FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA DO ESTADO DO RIO DE JANEIRO

INSTITUTO SUPERIOR DE TECNOLOGIA EM CIÊNCIA DA COMPUTAÇÃO

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA

SAULO VITOR BORBA EVANGELISTA

SISTEMAS DE DETECÇÃO DE INTRUSOS E SISTEMAS DE PREVENÇÃO DE INTRUSOS: PRINCÍPIOS E APLICAÇÃO DE ENTROPIA

PETRÓPOLIS
2008

SAULO VITOR BORBA EVANGELISTA

SISTEMAS DE DETECÇÃO DE INTRUSOS E SISTEMAS DE PREVENÇÃO DE INTRUSOS: PRINCÍPIOS E APLICAÇÃO DE ENTROPIA

Trabalho de conclusão de curso apresentado ao Instituto Superior de Tecnologia – IST, como requisito parcial para obtenção de título de tecnólogo em Tecnologia da Informação e da Comunicação.

ORIENTADOR: Prof. Fábio Borges

PETRÓPOLIS
2008

SAULO VITOR BORBA EVANGELISTA

SISTEMAS DE DETECÇÃO DE INTRUSOS E SISTEMAS DE PREVENÇÃO DE INTRUSOS: PRINCÍPIOS E APLICAÇÃO DE ENTROPIA

Trabalho de conclusão de curso apresentado ao Instituto Superior de Tecnologia – IST, como requisito parcial para obtenção de título de tecnólogo em Tecnologia da Informação e da Comunicação.

Aprovado em de de 2008

BANCA EXAMINADORA

_____

Prof. Fábio Borges de Oliveira
Laboratório Nacional de Computação Científica – LNCC

_____

Prof. Wagner Vieira Leo
Laboratório Nacional de Computação Científica – LNCC

_____

Prof. Paulo Cabral Filho
Laboratório Nacional de Computação Científica – LNCC

_____

Prof. Luis Rodrigo Oliveira Gonçalves
Laboratório Nacional de Computação Científica – LNCC

PETRÓPOLIS
2008

À minha família que muito me incentivou para conclusão deste trabalho.

Agradecimentos

Ao Professores Fábio Borges, Wagner Vieira, Paulo Cabral e Luis Rodrigo;
Aos Professores que também contribuíram para minha formação e às pessoas que de alguma maneira ajudaram para conclusão deste trabalho.

Quando Cérbero, a mais vil das bestas, nos descobriu, a boca escancarou, exibindo os dentes e outros membros, raivoso, agitado.

Dante Alighieri, A divina Comédia.

Resumo

A finalidade de um Sistema de Detecção de Intrusos é detectar uma invasão e a de um Sistema de Prevenção de Intrusos é de detectar e bloquear uma invasão. Para implementação de um sistema de detecção encontramos algumas dificuldades como os conflitos gerados com outros meios de segurança como criptografia e redes com switches. Existem hoje no mercado vários programas de detecção de intrusos, tais como o Snort, programa confiável e de fácil instalação, e equipamentos de prevenção de intrusos como os Appliances. Um novo conceito está sendo estudado para novas implementações mais eficazes para a detecção de intrusos, que é a introdução de Entropia em sistemas capazes de detectar intrusos, onde a Entropia é calculada para medir os níveis de distribuição de tráfego e assim analisar se há alguma anomalia no trafego de rede.

Palavras-chaves

Sistemas de Detecção de Intrusos, Sistemas de Prevenção de Intrusos, Snort, Appliances, Entropia.

Abstract

The purpose of an Intrusion Detection System is to detect an invasion and a system of Intrusion Prevention is to detect and block an invasion. To implement a system to detect find some difficulties as the conflicts generated by other means as security such encryption and networks with switches. There are various programs on the market today for detecting intruders, such as Snort, reliable and easy to program installation, and equipment for preventing intruders such as appliances. A new concept is being studied for new deployments more effective for detecting intruders, that is the introduction of entropy in systems that can detect intruders, where the entropy is calculated to measure the levels of distribution of traffic and thus examine whether there is an anomaly in the traffic network.

Índice

1 Introdução

O conceito de *Intrusion Detection System* (IDS) surgiu nos anos 80 em estudos do *Stanford Research Institute*. Conhecido como *Project* 6169 - *Statistical Techniques Development For An Audit Trail System*, o projeto utilizava um algoritmo de alta velocidade que analisava os usuários com base nos seus perfis de comportamento.[1]

A partir dos IDS surgiu o *Intrusion Prevention System* (IPS) sistema que além de detectar ataques, interrompe, e também será mencionado no trabalho. O IDS fornece uma camada extra de proteção para um sistema computacional. Ele auxilia na proteção da rede. Suponhamos que temos uma rede bem montada com *firewalls*, roteadores e *switches* colocados em locais bem estudados, pois bem, pode-se achar que temos segurança, mas não é bem assim, toda essa tecnologia precisa de configuração e às vezes não são tão bem configuradas pelo administrador do sistema. É aí que entra o IDS para fornecer essa proteção extra de que falamos. Em resumo IDS são ferramentas automatizadas e inteligentes para detectar tentativas de intrusão em tempo real e IPS são IDS que através de outros mecanismos vão interromper o invasor ou fazer algo para detê-lo.

O objetivo é explicar o que é um IDS e também mencionar os IPS, dando exemplos dos sistemas disponíveis no mercado e fazer uma análise dos mesmos, e por fim dar ênfase na utilização de Entropia em sistemas de detecção. Na introdução conceitua-se IDS e menciona-se a sua importância em uma rede de computadores. Dando início ao trabalho é feita uma sucinta abordagem dos protocolos para internet modelo *Open Source Interconnection* (OSI) e modelo *Transmission Control Protocol / Internet Protocol* (TCP/IP). Vale salientar que o IDS trabalha em cima do modelo OSI. No desenvolvimento é feito um comparativo do conjunto IDS/IPS analisando com mais aprofundamento estes sistemas e abordado funcionamento e composição, implementação e tipos de IDS/IPS. São mostrados os problemas na implementação desses sistemas tais como: IDS com *Secure Socket Layer* (SSL), redes com *switches*, redes de alta velocidade e etc. É feita a abordagem da Segunda Lei da Termodinâmica para o entendimento do conceito de Entropia e mostrado a utilização de Entropia num IDS. No último capítulo o Snort é detalhado, IDS muito utilizado devido a sua facilidade e bom desempenho, e é feito uma exposição de equipamentos e programas e suas características, chamando a atenção para os appliances como IPSs.

2 Camadas de Rede e Protocolos

Para falarmos de IDS/IPS torna-se necessário falarmos dos protocolos de Internet, como se relacionam e os modelos que formam, pois o IDS vai agir diretamente nas camadas desses modelos.

Os protocolos para internet são um grupo de protocolos de comunicação caracterizados como pilhas, que são padrões onde a Internet e a maioria das redes funcionam.

Existem dois padrões para internet mais conhecidos e utilizados. Um é o OSI e o outro é o TCP/IP, nomeado dessa forma, pois o protocolo TCP e o protocolo IP são os mais importantes. É de costume se comparar o modelo OSI com o TCP/IP, mas estes possuem algumas diferenças. O modelo TCP/IP é uma forma reduzida do modelo OSI. O primeiro tem cinco camadas e o segundo sete. Portanto, as camadas do modelo OSI não são iguais ao modelo TCP, existem algumas diferenças de funções.

O IDS baseado em rede, que será visto posteriormente, opera sobre camadas de rede do modelo TCP e do modelo OSI, o qual será explicado a seguir.

2.1 Modelo OSI

Para facilitar a comunicação entre computadores foi criado o padrão OSI com o objetivo de que diferentes máquinas funcionando com diversos sistemas pudessem se “entender”.

Cada camada desse modelo tem suas configurações de regras e protocolos para codificar e decodificar os dados que passam por essas camadas. Para se enviar uma informação os dados começam pela camada de aplicação e são passados para camadas de baixo, tendo cada camada instruções especificas, até que chegue a camada física. Para o recebimento o inverso é feito.

Este modelo compõe-se de sete camadas que são: física, enlace, rede, transporte, sessão, apresentação e aplicação.Veja figura 1:

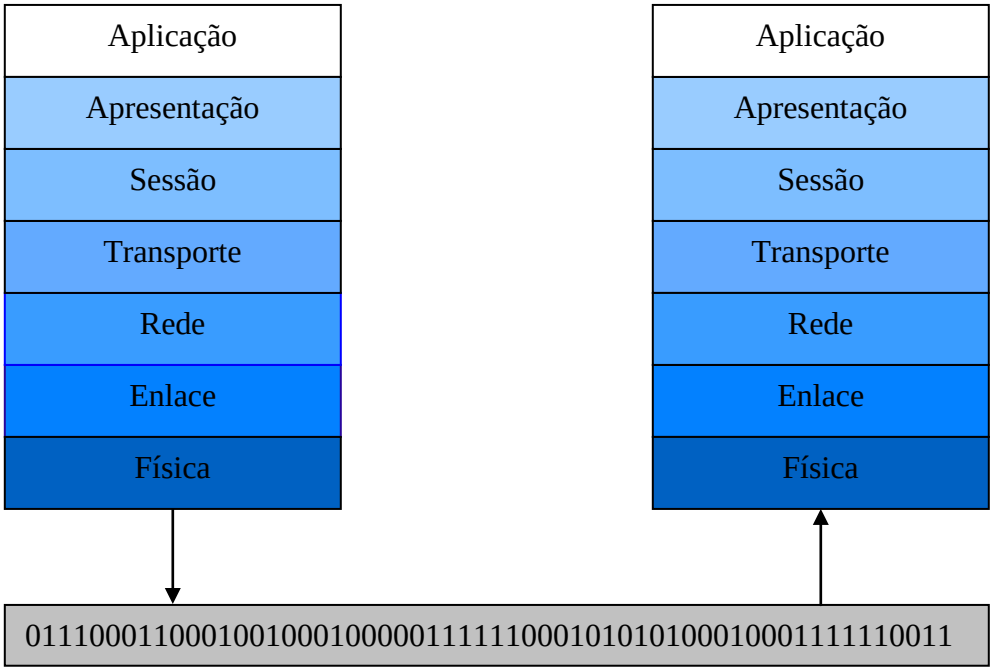


Figura 1: Modelo de comunicação OSI. A camada Física é onde teoricamente os dados estão mais próximos dos impulsos elétricos [2].

2.1.1 Camada Física

Camada de mais baixo nível do protocolo é ela quem define as características técnicas dos dispositivos elétricos. É quem controla a velocidade da transmissão, define as características elétricas e faz o controle de acesso. Este controle de acesso pode ser: centralizado ou distribuído. No modo centralizado, uma máquina controla o acesso à rede, um exemplo seria a topologia estrela. Já no modo distribuído todas as máquinas podem fazer o controle de acesso. Um exemplo seria a rede em anel).

2.1.2 Camada de Ligação de dados ou enlace

Esta camada detecta e corrige os erros que vieram da camada física ordenando os quadros. Faz a transmissão e recepção destes quadros e controla o fluxo de dados. Esta camada também trabalha com um protocolo de comunicação. Dentre os existentes temos: LAPB, PPP e NetBios.

2.1.3 Camada de Rede

Faz o endereçamento dos pacotes. Todos os endereços que eram lógicos passam a ser físicos. Define rota para que o pacote chegue ao destino fazendo análise de tráfego e definido qual é o melhor caminho. Também faz a fragmentação de pacotes, controle de congestionamento e sequenciamento de pacotes.

2.1.4 Camada de Transporte

Na transmissão esta camada pega os dados enviados pela camada de sessão e divide em pacotes. Na recepção esta camada pega os pacotes da camada de rede e reconstitui o dado para ser entregue a camada de sessão.

Esta camada faz a ligação das camadas de aplicação (níveis 5 a 7) e de nível físico (1 a 3). A camada de transporte pode trabalhar em dois modos: orientado a conexão e não orientado a conexão. Um exemplo de protocolo orientado a conexão é o TCP e não orientado temos o UDP. O modo orientado a conexão é mais confiável, pois, permite integridade e a correta seqüência ou ordenação dos dados.

2.1.5 Camada de Sessão

Esta camada permite que aplicações diferentes em diferentes computadores possam se comunicar. Também faz marcações nos pacotes para que caso a rede tenha problemas de comunicação, os dados sejam transmitidos de onde foram interrompidos.

2.1.6 Camada de Apresentação

Faz a conversão do formato do dado recebido pela camada de aplicação em um formato entendido pelo protocolo usado. Faz também a compactação dos dados e pode trabalhar também com algum tipo de criptografia que será descriptografado e descompactado na camada de apresentação do computador receptor.

2.1.7 Camada de Aplicação

Esta camada faz a ligação entre os aplicativos e o protocolo de comunicação utilizado. Por exemplo, entre um aplicativo de e-mail e o protocolo de comunicação responsável por este serviço.

2.2 Modelo TCP/IP

O modelo TCP/IP faz a união de camadas como é o caso das camadas de enlace e da camada física mudando o nome para interface de rede ou somente física ou enlace. Isto por sinal simplifica a entrega dos pacotes visto que a camada física define o tamanho dos frames da camada de enlace. O protocolo TCP não tem todos os recursos do modelo OSI, e nem se comunicam diretamente, porém a figura 2 visa apenas mostrar a mudança dos nomes das camadas e a redução do número de camadas no protocolo TCP. Uma das principais diferenças entre esses dois modelos é que o TCP não possui criptografia, por isso é introduzida uma forma de criptografia extra, o SSL.

