui valores, entre 0 e 10, para cada conexão estabelecida na rede monitorada. Esses valores são atribuídos considerando a frequência e o tipo de conexão estabelecida. Para a realização dos experimentos propostos foram criados *scripts* (seqüenciais e concorrentes) para simular cada uma das seguintes ações: transmitir um arquivo de senhas para uma outra estação da rede, tentar descobrir senhas através de programas ou por tentativa e erro e, ainda, explorar vulnerabilidades em alguma aplicação resultando em acesso privilegiado. A premissa existente por trás dessa estratégia é que o NSM atribuiria um peso maior para uma sessão composta por muitas atividades intrusivas (seqüenciais) do que para diversas sessões compostas por poucas atividades intrusivas (concorrentes).

Ataques	Scripts seqüenciais	Scripts Concorrentes
Transmissão do arquivo de senhas	7,472	7,472
Descoberta de senhas	3,160	3,160
Tentativa de diversas conexões	8,722	7,785
Exploração de vulnerabilidades	7,472	4,972

Tabela 1 - Avaliação do NSM: valores retornados [Puketza et al. 1997]

Como se observa na tabela 1, acima, para cada ataque realizado o IDS atribuiu um determinado valor, sendo que quanto maior o valor, maior a probabilidade de que essa atividade corresponda a um ataque. Segundo [Puketza et al. 1997], os resultados apresentados na tabela 1 demonstram que um intruso, uma vez “distribuindo” suas atividades pode reduzir as chances de ser detectado.

2.2 Lippmann et al. 2000

Desde 1998 o Laboratório Lincoln do *Massachusetts Institute of Technology* (MIT) conduz uma série de estudos em busca de uma metodologia para avaliação comparativa de IDS. De acordo com [Lippmann et al. 2000], os principais objetivos dos testes realizados em [Lippmann et al. 1998] e [Lippmann et al. 1999] foram (a) identificar os ataques detectados e as taxas de falsos positivos geradas pelos IDSs testados, a fim de que os desenvolvedores de tais sistemas obtivessem subsídios suficientes para corrigir possíveis falhas existentes nos mesmos e (b) prover uma forma imparcial de testar o comportamento dos IDSs submetidos à avaliação.

2.2.1 Método para classificação dos ataques

A classificação dos ataques no processo de avaliação de sistemas de detecção de intrusão permite descrever aspectos relevantes quanto ao contexto dos mais diferentes ataques [McHugh, 2000], isto é, entender a seqüência de passos e ações executadas para a realização de atividades não autorizadas e que de alguma forma coloquem em risco a estabilidade da rede e de seus serviços. A partir do conhecimento do contexto de um ataque é possível a realização de uma série de medidas para minimizar a probabilidade de que tais ataques sejam bem sucedidos.

A taxonomia utilizada em [Lippmann et al. 1998] e [Lippmann et al. 1999] classifica os ataques em relação a três aspectos: (1) nível de privilégio atual do usuário, (2) métodos de transição e (3) tipos de ações executadas pelo usuário. Esses três aspectos serão descritos a seguir.

2.2.1.1 Níveis de privilégios

Níveis de privilégios representam os tipos de acesso que um determinado indivíduo possui a um sistema ou estação. Segundo [Kendall, 1999], [Das, 2000] e [Korba, 2000] o conjunto de níveis de privilégios relacionados abaixo correspondem aos mais comuns e relevantes para avaliação de sistemas de detecção de intrusão baseados em rede.

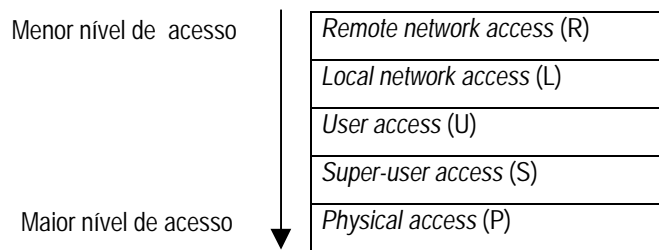


Figura 2 - Níveis de privilégios [Kendall, 1999]

- *Remote network access* (R), acesso remoto a rede;
- *Local network access* (L), acesso local a rede;
- *User access* (U), acesso de usuário, permite a execução de comandos de usuários no sistema;
- *Super-user access* (S), acesso de superusuário ou administrador, conjunto de privilégios reservados ao administrador do sistema;
- *Physical access* (P), acesso físico, corresponde ao acesso direto de um determinado usuário a servidores e/ou equipamentos de interconexão de rede;

2.2.1.2 Métodos de Transição

Os métodos de transição correspondem aos meios utilizados por um intruso para que o seu nível de privilégio sobre determinado sistema ou estação seja incrementado. A taxonomia em questão define um conjunto de cinco possíveis métodos, representados por um carácter, que podem vir a serem utilizados para a realização do processo de transição entre os diferentes níveis de privilégios existentes:

- *Masquerading* (m), método através do qual é possível fraudar um sistema fazendo-se passar por outro usuário ou *host*. Exemplos desse método incluem o uso indevido de contas de usuário legítimos do sistemas, que tenham sido roubadas, ou ainda o envio de pacotes TCP com endereço de origem forjado;
- *Abuse of feature* (a), consiste na realização de ações legítimas do usuário, mas que são executadas de forma a deixar o sistema, ou serviço num estado de falha ou de indisponibilidade. Exemplo: um usuário realiza diversas conexões a um determinado serviço até esgotar os recursos do mesmo, assim sendo, nenhum outro usuário conseguirá conectar-se a tal serviço até que essas conexões tenham sido excluídas, ou então, que o serviço tenha sido reiniciado;
- *Implementation bug* (b), a partir da exploração de falhas na implementação, tais como *buffer overflow* ou *formatstrings*, é possível que se obtenha acesso privilegiado ao sistema;
- *Social engineering* (s), consiste na habilidade de obter informações ou acesso indevido a determinado ambiente ou sistema com a utilização de técnicas de persuasão. Geralmente, ataques baseados em engenharia social são altamente eficazes e resultam na obtenção de informações que facilitarão o acesso a uma determinada organização;

- *System Misconfiguration* (c), método no qual o intruso explora as possíveis falhas de configuração existentes em determinados serviços e obtém acesso privilegiado ao sistema.

2.2.1.3 Tipos de ações

Existem muitas ações que podem ocorrer como parte de um ataque. Na taxonomia em questão as ações são identificadas por duas *strings*, sendo que a primeira representa a categoria da ação e a segunda determina o tipo específico da ação realizada. A tabela 2, abaixo, descreve os cinco tipos de ações contempladas nessa taxonomia:

Categoria	Tipo específico	Descrição
Probe	Probe (Machines)	Determina os tipos e o número de estações ativas na rede.
	Probe (Services)	Determina quais os serviços em execução e as respectivas portas.
	Probe (Users)	Fornecer informações sobre os usuários da rede.
Deny	Deny (Temporary)	Negação de serviço temporária.
	Deny (Administrative)	Negação de serviço que requer intervenção administrativa.
	Deny (Permanent)	Negação de serviço que requer intervenção .
Intercept	Intercept (Files)	Interceptação de arquivos.
	Intercept (Network)	Interceptação do tráfego da rede.
	Intercept (Keystrokes)	Interceptação do conjunto de teclas pressionadas por um usuário.
Alter	Alter (Data)	Alteração de dados armazenados.
	Alter(Intrusion-Traces)	Remoção dos sinais de uma intrusão, tais como os registros de um arquivos de log.
Use	Use (Recreational)	Uso do sistema para atividades tais como jogos e <i>Internet Relay Chat</i> , IRC.
	Use(Intrusion-Related)	Uso do sistema como um ponto de entrada para futuros ataques.

Tabela 2 - Tipos de ações possíveis [Korba, 2000]

Por exemplo, se um usuário com acesso remoto a rede (R), explorar uma falha no servidor de FTP (b) que resulte temporariamente em acesso negado a esse serviço (Deny), o ataque em questão será classificado como "R-b-Deny(Temporary)".

2.2.2 Tipos de tráfego

Por tráfego de fundo entende-se todo o fluxo de dados transmitido paralelo aos ataques a serem reproduzidos. O objetivo do tráfego de fundo é fazer com que, a partir da reprodução de um fluxo de dados que represente diferentes situações de um ambiente de rede, o processo de teste seja o mais real possível. Os tipos de tráfego de fundo normalmente reproduzidos a fim de realizar avaliação de IDSs são:

- Tráfego sintético: o fluxo de dados é gerado através de aplicações instaladas em diversas estações com o objetivo de reproduzir as ações diárias (previamente coletadas) de um conjunto de usuários;
- Tráfego fragmentado, nesse tipo de tráfego de fundo somente datagramas IP fragmentados são reproduzidos, com o objetivo de avaliar a capacidade do IDS de reconstruir esses datagramas.

Ainda que o processo de geração do tráfego de fundo esteja descrito muito superficialmente nas publicações referentes ao trabalho em questão, esse é um item que pode influenciar diretamente nos resultados obtidos [McHugh, 2000]. A primeira etapa da avaliação realizada em 1998 foi coletar amostras referentes ao tráfego existente nas bases da força aérea americana para, a partir dessa caracterização, criar um conjunto de dados sintéticos que representasse diversas atividades realizadas pelos usuários, tais como: navegar na Internet, ler e enviar *emails*, transferir arquivos via ftp, editar arquivos, compilar códigos e estabelecer conexões com outras estações. A partir de alterações no kernel dos sistemas operacionais das estações responsáveis pela geração do tráfego de fundo, foi possível simular, a partir de 10 estações, atividades geradas por dezenas de usuários. Na etapa seguinte, coletar tráfego de ataque, foram lançadas 300 instâncias de 58 diferentes tipos de ataques contra as estações *Linux* e *Solaris* da rede alvo para que o tráfego de ataque gerado fosse armazenado para posterior reprodução. Já na avaliação realizada em 1999 foram incluídos a esse tráfego ataques contra servidores *Windows NT*, ataques fragmentados e, ainda, ataques denominados *Stealthy attacks*, cujo objetivo é confundir o IDS em relação ao que é tráfego normal e o que de fato é um ataque real.

2.2.3 Ambiente de teste

Embora existam algumas divergências, quanto às configurações, entre o diagrama de rede disponível em [Kendall, 1999] e as informações existentes no *site* do laboratório Lincoln, a seção 3 do artigo [Lippmann et al. 2000] descreve o ambiente reproduzido na figura abaixo como sendo o ambiente de teste atualmente utilizado [McHugh, 2000].

000000

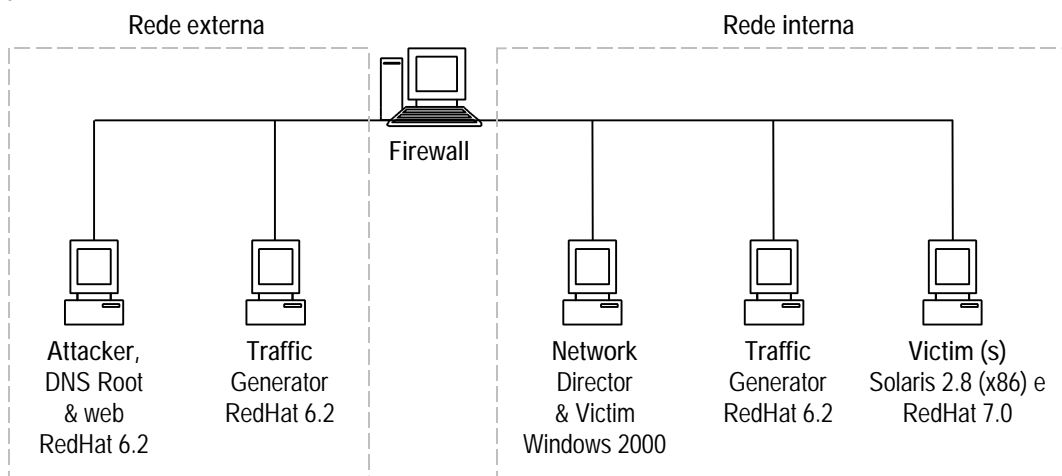


Figura 3 - Ambiente de teste usado na avaliação descrita em [Lippmann et al. 2000]

O lado esquerdo da figura 3 representa o ambiente externo (Internet) enquanto o lado direito representa o ambiente interno (Intranet) protegida contra diversos ataques por um firewall. A estação denominada *Attacker, Root DNS & web*, possui as funções de servidor de DNS (Domain Name Service), servidor *web* com a cópia de milhares de

sites e, ainda, promove os ataques contra a rede interna. As estações *Traffic Generator*, executando *Red Hat Linux 6.2*, são responsáveis pela geração do tráfego de fundo (tanto interno quanto externo) e possuem uma implementação especial do *kernel Linux* que permite, a partir de uma estação, simular o tráfego de dezenas de estações. As estações *Solaris 2.8 (x86)*, *Red Hat 7.0* e *Windows 2000* representam o conjunto de plataformas submetidas a diversos ataques e, portanto são denominadas *Victims*. Já a estação *Network Director* possui uma aplicação em *Java* denominada *Lincoln Adaptable Real-time Information Assurance Testbed* (LARIAT) que, embora não esteja disponível para utilização, permite a configuração (ataques, tipo de tráfego, taxa de reprodução, etc.) dos testes a serem executados [Lippmann et al. 2000].

2.3 Alessandri, 2000

Um dos trabalhos mais recentes desenvolvidos pela IBM [Alessandri, 2000] propõe uma abordagem de avaliação, cujo objetivo é testar as capacidades existentes nos IDSs e não as suas implementações ou base de assinaturas. Através dessa forma de avaliação é possível averiguar a capacidade de detecção dos IDSs, frente a um ataque para o qual o IDS, ainda, não tenha uma assinatura desenvolvida

2.3.1 Metodologia

Esse modelo de avaliação consiste em uma técnica, implementada em *prolog*, que permite descrever em forma de regras as características existentes nos IDSs a serem avaliados e as características exploradas em diferentes ataques. Segundo essa proposta é possível identificar, a partir do cruzamento das regras que representam tais características, o comportamento dos IDSs sem que seja necessária a realização de experimentos com os sistemas avaliados.

A descrição das características dos IDSs é realizada através de duas formas de classificação. A primeira forma separa as propriedades dos IDSs de acordo com o nível de detalhe das características descritas. Para tal, são considerados os seguintes grupos: (a) grupo de propriedades genéricas obtidas através da taxonomia proposta por [Hervé, 1999] e (b) grupo de propriedades detalhadas, cujo objetivo é descrever características relacionadas a protocolos e aplicações. A segunda forma de classificação separa as propriedades dos IDSs em função de características tais como: posicionamento dos sensores, técnicas utilizadas para reconhecimento de padrões e o tempo entre a ocorrência de uma atividade e a geração do respectivo alarme. Um exemplo simples de descrição de características de um IDS é apresentado abaixo. A partir dessa descrição é possível afirmar que este é um IDS baseado em rede (`p.infoSrc.type.net = true`) e que possui mecanismos capazes de analisar o protocolo IP (`p.IP.aware = true`). No entanto, o IDS não é capaz de tratar campos do cabeçalho TCP (`p.TCP.aware = false`).

```
p.infoSrc.type.net = true
p.IP.aware = true
p.TCP.aware = false
```

A descrição de uma atividade é representada por propriedades e regras que descrevem as características requeridas para que o IDS gere um alarme para essa atividade. A regra a seguir descreve uma atividade intrusiva denominada *A.SMTP.pipe*, que corresponde a exploração de uma vulnerabilidade a partir da qual é possível a execução de comandos arbitrários em versões antigas do *Sendmail* (servidor de email para sistemas operacionais padrão Unix).

```
A.SMTP.pipe -> r.alarm.SMTP.pipe = p.infoSrc.type.net & r.tech.patRec & r.proto.SMTP.aware
```

Essa regra deve ser interpretada da seguinte forma: a atividade denominada (A.SMTP.pipe) pode fazer com que um sistemas de detecção de intrusão gere um alarme (r.alarm.SMTP.pipe) se todas as premissas descritas forem verdadeiras, ou seja, a fonte de coleta de informações do IDS for a rede (p.infoSrc.type.net), esse sistema possuir algum tipo de algoritmo de reconhecimento de padrões para análise das assinaturas (r.tech.patRec) e, ainda, possuir a capacidade de tratar o protocolo SMTP (r.proto.SMTP.aware).

2.3.2 Avaliação do IDS

Uma vez que todas as características dos IDSs e das atividades tenham sido identificadas e devidamente representadas, a aplicação, desenvolvida em *prolog*, esta apta a executar a avaliação do IDS. A figura 4, abaixo, ilustra a interação entre os seguintes módulos: descrição do IDS, descrição da atividade, saída do IDS e avaliação do resultado. Essa interação entre módulos ocorre através dos avaliadores de regras e de resultado.

O “avaliador de regras” é responsável pelo cruzamento e análise das descrições das características do IDS e das atividades, resultando em uma lista de alarmes gerados pelo IDS que é armazenada no módulo de saída. Já o “avaliador de resultados” verifica a saída produzida e compara com o alarme que de fato deveria ser gerado (falso positivo, falso negativo, detecção correta) para essa atividade e que esta representado em forma de regras no módulo descrição da atividade. Portanto, através dessa segunda comparação é possível afirmar se o resultado que o IDS gerou está correto.

