UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIAS, GERÊNCIA E SEGURANÇA
DE REDES DE COMPUTADORES

THYAGO DOS SANTOS MEDEIROS

Proposta de uma Metodologia para Geração de dados para avaliação das ferramentas de detecção de intrusão

Trabalho de Conclusão apresentado como requisito parcial para a obtenção do grau de Especialista

Prof. Dr. Luciano Paschoal Gaspary
Orientador

Prof. Dr. Sérgio Luis Cechin
Prof. Dr. Luciano Paschoal Gaspary
Coordenadores do Curso

Porto Alegre, dezembro de 2008.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pelo dom da vida e sabedoria adquiridas durante estes dois anos e por estar chegando à conclusão de mais um conceituado curso.
As pessoas, que Deus me concedeu para estar ao meu lado nestes momentos e que foram uma benção e graça Dele, para me acompanhar:
A minha noiva Vivian, companheira e incansável amiga, que me esperava ansiosamente a cada final de sábado pela minha presença e que rezou e continua rezando e torcendo por nossas vitórias e conquistas.
A minha família, como um todo que rezou muito por mim, me incentivando e me cativando para ir além dos meus limites.
Ao meu orientador Luciano Gaspary, que foi muito atencioso e receptivo aos meus *e-mails* e conversas pessoais.
A Ida Rossi, que respondeu atenciosamente a todos meus *e-mails* e questionamentos e que faz um trabalho dedicado ao revisar normas em diversos trabalhos.
A professora Patricia Maria Seger de Camargo que se propôs a ler meu trabalho, dedicando seu tempo precioso, corrigindo erros de Português.
A amiga Luciana Marques que além de rezar por mim e meus próximos, auxilio-me na constituição das partes da monografia, além de revisar o português.
A todos que de alguma fora me ajudaram a concluir este trabalho o meu muito obrigado e que Deus abençoe e ilumine a todos.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

AFRL	*Air Force Research Laboratory*
ASP	*Active Server Pages*
BEEP	*Blocks Extensible Exchange Protocol*
CAIDA	*Cooperative Association for Internet Data Analysis*
CIDF	*Commom Intrusion Detection Framework*
CISL	*Common Intrusion Specification Language*
DARPA	*Defense Advanced Research Projects Agency*
DIDS	*Distributed Instrusion Detection Instrusion*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
DNS	*Domain Name System*
FTP	*File Transmission Protocol*
HIDS	*Host Intrusion Detection Systems*
HTTP	*Hypertext Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IDES	*Intrusion Detection Expert System*
IDMEF	*Instrusion Detection Message Exchange Protocol*
IDXP	*Intrusion Detection Exchange Protocol*

IP	*Internet Protocol*
IPS	*Intrusion Prevetion System*
MACE	*Malicious Traffic Composition Environment*
MIT	*Massachusetts Institute of Technology*
NLANR	*National Laboratory for Applied Network Research*
NIDS	*Network Intrusion Detection Systems*
NSM	*Network Security Monitor*
OSI	*Open Systems Interconnection*
PMA	*Passive Measurement and Analysis*
P2P	*Peer-to-Peer*
PHP	*Hypertext Preprocessor*
POP	*Post Office Protocol*
RFC	*Request for Comments*
RPC	*Remote Procedure Call*
SANS	*System Administration Networking and Security Institute*
SNMP	*Simple Network Management Protocol*
SMTP	*Simple Mail Transfer Protocol*
SYN	S*ynchronize*
TCP	*Transmission Control Protocol*
TFN	*Tribe Flood Network*
UDP	*User Datagram Protocol*
VEP	*Vulnerability Exploitation Programs*
VoIP	*Voice sob IP*

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Os sistemas de detecção de intrusão evoluíram de forma satisfatória nos últimos tempos. A eficiência - medida através de diversos tipos de métricas de análise – desses tem sido um dos principais pontos para sua implantação nas diversas instituições e empresas. A geração de tráfego está sendo estudado por pesquisadores para avaliar os sistemas de detecção de intrusão. No trabalho proposto é realizado um estudo das principais iniciativas para geração de tráfego - que na tentativa de simular um ambiente real - apresentam limitações e pontos positivos.

Posteriormente, utilizando-se destes estudos realizados, chega-se a um modelo que visa atender aos *traces* anômalos e normais tentando chegar a uma aproximação com ambiente real, com objetivo de desenvolver um banco de dados com o maior número de *traces* possível.

Palavras-Chave: intrusão, ameaça, sistemas de detecção de intrusão, geração de tráfego, segurança, avaliação, modelo de geração de tráfego.

Proposal for a Method for Generation of data for evaluation of intrusion detection tools

ABSTRACT

The intrusion detection systems developed satisfactory in the latest years. Its accuracy - measured through various types of metrics analysis - has been one of the main points for their deployment in various institutions and companies. The generation of traffic has being studied by researchers to evaluate the intrusion detection systems. In the proposed work is carried out a study of the major initiatives for generating traffic - which in an attempt to simulate a real environment - has limitations and strengths.

Later on, using these studies is presented a model that seeks to meet the normal and anomalous traces trying to reach an approach with real environment, aiming to develop a database with the largest number of traces possible.

Keywords: intrusion, threat, intrusion detection systems, traffic generation, security, evaluation, framework traffic generation.

1 INTRODUÇÃO

A segurança é um fator preocupante e recorrente na condição humana, uma vez que o homem, desde a sua concepção, ainda no ventre materno, recebe amparo e abrigo. Após nascer, necessariamente, adentra em um ambiente desconhecido, já sem a habitual proteção, e busca algo para ampará-lo, de forma a ter alternativas de suprimento de suas novas necessidades.

Fazendo-se uma analogia breve entre o ser humano e o computador, nota-se que quando um ser humano é exposto a ambientes de climas diferentes, o mesmo se expõe às bactérias e variáveis do local, sendo, então, susceptível à contração de um vírus. Isto faz com que o seu sistema de defesa (imunológico) gere um mecanismo de imunização no combate à ação dos microorganismos estranhos ao funcionamento do seu corpo.

Na informática e em seu campo amplo de atuação, dentre sistemas e componentes de *hardware* que se comunicam via redes de computadores e onde há informações transferidas em velocidades impressionantes, os mesmos precisam estar com seus mecanismos de segurança sempre de prontidão para evitar qualquer prejuízo ao seu tráfego.

Os ambientes corporativos que, nos últimos anos apresentam uma complexidade heterogênea de tecnologias e recursos e que buscam sempre a eficiência nas comunicações, podem apresentar várias preocupações e desafios a serem enfrentados.

Cada vez mais, se abrange o assunto de colaboração e unificação das tecnologias e programas computacionais nos ambientes das empresas e, ainda, conforme NAKAMURA (2007), os recursos disponibilizados na rede representam, para muitas empresas e instituições, os próprios negócios da mesma, podendo, assim, estar diretamente relacionados à prosperidade e ao aumento da produtividade e expansão de serviços e produtos.

Assim os conceitos de segurança como a confiabilidade[1], disponibilidade[2] e integridade[3]·, passam a ser essenciais para uma rede e seu bom funcionamento.

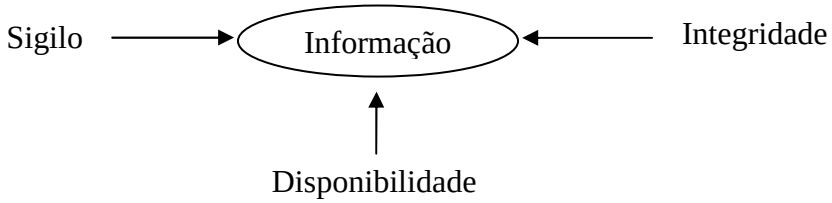

Figura 1.1: As propriedades importantes da Segurança
(NAKAMURA, 2007).

[1] Confiabilidade: ação de um sistema responder a uma dada requisição dentre de condições definidas.
[2] Disponibilidade: Sistema esteja funcionando a um dado instante.
[3] Integridade: não se ter a possibilidade de modificação de informações ou recursos.

Verifica-se que, nos últimos anos, houve uma maior preocupação com a questão da segurança nas empresas e instituições. Segundo NAKAMURA (2007), muitos processos de negócios corporativos não foram, inicialmente, concebidos em um contexto de ambientes distribuídos e muitos foram desenvolvidos sem nenhum enfoque em segurança. Sabe-se, ainda, que a rede *internet*, como um todo, foi concebida sobre os protocolos da suíte TCP/IP, que não apresenta uma estratégia de segurança. Cada vez mais, esta rede é alvo de tráfego malicioso.

Hoje, a segurança da informação e os negócios estão diretamente relacionados. Contudo, a dificuldade em aderir a sua importância ainda é grande. A abrangência da segurança e os aspectos que devem ser levados em consideração são muitos e, conforme NAKAMURA (2007), os principais riscos existentes e considerações a serem feitas são citados abaixo, mostrando o quanto a área é ampla:

- Falta de classificação das informações quanto ao seu valor e a sua confiabilidade;
- Controle de acesso mal definido;
- Dificuldade de controle do administrador sobre todos os sistemas da rede;
- Informações na rede estão sujeitas a capturas e os *e-mails* podem estar sujeitos a ameaças;
- Qualquer conexão a rede interna é um ponto que pode ser usado para ataques;
- A segurança envolve aspectos voltados a negócios, a recursos tecnológicos, humanos, processuais e jurídicos, sendo, portanto, muito complexa de ser tratada;

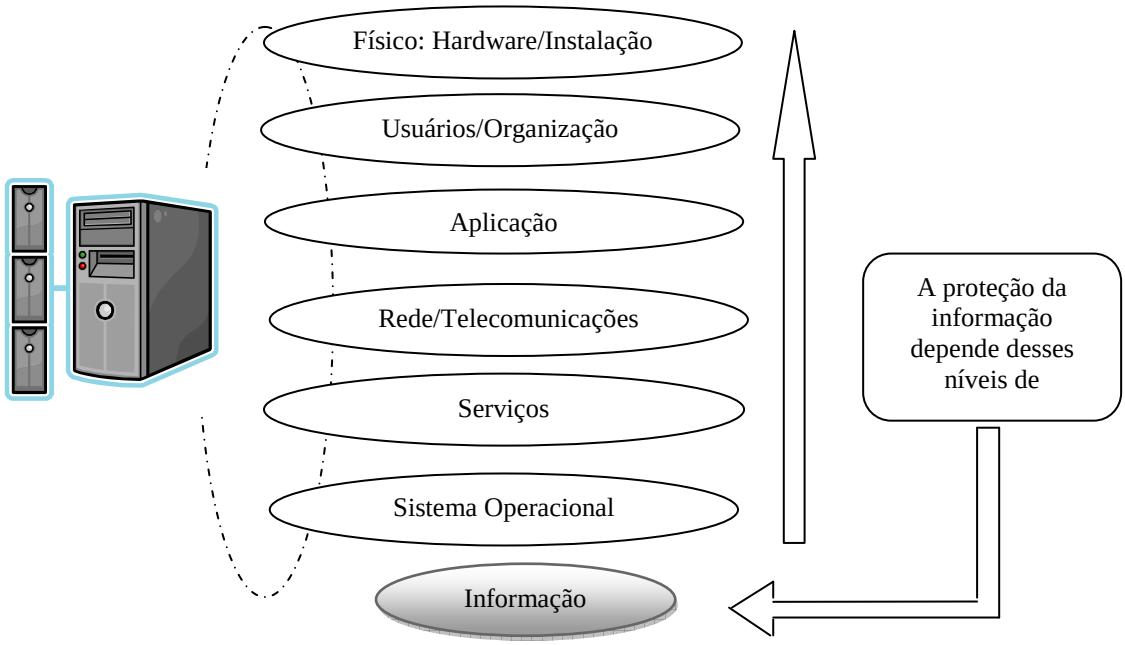

Figura 1.2: As propriedades mais importantes da Segurança (NAKAMURA,2007).

As invasões e ameaças nos últimos tempos, conforme CERTBR (2008), são destacadas na figura abaixo, mostrando as notificações realizadas pelos administradores de redes.

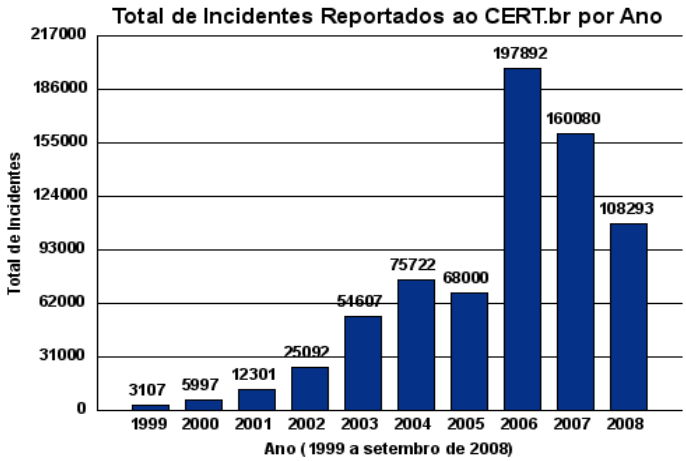

Figura 1.3: Total de incidentes por ano (CERTBR,2008).

No ano de 2006, conforme o gráfico e análise realizada pelo CERTBR, o número de fraudes *onlines* e infecções com *worms* atingiu 197 mil notificações, quase três vezes mais que em 2005. O número de incidentes de segurança, verificado na *internet* brasileira neste período, atingiu quase três vezes mais que o ano anterior. Praticamente no final do primeiro semestre de 2008, já foram notificados mais dados deste gênero que os de todo o ano de 2005. Abaixo, seguem os números de ataques e ameaças ocorridos de julho a setembro deste ano.

Nesta tabela, detalhada por tipos de ameaça, nota-se destaque maior para o número de fraudes realizadas, apresentado também uma porcentagem significativa.

Tabela 1.1: Classificação e percentual de incidentes por ameaças.

Mês	Total	Worm		DoS		Invasões		Ataques a servidores web		Fraude		Outros	
		Total	Percentual (%)	Total	Percentual (%)	Total	Percentual (%)	Total	Percentual (%)	Total	Percentual (%)	Total	Percentual (%)
Julho	**13735**	1806	13	2	0	20	0	232	1	7463	54	171	1
Agosto	**11488**	1859	16	0	0	19	0	530	4	5505	47	192	1
Setembro	**8686**	1619	18	5	0	36	0	262	3	2593	29	185	2
Total	**33909**	5284	15	7	0	75	0	1024	3	15561	45	548	1

Fonte: CERTBR, 2008.

Na tentativa de se utilizar mecanismos de segurança, empresas têm buscado ferramentas que visam detectar de maneira pró-ativa um possível incidente que no futuro possa causar um dano. Nesta monografia dar-se-á destaque a estas ferramentas, começando-se com a especificação dos termos que serão usados neste trabalho.

A intrusão, conforme COLE (2005) é tecnicamente definida como uma tentativa de uma entidade não autorizada de comprometer os recursos que envolvem a autenticidade, integridade e confiabilidade de um recurso.

A história da detecção de intrusão iniciou-se nos 1980 por James Anderson, onde, conforme LAUFER (2003) originou-se a partir da idéia de detecção de irregularidades quando ocorriam, usos impróprios de determinado sistema, e que serviu de base para as ferramentas de detecção de intrusão baseada em estação. Três anos depois a SRI Internacional, mas especificamente Dr. Dorothy Denning, desenvolveu um projeto para analisar trilhas de auditorias provindas dos *mainframes* do governo e ainda ajudou a desenvolver o modelo para detecção de intrusão, o Sistema Especialista em Detecção de Intrusão (IDES), que contribui para desenvolvimento de tecnologias e de ferramentas comerciais no futuro. Em 1988, o projeto *Haystack* da universidade da Califórnia, lançou uma nova versão para detecção de intrusão, produzindo a construção de uma ferramenta chamada Sistema de detecção de Intrusão Distribuído (DIDS), adicionando mais recursos à versão anterior - o IDES - através do monitoramento de máquinas clientes, além dos servidores. Nos anos 90, Todd Herberlein, da Universidade da Califórnia de Davis, introduziu a idéia de sistema de detecção de intrusão de rede, onde em 1990, ele criou o monitor de segurança de rede (NSM).

O desenvolvimento comercial de tecnologias de detecção de intrusão começou no início dos anos 90 mais começou a ganhar popularidade e gerar retorno somente por volta de 1997. Nos últimos tempos, destaca-se o aprimoramento das ferramentas de detecção de intrusão. Porém um dos aspectos que vem sido barreira em sua implantação nas empresas e demais instituições, é quanto à questão da análise da eficiência destas ferramentas. Muitas ainda apresentam erros de classificação que faz com que alertas falsos sejam emitidas. Testar estas ferramentas em um ambiente de rede real apresenta muitas resistências, visto que muitas empresas não podem parar seu ambiente de produção para fazer eventuais comparações de ferramentas.

A simulação de *internet* através da geração de tráfego, por determinados meios – elaboração de *scripts*, utilização de programas de geração de pacotes e tráfego - é uma das alternativas para os testes dos IDS, de forma que se possa realizar uma comparação da eficiência entre sistemas.

Progressos significativos têm sido feitos para as ferramentas para gerar tráfego. Conforme RUPP (2004) quando há uma avaliação de componentes de rede como sistemas de detecção de intrusão baseado em rede (NIDS), há duas tentativas de aproximação, da qual tem vantagens bem como desvantagens: enquanto capturar e repetir tráfego ativo provido com tráfego real, as características são fixas. Em outra mão, quando se gera *trace* de forma artificial (por exemplo, usando simulador de rede), selecionam-se parâmetros individuais, mas os resultados não mostram o lado de um tráfego atual.

Muitos bancos de dados com um conjunto de *traces* vêm sido criados ao longo dos tempos, mas nenhum ainda tem um conjunto completo de ataques que ocorrem, e são poucos que tem a geração de tráfego normal para uma rede. Tanto *trace* de ataque quanto o normal variam de uma empresa para outra, devida a políticas de segurança que são aplicadas.

1.1 Objetivo

O objetivo deste trabalho é a construção de um modelo para geração de tráfego, que englobe conteúdo de ações normais e de articulações de potenciais ataques. A partir das iniciativas já realizadas na área de geração de tráfego para avaliação da eficiência de

sistemas de detecção de intrusão, serão avaliadas as principais características e peculiaridades de cada tentativa, de forma a construir um modelo objetivo e claro.

1.2 Organização do trabalho

No Capítulo 2, serão mostradas as principais ameaças, planejamentos, potenciais atacantes e incidentes que um tráfego ou um *host* de uma rede pode receber. Nas partes finais, são mostradas as ferramentas atuais utilizadas, suas estruturas e respectivas classificações. No Capítulo 3, são apresentadas as principais iniciativas estudadas para a geração de tráfego, bem como uma análise dos modelos criados. Ao final, será visto o modelo criado. No Capítulo 4, as conclusões e trabalhos futuros são apresentados.

2 FUNDAMENTOS DE DETECÇÃO DE INTRUSÃO

Os fundamentos de detecção de intrusão envolvem ameaças, atacantes e incidentes possíveis em um tráfego de rede, e ferramentas utilizadas em determinado modelo para rede de computadores.

Nas primeiras seções deste capítulo serão mostradas as principais ameaças, planejamentos, potenciais atacantes e incidentes que um tráfego ou um *host* de uma rede pode receber. Nas partes finais serão mostradas as ferramentas atuais utilizadas, suas estruturas e respectivas classificações.

O modelo híbrido de rede adotado neste trabalho foi adaptado a partir dos modelos de referência internacionalmente conhecidos para rede de computadores, como o modelo OSI e TCP/IP. Tanembaum[4], atualizando tais modelos através de tecnologias e padrões recentes estabelece o seguinte modelo:

5	***Camada de aplicação***
4	***Camada de Transporte***
3	***Camada de rede***
2	Camada de Enlace de Dados
1	Camada física

Figura 2.1: Modelo hibrido proposto para esta monografia

(TANEMBAUM, 2004).

Deste modelo serão abordadas as camadas negritadas e seus respectivos protocolos que já foram alvo de ataques e ameaças, nos limites abaixo:

- **Camada de rede:** protocolo ICMP, que conforme TANENBAUM (2004), é considerando como um dos participantes do controle da *internet*, que possui mensagens de notificações para verificar a conectividade entre dois dispositivos de rede; e o IP, protocolo que projetado desde o início com o objetivo de interligar redes, fornece a melhor forma possível para entrega de datagramas da origem para o destino de dispositivos;
- **Camada de transporte:** protocolo TCP, que orientado à conexão, confiável, permite a entrega sem erros de fluxo de *bytes*; e UDP, protocolo sem conexão e não confiável, destinado a aplicações. Nestes protocolos constituem-se ameaças que se utilizam dos mecanismos de controle e transmissão de dados;
- **Camada de aplicação:** protocolo FTP, constituído para transferência de arquivos da *internet*; protocolo SNMP, voltado para comunicação de

[4] TANEMBAUM mostra em seu livro, na página 53, o modelo híbrido considerando o modelo TCP/IP e protocolos afins, bem como recursos mais recentes.

componentes de gerência de redes. Nesta camada se destacam aplicações – programas e sistemas - que são exploradas devido a vulnerabilidades existentes.

2.1 Tipos comuns de ataques e ameaças

Nesta seção serão destacados quem ocasiona os ataques, as principais ameaças associadas às camadas do modelo híbrido proposto e seus respectivos protocolos relacionados.

As ameaças são, normalmente, causadas por pessoas. O termo mais comumente usado para quem ataca um ambiente computacional é *hacker*. Esta generalização, conforme NAKAMURA (2007), apresenta várias ramificações, pois os ataques aos sistemas apresentam focos diferentes e o seu sucesso depende do grau de segurança dos alvos, e da conseqüente capacidade de atacar.

Um primeiro passo, para um ataque é a obtenção de informações sobre o sistema a ser atacado. As motivações para um *hacker* realizar causar uma ameaça são diversas, variando com o tipo de classificação que ele venha a ter, que podem ser semelhantes à seguinte:

- ✓ *script kiddies*: são pessoas inexperientes e novatas, que conseguem ferramentas - normalmente prontas na *internet* - que não possuem conhecimento da conseqüência que causam ao seu alvo;
- ✓ *cyberpunks*: são *hackers* que se dedicam a invasões de sistema por puro divertimento e desafio, sendo eles responsáveis por encontrar novas vulnerabilidades em serviços, sistemas ou protocolos;
- ✓ *insiders*: responsáveis pelos maiores e mais graves incidentes de segurança, são considerados uma nova modalidade de crime organizado;
- ✓ *coders*: são pessoas que compartilham informações sobre conhecimentos adquiridos escrevendo livros e participando de eventos de tecnologia;
- ✓ *white hat*: conhecidos como '*hackers* do bem', utilizam seus conhecimentos para descobrir vulnerabilidades nos sistemas e aplicar as correções devidas;
- ✓ *black hat*: conhecidos como *crackers*, este grupo utiliza seus conhecimentos para invadir sistemas e roubar informações secretas das empresas;
- ✓ *gray hat*: pessoas que tem conhecimento de atividades de *hacking* e fazem o papel dos *white hat*;
- ✓ *cyberterroristas*: são *hackers* que realizam ataques com cautela, tendo como objetivo transmitir uma mensagem para derrubar a infra-estrutura de comunicações ou para obter informações que podem comprometer a segurança de uma rede. Agem de três formas: (1) ataque semântico, que é 'pixação' de sites; (2) ataques sofisticados, como o de negação de serviços (*Distributed Denial-of-service*); (3) invasões de sistemas com o objetivo de obter informações privativas;

Os ataques a uma rede podem ser iniciados a partir de uma série de fontes diferentes. Segundo TECH FAQ (2008), as ameaças são divididas em grupos em internas ou externas. As ameaças externas são subdivididas em: ameaças desestruturadas e ameaças estruturadas, as quais são especificadas a seguir:

- **ameaças externas***:* são realizadas por indivíduos sem a assistência de colaboradores internos ou empreiteiros, sendo executadas por pessoas experientes com ou sem experiência. As ameaças externas são normalmente realizadas através de um plano pré-definido e com a utilização de técnicas de digitalização e recolhimento de informações. Pode-se, por conseguinte, detectar um ataque externo pelo exame de *logs* do *firewall*. As ameaças externas ainda podem ser:
    - o **estruturadas:** quando normalmente, são lançadas a partir da rede onde se tem um real dano premeditado e futuros prejuízos que se pode causar.

Possíveis motivos para estes ataques incluem ganância, política, terrorismo, racismo, dentre outros. Os atacantes são altamente qualificados na concepção da rede em questão, desenvolvendo métodos para evitar medidas de segurança e sistemas de detecção de intrusão. Eles têm competências necessárias para desenvolver novas técnicas e atacam realizando modificações de informações através de ferramentas de *hacking*;

- **desestruturadas:** são as provenientes de um atacante sem experiência, normalmente a partir de um *script kiddie*.

- **ameaças internas:** são ataques originários dos próprios empregados da empresa, que estão insatisfeitos dentro da organização. Os atacantes têm alguma forma de acesso aos sistemas da empresa e, geralmente, tentam ocultar seus ataques.

Uns dos principais pontos explorados nos ataques são por meio de técnicas que podem utilizar-se da engenharia social ou de invasões técnicas que exploram deficiências na concepção, implementação, configuração e no gerenciamento dos serviços e sistemas.