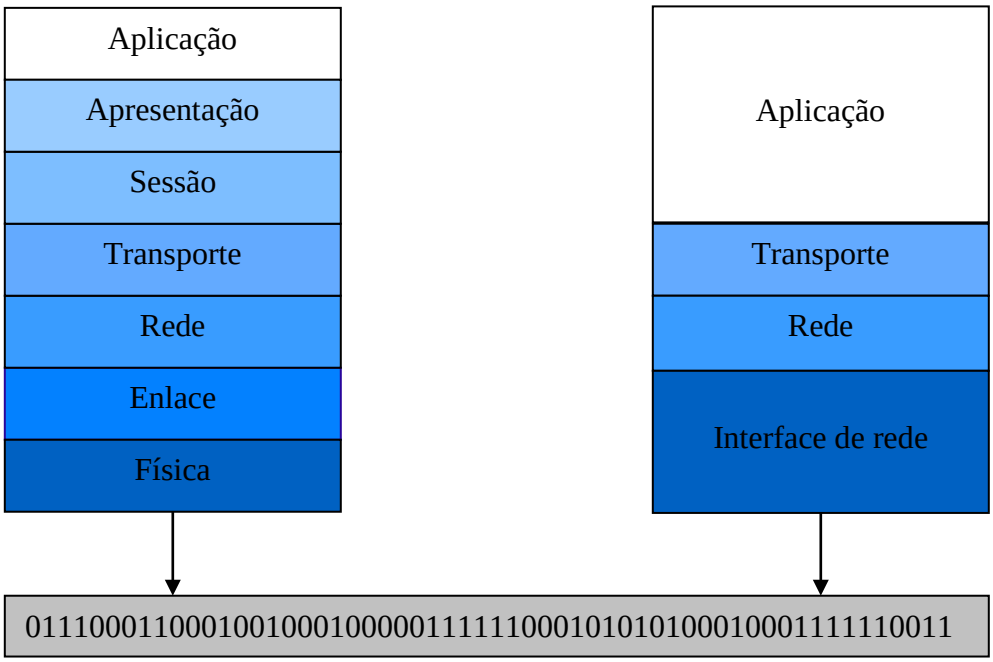


Figura 2: Modelo de Comunicação TCP/IP, comparativo com modelo OSI. A camada de interface de rede é a mais próxima teoricamente, dos impulsos elétricos [2].

Este padrão é composto de quatro camadas: Aplicação, Transporte, Internet ou Rede e Interface de rede.

A camada de Interface de rede utiliza o padrão Ethernet, a camada de rede o protocolo IP, a camada de Transporte pode ter muitos padrões utilizáveis como UDP e TCP, e a camada de aplicação utiliza, por exemplo, os protocolos HTTP (navegação na World Wide Web), FTP (transporte de arquivos), SMTP (envio de email) e SSH (login remoto seguro).

Para que seja feita a comunicação entre dois pontos na rede, sempre que os dados passam de uma camada para outra no modelo de protocolo eles são passados com um cabeçalho que permite a interação ou entendimento entre estas camadas. Por exemplo, os dados da camada zero ou camada de interface de rede iniciam o pacote com o cabeçalho para a camada um ou camada de rede. Os dados da camada um iniciam o pacote com o cabeçalho para a camada dois ou camada de transporte e assim sucessivamente. Veja figura 3:

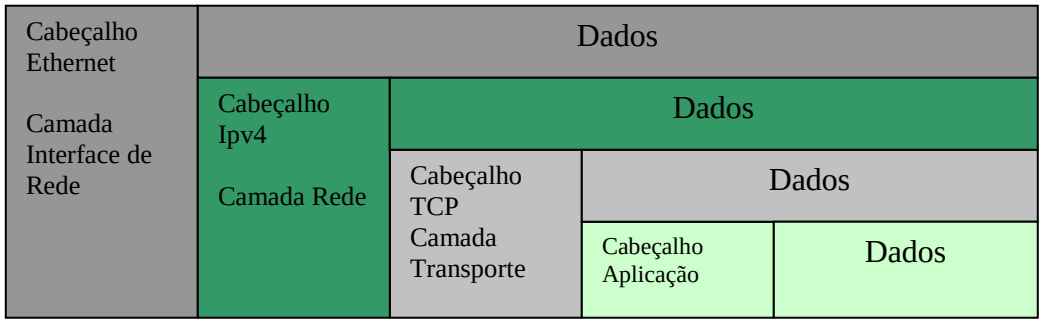


Figura 3: Padrão de pilha TCP/IP [2].

3 IDS e IPS - Comparativo

Neste capítulo estaremos explicando em conjunto IDS com IPS com o objetivo de esclarecer o entendimento de cada sistema. Estes sistemas serão mostrados tanto no que diz respeito a conceitos, quanto a funcionamento e implementação. Inicialmente segue noções básicas de intrusão para em seguida atingirmos o objetivo do capítulo.

3.1 Intrusos

Intrusos são os invasores de um sistema e se classificam de três maneiras:

- Mascarado (invasor de fora da rede ou sistema): Invasor que não está autorizado a entrar no sistema e o invade para obter privilégios de um usuário legítimo;
- Infrator (invasor de dentro da rede): É um usuário real ou legítimo que não está autorizado a usar determinados recursos, mas os utiliza, ou então, que está autorizado, mas não os utiliza de forma lícita;
- Usuário clandestino (invasor de dentro ou de fora da rede): Invasor que toma posse de privilégios de administrador de um sistema para que se esquive de auditorias e controles para acesso ou até mesmo para ludibriar provas auditórias contra este.

3.2 Técnicas de Intrusão

O principal objetivo de um intruso é aumentar alguns privilégios dentro de um sistema. Porém, esses privilégios são protegidos por senhas de usuário. Mas se um invasor tem acesso a estas senhas ele pode obter estes privilégios. O administrador mantém essas

senhas em um arquivo, por isso é necessário que este esteja bem protegido. Abaixo segue duas maneiras de se proteger esse arquivo de senhas:

Função unidirecional ou não reversível: A partir da entrada de uma senha o sistema a transforma em um valor de tamanho fixo que não pode ser revertido. Assim, o sistema não necessita de armazenar as senhas, mas somente os valores gerados a partir das mesmas.

Controle de acesso: Neste caso se limita ao extremo o acesso ao arquivo de senhas.

Agora descreveremos algumas técnicas que invasores utilizam para a descoberta de senhas:

1. Tentar senhas padrão geralmente entregues com o sistema;
2. Tentar senhas curtas;
3. Tentar palavras de dicionários *on-line;*
4. Tentar senhas com informações sobre seus usuários;
5. Tentar números de placa de automóveis;
6. Usar Cavalo-de-Tróia para ludibriar restrições de acesso;
7. Utilizar-se de escutas clandestinas entre acesso remoto e sistemas fixos;

Até o quinto método podemos ver que são métodos de tentativas e erros para descoberta de senha, porém são de fácil defesa para os administradores de sistemas. Já o sexto item é um pouco mais complicado, pois se trata de fazer com que um usuário legítimo instale um Cavalo-de-Tróia em sua máquina que fará com que o invasor obtenha acesso, por exemplo, a um arquivo de senhas que antes estaria protegido por criptografia, mas que agora está descriptografado, pois é um arquivo com privilégios para o usuário. Por fim, o sétimo item pode ser resolvido com criptografia do *link*.

As técnicas de invasão não se baseiam apenas na descoberta de senhas, mas também em várias outras maneiras, tais como sobrecarga de um sistema com envio constante de solicitações.

3.3 O que é IDS?

O IDS tem por finalidade detectar uma ameaça ou intrusão na rede. Pode se dizer por analogia que o IDS é como se fosse um alarme de um carro que soa quando alguém abre sua porta.

A monitoração e a detecção de intrusos eficientes são tão importantes quanto chaves e cadeados em nossas casas, assim como *firewalls* em nossas redes.

Imaginemos que tenhamos um servidor Web conectado a internet e queremos que clientes tenham acesso às páginas de Web. Pois bem, temos que pensar em segurança para não haver alteração por parte de invasores. Um *firewall* ou sistema de autenticação poderão prevenir os acessos sem autorização, mas às vezes pode ser quebrada a regra do *firewall* ou do sistema de autenticação. Sendo o IDS um sistema implementado na rede para alertar tentativas de acesso sem autorização aos computadores, este pode auxiliar ou complementar a segurança de um sistema já implementado com um *firewall*.

3.4 O que é IPS?

O IPS complementa um IDS bloqueando a intrusão e impedindo um dano maior para a rede. É uma ferramenta que detecta e bloqueia o invasor. Como foi dito o IDS é como se fosse um alarme de um carro que somente soa quando alguém abre sua porta. Já o IPS dispara o alarme e também trava as rodas para que o invasor não leve o carro.

Uma boa forma de se obter segurança em uma rede é fazer a prevenção de invasões. Porém para a instalação de um IPS deve-se levar em conta que temos um sistema limpo, ou seja, não esteja comprometido e deve-se possuir um conhecimento amplo do estado do sistema para que não se tenha problema posterior.

Para que o administrador da rede possa tomar alguma atitude quanto a uma invasão, deve-se obter informações no momento exato da invasão. A partir da detecção, um IPS executará ações para interromper o ataque e evitar ataques futuros. Essas ações podem ser desde o cancelamento de uma conexão até uma reconfiguração do *firewall* para interromper o ataque.

3.5 Terminologia referente a IDS/IPS

- **Alertas/Eventos**

É um aviso gerado pelo IDS quando este detecta determinada invasão. Este alerta pode ser dado tanto local quanto remotamente.

- **Evasão**

É um ataque ao IDS sem que este detecte o mesmo, é uma maneira que se encontra de ludibriar o sistema.

- **Fragmentação**

Fragmentação é uma maneira de dividir os pacotes de tal forma que não ultrapasse o limite da rede. A fragmentação é utilizada para a evasão ou também em ataques de negação de serviço.

- **Assinaturas**

Assinaturas são ataques conhecidos. Através das assinaturas ou regras pode-se gerar os alertas para atividades suspeitas. É feita comparação dos dados com as assinaturas e aí são gerados os alertas.

3.6 Por que usar um IDS?

Podemos nos perguntar por que usar um IDS. A resposta mais direta seria proteger os dados e a integridade do sistema. Para se ter proteção de integridade quanto a intrusos na Internet, somente senhas e segurança de arquivos não são suficientes. Devemos ter um bom sistema de segurança para a proteção dos dados. Não podemos somente entrar na Internet e achar que ninguém vai invadir. É importante que o sistema previna acessos a arquivos críticos ou bancos de dados de autenticação (como o NT SAM do Windows NT ou o Unix */etc/passwd*) exceto por administradores de sistemas autorizados [3].

3.7 IDS - Funcionamento e composição

O IDS faz análises na rede e no Sistema Operacional, verificando as atividades dos usuários, excesso de conexões, volume de dados, serviços de rede e etc. Esses dados são guardados em uma base de dados para que posteriormente de acordo com a configuração do sistema este possa alertar uma intrusão ou ameaça.

As ferramentas de IDS detectam diversos tipos de situações tais como:

- Scans: verifica se há portas do sistema que se encontram abertas;

- Ataques de comprometimento: o invasor obtém um *Shell* (terminal) com privilégios de *root* (superusuário, permite fazer qualquer tipo de alteração no sistema) para explorar vulnerabilidades;
- Ataques *Denial of Service* (DoS): é enviado um grande número de pacotes para sobrecarregar o desempenho do sistema comprometido [4].

De uma forma geral um IDS é composto dos seguintes elementos:

- Um dispositivo de acumulo de informações: Esse dispositivo deve ser capaz de colher dados. Por exemplo, ele deve ser capaz de detectar mudanças em um disco rígido, capturar pacotes em uma rede etc;
- Um mecanismo para monitoração de processos**:** Um IDS deve ser capaz de monitorar a si mesmo e a rede a qual está protegendo, fazendo verificações constantes, para que possa enviar informações para o administrador. O Snort, programa de IDS, pode avisar que ouve um problema na rede através de mensagens que no caso seriam enviadas para o arquivo */var/log/messages*;
- Capacidade de armazenamento de informações**:** A partir do momento que as informações foram capturadas pelo dispositivo de acumulo de informações essas informações devem ser armazenadas em algum lugar;
- Dispositivo de controle e comando: No que se diz respeito a controle e comando o IDS deve ser fácil de controlar seu comportamento;
- Um dispositivo de análise: Deve-se ter um dispositivo de análise para o administrador poder analisar seu acervo de dados utilizando um aplicativo.

3.8 Estratégias para IDS

3.8.1 Aplicativos IDS baseados em regras ou assinaturas

Este tipo de IDS é geralmente mais fácil de instalar. Para se ter um IDS baseado em assinaturas eficiente basta fazer com que o IDS carregue todas essas assinaturas e que se faça uma constante atualização destas. Assim, o IDS será capaz de detectar os ataques à rede.

Essas assinaturas são ataques reais que foram identificados. Já uma regra é uma linha de código que informa ao IDS sobre determinada assinatura.

3.8.2 Aplicativos IDS baseados em anomalias

Esse método é bem trabalhoso e não é tão seguro assim, pois o que é feito é uma coleta dos dados passados pela rede. O sistema reúne informações da atividade da rede e forma uma base de dados. A partir daí o sistema faz comparações das ocorrências da rede com essa base de dados e alerta sobre atividades que estão fora do que de costume, ou de normal acontece na rede. No entanto, se torna difícil configurar um sistema especificando o que é normal e o que é anormal em se tratando de tráfego de rede.

Pode-se pensar que uma constante detecção de invasão é o mais eficaz, porém o problema é se fazer análise de todas essas informações. Por isso o que se aconselha é fazer análises em intervalos de tempo, ou seja, baseadas em intervalos.