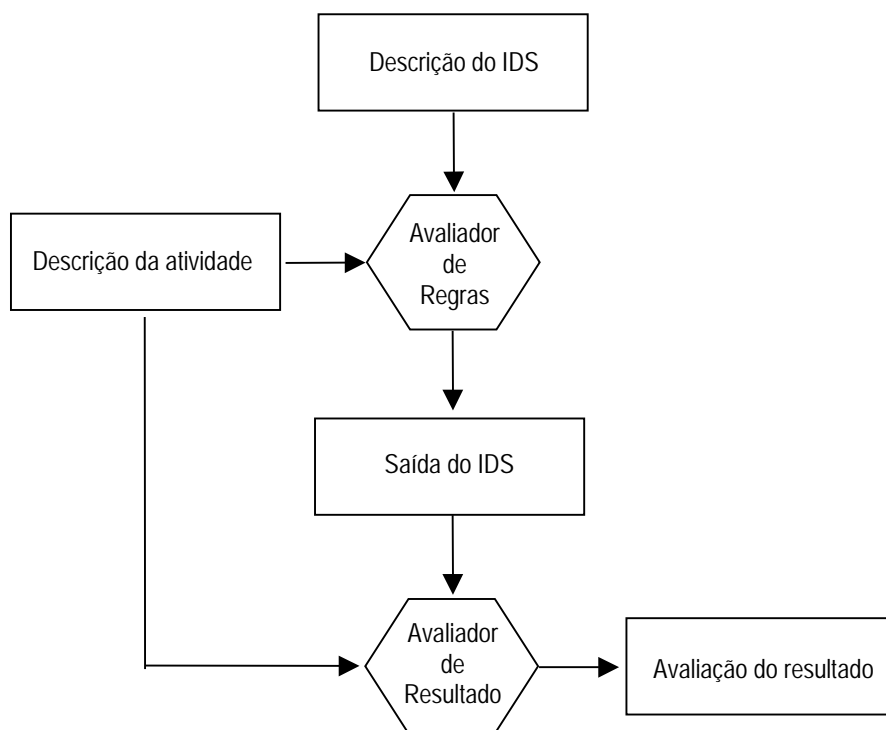


Figura 4 - Avaliação do IDS [Alessandri, 2000]

2.4 Barber, 2001

Conforme [Barber, 2001] o processo de escolha de um produto como um IDS requer a avaliação dos seguintes aspectos:

- Tecnológico: refere-se à forma com que a tecnologia utilizada no produto é implementada, ou seja, analisa aspectos técnicos do produto;
- Corporativo: o objetivo é verificar a saúde financeira da organização que desenvolveu o produto em questão;
- Comercial: a avaliação deste aspecto procura identificar as estratégias que a empresa responsável pelo produto adota no que se refere ao mercado e à concorrência.

A fim de realizar a avaliação dos aspectos corporativos e comerciais foram desenvolvidos alguns questionários a serem aplicados em entrevistas com os representantes dos produtos avaliados. A avaliação dos aspectos tecnológicos tem por objetivo identificar possíveis falhas em um IDS considerando dois grupos de critérios, um *quantitativo* e outro *qualitativo*. No grupo dos critérios quantitativos são analisadas (a) as taxas de detecção de ataques, (b) a habilidade de reconhecer ataques fragmentados e (c) o desempenho (com e sem fragmentação). Já o grupo de critérios qualitativos apresenta as seguintes análises: (a) facilidade de instalação e gerenciamento e (b) flexibilidade para definir os eventos que serão armazenados.

Para realizar os testes que consideram os critérios quantitativos foi construído um cenário com oito estações e um roteador, sendo três estações para realização de ataques (na primeira estação havia sido instalado Solaris 2.6, na segunda Windows NT e na terceira SuSe Linux 6.3), além de outras três estações com os serviços a serem atacados (nessas estações haviam sido instalados os mesmos sistemas operacionais das estações anteriores), e mais uma estação para gerenciamento do IDS, além de uma última estação para instalação dos sensores dos IDSs.

Para a realização dos ataques previstos foram utilizadas vinte e três ferramentas de ataque para promover basicamente dois tipos de ataques: negação de serviço e varredura de portas. O tráfego de fundo reproduzido nos testes de escalabilidade foi gerado a partir de ferramenta denominada *fragrouter* que permite monitorar a quantidade de pacotes IP fragmentados.

Segundo [Barber, 2001], a partir do estudo realizado é possível afirmar que os IDSs disponíveis não atendem satisfatoriamente às necessidades de mercado sendo necessário, ainda, o desenvolvimento de muitas outras potencialidades para que sejam obtidas melhorias nas taxas de detecção e escalabilidade dos IDSs. Quanto ao futuro dos sistemas de detecção de intrusão, Barber destaca dois aspectos: (a) essas ferramentas serão dotadas de mecanismos não apenas para detectar ataques de negação de serviço, mas também para responder de forma eficiente a esses ataques e (b) os IDSs devem ter o escopo de monitoramento ampliado para aplicações, ou seja, ao invés de monitorar a rede ou uma determinada estação, os IDSs deveriam monitorar e analisar especificamente o tráfego endereçado às aplicações, por exemplo um banco de dados.

2.5 Síntese das propostas de metodologias existentes

A tabela 3 apresenta uma síntese das principais características das metodologias anteriormente mencionadas. As características consideradas nessa análise foram: o tipo de avaliação realizada, a forma de classificação dos ataques, o tipo do tráfego de fundo gerado para realização dos experimentos, o tipo de ambiente de teste necessário e, por fim, as métricas de avaliação disponível em cada uma das metodologias.

Quanto ao tipo de avaliação, constatou-se que a maior parte das metodologias avalia apenas as bases de assinaturas dos IDSs ([Puketza et al. 1997] , [Lippmann et al. 2000] e [Barber, 2001]) o que, além de exaustivo considerando-se o tamanho destas bases, gera um resultado válido por um pequeno período de tempo, pois assinaturas são desenvolvidas muito rapidamente pelos fabricantes dos IDSs ou até mesmo pelos usuários destas ferramentas. Portanto, para que os resultados destas metodologias possam ser considerados confiáveis é necessário que sejam refeitos os experimentos previstos cada vez que surgem novas assinaturas. Por outro lado, metodologias tais como a proposta nessa monografia e em [Alessandri, 2000] que ao invés de testarem a base de assinaturas testam as capacidades de detecção dos IDSs , devem ter seus experimentos refeitos somente quando novos recursos forem implementados à essas ferramentas.

No que tange à classificação de ataques, a metodologia proposta nessa monografia utiliza a descrição técnica. A descrição técnica do ataque possibilita o conhecimento do mesmo a ponto de permitir não só o desenvolvimento de assinaturas, mas também uma seleção de ataques em função das características comuns exploradas pelos mesmos, o que resulta na redução dos cenários de teste, uma vez que os ataques que possuem as mesmas características são rapidamente identificados. Já a descrição do contexto dos ataques, possibilita o entendimento não da técnica em si, mas da forma como o ataque ocorre e os mecanismos utilizados para a realização do mesmo. Portanto, essa forma de classificação representa uma importante ferramenta de documentação de ataques [McHugh, 2000].

A caracterização do tráfego de fundo é uma característica fundamental na avaliação de IDSs, pois interfere diretamente nos resultados de alguns testes, tais como verificação das taxas de falsos positivos e escalabilidade. Algumas propostas como as publicadas em [Lippmann, 1998] e [Lippmann, 1999] não descrevem a composição do tráfego de fundo. Isto faz com que os resultados, principalmente da avaliação de falsos positivos, sejam contestados, pois não há como afirmar se de fato existem ou não ataques inseridos nesse tráfego, nem como identificar os motivos que levaram os IDSs a gerar tais resultados. Portanto, os resultados desses testes possivelmente representem valores incorretos. Já a metodologia proposta nessa monografia utiliza um tráfego de fundo homogêneo, isto é, um fluxo de dados composto por determinado tipo de pacote cujo tamanho e a taxa de reprodução são variáveis.

Com exceção da metodologia proposta em [Alessandri, 2000], todas as demais necessitam de algum tipo de ambiente de teste para a realização dos experimentos previstos na metodologia. Propostas como [Puketza et al. 1997] e [Lippmann et al. 2000] requerem ambientes de teste complexos, com dezenas de estações (atacantes, vítimas, sistemas avaliados, geradores e coletores de tráfego), diferentes equipamentos de interconectividade (*hub*, *switch* e roteadores) e até mesmo *firewalls*. Essas características de um ambiente de teste, muitas vezes, inviabilizam a reprodução dos mesmos, pois além de complexos demandam um grande intervalo de tempo e um ambiente dedicado até o término dos testes. Além disso, o uso de *firewalls* faz com que diversos ataques não sejam capturados pelos IDSs, pois são bloqueados antes de chegarem à rede interna da empresa. A utilização de *firewalls* é de extrema importância para qualquer instituição e deve estar presente em todos os estudos cujo objetivo seja avaliar a infra-estrutura de segurança. No entanto, para o propósito em questão é um componente que limita o processo de avaliação.

	Puketza	Lippmann	Alessandri	Barber	Abordagem proposta
Tipo de avaliação					
Avaliação exaustiva	X	X		X	
Potencialidade de detecção			X		X
Classificação de ataques					
Descrição técnica do ataque			X		X
Descrição do contexto do ataque		X			
Tráfego de fundo					
Tráfego de fundo sintético	X	X			
Tráfego fragmentado				X	
Tráfego homogêneo					X
Ambientes de teste					
Independente			X		
Específico	X	X		X	X
Resultados apresentados					
Taxa de falsos positivos		X	X	X	X
Tipos de ataques detectados	X	X	X	X	X
Escalabilidade				X	X
Aspectos de usabilidade				X	
Capacidade de detecção de ataques concorrentes	X				

Tabela 3 - Características das principais metodologias para avaliação de IDSs

A última característica considerada nessa síntese refere-se aos resultados apresentados. Embora a proposta descrita em [Barber, 2001] considere métricas que não são avaliadas em outras abordagens, essa proposta consiste em um conjunto de atividades exaustivas, uma vez que para cada critério avaliado é necessário executar novamente um a um dos ataques de forma manual, pois não há uma ferramenta de apoio ao processo de teste. Outro aspecto negativo dessa proposta é que os procedimentos não são sistematizados e a documentação existente não é clara o suficiente quanto à forma como os testes foram realizados. Abordagens como as desenvolvidas em [Lippmann et al. 2000] também são criticadas quanto ao esforço empregado nas atividades previstas, pois os mesmos resultados obtidos podem ser alcançados a partir de experimentos menos exaustivos [McHugh, 2000]. Propostas que prevêm o uso de ferramentas como mecanismos de auxílio no processo de avaliação, devem garantir que tais mecanismos estejam em perfeito funcionamento e que sejam facilmente utilizados para avaliação de diferentes IDSs mediante diferentes ataques. No entanto, não é o que se percebe em [Puketza et al. 1997], cuja ferramenta foi concebida e testada à luz de características existentes em um único IDS. Embora essa proposta consista no uso de uma plataforma de software baseada em *scripts*, que geram tanto o tráfego de fundo quanto os ataques, a quantidade de *scripts* desenvolvidos é bastante limitada. Assim, é função dos usuários dessa metodologia criar *scripts*, em linguagem *C* ou *TCL*, para simulação do tráfego de fundo característico da sua empresa, bem como para simular novos ataques. Outra abordagem que descreve o uso de uma ferramenta como principal mecanismo na realização dos experimentos e na geração do tráfego é [Lippmann et al. 2000]. No entanto passados aproximadamente dois anos a ferramenta ainda não possui uma versão estável para utilização o que inviabiliza a reprodução de tal abordagem. Na proposta desenvolvida pela IBM [Alessandri, 2000], a ferramenta de avaliação dispensa a necessidade de reproduzir um ambiente de teste, no entanto para que seja possível

aplicar essa metodologia de forma a obter resultados que de fato reflitam a realidade dos IDSs avaliados, serão necessários conhecimentos tão específicos de como os IDSs tratam determinadas características dos protocolos que a aplicação dessa proposta, na prática, limitada a IDSs de código aberto. Outro fator referente às ferramentas que diminuem a probabilidade de utilização das mesmas é a necessidade de conhecimentos específicos em determinadas linguagens como *prolog* e *TCL*.

De forma geral o que se observa nas metodologias citadas é a ausência de uma proposta, cuja aplicação seja voltada a empresas. Para que isto seja possível é necessário o desenvolvimento de uma abordagem com procedimentos bem definidos, facilmente realizáveis e que, de fato, reflitam a realidade dos critérios avaliados. As metodologias supracitadas falham em ambos os aspectos. Além de não fornecerem documentação adequada sobre a realização de alguns testes importantes, algumas destas abordagens ainda não foram devidamente validadas ou não apresentam a instrumentalização necessária para serem aplicadas.

3 A metodologia de avaliação proposta

O principal objetivo dessa metodologia é prover uma abordagem prática de fácil e rápida utilização para avaliação de IDSs. Para aplicar essa abordagem não é necessário o conhecimento dos sistemas de detecção de intrusão, pois esses são considerados caixas pretas. Os sistemas de detecção de intrusão submetidos a essa metodologia são avaliados quanto às seguintes características: (a) capacidade de detecção, (b) escalabilidade e (c) taxa de falsos positivos gerados. A metodologia proposta é composta por cinco etapas: seleção dos ataques, seleção de ferramentas, geração do tráfego dos cenários de avaliação, montagem do ambiente de avaliação e análise dos IDSs. A seguir serão descritas cada uma das etapas citadas.

3.1 Seleção dos Ataques

Nessa etapa o objetivo é selecionar um conjunto de ataques que explore características técnicas únicas entre si. Ao invés de simplesmente reunir um conjunto de ataques o que se busca, ao finalizar esta etapa, é selecionar ataques cuja detecção seja possível a partir diferentes mecanismos de detecção existentes em um IDS. Por exemplo, para que um IDS seja capaz de detectar um ataque de inserção, o *URL Encoding*, ele necessita mais do que simplesmente a capacidade de análise de um pacote HTTP, pois é necessário, ainda, um mecanismo de decodificação do conteúdo do pacote. Já o processo de detecção de um ataque de negação de serviço, como o *teardrop*, requer capacidades como remontar pacotes IP fragmentados. Dessa forma, os ataques selecionados nessa etapa, representam um conjunto de características ímpares que permitem avaliar as diferentes capacidades de detecção dos IDSs e não simplesmente a base de assinaturas dessas ferramentas. Essa forma de classificação denomina-se descrição técnica do ataque. Para que essa forma de seleção de ataques seja colocada em prática é necessário definir um conjunto inicial de ataques. Essa atividade está descrita na seção a seguir.

3.1.1 Ataques propostos

A primeira atividade prevista para essa etapa de seleção de ataques é definir quais os tipos de ataques que serão utilizados na avaliação. Nessa monografia foram considerados os seguintes tipos de ataques:

- Evasão: através de um ataque de evasão é possível obter desde o tipo e a versão de servidor *web* utilizados na estação alvo até a executar *scripts* que possam colocar em risco a segurança de tal servidor. Independente do tipo de ação executada o objetivo ao realizar este tipo de ataque é primeiramente evitar a detecção do mesmo, fazendo com que no processo de análise do conteúdo de um pacote os IDSs percam informações vitais para a detecção. Já as aplicações alvo recebem todas as informações contidas nos pacotes enviados e, portanto, serão vítimas do ataque.

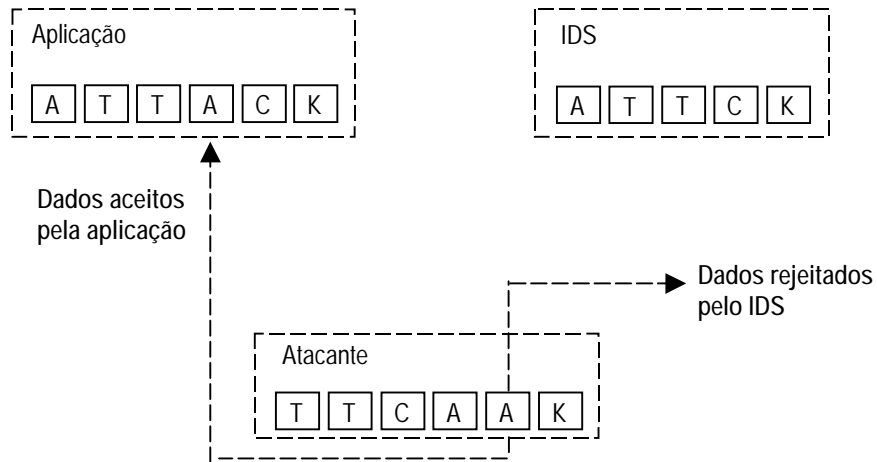


Figura 5 - Exemplo de evasão da letra “A” [Ptacek e Newsham, 1998]

A figura 5 ilustra a evasão da letra “A”, ou seja, o atacante envia para aplicação as seguintes informações (TTCAAK). A aplicação recebe os dados enviados e os organiza de forma que o resultado seja (ATTACK). No entanto o IDS no momento organizar as informações perde um dado importante, a letra “A”. Por conseguinte o IDS entende que a informação enviada (ATTCK) não corresponde a nenhum ataque existente na sua base de assinaturas, logo não gera nenhum alerta e o ataque é realizado [Ptacek e Newsham, 1998].

- Inserção: a exemplo do ataque de evasão, as atividades realizadas podem variar desde a sondagem de servidores *web* até a tentativa de execução de *scripts* que coloquem em risco a segurança do servidor em questão. Novamente, são exploradas as inconsistências existentes entre as informações coletadas pelo IDS e as informações que de fato chegam à aplicação. No entanto, diferentemente dos ataques de evasão, os ataques de inserção fazem com que o IDS ao analisar um pacote receba mais informações do que o sistema alvo. Dessa forma, o ataque é realizado e o IDS não identifica na sua base de assinaturas um padrão que corresponda ao ataque [Ptacek e Newsham, 1998].

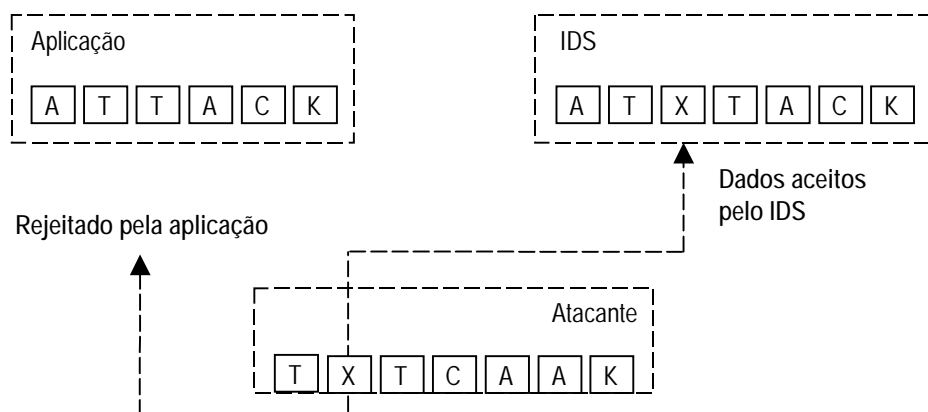


Figura 6 - Exemplo de inserção da letra “X” [Ptacek e Newsham, 1998]

A figura 6 ilustra a inserção da letra “X”, ou seja, o atacante envia à aplicação o seguinte conteúdo (TXTCAAK), o IDS, por sua vez, aceita (TXTCAAK)

sofrendo então a inserção da letra “X” o que logicamente impede a detecção do ataque. Já a aplicação, ao receber os pacotes, organiza as informações e descarta a letra “X”, por ser um dado irrelevante para aplicação, obtendo então (ATTACK).