O detalhamento das ameaças fica bem destacado na figura 2.2, onde se pode observar, a estrutura de um incidente, que abrange as etapas da constituição de um ataque a partir de um atacante que atinge um objetivo final, desenvolvendo-se assim as seguintes etapas:

(1) um incidente é iniciado por um atacante;
(2) início do ataque: o atacante utiliza uma ferramenta para realizar o ataque;
(3) a partir da ferramenta verifica-se que tipo de vulnerabilidade é explorado;
(4) início do evento do ataque: uma ação é realizada a partir da vulnerabilidade explorada;
(5) finalização do evento: a ferramenta utilizada leva à identificação do alvo atacado;
(6) finalização do ataque: o ataque ao alvo se gera um resultado não autorizado;
(7) o resultado não autorizado realiza o objetivo do atacante.

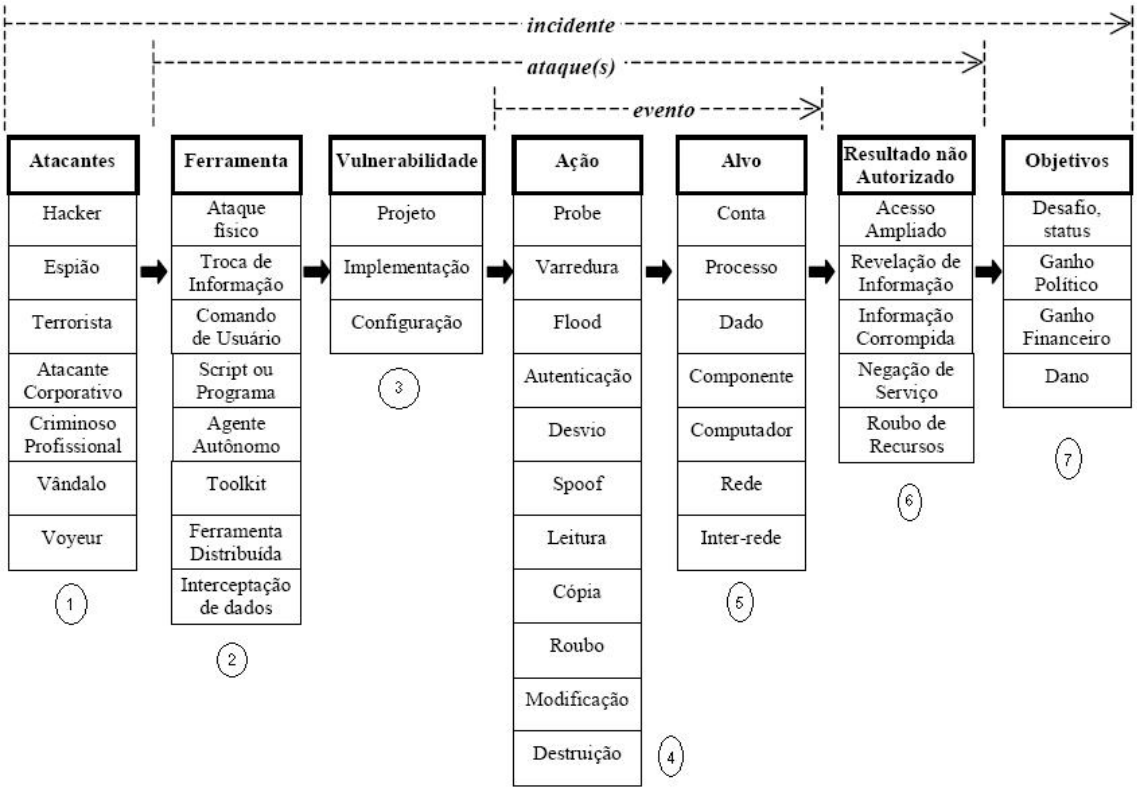

Figura 2.2: A concepção de um incidente e suas estruturas (ARVIDSSON, 2008).

A figura permite, portanto, a demonstração de um incidente desde seu início e geração até o alcance do seu objetivo final, descrevendo as etapas do ataque (etapas 2 até 6) e do evento (etapas 4 e 5).

Passa-se agora a mostrar os principais ataques relacionados às camadas de rede, transporte e aplicação, e respectivos protocolos, que são explorados através das características de sua concepção causando algum tipo de dano a uma comunicação estabelecida.

2.1.1 Ataques e ameaças a protocolos de nível de rede

As ameaças aos protocolos de nível de rede ocorrem, normalmente, a partir dos ataques de negação de serviço (*DoS*) ou ataque de negação de serviço distribuído (DDoS). Tais tipos de ameaças são confirmadas pelo envio de inúmeras solicitações de resposta a um servidor e/ou rede de computadores sem este ou aquele tenha recursos suficientes para atender a demanda das requisições pedidas.

Segundo NORTHCUTT (2003), o ICMP, para receber os ataques, foi alterado na sua constituição a fim de, através do uso de um canal de comunicação entre dois dispositivos de rede, o atacante, consegue interceptar e realizar ataques destinados à rede ou a um alvo (*host*) específico. Descrevem-se, abaixo, ataques que exploram as características deste protocolo:

- **Ataque Smurf:** o ataque, do tipo DDoS faz com que a vítima (na figura 2.3 estação vítima) utilize o recurso do protocolo ICMP para enviar pacotes ao endereço de *broadcast*. Muitas estações podem "escutar" e responder a um simples pedido de resposta (denonimado *echo request*) para este endereço geral da rede. Assim, esta característica é usada para executar um ataque contra uma estação ou uma rede alvo. Na ilustração abaixo segue um exemplo de uma rede fictícia com endereçamento de sub-rede 192.168.0.0.

O computador malicioso efetua remete o ICMP *request* com um endereço IP de origem alterado, enviando mensagens de resposta para o endereço de *broadcast* da rede intermediária. Os seguintes passos mostram como ocorre o ataque *smurf*:

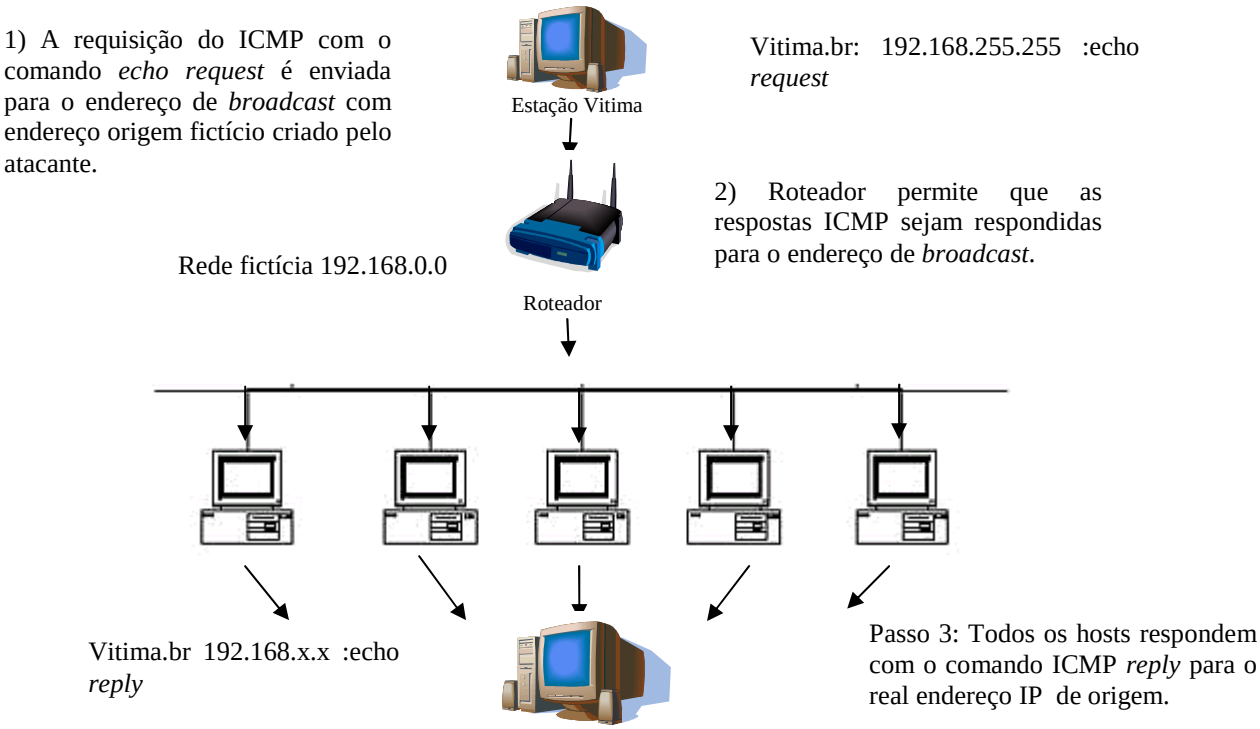

Figura 2.3: Passos do ataque smurf
(NORTHCUTT,2003).

Os seguintes passos são executados durante a realização do ataque *smurf*:

1) o endereço IP origem foi alterado pelo atacante, que escolhe o computador ou rede destino;
2) o roteador local permite a atividade de *broadcast* na rede. Se ele aceita a resposta do ICMP (ICMP *request*), a notificação é enviada para todos os *hosts* da sub-rede no qual o *broadcast* foi enviado;
3) todos os computadores da sub-rede respondem com um ICMP *reply*, pelo através do qual se acredita estar enviando para o remetente correto, ou no exemplo anterior, a estação vítima;
4) assim, a rede ou o computador que está recebendo as respostas pode ter seu desempenho prejudicado, afetando suas atividades e levando a uma possível degradação ou recebendo um ataque de negação de serviço (DoS).

- **Tribe Flood Network:** O ataque de tribo de inundação de rede (TFN), também do tipo DDoS, utilizando-se do protocolo ICMP difere-se do ataque *smurf*, no qual a origem utiliza apenas a comunicação da rede, como um ponto de distribuição do ataque. No TFN o atacante utiliza além da rede, o auxílio de outros computadores distribuídos, conhecidos como *daemon* ou *zombies*.

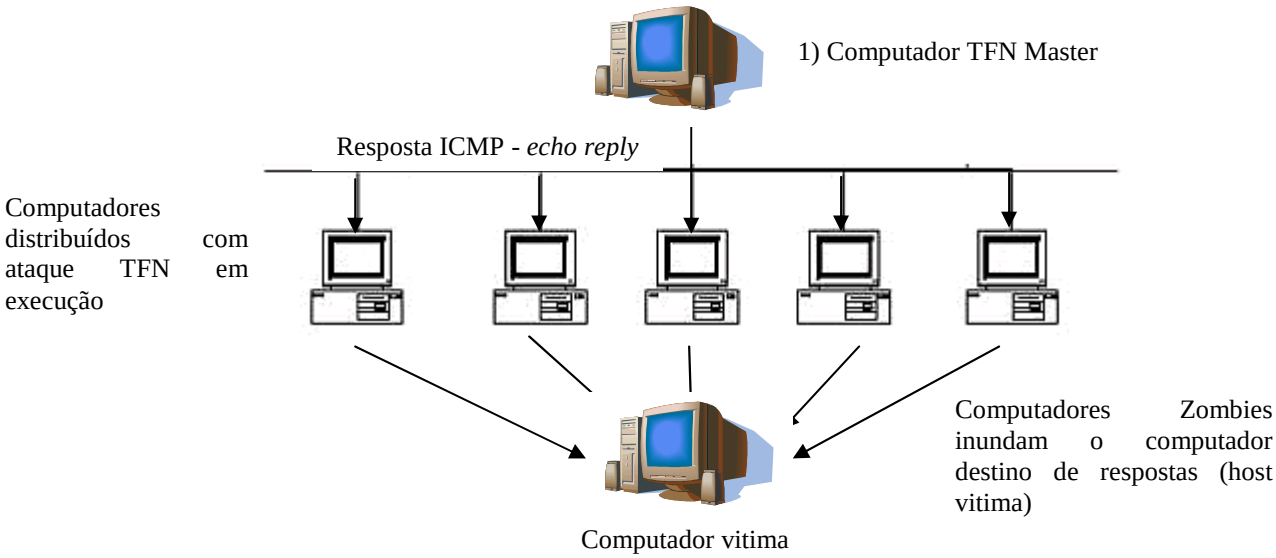

Figura 2.4: Passos do ataque Tribe Flood Network

(NORTHCUTT, 2003).

Os passos descritos abaixo mostram como funciona o ataque TFN:

1) este tipo de ataque requer um servidor denominado TFN Master e computadores alvos, cujo servidor recebe o ataque;
2) o servidor TFN instrue os *hosts* z*ombies* a atacarem o computador vítima, com possibilidade de ocorrência simultânea. A comunicação entre o master e os *hosts zombies* é feito com a utilização do ICMP *echo reply*. Os *hosts* distribuídos podem emitir o computador vítima usando uma inundação de protocolos como o UDP, TCP SYN, um ICMP *echo request* ou um ataque *smurf*;
3) O computador mestre instrui o *zombie* para fazer o envio de comandos com o ICMP *echo reply*. Neste caso é usado um cabeçalho do ICMP, que é um campo denominado número de identificador do *echo reply*, sendo usado para direcionar os *daemons* da ação a ser tomada.

- ***WinFreeze*:** este tipo de ataque de negação de serviço utiliza o recurso do ICMP, de redirecionamento, informando ao *host* - que utiliza um roteador para comunicação de *internet* – que ele deve realizar a adição de rotas adicionais para sua tabela de roteamento. Este ataque ocorre em ambientes com o sistema operacional Microsoft, onde o *host* recebe um ataque de negação de serviço por inundação de ICMP, através das mensagens de redirecionamento.
- **Loki**: O atacante utiliza o protocolo ICMP para realizar um tunelamento através de um canal de transporte a ser estabelecido de forma inesperada ou utilizando um campo de dados do ICMP. O atacante age através de uma aplicação cliente/servidor. Se o servidor está comprometido com o ataque, o mesmo responde ao tráfego enviado para ele, através de uma ação cliente (cliente loki), através de uma notificação. O perigo neste ataque é que o protocolo, aparentemente, inofensivo pode estar sendo usado para fazer alguma mudança sotisficada e potencialmente danosa, visto que o número de respostas (ICMP *echo replys*) é devolvido a um único servidor com o ataque loki sendo executado. Na figura 2.5 segue uma breve ilustração do funcionamento do ataque Loki, onde a aplicação cliente-servidor

realiza a criação de um túnel, utilizando uma parte da informação do protocolo ICMP, nos comandos - ICMP_ECHO e o ICMP_ECHO REPLY – em que o Loki explora o canal estabelecido no tráfego do ICMP_ECHO;

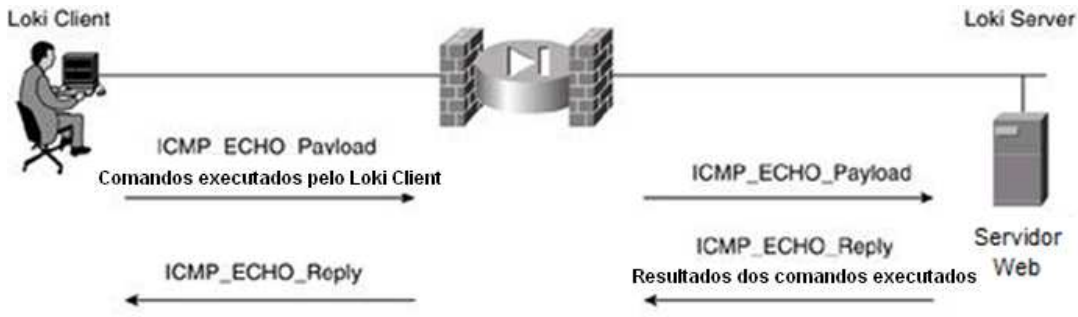

Figura 2.5: Ataque com a aplicação Loki

(TCP/IP SUITE WEAKNESSES,2008).

- **Ping Flood (ICMP Flood):** O utilitário ping é um dos usos mais comuns do ICMP. Ele envia um "Echo Request" (solicitação de eco) para um host e espera até que este host devolva uma mensagem "*echo reply*" (resposta de eco). Como a máquina é obrigada a responder, um atacante simplesmente manda uma série de "*echo request*"[5] para a vítima. Obrigado, então a responder sem parar, o sistema do *host* destino entra em colapso ou fica muito lento. Na figura 2.6 segue um exemplo do funcionamento do ICMP *flooding*, onde a estação vítima recebe uma série de requisições;

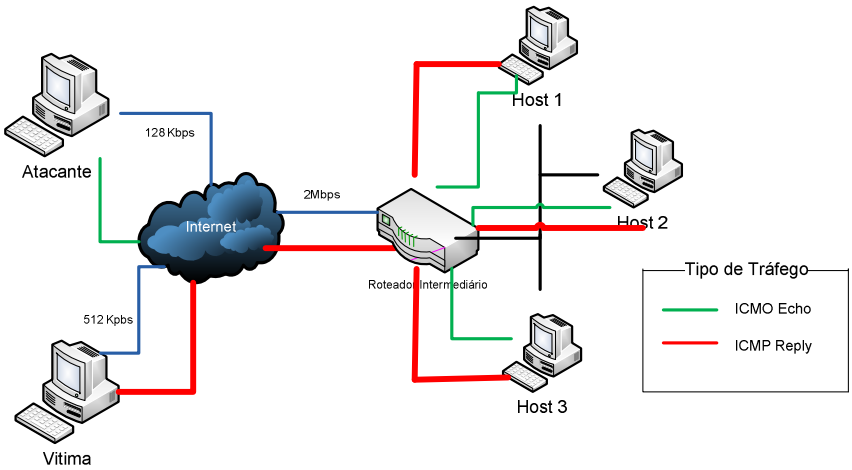

Figura 2.6: Ataque ICMP Flood

(JAVVIN TECHNOLOGIES,2008).

[5] Inundação de “ping” ou ICMP: Quando uma série de requisições do tipo ICMP *request* ocorre simultaneamente a um dispositivo de rede.

- ***Ping of Death*:** No ataque ping da morte, um atacante envia um pacote ICMP solicitando "echo" muito maior do que o tamanho máximo permitido para os pacotes IP da vítima. Como o pacote de requisição é muito maior do que o normal, a vítima não consegue recompor os pacotes e o sistema operacional trava ou é reiniciado;
- ***Firewalking***: técnica implementada em uma ferramenta similar ao *traceroute*[6] que se utiliza do protocolo UDP ou ICMP, normalmente, para obtenção de informação como regras de filtragem de um *firewall*. Esta aplicação, normalmente, usa o campo Time-To-Live (TTL) do ICMP alterado para determinar qual a porta do *firewall* permite uma passagem direta, sem requerer que o destino responda de algum comando.

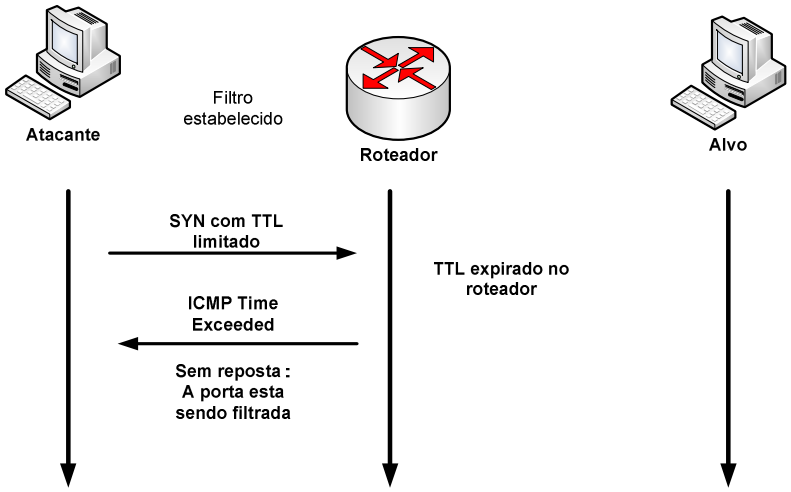

Figura 2.7: Técnica de Firewalking

(ANDERSEN,2003).

Observa-se que o ICMP, protocolo característico por notificações ao nível de rede pode ser explorado através de sua principal funcionalidade que é a emissão de notificações de rede. O próximo protocolo a ser abordado é o IP, que normalmente sofre ataques devido a suas funcionalidades de fragmentação e o reagrupamento dos pacotes.

- **Ataque *Tear drop***: Neste tipo de ataque de negação de serviço, conhecido como "ataque da lágrima", o atacante usa o programa *teardrop* para enviar fragmentos IP que não podem ser reagrupados porque o valor do *offset* do pacote foi adulterado. A grande quantia de fragmentos dispersos provoca uma reinicialização ou congela o sistema da vítima.
- **IP *Spoofing*** : O atacante envia mensagens para um host com um endereço IP imitado (não original), indicando que a mensagem foi enviada por um *host* confiável, com o propósito de conseguir um acesso não autorizado na máquina alvo ou até em outros *hosts* de uma rede. Para preparar um ataque de imitação de IP o *hacker* usa várias técnicas para descobrir o endereço IP de um host confiável e para modificar os cabeçalhos dos pacotes, a fim de que pareçam originar-se do *host* confiável. Na figura 2.6 destaca-se como funciona este tipo de ataque, no qual o objetivo não é recuperar nenhuma informação, mas sim negar o uso dos serviços do

[6] Traceroute: Ferramenta que permite verificar a origem e destino de pacotes em uma rede.

servidor web, que prove dados para os usuários válidos. Verifica-se que o endereço de retorno pode ser adulterado.

Segundo TCP/IP SUITE WEAKNESSES (2008), o hacker através de um acesso a *internet*, com IP origem (168.12.25.5) realiza a conexão ao servidor web (200.198.20.2) que retorna a página para a estação do *hacker*. Com a entrada na rede, o *hacker* com o IP alterado (192.168.0.2) de uma estação de trabalho válida, começa a realizar ataques de negação de serviço a partir do servidor web, onde a estação recebe inúmeras requisições de tentativas de conexões não solicitadas.

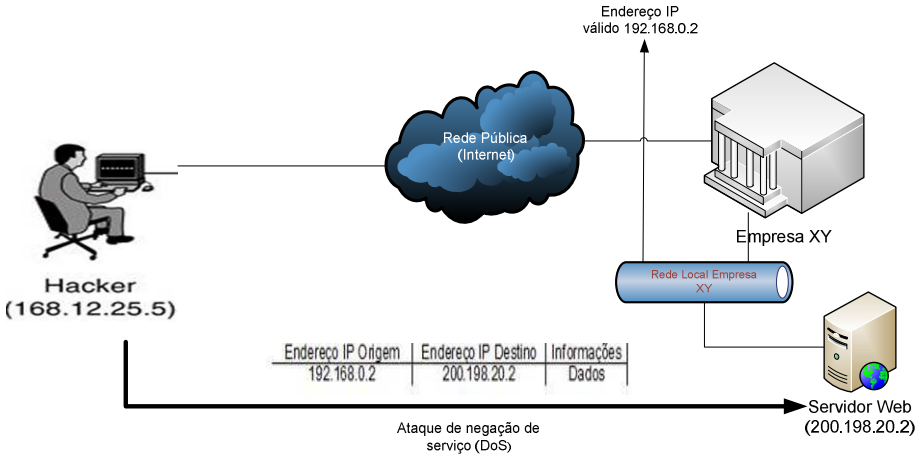

Figura 2.8: Ataque IP Spoofing

(TCP/IP SUITE WEAKNESSES, 2008).

Destacou-se nesta seção os principais ataques aos protocolos da camada de rede e onde realmente se pode realizar a exploração das vulnerabilidades deles. Na próxima seção se dará atenção aos protocolos da camada de transporte.

2.1.2 Ataques e ameaças a protocolo de transporte

Nesta seção serão abordadas as principais ameaças aos protocolos da camada de transporte. Conforme NAKAMURA (2007), um dos problemas da suíte de protocolos TCP/IP é quanto à autenticação entre os *hosts* ser baseada em endereços IP e tendo como outros problemas, os relacionados ao controle da rede e a protocolo de roteamento.

- **Man-in-the-middle (TCP)**: Este tipo de ataque ocorre por meio da interrupção do atacante na comunicação executada entre *hosts*, através do protocolo TCP.

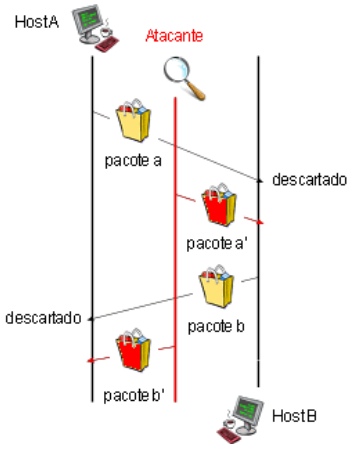

Figura 2.9: Ataque Man-in-the-middle (CREATIVE COMMUNS, 2008).

O atacante permite uma autenticação normal entre dois computadores (no exemplo HostA e HostB) e depois assume o controle da conexão. Existem dois modos de atuação: um é durante o *handshake* - aperto de mão - de três etapas do TCP e o outro é no meio de uma conexão já estabelecida. Segundo CREATIVE COMMUNS (2008), o seqüestro de conexão se aproveita de um "estado dessincronizado" na comunicação. Quando dois *hosts* não estão adequadamente sincronizados eles descartam (ignoram) pacotes um do outro. Neste momento, um atacante pode injetar pacotes forjados que tenham os números seqüenciais corretos e potencialmente modificar ou adicionar comandos na conexão. Isto exige que o atacante esteja no caminho da comunicação, exatamente entre os dois *hosts*, para poder espionar e reproduzir pacotes que estejam sendo enviados.