Esses intervalos podem ser no horário que não tiver expediente, e também em intervalos aleatórios durante o expediente. Todas essa informações geradas podem ser gravadas em uma base de dados de arquivos pequenos com alguns Mega Bytes.

3.9 Tipos de IDS

3.9.1 Host Based Intrusion Detection System (HIDS)

O IDS de *Host* é instalado em determinada máquina para avaliar o próprio *host*. Analisa os eventos do sistema operacional, eventos de acesso e eventos de aplicação, monitora as entradas, ou qualquer outra parte que represente tentativa de intrusão. Bloqueia também ataques que não são detectados pelo *firewall* como, por exemplo, protocolos criptografados. O IDS também acusa uma tentativa suspeita como um usuário tentando utilizar algo que ele não tenha permissão.

Existem dois tipos de aplicativos IDS baseados em *host*:

- Analisadores de eventos (listagem das ocorrências em uma rede ou num computador): Procura por conexões abertas de rede e monitoram portas do sistema;
- Analisadores de unidades de disco do sistema: Analisa unidade de disco e outros periféricos do sistema e cria uma base de dados. Essa base de dados é como se fosse

a situação original do sistema e sempre que ocorrer uma mudança o IDS pode gerar um alerta ou registrar a mudança.

3.9.2 Network Based Intrusion Detection System (NIDS)

O NIDS é instalado em um segmento de rede onde através de uma base de dados faz comparações necessárias com os pacotes de rede ou então faz a decodificação e verifica os protocolos de rede. O NIDS verifica os usuários externos não autorizados a entrar na rede, DoS ou roubo de base dados.

O IDS baseado em rede opera sobre as camadas de rede do modelo (OSI/RM). Esse tipo aplicativo baseado em rede é bem interessante quando se quer analisar o tráfego da rede.

Um IDS baseado em rede se torna muito mais eficiente e de fácil controle pelo administrador quando se utiliza vários servidores para aplicação do IDS. Um servidor para captura de dados, outro para monitoração e armazenamento e um para análise atenderia à necessidade de processamento e armazenagem dos dados. O servidor sensor detecta os dados que passam pela rede e os envia para o servidor de armazenagem, este envia para o servidor de análise o arquivo que contém os pacotes enviados pelo sensor. O servidor de análise pode então ler os pacotes onde estão armazenados ou selecionar os eventos da estação de armazenagem.

O dispositivo de armazenagem pode deixar todos os eventos prontos para serem enviados através de um servidor Web. O servidor para análise pode ser um servidor Linux com um navegador de Web. O administrador poderá utilizar o navegador Web para acessar o servidor Web do dispositivo de armazenamento. O administrador pode ainda utilizar um *Secure Shell* (SSH) para acessar diretamente os eventos, ou seja, à base de dados.

Vale salientar que um IDS precisa de muito processamento para seu funcionamento e os arquivos de registros precisam de grande quantidade de espaço no disco rígido. Assim deve se levar em consideração, a idéia de dividir o trabalho realizado por um IDS em servidores diferentes.

3.9.3 IDS Distribuído – Sistema de Detecção de Intrusos Distribuído (SDID)

Neste modelo são utilizados sensores NIDS localizado onde se proteja os servidores públicos e outros sensores analisando os *hosts* onde a rede é teoricamente mais confiável. Todos esses IDS se comunicam diretamente com uma estação de gerenciamento centralizada. Os *uploads* (envio de dados) dos eventos de ataque podem ser feitos através da estação de gerenciamento e armazenados em um banco de dados central. O *download* (retirada de dados) de assinaturas de ataque são feitos pelos próprios sensores. Cada sensor pode ter regras especificas para atender suas necessidades.

A comunicação entre os sensores e o gerenciador pode ser feita através de uma rede privada ou através da própria rede existente. Mas neste caso deve ser usada criptografia ou *Virtual Private Network* (VPN).

3.10 Tipos de IPS

Existem dois tipos de sistemas de prevenção de intrusos no mercado:

3.10.1 *Host-Based* (Baseados em host)

Os sistemas baseados em host são programas de prevenção de intrusos para serem instalados diretamente em computadores;

3.10.2 *InLine* (Em linha)

É todo dispositivo de hardware ou software habilitado a detectar e impedir ataques maliciosos, verificando anomalias.

3.11 Implementação de um IDS

Alguns fatores são importantes para implementação de um IDS:

- Política de Segurança - Para a implementação de um IDS é necessário uma política de segurança abrangente;
- Análise de custo - Existem vários IDS em código aberto, porém para implementação de um IDS com vários servidores há necessidade de maiores recursos;

- Pessoal de suporte - É necessário pessoal específico para implementação, manutenção e análise do IDS.

3.12 IDS - Serviços

3.12.1 Identificação de Tráfego

Um IDS deve sempre saber de onde veio a invasão, mostrando a porta e o endereço de origem e de destino.

A possibilidade de informar todos os detalhes de um pacote que trafega pela rede é o elemento mais importante de um sistema IDS que registra o tráfego de rede. Esses detalhes serão descritos a seguir:

Tipo de protocolo – O IDS informará se o pacote é UDP, TCP, ICMP e etc;

Origem – É o endereço de IP de origem;

Destino – É o endereço de IP de destino;

Porta de origem – Caso seja um pacote UDP ou TCP, o aplicativo dirá que porta o host de origem utilizou;

Porta de destino – É a porta de host de destino;

Checksums (tipo de analisador de integridade) – São os *checksums* que preservam a integridade dos pacotes transmitidos;

Número de seqüência – O IDS informa à ordem que os pacotes são gerados através dos números de seqüência. Essa ordem pode ser importante para entender a natureza de um ataque;

Informações sobre os pacotes – O IDS pode fazer pesquisas nos pacotes e analisar seus conteúdos.

3.12.2 Aplicação nos registros e definição de limites

Um IDS atualizado periodicamente coloca informações em um arquivo de registro ou em uma base de dados e define limites. Caso esse limite seja excedido o IDS poderá enviar um alerta.

Um IDS pode armazenar suas informações em diversos lugares:

- Arquivos de eventos do sistema – Mensagens podem ser enviadas para arquivos de registros já existentes como */var/log/messages e /var/log/security*, no Red Hat Linux;
- Arquivos de texto simples e diretórios – São diretórios e arquivos de textos que funcionam como */var/log/messages*, mas que são criados especificamente pelo IDS. Cada novo host detectado poderá ser nomeado com o endereço IP deste host. O IDS então separará os arquivos de acordo com o protocolo específico;
- Base de dados – São armazenadas as informações de maneira lógica e permite que sejam pesquisadas de forma eficiente. Depois de armazenadas essas informações elas podem ser passadas para um servidor Web e acessadas normalmente com um navegador Web.
- Alertas – Os IDS trabalham com alertas para chamar a tenção dos administradores para uma possível invasão. Esses alertas podem ser o envio de um evento a um arquivo de registro de alertas, um alerta a um sistema remoto ou o envio de um e-mail.

3.12.3 Configuração do sistema

Alguns aplicativos como o PortSentry oferece a possibilidade de reconfigurar o Sistema Operacional ou o *firewall* em caso de ataque;

3.12.4 Verificação de unidades de disco

Possibilidade de se obter uma imagem da rede e do Sistema Operacional e assim enviar alertas quando houver um evento anormal. É obtida uma imagem instantânea do sistema de arquivo e após é feita uma comparação com uma outra imagem tirada posteriormente. Aplicativos como o *Tripware* protegem o sistema contra os Cavalos de Tróia que são aplicativos projetados para parecerem legítimos.

3.13 Problemas comuns com IDS/IPS

3.13.1 Falsos Positivos

O IDS alerta determinada invasão, mas ela não existe, se trata de um alarme falso;

3.13.2 Falsos Negativos

O IDS não detecta uma intrusão, o sistema acha que o pacote é de fluxo normal do sistema;

3.13.3 Desenho da Arquitetura

Quando o tamanho da rede dificulta a implantação e controle do combinado IDS/IPS;

3.13.4 Freqüentes Updates

Há necessidade de que todo o sistema esteja atualizado para que se tenha defendido toda a infra-estrutura da rede.

3.14 Implementação na rede

É de costume colocar um NIDS antes do firewall para impedir que um usuário externo venha conhecer a topologia de rede e um depois do *firewall* na Zona Desmilitarizada (DMZ) para detectar algum ato que o *firewall* não tenha detectado. Coloca-se um também para detectar ataques advindos da rede interna e por fim HIDS para servidores de risco, tais como *WebServer* e servidores de email. Veja figura 4:

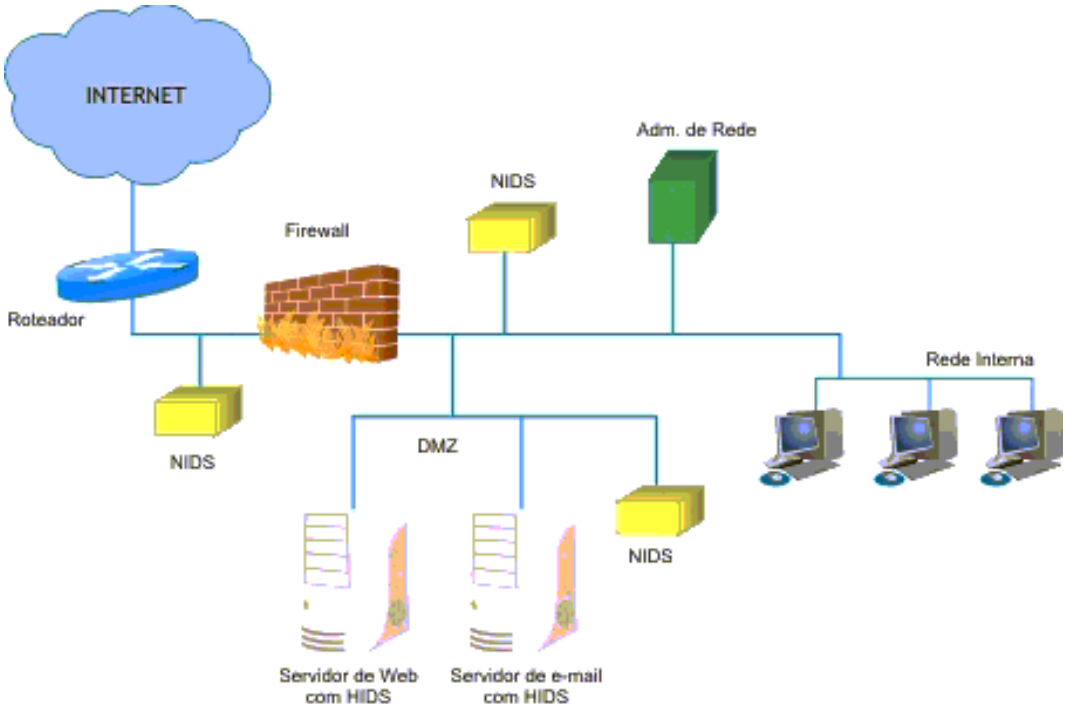


Figura 4: Modelo de arquitetura com NIDS e HIDS [5].

4 Desafios para um IDS

Existem muitos desafios para um IDS. O projeto ou adaptação de um IDS em ambientes diversos onde encontramos novas tecnologias de protocolos, tais como SSL e *Secure Internet Protocol* (IPSec), evoluções em infra-estrutura de redes comutadas e implantação de IDS em redes de alta velocidade com, como a tecnologia ATM, são algum deles. Existem também, outros problemas ou desafios, como a detecção de *Distributed Denial of Service* (DDoS) e a utilização de IDS em conjunto com firewalls. Diante dessa problemática será exposto aqui, algumas alternativas de implementação e abordado o tema de forma que o leitor tire conclusões se é viável ou não esta implementação.

4.1 IDS x SSL, IPSec

Sabe-se que um IDS faz monitorações tanto nos cabeçalhos dos pacotes quanto nos campos de dados. Porém, com a necessidade de sigilo e proteção dos dados que transitam pela rede se torna necessário o uso de criptografia, o que dificulta a utilização de IDS. Dados que antes seriam analisados pelo IDS e que poderiam estar sendo alvos de ataques podem vir a ser escondidos devido ao uso da criptografia.

4.1.1 SSL

Este protocolo é executado entre a camada de transporte e a de aplicação.Veja figura 5:

Aplicação	HTTP	LDAP	IMAP
	SSL (SECURE SOCKET LAYER)		
Transporte	TCP		
Rede	IP		
Interface de rede	Ethernet, PPP, Token Ring etc		

Figura 5: Localização do SSL nas camadas do protocolo TCP/IP.

A criptografia dos dados do pacote TCP faz com que todo conteúdo das conexões seja criptografado.

Os ataques na camada de aplicação são usados para invasão ou para DoS. Com a criptografia o IDS não terá como registrar o ataque, e nem enviar um pacote TCP RESET para ambos os participantes para terminar a conexão. Também não poderá interagir com o *firewall* para que este bloqueie endereços ou portas conforme configurado.

4.1.2 IPSec

O IPSec é uma extensão do protocolo IP empregado em implementações VPN. Ele trabalha com criptografia e assinatura digital.

Existem dois modos de funcionamento do IPSec: modo transporte e modo túnel. No modo transporte ele fornece proteção para protocolos de camada superior antecipadamente. Já no modo túnel os protocolos são empregados como um túnel de pacotes IP.

Existem dois protocolos: o *Authentication Header* (AH) e o *Encapsulating Security Payload* (ESP).