- Varredura de portas ou *port scan*: normalmente essa é uma das primeiras atividades realizadas por um intruso e consiste basicamente em uma coleta de dados, cujo objetivo é reunir o maior número possível de informações sobre a rede ou o servidor alvo. Essas informações podem variar desde o tipo de sistema operacional instalado em uma determinada estação até quais serviços estão em execução e as suas respectivas versões. Conforme o relatório publicado no dia 1º de outubro de 2001 pelo instituto SANS (*System Administration, Networking and Security*) juntamente com o NIPC/FBI (*National Infrastructure Protection Center, FBI*), as varreduras de portas estão entre as vinte vulnerabilidades mais críticas de segurança na Internet [Cert, 2002].
- Negação de serviço ou *Deny of Services (DoS)*: ataque cujo objetivo é esgotar os recursos de um serviço ou rede tornando-o inacessível ou com respostas muito lentas. Os ataques de negação de serviços são normalmente executados usando ferramentas que enviam de forma indiscriminada requisições a um determinado servidor, sobrecarregando os recursos do mesmo e, por vezes, tornando o sistema inoperável. Na grande maioria desses ataques o endereço de origem é forjado (*spoofing*) e, portanto, dificultam o processo de auditoria. Todos os sistemas conectados à Internet e que estejam executando serviços de rede baseados no protocolo TCP estão sujeitos a ataques de negação de serviços [Newsham e Ptacek, 1998]. Esse tipo de ataque possui uma variante denominada negação de serviço distribuída, *Distributed Denial of Service (DDoS)*, que corresponde a ataques de DoS realizados em larga escala, partindo de várias estações e disparados simultaneamente de forma coordenada sobre um ou mais alvos.

Uma vez definidos os tipos de ataques a serem utilizados na avaliação é o momento de selecionar um conjunto de atividades intrusivas que irá representar cada um desses tipos de ataques. A seguir serão descritos os ataques que compõe os cenários de ataque propostos nessa monografia. É importante ressaltar que caso esses ataques sejam de conhecimento, o leitor pode passar a etapa seguinte, a descrição técnica do ataque.

3.1.1.1 Evasão

Case Sensitivity

Os sistemas operacionais tais como o Windows 98 e Windows 2000 Server não diferem letras maiúsculas de letras minúsculas (não são *case sensitivity*), ou seja, o arquivo *phf.cgi* pode ser referenciado tanto como *PHF.cgi* quanto como *PHF.CGI*. Logo os IDSs devem ser capazes de detectar ambas as requisições, caso contrário o ataque será bem sucedido.

Method Matching

No intuito de explorar as vulnerabilidades de um *script* e, ainda, tentar inviabilizar a detecção de tal ataque, pode-se utilizar métodos alternativos de solicitações tais como *Put*, *Head* e *Post*. Dessa forma, embora alguns IDSs identifiquem a requisição `"Get /cgi-`

bin/phf.cgi HTTP/1.0" podem vir a não identificar uma requisição do tipo "Put /cgi-bin/phf.cgi HTTP/1.0".

Session Splicing

Diferentemente dos ataques de fragmentação, este ataque consiste no envio de diversos pacotes. Por exemplo, a solicitação "Get /cgi-bin/phf.cgi HTTP/1.0" pode ser dividida em múltiplos pacotes "Ge", "t", "/", "cgi", "-bin", "p", "hf.c", "gi", "H", "T", "TP", "/1", ".0". Sendo assim o processo de detecção é ainda mais difícil e para que seja possível os IDSs devem ser capazes de analisar uma seqüência de pacotes.

HTTP Mis-formatting

Conforme a RFC 2616 a estrutura de uma requisição a um servidor *web* deve seguir o seguinte formato: método <espaço> URL <espaço> versão CRFL CRFL; onde CRFL corresponde a uma linha em branco obrigatória. No entanto muitos servidores *web* aceitam requisições que não estejam plenamente em conformidade com essas especificações, por exemplo: método <tabulação> URL <tabulação> versão CRFL CRFL. Portanto existe a possibilidade de iludir alguns IDSs, pois ao ser realizada a comparação entre o pacote e a base de assinaturas de ataques não haverá relação, logo o pacote será considerado uma solicitação normal e não um ataque.

DOS Directory Syntax

Em plataformas Windows o separador de diretórios é representado por '\', enquanto que a especificação do protocolo HTTP determina que o separador de diretórios web seja '/'. Isso faz com que toda vez que uma requisição do tipo "Get /cgi-bin/phf.cgi" é enviada a um servidor web da Microsoft, esse tenha que converter '/' para '\', de forma que a aplicação em questão interprete a requisição como "Get \cgi-bin\phf.cgi". Portanto, este servidor aceita requisições como, "\cgi-bin\phf.cgi", o que faz com que o IDS, ao analisar o pacote HTTP, não encontra a assinatura de um ataque conhecido.

3.1.1.2 Inserção

Long URLs

Existem várias técnicas que visam melhorar o desempenho dos IDSs. Uma dessas técnicas limita a quantidade de informações de uma requisição HTTP a serem analisadas. Dessa forma, é possível a inserção de uma quantidade suficiente de caracteres para mover o código de ataque para além do escopo da análise do IDS, o que faz com que o ataque não seja detectado.

Self Reference

Os caracteres ".." quando utilizados para acessar diretórios conduzem o usuário para um diretório superior (diretório pai) ao diretório atual. Já o caracter "." faz referência ao diretório atual. Sendo assim, então "c:\temp\.\.\.\." é equivalente a "c:\temp\". O objetivo da técnica denominada *Self Reference* é confundir os mecanismos de análise de assinaturas dos IDSs enviando o seguinte tipo de requisição "Get ./cgi-bin/./phf".

URL Encoding

Conforme a RFC 2616, caracteres binários arbitrários podem ser passados em uma requisição HTTP desde que estejam na seguinte notação: %xx, onde xx corresponde ao valor hexadecimal do caracter. Uma vez que a requisição "Get /cgi-bin/teste.cgi HTTP/1.0" seja

codificada torna-se "Get /cg%69-b%69n/t%65st%65.cg%69 HTTP/1.0". Portanto, os IDSs, antes de analisarem, qualquer *string* devem decodificar a mesma.

Multiple Slashes

Os servidores *web* aceitam requisições contendo múltiplas barras, *slashes*, como em "Get /cgi-bin/scripts///phf.cgi HTTP/1.0". Contudo, existe a possibilidade que alguns IDSs ao analisarem esses tipos de requisições falhem devido ao fato de que a assinatura existente contenha apenas uma barra.

Parameter Hiding

Uma requisição de página *web* pode conter informações adicionais, parâmetros, que serão utilizadas para construir o conteúdo de páginas dinâmicas. Esses parâmetros são determinados após um sinal de interrogação no identificador uniforme de recursos, *Uniform Resource Locator* (URL), como em "Get /index.htm?user=normal HTTP/1.0". Alguns IDSs, a fim de otimizar o processo de análise de pacotes ignoram todas as informações após a indicação de parâmetros; sendo assim, há a possibilidade da inserção de um código malicioso após essa parte da requisição.

Reverse Traversal

Consiste em uma tentativa de iludir os IDSs através de uma requisição na qual existam referências a outros diretórios que não estejam especificados na base de assinaturas dos IDSs. Por exemplo, a requisição "Get /cgi-bin/phf.cgi HTTP/1.0", é facilmente detectada. Alguns IDSs, porém, podem desconsiderar esta mesma requisição quando requerida da seguinte forma: "Get /cgi-bin/scripts/./phf.cgi HTTP/1.0".

3.1.1.3 Varredura de portas e serviços

TCP connect

A chamada de sistema *connect* (), provida pelo sistema operacional, é usada para estabelecer conexão com um conjunto de portas na máquina alvo. Caso a porta esteja no estado *listening*, *connect* irá estabelecer uma conexão; caso contrário o usuário receberá a mensagem de porta inalcançável.

SYN Scan

Técnica conhecida como "*half-open scanning*" por não estabelecer uma conexão TCP completa. O primeiro pacote a ser enviado está com o *flag* SYN configurado para estabelecer uma conexão real e, portanto, deve aguardar uma resposta da estação para a qual o pacote foi enviado. Ao receber uma resposta com os *flags* SYN/ACK ligados isto indica que a porta está no estado *listening*. Já uma resposta com o *flag* RST ligado é uma indicação que a porta está fechada. Se o *flag* SYN/ACK é recebido, o *flag* RST é imediatamente enviado para encerrar a conexão.

ACK Scan

Este tipo de sondagem envia pacotes com o *flag* ACK para uma porta específica. Caso seja retornado um pacote TCP com *flag* RST ligado, a porta é classificada como "não filtrada"; caso seja retornado um ICMP *unreachable*, a porta é classificada como "filtrada".

Window Scan

Este tipo de *scan* é muito similar ao *ACK scan*, no entanto é possível detectar portas abertas mesmo quando essas estão sendo filtradas por um *firewall*. Isso ocorre

devido ao tamanho da janela TCP existentes em diversos sistemas operacionais (por exemplo: FreeBSD, SunOS e OpenVMS).

FIN Scan

Esta técnica consiste em enviar um pacote com o *flag* FIN habilitado para uma determinada porta. Segundo a RFC 793 as portas que estiverem fechadas devem responder com um pacote TCP com o *flag* RST ligado, enquanto que as portas que estiverem abertas devem ignorar o pacote em questão.

UDP Scan

Este método é usado para determinar quais as portas UDP (*User Datagram Protocol*) estão abertas. A técnica implica em enviar 0 bytes de dados de pacotes UDP para cada porta da estação alvo. Caso a resposta seja uma mensagem ICMP *port unreachable* então a porta está fechada.

Null Scan

Esta técnica consiste em enviar um pacote TCP com todos os *flags* desabilitados para uma determinada porta na estação alvo, sendo que as portas que estiverem fechadas devem responder com um pacote TCP com o *flag* RST habilitado, enquanto que as demais devem ignorar o pacote em questão.

Xmas

Ao contrário do método denominado *null scan* esta técnica envia um pacote para cada porta da estação alvo a ser sondada com todos os *flags* habilitados exceto o *flag* SYN.

TCP Ping

Através desta técnica é possível determinar quais as estações que estão ativas no momento. Para tal, ao invés de enviar pacotes ICMP *echo request* e aguardar pelas respostas são enviados pacotes com *flag* ACK habilitado por toda a rede. Estações que estiverem ligadas devem responder com um pacote TCP com o *flag* RST habilitado.

TCP fragmentation scanning

Esta forma de sondagem utiliza várias outras técnicas de varredura de portas tais como *SYN scan*, *FIN scan*, *Xmas* e *Null scan*. Os pacotes enviados a estação alvo são fragmentados, ou seja, o cabeçalho TCP é dividido em vários pacotes. Com isso, os IDSs que não possuem mecanismos eficientes de remontagem de pacotes não conseguem identificar essa forma de ataque, pois os diversos datagramas enviados individualmente não correspondem a uma ameaça..

Sondagem do protocolo IP

Este método determina quais protocolos da família TCP/IP estão sendo utilizados na estação alvo. A técnica consiste em enviar pacotes IP *raw* sem nenhum cabeçalho para cada porta na estação alvo; caso a resposta seja ICMP *unreachable*, o protocolo não está sendo utilizado.

Identificação Remota de Sistemas Operacionais (fingerprinting)

A fim de identificar o sistema operacional instalado em uma determinada estação, se utiliza um conjunto de técnicas que detectam características da implementação do protocolo TCP/IP do sistema operacional que está instalado na estação alvo. Uma vez que essas características tenham sido identificadas é realizada uma comparação dessas

informações com a base de dados da ferramenta de ataque, a fim de definir qual o sistema operacional da estação em questão. Atualmente as técnicas mais utilizadas para determinar o tipo de sistema operacional são: sondagem FIN (*FIN probe*), identificação de padrões do número inicial de seqüência (*Initial Sequence Number - ISN*) escolhido pelo TCP ao responder um pedido de conexão, verificação da existência ou não do bit de não fragmentação, análise do tamanho da janela entregue pelos pacotes de retorno (*TCP Initial Window*), valor do *flag* ACK, tamanho da mensagem ICMP de erro, verificação do valor do tipo de serviço retornado pelas mensagens de ICMP *port unreachable* e, ainda, análise da forma como é feito o controle de fragmentação e das informações existentes no campo de opções do cabeçalho TCP.

IDENT Reverso TCP

O protocolo IDENT (RFC 1413) retorna nomes de usuários válidos e é consultado por diversos serviços (IRC, FTP, SMTP, etc.), além de servir como mecanismo de restrição de acesso baseado na relação usuário e endereço IP. Porém, conforme notificado por Dave Goldsmith em 1996, o rastreamento do protocolo IDENT permite revelar o nome dos usuários donos dos processos conectados via TCP [Ref].

3.1.1.4 Negação de Serviços

Smurf

Esse ataque se utiliza de redes que permitam tráfego na interface de *broadcast*. O ataque consiste na falsificação do endereço de origem de um pacote ICMP *echo request*, fazendo com que uma grande quantidade de respostas, pacotes ICMP *echo reply*, sejam direcionadas ao endereço que foi falsificado.

UDP Storm

A exemplo do ataque anterior o objetivo é congestionar a rede e, por conseguinte, diminuir a largura de banda da mesma. Ao se estabelecer uma conexão entre dois serviços UDP, por exemplo, *echo/UDP* e *chargen/UDP*, serão gerados uma grande quantidade de pacotes na rede até que ocorra uma intervenção externa, como, por exemplo, reiniciar o serviço *inetd*.

SYN Flood

Este ataque explora as limitações do processo inicial de uma conexão denominado *handshake*, procedimento que, através do envio de pacotes TCP com os *flags* SYN e ACK habilitados entre cliente e servidor, permite efetuar o início de uma conexão a um serviço. O objetivo é exceder os limites definidos para o número de conexões que podem ser estabelecidas à um determinado serviço. Isso faz com que não seja possível estabelecer quaisquer outras conexões a esse serviço até que o número de conexões em espera seja reduzido. O problema mais crítico que envolve esse tipo de ataque e os IDSs é a alta probabilidade de falsos positivos, uma vez que nem todas as tentativas de conexões em um pequeno intervalo de tempo, podem ser consideradas tentativas de *SYN Flood*.

Teardrop

Ao contrário do ataque denominado *Smurf*, que utiliza a “força bruta” para gerar o ataque, o *teardrop* executa um ataque de DoS utilizando-se de falhas em diferentes implementações da pilha TCP/IP. Este ataque explora a incapacidade de alguns sistemas operacionais de reconstituir pacotes IP fragmentados. Como resultado, os sistemas

suscetíveis a esse ataque têm o seu funcionamento prejudicado, podendo inclusive travar o sistema operacional.

ICMP Fragmentation

Para ser transmitido entre redes locais, um pacote IP deve ser fragmentado toda vez que exceder o limite do maior quadro que uma determinada rede local é capaz de transmitir (*Maximum Transfer Unit* - MTU). Nesse caso, é necessário dividir o pacote IP em fragmentos menores que a MTU. O protocolo ICMP é um protocolo auxiliar ao IP, que carrega informações de controle e diagnóstico, informando falhas como TTL do pacote IP excedido, erros de fragmentação e roteadores congestionados. O ataque em questão consiste no envio de um pacote ICMP mal formado para uma determinada estação, fazendo com que a estação de destino ao receber este pacote reduza a MTU desnecessariamente. Isto faz com que a conexão entre essas duas estações fique extremamente lenta. Além disso, em função da quantidade de pacotes enviados uma razoável quantidade da largura de banda é consumida.

3.1.2 Descrição técnica do ataque

Para realizar a descrição técnica do ataque proposta por essa metodologia é de fundamental importância o conhecimento das características técnicas do ataque. Essas características foram obtidas através do estudo dos ataques descritos na seção anterior. A partir desse estudo identificou-se as seguintes características: tipo e quantidade de pacotes enviados, campos do cabeçalho desse pacote que são importantes para que o ataque seja realizado, a quantidade e o tamanho dos pacotes a serem gerados. Sendo assim, uma vez que os ataques propostos tenham sido estudados e as suas características forem conhecidas é o momento de classificar os ataques em função das suas características técnicas. Isto deve reduzir o cenário inicial de ataques a uma quantidade que represente apenas ataques que explorem características diferentes uns dos outros.

A tabela 4 abaixo está organizada da seguinte forma: nas linhas foram representados todos os ataques propostos no primeiro cenário de ataques, conforme descrito na seção 3.1.1. Já nas colunas aparecem às características exploradas em cada ataque. Estas características estão agrupadas em função dos tipos de pacotes utilizados (HTTP, IP, TCP, ICMP e UDP) e de aspectos relevantes para a realização do ataque tais como: a quantidade de pacotes utilizados e o estabelecimento ou não de conexões com a estação a ser atacada. Assim que todos os ataques e todas as características forem colocadas nessa tabela é o momento de relacionar os ataques com as suas respectivas características. Como resultado tem-se a visão detalhada de cada ataque, o que é fundamental na etapa seguinte para seleção de quais ataques devem compor o cenário de avaliação.

				HTTP			IP			TCP							ICMP		UDP						
	Múltiplos pacotes	Não estabelece conexão(<i>Half-Open</i>)	Estabelece conexão	Linha de Requisição	Codificação da requisição	Tamanho da requisição	Off-set de fragmento	Pacote IP com DF bit habilitado	Identificação do fragmento	Pacote IP raw	FLAGS de controle da fragmentação	Opções (NOP, MSS, Window, Timestamp)	Campo: Tipo de serviço	Número de seqüência	TCP Initial Window	Pacote com <i>flag</i> SYN	Pacote com <i>flag</i> FIN	Pacote com <i>flag</i> ACK	Pacote com <i>flag</i> URG	Pacote com <i>flag</i> PUSH	Pacote com todos os <i>flags</i> desativados	Pacote ICMP	Tamanho da mensagem ICMP de Erro	Pacote UDP de zero bytes	
Evasão																									
Case Sensitivity				X																					
Method Matching				X																					
Session Splicing	X			X																					
HTTP Mis-formatting				X																					
Dos Directory Syntax				X																					
Inserção																									
Long URLs				X	X																				
Self Reference				X																					
URL Encoding				X	X																				
Multiple Slashes				X																					
Parameter Hiding				X																					
Reverse Traversal				X																					
Varredura de Portas																									
TCP Connect	X		X																						
Syn Scan	X	X													X										
Ack Scan	X	X																		X					
Window Scan	X	X																		X					
Fin Scan	X	X																		X					
UDP Scan	X	X																							X
Null Scan	X	X																							X
Xmas	X	X																		X	X	X	X		
TCP Ping	X	X																		X					
TCP Fragmentation	X	X					X	X	X						X	X	X	X	X						
Varredura do protocolo IP	X	X								X															
Fingerpriting	X	X						X				X	X	X	X		X	X					X		
Ident Reverso TCP	X		X																X						
Negação de Serviço																									
Smurf	X																						X		
UDP Storm	X																								X
Syn Flood	X	X														X									
Teardrop	X						X	X	X																
ICMP Fragmentation	X						X	X	X														X		

Tabela 4 - Descrição técnica do cenário inicial de ataques proposto

Uma vez que a descrição técnica, representada na tabela 4, foi finalizada o próximo passo era realizar a simplificação dessa tabela. A etapa de simplificação corresponde a uma visão macro das características de cada um dos ataques. Para tal, foram identificadas as características genéricas dos ataques. Essas características estão indicadas nas colunas da tabela 5. Estando a atividade de simplificação concluída, iniciou-se a seleção dos ataques que irão compor o cenário de avaliação. A seguir serão apresentados os motivos que conduziram à seleção dos ataques em destaque na tabela 5.