- **Sessão de seqüestro TCP (TCP *Hijacking)***: Neste ataque, conforme COLE (2005), o atacante interrompe - seqüestra - uma seção entre uma conexão confiável estabelecida entre cliente e servidor. O computador atacante substitui o endereço IP da estação cliente e então o servidor continua a comunicação com a máquina invadida. Com a conexão estabelecida ocorrem os seguintes passos:
  - **(a)** O cliente confiável conecta um servidor;
  - **(b)** O atacante toma o controle do cliente confiável;
  - **(c)** O atacante desconecta a conexão entre cliente e servidor;
  - **(d)** O computador atacante substitui o IP do cliente com seu próprio IP e seus números de seqüência;
  - **(e)** O atacante continua a comunicação com o servidor da rede (o servidor acredita estar fazendo comunicação com o cliente correto);
- **Syn Ataque TCP**: Este é um ataque do tipo negação de serviço. Este ataque ocorre quando um serviço fica tão sobrecarregado que não é mais capaz de responder ou atuar.

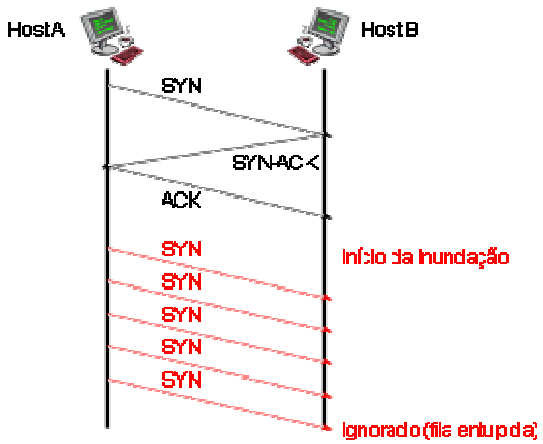

Figura 2.10: Funcionamento do ataque de negação de serviço - SYN TCP

(CREATIVE COMMUNS, 2008).

Após uma conexão efetivada, o protocolo TCP usa um método de "aperto de mão" (*handshake*) de três etapas. Assim ocorrem as seguintes etapas:

- Primeiro um host A envia um sinal SYN (pedido de sincronização) para o host B. O host B precisa controlar esta e outras conexões parcialmente abertas numa "lista de escuta";

- Depois de enviar o sinal SYN, o host A espera por uma resposta de aceitação de sincronização (SYN-ACK) do host B e, quando receber esta resposta, encerra o *handshake* enviando um (ACK) para o host B.
- O atacante então envia, em um curto intervalo de tempo, um grande volume de sinais de sincronização para o host B que não consegue responder com um ACK as respostas SYN-ACK recebidas, sendo que a fila de solicitações de sincronização do host B fica saturada de pedidos, e o host destino não consegue atender a todos os pedidos em tempo hábil e alguns acabam sendo negados.

- **Port Scanning:** As ferramentas de *port scanners* são utilizadas para obtenção de informações referentes a serviços acessíveis por meio das portas TCP e UDP. Normalmente esses ataques são avaliados pelo atacante a partir de ferramentas. Conforme NAKAMURA (2007) estes ataques são caracterizados conforme abaixo:
  (a) **TCP connect() :** É o tipo mais básico de *scanning* TCP. A system *call connect* () é utilizada para abrir uma conexão nas portas do alvo. Neste método não é necessário nenhum privilégio especial;
  (b) TCP SYN (half open): esse método não abre uma conexão TCP completa. Um pacote SYN é enviando como se ele fosse abrir uma conexão real. Caso um pacote SYN-ACK seja recebido, a porta está aberta, enquanto um RST(reset) é enviado, no entanto indicando que a porta está fechada;
  (c) FIN (modo stealth): o FIN no modo *stealth* (modo secreto) elimina a possibilidade de detecção do pacote SYN, sendo que as portas enviam um pacote RST como respostas ao pacote FIN, enquanto portas abertas ignoram tais pacotes;
  (d) UDP: esse método envia um pacote UDP, normalmente de 0 byte, para cada porta do alvo. Caso ele receba como resposta *ICMP port unreachable*, notifica-se que a porta está fechada. A partir disso verifica-se se o alvo está apto para receber um ataque ou não;
  (e) ICMP (ping sweep): esse método envia pacotes ICMP *echo request* para os *hosts*, sendo que para estabelecer a comunicação utiliza-se de um pacote TCP ACK e obtém um pacote RST de volta; assim, o alvo estará pronto para ser atacado;
  (f) Xmas Tree: Portas fechadas enviam um pacote RST como resposta a pacotes FIN, enquanto portas abertas ignoram esses pacotes. São utilizadas as flags do TCP, FIN, URG e PUSH, que são enviadas ao alvo.

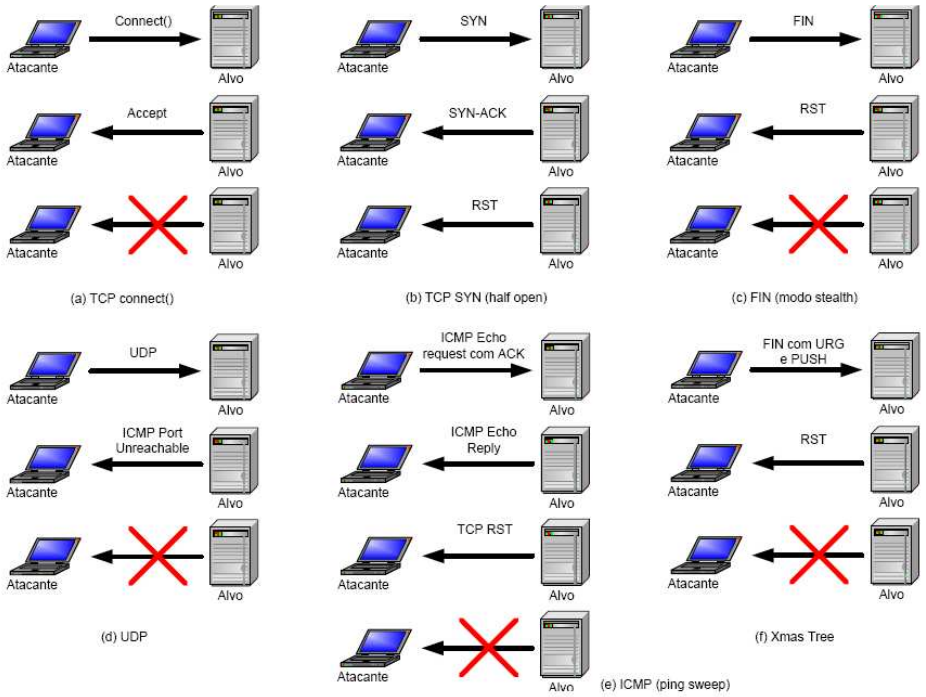

Figura 2.11: Funcionamento dos ataques tipos TCP e UDP Port Scanning (JUNIOR,2006).

- **Ataque do número de sequência do TCP ou Prognóstico do número de sequência TCP:** Neste tipo de ataque o atacante faz o destino acreditar que ele está conectado a um *host* válido, e então sequestra a seção por prognóstico da sequência númerica inicial do protocolo TCP do computador destino. Esta mesma conexão, frequentemente, é usada para iniciar ataques a outros computadores da rede. Também possibilita a construção de pacotes TCP de uma conexão, de modo a injetar tráfego passando-se por um outro equipamento;

- **Ataque de inundação UDP (*Fraggle*):** Um ataque de inundação UDP acontece quando um atacante envia um pacote UDP para qualquer uma das portas do sistema vítima. Quando o alvo recebe o pacote UDP ele tenta descobrir qual é o aplicativo que está aguardando na porta indicada. Quando percebe que não há aplicativo algum ele cria um pacote ICMP de "destino não alcançável" e o envia para o endereço forjado. Se uma quantidade suficientemente grande de pacotes UDP deste tipo forem enviados para diversas portas da vítima, pode deixar o host alvo com o seu desempenho prejudicado.

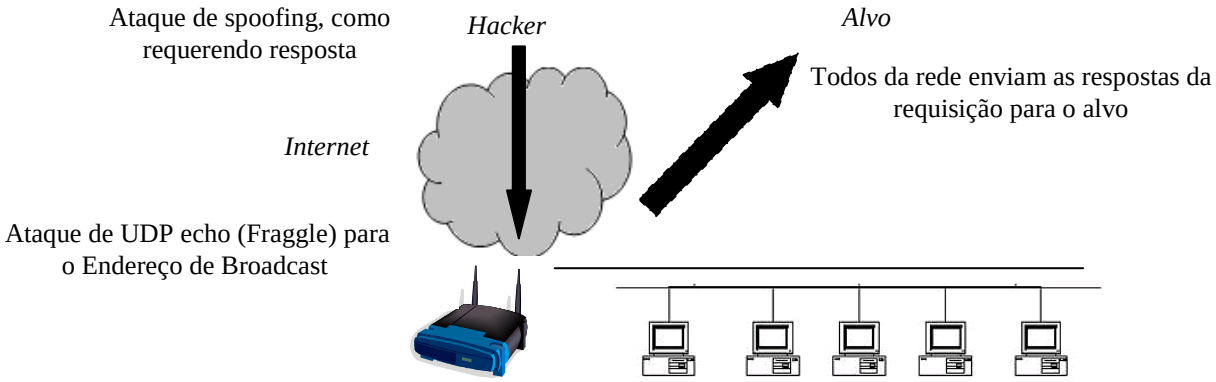

Figura 2.12: Ataque de Fraggle UDP
(COLE,2005).

- **RPC Scan:** ataque que combina vários métodos de *port scanning*. Ele considera todas as portas UDP e TCP abertas e envia comandos NULL SunRPC na tentativa de que eles sejam portas RPC, e façam com que ocorra ataque no tipo inundação;
- **Ataque Mitnick**: Foi o ataque realizado por Mitnick, famoso *hacker* que segundo NAKAMURA (2007), foi preso pelo crime de fraude a computadores e interceptação ilegal de comunicação eletrônica. Ele elaborou um ataque contra Shimomura, um famoso Analista de sistemas. Este ataque é usado como exemplo clássico, além de envolver o uso de diferentes técnicas como *IP Spoofing*, negação de serviço e prognóstico do número de seqüência.

Nesta seção foram citadas as principais ameaças em nível de camada de transporte. Na seguinte serão destacadas as ameaças em nível de aplicação.

2.1.3 Ataques e ameaças a protocolos de nível de aplicação

As ameaças citadas nesta seção serão referentes à camada de aplicação, mostrando os principais protocolos usados para causar ameaças, bem como as aplicações atingidas. Basicamente, estes ataques resumem-se na exploração de vulnerabilidades em aplicações, serviços e protocolos. Abaixo, seguem citadas as principais ameaças:

- **Vírus:** Conforme COLE (2005), esses códigos maliciosos são planejados para causar um dano, interromper funções de computadores e trazer algum tipo de prejuízo ao tráfego de redes. Esses códigos podem ser móveis, como um *applet* Java ou em um ambiente de código *Active*X. Eles também podem ser anexados junto a um código legitimo e efetuar uma propagação.

  Ainda verifica-se, conforme gráfico abaixo, que o número de vírus conhecidos vem aumentado ao longo dos anos:

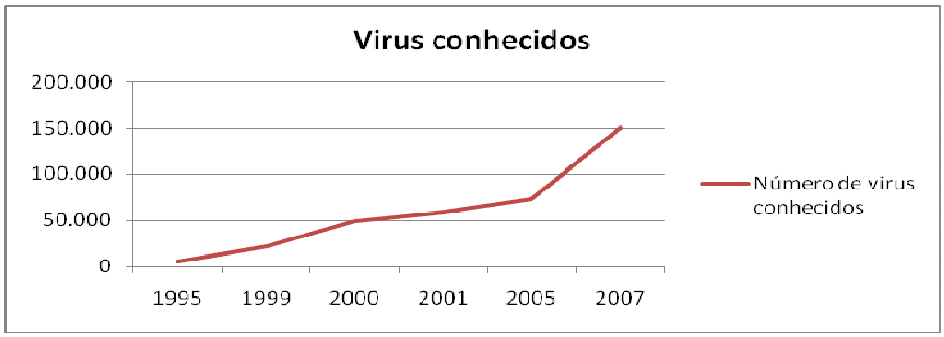

Figura 2.13: Evolução da quantidade de Vírus informático ao longo dos anos (VÍRUS, 2008).

Conforme VÍRUS (2008), o número de vírus conhecidos no ano de 2008 já chega a mais de 530 mil. Os vírus apresentam classificações e categorias diferenciadas conforme a sua forma de propagação:

Tabela 2.1: Tipos de vírus.

Tipo de vírus	**Descrição**
Vírus de Macro	Este vírus é um dos mais comuns encontrados e infecta aplicações como as da fabricante Microsoft (Word e Excell);
Infectores de arquivos	As viroses de arquivos usualmente, atacam arquivos executáveis como de extensão .com e.exe para plataforma Windows. Estes vírus são executados quando os códigos são executados. Outra versão do arquivo com mesmo nome com extensão *.com e *.exe é criada, e quando é aberta o vírus se executa.
Infectores de sistema ou registro de boot	Estes vírus são anexados junto ao registro mestre de boot no disco rígido ou em mídias de disquetes. Quando o sistema realiza o processo de inicialização do sistema no disco rígido, ele primeiro verifica o setor de boot e carrega os vírus para a memória RAM, onde ele se propagam para outros discos e computadores.
Vírus Polimórficos	Estes vírus ocultam-se através de vários ciclos de criptografia e descriptografia. Eles empregam-se em variados tipos de estruturas de criptografia, requerendo diferentes rotinas de descriptografia. Na prática, o vírus criptografado é associado a um processo de mutação, sendo inicialmente descriptografado. O vírus infecta uma área de código. O processo de mutação desenvolve uma nova rotina de descriptografia e o vírus executa a criptografia deste processo, fazendo a cópia do vírus para o algoritmo correspondente para a realização de uma nova rotina de descriptografia. Este pacote de criptografia ocorra diversas vezes.
Stealth viruses	As viroses secretas tomam ação sobre as funções dos sistemas para se esconder. Elas fazem isto para comprometer as ações dos antivírus relatando que há uma área sem infecção onde, na verdade há. Esses vírus podem aumentar o tamanho de arquivos ou mudar informações.
Bombas Lógicas	É um código malicioso que é adicionado para uma aplicação e é engatilhado por uma ocorrência específica, a exemplo uma condição lógica, tal como o tempo ou data, dentre outros.
Worms	Do português Verme, são programas que têm a condição de se replicar para uma rede de computadores. São normalmente, arquivos anexados a um e-mail, os quais abertos ativam-no, realizando a copia dele mesmo para o computador infectado e para o catálogo de endereços do usuário. Este vírus é encaminhado pela *internet* e, sobrecarregando servidores de *e-mail*, pode causar ataques de negação de serviço.
Droppers (Conta-gotas)	É um programa usado para auxiliar a instalação de vírus em outros computadores. Em muitas instâncias este vírus não está infectado com código malicioso e, por isso, não é detectado pela maioria das ferramentas antivírus. Ele pode ter funções para conectar a internet, estabelecendo uma parte da conexão para um posterior uso do *malware* (*backdoor*, vírus).

Fonte: Adaptado de Cole, 2005.

- **Buffer overflow*:*** São ataques que podem executar códigos arbitrários nos sistemas, sendo considerados de alto risco. Este tipo de ataque realiza a exploração de funcionalidades mal implementadas nos programas, nos quais o controle do *buffer* (memória temporária de armazenamento de dados) não é feito de forma adequada. Assim o atacante pode enviar mais informações do que o *buffer* pode suportar, preenchendo o espaço de pilha de memória RAM do computador alvo. Abaixo segue descrito o modo como é realizado este tipo de ataque.

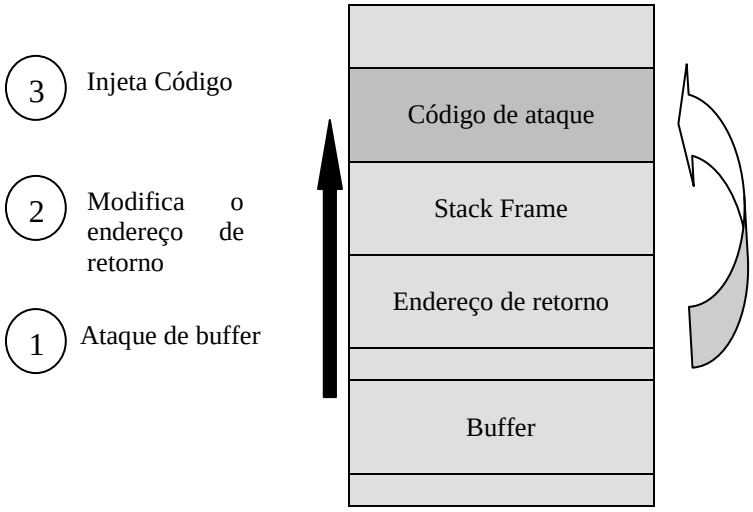

Figura 2.14: Ataque de buffer Overflow
(NAKAMURA,2007).

1) O ataque é feito com a inserção de uma *string* grande em uma rotina que não verifica limites do *buffer*, ultrapassando o limite e sobrescrevendo áreas da memória;
2) O endereço de retorno é sobrescrito por um outro endereço que inclui uma string e aponta para o código do ataque;
3) O código do ataque é injetado na posição de memória que foi sobrescrita no passo 2. A função, então, executa o código do ataque injetado, baseada também no endereço de retorno inserido, fazendo com que o código esteja pronto para ser executado;

- **Ataque replay (repetição):** Ocorre quando o atacante intercepta, salva antigas mensagens e tenta enviar posteriormente de forma anônima para um dos participantes. Um método de dificultar este ataque é através da geração de números e textos randômicos;
- **Ataque de aniversário (*birthday attacks*):** este tipo de ataque criptográfico é feito contra algoritmos de *hash*, que são usados para verificar a integridade de assinaturas digitais. São ataques de tipo força-bruta, onde uma mensagem processada por uma função hash, chamada *message digest,* produz uma saída de tamanho fixo independente do tamanho da saída de um *e-mail*. A função *hash* unicamente caracteriza a mensagem. O ataque de aniversário se refere à

probabilidade de encontrar duas mensagens que geram o mesmo número *hash* quando processado pela função.

- **Adivinhação de senha:** As senhas são muito comumente usadas para autenticação de usuários para sistemas de informação e, a obtenção de senhas pode lvear a um efetivo ataque. Vários artifícios são utilizados para obter-se informações confidenciais, como a utilização de *sniffer* e ataques de engenharia social. Abaixo seguem listadas algumas técnicas para descoberta de senhas:
    - **Força Bruta:** Este tipo de ataque caracteriza-se por tentativas diferentes de obtenção de senhas e algumas palavras. Senhas mais lógicas podem ser aplicadas, como nome de pessoas, título de emprego, *hobbies*, dentre outras; e
    - **Ataque de Dicionário:** neste ataque é usado um dicionário de palavras comuns para a tentativa de obter acesso a um arquivo ou alguma rede;
- **Ataques Matemáticos:** Referem-se a ataques que usam cálculos matemáticos para quebrar senhas ou algoritmos de criptografia. Um bom exemplo de ataque matemático é a utilização de fatoração de algoritmos para quebrar a chave pública do RSA. Um exemplo deste tipo de ataque é o *weak keys*, ataque caracterizado pela exploração de chaves de criptografia encriptadas de maneira fraca. Esta chave é usada quando um específico sistema de criptografia, faz com que ocorra uma maneira desorientada de cifragem;
- ***Dumpster Driving*:** também conhecido como *trashing*, é atividade na qual são verificadas informações não mais usadas, espécie de "lixo" (informações que são apagadas), para encontrar informações confidenciais de uma rede. Um ponto a destacar nesta técnica é que ela não possui uma lei que a regule, haja vista que as informações são coletadas em um repositório não mais utilizado.
    - Engenharia Social: É técnica que explora as pessoas e seus sentimentos, tendo o objetivo de enganar e, assumindo muitas vezes, uma falsa identidade, a propósito de coletar informações que comprometam a segurança da informação.
    - *Packet Sniffing (ou passive eavesdropping*): técnica de "escuta" de pacotes ou técnica de escutar clandestinamente ("escondido"). Consiste na captura de dados diretamente pelo fluxo de pacotes na rede. As informações que podem ser coletadas variam; podem ser estabelecidos filtros pelos programas.
- **FTP proxy (bounce attack scanner)**: este protocolo permite que um servidor seja utilizado como *proxy* entre o cliente e outro endereço qualquer, sendo o servidor utilizado como ponto de acesso a outras conexões. Com isso, caso ele seja utilizado como referência de ataque, o *hacker* pode mascarar sua origem, pois, para a vítima, o ataque origina-se do servidor. Este ataque é utilizado para envio de *e-mails* sem passar pelos *firewalls*. Abaixo, segue uma breve ilustração do ataque usando o protocolo FTP, onde se observa que o servidor atacado é apenas um intermediário entre o atacante e seu alvo;

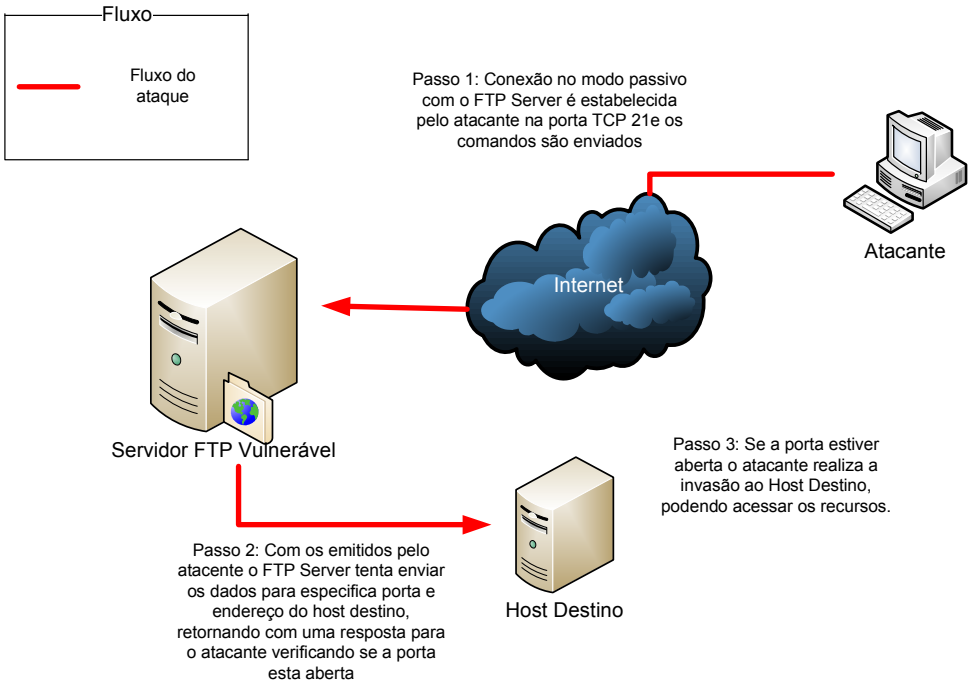

Figura 2.15: Ataque FTP Bounce Scanner
(O'REILLY MEDIA, 2008).

- ***War Dialing*:** ferramenta utilizada pelos *hackers* para fazer a varredura de números de telefones, através de comunicação estabelecida com um local desconhecido, com a utilização de um modem. Conforme NAKAMURA (2007), é uma técnica utilizada em conjunto com a engenharia social, para realizar a descoberta de números telefônicos em que os modems atendem chamadas;
- **Problemas com protocolo SNMP**: Este protocolo pode oferecer informações valiosas que podem ser exploradas, tais como tabela de rotas, tabelas de ARP e conexões UDP e TCP, facilitando o planejamento de ataques. Conforme NAKAMURA (2007), na manipulação de *traps*[7]·, praticada pelo gerenciador - o gerente - existem vulnerabilidades que fazem com que haja condições de se realizar a negação de serviço;
- **Ataques na Web**: Exploração de falhas em servidores web, navegadores de *internet*, *scripts*, fazem com que os *hackers* consigam realizar os seguintes tipos de ataques:
    - *Web defacement* (desfiguração web): modificações de arquivos e páginas, tendo como conseqüência a degradação da imagem das organizações;
    - *Poison Null*: permite que os conteúdos dos diretórios sejam vistos, sendo que em alguns casos pode-se ler e modificar arquivos dos servidores *web*;
    - *Upload Bombing*: afeta sites que oferecem recursos para *upload* de arquivos, tendo como objetivo preencher o conteúdo do disco rígido do servidor com arquivos inúteis;
    - *Web Spoofing*: O usuário é iludido a pensar que está em uma página autêntica. Este tipo de ataque vem sendo muito utilizado contra usuários de i*nternet banking*, para capturar suas informações confidenciais.

[7] *Traps :* Decodificação e processamento de mensagens utilizadas pelo SNMP para comunicação.

- **SQL Injection**: Conforme SANS INSTITUTE (2006), inserções de código, particularmente de códigos SQL, são muito comuns em aplicações web. As inserções são possíveis devido à mistura de dados fornecidos pelo usuário dentro de consultas dinâmicas ou dentro de procedimentos de armazenamento construído de forma simplista. As inserções de código SQL permitem aos atacantes:
    - Criar, ler, atualizar ou apagar qualquer dado arbitrário disponível à aplicação;
    - No pior cenário, comprometer completamente o sistema de banco de dados e os sistemas próximos a ele.

Destacaram-se nesta seção as principais ameaças ao nível de aplicação, caracterizando-se os pontos mais usuais a serem explorados.