Um protocolo no modo transporte é parecido com o SSL, protegendo somente a porção de dados. Já o modo túnel criptografa todo o pacote IP. Veja figura 6 e 7:

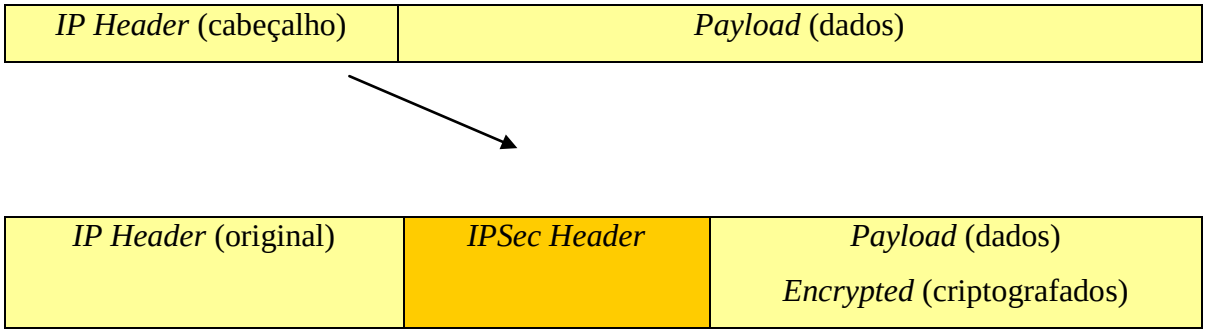


Figura 6: IPsec no modo de transporte.

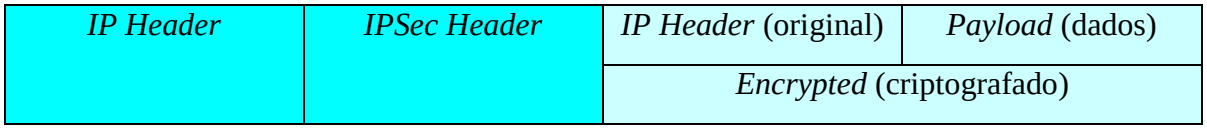


Figura 7: IPSec no modo túnel.

O AH provê integridade sem conexão, autenticação de dados, e um serviço para prevenção de reenvio de pacotes.

O protocolo ESP provê criptografia, limitado fluxo de tráfego confidencial e ainda as características do AH. A diferença é que o ESP não atua no cabeçalho dos pacotes.

Resumindo:

- Modo Transporte: protege somente porção de dados;
- Modo Túnel: protege cabeçalho e porção de dados;
- AH: protege cabeçalho e porção de dados;
- ESP: protege somente porção de dados

Em um modo máquina a máquina o monitoramento não é possível com IDS baseado em rede para fins de análise de dados, pois os mesmos estariam criptografados. Veja figura 8.

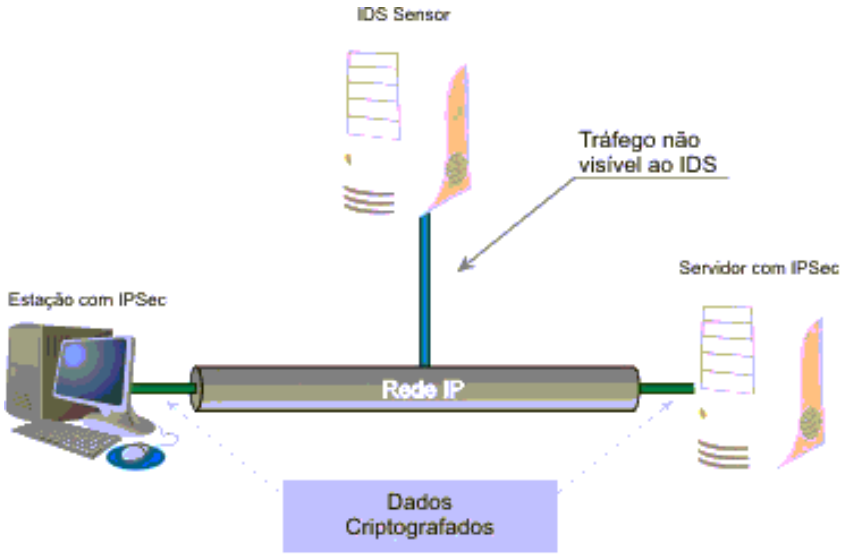


Figura 8: IPSec máquina a máquina [6].

Uma solução para o problema é colocar o IDS de rede após o dispositivo IPSec, como mostrado no IPSec gateway-a-gateway na figura 9.

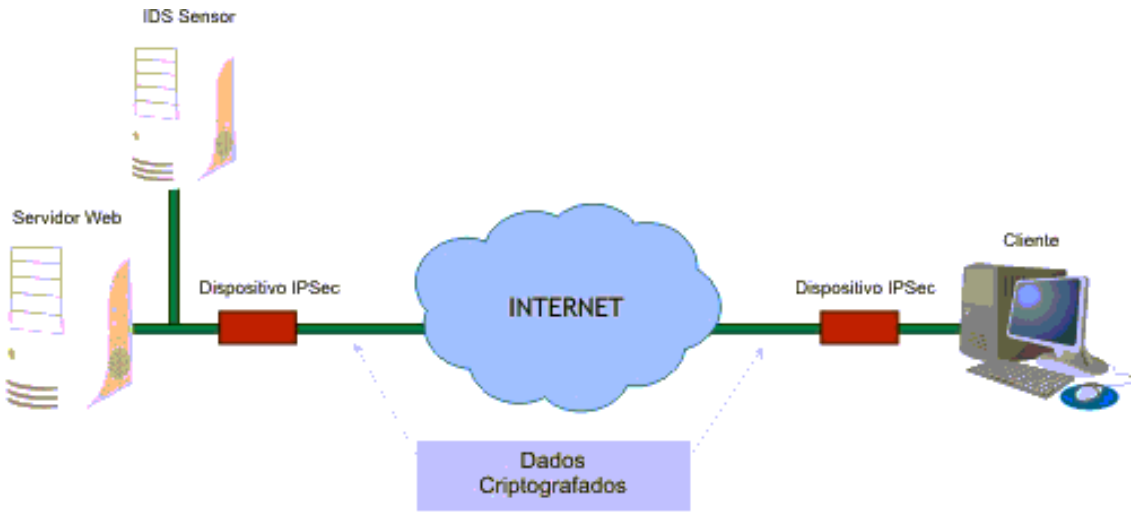


Figura 9: IPSec gateway-a-gateway [6].

4.2 IDS em redes com switches

O switch é um dispositivo que permite a comutação de dados, ou seja, somente o próprio destinatário recebe a mensagem, os outros usuários da rede não vêem tais dados. Essa medida aumenta em muito o desempenho da rede, porém dificulta a implantação e análise da rede com um IDS.

A seguir será mostrado três possibilidades de implementar um IDS em redes comutadas.

4.2.1 PortSPAN

Suponhamos que alguns dispositivos de rede tais como servidores e roteadores estejam ligados a um *switch* com velocidade baixa, então se coloca um IDS à porta *Switched Port Analyzer* (SPAN) de alta velocidade recebendo todo o tráfego do *switch*. Assim, toda a rede será monitorada.

Um exemplo de *switch* que disponibiliza essa possibilidade é o *switch* CataLyst da Cisco.

4.2.2 Splitting Wire/Optical Tap

A utilização de um *Splitting Tap* é a colocação de uma escuta para monitoração de tráfego. Uma boa idéia é a colocação de um *hub* entre o *switch* e o equipamento de rede para que seja enviada uma cópia do tráfego que passa pelo *hub* para o IDS. Veja figura 10.

Pode se utilizar este recurso tanto para cabos UTP, utilizado em redes Ethernet, como para fibras óticas em redes ATM. Neste caso, pode-se utilizar um dispositivo chamado *Optical Tap.* Veja figura 11.

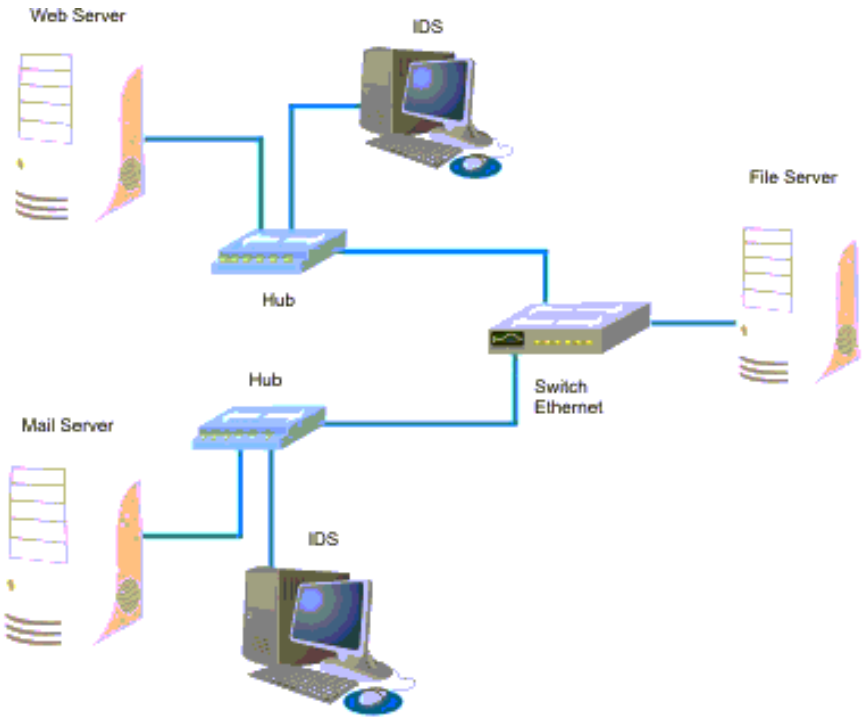


Figura 10: Monitoração de pacotes em rede Ethernet, utilizando *hubs* ou *wire tap*.[6]

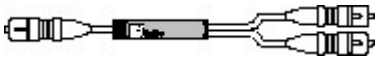


Figura 11: Optical Tap. [6]

4.2.3 Port Mirror

Esta opção consiste em fazer um espelhamento de uma porta para outra que serve de monitoração. Esta medida é um pouco inviável, pois se coloca apenas um dispositivo por IDS. Porém, o espelhamento de portas (*port mirroring*) em redes hierárquicas é o mais sensato. Veja figura 12.

Uma maneira de resolvermos problemas com relação a redes comutadas é a instalação de *host based*.

Sabemos que há duas variações de host based: detecção de anomalia/atividade suspeita e detecção de ataque baseado em rede. É interessante análise híbrida, porém é difícil encontrar um produto que faz as duas coisas: detecção de anomalias e tráfego de redes.

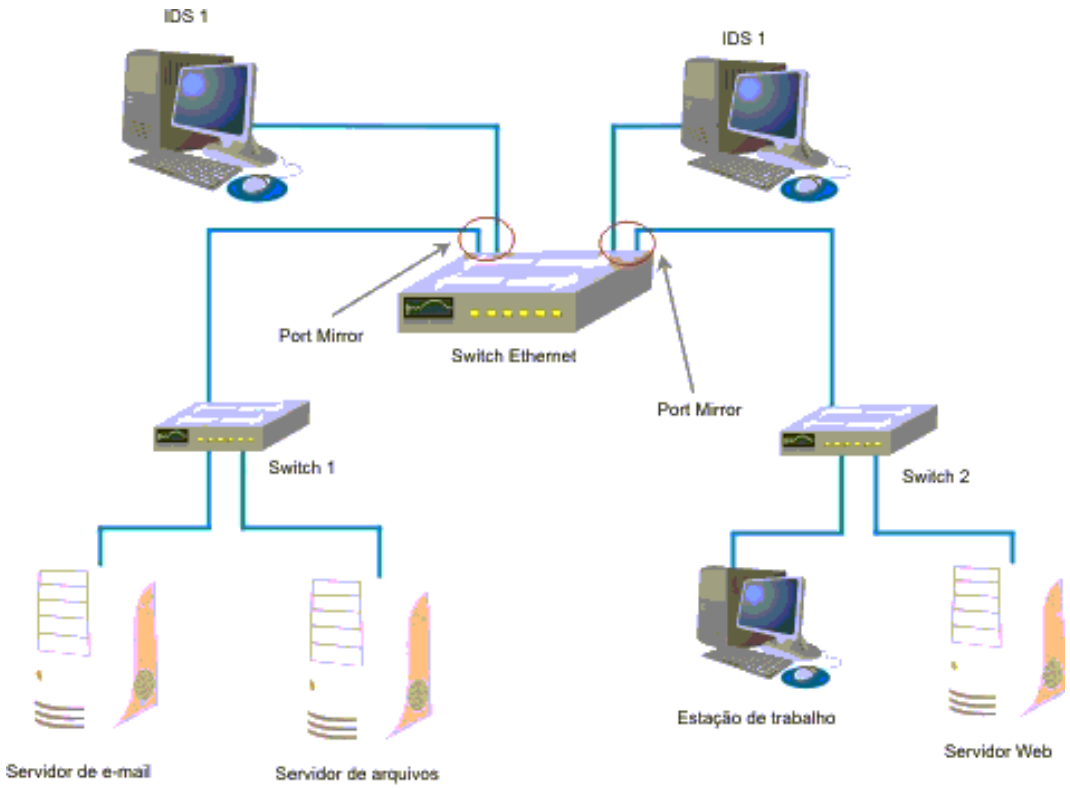


Figura 12: Switch com *port mirror* em uma estrutura hierárquica de *switches* sem *port span*. [6]

4.3 IDS em redes de alta velocidade

As redes de alta velocidade se tornam um problema devido à dificuldade para monitoração. Outro problema é o tamanho dos pacotes, pois influencia diretamente na análise pelo IDS.