Evasão

O ataque *Session Splicing* é o único entre os ataques de evasão que utiliza múltiplos pacotes. Todos os demais utilizam somente um pacote HTTP. Entre os ataques que apresentam as mesmas características, *Method Mathing* foi arbitrariamente selecionado.

Inserção

Os ataques selecionados, *Long URLs* e *URL Encoding*, apresentam como características exclusivas em relação aos demais ataques, respectivamente, uma requisição HTTP com uma grande quantidade de caracteres e a presença de um padrão de codificação da requisição. Entre os ataques *Self Reference*, *Multiple Slashes* e *Reverse Traversal* que exploram as mesmas características foi escolhido arbitrariamente o *Self Reference*.

Varredura de portas

Os ataques *UDP scan*, varredura do protocolo IP e *TCP Fragmentation* foram selecionados pois geram, respectivamente, pacotes UDP, IP *raw* e IP fragmentados. Entre os ataques que estabelecem conexão (*TCP connect* e *Ident Reverso TCP*) ambos utilizam pacotes TCP, no entanto *Ident Reverso TCP* foi o escolhido para representar este tipo de técnica de varredura de portas. O último ataque selecionado *Fingerprinting* utiliza três tipos diferentes de pacotes em uma mesma seção de ataque (IP, TCP e ICMP). Já *Xmas* foi o ataque selecionado para representar todos os ataques de varredura de portas que utilizam somente pacotes TCP e que não estabelecem conexões.

Negação de serviço

Os ataques *Smurf*, *UDP Storm*, *Syn Flood* e *Teardrop* foram selecionados, pois diferem um dos outros quanto aos tipos de pacotes utilizados, ou seja, ICMP, UDP, TCP e IP, respectivamente. Já o ataque identificado como *ICMP Fragmentation* foi incluso nessa seleção, devido ao fato de que entre esses ataques é o único que utiliza pacotes do tipo ICMP fragmentados.

					HTTP		IP	TCP	ICMP	UDP
		Múltiplos pacotes	Não estabelece conexão(Half-Open)	Estabelece conexão		Codificação da requisição	Tamanho da requisição			
Seleção	Evasão HTTP									
	Case Sensitivity			X	X					
b	Method Matching			X	X					
b	Session Splicing	X		X	X					
	HTTP Misformatting			X	X					
	Dos Directory Syntax			X	X					
	Inserção HTTP									
b	Long URLs			X	X		X			
b	Self Reference			X	X					
b	URL Encoding			X	X	X				
	Multiple Slashes			X	X					
	Parameter Hiding			X	X					
	Reverse Traversal			X	X					
	Varredura de Portas									
	TCP Connect	X		X				X		
	Syn Scan	X	X					X		
	Ack Scan	X	X					X		
	Window Scan	X	X					X		
	Fin Scan	X	X					X		
b	UDP Scan	X	X							X
	Null Scan	X	X					X		
b	Xmas	X	X					X		
	TCP Ping	X	X					X		
b	TCP Fragmentation	X	X				X	X		
b	Varredura do protocolo IP	X	X				X			
b	Fingerprinting	X	X				X	X	X	
b	Ident Reverso TCP	X		X				X		
	Negação de Serviço									

b	Smurf	X	X						X	
b	UDP Storm	X	X							X
b	Syn Flood	X	X					X		
b	Teardrop	X	X				X			
b	ICMP Fragmentation	X	X				X		X	

Tabela 5 - Seleção do cenário de avaliação

3.2 Seleção de ferramentas

Essa etapa foi dedicada à obtenção de ferramentas que permitissem reproduzir o cenário de avaliação proposto na etapa anterior. Essa atividade pôde ser realizada num curto espaço de tempo devido à facilidade com que atualmente se pode encontrar e utilizar tais ferramentas.

A figura abaixo está baseada no relatório anual do *General Account Office* (GAO), órgão ligado ao congresso norte-americano, ilustra a relação entre a sofisticação das ferramentas de ataques atuais e o conhecimento técnico necessário para utilizá-las [Durst et al. 1999].

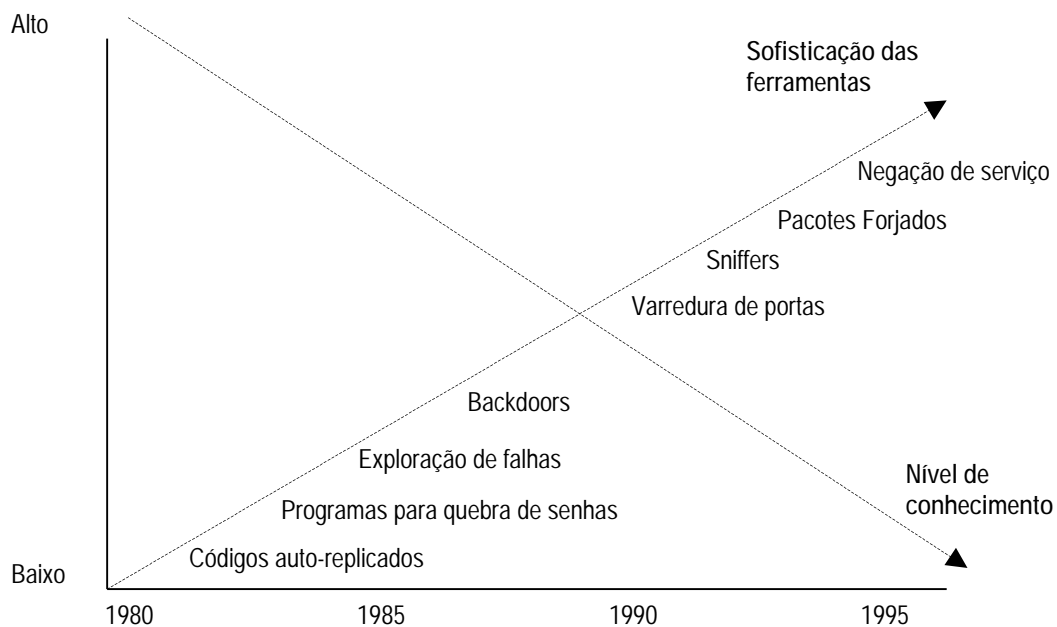


Figura 7 - Evolução das ferramentas de ataque

Como se pode observar, a relação entre esses dois aspectos é inversamente proporcional, ou seja, quanto mais sofisticadas as ferramentas de ataques, menor o conhecimento necessário para operá-las e, dessa forma, promover ataques. Embora diversas técnicas intrusivas sejam conhecidas há muitos anos, somente a partir da metade da década de noventa as ferramentas que implementam tais técnicas tornaram-se amplamente disponíveis por meio da Internet [Durst et al. 1999].

Ao longo dessa segunda etapa as principais fontes utilizadas para busca de ferramentas e aprendizado sobre as mesmas foram: (a) *sites* que disponibilizam informações sobre segurança de sistemas operacionais [Linuxsecurity, 2002], (b) artigos

publicados em [Cert, 2002], (c) portais de segurança da informação [Securityfocus, 2002] e finalmente (d) os *sites* dos desenvolvedores de algumas dessas ferramentas [Nmap, 2002] e [Whisker, 2000].

3.3 Ferramentas de ataque

A tabela 6 exibe o nome dos ataques a serem reproduzidos (conforme determinado na seção 3.2.2), as ferramentas utilizadas para tal e os pré-requisitos básicos para instalação e configuração dessas ferramentas.

Ataques	Ferramentas	Pré-requisitos
Evasão		
Method Mathing	Whisker 1.4	GNU/Linux, Perl 5.0 e Libwhisker
Session Splicing		
Inserção		
Long URLs	Whisker 1.4	GNU/Linux, Perl 5.0 e Libwhisker
Self Reference		
URL Encoding		
Varredura de portas		
TCP Fragmentation	Nmap 2.54	GNU/Linux
UDP Scan		
TCP Connect		
Varredura do protocolo IP		
Fingerprinting		
Ident Reverso TCP		
Negação de serviços		
Smurf	Smurf 4.0	GNU/Linux e gcc 2.9.6
UDP Storm	Udpstorm 1.0	GNU/Linux e gcc 2.9.6
SYN Flood	Synk4	GNU/Linux e gcc 2.9.6
Teardrop	Teardrop 2.0	GNU/Linux e gcc 2.9.6
ICMP Fragmentation	Jolt2	GNU/Linux e gcc 2.9.6

Tabela 6 - Ferramentas para reproduzir os ataques selecionados

A seguir descreve-se cada uma das ferramentas utilizadas para promover os ataques propostos na etapa anterior.

Whisker 1.4

É um conhecido *scanner* de vulnerabilidades, cujo objetivo é procurar por falhas em servidores *web*, através da execução de diversos *scripts* desenvolvidos em *perl*. Esta ferramenta implementa técnicas, anti-IDSs, que dificultam a detecção dos ataques por parte dos sistemas de detecção de intrusão. Essas técnicas são denominadas, respectivamente, evasão e inserção e foram descritas na seção 3.2. Através dessa ferramenta é possível obter, entre outras informações, o tipo de servidor *web* utilizado, a versão do mesmo e as vulnerabilidades às quais esse servidor está exposto. Além disso, a ferramenta possui uma arquitetura modular que permite a instalação de novos *plugins*, o que viabiliza a detecção de novas vulnerabilidades assim que as mesmas são descobertas.

Nmap 2.54

É uma ferramenta de varredura de portas, desenvolvida para diversas plataformas Unix, capaz de realizar (a) sondagem convencional de porta, para descobrir quais portas estão abertas em uma estação, e (b) sondagem furtiva, também conhecida como *stealth*

scan, que ao mesmo tempo em que procura identificar portas abertas tenta evitar a detecção por parte dos IDSs.

Smurf 4.0

A ferramenta *Smurf* permite realizar o ataque de negação de serviço que possui o mesmo nome da aplicação. Através dessa ferramenta é possível definir os endereços IP de origem (IP forjado), de destino e o endereço de *broadcast* da rede.

UDPstorm

É uma ferramenta através da qual o ataque *UDP Storm* é rapidamente realizado. Para tal, o usuário dessa ferramenta deve informar os endereços IP de origem (IP forjado) e de destino. Dessa forma será gerada, em um pequeno intervalo de tempo, uma grande quantidade de pacotes UDP entre as duas estações.

Teardrop 2.0

Esta é uma pequena aplicação que possibilita a realização do ataque *Teardrop*. Nessa versão a ferramenta suporta endereços IP forjados para definir a estação de origem dos pacotes, além do endereço IP da estação destino.

Synk4

A ferramenta em questão reproduz o ataque *Syn Flood*. O usuário define, a exemplo das ferramentas anteriores, os endereços IP de origem (IP forjado) e de destino, além do intervalo de portas para as quais os pacotes devem ser enviados.

Jolt2

Essa ferramenta reproduz o ataque de negação de serviço *ICMP fragmentation*. Igualmente, como nas demais ferramentas, são passados como parâmetros os endereços IP de origem (IP forjado) e de destino, além da quantidade de pacotes ICMP a serem gerados.

3.4 Geração do tráfego do cenário de avaliação

O cenário de avaliação é formado pelos ataques selecionados na seção 3.2.2 e pelo tráfego de fundo necessário para a análise de escalabilidade. A seguir serão descritas as formas propostas nessa metodologia para armazenar o tráfego de ataque e gerar o tráfego de fundo.

3.4.1 Coleta do tráfego de ataque

Os testes previstos na etapa de análises dos IDSs são realizados a partir da reprodução do tráfego de ataque, fazendo com que não seja necessária a utilização de cada uma das ferramentas de ataque cada vez que os testes tiverem que ser reproduzidos.

A primeira atividade prevista nessa etapa corresponde à montagem do ambiente de rede, conforme a figura 7. Esse ambiente é composto por apenas três estações, todas com sistema operacional Linux, distribuição RedHat 7.3. Na estação *Ataque* estão instaladas todas as ferramentas de ataque selecionadas na etapa de seleção de ferramentas. Já na estação *Vítima* estão instalados todos os serviços a serem atacados, por exemplo, o servidor *web Apache* para ataques de evasão e inserção. A estação *sniffer* possui como atribuição coletar o tráfego gerado pelas ferramentas de ataque. Para tal, o *sniffer tcpdump* [Tcpdump, 2002] precisa ser instalado nessa estação.

0

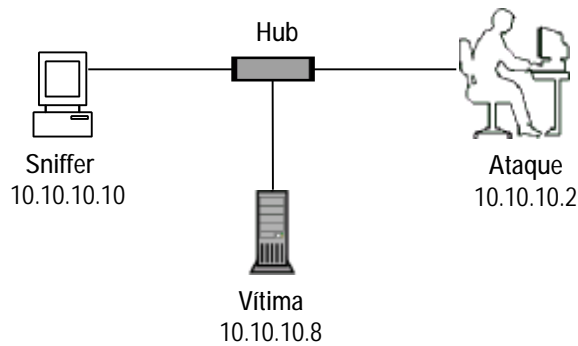


Figura 8 - Ambiente de rede para geração do tráfego dos cenários de teste

O *tcpdump* é uma ferramenta Unix utilizada para coletar dados de uma interface de rede. Todo o tráfego coletado é armazenado para, posteriormente, ser reproduzido. A fim de realizar essa coleta de dados a linha de comando a seguir deve ser utilizada.

```
#:tcpdump -w nomearquivo
```

O parâmetro *w* indica que os registros estão gravados em formato binário e *nomearquivo* refere-se ao nome do arquivo gravado. Para ler esse arquivo é necessário executar a seguinte linha de comando:

```
#:tcpdump -r nomearquivo
```

```
09:30:45:920000 10.10.10.2.1173 > 10.10.10.8.21: S 72797701: 72797701(0) win 512
```

A saída gerada pelo *tcpdump* segue o seguinte formato: dois dígitos para hora, dois dígitos para minuto, dois dígitos para segundos e seis dígitos para a parte fracionária de um segundo. Em seguida são exibidos os endereços IP e as portas de origem e destino, separados pelo caracter “>” que indica o sentido do fluxo de dados. O *flag* TCP indicado pela letra “S” (SYN) representa uma requisição de conexão e os números (72797701:72797701(0)) representam, respectivamente, o número de seqüência TCP inicial; e o número de seqüência TCP final, o valor zero entre parênteses corresponde ao número de bytes enviados para uma requisição de estabelecimento de conexão. O último dado fornecido (win 512) é o tamanho do *buffer* TCP da estação destino.

Uma vez que todas as estações foram configuradas iniciou-se as atividades de coleta do tráfego de ataque. O fluxograma abaixo, ilustrado na figura 8, representa a seqüência em que essas atividades foram executadas para cada um dos ataques selecionados.

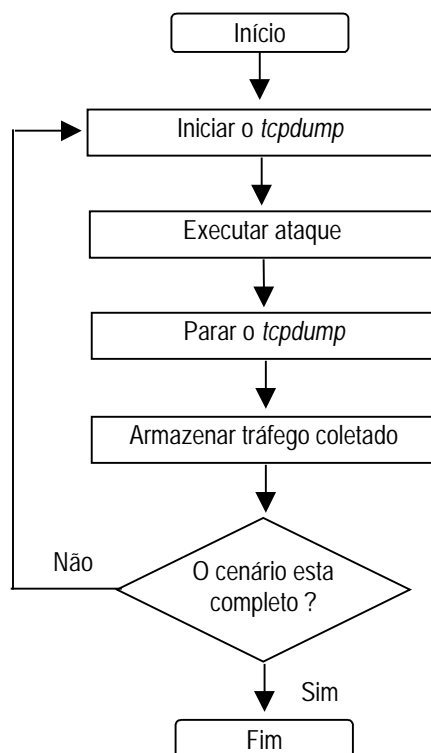


Figura 9 - Seqüência de atividades a serem realizadas para coletar o tráfego de ataque

Antes de reproduzir um determinado ataque o *tcpdump* deve ser iniciado, conforme descrito anteriormente; só então o ataque é executado. Assim que o ataque estiver concluído o *tcpdump* é parado e o arquivo gerado por esse *sniffer* é armazenado para que posteriormente possa ser reproduzido. Essa seqüência de atividades é realizada até que o último ataque tenha sido reproduzido.

3.4.2 Geração do tráfego de fundo

Conforme já mencionado, o tráfego de fundo é necessário para realizar a análise de escalabilidade dos IDSs. Inicialmente, foi considerada a hipótese de compor o tráfego de fundo por pacotes UDP de tamanhos (*payload*) variados (256, 512 e 1024 bytes). No entanto, ao realizar os testes de escalabilidade, verificou-se que a variação no tamanho dos pacotes praticamente não influenciou nos resultados obtidos pelos IDSs. Sendo assim, optou-se por utilizar um tráfego de fundo composto somente por pacotes UDP de 256 bytes. Esse tráfego é reproduzido em diferentes taxas de transmissão. A primeira dessas taxas corresponde a 4 (Mbps), taxa na qual os IDS geralmente ainda não descartam pacotes. Sugere-se que as demais taxas sejam reproduzidas em uma seqüência linear, por exemplo: 6, 8, 10 e 12 Mbps. Esta metodologia não determina uma taxa limite para reprodução desse tráfego. No entanto, o gerador de pacotes UDP utilizado nos experimentos tem capacidade para reproduzir até 100 Mbps.