2.1.4 Exploração de vulnerabilidade e uso inapropriado de software

Nesta seção serão apresentadas as principais ameaças relacionadas à exploração de vulnerabilidades de sistemas. Conforme NAKAMURA (2007), normalmente são utilizados sistemas que visam realizar testes na rede à procura de falhas de segurança, sejam em protocolos, serviços, aplicativos ou sistemas operacionais. Abaixo seguem destacados:

- **Exploração de vulnerabidades encontradas:** Os riscos existentes em grandes empresas são analisados pelos *scanners* de tráfego, conhecidos também por *sniffers*, através dos quais se podem observar conteúdos de equipamentos como roteadores, *switches*, com a coleta das seguintes informações:
    - Compartilhamento de arquivos não protegidos por senha;
    - Configuração incorreta;
    - Software desatualizado;
    - Configuração de políticas nos navegadores de Internet;

  Neste tipo de exploração também pode ocorrer o ataque de *eavesdropping*, do inglês, “escuta clandestina”, que ocorre através da interceptação do tráfego de uma rede. Esta situação particularmente prevalece quando as redes incluem conexões sem fio e dispositivos de acesso remoto. Assim, o atacante pode obter senhas, números de cartões de crédito e outras informações confidenciais que os usuários enviam pela rede. Conforme COLE (2005), exemplos de várias maneiras de inclusão desse tipo de ataque são citados abaixo:
    - Passivo *eavesdroping*: sem autorização, com o objetivo de monitoração;
    - Ativo *eavesdropping*: proibindo, varrendo ou falsificando informações em um canal de transmissão;

  As vulnerabilidades em softwares podem ser exploradas para obtenção de acessos não autorizados a recursos e dados. Alguns exemplos são citados no relatório da instituição SANS, que apresenta as principais explorações de falhas de segurança ocorridas em sistemas e aplicações. A maioria das organizações confia nesta lista e prioriza esforços na tentativa de tratar tais incidentes. Abaixo segue as principais categorias avaliadas no último relatório:
    - **Aplicação do lado servidor:** Os sistemas operacionais apresentam poucas vulnerabilidades que tratam *worms*. Por exemplo, durante os anos de 2002 até 2005, os vermes do Microsoft Windows como Blaster, Nachi, Sasser e Zotob infectaram um bom número de serviços dos sistemas. O mais recente verme foi direcionado à exploração do antivírus da Symantec usando ataque de *buffer overflow*. No relatório constam falhas nos serviços de sistemas: MAC OS,

Microsoft e Unix, *software* de banco de dados, *software* de backup, *software* antivírus e gerenciamento de servidores;

- **Aplicações lado cliente:** Ao crescimento do número de vulnerabilidades do lado cliente incluem-se os navegadores de *internet*, aplicações de escritório, tocadores de mídia e outras aplicações;
- **Abuso no uso de aplicação:** As aplicações citadas são as de **c**ompartilhamento de arquivos usando a rede P2P e Mensagens Instantâneas. As vulnerabilidades, nesta categoria, ocorrem onde os arquivos são trocados por códigos executáveis com *malware*;
- **Nos dispositivos de rede:** são analisadas fraquezas de configuração dos dispositivos de rede. A tecnologia usada em servidores VoIP e telefones já está sendo atingida com ataques de *phishing*, em que é trocada a numeração de telefone. Nesta forma de *phishing* um *e-mail* orienta a pessoa a telefonar para um número específico, onde uma unidade de resposta de áudio, no outro lado de uma linha de telefone VoIP, fica comprometida, à espera para coletar seu número de conta, identificação pessoal, senha ou outros dados pessoais;
  - **Política de Segurança e Pessoal:** São abordados nesses tópicos dois tipos de ameaças:
    - **Permissões de Usuário e Dispositivos Não Autorizados**: Programas não gerenciados introduzem múltiplos riscos para a corporação. Os melhores esforços para tornar um sistema de informações seguro são fúteis se os dispositivos não autorizados são capazes de conectar à rede;
    - **Ataques de *phishing* e *phishing* Direcionado:** Normalmente são voltados a um alvo (usuários), através de um *e-mail* que inclui informações sobre funcionários ou problemas da organização e faz com que a comunicação pareça genuína aos empregados ou membros da empresa. Os atacantes obtêm informações sobre o nome de usuário e sua senha, e depois invadem para ter acesso a informações confidenciais;
- **Ataques Zero Day:** Esse tipo de ataque ocorre quando há uma falha em código de software;
- **Uso inapropriado de software:** Este tipo de atividade refere-se ao uso de negócios e recursos de uma empresa, inadequadamente, como o *download* de materiais inapropriados da *internet*, condução de pessoas a realizar transações bancárias. É um ataque promovido direto à organização, sendo complementando por ataques físicos, como roubos e assaltos dentre outros. São destacados também configurações mal realizadas por administradores de redes e, às vezes, exploradas por atacantes.

- **Inserção e evasão de um sistema de detecção de intrusão:** As ferramentas de detecção de intrusão que operam no modo de sistemas baseados em rede, no modo passivo, possuem alguns problemas, como a falta de informações suficientes para uma conclusão dos ataques que estão sendo executados. Conforme NAKAMURA (2007), existem algumas classes que exploram o modo de execução passivo desses tipos de ferramentas como:

  a. Inserção: envio de pacotes inválidos à rede, que o IDS aceita, mas o sistema destinatário não. São usados para driblar os sistemas baseados em assinatura, que normalmente usam um conjunto de caracteres para detectar um ataque. Conforme o exemplo abaixo, um pacote recebido somente pelo IDS, com uma assinatura

baseada em um conjunto de caracteres ‘ATTACK’, não detectará um ataque que use a técnica, pois encontrará um conjunto desconhecido.

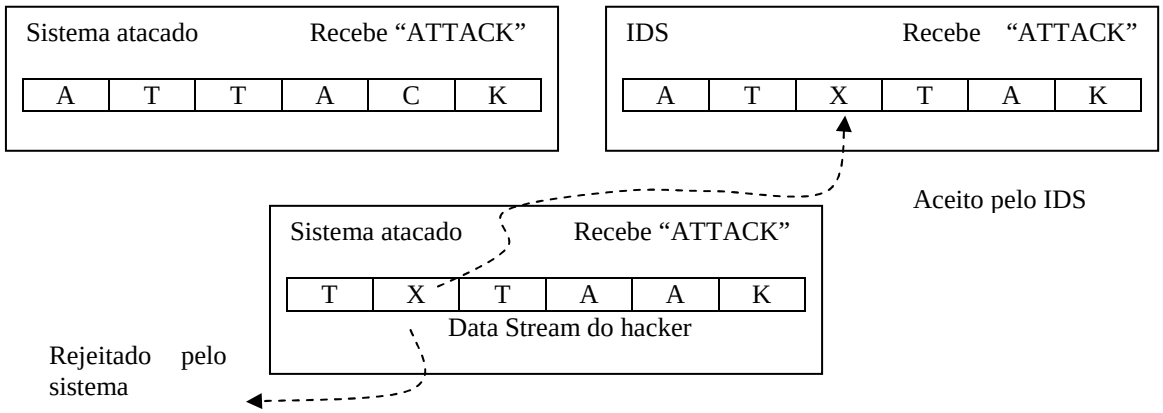

Figura 2.16: Ataque de inserção
(NAKAMURA, 2007).

b. Evasão: exploração de vulnerabilidades entre o IDS e o sistema destinatário, sendo que o IDS não analisa pacotes que chegam ao destinatário. Nesta técnica, o sistema aceita os pacotes que o IDS rejeita, fazendo com que o IDS analise um tráfego diferente do sistema. Verificando-se o exemplo abaixo, nota-se que uma assinatura que detecta o conjunto ‘ATTACK’ não detectará o ataque, visto que analisa ‘ATTCK’.

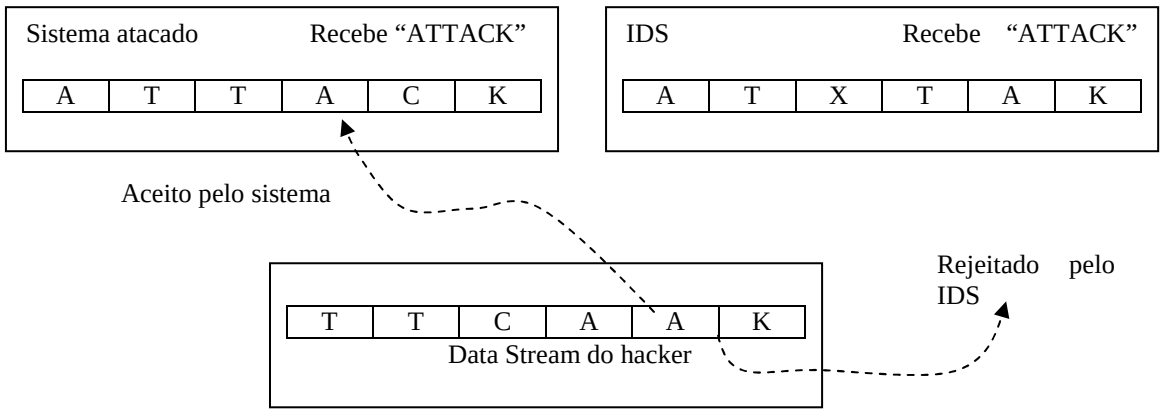

Figura 2.17. Ataque de evasão
(NAKAMURA, 2007).

Nesta seção foram descritos os principais tipos de ameaças que utilizam as técnicas de exploração das falhas de segurança dos sistemas. Na próxima seção serão mostradas as técnicas para detecção e a análise das ameaças mencionadas nas últimas seções.

2.2 Estrutura básica de mecanismos de detecção de intrusão e prevenção de ameaças

Os sistemas de detecção de intrusão são meios para prever problemas na origem da saída de um ataque. Conforme COLE (2005), a retificação do dano feito por um

atacante e sua subseqüente finalidade pode ser muito custosa e o tempo de consumo para detectar a presença do atacante e a remoção da ameaça, pode chegar tardiamente.

Esses sistemas trabalham como um alarme, podendo realizar a detecção com base em algum tipo de conhecimento, como a assinaturas de ataques, ou em desvios de comportamentos.

Modelos e padrões de estrutura desses sistemas foram debatidos por instituições e vários esforços elaborados, apresentaram boas contribuições. Abaixo seguem os principais modelos de estrutura de um sistema de detecção de intrusão:

- **Modelo CIDF:** A estrutura comum de detecção de intrusão foi um dos principais e primeiros esforços, conforme SILVA (2006), elaborada pela funcionária da DARPA, Teresa Lucent, no final da década de 90. Foi uma necessidade de padronização por surgirem na época, ataques sotisficados. Dentre os objetivos desse modelo estava o desenvolvimento de protocolos e modelos de programação. O esforço principal do CIDF Work Group foi definir uma linguagem da camada de aplicação, a qual foi chamada de CISL (Linguagem comum de especificação de intrusão) e um protocolo para codificar e partilhar informações entre os componentes descritos a seguir:

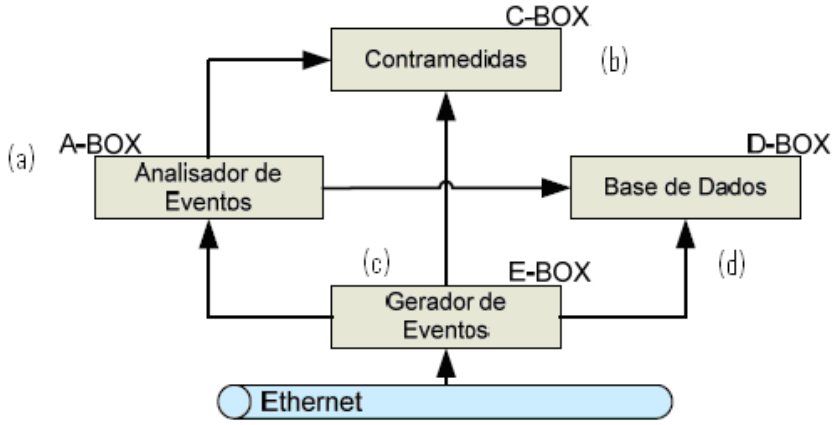

Figura 2.18: Modelo CIDF
(SILVA, 2006).

(a) analisadores de eventos (*A-boxes*): componente responsável pela detecção de intrusão, recebendo dados provenientes dos geradores de eventos e buscando padrões que caracterizem um ataque;
(b) unidades de resposta (C-*boxes*): responsáveis pela tomada de ações em resposta a determinados alertas, como encerrar processos, cortar conexões ou alterar as permissões de algum arquivo;
(c) geradores de eventos (*E-boxes*): elemento responsável pela obtenção de dados gerados fora do IDS e pela posterior padronização do formato desses dados;
(d) bases de dados de eventos (*D-boxes*): elemento responsável pelo armazenamento de eventos reunidos para análise futura ou de relativa importância para o sistema;

- **Modelo do IDWG**: O Grupo de trabalho de detecção de intrusão, elaborado pelo IETF teve como objetivo definir a formatação dos dados e procedimentos para a troca de mensagens entre IDS, resultando nas seguintes definições:
  (a) IDMEF: formato para a troca de mensagens;
  (b) IDXP: formato da camada de transporte de mensagens;

Conforme (SILVA, 2006), o IDWG determinou uma arquitetura para IDS mais detalhada mostrada abaixo:

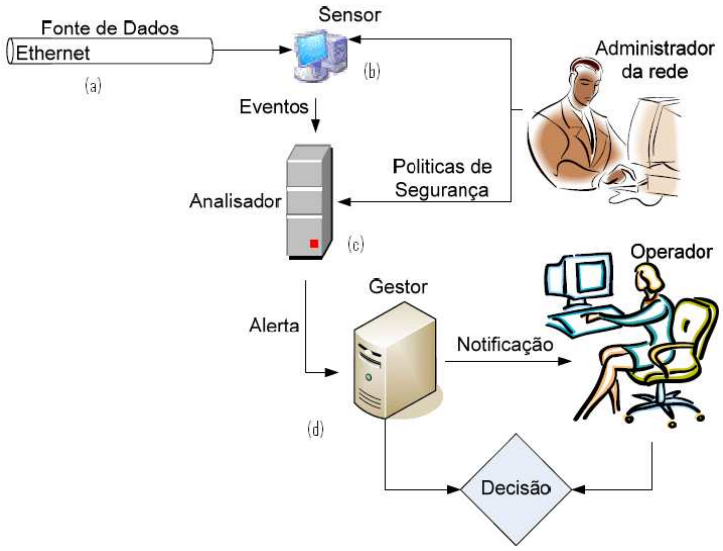

Figura 2.19: Modelo IDWG
(SILVA, 2006).

(a) Fontes de dados: representam os dados que serão avaliados por algum processo de auditoria;

(b) Sensores: componentes instalados nas estações a serem avaliadas para reunir informações e repassá-las posteriormente ao **analisador;**

(c) Analisador: é responsável por receber eventos capturados e passados pelos sensores que podem então gerar algum tipo de alerta, registrar algum evento e processá-los. Os eventos suspeitos são classificados de acordo com a política de segurança - realizado pelo administrador da rede;

(d) Gestor: responsável por receber as notificações de um evento suspeito relacionado a uma política de segurança. Ele também notifica ao **operador** através de um *e-mail* com alerta sonoro, e o mesmo tomará providência, seja automática ou não.

O IDXP possibilita a troca de mensagens IDMEF e de dados no formato binário entre os componentes de um IDS, sendo especificado um protocolo denominado BEEP. O protocolo BEEP é um modelo de aplicação genérica, que funciona com interações assíncronas e orientadas à conexão.

Esse grupo de trabalho parou suas pesquisas e desenvolvimentos deixando uma RFC experimental.

As ferramentas de detecção de intrusão apresentam uma variedade de categorias e técnicas. Por isto a localização dentro de uma rede de computadores é um ponto crucial a ser planejado e programado.

Nesta seção foram descritas as principais estruturas desenvolvidas ao longo do tempo para padronizar os sistemas de detecção de intrusão. Na próxima seção serão destacadas as principais classificações das ferramentas elaboradas para detecção de intrusão.

2.3 Classificações de um sistema de detecção de intrusão.

As diferentes ferramentas existentes atualmente apresentam diferentes tipos de métodos para analisar os dados. Uma abordagem apontada por SILVA (2006) e CAMPELLO (2001), categoriza as diferentes abordagens e se classifica em quatro critérios:

1. Quanto a metodologia de detecção, que expõe a maneira como as ferramentas analisam seu tráfego;
2. Quanto a localização, que mostra a maneira de posicionamento dos sensores de avaliação;
3. Quanto a arquitetura, que mostra como os componentes das ferramentas estão ajustados e configurados;
    - Quanto ao comportamento após a detecção, que é uma resposta das ferramentas frente a detecção realizada, que as vezes não é uma característica do próprio sistema de detecção de intrusão. Conforme encontrado em NAKAMURA (2007), as respostas pode ser: (a) Reconfiguração do *firewall*; (b) Alarme (som); Aviso de SNMP para sistemas de gerenciamento de redes; (c) Envio de *e-mail*, de mensagens para o paper; (d) Geração de *logs*; (e) Gravação das informações sobre o ataque e de evidências do ataque para análise posterior; (f) Execução de um programa capaz de manipular um evento e (g) finalização da conexão estabelecida.

A estrutura poderia ser montada de acordo com a proposta abaixo:

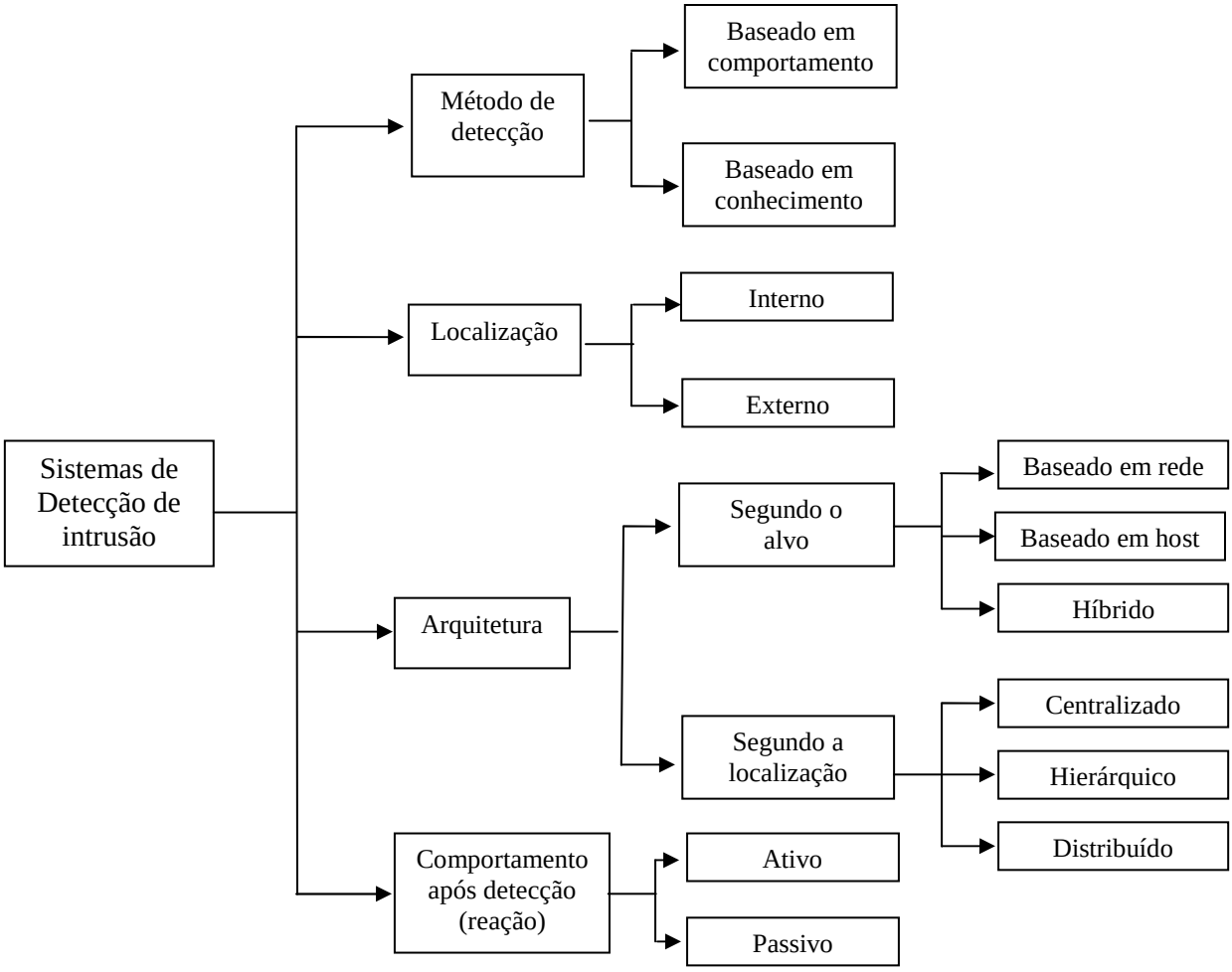

Figura 2.20: Estrutura da classificação das ferramentas de detecção de intrusão adaptado (CAMPELLO ,2001).

Acima foi mostrada a estrutura apresentada para as próximas seções onde ser apresentando as principais características das estruturas e componentes de um sistema de detecção de intrusão.

2.3.1 Quanto à natureza do processo de detecção

Considerando-se que há varias abordagens que podem ser utilizadas em conjunto entre si, a natureza de uma detecção deve considerar o comportamento do tráfego que está sendo analisado para realizar uma posterior comparação com uma base de assinaturas ou por um desvio de comportamento.

As metodologias utilizadas por um IDS para detecção de um ataque, segundo NAKAMURA (2007), são dois:

- Sistema de detecção de intrusão baseada em conhecimento, também conhecida como sistema de detecção por mau-uso e, Sistema de detecção de intrusão baseada em comportamento, também conhecido como sistema de detecção de anomalia.

Conforme COLE (2005), dependendo do tipo de alarme do IDS e o atual cenário, os seguintes tipos de resultados são possíveis, considerando a figura abaixo:

- **Verdadeiros positivos:** São as classificações corretas de comportamento normal, quando ocorre um ataque atual e o IDS respondem o mesmo com um alarme apropria dado;

- **Verdadeiros negativos:** Atividade normal como aguardada pelo administrador do IDS. Quando não acontece um ataque, o IDS não tem razão para ativar um alarme ou uma resposta. É quando um tráfego suspeito é detectado;
- **Falsos positivos:** Tipicamente ocorre quando tem falsos alarmes, onde o IDS analisa um tráfego como legítimos, sendo que um ataque está ocorrendo e a ferramenta não detecta;
- **Falsos negativos:** Quando um potencial ou um real ataque não é detectado pelo IDS;

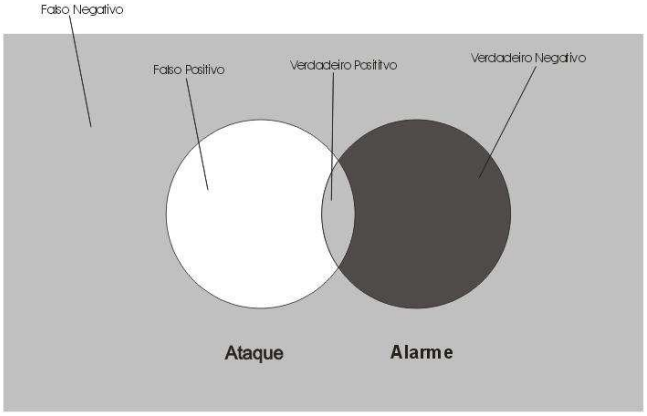

Figura 2.21: Categorização de alarmes em IDS adaptado

(COLE, 2005).

Conforme as possíveis classificações, podemos destacar ainda duas estruturas normalmente utilizada pelas ferramentas de detecção de intrusão.

- **Análise baseada em assinaturas ou conhecimento:** também chamada por detecção por conhecimento ou detecção por mau-uso, pode-se identificar as técnicas que possuem certo conhecimento, como uma base de dados, e dividem as ações possivelmente desempenhadas no sistema em aceitáveis e não aceitáveis. O funcionamento desse tipo de IDS é semelhante ao antivírus, no qual o IDS procura por um padrão ou assinatura de ataque que esteja na base de informação. Estas ferramentas possuem uma base de dados que contém o conhecimento acumulado sobre ataques específicos e vulnerabilidades do sistema. Um comparativo com os pontos fracos e fortes deste tipo de análise conforme podemos destacar abaixo:

Tabela 2.2: Características das ferramentas com análise baseada em conhecimento

Pontos fortes	Pontos Fracos
Poucos números de falsos-positivos: Esta metodologia é mais rápida e não gera tantas classificações errôneas (falso positivos) em comparação com análise baseada em comportamento, pois a verificação 'entende' que tipo de ataque não pode ocorrer, visto que o mesmo possue uma política bem definida e sabe como a rede funciona.	**Não detecção de ataques não conhecidos:** A base de assinaturas precisa ser atualizada pelo fabricante do sistema (assim como ocorre com ferramentas de antivírus);
Facilidade para entender as regras e ter possível adoção de contra-medidas imediatas: As ferramentas de posse de uma base de conhecimento têm a capacidade de se emitir respostas mais rápidas. Além disso, os administradores de rede podem criar regras novas conforme o seu ambiente.	**Dificuldade de manutenção:** Quando se sofre um ataque distribuído coordenado, a análise em tempo real pode ficar comprometida. É necessário manter atualizada, completa e adaptada a base de informação para a organização.
Redução na quantidade de informação tratada	**Possível estudo da base de assinaturas para explorar vulnerabilidades**
Melhor desempenho, mesmo com grandes bases: Não se usa muitos recursos computacionais e uso pouco freqüente de operações de ponto flutuante	**Dificuldade para detecção de abusos de privilégio**: As ações mesmo sendo disparadas por usuários legítimos representam ameaças à organização.