Uma solução para problemas como estes em sistemas IDS é a separação do tráfego através de *switches* de balanceamento de tráfego para IDS. Esses switches *TopLayer* podem dividir uma porta Gigabit-Ethernet em várias portas *Fast-Ethernet* dividindo o tráfego da rede. Veja figura 13:

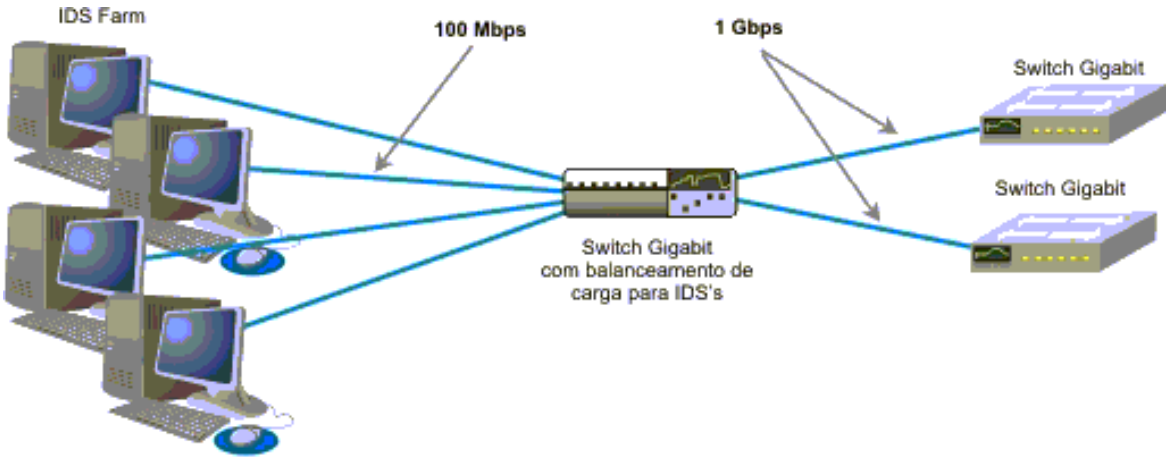


Figura 13: O tráfego entre os *switches* Gigabit é distribuído entre vários IDS [6].

Deve-se atentar para esse tipo de medida, pois é necessário fazer uma consolidação dos vários eventos gerados pelos sensores. Hoje em dia, existem vários ataques advindos dos mais diversos locais com endereço IP diferente visando um objetivo comum. Por isso, utilizando-se um sistema como o proposto acima se torna difícil descobrir esse ataque, pois ele seria diluído no ambiente de rede.

Uma outra abordagem é analisar somente elementos de interesse do administrador como, por exemplo, roteadores (implementação chamada de *Target* IDS). Outra opção é a segregação de IDS por serviço, onde um IDS é configurado somente para analisar eventos relativos a email, outro somente HTTP etc.

4.4 Distributed Denial of Service (DDoS)

Hoje há uma constante preocupação quando tratamos de segurança, principalmente nos ataques feitos por invasores que acabam afetando *sites* famosos, e é nesse meio que escutamos muito sobre negação de serviço ou DoS.

A idéia desse tipo de ataque é enviar um número excessivo de requisições a um computador alvo e tornar este indisponível para os serviços que são disponibilizados.

Intrusão Distribuída é a forma de se invadir determinado sistema utilizando máquinas espalhadas pelo mundo inteiro e através dessa união de máquinas fazer uma intervenção em conjunto no alvo em foco.

Os ataques de DDoS são resultantes da união de intrusão distribuída e negação de serviço. Computadores de diversos lugares trabalham em conjunto para inutilizar um alvo. Pode-se dizer que é um ataque DoS só que em escala muito maior, utilizando diversas máquinas.

Os primeiros ataques DDoS documentados surgiram em agosto de 1999, no entanto, esta categoria se firmou como a mais nova ameaça na Internet na semana de 7 a 11 de Fevereiro de 2000, quando vândalos cibernéticos deixaram inoperantes por algumas horas *sites* como o Yahoo, EBay, Amazon e CNN. Uma semana depois, teve-se notícia de ataques DDoS contra *sites* brasileiros, tais como: UOL, Globo e IG, causando com isto uma certa apreensão generalizada [7].

Outras repercussões causadas por ataques DDoS foram o caso do *site* da Alldas, da RIAA, da SCO e o *site* da Al Jazira. Isso mostra que os ataque DoS e DDoS geram prejuízos de formas incalculáveis, tanto na parte financeira como na parte das informações [8]. Assim a idéia aqui é falar do que se trata um DDoS e como um IDS pode auxiliar contra essa ameaça.

4.4.1 Entendendo o ataque

4.4.1.1 Personagens

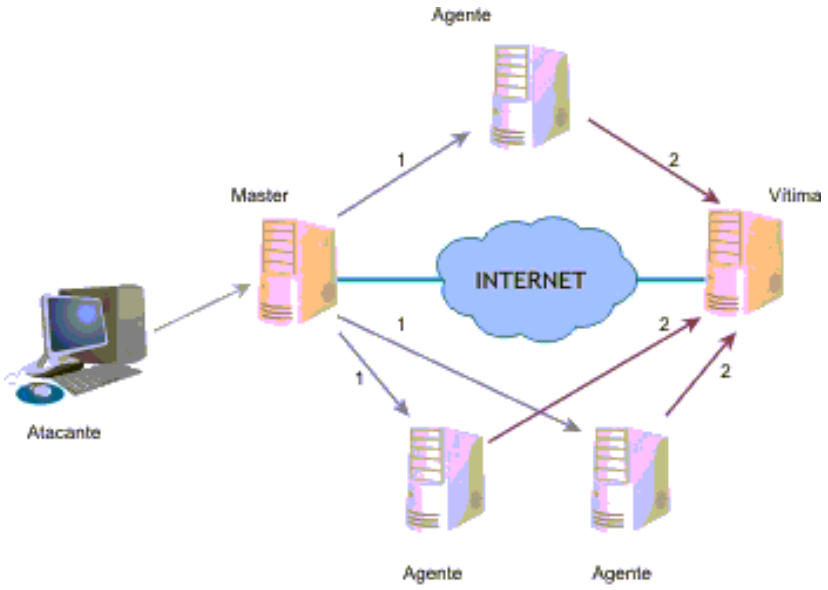


Figura 14: Ataque DDoS [7].

Para entendermos o funcionamento de um ataque vamos conhecer os personagens principais e usaremos a nomenclatura abaixo:

Atacante: É o invasor. O indivíduo que realmente planeja e coordena o ataque. Possui sua máquina individual que pode ser um computador de mesa ou um *Notebook*;

Master: Computador que é programado para comandar os agentes.

Agente: Computador que realmente realiza o ataque DoS contra um ou mais alvos ou vítimas, conforme o desejo do atacante.

Vítima: Máquina que recebe o ataque. É ocupada por um grande número de pacotes, sobrecarregando toda a rede e resultando na paralisação desta e conseqüentemente dos serviços oferecidos.

Além destes personagens existe ainda mais dois que também são importantes:

Cliente: Aplicação que está no master e que realmente tem o controle dos ataques e envia comandos aos *daemons*.

***Daemon*:** Processo que está sendo executado no agente, que recebe e executa os comandos advindos do cliente.

4.4.1.2 O ataque

Um ataque DDoS é feito em três fases: a primeira é quando se faz a intrusão em máquinas para se obter acesso privilegiado, ou seja, acesso de *root*. Na segunda são instalados softwares nestas máquinas invadidas para a montagem da rede de ataque. E a terceira é quando se envia um número grande de pacotes contra as vítimas, concretizando o ataque.

Fase 1: Obtendo acesso de *root* segue-se as seguintes etapas:

1. É feito um estudo das portas e das vulnerabilidades das redes alvo, ou seja, onde se quer fazer o ataque. Costuma-se focar redes de banda larga e com pouco monitoramento.
2. Em seguida o atacante explora as vulnerabilidades encontradas com o objetivo principal de obter o acesso de root;
3. Por fim com o endereço IP das máquinas invadidas é montada a rede de ataque.

Fase 2: Para a instalação do software DDoS segue-se as seguintes etapas:

1. É utilizada uma conta de um usuário para instalar as versões compiladas das ferramentas de ataque;
2. As ferramentas de DDoS sendo instaladas vão permitir agora que as máquinas sejam controladas remotamente. Essas máquinas que poderão ser agentes ou masters.
3. A definição de qual máquina será master e qual será agente quem escolhe é o atacante. As máquinas master não são muito manuseadas e nem monitoradas pelos administradores. Já as máquinas agentes têm acesso a Internet por *links* rápidos.
4. Com o *daemon* instalado nos agentes, estes ficam prontos para receber os comandos dos masters. A máquina master registra uma lista de IP das máquinas agentes ativas. Com essa conexão entre masters e agentes já se pode organizar os ataques;

As fases 1 e 2 são realizadas imediatamente uma após a outra e de forma automatizada. Por isso as informações de vulnerabilidade são de suma importância para a instalação rápida das ferramentas de DDoS.

Fase 3: Realizando o ataque:

O atacante pode controlar uma ou mais máquinas master e estas por sua vez podem controlar muitas máquinas agentes. A partir daí pode-se disparar os pacotes consolidando o ataque. Os agentes ficam aguardando as instruções do master para atacar um ou mais de um endereço IP em determinado espaço de tempo.

O atacante ordenando o ataque, as vítimas terão suas máquinas congestionadas por um grande número de pacotes, interrompendo o *link* de rede e assim paralisando os serviços.

4.4.2 Classificação dos ataques DDoS

Sabemos que o objetivo de um ataque de DDoS é consumir os recursos do alvo enviando um grande número de pacotes com a finalidade de que este não possa fornecer seus serviços. Assim podemos classificar um ataque DDoS em termos de recurso consumido. Este é dividido em recursos internos do *host* ou capacidade de transmissão de dados.

Um exemplo específico de ataque de recursos interno seria o ataque por inundação de SYN.A. Segue-se no ataque os seguintes passos:

1. O atacante por meio de seus escravos ou agentes os instrui a entrar em contato com o servidor Web do alvo;
2. Os agentes enviam por comando do atacante, pacotes SYN (sincronismo/inicialização), com endereçamento IP errado de retorno, para o alvo;
3. Para cada pacote SYN o alvo retorna um pacote SYN/ACK (sincronizar/confirmar) tentando formar uma conexão TCP, porém este endereço de origem não existe, então o alvo fica esperando finalizar falsas conexões.

Um exemplo para o ataque que limita os recursos para a transmissão de dados é o ataque distribuído utilizando o protocolo *Internet control Message Protocol* (ICMP). Seguem-se as seguintes etapas para o ataque:

1. O atacante por meio das máquinas mestres instrui as máquinas escravas ou agentes a enviar pacotes ICMP ECHO (pacote que solicita aos destinatários um resposta) com um endereço de IP falsificado do alvo para várias máquinas que atuam como refletoras;

2. Essas máquinas refletoras de posse do endereço IP do alvo respondem com um pacote chamado ECHO REPLY tentando manter uma conexão;
3. O alvo que pode ser um roteador fica congestionado com os pacotes ECHO REPLY das máquinas refletoras.

Outra maneira de classificar um ataque DDoS é dividi-lo em diretos ou refletores. No ataque direto o atacante instala softwares zumbis em sites da Internet. Instala softwares zumbis em máquinas mestres que por sua vez instalam softwares zumbis em máquinas escravas ou agentes, que vão disparar seu ataque contra o alvo ou vítima. Já no ataque refletor é acrescentado uma nova camada de máquinas em relação ao ataque direto. Essa camada é chamada de refletora. Os zumbis escravos enviam pacotes para as máquinas refletoras cujas respostas serão enviadas para a máquina alvo. Esse tipo de ataque é muito mais prejudicial do que um ataque direto, pois envolve um número muito maior de máquinas.

4.4.3 Vulnerabilidades

Um atacante precisa conhecer as vulnerabilidades das máquinas nas quais deseja invadir. Esse processo é conhecido com varredura, mas para isso torna-se necessário ter uma estratégia que permita o conhecimento destas vulnerabilidades. Abaixo segue algumas estratégias de varredura:

- Aleatória: Máquinas comprometidas sondam endereços IP utilizando endereços de origem diferentes;
- Lista de acerto: Uma lista de máquinas com grande possibilidade de vulnerabilidade são analisadas. Este processo é lento. Porém, feita a análise, e assim que as máquinas começarem a ser infectadas, estas também passam a auxiliar na análise de vulnerabilidades e a infectar outras máquinas, o que torna a varredura muito mais eficiente;
- Topológica: A partir de uma máquina infectada tira-se informação desta para se analisar outras máquinas;
- Subrede local: Logo que uma máquina que está atrás de um *firewall* é invadida, esta procurará alvos na rede local. Para isto ela utilizará os endereços de sub-rede que anteriormente estavam protegidos pelo *firewall*.

4.4.4 Ferramentas de ataque DDoS

Os ataques DDoS não são nenhuma novidade. Desde seu surgimento em 1998 as ferramentas de DDoS vêm se desenvolvendo constantemente, ficando mais sofisticadas e com melhores interfaces. Veja tabela 1:

1ª – Fapi
2ª – Blitznet
3ª – Trin00
4ª – TFN
5ª – Stacheldraht
6ª – Shaft
7ª – TFN2K
8ª – Trank
9ª – Trin00 win version

Tabela 1: Ordem de surgimento das ferramentas de DDoS.

Dentre estas aqui mostradas as principais ferramentas de DDoS são: Trin00, TFN, Stacheldraht e TFN2K.