A ferramenta utilizada para geração do tráfego de fundo é denominada *gerador_udp* e foi desenvolvida pelo Laboratório de Modelagem/Análise e Desenvolvimento de Sistemas de Computação (LAND) da Universidade Federal do Rio de Janeiro (UFRJ). Através dessa ferramenta é possível determinar (a) o endereço IP da estação destino, (b) o tamanho do pacote a ser enviado, (c) a taxa de transmissão em Kbps e (d) o tempo de geração do tráfego. Caso este tempo não seja determinado o tráfego é gerado até que seja cancelada a execução do comando. A seguir, é exibida a

linha de comando que permite reproduzir pacotes UDP de 256 bytes a uma taxa de 4 Mbps.

```
./gerador_udp 10.10.10.8 256 8000
```

```
Duração da sessão: 42 segundos  
Total de PDUs transmitidas: 90560  
Erros de escrita: 0  
Tamanho das PDUs: 256 bytes  
Intervalo entre gerações: 0,010240 segundos  
Banda teórica: 8000 Kbps  
Banda efetivamente gerada: 4098 Kbps
```

A saída gerada por essa aplicação informa que o tráfego de fundo foi gerado por 42 segundos. Durante esse intervalo de tempo foram transmitidos a cada 0,010240 segundos exatamente 90560 pacotes UDP de 256 bytes a uma taxa efetiva de 4098 Kbps o que corresponde a aproximadamente 4Mbps.

3.5 Montagem do ambiente de avaliação

Esse ambiente foi composto pelos IDSs a serem avaliados, pelas estações-alvo (*vítimas*) que sofreram os ataques, além de uma estação para reproduzir o tráfego de ataque (*Trf_ataque*) e outra para reproduzir o tráfego de fundo (*Trf_fundo*) ao longo dos testes de escalabilidade.

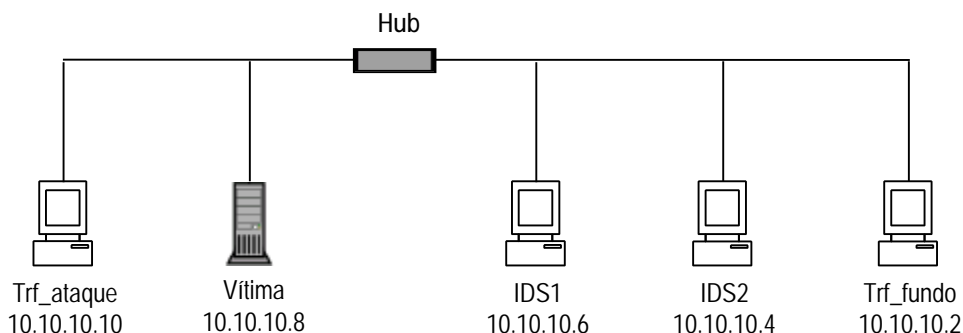


Figura 10 - Ambiente de rede para o cenário de avaliação

A figura acima representa o ambiente utilizado nessa avaliação, cujos resultados serão apresentados no capítulo 4. Para a avaliação proposta nessa monografia esse ambiente é adequado. No entanto, a quantidade de estações vítimas e o sistema operacional instalado nessas estações podem variar conforme os ataques selecionados para compor o cenário de avaliação. Por exemplo, caso o cenário de avaliação seja constituído por ataques a estações *Solaris* e *Windows 2000 Server*, o ambiente de rede representado na figura acima deverá contar com mais duas máquinas vítimas, nas quais esses sistemas devem estar instalados e devidamente configurados. Além disso, a quantidade de IDSs avaliados também pode variar, por conseguinte o número de estações para esses sistemas deverá ser maior do que ilustrado na figura 9. As estações responsáveis pela geração dos tráfegos de ataque e de fundo devem conter, respectivamente, a ferramenta *tcpreplay* [Tcpreplay, 2002] e o gerador de pacotes UDP.

O *tcpreplay* é uma ferramenta que possibilita a reprodução dos pacotes capturados, via *tcpdump*. A linha de comando abaixo demonstra como proceder para reproduzir um determinado tráfego na mesma taxa em que foi capturado.

`$.tcpdump -i interface_de_rede arquivo_tcpdump`

3.6 Análises dos IDSs

A metodologia aqui proposta avalia as seguintes características dos IDSs: capacidade de detecção, escalabilidade e taxa de falsos positivos gerados por esses sistemas. Capacidade de detecção é o teste a partir do qual é possível identificar as potencialidades de detecção dos IDSs, ou seja, quais os tipos de ataques que esse sistema está apto a detectar. O teste de escalabilidade permite identificar a partir de qual taxa os IDSs começam a descartar pacotes. Já a taxa de falsos positivos identifica a tendência desses sistemas em gerar alarmes falsos, isto é, confundir um tráfego considerado normal com um ataque ou, ainda, quando submetido a um ataque gerar alarmes referentes a outros ataques [Durst, et al. 1999].

3.6.1 Capacidade de detecção

O fluxograma apresentado na figura 10 ilustra a seqüência de atividades que deve ser realizada para analisar a capacidade de detecção dos IDSs avaliados. A primeira atividade a ser realizada é certificar-se que os arquivos de *log* estejam vazios. Tão logo essa verificação tenha sido concluída, o serviço de *log* dos IDSs deve ser iniciado. Em seguida, o primeiro ataque deve ser reproduzido a partir da estação *Trf_ataque* via *tcpdump* conforme já exemplificado.

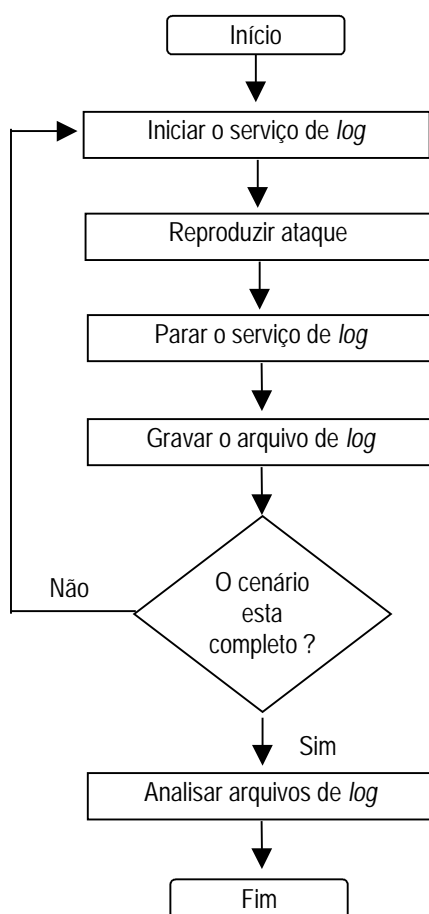


Figura 11 - Seqüência de atividades a serem realizadas para analisar capacidade de detecção dos IDSs

Assim que o ataque estiver finalizado o serviço de *log* de cada um dos IDSs deve ser parado e os arquivos gerados, armazenados para posterior análise. Caso o cenário de

avaliação não tenha sido completamente reproduzido, essa seqüência de atividades (figura 10) deve ser repetida até que todos os ataques tenham sido reproduzidos. O passo seguinte corresponde à análise dos arquivos de *log* dos IDSs. A tabela 7 representa a forma como os resultados obtidos nesse teste devem ser expressos. Esta tabela, para fins de exemplo, está considerando resultados obtidos para dois IDSs fictícios (IDS1 e IDS2) e um cenário de avaliação apenas com ataques de negação de serviço.

Ataques	IDS1	IDS2
Smurf	X	-
UDP Storm	X	-
Syn Flood	X	X
Teardrop	X	X
ICMP Fragmentation	-	-

Tabela 7 - Exemplo de resultados da análise de capacidade de detecção

A capacidade de detecção é representada nessa metodologia através de dois tipos de indicadores, sendo que “-” significa que o ataque não foi detectado e “X”, que o ataque foi detectado.

3.6.2 Escalabilidade

Para que os resultados da análise de escalabilidade sejam os mais confiáveis possíveis, é fundamental que sejam reproduzidos apenas os ataques que cada um dos IDSs detectou na análise anterior. A figura 11 representa a relação entre o tráfego de ataque e o tráfego de fundo que é reproduzido nessa análise.

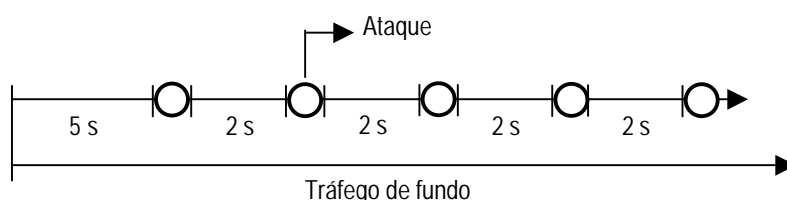


Figura 12 - Seqüência de reprodução dos tráfegos utilizados na análise de escalabilidade

Cada um dos tipos de ataques considerado na avaliação (negação de serviços, evasão, inserção e varredura de portas) foi executado separadamente um do outro, mas sempre em paralelo ao tráfego de fundo. Por exemplo, considerando que a figura acima seja referente a análise de escalabilidade de um IDS frente aos ataques de negação de serviço, tem-se cinco ataques (*Smurf*, *UDP Storm*, *Syn Flood*, *Teardrop* e *ICMP Fragmentation*), ambos representados na figura 11 pelos círculos, e sendo reproduzidos em paralelo ao tráfego de fundo. Esses ataques são executados uns após os outros separados em intervalos de dois segundos, o que facilita a leitura dos *logs*, posteriormente, pois os ataques ocorrem em instantes distintos. A fim de que o tráfego de fundo já esteja sendo gerado no momento em que o primeiro ataque for realizado, é previsto um período de cinco segundos antes do início da execução dos ataques. A geração do tráfego de fundo é finalizada somente após todos os ataques terem sido reproduzidos.

O *shell script* abaixo ilustra a seqüência em que os ataques de negação de serviço, do exemplo em questão, devem ser reproduzidos.

```
# /bin/bash

# Identificação: script_dos

# Este script reproduz os ataques de negação de serviço
# selecionados. A sintaxe do comando tcpreplay utilizado nesse
# script informa a interface de rede utilizada para reproduzir o
# tráfego e o nome do arquivo (tcpdump) que representa o tráfego de # ataque a ser gerado.

sleep 5
tcpreplay -i eth0 smurf
sleep 2
tcpreplay -i eth0 udpstorm
sleep 2
tcpreplay -i eth0 synflood
sleep 2
tcpreplay -i eth0 teardrop
sleep 2
tcpreplay -i eth0 icmpfrag
```

Para cada tipo de ataque deve haver um *shell script* que reproduza um a um dos ataques selecionados para este grupo de ataques. Ao término da reprodução de um determinado tipo de ataque à uma determinada taxa, o sistema de *log* deve ser parado e deve-se registrar o número de alertas gerado pelo IDS. A tabela 8 abaixo ilustra a forma como esses dados foram registrados. Inicialmente registrou-se o número de alertas gerados para cada ataque, quando submetidos a uma determinada taxa de transmissão do tráfego de fundo. Em seguida, foram calculados os totais de alarmes gerados a cada taxa e o percentual de ataques detectados.

Identificação: IDS1
Tipo de ataque: negação de serviço

Taxas (Mbps)	Smurf	UDP Storm	Syn Flood	Teardrop	ICMP	Total	Percentual
4	100	150	125	115	1500	1990	100%
6	80	75	65	62	750	1032	51,6%
8	40	35	32	31	250	388	19,5%

Tabela 8 - Exemplo de resultados da análise de escalabilidade

A partir dos dados exibidos na tabela 8 observou-se que a 4Mbps este IDS não perdeu alarmes. Já nas taxas de transmissão seguintes o percentual de ataques detectados foi reduzindo bruscamente.

Uma vez que todos os tipos de ataques tenham sido reproduzidos em todas as taxas estipuladas, pode-se gerar os gráficos referentes aos resultados obtidos por esses IDSs para cada tipo de ataque, conforme o gráfico ilustrado a seguir.

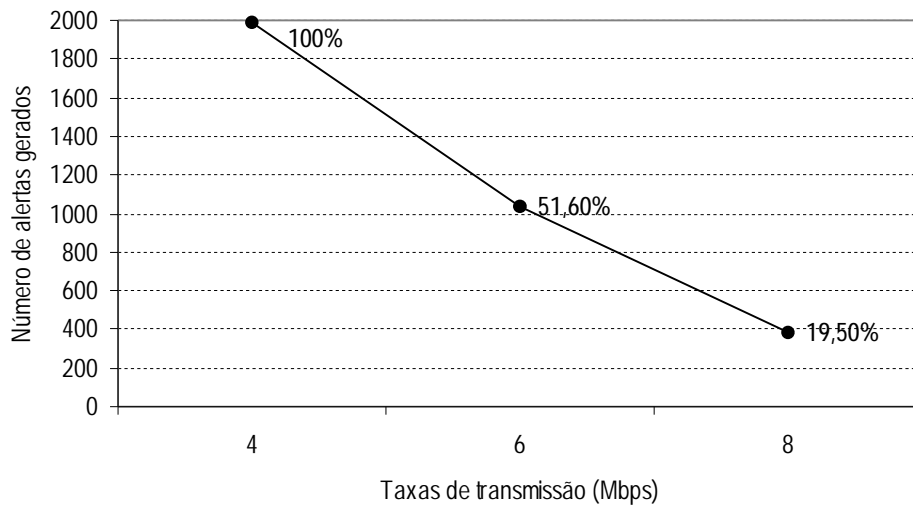


Figura 13 - Exemplo dos resultados da análise de escalabilidade em relação aos ataques de negação de serviço

Em seguida, os resultados alcançados pelos IDSs avaliados foram dispostos a fim de comparar os valores obtidos. A tabela 9 representa esta comparação.

Taxas (Mbps)	IDS1	IDS2
4	100%	100%
6	51,60%	31,60%
8	19,50%	9,50%

Tabela 9 - Exemplo dos resultados da análise de escalabilidade em relação aos ataques de negação de serviço

Os dados da tabela acima podem ser representados através figura abaixo.

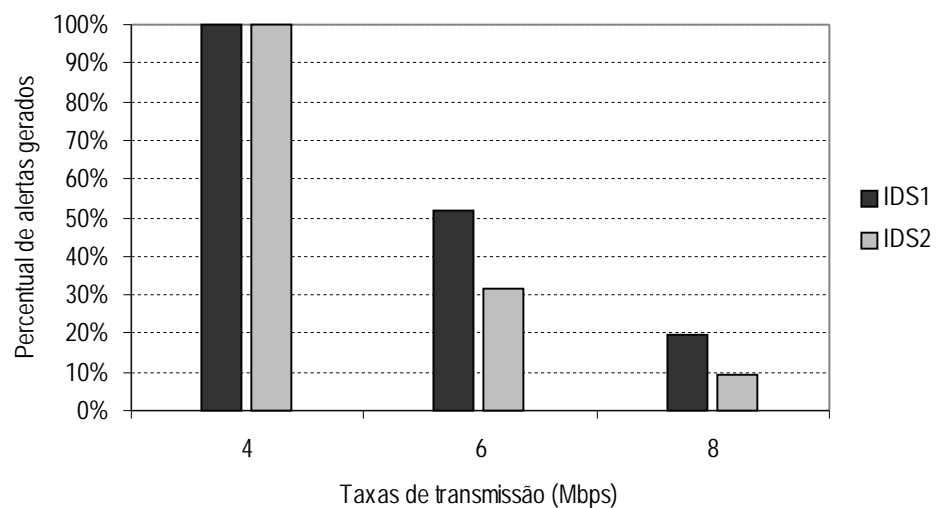


Figura 14 - Exemplo de comparação dos resultados da análise de escalabilidade em relação aos ataques de negação de serviços

3.6.3 Taxa de falsos positivos

Falsos positivos são todos os alarmes que indicam que um determinado ataque está em execução, quando de fato o que está ocorrendo é outro tipo de atividade. Por exemplo, um usuário do suporte executa um comando *ping* para um servidor, o IDS registra este evento como um ataque. Outro exemplo de falsos positivos é quando a rede está sofrendo um determinado tipo de ataque (*UDPstorm*), e o IDS gera alarmes tanto para este ataque, quanto para outros tipos de ataques (*ICMP fragmentation*) que não estão ocorrendo e não possuem relação com o evento em questão [McHugh, 2000].

Ao longo dos estudos desenvolvidos durante essa monografia, verificou-se que para avaliar a incidência de falsos positivos de forma a seguir a proposta dessa metodologia, de ser uma abordagem prática de fácil execução e na qual o IDS é visto como uma caixa preta, as alternativas mais prudentes são: (a) realizar essa análise em relação ao tráfego de fundo coletado junto a rede das instituições ou (b) em função do tráfego de ataques utilizado nos experimentos anteriores.

A análise de falsos positivos, tendo como parâmetro o tráfego de fundo coletado junto a rede de uma instituição é conduzida da seguinte forma, após coletar durante um determinado período (dias ou semanas) o tráfego diário de um segmento de rede de uma instituição, esse tráfego é reproduzido em paralelo ao conjunto de ataques detectados na análise da capacidade de detecção. Dessa maneira, após analisar os *logs* é possível determinar quais alertas correspondem a falsos positivos e, ainda, quais os alertas de fato representam os ataques gerados. Essa forma de análise apresenta como limitação, a impossibilidade de garantir que no tráfego coletado não estejam inseridos ataques, que por ventura tenham ocorrido durante o período de coleta desse tráfego. Caso isto ocorra a análise de falsos positivos está comprometida, uma vez que o número de ataques inseridos no tráfego de fundo é desconhecido. Já a análise de falsos positivos em relação a um tráfego de ataque conhecido é uma abordagem que permite, a partir da análise dos *logs* gerados nos testes de capacidade de detecção, identificar a quantidade de falsos positivos resultante. Por exemplo, através da ferramenta *jolt2*, conforme descrito na seção 3.3.1, é possível determinar o número de pacotes ICMP que irão compor o ataque conhecido por *ICMP fragmentation*. Caso o *log* do IDS avaliado registre um número maior de alarmes do que o número de pacotes gerados, ou ainda, apresente qualquer outro tipo de registro referente a outros ataques, estes serão os falsos positivos.

A forma de análise utilizada nesta metodologia obtém a taxa de falsos positivos em relação ao tráfego de ataque utilizado nas análises de capacidade de detecção e escalabilidade. Para tal, os arquivos de *log* resultantes dos testes de capacidade de detecção foram novamente consultados. No entanto, os registros contabilizados nessa consulta foram todos aqueles que não faziam referência a nenhum dos ataques executados. A tabela abaixo ilustra a forma como esses dados foram registrados.