- **Análise baseada em comportamento:** também chamado por detecção por anomalia, e assumi-se que as intrusões podem ser detectadas por meio de desvios de comportamento dos usuários ou dos sistemas.

  Segundo NAKAMURA (2007), a decisão é tomada por meio de uma análise estatística ou heurística, com o intuito de encontrar possíveis mudanças de comportamento, tais como o aumento de tráfego, utilização da CPU, atividade de disco, *logon* de usuários, acesso a disco, dentre outros.

  A abordagem utilizada é de que tudo o que não foi visto é perigoso e,portanto, deve ser evitado. Assim, conforme CAMPELLO (2001), pode-se se destacar os seguintes pontos destas ferramentas:

Tabela 2.3: Características das ferramentas baseada em comportamento

Pontos fortes	Pontos Fracos
Independência de rede: Não se depende da forma de comunicação adotada entre as máquinas (seja segura, criptografada), as tarefas de um IDS baseado em *host* não são afetadas diretamente.	**Classificações errôneas das ferramentas:** Podem-se gerar falsos negativos (quando o ataque não causa mudança significativa no tráfego) e grande número de falsos positivos (problemas no sistema de monitoramento, erro no modo de análise da medição, dentre outros).
Detecção de ataques internos: Apresenta grande facilidade de classificação destes tipos de ataques, podendo detectar atividades não autorizadas que representam abusos de privilégio ou programas.	**Dificuldade de tratar ataques que venham da rede:** Dificilmente consegue-se se tratar ataques que venham da rede
Capacidade de reação: Pode-se ter uma maior eficiência e facilidade, confinar e avaliar danos e recuperar erros.	**Dificuldade de manutenção e instalação, desempenho e dependência de plataforma:** 1) Cada máquina monitorada deve conter ao menos um elemento do sistema de detecção de intrusão; 2)Tarefas de manutenção periódicas são prejudicadas. 3)O desempenho da estação monitorada é diretamente comprometido visto que as vezes precisa-se de uma interação

2.3.2 Quanto a sua localização

Conforme SILVA (2008), a localização das tecnologias para coleta de informações pode servir para identificar e classificar um sistema de detecção de intrusão. Nesta classificação se aborda o posicionamento dos sensores e sua localização dentro da estrutura. Basicamente se constitui de dois elementos:

- Sensores externos: Normalmente estes sensores são usados em ferramentas de detecção por comportamento e se enquadram nesta qualificação todos os componentes de monitoração separados das aplicações e dos objetos para serem monitorados. São programas independentes, facilmente alterados, que sofrem modificações e independe de plataforma ou linguagem de programação. Porém esta facilidade pode abrir brechas para um intruso pode desativar recursos.
- Sensores internos: Nesta categoria se enquadram os sensores que são embarcados juntos as aplicações desenvolvidas e incorporadas no código da aplicação a ser monitorada. Eles apresentam uma desvantagem significativa por serem difíceis de programar e por poderem causar sérias conseqüências no desempenho do sistema.

2.3.3 Quanto a arquitetura - alvo da análise

Os sistemas de detecção de intrusão apresentam variadas funcionalidades e cada tipo de ferramenta apresenta suas próprias funcionalidades especializadas. Uma organização desejando instalar um IDS normalmente precisa ter uma compreensiva revisão de suas necessidades e requisitos de segurança antes de escolher um IDS. Esses sistemas podem ser classificados sobre o alvo da análise nas seguintes categorias, conforme NAKAMURA (2007):

- Detecção de Intrusão baseada em rede (NIDS):
- Detecção de intrusão baseada em hosts (HIDS);
- Solução Hibrida (NIDS e HIDS);
- Solução de prevenção de intrusão baseada em *host*;
- Solução de prevenção de intrusão baseada em rede;

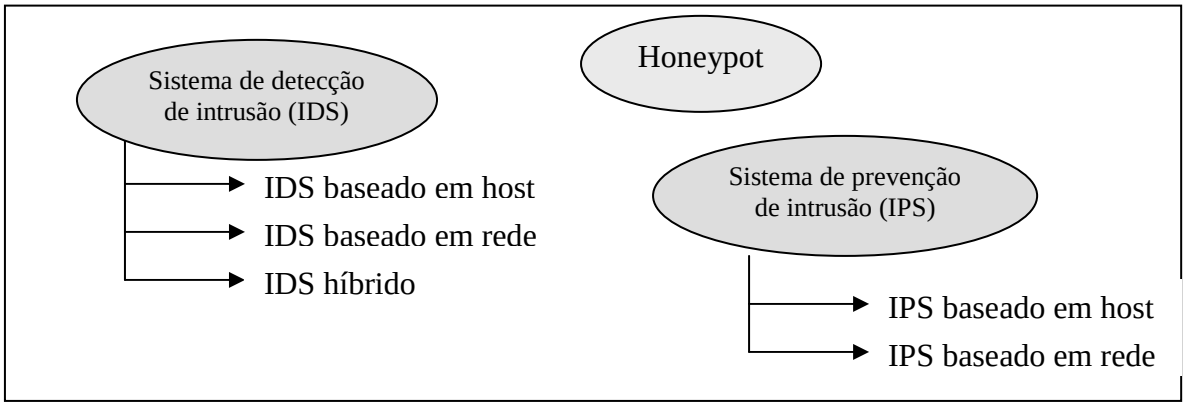

Figura 2.22: Ferramentas de Detecção de Intrusão e seus tipos

(NAKAMURA, 2007).

O processo evolutivo das ferramentas levou ao desenvolvimento do IDS hibrido que aproveita o melhor das características do HIDS e do NIDS. Segundo NAKAMURA (2007), o *honeypot* não é necessariamente um IDS, porém ele pode ser usado para que o administrador de segurança aprenda mais sobre ataques, detectando e analisando e armazenando todos os tipos de ataques. Mais recentemente ainda, a identificação de

pontos fracos levou ao desenvolvimento de sistemas de prevenção de intrusão (IPS), que buscam prevenir ataques.

Mais detalhes dos tipos de ferramentas com as características do alvo de análise serão mostrados abaixo:

- **Detecção de Intrusão baseada em rede (NIDS):** As ferramentas de IDS neste modelo capturam o tráfego de uma rede - usualmente na rede como um todo de largos segmentos - para operações de detecção de intrusão. Muito freqüentemente, estes sistemas trabalham como um *sniffer* de pacotes que analisam informações através do tráfego de entrada e usam métricas especificas para concluir que a rede está comprometida. A detecção é realizada com a captura e análise dos cabeçalhos e conteúdos dos pacotes, que são comparados com padrões ou assinaturas conhecidos. Exemplos de NIDS são o RealSecure, NFR e o Snort. Os NIDS são eficientes para ataques como *port scannig*, IP *spoofing* ou *SYN flooding* e buffer overflow. São ferramentas capazes de detectar a intrusão em tempo real Exemplos destes sistemas são o RealSecure, NFR e o Snort. Segue uma exemplificação de onde normalmente os NIDS são instalados nas empresas:

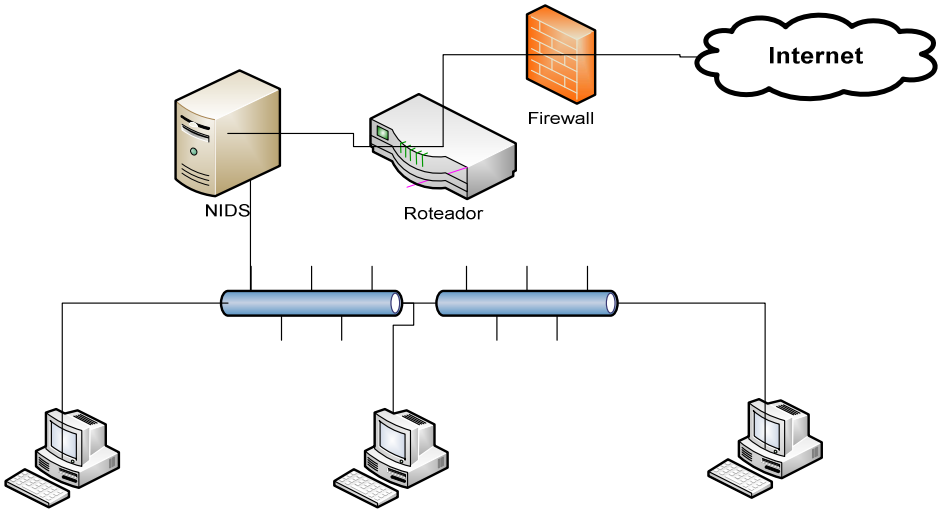

Figura 2.23: Ferramentas de detecção de intrusão baseada em rede (COLE, 2005).

- **Detecção de intrusão baseada em host (HIDS):** sistemas que fazem o monitoramento baseado normalmente em um banco de informações, realizando análise de *logs* ou de agentes de auditoria. Conforme NAKAMURA (2007), o HIDS pode ser capaz de monitorar acessos e alterações em importantes arquivos do sistema, modificações dos privilégios dos usuários, processos do sistema, programas que estão sendo executados, uso do processador, como a detecção do *port scannig*. Além disso, através da análise de *checksum*, a verificação dos arquivos de sistema. Na maioria das vezes o HIDS é considerado ferramentas utilitárias, visto que não consegue se monitorar em tempo real. Alguns exemplos destes sistemas são Abacus Project e o *Tripwire*.

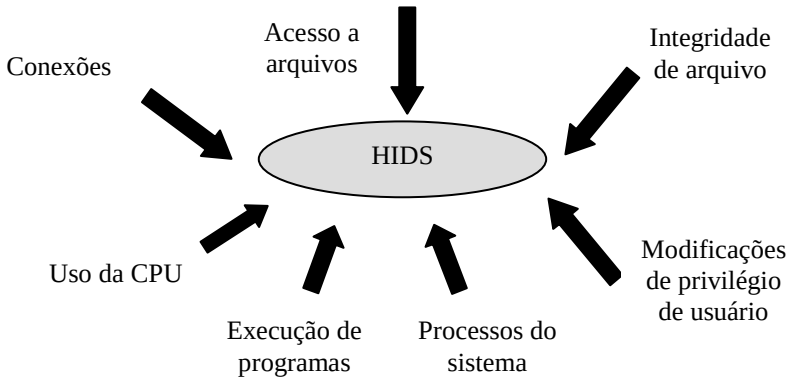

Figura 2.24: Avaliação de sistema de detecção baseado em host

(NAKAMURA, 2007).

- **Sistema de detecção de intrusão híbrido:** São sistemas que se utilizam de duas metodologias de análise para identificar as intrusões, tendo como objetivo combinar os pontos fortes do HIDS e do NIDS. Conforme NAKAMURA (2007), estes sistemas operam com o NIDS, coletando o tráfego de rede, processando os pacotes e detectando e respondendo a ataques. A diferença ela processo as informações como um HIDS. Os dois tipos de detecção de intrusão apresentados se diferenciam bastante, mas se complementam. A medida que os HIDS atuam somente em estações críticas, o NIDS atua analisando todo o tráfego de rede, inclusive aquele para estações que não contem um sistema de detecção rodando. Uma configuração interessante, conforme LAUFER (2003), bastante comum seria a de um sistema de detecção baseado em rede para a rede local e os HIDS rodando nos servidores principais, como é mostrado na figura abaixo.

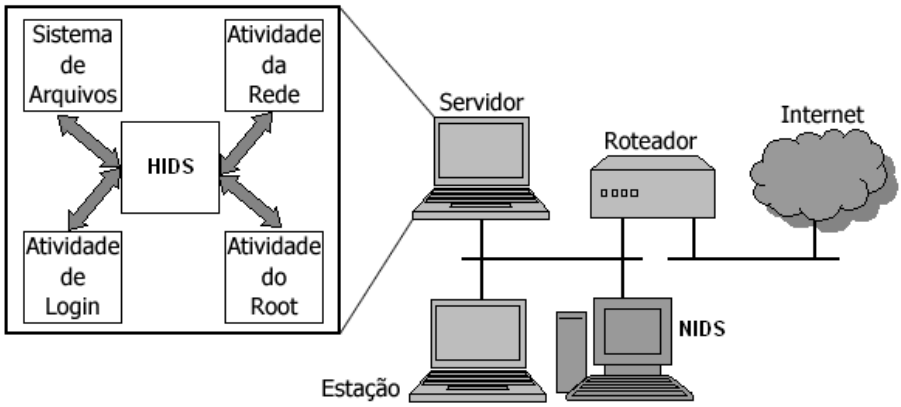

Figura 2.25: Sistema de detecção híbrido

(LAUFER, 2003).

- **Sistema de prevenção de intrusão:** Uma categoria não muito citada, mas que se pode fazer presente em algumas redes e sistemas de monitoramento são os sistemas para prevenir ataques e ameaças. Esses sistemas, conforme NAKAMURA (2007) permite duas abordagens:
    - **IPS baseado em rede:** Neste modelo trata de alguns problemas com sistemas que trabalham como *sniffer*, que realiza a tarefa de capturar e

analisar a comunicação do segmento de rede, possuem alguns problemas, como o fluxo de pacotes fragmentados, não confiáveis e que as vezes chegam fora de ordem. Algumas soluções tomadas para resolver estes problemas de análise são citadas abaixo:

- *Ip de fragmentation* : combinar fragmentos em pacotes;
- *TCP reassembly* (reagrupar): recolocar os segmentos TCP na ordem inicial, eliminando dados duplicados e em *overlapping* (sobreposição de pacotes);
- *Flow tracking*: identificar os fluxo e associá-los com uma sessão única de comunicação;
- Normalização: interpretação e manipulação de representações codificadas e caracteres especiais na reconstrução das mensagens.

Sistemas que se utilizam desta característica são baseados em estados e trabalham no modo *inline*, ou seja, possui um papel semelhante a de um *firewall*, onde todo o tráfego passa por este sistema. Este comportamento permite a estes sistemas enviarem mensagens de '*drop*' das conexões que ele classifica e caracteriza como uma ameaça, fazendo com que as conexões não sejam estabelecidas.

- **IPS baseado em estação**: Este tipo de IPS trabalha, conforme NAKAMURA (2007), com as seguintes características como:
  - **Abordagem heurística**, voltado para detecção via redes neurais;
  - **Abordagem baseada em *sandbox*,** no qual uma área do sistema tem acesso restringindo, alarmando quando uma ação viola os limites dessa área.
  - **Abordagem baseada no *kernel***, onde o acesso ao kernel é controlado pelo IDS, prevenindo a execução de chamadas maliciosas.

  Estes sistemas funcionam integrados ao *kernel* do sistema operacional, inspecionando as chamadas de sistema de acordo com o conjunto de regras definidas rejeitando problemas como *buffer overflow*, mudanças em registros e vírus, cavalos de tróia, *rootkits* e *backdoors*.

Nesta seção foi destacado como é realizado a análise dos sistemas de detecção de intrusão. Na próxima se destacará as principais arquiteturas que as ferramentas se adéquam.

2.3.4 Quanto à arquitetura – localização dos componentes

Nos sistemas de detecção de intrusão outro fator de primordial importância é quanto sua arquitetura, a forma de como o sistema está organizado quanto a forma como seus componentes funcionais estão apresentados e distribuídos. Conforme CAMPELLO (2001), dois fatores influem diretamente na arquitetura: localização e alvo. Abaixo veremos as principais arquiteturas, representadas para os sistemas de detecção de intrusão.

- **Arquitetura centralizada:** Usada na grande maioria das implementações dos sistemas de detecção de intrusão. Estruturas centralizadas compartilham vantagens inegáveis em vários aspectos, seja na operação ou no desenvolvimento dessas ferramentas. A facilidade de instalação e configuração de um IDS centralizado, aliada ao seu desempenho, são exemplos de vantagens operacionais em relação a outras abordagens. No desenvolvimento, sua simplicidade, comparada a complexas técnicas distribuídas, garante vantagens a projetistas e responsáveis pela sua implementação;

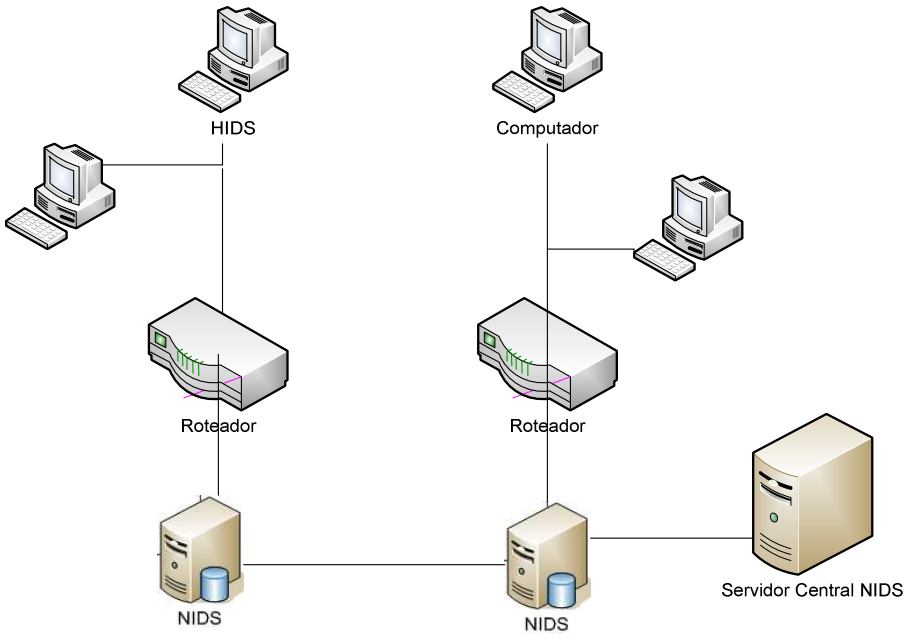

Figura 2.26: Sistema de detecção de arquitetura centralizada (COLE, 2005).

- **Arquitetura distribuída:** Em uma arquitetura distribuída remete-se a módulos independentes cooperando através da troca de mensagens e garantindo uma redundância inerente. Suas vantagens apresentam-se maior robustez, incluindo a facilidade de crescimento modular, possibilitando agregar novos mecanismos ao sistema de acordo com a necessidade, realizando também a distribuição de tarefas, retirando de um único ponto o custo e a responsabilidade de todo o processamento, e a maior abrangência de detecção, com módulos espalhados pelos mais diferentes pontos do sistema.

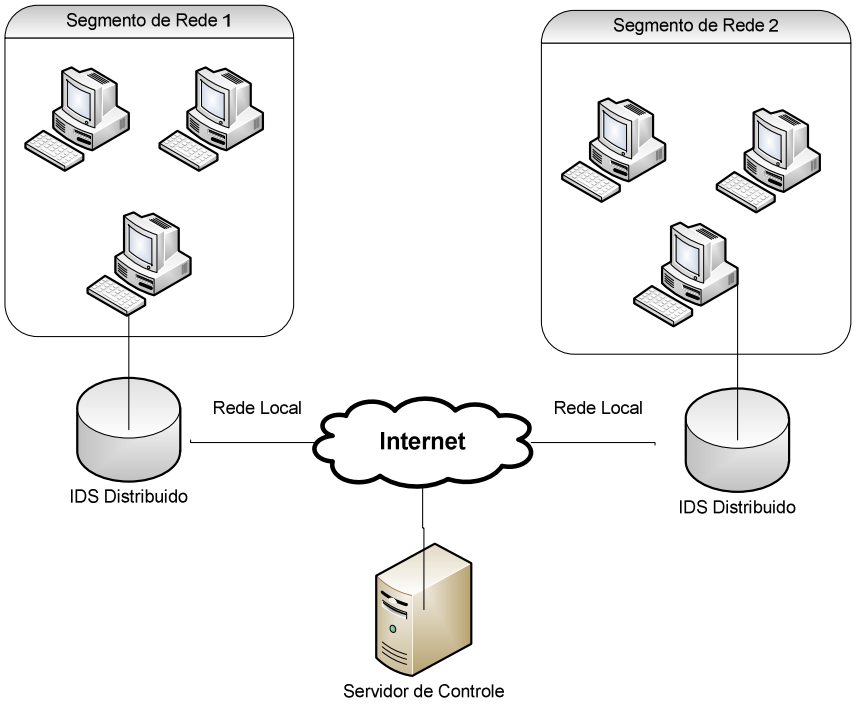

Figura 2.27: Sistemas de detecção de arquitetura distribuída (COLE, 2005).

- **Arquitetura hierárquica:** É uma estrutura cujos componentes participantes apresentam uma forte relação. Há uma interação entre os módulos do sistema sendo regida por relações de subordinação. Esta estrutura apresenta um grande grau de dependência entre os componentes da arquitetura. Se um módulo falhar, todo o sistema fica indisponível.

Nesta seção foi apresentando as principais arquiteturas utilizadas pelos sistemas de detecção de intrusão. Na seção subseqüente irão se mostrar algumas ferramentas existentes.

2.4 Ferramentas existentes

Serão destacadas as principais ferramentas utilizadas segundo as arquiteturas e classificações dadas nas seções deste capítulo. Serão abordadas ferramentas proprietárias, bem como as ferramentas open-source.

2.4.1 Snort

O Snort é uma ferramenta de plataforma livre que se utiliza da arquitetura NIDS. Ele tem a capacidade de analisar todo o conteúdo dos pacotes, comparando-o com um vasto conjunto de regras em tempo real, tornando um IDS eleição.

O Snort foi desenvolvido em linguagem de programação C baseando na biblioteca *libcap* (biblioteca para captura de pacotes de rede). Os pacotes que coincidem com alguma das regras da base podem ser simplesmente descartados, armazenados ou podem gerar algum alerta aos responsáveis pelo sistema. Há ainda a possibilidade de utilizar regras de filtragem durante a coleta dos pacotes (*libpcap*), antes que eles passem pelo analisador.

Esta ferramenta, segundo SILVA (2008), logicamente possui variados componentes que trabalham em conjunto para detectar ataques e gerar as saídas no formato especificado nos componentes:

- **Mecanismo de captura/descodificação:** através da função libpcap começando verificando a interface de rede no modo promíscuo faz as chamadas funções que realizam a descodificação para os protocolos que são analisados;
- **Plugins de pré-processador:** Após a primeira etapa de descodificação, vêm as etapas de plug-in, onde os pacotes sofrem ajustes e reagrupamentos para quando forem enviadas para o mecanismo de detecção, as regras possam ser aplicadas de forma otimizada;
- **Mecanismo de detecção:** Um dos mecanismos mais importantes onde os dados provenientes dos pré-processadores são verificados através de um conjunto de regras. Quando o snort é executado são criadas várias listas em memória com a informação de todas as regras, sendo estas percorridas quando se pretende comparar os dados dos pacotes**;**
- **Plugins de saída:** Depois de realizar todo o processamento e passagem por todas as fases citadas anteriormente (pacotes capturados, descodificados, passarem pelos pré-processadores, serem analisados pelo mecanismo de detecção), ocorrendo de coincidir com uma regra é necessário que o alerta e o conteúdo do pacote sejam guardados para posterior análise.

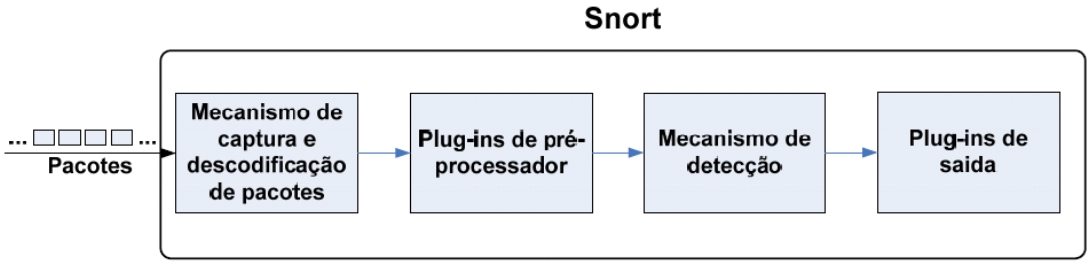

Figura 2.28: Estrutura Snort
(SILVA, 2008).

2.4.2 Bro

Esta ferramenta apresenta bastantes semelhanças com o Snort, tendo uma arquitetura centralizada, de rede e baseada em assinaturas, possuindo como diferencial o formato de sua base de ataques. Neste sentido, toda análise é feita utilizando *scripts*, descritos em linguagem própria. Ele funciona com uma máquina de estados divido em dois componentes: uma máquina de eventos responsável por reduzir um fluxo de pacotes já previamente filtrados e um interpretador de *scripts*, responsável pelo processamento de linguagem de descrição de políticas.

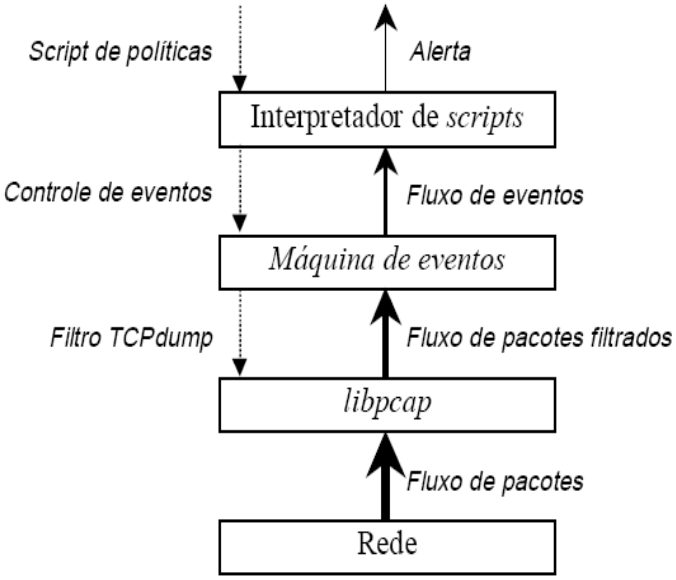

Figura 2.29: Estrutura Bro

(CAMPELLO, 2001).