4.4.4.1 Trin00

Ferramenta distribuída que lança ataques coordenados de DDoS do tipo UDP *flood*. Geralmente uma rede Trin00 é composta por uma proporção muito maior de agentes do que de masters. Algumas características são:

- O controle remoto do master pelo atacante é feito por conexão TCP;
- A comunicação master - agente é feita via pacotes UDP ou TCP;
- A comunicação agente - master é feita por pacotes UDP.
- Ao ser inicializado um *daemon*, este anuncia para o master sua disponibilidade enviando uma mensagem, ao master. Este por sua vez tem uma lista de IP das máquinas agentes que estão ativas e que estão sendo controladas pelo master. Um nome comum encontrado no master como cliente é o “master.c” e alguns dos nomes de *daemons* que têm sido vistos nos agentes são: “ns”, “http”, “trinix” e etc. Essas aplicações não necessitam de privilégios de *root*.

4.4.4.2 TFN – Tribe Flood Network

Ferramenta distribuída que lança ataques DoS a uma ou mais de uma máquina vítima, por meio de outras máquinas que já estão comprometidas. Algumas características são:

- Pode gerar ataques UDP flood, SYN flood, ICMP flood e Smurf/Fraggle (mecanismo de envio de pacotes para endereços de *broadcasting*);
- Permite esconder o endereço origem dos pacotes o que dificulta a identificação do atacante;
- O controle remoto do master feito pelo atacante é realizado por comandos executados por meio do programa cliente e pode ser utilizado qualquer meio de conexão, tais como rsh, telnet e etc;
- A comunicação cliente – *daemons* ou master – agente é feita por meio de pacotes ICMP_ECHOREPLY;
- O TFN assim como o Trin00 trabalha com uma lista do IP das máquinas com os daemons instalados;
- O nome “tribe” é utilizado para a aplicação cliente e o nome “td” é usado para identificar os *daemons* instaladas nos agentes. Estas aplicações devem ser executadas com privilégio de *root*.

4.4.4.3 STACHELDRAHT

Ferramenta distribuída que lança ataques DoS a uma ou mais de uma máquina vítima, por meio de outras máquinas que já estão comprometidas. Pode-se dizer que esta ferramenta é uma junção da Trin00 e da TFN adicionada de mais alguns itens como criptografia na comunicação entre atacante e master (telnet criptografado) e constante atualização dos agentes. Assim como a Trin00, a STACHELDRAHT é composta por uma proporção maior de agentes do que de masters. Algumas características são:

- Gerar ataques UDP *flood*, SYN *flood*, ICMP *flood* e Smurf/Fraggle;
- O controle remoto do master pelo atacante é feito por conexão TCP;
- A comunicação entre masters e agentes e vice-versa é feita por meio de pacotes TCP e ICMP;

- Programas clientes costumam vir com o nome de “mserv” e *daemons* com nomes de “leaf” ou “td”. Estas aplicações devem ser executadas com privilégio de root.

4.4.4.4 TFN2K - TRIBLE FLOOD NETWORK 2000

Ferramenta distribuída de DoS que foi escrita pelo mesmo autor da TFN, Mixter. Algumas características são:

- Pode gerar ataques UDP *flood*, SYN *flood*, ICMP *flood* e Smurf/Fraggle ou ainda a daemon pode ser instruída para alternar entre estes ataques;
- O controle remoto do master pelo atacante é feito via pacotes TCP, UDP, ICMP ou os três aleatoriamente;
- Não há confirmação (ACK) de recepção de comandos como no TFN, ao invés disso são enviados vários comandos iguais para que pelo menos um chegue ao objetivo.

Abaixo segue a tabela 2 que é um comparativo de meios de comunicação entre os personagens de um ataque DDoS para cada ferramenta de DDoS:

Comunicação	**Trin00**	**TFN**	**Stacheldraht**	**TFN2K**
Atacante-Master	27665/tcp	icmp_echoreply	16660/tcp	icmp/udp/tcp
Master-Agente	27444/udp ou 1524/tcp	icmp_echoreply	65000/tcp, icmp_echoreply	icmp/udp/tcp
Agente-Master	31335/udp	icmp_echoreply	65000/tcp, icmp_echoreply	icmp/udp/tcp

Tabela 2: Comparativo de meios de comunicação entre os personagens de um ataque DDoS para cada ferramenta de DDoS.

4.4.5 Linhas de defesa

Basicamente pode-se dizer que existem três linhas de defesa contra ataque DDoS:

- Prevenção do ataque: O objetivo é não negar o serviço para usuários legítimos, mas permitir a resistência do alvo. A técnica é determinar o consumo de recursos e o fornecimento recursos reservas;

- Detectar o ataque: O objetivo é fazer a detecção o mais cedo possível. A técnica é fazer uma busca por comportamentos suspeitos;
- Rastear e identificar a origem do ataque: O objetivo é prevenir ataques futuros a partir da origem do ataque. A técnica de análise da origem não é tão viável quanto a ataques em andamento, pois não traz resultados imediatos.

4.4.6 Como se defender do DDoS

Não existe uma maneira totalmente eficaz para se defender de um ataque de DDoS devido a força e arquitetura do mesmo. A solução perfeita seria configurar os computadores para que estes não permitissem a invasão com o objetivo de formação de uma rede de DDoS.

Apesar das dificuldades de se encontrar uma forma ideal de se proteger, existem métodos de defesa como, por exemplo, um plano de contingência que é uma forma de defesa contra ataques de força-bruta que consome recursos da rede devido a sua origem ser de redes numerosas e com muitos recursos.

Uma política de segurança pode proteger contra ataques de DDoS, porém, ela deve garantir que:

- Usuários legítimos não venham a colaborar para possíveis ataques;
- As senhas escolhidas devem ter um tamanho adequado e devem ser trocadas periodicamente;
- O acesso à parte físico deve ser feito apenas por administradores;
- Deve-se fazer constante atualização do sistema;
- Os programas antivírus devem estar atualizados;
- Portas abertas deverão somente ser as que estão sendo utilizadas;
- A largura de banda deve ser estipulada por servidor;
- Deve-se criar diretrizes para recebimentos de pacotes para endereço de broadcasting;
- Pode ser feito o bloqueio de endereços de internet na possibilidade de um ataque;
- Um plano de reação é ideal para garantir uma boa resposta no momento crucial;

- A implantação de um IDS, tomando o cuidado de se fazer uma verificação da rede para ver se a mesma não está comprometida.

4.4.6.1 Detecção

No que se diz respeito à detecção de ataques DDoS temos como principal dificuldade a criptografia. Por outro lado, há possibilidade da modificação do código fonte de forma que senhas e portas sejam alteradas quando se trata de prevenção.

No entanto, há possibilidade de detecção e existem várias maneiras de se fazer isso, desde métodos convencionais como Auditoria e Ferramentas de Detecção Específicas até métodos mais modernos como a instalação de IDS.

4.4.6.2 Detecção de anomalias de tráfego (DDoS) usando Entropia Não-Extensiva

Podemos fazer uma analogia quando falamos de anomalias de tráfego e DDoS e chegamos a conclusão que se trata da mesma coisa. A principal característica das anomalias de tráfego são as alterações danosas que estas podem ocasionar à rede. A Entropia Não-Extensiva analisa esse tipo de problema e existem formas de Entropia (que é a avaliação do padrão de comportamento do tráfego) como a de Shannon e a de Tsallis. Vários estudos foram feitos e concluiu-se que este tipo de detecção é flexível e permite um bom desempenho. Mais adiante falaremos com detalhes desta utilização de Entropia na detecção de anomalias.

4.4.6.3 Algumas Ferramentas de Detecção de Negação de Serviço

O Zombie Zapper bloqueia um ataque em andamento. Se um IDS detecta que a rede está sendo usada como base de ataque, o Zombie Zaper por meio de comandos enviados ao Agente interrompe o ataque.

Find DDoS é uma ferramenta desenvolvida pelo FBI justamente pelo grande número de ataques de DDoS. O Find DDoS faz a localização do Master e do Agente das ferramentas de ataque DDoS, tais como Trin00, TFN2K, Tribe Flood Network e Stacheldraht.

DDoS Ping que é um programa desenvolvido pelo Robin Keir, torna mais fácil e acessível sua utilização por possuir boa interface gráfica. Detecta Agente com as seguintes ferramentas Trin00, Stacheldraht e Tribe Food Network. A busca é feita por meio de mensagens ICMP e UDP enviadas para endereços IP que foram definidos pelo usuário.

4.4.7 Conclusões relativas a ataques DDoS

O DDoS passaram a preocupar quando suas ferramentas de ataque se popularizaram na Internet. Estas ferramentas facilitam tanto a execução de um ataque DDoS que mesmo pessoas pouco experientes conseguem lançar um ataque desse tipo.

Ataques a protocolos podem ser evitados após a descoberta de suas vulnerabilidades. Já os ataques de força-bruta, ou seja, de DDoS são um pouco mais complicados, pois devem ser detectados e combatidos em sua origem. Ameaças DDoS sempre existirão quando uma máquina está conectada porque sempre há possibilidade de se receber um grande número de dados.

Com a Internet nada é mais cem por cento seguro, quando se está conectado. Existem maneiras de se diminuir um ataque DDoS, mas nada infalível. Tudo vai depender da disponibilidade de tempo e experiência do atacante.

Uma solução que está sendo estudada é a implementação de vários sistemas de segurança que serão colocados em lugares estratégicos na Internet cuja finalidade seria interromper esse tipo de ataque em sua origem. A partir do momento que um ataque DDoS fosse detectado então roteadores reduziriam ou bloqueariam o tráfego. Posteriormente este poderia liberar o tráfego de forma que não inundasse os destinatários.

Ataques recentes mostraram que tanto as redes locais quanto a Internet estão vulneráveis a ataques DDoS. Assim é de grande importância, principalmente para os administradores de rede, uma maior atenção no desenvolvimento dos métodos de ataque de prevenção de DDoS, ou seja, novidades em IDSs.

4.5 IDS x *Firewalls*

4.5.1 Definindo *Firewall*

Um *firewall* é um dispositivo de rede de computadores que analisa o tráfego entre redes. Ele pode impedir que dados não autorizados ou que possam vir a prejudicar a rede entrem na mesma.

Costuma-se relacionar o conceito de *firewall* com proteção completa de uma rede de computadores, porém um *firewall* se restringe até o nível quatro do modelo OSI.

A idéia do Termo “parede de fogo” tem sua origem justamente pela função que desempenha esse equipamento. O que ele faz é o mesmo que uma parede, pois não deixaria passar o fogo. No entanto, neste caso o equipamento não deixa passar os dados nocivos que prejudicariam a rede.

Os *firewalls* são encontrados tanto em hardware quanto em software, ou então na junção dos dois. Sua instalação é muito relativa, pois depende do tamanho, da complexidade de regras de entrada e saída e da segurança desejada na rede.

Os *firewalls* se classificam da seguinte forma:

- Filtro de pacotes;
- *Proxy firewall*;
- *Stateful firewall*;
- *Firewall* de aplicação;
- Comandos e opções de *firewall*.

4.5.1.1 Filtro de pacotes

Este tipo de sistema analisa os pacotes que passam da camada 2 de enlace para a camada 3 de rede do modelo OSI.

As regras podem envolver endereço de origem e destino e as portas TCP/IP. Uma das desvantagens seria a falta de controle dos tipos de pacotes o que permitiria a injeção de pacotes simulados na sessão.

4.5.1.2 Proxy firewall

Iniciado em 1995 por Marcus Ranum o *proxy* recebe os dados de uma conexão e antes de enviar para o solicitante faz uma análise destes dados. Para que seja enviado para o solicitante é feito um novo pedido o qual quem controla é o *firewall*.

Algumas desvantagens são:

1. A cada nova solução na internet é necessário um modelo compatível de *proxy*;
2. Os *proxies* trazem perda de desempenho na rede, pois teria que ter o processamento tanto do *gateway* quanto do *proxy*;

4.5.1.3 Stateful firewall

Trata-se de um tipo de *firewall* que inspeciona todos os pacotes em todas as camadas do padrão OSI. Possui a segurança de um *gateway* e a velocidade de um filtro de pacotes, é transparente aos usuários e permite ao administrador a adição de novos serviços de Internet com muita facilidade, somente por definição de novas Tuplas. Há facilidade na manutenção e instalação e é de baixo custo.

4.5.1.4 Firewall de Aplicação

Analisa o protocolo da aplicação e define decisões dentro de suas particularidades. Este tipo de *firewall* exige um conhecimento muito grande dos protocolos de internet e no processamento dos ataques focados na camada de aplicação. Exige ainda uma constante evolução na tecnologia das necessidades computacionais de um *firewall* de aplicação. Estas necessidades advêm dos algoritmos e das regras de detecção e também da velocidade do trânsito de pacotes pela rede. Há ainda grande necessidade de altos recursos computacionais para o processamento da criptografia e descriptografia dos pacotes que passam por ele.

Para mais conhecimento a respeito do *firewall* de aplicação consulte a ApRisco (Associação Profissional de Risco).

4.5.1.5 Comandos e Opções de Firewall

Masquerade: opção que associada ao comando Iptables traduz endereços de rede dos pacotes que passam pelo servidor *firewall*.

Redirect: opção que associada ao comando Iptables ou Ipchains configura um sistema *transparent proxing.*

4.5.2 Firewall ou IDS

Costumamos nos perguntar o porquê de usarmos um IDS se já temos um *firewall*. Sabendo que o *firewall* permite conexões com o servidor de FTP e se alguém tenta baixar o passwd do servidor de ftp o *firewall* poderá até reconhecer o tráfego, mas não fará o bloqueio. Já o IDS detectará essa movimentação e gerará um alerta. Se estivermos falando de IPS este poderá bloquear o tráfego.