Identificação: IDS1

Tipo de ataque: negação de serviço

Alertas	Smurf	UDP Storm	Syn Flood	Teardrop	ICMP	Total
Positivos	100	150	125	115	1500	1990
Falsos positivos	25	0	100	96	750	975
Total de alarmes:						2965
Taxa de falsos positivos:						32,80%

Tabela 10 - Exemplo dos resultados da análise das taxas de falsos positivos em relação aos ataques de negação de serviço

Uma vez conhecidas, as quantidades de alertas reais e de falsos positivos obtêm-se a taxa de falsos positivos em relação ao tráfego de ataque em questão (negação de serviço). Este procedimento é repetido para cada um dos tipos de ataques considerados na avaliação. Sendo assim, as taxas de falsos positivos obtidos para cada um dos tipos de ataques são apresentadas conforme o exemplo a seguir.

Alertas	Evasão	Inserção	Varredura de portas	Negação de serviço
IDS1	15,20%	17,68%	27,70%	32,80%
IDS2	10,70%	15,00%	21,00%	31%

Tabela 11 - Exemplo dos resultados da análise das taxas de falsos positivos obtidos por dois IDSs

Os dados obtidos nessa análise podem ser representados através de um gráfico de barras conforme o exemplo da figura abaixo.

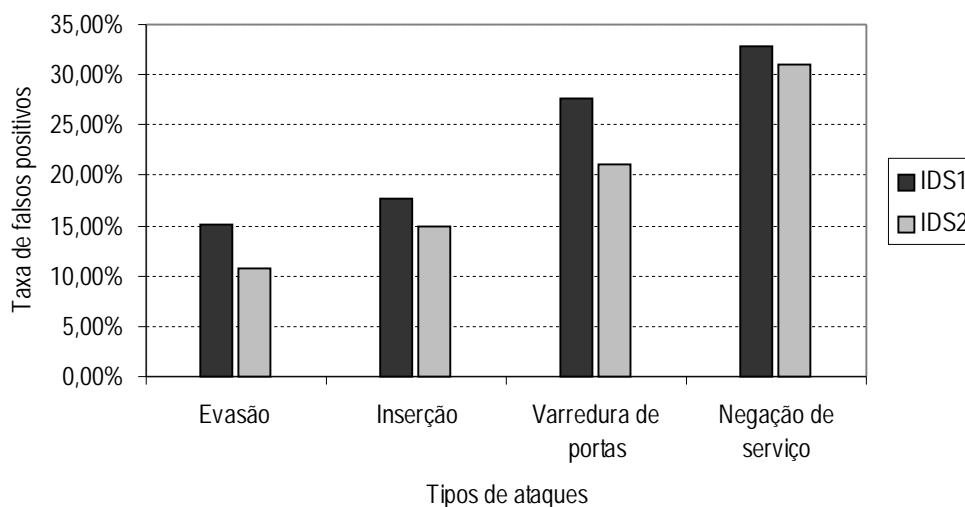


Figura 15 - Exemplo de comparação entre as taxas de falsos positivos geradas pelos IDSs avaliados

4 Estudo de caso

Nesse capítulo serão apresentados os resultados alcançados pelos dois IDSs submetidos à metodologia descrita no capítulo anterior. Os sistemas de detecção de intrusão utilizados nesse estudo de caso foram os seguintes: o *Snort 1.83* e o *Firestorm 0.4.6*, ambos disponíveis sobre licença GNU GPL versão 2.

O *Snort* é um dos sistemas de detecção de intrusão mais utilizados no momento, esta ferramenta combina simplicidade e eficiência. De livre distribuição, desenvolvido por Marty Roesch, esse IDS coleta os dados junto a rede e os compara com uma base de assinaturas. A base de assinatura do *Snort* é atualizada semanalmente. Atualmente, existem versões desse IDS para *Linux* e *Windows* [Campello, 2001].

Semelhante a ferramenta anterior, o *Firestorm* é um sistema de detecção de intrusão baseado em rede, cuja análise dos dados é realizada a partir de uma base de assinaturas. Este IDS é distribuído livremente e foi desenvolvido exclusivamente para ambiente *Linux* [Tedesco, 2002].

As duas ferramentas escolhidas apresentam, segundo os seus desenvolvedores, capacidade para detectar todos os tipos de ataques conhecidos até o presente momento. Além disso, tanto o responsável pela criação do *Snort* (Marty Roesch) quanto do *Firestorm* (Gianni Tedesco) afirmam que os seus IDSs são extremamente velozes e que apresentam uma baixa taxa de falsos positivos.

A seguir os resultados obtidos pelo *Snort* e pelo *Firestorm* serão apresentados. A apresentação desses dados será através de tabelas e gráficos.

4.1 Capacidade de detecção

A análise da capacidade de detecção foi realizada em relação à seleção de ataques apresentada na seção 3.2.2. Essa análise foi realizada simultaneamente com os dois IDSs escolhidos. Os resultados obtidos tanto pelo *Snort* quanto pelo *Firestorm* são apresentados na tabela abaixo.

Ataques	Snort	Firestorm
Evasão		
Method Matching	X	X
Session Splicing	X	X
Inserção		
Long URLs	X	X
Self Reference	X	X
URL Encoding	X	-
Varredura de Portas		
UDP Scan	X	-
Xmas	X	X
TCP Fragmentation	X	X

Varredura do protocolo IP	X	-
Fingerpringing	X	X
Ident Reverso TCP	X	X
Negação de serviço		
Smurf	X	X
UDP Storm	X	X
Syn Flood	X	X
Teardrop	X	X
ICMP Fragmentation	X	X

Tabela 12 - Resultados obtidos na análise da capacidade de detecção

Conforme mostrado na tabela 12 o *Snort* detectou todos os ataques aos quais foi submetido. Já o *Firestorm* não detectou um ataque de inserção (*URL Encoding*) e dois ataques de varredura de portas (*UDP Scan* e sondagem do protocolo IP). Quando submetido ao ataque de inserção, o *Firestorm*, ao invés de identificar esse ataque, gerou diversos alertas de varredura de portas, tais como "*Portscan Detected*". Em relação ao ataque *UDP Scan* foram gerados alertas do tipo "*UDP echo+chargen*" que correspondem ao ataque de negação de serviço *UDPstorm*. Para a varredura do protocolo IP não foram gerados alertas.

Os resultados obtidos nessa análise demonstram que o *Snort* é uma ferramenta de detecção de intrusão capaz de detectar ataques de inserção, evasão, varredura de portas e negação de serviço de forma bastante eficiente. Em relação ao *Firestorm* constatou-se que não há um mecanismo eficiente de decodificação de requisições HTTP. O desenvolvimento de melhorias em relação a este mecanismo consta como um dos objetivos para as futuras versões deste IDS [Tedesco, 2002].

4.2 Escalabilidade

As análises de escalabilidade foram realizadas mediante uma limitação no que tange as taxas de transmissão em que o tráfego de fundo foi reproduzido. Esta limitação consistiu na utilização de um *hub* de 10 Mbps, o que, por conseguinte não possibilitou reproduzir o tráfego de fundo a velocidades mais elevadas. A reprodução de taxas de transmissão superior possibilitaria uma análise mais ampla.

Os testes de escalabilidade foram realizados individualmente para cada um dos IDSs avaliados. Isto ocorreu em função dos diferentes resultados apresentados na análise da capacidade de detecção.

Conforme descrito anteriormente através dos experimentos realizados constatou-se que a uma taxa de transmissão de 4 Mbps, os IDSs avaliados não apresentam descartes de pacotes. Sendo assim os números de alertas gerados nessa taxa correspondem ao número de pacotes efetivamente gerados. A seguir serão apresentados os resultados obtidos pelos sistemas de detecção de intrusão avaliados.

4.2.1 Análise de escalabilidade do *Snort*

A tabela 13 apresenta o número de alertas gerados para cada um dos ataques de inserção, detectados pelo *Snort* na análise anterior, ao longo das três taxas de transmissão nas quais o tráfego de fundo foi reproduzido.

Taxa (Mbps)	Long URL	Self Reference	URL Encoding	Total de alertas	Alertas
4	642	96	101	839	100%
6	638	92	99	829	98,81%
8	595	77	83	755	89,99%

Tabela 13 - Resultados obtidos na análise de escalabilidade do *Snort* em relação aos ataques de inserção

Conforme ilustrado na figura 16, a diferença entre o número total de alertas gerados nas duas primeiras taxas é de 1,19%. Já a 8 Mbps a quantidade de alertas gerados reduz aproximadamente 10% em relação à primeira taxa de transmissão(4 Mbps).

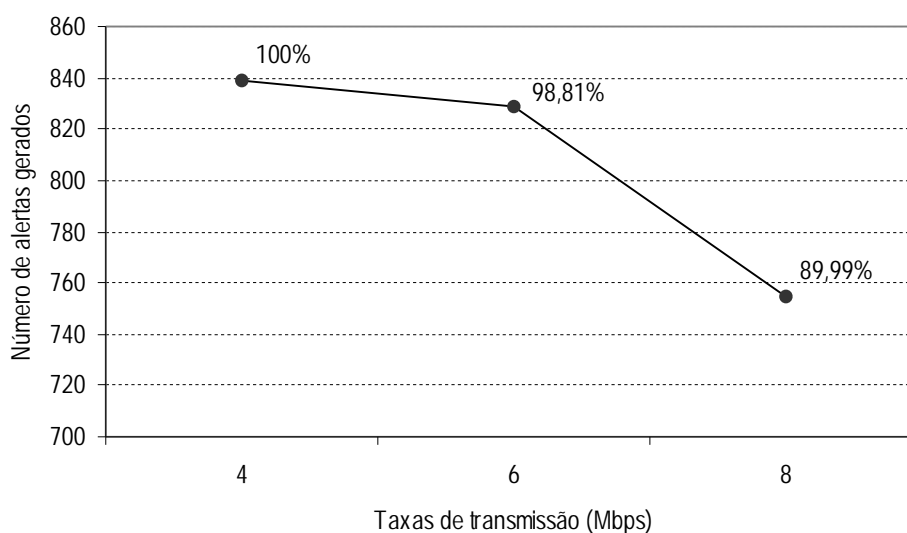


Figura 16 - Análise de escalabilidade do *Snort* em relação aos ataques de inserção

Em relação aos ataques de evasão observou-se, conforme os dados apresentados na tabela 14, que a 6 Mbps o IDS em comparação a primeira taxa gerou 2,74% a menos de alertas. Entretanto, uma vez que o tráfego de fundo foi reproduzido na taxa máxima prevista nesse estudo de caso, o número de alertas em relação a taxa de 4 Mbps foi reduzido em aproximadamente 14%.

Taxa (Mbps)	Method Matching	Session Splice	Total de alertas	Alertas
4	111	76	187	100%
6	109	74	183	97,86%
8	98	63	161	86,10%

Tabela 14 - Resultados obtidos na análise de escalabilidade do *Snort* em relação aos ataques de evasão

A figura 6, a seguir, ilustra os resultados supracitados.

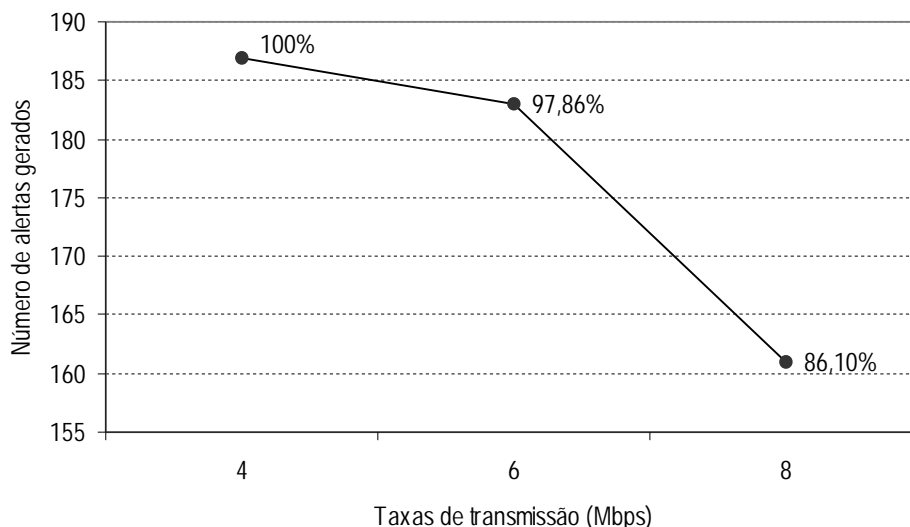


Figura 17 - Análise de escalabilidade do *Snort* em relação aos ataques de evasão

Quanto aos ataques de varredura de portas, cujos percentuais de alertas gerados a cada taxa de transmissão constam na tabela 15, constatou-se que o *Snort* gerou praticamente o mesmo número de alertas nas duas primeiras taxas. Sendo que somente a 8 Mbps a diferença entre o número de alertas gerados, em relação a primeira taxa, pode ser considerada significativa como mostra a figura 18.

Taxa (Mbps)	Fragmentation	Fingerprinting	Ident Reverso	Protocolo IP	Xmas	UDP Scan	Total de alertas	Alertas
4	1115	15	9	251	942	35	2367	100%
6	1109	13	9	249	936	35	2351	99,32%
8	1084	9	9	237	877	29	2245	94,85%

Tabela 15 - Resultados obtidos na análise de escalabilidade do *Snort* em relação aos ataques de varredura de portas

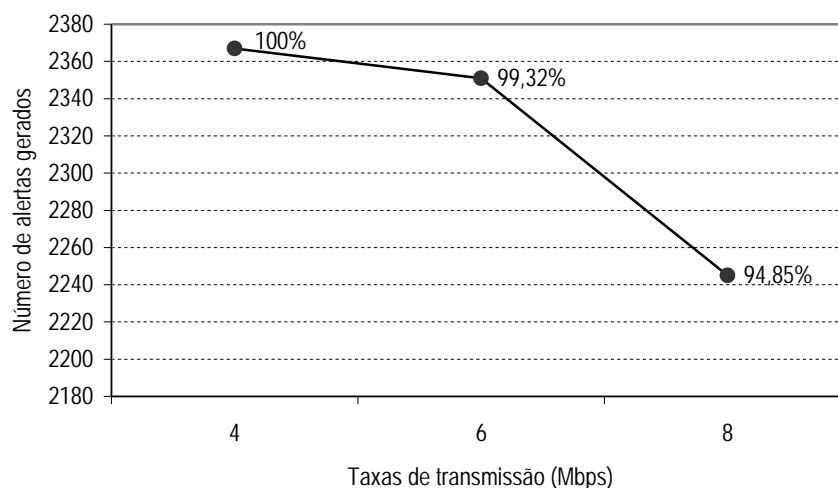


Figura 18 - Análise de escalabilidade do *Snort* em relação aos ataques de varredura de portas

Na análise de escalabilidade em relação aos ataques de negação de serviços, novamente o *Snort* apresentou resultados bastante próximos nas duas primeiras taxas de transmissão, como pode ser observado na tabela 16.

Taxa (Mbps)	Smurf	Teardrop	UDP	ICMP	Syn Flood	Total de alertas	Alertas
4	2000	110	215	2010	1000	5335	100%
6	1996	108	215	2008	999	5326	99,83%
8	1870	95	212	1909	951	5037	94,41%

Tabela 16 - Resultados obtidos na análise de escalabilidade do *Snort* em relação aos ataques de negação de serviços

Os dados da tabela acima estão representados na figura 19. A partir desse gráfico percebe-se que há uma queda de aproximadamente 5,5% na quantidade de alertas gerados entre a primeira e a última taxa de transmissão.

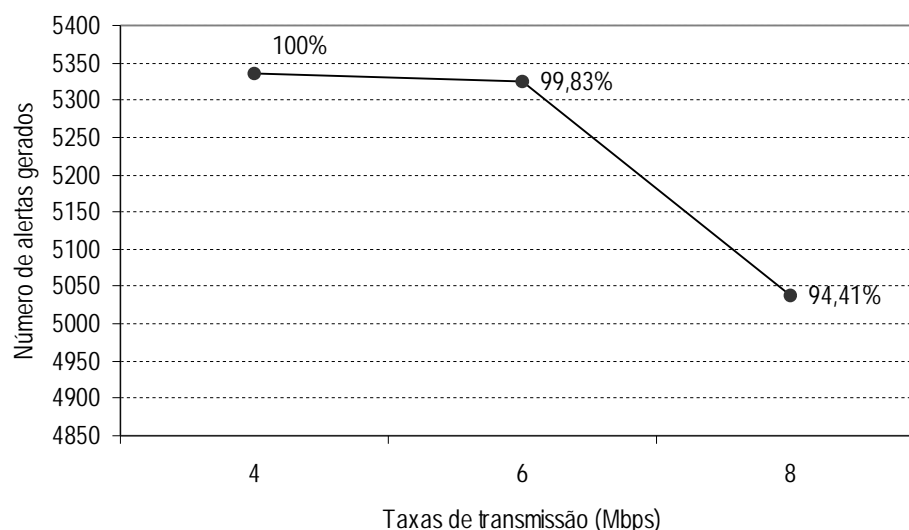


Figura 19 - Análise de escalabilidade do *Snort* em relação aos ataques de negação de serviço

Na tabela 17 constam as quantidades de alertas geradas para cada um dos ataques de inserção, detectados pelo *Firestorm* na análise da capacidade de detecção ao longo das três taxas de transmissão nas quais o tráfego de fundo foi reproduzido.

Taxa (Mbps)	Long URL	Self Reference	Total de alertas	Alertas
4	642	96	738	100%
6	630	90	720	97,56%
8	560	77	635	86,04%

Tabela 17 - Resultados obtidos na análise de escalabilidade do *Firestorm* em relação aos ataques de inserção

Conforme ilustrado na figura 20, a diferença entre o número total de alertas gerados nas duas primeiras taxas é de 2,44%. Já a 8 Mbps a quantidade de alertas gerados reduz aproximadamente 14% em relação à primeira taxa de transmissão(4 Mbps).