O Bro, assim como o Snort, também utiliza a biblioteca *libpcap* para fazer a captura de pacotes. Alguns filtros no formato da ferramenta TCPdump são aplicados a essa biblioteca para fazer o primeiro nível de redução de dados, agilizando o trabalho das camadas superiores.

2.4.3 Tripwire

O Tripwire é um sistema de detecção de intrusão baseado em host. O projeto de construção desta ferramenta se iniciou no ano de 2000, pela então empresa Tripwire Inc.

Abaixo segue uma breve estrutura do sistema. Conforme LITT (2003), o *Tripwire*

detecta e reporta mudanças na grande maioria dos sistemas de arquivos. Se a mudança do arquivo é detectada, se determina se a mesma pertence as atividades normais do sistema ou como um comportamento não convencional.

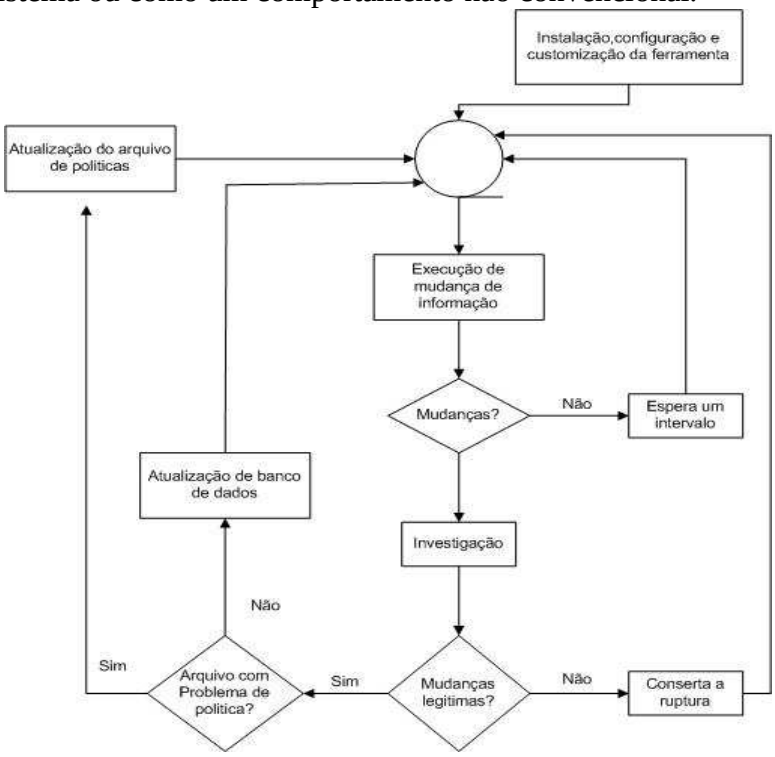

Figura 2.30: Estrutura Tripwire

(LITT, 2003).

Nesta seção foram destacadas algumas ferramentas de detecção de intrusão existentes que permite estudos mais detalhados de seu funcionamento perante as ameaças existentes. Na próxima seção será destacado o estudo do funcionamento de ataques utilizando os *honeypots*.

2.5 Entendendo comportamentos maliciosos usando Honeypots.

Os *honeypots* - potes de mel - têm como principal função serem utilizados para conter uma fonte de aprendizado sobre ataques. Ele não contém dados ou aplicações muito importantes para a organização e seu único propósito são passar-se por um legitimo equipamento da organização para interagir com um *hacker*.

Assim, detalhes de técnica das ameaças podem ser capturados e devidamente estudados. Uma característica interessante nestes sistemas é que não existem falsos positivos, pois todo trafego direcionado ao sistema é real. Os *honeypots* podem ser de diferentes tipos, recebendo uma classificação, conforme NAKAMURA (2007):

- *Sacrifical Lambs* (“cordeiro sacrificado”): são sistemas disponibilizados praticamente com a sua configuração padrão, para serem atacados;
- Facades: emulam serviços ao invés de disponibilizarem servidores reais para serem atacados;
- *Instrumental Systems*: previne que o sistema seja usado para novos ataques e provê muitas informações sobre eles.

O posicionamento destes sistemas também pode influir diretamente no tipo de análise pretendido e resultados. Algumas estratégias seguem abaixo, conforme a sua localização em uma rede:

a) ***Minefield*** **(campo minado)**: o *honeypot* é inserido juntamente com os servidores reais de uma DMZ. A detecção é feita partindo-se do principio que, quando um sistema é atacado, ele é usado para descobrir outros sistemas de rede e atacá-los também.

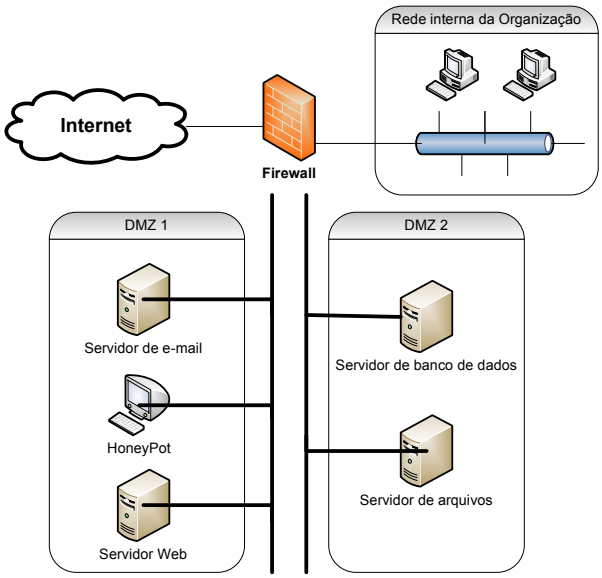

Figura 2.31: Honeypot do tipo Minifield
(NAKAMURA, 2007).

b) ***Shield*** **(escudo)**: o *honeypot* recebe os tráfegos considerados suspeitos, baseado nos serviços. O *firewall* ou o roteador direciona todo o tráfego não condizente com cada sistema para o *honeypot*, que passa a receber as informações do atacante.

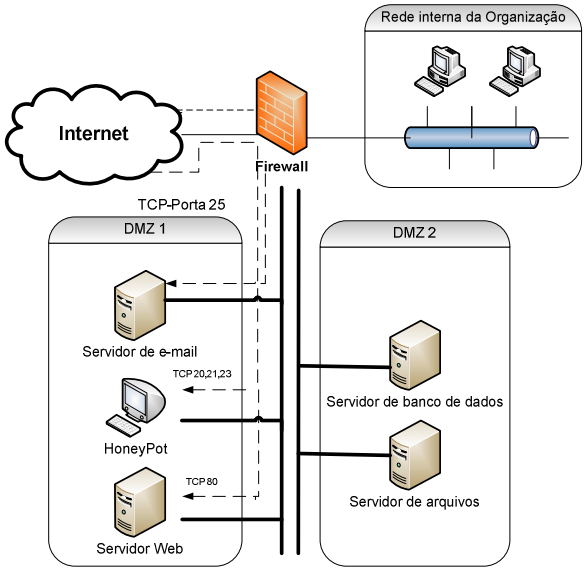

Figura 2.32: Honeypot do tipo Shield
(NAKAMURA, 2007).

c) ***Honeynet*** **(“rede de mel”):** é considerada uma rede de *honeypots*, com diferentes sistemas interagindo no ambiente. Esta rede pode conter facades, *sacrifica lambs* e *instrumeted systems*

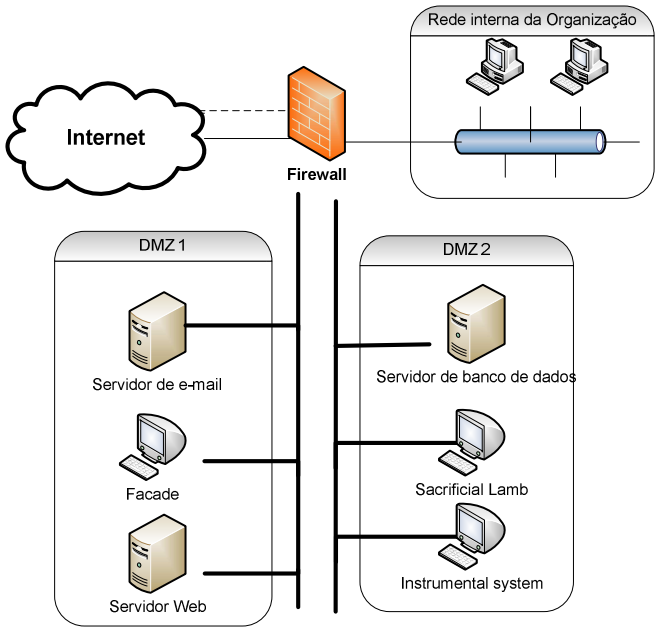

Figura 2.33: Honeypot do tipo Honeynet
(NAKAMURA, 2007).

Com a descrição dos *honeypots* se podem se prevenir de ameaças e através da correta configuração das ferramentas de detecção de intrusão. Aliada de boa estratégia de segurança estes sistemas permite elaborar uma boa base de conhecimento.

Portanto se encerra este capitulo que mostrou as principais ameaças e detalhou as características e arquiteturas das ferramentas utilizadas para realizar a detecção e prevenção, fornecendo informações referente as principais ameaças e ataques aos ambientes corporativos. No próximo capítulo irá se apresentar um dos principais desafios das ferramentas de detecção de intrusão.

3 GERAÇÃO DE DADOS PARA AVALIAÇÃO DE SISTEMAS DE DETECÇÃO DE INTRUSÃO

Serão abordadas as principais técnicas, tentativas e soluções para geração de *traces* para a avaliação das ferramentas de detecção de intrusão.

As ferramentas de detecção de intrusão, normalmente analisam informações que são armazenadas em determinado banco de dados de informações.

As principais formas de medidas dos tráfegos de pacotes (*traces*), que visam avaliar a eficiência das ferramentas, têm sido identificadas como um dos pontos críticos das pesquisas na área de segurança de redes de computadores.

Segundo MCHUGH (2000), os trabalhos mais antigos que realizaram a avaliação das ferramentas de detecção de intrusão, como de Puketza e das universidades da Califórnia, reportam claramente os esforços para geração de tráfego.

Serão descritas, a seguir, as principais iniciativas para elaboração de tráfegos na tentativa de simular um ambiente real.

3.1 As principais iniciativas para geração de dados

Nesta seção serão apresentadas as principais metodologias criadas para reproduzir um tráfego que posteriormente é usado para avaliar uma ferramenta de detecção de intrusão.

Segundo RUOMING (2006), o compartilhamento dos dados gerados pelos pesquisadores e instituições em seus laboratórios permite: (i) Verificação dos resultados anteriores; (ii) comparação direta de idéias e (iii) uma larga visão para um investigador pode comumente ter ao verificar os dados.

A importância do compartilhamento dos dados elaborados tem fornecido aos pesquisadores maiores subsídios para seus laboratórios.

Assim serão destacadas nas próximas subseções as iniciativas que foram desenvolvidas para elaboração de tráfego

3.1.1 Iniciativas de geração de tráfego legítimo

Serão mostradas considerações para gerar-se um tráfego legítimo para experimentos a serem realizados.

- **2005 – Sobre a geração de dados artificiais para tráfego normal**

Considerações importantes foram levantadas por SEEBERG (2005), referente a criação de um tráfego legitimo. Programas de geração de tráfego, como SmartBits e o Antara Flamethrower, geram tráfego legítimo de acordo com regras pré-definidas tendo a possibilidade da inserção de futuros ataques durante a simulação do tráfego. A principal vantagem deste tipo de pesquisa é a de ser reproduzível. Os testes de seleção de dados podem ser livremente distribuídos. Há algumas desvantagens destacadas:

- ✓ O custo de simulação do tráfego é alto para pesquisas, por onde o avaliador deve planejar e assegurar que a simulação fará um teste válido, seguindo pré- requisitos a serem estabelecidos;
- ✓ Se há necessidade de simular uma alta carga de tráfego de rede, será um problema gerar o que seja suficiente;
- ✓ Redes geram diferentes tipos de *traces*, sendo que não é possível gerar uma seleção de dados genérica que se adapte a todas as instituições e empresas;
- ✓ Os geradores de tráfegos de rede têm a tendência de gerar informações conforme as definições dos protocolos. Os tráfegos anormais têm a tendência de gerar falso-positivos. Simular tráfego legítimo traz a idéia de ter certa proximidade com a realidade, porém às vezes não se consegue chegar ao resultado esperado.

Notam-se as dificuldades para tratar e desenhar cenários que simulam uma realidade. Os tráfegos considerados benignos variam de uma instituição para outra, o que torna difícil definir um padrão.

Destacou-se nesta subseção considerações de geração de tráfego de *background* que normalmente são utilizados nos experimentos para posterior inserção de tráfego de ataques.

3.1.2 Iniciativas de geração de tráfego malicioso

Serão citadas nesta subseção as principais maneiras de gerar um tráfego malicioso. Muitos dos ataques são implementados de forma a fornecer e elaborar os tráfegos de forma a se tornar de forma real.

A grande maioria das simulações elaboradas dos tráfegos maliciosos, conforme MCHUGH (2000), foram desenvolvidos via *script* e programas, de variadas origens, que automatizam os ataques. No que pode ser determinado, a partir das descrições disponíveis, é que não foram feitas tentativas para assegurar que os ataques elaborados artificialmente foram realmente distribuídos no contexto geral.

- **1997 - Plataforma de software para geração de tráfego malicioso:**

Uma das primeiras tentativas para geração de tráfegos foi feito por PUKETZA (1997), em meado final dos anos 90.

No laboratório realizado na época, foi desenvolvida uma plataforma de *software* que criava uma série de *scripts* com variados parâmetros, que realizava a simulação de um tráfego de ataques. Abaixo na figura 3.1, mostra o funcionamento desta plataforma de *software* e interação de um atacante, que se inicia através de uma conexão via *telnet*[8].

[8] **Telnet** é um protocolo cliente-servidor de comunicações usado para permitir a comunicação entre computadores ligados numa rede.Telnet é um protocolo de login remoto

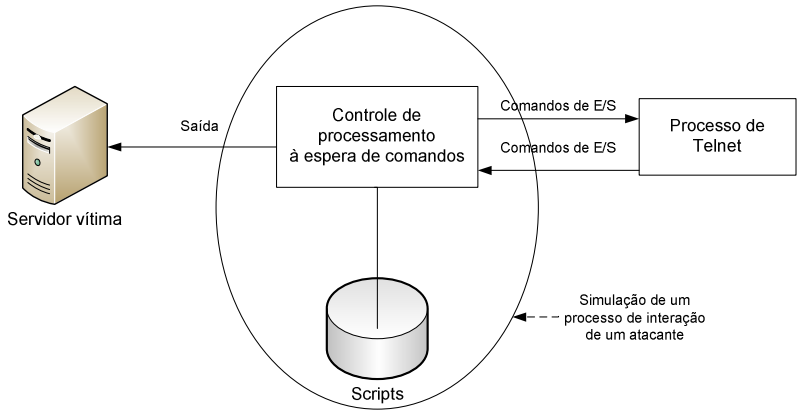

Figura 3.1: Plataforma de Software para simulação de tráfego (PUKETZA, 1997).

- **2006 - Geração de Tráfego utilizando uma infra-estrutura de máquina virtual**

Uma tentativa de elaborar tráfegos foi utilizando uma infra-estrutura de máquinas virtuais, em MASSICOTE (2006), no qual se utilizou uma estratégia rápida para gerar e coletar um largo número de tráfego de ataques. Para desenvolver esta seleção de dados em larga escala, um infra-estrutura de rede controlada foi desenvolvida, que permite:

- Armazenamento de todos os tráfegos de redes: pode ser usado para estudar comportamento de ataques;
- Controle do tipo de trafego: Consegue se ter um controle do trafego de modo a se ter um *trace* " limpo" que contenha somente trafego relevante para cenários de ataques;
- Controle de propagação do ataque: A propagação do ataque é confiando para somente o ambiente estruturado, prevenindo infecções de maquinas físicas;
- Uso do real e heterogênea configuração de sistemas: com o ambiente de máquina virtual, criam-se diferentes modelos (*templates*) de configurações de softwares (sistema operacional, serviços, dentre outros). Com isto conseguiu-se formar um banco de dados de modelos de máquina virtual que pode ser rapidamente desenvolvido.
- Rápida restauração para condições padrões (iniciais): Com funcionamento de alterações para recuperar um ambiente operacional de maneira instantânea, os testes com diferentes ataques são facilitados. Todos os cenários de ataques podem ser realizados sob as mesmas condições.

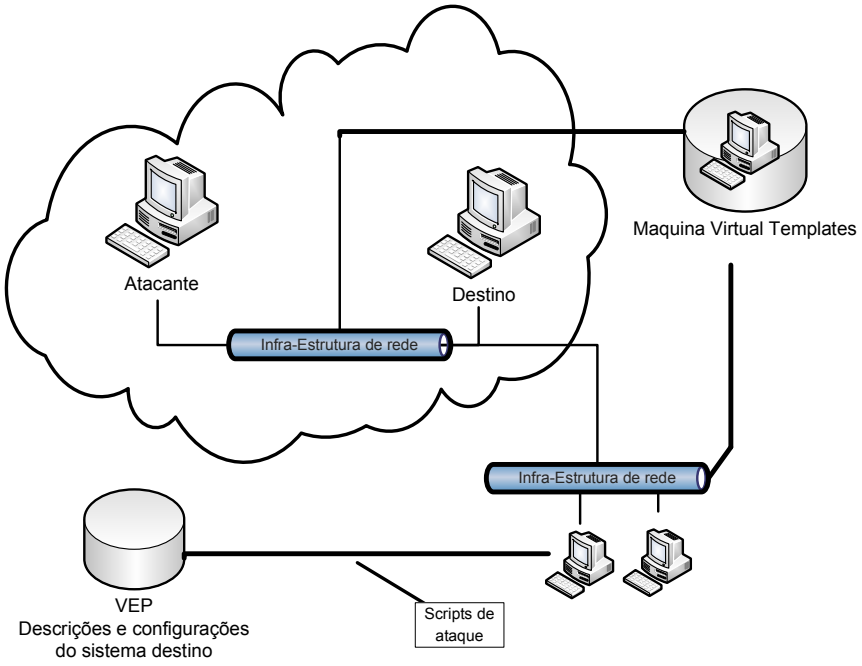

Figura 3.2: Ambiente de simulação de tráfego de ataques utilizando Maquinas Virtuais (MASSICOTE, 2006).

O ambiente virtual, conforme figura 3.2, contém um sistema de ataque, tendo, como alvo os sistemas e infra-estrutura de serviços de rede e equipamentos, criados em maquinas virtuais. Os sistemas de ataques são usados através de programas que exploram vulnerabilidades (VEP). Para infra-estrutura, como um todo, é assegurada a comunicação para todos os componentes integrantes enquanto um ataque está em progresso. Na etapa de geração de *scripts*, na seleção de dados, usou-se um banco de dados para automatizar a execução de cada programa de exploração de vulnerabilidade.

3.1.3 Iniciativas de geração de tráfego híbrido

São destacados os principais esforços criados para gerar um tráfego que englobe tanto um *trace* com ataques como um de saída adequada.

- **1997-1998 – Laboratório IBM Zerich para trafego híbrido:**

Uma plataforma de teste de IDS foi desenvolvida neste laboratório. Foram automatizados ataques e o tráfego normal para servidores de FTP. Os tráfegos foram gerados somente para servidores FTP e um pequeno número de ataques foi desenvolvido. Segundo MELL (2003), nos testes, uma inicial baixa detecção foi notificada e a alta média de alarmes foi improvisada por um cíclico, sendo repetido o teste, que gerava um tráfego real e normal. Este último foi destacado como complexo e com um alto tempo de consumo.

- **1998 – Laboratórios da Universidade da Califórnia em Davis (UCD):**

Um dos primeiros esforços de pesquisa foi o da Universidade da Califórnia em uma plataforma de teste, que automaticamente, executava ataques, usando interações via sessões de telnet, FTP e rlogin[9]. Conforme LIPPMANN (2000), os *scripts* de sessões de tráfego normal e ataques foram executados durante testes que avaliavam a habilidade de um IDS para distinguir intrusões de um tráfego normal. A ferramenta usada para o teste

[9] **rlogin** é um programa utilitário para o sistema operacional Unix que permite aos usuários realizar log in em outro computador via rede via protocolo TCP na porta 513

foi chamada de Network Security Network e utilizado em poucos ataques como quebra de senhas, transmissão de um arquivo de senhas para uma estação e exploração de vulnerabilidades. Com uma alta carga de processamento, pacotes foram perdidos e ataques não foram detectados.

- ***1998 - MIT Lincoln Laboratory (MIT/LL)***

Neste laboratório, foram simulados 300 ataques, durante 9 semanas, que foram coletados para uma posterior avaliação. Conforme LIPPMANN (2000), estes 300 ataques foram desenhados para 32 diferentes tipos de ataques em 7 cenários distintos. Estes tipos de ataques cobriram as diferentes classes de antigos ataques e também aqueles que eram conhecidos.

A Figura 3.2 mostra o detalhamento de um diagrama de blocos de realização dos testes e as respectivas coletas de dados dos tráfegos. A entrada do tráfego é simulada por uma base da *Air Force*, que contém 3 *hosts* que tem o papel das vítimas dos ataques (Linux 2.0.27, SunOS 4.1.4, Sun Solaris 2.5.1), e um gateway para 100 de outros *hosts* que são simulados por máquinas virtuais.

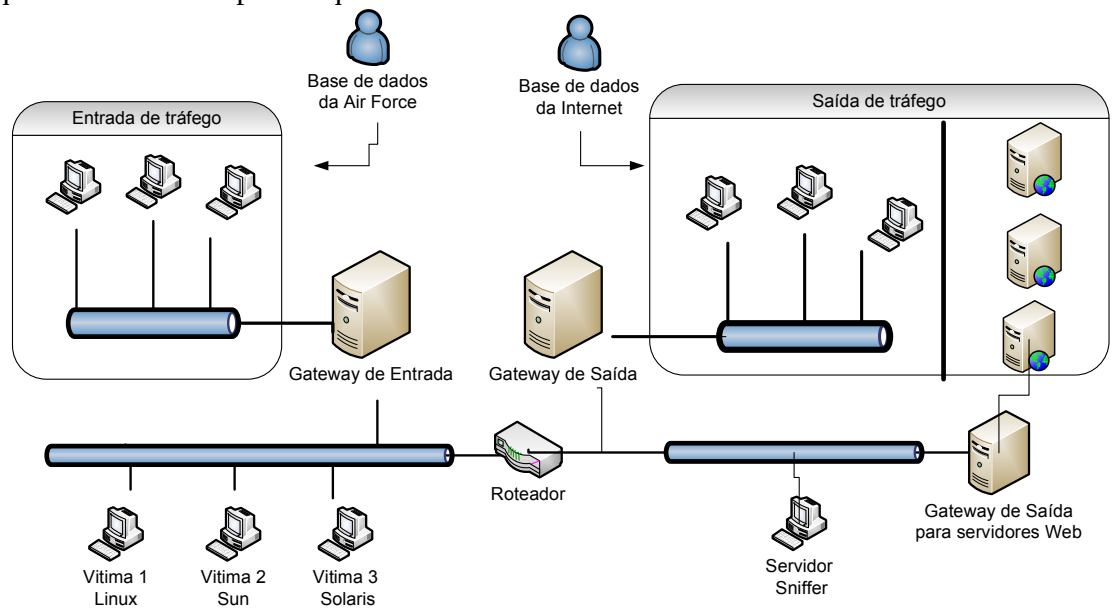

Figura 3.3:Criação de muitos tipos de tráfego usando 1000 hosts virtuais

(LIPPMANN, 2000).

Para o tráfego legitimo, os dados gerados foi usando mais de 20 serviços, da rede da *Air Force*, incluindo DNS, FTP, HTTP, ident[10], ping, POP, SMTP, SNMP, telnet, dentre outros. Em ordens de exatidão, os tráfegos coletados foram em um ambiente de rede com mais de 50 computadores na rede.

O tráfego de saída simula a i*nternet*. Para coletar este tráfego se continha um *sniffer*. Além disso, existia um *gateway* para os 100 *hosts* simulados em outras sub-redes.Uum segundo *gateway* emulava milhares de servidores web.

Em KENDALL (1999), verifica-se como funcionou o mecanismo de geração de tráfego, através da criação do automatizado processo de sessões de ataques artificiais,

[10] O protocolo **Ident**, especificado na RFC 1413, é um Protocolo de *internet* que auxiia a identificar o usuário de uma particular conexão TCP-IP.

sob vários *traces*, incluindo dados considerados legítimos (normal). O mecanismo de regeneração de tráfego foi designado para ser:

- Automático: de forma que a não requisitar alguma interferência humana;
- Reprodutível: quando repetido, sessões produziam resultados iguais e,
- Robusto: Pode executar por um longo período, sem haver necessidade de supervisão humana.

A parir da linguagem denominada *expect*[11], que tinha como objetivo a automatização da interação entre sistema e o usuário, se permitia sessões independentes para serem executadas como se o usuário estivesse digitando em um teclado.