Em se tratando de modelo OSI os filtros de pacotes dos *firewalls* geralmente trabalham nas camadas de rede e de transporte. Após as aplicações das regras, que são checagem de endereços IP, protocolos e número de porta, os pacotes são filtrados com base nessas checagens.

Os *firewalls* simplesmente fazem essa checagem e filtram os pacotes que ele achar fora da normalidade com base em regras pré-estabelecidas, assim, o *firewall* não faz uma análise do que está sendo feito pelo usuário. Já o IDS trabalha tanto nas camadas três e quatro do modelo OSI como também na camada sete, ou seja, de aplicação. Ele busca por Trojans, ataques de negação de serviço etc. Como visto anteriormente existem dois tipos de *firewalls* que atuam também na camada de aplicação: é o stateful *firewall* que é de baixo custo e o *firewall* de aplicação que exige um grande recurso computacional.

Podemos então fazer a seguinte conclusão:

Firewall = Trabalha de forma estática (na maioria dos casos);

IDS/IPS = Trabalha de forma dinâmica.

5 Segunda Lei da Termodinâmica e Entropia

O leitor pode estar se perguntando o porquê de um capítulo falando de Termodinâmica e Entropia, o fato é que este conceito fará com que possamos ver como a Entropia pode ser usada em aplicações de detecção de intrusos.

Um grande nome da astrofísica, o britânico Arthur Eddington fez a seguinte conclusão: “Se a sua teoria contrariar alguma lei da física tudo bem, é possível que a lei deva ser modificada. Mas se essa lei for a Segunda Lei da Termodinâmica, pode jogar sua teoria no lixo”.

5.1 Relação entre a Segunda Lei da Termodinâmica e a Entropia

A segunda lei da termodinâmica é uma das leis naturais mais importantes que existe. Na sua forma simplória, que teve origem no século XIX proposta por Rudolf Clausius, médico alemão, e Lord Kelvin, físico inglês, ela fala que o calor se transfere de um corpo mais quente para um mais frio.

Por mais simples que possa parecer essa lei, ela nos trás uma informação de grande valia, pois, nos mostra a causa de a desordem sempre tender a crescer e a ordem tender a decrescer. Também nos dá uma idéia da passagem do tempo e do porquê do nosso envelhecimento e outras questões também importantes sobre o mundo e a vida. Porém, o que nos interessa mesmo é a idéia de ordem e desordem conhecida como entropia.

Vamos por partes, começando com fatos que são familiares para todo mundo. Quando você põe um cubo de açúcar no café, o cubo dissolve. Uma vez dissolvido você não verá os grãos de açúcar voltarem a formar o cubo [9].

Se você abrir uma garrafa de perfume em um quarto fechado, você sentirá o cheiro agradável se espalhando pelo quarto. Isso ocorre por que as moléculas de perfume chocam-se entre si, escapando da garrafa, e, aos poucos, vão se chocando também com as moléculas de ar no quarto, e o perfume vai se difundindo. Você não verá o aroma agradável desaparecer devido ao fato de todas as moléculas espontaneamente não terem resolvido voltar para a garrafa [9].

Mais um exemplo: você quebra um ovo e prepara uma omelete. Jamais você verá a omelete se transformar de volta em um ovo. Todos esses processos mostram que existe uma

direção preferencial para a passagem do tempo. Se você visse uma omelete se transformando em um ovo, você imediatamente concluiria, por mais estranho que fosse que o tempo estivesse voltando. [9]

Estes exemplos (o ovo, o perfume e o cubo de açúcar) têm uma característica em comum, pois todos eles passam por um processo e terminam desorganizados (a omelete, o perfume espalhado e o cubo de açúcar dissolvido). Esse processo não ocorre especificamente com esses exemplos, pois isso ocorre com todo sistema que não troca energia com o exterior. [9]

Sabe-se que a entropia é a quantidade de desordem de um sistema, assim quanto mais ordem, menor é a entropia. No exemplo dado, o cubo de açúcar e a xícara de café possuem uma entropia menor do que a dos grãos de açúcar dissolvidos no café. Essa comparação, ou seja, crescimento na entropia define a segunda lei da termodinâmica: dado um sistema isolado a entropia nunca tende a diminuir, porém, pode crescer ou ainda se manter. A segunda lei também tem relação com a passagem de tempo, pois é de costume definir a passagem de tempo com o crescimento da entropia.

Podemos nos perguntar se a segunda lei não está em contradição com a teoria da evolução, pois, nós viemos de seres unicelulares totalmente simples e hoje somos seres com formação biológica muito organizada. A resposta já foi dada acima quando se fala na segunda lei onde somente sistemas isolados que não trocam informação e energia com o exterior e conseqüentemente tendem a não se organizarem. E sabemos que este não é o caso dos seres vivos.

Todos os animais precisam de alimentação para produção de energia. Para se ter vida, precisa-se de harmonia com outros seres, assim, não se pode viver isoladamente.

5.2 A Segunda Lei da Termodinâmica e a Entropia – Conceitos

5.2.1 Entropia

Entropia é definida como uma grandeza termodinâmica que se associa ao grau de desordem. É uma medida de parte da energia que não se transforma em trabalho. É uma função de estado que aumenta seu valor durante um processo natural em um sistema capaz de não fazer trocas de energia.

5.2.2 Segunda Lei da Termodinâmica

De acordo com o Princípio da Conservação da Energia em qualquer transformação natural, a energia total ou final é sempre constante. A primeira Lei da Termodinâmica reafirma essa idéia, porém não prevê a possibilidade da realização dessa transformação. Existem muitos eventos que satisfazem essa Lei, mas que são praticamente impossíveis de acontecer.

A Segunda Lei da Termodinâmica define que um corpo de maior temperatura passa seu calor para um de menor temperatura, porém a possibilidade de o inverso acontecer segundo a Primeira Lei, existe, mas é praticamente impossível, devido aos sistemas tenderem ao equilíbrio.

Em resumo a Segunda Lei da Termodinâmica define que em uma transformação natural, a energia vai de uma forma organizada para uma forma mais desorganizada, conceito anteriormente visto como Entropia e que tem íntima ligação com essa Lei.

Essa Lei foi definida por Clausius da seguinte forma:

“O calor não passa espontaneamente de um corpo para outro de temperatura mais alta”.

Visto que o calor é uma forma de energia que sofreu certa degradação, a sua conversão em alguma outra forma de energia não é tão simples, embora a Primeira Lei defenda essa possibilidade. Assim Kelvin e Planck definiram a Segunda Lei da Termodinâmica da seguinte forma:

“É impossível construir uma máquina, operando em ciclos, cujo único efeito seja retirar calor de uma fonte e converte-lo integralmente em trabalho”.

Imaginemos um recipiente com cem bolinhas vermelhas num recipiente fechado e cem bolinhas azuis acima destas. Em seguida pegamos o recipiente e o agitamos. Existe a possibilidade dessas bolinhas retornarem a posição inicial, porém essa possibilidade é muito pequena. Outro exemplo seria um baralho ordenado por naipes e valores que após ser embaralhado, para retornar a posição inicial seria praticamente impossível sua ordenação depois de uma nova tentativa de ordenação por meio de embaralhamento.

Os fenômenos naturais tendem a irem sempre para os estados mais prováveis. Por isso a idéia de tudo sempre passar de um sistema ordenado para um desordenado. Retira-se então a seguinte conclusão:

“À medida que o Universo evolui, a desordem sempre aumenta”.

“Em todos os fenômenos naturais, a tendência é uma evolução para um estado de maior desordem”.

Clausius inseriu o conceito matemático de Entropia no conceito estatístico de desordem. Com essa relação, a Entropia aumenta quando aumenta a desordem nos processos naturais. Assim:

“As transformações naturais sempre levam a um aumento na Entropia do Universo”.

Uma variação de entropia entende-se com sendo a ineficácia da energia do sistema em uma evolução natural. Assim, um sistema sempre tende a diminuir a possibilidade de se conseguir uma energia aproveitável.

Falando em transformação natural, quando esta ocorre, uma forma de energia se converteu em calor, diminuindo à energia total do sistema e aumentando a Entropia do mesmo. Em resumo podemos dizer que essa quantidade de calor é uma medida parcial do aumento de Entropia.

6 Entropia Não-Extensiva de Tsallis e sua utilização na Detecção de Anomalias de Tráfego

Anomalias são alterações nos enlaces de rede não muito comuns e que merecem um pouco de atenção, pois, podem trazer complicações posteriores. Para se fazer detecção desse tipo de anomalias falaremos neste capítulo da Entropia Não-Extensiva de Tsallis que é uma variação da Entropia de Shannon. Esse tipo de Entropia é uma proposta bem atraente devido a sua flexibilidade no nível de detecção, pois podemos ajustar o nível de acordo com a necessidade, e também pelo seu desempenho se comparado com outras abordagens.

A partir da metrologia de redes podemos obter várias conclusões no que diz respeito à Internet e a redes em geral de pequeno porte. Alterações significativas e pouco comuns nos enlaces de rede, com ou sem intenção. Essas alterações são provenientes de DDoS e problemas com encaminhamentos de endereços IP, que inclui falha em equipamentos e má configuração de roteadores. Para se obter um diagnóstico de anomalias, esbarramos em algumas dificuldades tais como a variedade das anomalias e o volume de dados, pois dificulta a análise.

Este diagnóstico é a detecção, identificação e quantificação das anomalias. Detectar seria analisar a rede e verificar alguma anomalia em determinado período. Identificar é o mesmo que, a partir de uma base de dados fazermos a classificação da anomalia. Por fim, a quantificação é a contagem do volume de situações anômalas. A detecção é de grande valia porque permite tomarmos uma linha de ação rapidamente após a identificação.

Sabe-se que a probabilidade do nível de tráfego nos nós de entrada e saída de uma rede pode ser utilizada para quantificação das anomalias através de Entropia. A proposta, contudo não é trabalhar com a Entropia de Shannon, idéia muito utilizada para detecção de anomalias, mas sim com uma generalização da mesma que é Entropia Não-Extensiva de Tsallis.

A Entropia Não-Extensiva de Tsallisé bastante flexível devido à possibilidade da variação do nível de sensibilidade das detecções. Ela melhora o desempenho quanto à detecção e diminui os falsos negativos.

6.1 Cálculo de Entropia

O cálculo de Entropia é feito para quatro categorias:

1- Entropia de portas de origem;
2- Entropia de portas de destino;
3- Entropia de endereços de origem;
4- Entropia de endereços de destino;

A classificação da anomalia é obtida de um cruzamento entre os quatro valores de Entropia. As características de uma anomalia são obtidas por meio do nível de concentração e dispersão das categorias mencionadas anteriormente. A Entropia não-extensiva de Tsallis apesar de ser utilizada hoje em dia em diversas situações, com esta abordagem a que estamos nos referindo que é a detecção de anomalias, não é ainda difundida e nem abordada. Posteriormente falaremos do cálculo prático, ou matemático de Entropia.

6.2 Entropia de Shannon e Entropia Não-Extensiva de Tsallis

Neste tópico mencionaremos o conceito de Teoria da Informação para entendimento da Entropia de Shannon e sua utilização na detecção de anomalias. Em seguida faremos a abordagem da Entropia Não-Extensiva de Tsallis para obtenção de uma visão amplificada e de melhor desempenho para análise de anomalias.

6.2.1 Teoria da Informação

Teoria da informação é um ramo da teoria da probabilidade e da matemática estatística que lida com sistemas de comunicação, transmissão de dados, criptografia, codificação, teoria do ruído, correção de erros, compressão de dados e etc. Ela não deve ser confundida com tecnologia da informação e biblioteconomia [10].

Claude E. Shannon conhecido como o pai da teoria da informação foi o primeiro a tratar da comunicação como um problema matemático embasado na estatística onde determina o nível de eficiência de um canal de comunicação através das ocorrências dos bits. Esta teoria tem relação com a perda de informações quando há compressão de dados e também quando se transmite um sinal em um canal com problemas de ruído. Através dessa teoria Shannon definiu a medida de Entropia.

Esta entropia que foi definida por Shannon tem ligação direta com a entropia definida pelos físicos. Há uma relação entre a entropia definida na termodinâmica e na teoria da informação. Esta teoria de Shannon mede a incerteza em um espaço desordenado.

6.2.2 Entropia de Shannon

Pode-se dizer que é uma medida ligada à quantidade de informações e de incerteza em um dado sistema com base na probabilidade de um determinado fenômeno acontecer [11]. A Entropia é utilizada para determinar através de um volume de dados o comportamento destes, onde esse volume pode ser de fluxos IP ou então quantidade de bytes. Através desse volume de dados determinamos se o fluxo de dados está concentrado, onde grande parte dos dados está indo para um único ponto de rede, ou disperso, quando o tráfego está distribuído. Através desse tráfego podemos calcular a probabilidade do fluxo de dados nos nós e determinar o nível de entropia em cada ponto de rede.
Entropia de Shannon:

$$Hs = -\sum_{i-1}^{n} P_i \log_2 P_i$$

Onde: n é número de eventos;

P_i é a distribuição da probabilidade;

Hs varia entre 0 $\log_2 n$ que determina o grau de caoticidade da distribuição de probabilidade Pi e pode ser usada para determinar a capacidade do canal necessária para transmitir a informação;

Observações:

Hs = 0, concentração máxima (todo o tráfego para um único ponto);

Hs = $Hs^{\max}$, disperção total (tráfego distribuído uniformemente com probabilidade de 1/n).