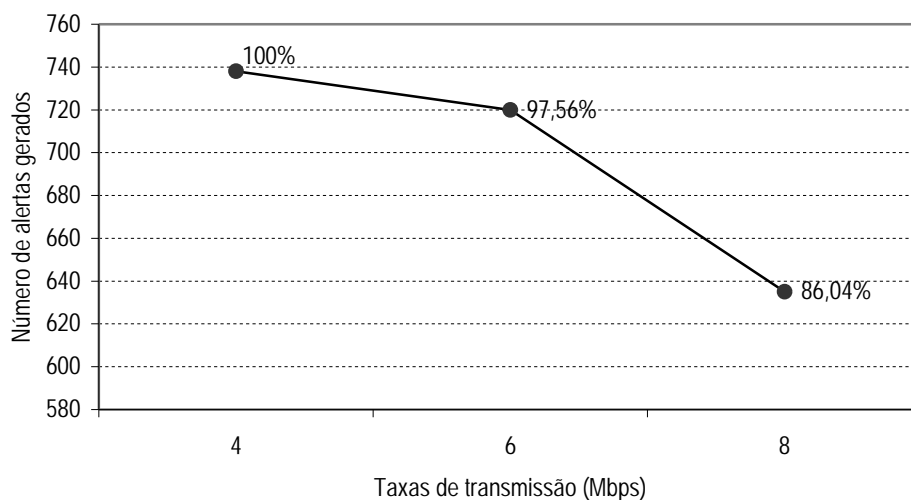


Figura 20 - Análise de escalabilidade do *Firestorm* em relação aos ataques de inserção

Referente aos ataques de evasão cujos percentuais de alertas gerados a cada taxa de transmissão constam na tabela 18, verificou-se que o *Firestorm* apresentou uma diferença de aproximadamente 5% em relação à quantidade de alertas gerados nas duas primeiras taxas. Quando submetido a um tráfego de fundo reproduzido a uma taxa de transmissão de 8 Mbps, a diferença entre o número de alertas gerados em relação à primeira taxa foi de praticamente 16,6%. A figura 21, abaixo, representa os resultados obtidos nessa análise.

Taxa (Mbps)	Method Matching	Session Splice	Total de alertas	Alertas
4	111	76	187	100%
6	105	73	178	95,19%
8	91	65	156	83,42%

Tabela 18 - Resultados obtidos na análise de escalabilidade do *Firestorm* em relação aos ataques de inserção

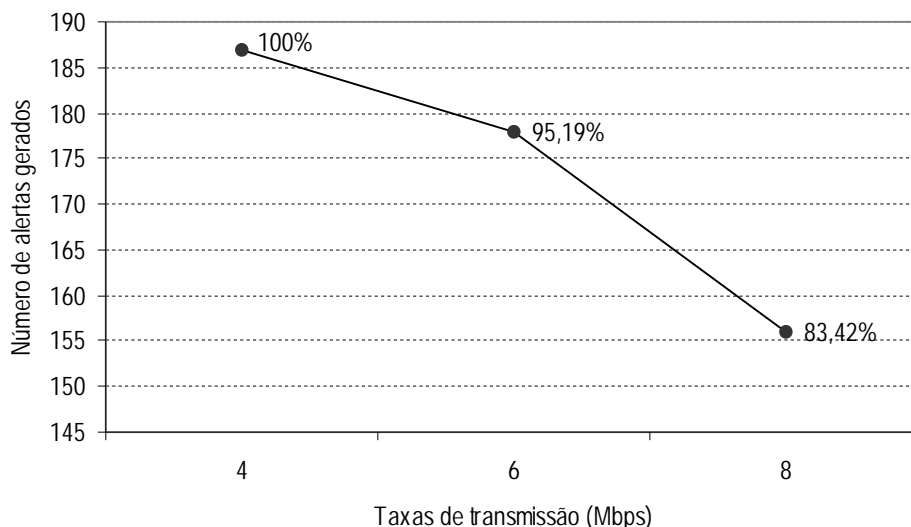


Figura 21 - Análise de escalabilidade do *Firestorm* em relação aos ataques de evasão

Em relação aos ataques de varredura de portas observou-se, conforme os dados apresentados na tabela 19, que nas duas primeiras taxas nas quais o tráfego de fundo foi reproduzido o percentual de alertas gerados é praticamente o mesmo. Entretanto, a uma taxa de 8 Mbps o percentual de alertas gerados foi reduzido a aproximadamente 10% em relação às duas primeiras taxas. A figura 22 ilustra os resultados obtidos na análise.

Taxa (Mbps)	TCP Fragmentation	Fingerprinting	Ident Reverso	Xmas	Total de alertas	Alertas
4	1115	15	9	942	2081	100%
6	1113	14	9	936	2072	99,57%
8	998	10	8	867	1883	90,49%

Tabela 19 - Resultados obtidos na análise de escalabilidade do *Firestorm* em relação aos ataques de varredura de portas

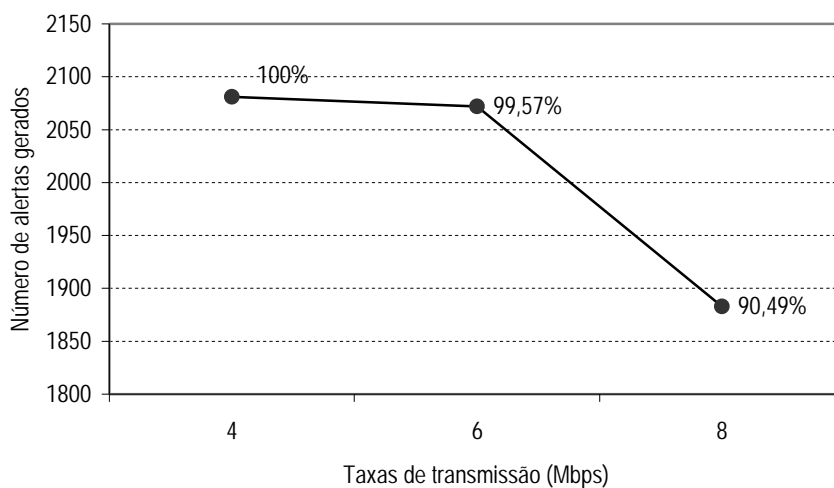


Figura 22 - Análise de escalabilidade do *Firestorm* em relação aos ataques de varredura de portas

Na análise de escalabilidade em relação aos ataques de negação de serviço, novamente os percentuais de alertas gerados nas duas primeiras taxas foram praticamente iguais. Já o que se constatou em relação ao tráfego de fundo reproduzido a 8 Mbps é uma queda de aproximadamente 8% no percentual de alertas gerados. A figura 23 representa estes resultados.

Taxa (Mbps)	Smurf	Teardrop	UDP	ICMP	Syn Flood	Total de alertas	Alertas
4	2000	110	215	2010	1000	5335	100%
6	1991	101	212	1998	999	5301	99,36%
8	1817	91	205	1867	941	4921	92,24%

Tabela 20 - Resultados obtidos na análise de escalabilidade do *Firestorm* em relação aos ataques de negação de serviços

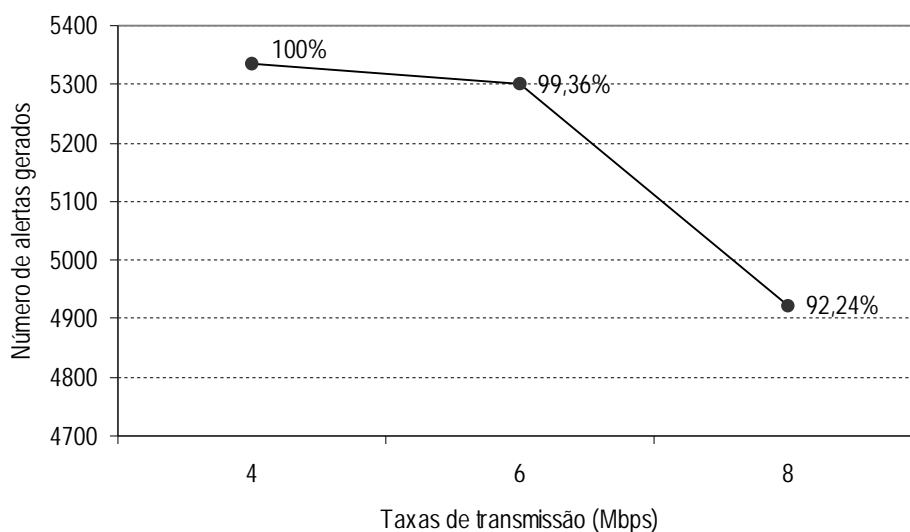


Figura 23 - Análise de escalabilidade do *Firestorm* em relação aos ataques de negação de serviço

4.2.2 Comparação dos resultados da análise de escalabilidade

A tabela 21 reúne os percentuais de alertas gerados pelos IDSs avaliados, em relação aos ataques de evasão a cada taxa de transmissão estipulada.

Taxa(Mbps)	Snort	Firestorm
4	100%	100%
6	98,81%	97,56%
8	89,99%	86,04%

Tabela 21 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de evasão

Conforme os dados representados na figura 24, nota-se que o *Snort* apresentou uma pequena superioridade em relação ao *Firestorm* quanto à análise de escalabilidade frente aos ataques de evasão. É importante ressaltar que, embora os percentuais apresentados sejam bastante próximos, o *Firestorm* não detectou um dos ataques de evasão (URL Encoding).

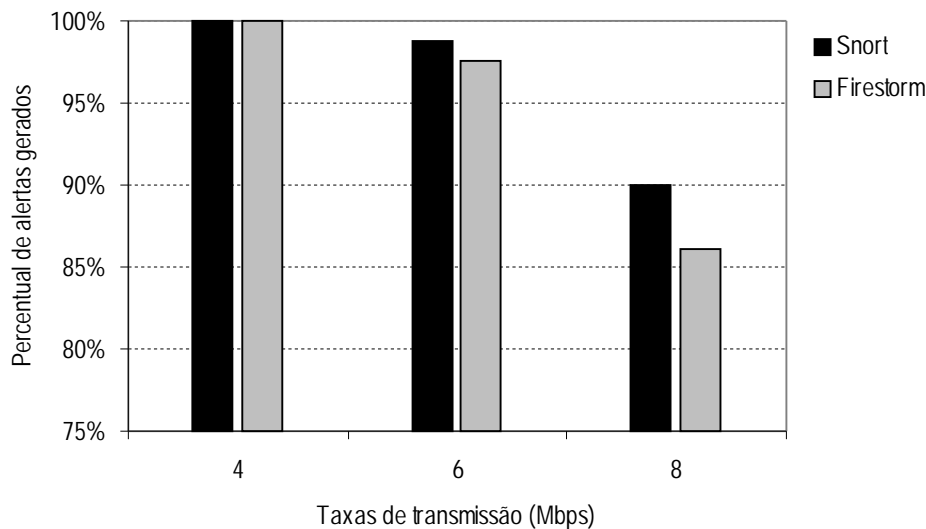


Figura 24 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de evasão

A análise de escalabilidade realizada em relação aos ataques de inserção apresentou resultados favoráveis ao *Snort*, embora a diferença entre os percentuais de alarmes gerados entre os dois IDSs, novamente, tenha sido muito pequena.

Taxa(Mbps)	Snort	Firestorm
4	100%	100%
6	97,86%	95,19%
8	86,10%	83,42%

Tabela 22 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de inserção

O gráfico ilustrado na figura 25, abaixo, representa os resultados obtidos nessa análise pelos dois IDSs avaliados.

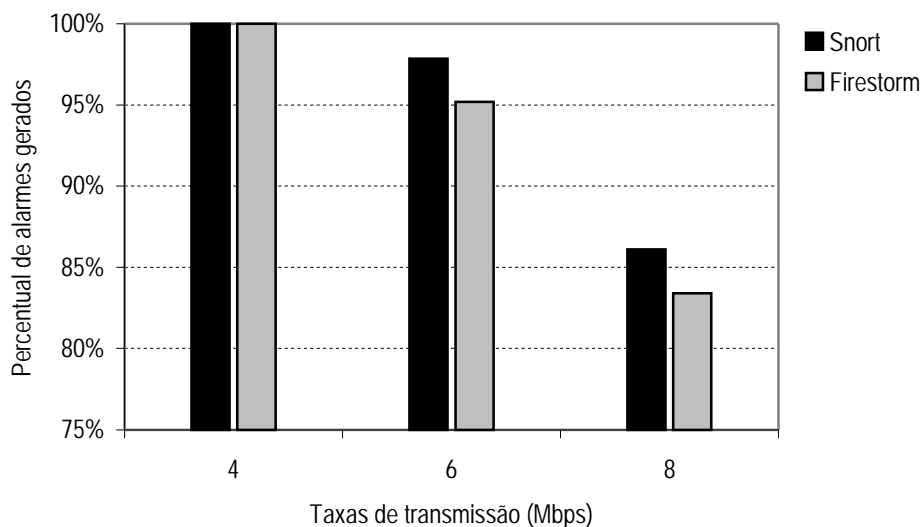


Figura 25 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de inserção

Na análise de escalabilidade que considera os ataques de varredura de portas, novamente registrou-se muito equilíbrio entre os resultados obtidos. No entanto, conforme já mencionado na seção anterior, o *Firestorm* na análise da capacidade de detecção não detectou dois ataques de varredura de portas.

Taxa(Mbps)	Snort	Firestorm
4	100%	100%
6	99,32%	99,57%
8	94,85%	90,49%

Tabela 23 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de varredura de portas

Os resultados obtidos nessa análise foram representados na figura 26 apresentada a seguir.

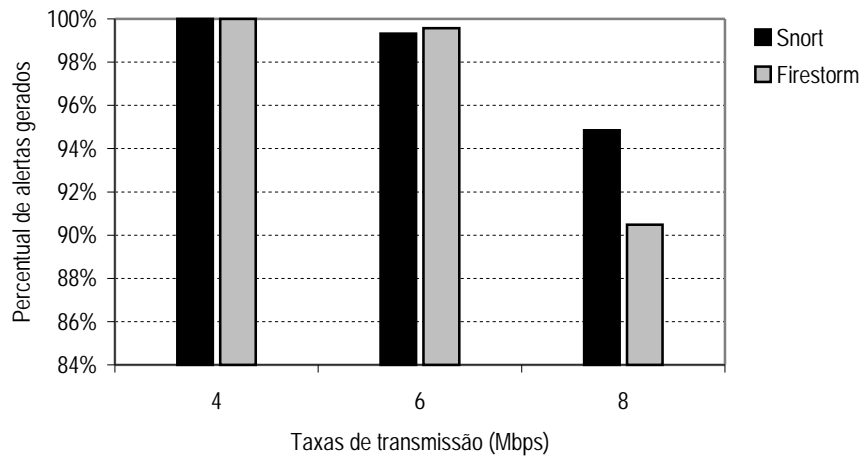


Figura 26 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de varredura de portas

Os teste de escalabilidade em relação aos ataques de negação de serviço confirmaram o equilíbrio existente ao longo dessa análise. No entanto, novamente o sistema de detecção de intrusão que apresentou os melhores resultados foi o *Snort*.

Taxa(Mbps)	Snort	Firestorm
4	100%	100%
6	99,83%	99,36%
8	94,41%	92,24%

Tabela 24 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de negação de serviço

O equilíbrio entre os resultados obtidos pelos dois IDSs nessa última análise de escalabilidade é representado na figura 27, a seguir.

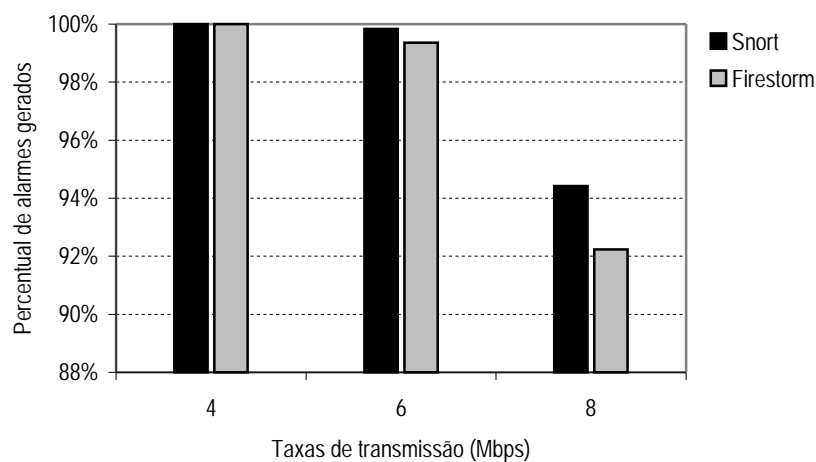


Figura 27 - Comparação dos resultados obtidos na análise de escalabilidade em relação aos ataques de negação de serviço

4.3 Taxas de falsos positivos

A análise das taxas de falsos positivos, conforme descrita no capítulo anterior, é realizada mediante consultas aos arquivos de *logs* criados pelos IDSs no momento da análise da capacidade de detecção.

4.3.1 Análise das taxas de falsos positivos gerados do *Snort*

A tabela 25 apresenta o percentual de falsos positivos gerados pelo *Snort* em relação aos ataques de evasão. Este percentual de 36,73% é formado principalmente pela grande quantidade de alarmes falsos gerados em função do ataque *Session Splice*.

Alertas	Method Matching	Session Splice	Subtotal
Verdadeiros	111	75	186
Falsos positivos	16	92	108
Total:			294
Taxa de falsos positivos:			36,73%

Tabela 25 - *Snort*: percentual de falsos positivos em relação aos ataques evasão

A taxa de falsos positivos em relação aos ataques de inserção é mostrada na tabela 26. Esta taxa é de aproximadamente 30% e o ataque que gerou a maior quantidade de alarmes falsos foi o *Long URL*.

Alertas	Long URL	Self Reference	URL Enc	Subtotal
Verdadeiros	642	96	101	839
Falsos positivos	321	25	13	359
Total:				1198
Taxa de falsos positivos:				29,97%

Tabela 26 - *Snort*: percentual de falsos positivos em relação aos ataques inserção

Em relação aos ataques de varredura de portas, tabela 27, foram gerados apenas 4,71% de falsos positivos, sendo que para as sondagens de protocolo IP e UDP não foram gerados alarmes falsos.

Alertas	TCP Fragmentation	Fingerprinting	Ident Reverso	Protocolo IP	Xmas	UDP Scan	Subtotal
Verdadeiros	1115	15	9	251	942	35	2367
Falsos positivos	24	10	6	0	77	0	117
Total:							2484
Taxa de falsos positivos:							4,71%

Tabela 27 - *Snort*: percentual de falsos positivos em relação aos ataques de varredura de portas

Ao exemplo dos ataques de varredura de portas os ataques de negação de serviços, tabela 28, geraram uma pequena taxa de falsos positivos. Além disso, o *Snort* não gerou alarmes falsos para o ataque *UDPstorm*.