- ***1998 – 1999:*** **Laboratório de Pesquisa Força Área (AFRL)**

Na instituição AFRL, sob a sujeição da DARPA, participou das pesquisas dos IDS com MIT/LL nos anos de 1998 e 1999. Este laboratório foi similar ao MIT/LL, mas foi realizado em tempo real em uma complexa estrutura hierarquia de rede. Os sistemas de detecção de intrusão foram instalados em um "teste de mesa", por 4 horas onde o tráfego normal era executado e os ataques eram disparados contra *hosts* em meio ao tráfego normal. A AFRL simulou a larga rede desenvolvida por um *software* (usada também no laboratório de MIT/LL) para dinamicamente associar um arbitrário endereço IP origem para os computadores que executavam teste.

- **2004 – Framework para manipulação de tráfego de pacotes**

Neste laboratório se descreve um sistema que prove um modelo para realizar a manipulação do trafego, baseando-se, conforme RUPP (2004), em uma linguagem de configuração que convenientemente especifica uma seleção que provê a realização de operações básicas, através de componentes de processamento de dados.

Foram assim, criados *plug-in* para facilitar a reutilização e habilitação de complexos sistemas de manipulação. Assim existe neste sistema um *plug-in* de entrada e de saída para facilitar o usuário para estender o sistema para atribuições de leitura e escrita dos dados de *traces* de formatos arbitrários.

O modelo distingue entre diferentes tipos de componentes de processamento dados:

- *DataSources :* provem um sistema de filtro com dados de entrada;
- *DataSinks :* Modificações de dados salvos por um sistema de filtro;
- Componentes Intermediários*:* como os filtros realizam algumas operações de manipulações sendo que um p*ipe*[12] pode ser usado como buffer de dados entre dois componentes;

Foi definida uma interface genérica para que todos estes componentes sejam implementados. Eles podem ser desenvolvidos independentemente do sistema principal.

A arquitetura prove plug-ins de dois tipos de mecanismo para transferência de dados. O controlador constitui a mais importante parte da arquitetura. Ele prove todos os métodos necessários de construção e controle do FilterNetwork. Na figura 3.4 é mostrada a estrutura da aplicação.

---

[11] **Expect** é uma ferramenta de automação do sistema operacional Unix, escrito por Don Libes como uma extensão para linguagem de *script* TCL, para interação de aplicações como telnet, FTP, dentre outros.

[12] ***Pipe***, ou "canalização", é o redirecinamento da saída padrão de um programa para a entrada padrão de outro.

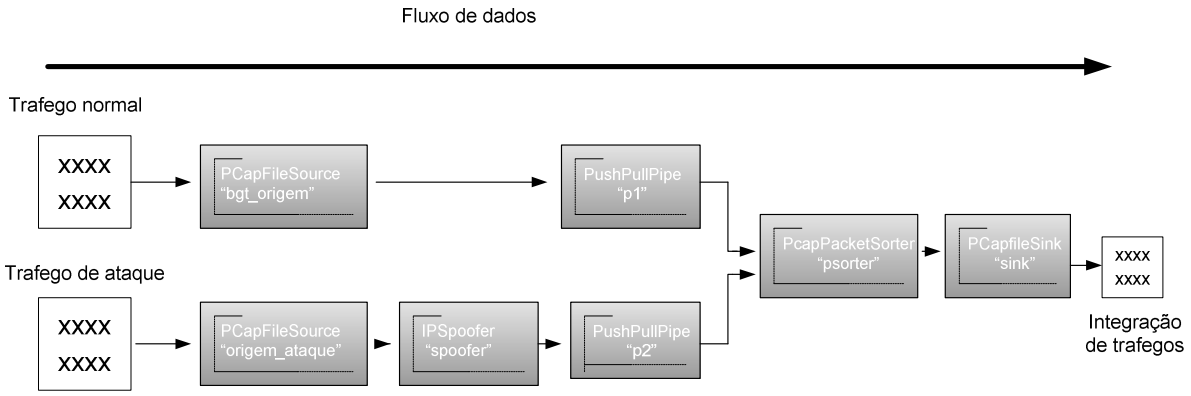

Figura 3.4: Aplicação NetworkFiter

(RUPP, 2004).

A aplicação de geração de tráfego consiste de:

- **plug-in PCapFileSource** : Pode suportar um filtro de pacotes de rede de arquivos da biblioteca libpcap, ajustando o timestamp[13] e o mapa de pacotes (da camada de transporte) para fluxos na base principal suprindo as especificações de uso.
- **plug-in PCapFileSink**: escreve pacote que tem sido manipulado pelo *NetworkFilter* sendo compatível com a biblioteca libpcap;
- **plug-in IPSpoofer**: pode adaptar os endereços e portas dos fluxos de pacotes de acordo com a seleção aplicado pelo usuário, das regras de *spoofing* que tem a sintaxe BPFlike;
- **plug-in PCapPacketSorter**: intercala, cronologicamente, pacotes de múltiplas InputSides;
- **plug-in PushPullPipe :** pode simplesmente agir como mecanismo de transferência convertendo dados entre as interfaces do PCapPacketSorter e colocando nas interfaces do PCapFileSourcer , IPSoofer ou outras instâncias.

- **2006 - *Framework* Trident para geração de tráfego híbrido:**

Esta proposta inclui um modelo que gera dados através de um ferramentas que automatizam o tráfego. O modelo *Trident*, pode gerar pacotes tradicionais para avaliações convencionais e também pode ser usado em laboratórios com configurações controladas para avaliar o desempenho online das características dos NIDS. As capacidades do modelo criado incluem, conforme SOMMERS (2006):

- A habilidade para gerar um representativo tráfego legítimo, incluindo *payloads*;
- A habilidade para construir e gerar novos tipos de tráfego malicioso;
- Combinação entre os tráfegos de testes maliciosos e os legítimos;
- Modular o volume dos tráfegos de testes maliciosos e legítimos, e
- Modular o tempo de chegada dos processos de ambos os tráfegos.

Neste modelo foi realizada a simulação do tráfego baseado em *payload*[14] intercalados. Esses, eram dinamicamente construídos através de fluxos de pacotes de

[13] Um **timestamp** é uma seqüência de caracteres, indicando os dados e/ou tempo indicando datas e/ou tempo de acordo quando certo evento ocorre.

[14] **Payload**, ou **carga útil**, em protocolos de comunicação refere-se ao dado real sendo transmitido. Ele é seguido por um cabeçalho que identifica o transmissor e o receptor do dado sendo transportado e é logo descartado assim que chega ao destinatário.

forma randômica, que correspondem a um estado particular no serviço de geração automático. Os pacotes (cabeçalhos e *payloads*) com os fluxos são extraídos dos *traces* com conteúdo malicioso.

Conforme (SOMMERS 2006), foi discutido miniciosamente os benefícios e desvantagens adotando três estratégias para o crescimento e geração do tráfego de pacotes no modelo Trident:

1) Sinteticamente gerar *traces*, usando modelos estatísticos desenvolvidos de *traces* reais - ao vivo – como foi feito na criação da seleção de dados da DARPA;
2) Usar um NIDS com seleção de regras para extrair pacotes legítimos de uma empírica coleção de *traces*;
3) Baseada na noção de uma matriz verídica o qual, simula as regras de um NIDS, é usada para extrair pacotes benignos de coleção de tráfegos de pacotes. Ele é importante para notificar que nenhum destas estratégias pode ser absolutamente garantido, sendo que os resultados dos *traces* benignos possa ser livre dos pacotes maliciosos. Trident se constitui dos seguintes elementos mostrados na figura 3.4:

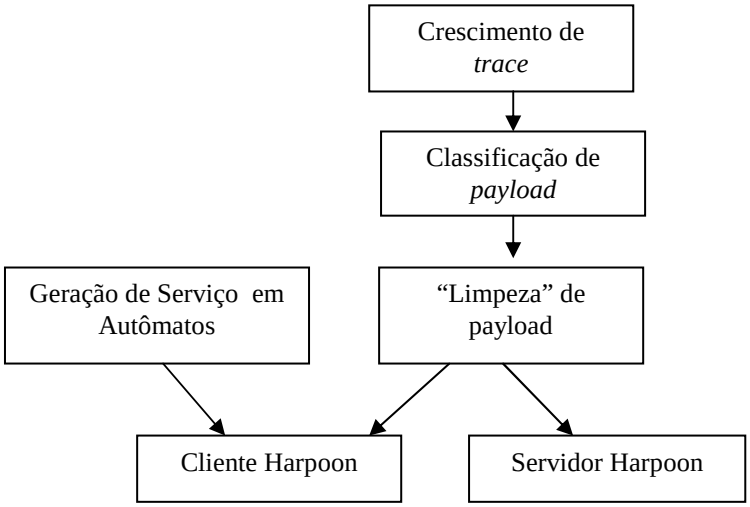

Figura 3.5: Passos para geração de tráfego no modelo Trident

(SOMMERS, 2005).

Seguem-se destacados os elementos do modelo:

- **Crescimento de trace:** O objetivo desta fase é criar um largo número de pacotes reais e diversificados;
- **Classificação de Payload**: Os tráfegos "crus" classificados como legítimo é dado como entrada para o modulo de classificação, chamado de *payload-gen*. O propósito deste é classificar nos tráfegos correntes, estados de diferentes serviços automatizados. Neste passo, os pacotes gerados no estado da aplicação, são distribuídos em diferentes fluxos que são agregados ao mesmo tráfego corrente;
- **Limpeza de Payload.** Após a classificação, *payloads* são descartados ou modificados para assegurar que eles não tenham violados requisitos estabelecidos. As modificações aqui são feitas para simplificar a definição dos serviços nos autômatos e seus processamentos. O efeito desta limpeza é que se deve reduzir o nível de falsos alarmes gerados pelos NIDS;
- **Geração de serviços autômatos**: É chamado o coração do gerador de tráfego legítimo, sendo uma coleção de autômatos com estados que descrevem as classes de

pacotes observadas em um especifico serviço. Foram criados autômatos que modelam protocolos como HTTP, SMTP, DNS, Telnet, FTP e SSH. Estes autômatos descrevem cada serviço em três fases de abstração que é tipicamente encontrada nos protocolos de rede.

- **Geração de contéudo de tráfego via Harpoon**: Ocorre por meio da utilização de um plug-in para ferramenta de geração de tráfego Harpoon [15], objetivando a execução do estado dos autômatos para transmissão dos *payload* "limpos", sendo que o controle destas é efetuado pelos clientes Harpoon. Os servidores Harpoon, simplesmente, respondem às requisições emitidas, enviando certos números de intercalados *payload* da especificada aplicação. A mudança dos *payloads* da camada de aplicação é dada por um padrão de *sockets*, acima da camada de transporte. Fluxos de tráfegos legítimos, produzidos pelo *Trident*, são destinados a um IDS.

A geração de tráfego malicioso foi criado utilizando um *framework* denominado MACE que consiste de 3 componentes: (I) Exploit, (II) ofuscação e (III) propagação. Além disso, possui um número de funções para suporte a interpretação, execução e exceções de manipulação de perfis de ataque. Conforme SOMMERS (2004) este modelo é mostrado abaixo:

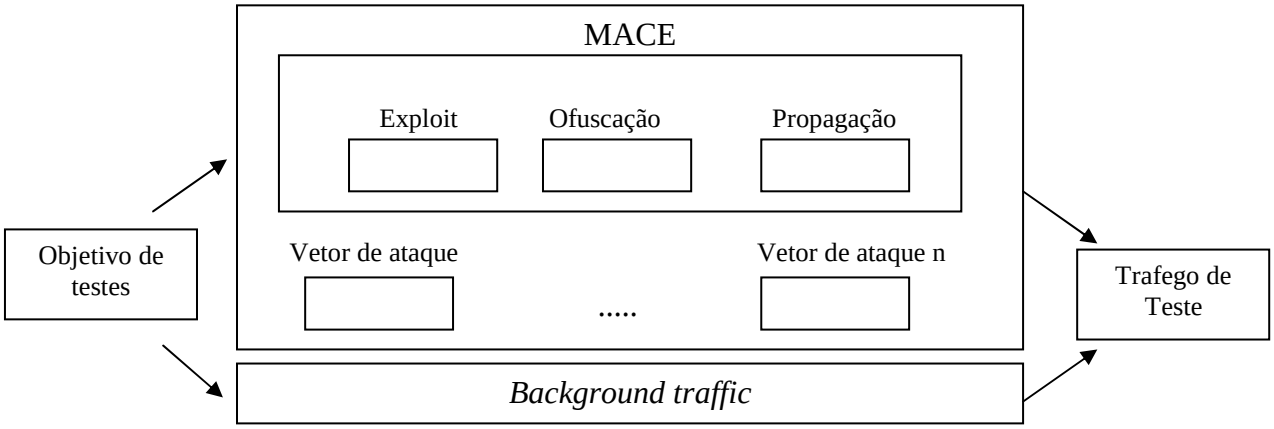

Figura 3.6: MACE arquitetura

(SOMMERS, 2004).

- Modelo Exploit — seleção de vulnerabilidades que são parte da seqüência dos ataques;
- Modelo Ofuscação — modificar no cabeçalho ou payload para habilitar o exploit para iludir o NIDS. Este poderá ser voltado à camada de rede ou aplicação;
- Modelo de propagação – A ordem na qual as vitimas são escolhidas para serem atacadas;
- Modelo de tráfego normal (*background*) — Fluxo de Tráfego legítimo na rede.

A arquitetura do MACE mostrada acima tem como objetivos: testes com a informação, seleção e composição das fases exploit, ofuscação e propagação em cima de uma série de vetores de ataques. Ferramentas existentes como Harpoon produzem o tráfego legítimo.

[15] Conforme (SOMMERS, 2004), Harppon é uma aplicação independente para gerar um representativo tráfego de pacotes em nível de fluxo do protocolo IP. Geram-se com ela fluxos de pacotes TCP e UDP que tem o mesmo *byte*, pacote, características temporais e espaciais

- **2006 - Cenário de criação de *traces* hibridos de Wang e Zhao**

Um das principais maneiras para tentar gerar tráfegos maliciosos, é entendendo como funciona um ataque. Assim foi conduzido para a constituição deste laboratório o seguinte ambiente apresentado na figura

Neste ambiente proposto encontra-se a geração de *traces* de ataques através dos seguintes critérios:

- Ataques indicados pela Security Focus (14 categorias de ataques em Janeiro até Junho de 2002);
- As técnicas de ataques empregadas são típicas;
- Depois do ano de 1999, novas técnicas de técnicas de *hacking* foram desenvolvidas ***rapidamente conforme*** WANG (2006)*:*

> *(1) utilização de sniffers para obter senhas e outras informações; (2) Atacar vulnerabilidades de scripts em linguagens de programação como ASP, PHP, Perl entre outras; (3) utilização de compartilhamentos da plataforma Microsoft; (4) Codigos maliciosos embutidos em páginas web ou anexos aos e-mails; Sessões de seqüestro incluindo falsos IP, DNS ou ARP, etcs.. ; (6) propagação de worms através de ataques a redes ou vulnerabilidades de sistemas ; (7) Ataques distribuídos de negação de serviço e, (8) ataques a nível de aplicação associados a bug ou vulnerabilidades.*

O desenho do laboratório mostrado na figura 3.6 apresenta os seguintes componentes caracterizados para geração dos tráfegos de ataques e legítimos:

- agendadores de ataques: inserção de ataques automatizados e manualmente, onde se é utilizado, um programa com um conjunto de *scripts* coletados e prontos para serem executados. Os ataques possuem parâmetros definidos através do *script*;
- inserção de ataques manuais: Um servidor destinado a ataques manuais, através da intervenção de um usuário;
- *sniffer:* Utilizado para captura dos tráfegos gerados;
- NIDS e HIDS: utilizados para avaliação da eficiência das ferramentas testadas.

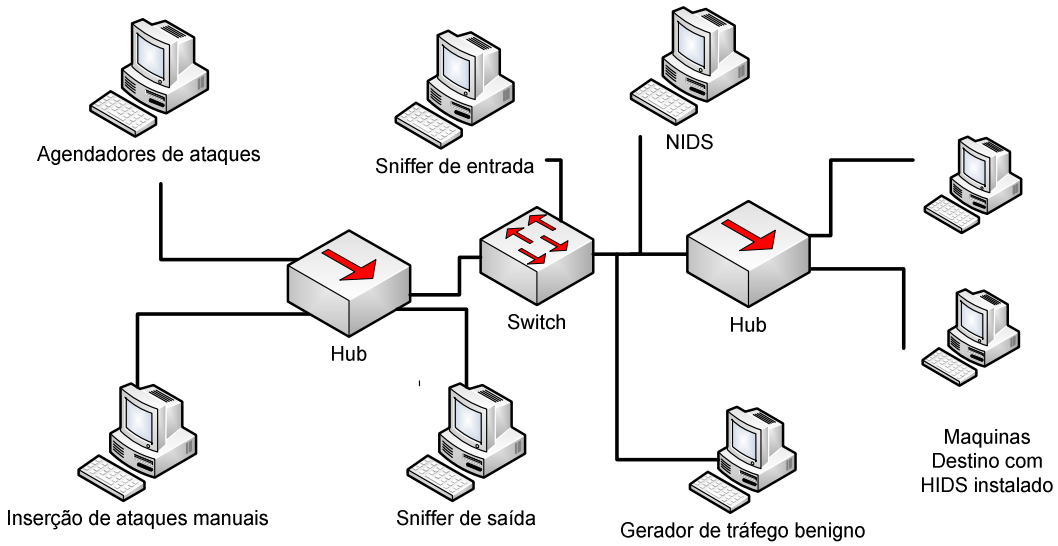

Figura 3.7: Ambiente de trafego híbrido

(WANG, 2006).

No experimento foram utilizados juntos 9 servidores. Dois deles eram responsáveis para simular 40 ataques automaticamente. Outro serviria para inserção manual dos ataques. Foi utilizando o ambiente *Vmware* em um dos servidores para aumentar o número de máquinas destinos que podem ser simulados no experimento. Depois de iniciar o experimento foi ainda utilizado uma seleção de dados, da segunda semana de tráfego da seleção de dados do MIT de 1999. Foram invocados 2015 ataques e 19267 sessões de seqüestros foram geradas. Para o tráfego normal foram geradas 468.093 sessões. Foi executado o experimento durante 5 dias e 22 horas. Durante o experimento todas as sessões foram gravadas utilizando o utilitário tcmdump[16].

Conforme foi apresentado, tentativas foram realizadas no intuito de gerar tráfegos para avaliação da eficiência das ferramentas de IDS. Na próxima seção será mostrada a análise das iniciativas de geração de tráfego.

3.2 Análises das propostas para geração de dados

As propostas apresentadas foram colocadas em cima de cenários simulados, sendo utilizadas em sua maioria, ferramentas para geração de tráfego legitimo e malicioso.

Dificilmente as avaliações feitas apresentam resultados necessariamente completos para qualificar um sistema de detecção de intrusão.

Verifica-se que os tráfegos de hoje são muito diferente de um ano para outro ou de anos atrás.

Os resultados das avaliações na maneira que foram apresentados e gerados, conforme MCHUGH (2000), tem sido apresentando por dificuldades e limitações.

3.2.1 Análise sobre a geração de dados de forma real

Algumas pesquisas têm testado IDS pela injeção de ataque em um fluxo real de trafego normal. Conforme MELL (2003), isto é uma técnica muito efetiva para determinar a eficiência de um IDS, dado um particular nível de atividade em *background*. A média dos testes que usam esta técnica deve ser bem recebida por que as atividades em segundo plano são reais e contêm todos os tipos de atividades de anomalias. Há algumas desvantagens referentes a esta técnica:

- Usualmente não é possível ter um teste repetidos usando tráfego real pois há dificuldades políticas e técnicas para armazenar e repetir uma larga soma de tráfego (especialmente em um ambiente de *backbone*). O *hardware*, normalmente, tende a falhar repetidamente em pacotes de rede e devido a isto, tenta-se paralisar este processo devido a seqüência de problemas;
- Estes experimentos usualmente têm-se uns pequenos números de máquinas vítimas que são iniciadas por um único propósito de iniciar um ataque durante o teste. Alguns IDS devem ser capaz de detectar somente máquinas que são atacadas e assim artificialmente consegue elevar seu desempenho no teste.
- Há privativos interesses relacionados para publicar informações dos testes para ser usado para real atividade de tráfego real;
- Repetições dos testes devem causar algum prejuízo, pois a regulagem do tempo deve ser refinada, de modo que haja um maior controle;

Verifica-se que as considerações passadas, em geral, são consideradas até os dias de hoje, pois realmente é difícil você conseguir agregar todos os componentes que influenciam um análise com estudos mais aprofundados. Assim será abordada, uma

[16] **tcpdump** é uma ferramenta utilizada para monitorar os pacotes tráfegados numa rede de computadores. Ela mostra os cabeçalhos dos pacotes que passam pela interface de rede

alternativa que não sofre restrição política, que é a geração de tráfego artificial, porém é custosa de ser elaborada.

3.2.2 Análise sobre testes que usaram a geração de dados de forma artificial

Superando os problemas de política e privacidade de usar, analisar, e/ou distribuírem atividade de tráfego real, alguns pesquisadores têm propondo a "limpeza" para atividades de tráfego de *background.* Exemplos citados de limpeza do trafego são utilizando os cabeçalhos dos pacotes dos protocolos. Os cabeçalhos, conforme (MELL, 2003), foram armazenados durante um tráfego normal para ser usado em um modelo de estatísticas de rede e não para avaliar um sistema de detecção de intrusão.

No experimento mostrado por SOMMERS (2006), no *framework* denominado *Trident* que o tráfego que é coletado, sendo pré-gravado e realizado uma espécie de higiene, removendo dados sensitivos. Então os dados de ataque são injetados com o fluxo de dados limpos, podendo ser acompanhados pela repetição e execução de ataques concorrentemente. Assim a vantagem deste tipo de abordagem é os dados de teste podem ser livremente distribuídos. As principais dificuldades são listadas a seguir:

- As tentativas de sanatização devem acabar com a remoção de muito conteúdo de tráfego de *background,* criando um ambiente não real;
- Tentativas de sanatização podem falhar causando um inesperado lançamento de dados. Este cenário é muito possível desde que o dado não seja classificado por um humano para verificar a sanatização em grande volume dados. Este risco, conforme MELL (2003), é que as organizações não tolerarão que aconteça para um projeto de pesquisa.
- Desde que um ataque tenha sido injetado artificialmente em fluxo de dados sanatizados, os ataques não interagem realisticamente com atividade de *background*.
- Quando é feito a limpeza do tráfego real, pode ser difícil remover ataque que existem no fluxo de dados. As adequações realizadas nos dados, às vezes, podem remover informações que são necessários para detectar um ataque.

As técnicas utilizadas nesta abordagem visam simular a realidade que as empresas e de mais instituições e órgãos passam no seu dia-a-dia, porém se trata de técnicas

3.2.3 Análise sobre a geração de dados sobre uma rede de testes criada.

Um das pesquisas mais comuns para testar os sistemas de detecção de intrusão é criando uma rede de testes, com *hosts* e uma infra-estrutura que pode ataques bem sucedidos e podendo-se gerar um tráfego de *background*. Estes testes na rede incluem vitimas sob um tráfego normal gerado por geradores de tráfego, que modelam as atuais estatísticas da rede. Conforme MELL (2003), a vantagem deste tipo de pesquisa é que:

- os dados podem ser distribuído livremente desde que não contenham qualquer informação privativa ou sensitiva;
- garante que as atividades não contenha qualquer ataques desconhecidos.

Ultimamente, os testes de IDS que usam tráfegos simulados são usualmente repetidos, e desde então ele passa ser reproduzido previamente gerando atividades de *background* ou ter um simulador que regenera a mesma atividade usada em um teste anterior.

As principais dificuldades neste tipo de abordagem são:

- Ë muito custosa e difícil de criar uma simulação;
- Pode ser difícil simular com alto poder de tráfego devendo haver algum limite a ser atingido;

- A necessidade por diferentes tipos de tráfegos para modelar vários ataques; Por exemplo, o tráfego de uma empresa é diferente de uma universidade.

Encerram-se nesta seção as analises feitas as principais iniciativas de laboratórios para geração de tráfego. A seguir, apresentar-se-á o modelo proposto para geração de tráfego.

3.3 Proposta de uma metodologia para geração de dados para avaliação de sistemas de detecção de intrusão

Uma metodologia que visa simular a construção de algo real precisa ser entendida de forma minuciosa e cautelosa, para verificar o funcionamento do mecanismo de geração de tráfego, desde a sua constituição até a sua finalização.

A simulação não é algo muito simples. Conforme WANG (2006), a dificuldade de vir a elaborar um *trace* de ataque, vem a ser complicado visto que simular novos e desconhecidos ataques, pois é muito difícil modelar e identificar os mecanismos de *hacking* utilizados.

Para a constituição de um tráfego, normalmente, utiliza-se de quatro fatores variáveis que diretamente influenciam para uma futura avaliação dos mesmos:

- Recursos de *hardware*: tipo de memória, espaço em disco;

- Recursos de *software*: Programas que foram instalados durante a execução do experimento: (1) Sistema operacional: Instalação de sistema operacional, atualização de pacotes de atualização, serviços que se executam; (2) Sistema de detecção de intrusão: Tipo NIDS ou HIDS;

- Espaço de tempo: dimensionar um tempo hábil para capturar um tráfego também é um ponto a ser verificado. Verificar o tempo e medir a duração de um tráfego (em unidade de tempo - dias, horas, segundos e minutos);

Com estes três parâmetros citados pode-se mencionar que eles influenciam diretamente na constituição do tráfego como um todo.

Apresenta-se, na figura 3.8, um modelo para geração de dados para ambas as situações que um tráfego pode conter: normal (convencional) e anômala (não-convencional).

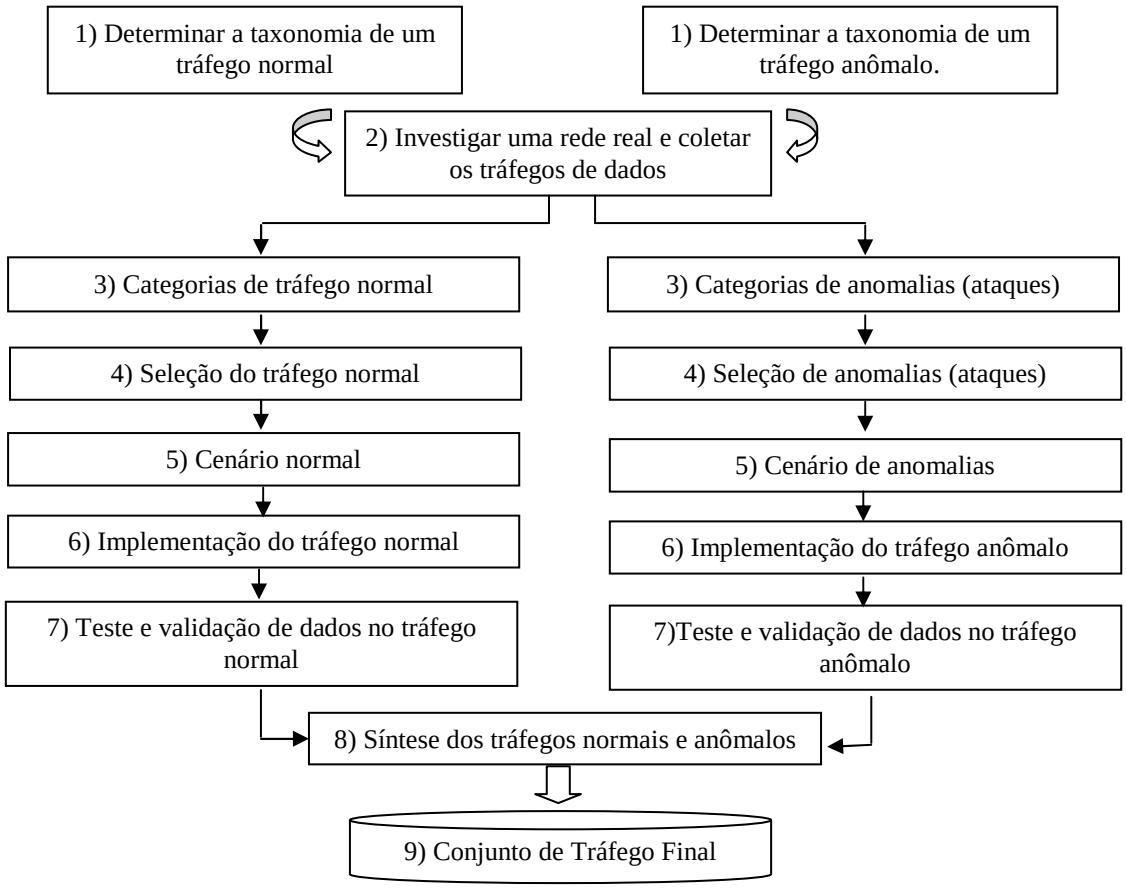

Figura 3.8: Proposta para geração de tráfego proposto adaptado (WANG, 2006).

Neste modelo apresentado, tentando ter uma aproximação da realidade, e verificando a geração de *traces* de ataque e dados normais devem-se destacar os seguintes pontos nesta proposta:

1) **Determinar a taxonomia**: a seleção ou a definição de uma apropriada taxonomia de ataques e de dados normais para ser usado no experimento.
2) **Investigar uma rede local:** a construção de um ambiente que se assemelhe e que simule o ambiente real, com coleção de dados com *traces* de comum uso a empresas e demais instituições;
3) **Elaboração de Categorias:** separar em categorias a seleção de *trace*s normal e anômalo é importante para determinar o que a ferramenta de detecção de intrusão precisa classificar de forma correta, durante uma avaliação;
4) **Seleção de tráfego:** realizar uma criteriosa seleção de dados de ambos os tipos de *traces*, permite ter uma análise mais objetiva do que é gerado e do que precisa ser classificado;
5) **Criação de um cenário:** A criação de cenários é importante para assegurar que os passos para constituição de uma anomalia e também de um tráfego normal. Conforme WANG (2006), os passos para constituição de um ataque e conseqüentemente um tráfego de seguem abaixo, considerando um processo genérico de ameaça:

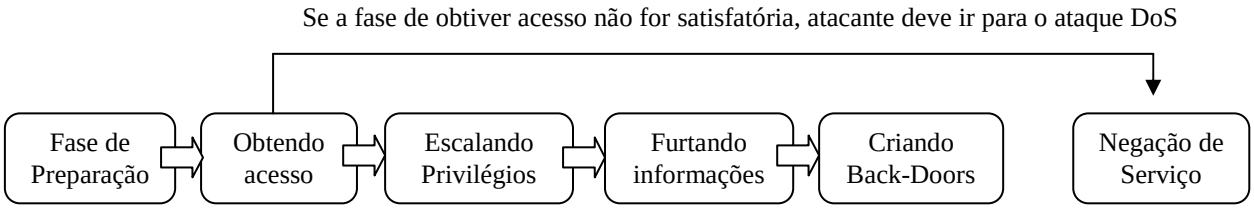

Figura 3.9: Fases de um ataque
(WANG, 2006).

Figura 3.10: Fases de acesso normal.

Assim nesta fase é importante para assegurar que o passo para criação de um ataque ou de uma ação normal mantenha as fases correlacionadas e integradas, sendo considerados dois pontos importantes:

(a) Distribuição de tempo e freqüência de cada categoria de *trace*;
(b) Correlações intercaladas em meio aos passos de um ataque ou de uma ação normal;

6) **Implementação de tráfego**: Estrutura que gera o tráfego de *background*, devendo ser conformado com os padrões estabelecidos em análise através de um modelo de dados reais. Utiliza-se nesta etapa geradores de tráfego que elaboram os dados que irá trafegar pela rede do cenário estruturado. Nesta fase então são inseridos *traces* em um tráfego de segundo plano. Dois pontos importantes a serem considerados:
   - Para ferramentas de detecção de intrusão de rede, baseada em comportamento: Algumas das ferramentas de geração de ataque, segundo (WANG, 2006), podem ser invocadas automaticamente, mas outras necessitam de uma intervenção humana.
   - Para ferramentas de detecção de intrusão de *host*, baseada em comportamento: Muitas tarefas que um atacante ou um usuário comum, usualmente fazem, precisam de um inicial acesso para o sistema destino. Muitas atividades de um *host* requerem uma intervenção humana e não pode ser executado automaticamente.
7) **Teste e validação de tráfego**: Após a geração de *traces* para a constituição do tráfego como um todo, deve-se validar o funcionamento dos mesmos junto a uma ferramenta de detecção de intrusão, bem como a estrutura de cenário elaborado para o experimento.
8) **Síntese de tráfego normal e tráfego anômalo:** Conforme (WANG, 2006), há dois diferentes métodos para combinar as duas origens de tráfego: *offline* ou *online*.
   - Método *offline*: é para gerar os tráfegos respectivos, combinando as duas seleções de dados *offline* de acordo com os *timestamps* dos pacotes.
   - Método *online*: envolve estabelecer um ambiente experimental contendo a ativação conjunta de um tráfego de segundo plano e de um gerador de tráfego de

ataque com a inserção manual de processos de ataque ao mesmo tempo em que os pacotes são injetados nos *links* de *internet*.

9) **Conjunto de tráfego Final**: Destaca-se neste componente um banco de dados com as informações das seleções de dados normais e de ataques. Como ela se deve permitir a avaliação por um banco de dados de fácil acesso que contenha informações precisas de todas as simulações elaboradas.

Como há limitações para elaborar um tráfego real, se utiliza a área de simulação de *traces*, que evoluiu bastante nos últimos tempos. Há existência de inúmeras ferramentas para geração de tráfego

Assim, nesta seção foi descrito um escopo do modelo que pode ser usado para gerar dados que se aproximam da realidade para *traces* normais e anômalos. A estrutura apresentada deve acompanhar as fases da elaboração de um *trace*, de forma a ser interpretado pelas ferramentas de detecção de intrusão.

4 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O trabalho presente constou em apresentar sobre uma área que está em constante estudo para a área de segurança computacional que é a avaliação da eficiência dos sistemas de detecção de intrusão.

Inicialmente foi mostrado, conceitos de segurança da informação que estão relacionados diretamente aos negócios das empresas e demais instituições. Contextualizou-se a inserção dos sistemas de detecção, em uma pequena cronologia histórica, desde o seu princípio até os dias de hoje. Estes sistemas foram detalhadamente estudados, associando-se eles as principais ameaças e as camadas do modelo de referência proposto para este trabalho.

Pode-se verificar que esses sistemas são úteis e podem ser utilizados em ambientes complexos e simples de segurança, podendo prover mecanismos de detecção de ameaças em uma medida pró-ativa.

A avaliação da eficiência destas ferramentas de detecção de intrusão têm sido um dos pontos mais cruciais para definição da implantação destes mecanismos em um ambiente de rede.

Para avaliação destas ferramentas são utilizados ambientes de simulação que visam aproximar-se de uma realidade, visto que, muitas vezes é impraticável realizar testes em um ambiente real. Assim foram mostrados, alguns trabalhos que se utilizavam de técnicas para elaboração de tráfego. Constatou-se que estas técnicas evoluíram e a aproximação com *traces* reais já pode se tornar uma realidade muito breve.

Por fim, mostrou um modelo que consiste da elaboração de um ambiente simulado, separando cenários de tráfegos em normais e de ataques, sendo constituídos conforme a necessidade do criador, no que se deseje simular, visto que os *traces* variam de uma instituição para outra, conforme a política de segurança que cada uma emprega.

Na seção desse capítulo será destaque a trabalhos futuros que se pode realizar com a iniciativa deste trabalho.

4.1 Trabalhos Futuros

Atualmente existem técnicas para geração de tráfego que se aliam de mecanismos híbridos para suas constituições, na tentativa de elaborar *traces* qualificados. Como sugestões para trabalho futuros apresentam-se os seguintes tópicos:

- **Elaboração de ambiente de testes para o modelo apresentado**

Através das ferramentas de elaboração de tráfego existentes e utilizando uma infra-estrutura de máquina virtual, se consegue isolar bem os *traces* para os cenários de *traces* normais e de ataques. Após a elaboração dos *traces*, em cenários separados, podem se unir ambas bases em uma única, para formar um tráfego híbrido para ser avaliado por um NIDS.

- **Detecção de intrusão usando traces dinâmicos**

Uma proposta apresenta por SCHUBA (2007) apresenta uma estrutura interessante para composição de uma ferramenta de detecção de intrusão junto ao kernel[17] de um sistema operacional, que tem a característica de ser dinâmico para o *traces* em nível de usuário. Esta técnica poderá ser empregada, eficientemente, para caracterizar em tempo de execução o comportamento de determinado código. Usando a técnica de *traces* dinâmicos, as seqüências das instruções do kernel podem ser instrumentadas de forma a não precisar ter acesso ao código de origem. Usando o desenvolvimento de técnicas, para verificar o estado dos dados, dos parâmetros passados e/ou informação do tempo, pode ser exemplo para prover mais detalhes de atuais comportamentos do sistema em questão. Em explorações baseadas em assinaturas, mais assinaturas são possíveis e aquelas baseada em anomalia, mais detalhe de sensitividade pode ser desenvolvido para discriminar entre normal e anômalo comportamento.

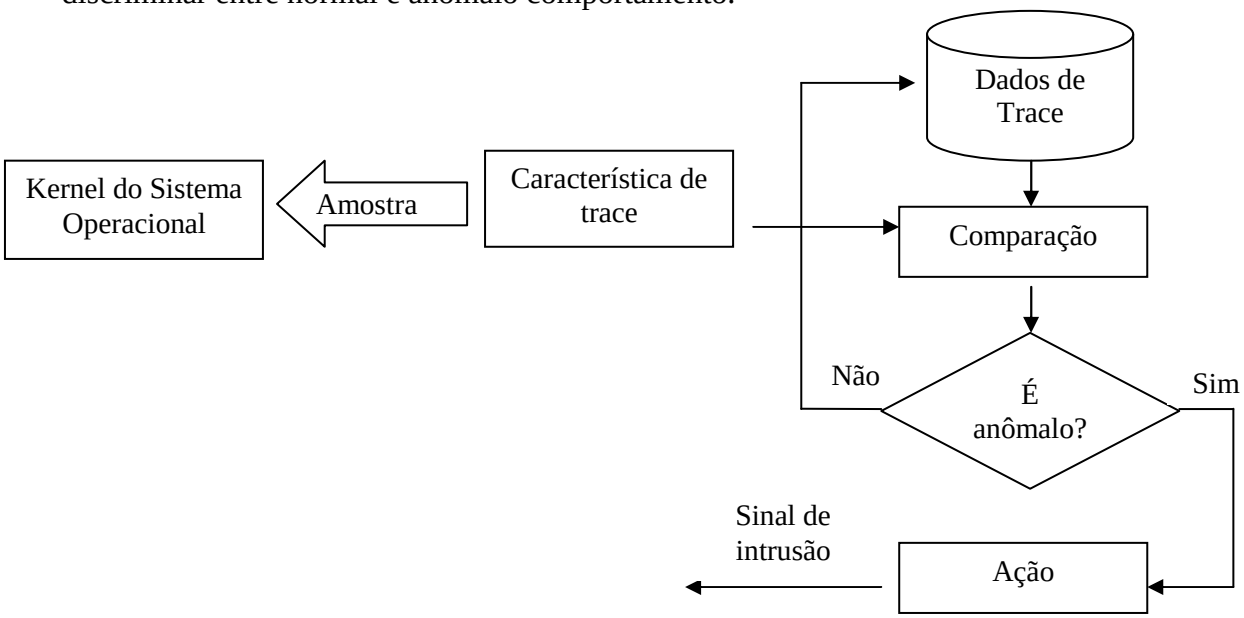

Figura 4.1: Proposta de IDS com trace dinâmico (SCHUBA, 2007).

- **Novas tecnologias a serem empregadas para Sistemas de Detecção de intrusão**

Novas técnicas, conforme MELL (2003), para IDS incluem as tecnologias de meta-IDS que tenta facilitar o gerenciamento de alertas extraindo informações dos ataques. O uso de *Appliances*[18] para IDS tem crescido ultimamente e promete-se incrementar a força de processamento, trazendo maior força e robustez para o gerenciamento de capacidade. Estas novas direções focadas em novas tecnologias representam exemplos de pesquisas e esforços para tratarem dos problemas de falso-positivos e distinção de erros para ataques que não são conhecidos.

[17] O **Kernel** de um sistema operacional é entendido como o **núcleo** deste ou, numa tradução literal, **cerne**. Ele representa a camada de *software* mais próxima do *hardware*, sendo responsável por gerenciar os recursos do sistema computacional como um todo.

[18] *APPLIANCE* é um *hardware* com características de uma arquitetura particular, onde é instalado um sistema operacional (Microsoft, Unix, Linux, etc) com recursos e configuração especificas, que executa um *software* (como um IDS) com finalidade definida.

Constata-se que as técnicas utilizadas pelos IDS aliados a variados mecanismos, procuraram filtrar o maior número de intrusões, gerando pouco ou nenhum erro de classificação.

Os esforços de Indústrias, empresas, comunidades de *softwares* abertos, e as universidades estão em estudos persistentes de maneiras a realizar a disseminação maior para uso dos sistemas de detecção de intrusão

Espera-se que o progresso das técnicas de sistemas de detecção de intrusão, aliados a medidas de segurança superem as tentativas e alternativas de atacantes de explorar vulnerabilidades e se utilizarem de técnicas variadas para causarem danos a instituições e empresas.

REFERÊNCIAS

ANDERSEN, D. G. Mayday: Distributed Filtering for Internet Services**.** In: USENIX SYMPOSIUM ON INTERNET TECHNOLOGIES AND SYSTEMS, 4., 2003. **Proceedings...** [S.l.:s.n.], 2003. p. 31-42

ARVIDSSON, J. **Taxonomy of the Computer Security Incident related terminology**. Disponível em: <http://www.terena.org/activities/tf-csirt/iodef/docs/i-taxonomy_terms.html>. Acesso em: jul. 2008.

CAMPELLO, R. S.; WEBER, R. F. Sistemas de Detecção de Intrusão: In: MACÊDO, R. J.; FARINES, J. (Org.). **Livro Texto dos Minicursos**: SBRC'2001. Florianópolis: UFSC, 2001. p 1-43.

CORREA, A. C. **Metodologia para análise comparativa de Sistema detecção de intrusão**. 2005. 92f. Dissertação (Mestrado em Engenharia da Computação) – Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo.

CERTBR: Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, Registro Br, São Paulo. **Estatísticas dos Incidentes Reportados ao CERT.br** Disponível em : <http://www.cert.br/stats/incidentes/> e <http://www.cert.br/stats/incidentes/2008-apr-jun/total.html>. Acesso em: nov. 2008.

COLE, E. et al. **Network Security Bible**. Indianapolis: Wiley, 2005. 660p.

CREATIVE COMMUNS**.** **Ataques a camada 3.** Disponível em: <http://www.numaboa.com/informatica/seguranca/165-exploits/714-ataques-camada3>. Acesso em: ago. 2008.

CREATIVE COMMUNS. **Ataques a camada 4**. Disponível em: <http://www.numaboa.com/informatica/seguranca/165-exploits/718-camada4>. Acesso em:ago.2008.

FLOYD, S.; PAXSON V. Difficulties in Simulating the Internet**. IEEE/ACM Transactions on Networking**, [S.l.], v.9, n.4, p.392-403, Aug. 2001.

JAVVIN TECHNOLOGIES. **ICMP Attacks**. Disponível: <http://www.javvin.com/networksecurity/ICMPAttacks.html>**.** Acesso em: nov. 2008**.**

JUNIOR, L. F. S. S. **Um framework baseado em grupos de agentes de software especializados em sistemas distribuídos para detecção de intrusão em redes de computadores.** 2006. Dissertação de Mestrado, Publicação 206/06. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF. Disponível em: < http://bdtd.bce.unb.br/tedesimplificado/tde_busca/arquivo.php?codArquivo=952>. Acesso em: nov. 2008.

KENDALL, K. **A database of computer attacks for the evaluation of intrusion detection systems**. [S.1.]: MIT, 1999. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=50A14E90CF967E1033F0B998E1CBD969?doi=10.1.1.15.7102&rep=rep1&type=pdf>. Acesso em: nov. 2008.

LAUFER, R. P. **Introdução a Sistemas de Detecção de Intrusão.** 2003**.** Disponível em: <http://www.gta.ufrj.br/grad/03_1/sdi/sdi-1.htm>. Acesso em: ago. 2008.

LIPPMANN, R. et al. **Evaluating intrusion detection systems**: The 1998 darpa off-line intrusion detection evaluation. [S.l.]: IEEE, 2000. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4DEE9A1D7E0ABC08196791E2CD83892E?doi=10.1.1.36.6039&rep=rep1&type=pdf>. Acesso em: nov. 2008.

LITT, S. Tripwire Overview. **Linux Productivity Magazine**, [S.l.], 2003**.** Disponível em: <http://www.troubleshooters.com/lpm/200304/200304.htm>. Acesso em: out.2008.

MELL, P. et al. **An Overview of Issues in Testing Intrusion Detection Systems.** [S.l.]: NIST, 2003. (NIST IR 7007). Disponível em: <http://csrc.nist.gov/publications/nistir/nistir-7007.pdf>. Acesso em: nov.2008.

MASSICOTE, F. et al. **Using a VMware Network Infrastructure to Collect Traffic Traces for Intrusion Detection Evaluation**. [S.l.]: 2006. Disponível em: <http://cg.scs.carleton.ca/~mathieu/ACSAC06.pdf>. Acesso em: out. 2008.

MCHUGH, J. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. **ACM Transactions on Information and System Security**, New York, v.3, n.4, Nov. 2000.

NORTHCUTT, S. **Network Intrusion Detection**. 3rd ed. Indianapolis: New Riders, 2003. 489 p.

NAKAMURA, E. T.; GEUS, P.L. **Segurança de Redes em Ambientes Corporativos**. São Paulo: NovaTec, 2007.

O'REILLY MEDIA. **Third Party IP Network Scanning Methods**. Disponível em: <http://www.codewalkers.com/c/a/Server-Administration/Third-Party-IP-Network-Scanning-Methods/>. 2008. Acesso em: nov. 2008

PUKETZA N. et al. **A software platform for testing intrusion detection systems.** [S.l.], 1997. Disponível em: <http://citeseer.ist.psu.edu/cache/papers/cs/33082/http:zSzzSzwww.cs.ucdavis.eduzSzzCz7EolssonzSzpubszSz1997zSztids_software.pdf/puketza97software.pdf>. Acesso em: nov. 2008.

RUOMING, P. et al. The Devil and Packet Trace Anonymization. **ACM SIGCOMM Computer Communication Review**, New York,v.36, n.1, p.29-38, Jan. 2006.

RUPP, A. et al. Packet Trace Manipulation Framework for Test Labs**.** In: ACM SIGMOD INTERNET MEASUREMENT CONFERENCE, IMC, 4., 2004, Taormina, Sicily, Italy. **Proceedings...** New York: ACM, 2004. p. 251-256. Disponível em: <http://www.imconf.net/imc-2004/papers/p251-rupp.pdf>. Acesso em: out.2008.

SCHUBA, C.L. et al. **Intrusion Detection using dynamic tracing**. [S.l.], 2007. Disponível em: <http://www.freepatentsonline.com/y2007/0107058.html>. Acesso em: nov.2008.

SEEBERG,V.E. **Generation and use of test data sets in IDS testing**. [S.l.], 2005. Disponível em: <http://infosikring.dynalias.com/writings/Seeberg_IDS_testing.pdf>. Acesso em: nov. 2008.

SILVA, M. C; SARAIVA, M.N.S. **Estudo de sistemas de detecção de intrusão – Uma abordagem Open Source**. Disponível em: < http://mosel.estg.ipleiria.pt/files/Paper_-_IDS_&_Snort.pdf >. Acesso em: set. 2008.

SILVA, M.P.C.; SAMPAIO, M.N.S. **Estudos de Sistemas de Detecção de Intrusões Uma Abordagem Open Source.** 2006. 128 f. Relatório final da Disciplina Projeto I (Licenciatura em Engenharia em Informática e Comunicações) – Instituto Politécnico de Leiria, Escola Superior de Tecnologia e Gestão. Disponível em: <http://mosel.estg.ipleiria.pt/files/Estudo_de_Sistemas_de_Deteccao_e_Prevencao_de_Intrusoes.pdf>. Acesso em: nov. 2008.

SANS INSTITUTE. **Top-20 Internet Security Attack Targets**. [S.l.], 2006. Disponível em: <http://www.sans.org/top20/2006/top20-v70-portuguese.pdf>. Acesso em: nov. 2008.

SOMMERS, J. et al. A Framework for Malicious Workload Generation. In: CONFERENCE ON ACM SIGCOMM INTERNET MEASUREMENT, IMC, 4., 2004, Taormina, Italy. **Proceedings…** New York: ACM, 2004. p. 82-87. Disponível em: < http://www.imconf.net/imc-2004/papers/p82-sommers.pdf>. Acesso em: nov. 2008.

SOMMERS, J. et al. Toward **Comprehensive Traffic Generation for Online IDS Evaluation**. 2005. Disponível em: <http://pages.cs.wisc.edu/~pb/trident_final.pdf>. Acesso em**:** nov. 2008.

SOMMERS, J. et al. Recent Advances in Network Intrusion Detection System Tuning. In: IEEE CONFERENCE ON INFORMATION SCIENCES AND SYSTEM, 2006. **Proccedings...** [S.l.]: IEEE, 2006. p. 1490-1495.

SOMMERS J.; BARFORD P. Self-Configuring Network Traffic Generation. In: CONFERENCE ON ACM SIGCOMM INTERNET MEASUREMENT, IMC, 4., 2004, Taormina, Italy. **Proceedings…** New York: ACM, 2004. p. 68–80. Disponível em: < http://pages.cs.wisc.edu/~jsommers/pubs/p68-sommers.pdf>. Acesso em: nov. 2008.

TANEMBAUM, A. S. **Redes de Computadores**. Rio de Janeiro: Campus, 2003. 945p.

TCP/IP Suite Weaknesses: Telecommunications & Internetworking. Disponível em: <http://mudji.net/press/?p=152>. Acesso em: out. 2008.

TECH FAQ**. Compreendendo Ataques de redes.** Disponível em: <http://www.tech-faq.com/lang/pt/network-attacks.shtml&usg=ALkJrhiKs0OlGpzZPdH2JJaZc4Fw4-3Atw>. Acesso em**:** jul.2008.

WANG, J.;ZHAO, L. **Experimental Design for Attack Scenario Traces to Validate Intrusion Detection Alert Correlations. University of Pensilvania.** Singapore: Management University, 2006.

VÍRUS de computador. Disponível em: <http://pt.wikipedia.org/wiki/V%C3%ADrus_de_computador>. Acesso em**:** nov. 2008.