Alertas	Smurf	Teardrop	UDP	ICMP	Syn Flood	Subtotal
Verdadeiros	2000	110	215	2010	1000	5335
Falsos positivos	105	7	0	89	58	259
Total:						5594
Taxa de falsos positivos:						4,63%

Tabela 28 - *Snort*: percentual de falsos positivos em relação aos ataques negação de serviço

4.3.2 Análise das taxas de falsos positivos gerados do *Firestorm*

Em relação aos ataques de evasão, o *Firestorm* gerou aproximadamente 41% de alarmes falsos (tabela 29). Sendo que a exemplo do que ocorreu na análise com o *Snort*, o ataque denominado *Session Splice* foi o responsável pela maior quantidade de falsos positivos gerados.

Alertas	Method Matching	Session Splice	Subtotal
Verdadeiros	111	75	186
Falsos positivos	35	96	131
Total:			317
Taxa de falsos positivos			41,32%

Tabela 29 - *Firestorm*: percentual de falsos positivos em relação aos ataques evasão

A tabela 30 mostra a quantidade de falsos positivos gerados para os ataques de inserção. Esta taxa corresponde a 42,97% de falsos positivos gerados. O ataque denominado *URL Enconding*, embora não tenha sido detectado pelo *Firestorm* foi o responsável por 114 alarmes falsos. Além disso, a exemplo da análise com o *Snort* o ataque *Long URL* gerou um elevado percentual de falsos positivos.

Alertas	Long URL	Self Reference	URL Enconding	Subtotal
Verdadeiros	642	96	0	738
Falsos positivos	397	45	114	556
Total de alertas:				1294
Taxa de falsos positivos:				42,97%

Tabela 30 - *Firestorm*: percentual de falsos positivos em relação aos ataques inserção

O IDS em questão gerou uma taxa de aproximadamente 13% de alarmes falsos em relação aos ataques de varredura de portas. Conforme os dados da tabela 31, os ataques que contribuíram significativamente para esta taxa foram *Fingerprinting* e *UDP Scan* que, embora não tenha sido detectado, gerou falsos positivos.

Alertas	TCP Fragmentation	Fingerprinting	Ident Reverso	Xmas	UDP Scan	Subtotal
Verdadeiros	1115	15	9	942	0	2081
Falsos positivos	98	15	7	154	35	309
Total de alertas:						2390
Taxa de falsos positivos:						12,93%

Tabela 31 - Firestorm: percentual de falsos positivos em relação aos ataques de varredura de portas

A taxa de falsos positivos gerados em relação aos ataques de negação de serviço foi de 7,47%. A exemplo do que ocorreu com o *Snort*, não foram gerados falsos positivos para o ataque *UDPstorm*.

Alertas	Smurf	Teardrop	UDP	ICMP	Syn Flood	Subtotal
Verdadeiros	2000	110	215	2010	1000	5335
Falsos positivos	181	17	0	64	169	431
Total de alertas:						5766
Taxa de falsos positivos:						7,47%

Tabela 32 - Firestorm: percentual de falsos positivos em relação aos ataques de negação de serviço

4.3.3 Comparação dos resultados da análise de taxas de falsos positivos

Ao analisar os resultados da tabela 33 conclui-se que o *Snort*, em geral, retornou um número de falsos positivos menor do que o *Firestorm*.

Alertas	Evasão	Inserção	Varredura de portas	Negação de serviço
Snort	36,73%	29,97%	4,71%	4,63%
Firestorm	41,32%	42,97%	12,93%	7,47%

Tabela 33 - Comparação entre os resultados obtidos na análise das taxas de falsos positivos

Conforme se pode observar na figura 28, a diferença entre as taxas de falsos positivos gerados por esses IDSs nos ataques de evasão e negação de serviço não ultrapassaram a margem dos 5%. Entretanto, para os ataques de varredura de portas, a diferença entre as taxas de falsos positivos geradas é de aproximadamente 8% a favor do *Snort*. Já em relação aos ataques de inserção, registrou-se a maior diferença entre os dois IDSs avaliados; praticamente 12%. Novamente o *Firestorm* gerou uma quantidade mais elevada de falsos positivos.

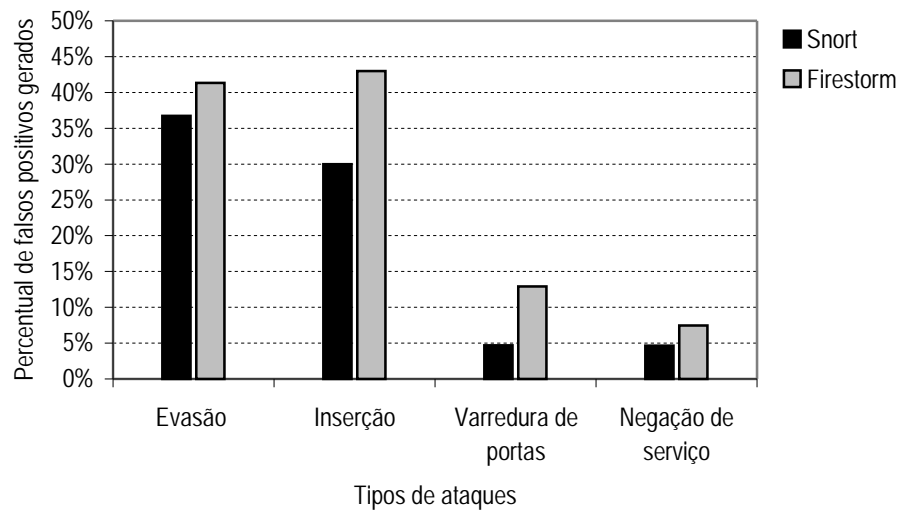


Figura 28 - Comparação entre os resultados obtidos na análise das taxas de falsos positivos

5 Considerações finais

No decorrer desta monografia apresentou-se uma metodologia para avaliação de sistemas de detecção de intrusão. Esta nova abordagem constitui-se em uma metodologia factível fora do ambiente acadêmico, que possui um conjunto de procedimentos sistematizados, realizáveis em um curto espaço de tempo e que não exige o conhecimento prévio das ferramentas de detecção de intrusão a serem avaliadas. A metodologia proposta, ao contrário de outras abordagens, avalia as potencialidades dos IDSs e não a implementação ou a base de assinaturas dos mesmos. As análises previstas nessa metodologia avaliam as seguintes características: capacidade de detecção, taxas de falsos positivos e escalabilidade dos sistemas avaliados.

Este trabalho foi estruturado da seguinte forma: o capítulo 1 apresentou uma introdução ao tema proposto e os aspectos que motivaram esse estudo. No capítulo 2 realizou-se uma síntese das principais publicações relacionadas ao tema em questão. Já o capítulo 3 corresponde a descrição da metodologia proposta para avaliação de IDSs. Os resultados obtidos no estudo de caso foram apresentados no capítulo 4. As principais contribuições da metodologia desenvolvida neste trabalho são abordadas a seguir.

5.1 Seleção de ataques

Conforme descrito no capítulo 2 as principais metodologias voltadas à avaliação de sistemas de detecção de intrusão, partem da premissa de que quanto maior a quantidade de ataques executados mais detalhado é o processo de avaliação. No entanto, o que de fato pode-se observar é que não existem critérios pré-estabelecidos para a seleção destes ataques. Sendo assim, muitos dos ataques utilizados nessas metodologias exploram as mesmas características e, portanto, não possibilitam uma avaliação ampla e detalhada das potencialidades dos IDSs. Além disso, este tipo de seleção de ataques torna os experimentos previstos extremamente exaustivos dada a quantidade de ataques a serem realizados.

A metodologia desenvolvida ao longo desta monografia descreve um método de seleção de ataques, a partir do qual o cenário de ataques utilizado no processo de análise dos IDSs foram compostos apenas por ataques que apresentam características únicas entre si. Através desta proposta de seleção de ataques (descrita no capítulo 3), reduziu-se o cenário inicial de ataques (tabela 4) em aproximadamente 50% (tabela 5). Isto significa que em relação a abordagens como [Puketza et al. 1997], [Lippmann et al. 2000], [Alessandri, 200] e [Barber, 2001] esta metodologia apresenta uma forma eficiente e sistemática de seleção de ataques que permite avaliar as potencialidades de detecção presentes nos IDSs.

5.2 Capacidade de detecção

A análise da capacidade de detecção desenvolvida nessa metodologia difere-se das abordagens supracitadas pelo fato de avaliar as potencialidades de detecção existentes nos IDSs ao invés de avaliar a base de assinaturas dessas ferramentas. O desenvolvimento de novas assinaturas ocorre diariamente. Já a implementação de novos recursos (por exemplo, um decodificador HTTP) é um processo mais lento. Portanto, as

metodologias que avaliam os IDSs somente em relação à base de assinaturas têm seus resultados desatualizados rapidamente. Outra contribuição desta metodologia, quanto à análise em questão, refere-se ao fato de que, ao contrário do que é proposto em [Ref], não é necessário o conhecimento da implementação dos IDSs para avaliá-los.

5.3 Análise de escalabilidade

Entre as metodologias referenciadas no capítulo 2, somente [Barber, 2001] apresentou resultados referentes a escalabilidade dos IDSs avaliados. Todavia, a forma como essa análise foi realizada não está documentada. A análise de escalabilidade desenvolvida nessa monografia está baseada em procedimentos simples tais como, reproduzir o tráfego de fundo definido na seção 3.4.2 em paralelo ao tráfego de ataque e, em seguida, analisar os arquivos de *logs* dos sistemas avaliados.

5.4 Análise das taxas de falsos positivos

Algumas metodologias como [Puketza et al. 1997] e [Barber, 2001] não descrevem a forma exata como foram realizados os experimentos de análise das taxas de falsos positivos. Outras abordagens como [Lippmann et al. 2000] utilizam técnicas questionáveis para a realização desta análise conforme descrito no capítulo 2. Já na metodologia descrita nessa monografia a análise de falsos positivos é realizada a partir do tráfego de ataque previamente capturado. Dessa forma, é possível verificar a taxa de falsos positivos em relação a um tráfego composto por ataques. Entretanto, somente esta análise não é suficiente para determinar as taxas de falsos positivos gerados por um IDS, sendo necessário ampliar o escopo dessa análise para avaliar de forma mais precisa o comportamento destes sistemas em relação às taxas de falsos positivos.

5.5 Possibilidade de expansão da metodologia

No que tange a utilização desta metodologia, o usuário pode tanto utilizá-la entendendo que o cenário de avaliação proposto (tabela 5) é suficiente para a realização da avaliação, quanto aumentar o número de ataques que serão submetidos ao processo de seleção. A partir do estudo desses ataques o cenário de avaliação pode ser ampliado.

Os usuários que optarem pela primeira opção devem iniciar o uso desta metodologia pela preparação do ambiente de teste e realização das análises da capacidade de detecção, escalabilidade e taxas de falsos positivos conforme descrito na seção 3.6. Aqueles usuários que desejarem ampliar o escopo de ataques propostos por esta metodologia, devem submeter novos ataques a etapa de seleção, a fim de verificar a necessidade de testar os IDSs contra os mesmos. Uma vez que estes ataques tenham de ser reproduzidos, há um novo cenário de avaliação definido. Portanto, o próximo passo é pesquisar por ferramentas que implementem esses ataques e em seguida realizar a preparação da etapa de geração do tráfego do cenário de avaliação. Tendo os novos tráfegos de ataques devidamente coletados é o momento de montar o ambiente de avaliação e realizar as análises citadas anteriormente.

5.6 Trabalhos futuros

Embora os objetivos propostos nesse trabalho tenham sido atingidos existem diversas melhorias a serem realizadas. Uma dessas melhorias corresponde a ampliação do cenário de avaliação, através da seleção de outros tipos de ataques. Além disso, o desenvolvimento de uma ferramenta para instrumentalizar essa metodologia é algo de extrema importância.

A exemplo de outras áreas da computação, como a escolha de qual o banco de dados mais adequado para uma determinada instituição, a seleção de um IDS está relacionada com as peculiaridades existentes em cada caso. Portanto, o sistema de detecção de intrusão ideal é aquele que melhor atende as necessidades no tocante a segurança. Para que essas necessidades sejam definidas é necessário o conhecimento da política de segurança da instituição.

Uma vez que os objetivos a serem atingidos estejam claramente definidos, é o momento de realizar a avaliação dos IDSs. Embora a metodologia proposta nessa monografia avalie características fundamentais dessas ferramentas, existem outros aspectos tais como: o nível de conhecimento necessário para administrar essas ferramentas, o suporte fornecido a esta solução, as possibilidades de integração com outros mecanismos de segurança, e a capacidade de resposta dos IDSs devem ser considerados no processo de avaliação.

Estudos que avaliem a eficiência dos recursos de integração dos IDSs com outros mecanismos de segurança e a capacidade de resposta desses sistemas, são estudos de fundamental importância, a fim de aprimorar o processo de seleção de um sistema de detecção de intrusão.

Bibliografia

- [Puketza et al. 1997] Puketza, Nicholas; Chung, Mandy; Olsson, Ronald A. and Mukherjee, Biswanath. **A software platform for testing intrusion detection systems**. IEEE Software vol. 14, no. 5, pp. 43 – 51, 1997.
- [Lippmann et al. 1998] Lippmann, Richard P.; Fried, David J.; Graf, Isaac; Haines, Joshua W.; Kendall, Kristopher R. McClung, David; Weber, Dan; Webster, Seth E.; Wyschogrod, Dan; Cunningham, Robert K. and Zissman, Marc. **Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation**. In proceedings of the on DARPA Information Survivability Conference and Exposition (DISCEX) 2000, IEEE Computer Society Press, Los Alamos, CA.
- [Ptacek e Newsham, 1998] Ptacek, Thomas H.; Newsham, Timothy N. **Insertion, Evasion, and deny of service: Eluding network intrusion detection**. [online], 1998. Disponível em http://www.clark.net/pub/roesh/public_html/IDSpaper.pdf.
- [Durst et al. 1999] Durst, Robert; Champion, Terrence; Witten, Brian; Miller, Eric and Spagnuolo, Luigi. **Testing and evaluating computer intrusion detection systems**. Communications of ACM vol. 42, no. 7, pp. 53 – 61, 1999.
- [Hervé, 1999] Hervé, Debar. **Towards a taxonomy of intrusion detection systems**. Computer Networks, vol. 31, pp. 805-822, 1999.
- [Iss, 1999] Internet Security Systems, Inc. **Real Secure**. [online], 1998. Disponível em <http://iss.net/prod/realsecure.pdf>.
- [Kendall, 1999] Kendall, Kristopher. **A database of computer attacks for the evaluation of intrusion detection systems**. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA, 1999.
- [Mutaf, 1999] Mutaf, Pars. **Defending against a deny of service attack on TCP**. Recent Advances in Intrusion Detection (RAID). West Lafayette, Indiana, USA, 1999.
- [Paxson, 1999] Paxson, Vern. **Bro: a system for detecting network intruders in real-time**. Computer networks, vol. 31, pp. 2435 – 2463, 1999.
- [Lippmann et al. 1999] Lippmann, Richard; Haines, David; Fried, David J.; Das, Kumar J.; Korba, Jonathan. **Evaluating intrusion detection systems: The 1999 DARPA off-line intrusion detection evaluation**. Computer Networks, vol. 34, pp. 579 - 595, 2000.
- [Roesch, 1999] Roesch, Martin. **Snort – Lightweight intrusion detection for networks**. USENIX LISA Conference 1999. Seattle, WA.
- [Lippmann et al. 2000] Lippmann, Richard P.; Haines, Joshua W. **Extending the DARPA off-line intrusion detection evaluations**. Submitted for consideration by DISCEX-II. 2000.
- [Korba, 2000] Korba, Jonathan. **Windows NT attacks for the evaluation of intrusion detection systems**. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA, 2000.

- [Das, 2000] Das, Kumar J. **Attack development for intrusion detection**. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA, 2000.
- [Northcutt, 2000] Northcutt, Stephen. **Como detectar invasão em rede – uma guia para analistas**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2000.
- [Alessandri, 2000] Alessandri, Dominique. **Using rule-based activity descriptions to evaluate intrusion-detection systems**. Recent Advances in Intrusion Detection (RAID). ONERA: Toulouse, France, 2000.
- [McHugh, 2000] McHugh, John. **Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection systems evaluations as performed by Lincon labotatory**. Transactions on Information and System Security. ACM vol. 3, pp. 262 – 294, 2000.
- [Whisker, 2000] Rain Forest Puppy. **Anti-IDS tools and tactics**. [online], 2000. Disponível em <http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html>.
- [Barber, 2001] Barber, Richard. **The evolution of intrusion detection systems – the next step**. Computer & Security, vol. 20, pp. 145, 2001.
- [Campello, 2001] Campello, Rafael. **Sistemas de detecção de intrusão**. 19º Simpósio Brasileiro de Redes de Computadores. UFSC: Florianópolis, 2001.
- [Nfr, 2001] Network Flight Recorder, Inc. **Overview of NFR intrusion detection systems**. [online], 1998. Disponível em <http://www.nfr.com/>.
- [Cert, 2002] Cert Coordination Center. **Cert security improvement modules**. [online], 2002. Disponível em <http://www.cert.org/security-improvement/>
- [Enterasys, 2002] Enterasys Networks, Inc. **Sistemas de detecção de intrusão**. [online], 2002. Disponível em <http://www.enterasys.com/ids/>.
- [Linuxsecurity, 2002] Linuxsecurity. **Linux Security – The community's center for security**. [online], 2002. Disponível em <http://www.linuxsecurity.com/>.
- [Nmap, 2002] Nmap. **Network mapper**. [online], 2002. Disponível em <http://www.insecure.org/nmap/index.html>
- [Securityfocus, 2002] SecurityFocus. **SecurityFocus corporate site**. [online], 2002. Disponível em <http://www.securityfocus.com>.
- [Tcpdump, 2002] Tcpdump. **TCPDUMP public repository**. [online], 2002. Disponível em <http://tcpdump.org>.
- [Tcpreplay, 2002] Tcpreplay. **Tool to replay captured network traffic**. [online], 2002. Disponível em <http://sourceforge.net/projects/tcpreplay/>.
- [Tedesco, 2002] Tedesco, Gianni. **Firestorm network intrusion detection system**. [online], 2002. Disponível em <http://www.scaramanga.co.uk/>.