Podemos concluir que quanto maior a Entropia, mais disperso está o tráfego e melhor funciona o sistema. Quanto maior a proximidade da probabilidade de um evento ocorrer em determinado ponto do sistema, mais disperso este será, e assim vice-versa. Fica então a fórmula definida da seguinte maneira:

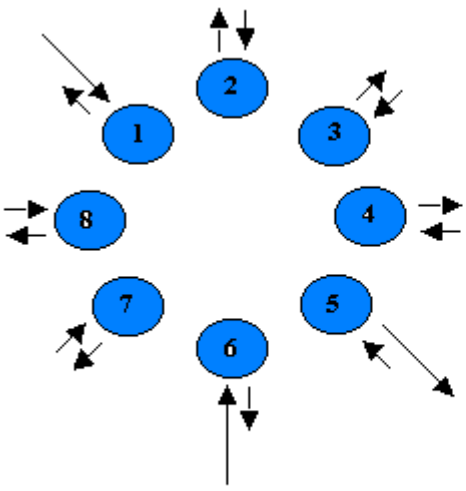
$$Hs^{\max} = -\sum_{i=1}^{n}\left(\frac{1}{n}\log_2\frac{1}{n}\right) = \log_2 n$$

6.2.3 Entropia e sua utilização na detecção de anomalias

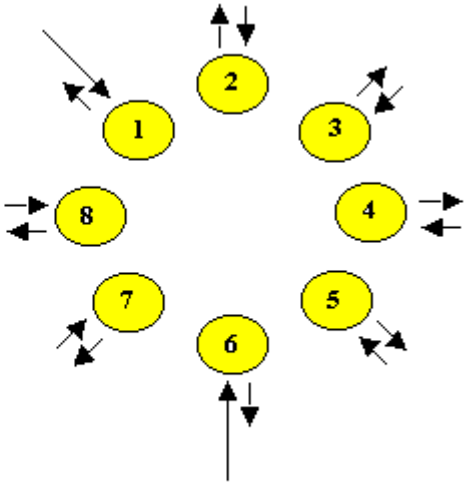
Através do fluxo de dados podemos agrupar o fluxo de entrada e de saída por cada ponto de domínio. Assim, podemos calcular as probabilidades de cada ponto de entrada chamado origem e cada ponto de saída chamada destino. A partir desse conceito podemos definir quatro categorias importantes quanto à detecção de anomalias:

1. Origem concentrada e destino concentrado (CC);
2. Origem concentrada e destino disperso (CD);
3. Origem dispersa e destino concentrado (DC);
4. Origem dispersa e destino disperso (DD).

Analisando a ilustração abaixo veremos a indicação de tráfego de entrada e saída nos pontos de presença: Na figura 15, as setas maiores indicam um grande volume de tráfego, enquanto que as setas menores indicam menor tráfego nos demais pontos. Se todas as setas estiverem com o mesmo tamanho então isso indicará que o padrão de tráfego está uniforme. Veja figura 15:



1- Concentração - Concentração (CC)



2- Concentração - Dispersão (CD)

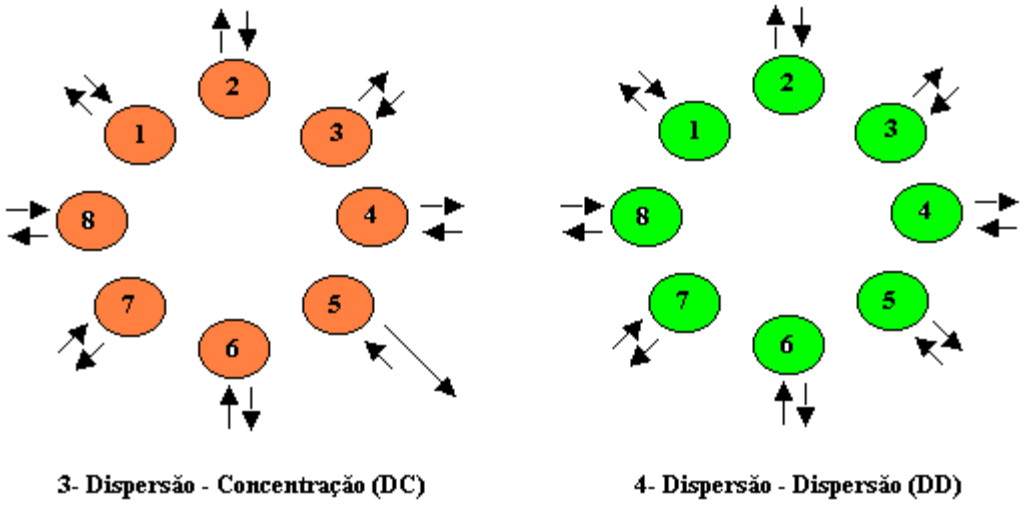


Figura 15: Demonstrativo de concentração – dispersão de tráfego num domínio IP.

Análise:

1º caso: O tráfego entra predominantemente pelos pontos 1 e 6 e sai com maior concentração pelo ponto 5, caracterizando um padrão definido como Concentrado-Concentrado (CC);

2º caso: O tráfego entra predominantemente pelos pontos 1 e 6 e sai após o roteamento distribuído uniformemente por todos os pontos de presença restantes caracterizando o padrão Concentração-Dispersão (CD);

3º caso: O tráfego entra uniformemente distribuído por todos os pontos e sai de forma concentrada pelo ponto 5 caracterizando o padrão Dispersão-Concentração (DC);

4º caso: O tráfego entra e sai uniformemente distribuído por todos os pontos, caracterizando o padrão de tráfego Dispersão-Dispersão (DD).

Na figura 16 mostraremos como um padrão de tráfego pode determinar algumas situações em que há ocorrência de anomalias.

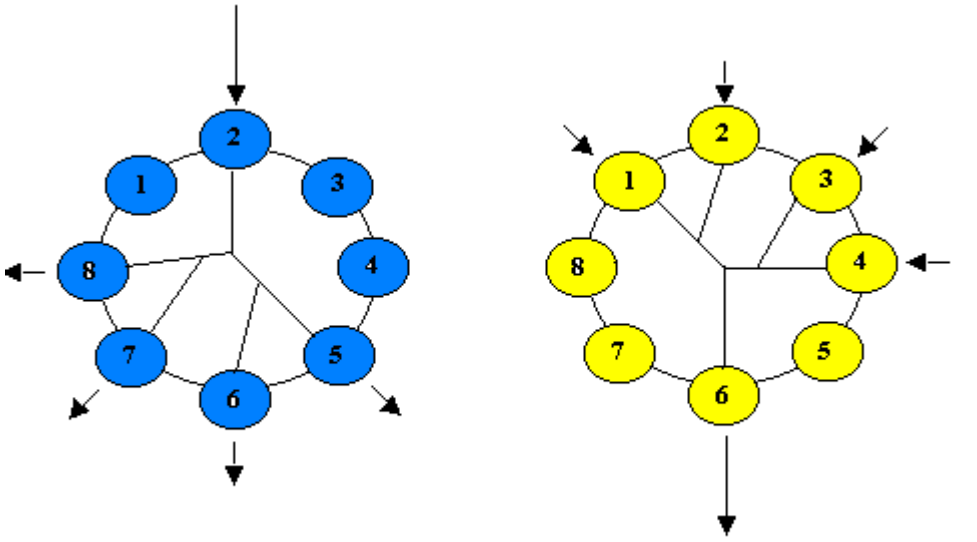


Figura 16: Como se caracteriza uma anomalia através dos padrões de tráfego.

Análise:

1º caso: Transmissão Multicast onde o ponto de presença 2 recebe todo o tráfego e após sua replicação no domínio é retransmitida pelos ponto 5, 6, 7 e 8 definindo um tráfego CD (Concentração - Dispersão);

2º caso: Trata-se de um Ataque Distribuído de Negação de Serviço (Distributed Denial of Service - DDoS) e neste caso é feito por intermédio dos pontos 1, 2, 3 e 4 onde estes recebem os fluxo com os endereços destinados a um único ponto, neste caso o ponto 6, assim sendo define-se um padrão de tráfego DC (Dispersão – Concentração).

6.2.4 Entropia Não Extensiva de Tsallis – Cálculo e Utilização na detecção de anomalias

6.2.4.1 Cálculo da Entropia Não Extensiva de Tsallis

A Entropia Não Extensiva de Tsallis é uma generalização da Entropia de Shannon.

$$Hq = \frac{1 - \sum_{i=1}^{n} Pi^{q}}{q - 1}$$

Onde: *n* é o total de elementos;

P_i é a probabilidade de um evento ocorrer;

q define o grau de extensividade do sistema, ou seja, a sensibilidade do sistema quanto à detecção de anomalias.

logo: $0 <= P_i <= 1$ e $\sum_i P_i = 1$;

O cálculo de Entropia de Tsallis tem uma certa equivalência com a de Shannon com isso podemos comprovar que se trará de uma generalização.

O resultado do valor de Entropia varia entre 0 que significa concentração máxima e $H_q^{\max}$ que significa dispersão total onde:

$$H_q^{\max} = \frac{1 - n^{1-q}}{q - 1}$$

A diferença dessa fórmula de Tsallis para a de Shannon é a introdução do parâmetro "q" cujo termo vai determinar o grau de extensividade do sistema. Também cabe mencionar que a variação da probabilidade dos eventos no cálculo de Shannon não influencia tanto no resultado final da Entropia. Já no cálculo de Tsallis define-se o valor de "q" onde:

$q > 1$: eventos com maiores probabilidades acarretam maiores alterações no valor da Entropia do que eventos de menores probabilidades;

$q < 1$: eventos com menores probabilidades acarretam maiores alterações no valor da Entropia do que eventos de maiores probabilidades;

6.2.4.2 Uso da Entropia Não Extensiva de Tsallis na detecção de anomalias

A idéia da aplicação da Entropia Não extensiva de Tsallis se dá de forma prática pela utilização direta do parâmetro "q", existente na fórmula, que define a sensibilidade do sistema quanto à detecção de anomalias. Pode-se definir o quanto de detecção se quer no nosso sistema. Portanto, como estamos sempre em busca do melhor controle no sistema de detecção, buscaremos também sempre um "q" ótimo que definirá o melhor controle de sensibilidade do sistema no problema em questão, que no nosso caso é a detecção de intrusos.

7 Equipamentos e Programas

Vários equipamentos e programas de IDS e IPS estão disponíveis atualmente. Como IDS destacamos o Snort, e citamos o Ossec HIDS e como IPS citamos os Appliances e o HLBR. Cabe salientar que o objetivo é dar maior destaque para o Snort como IDS.

7.1 Snort

O Snort é um sistema de detecção de intruso baseado em rede, onde é aplicado em segmentos de rede para detecção de intrusos. É uma ferramenta muito utilizada por administradores de rede, pois é uma ferramenta leve e de fácil instalação que é capaz de fazer o escaneamento de rede com grande confiabilidade. Assim, o objetivo aqui é fazer uma descrição mais detalhada do programa e mostrar onde baixar e como instalar.

O código fonte utilizado é o C e é de domínio público, por isso o código é constantemente atualizado e as regras de detecção também.

Os módulos do Snort são capazes de monitorar tanto o cabeçalho quanto o conteúdo dos pacotes e ainda disponibiliza a opção de capturar uma sessão inteira.

O Snort faz monitoração do tráfego de pacotes em redes IP, sobre diversos protocolos (rede e aplicação) e sobre conteúdo (hexadecimal e ASCII). Através de argumentos na linha de comando é possível ativar o subsistema de registros e alertas, onde existem três opções de registros e cinco de alertas. Esses registros podem ser configurados para armazenar pacotes decodificados em uma estrutura de diretórios baseados em IP, ou no formato binário do tcpdump. Esse registro pode ser desabilitado para um melhor desempenho permanecendo assim somente os alertas.

Alertas poderão ser enviados a um editor de texto como texto puro. Existe a opção também de desabilitar os alertas.

Uma boa opção para aumentar a base de dados do Snort é utilizar o Sniffer. Este irá capturar uma parte do texto ou uma *string* binária que passou pela ferramenta de ataque para o servidor. Assim basta adicionar os caracteres significativos dessas *strings* como um descritor de conteúdo do Snort.

7.1.1 Requisitos de sistema

Para a implementação do Snort, deve-se levar em consideração alguns fatores. Por exemplo, os dados gerados pelo Snort necessitarão de espaço em disco suficiente para sua armazenagem. Caso o administrador queira uma monitoração remota deve-se instalar o SSH e Apache com SSL, quando estamos trabalhando com Linux ou Unix. Caso o sistema seja Windows temos que ter um Terminal Services com limitações sobre quais usuários e máquinas podem se conectar e servidores IIS (Internet Information Server).

7.1.2 Componentes de Hardware

Se o Snort estiver no modo de sistema de detecção baseado em rede (SDIR), será necessária uma unidade de disco rígido bastante grande devido ao número de dados gerados.

É interessante uma segunda interface Ethernet. Uma interface para conectividade típica (SSH, serviços da Web e etc) e outra para o uso no Snort que funcionará como um sensor.

A placa de interface de rede também é importante, pois, se a rede for de 100MB a placa deverá ser também de 100MB, para se poder utilizar toda a largura de banda.

7.1.3 Plataforma

O Snort funciona em praticamente todos os sistemas operacionais da atualidade. Entre eles estão: Linux, FreeBSD, NetBSD, OpenBSD, Windows, e também, Sparc Solaris, PowerPC MacOS X e MKLinux, e PA-RISC HP-UX.

7.1.4 Opcionais

Alguns softwares podem ser instalados para acrescentar funcionalidades na utilização do Snort:

MySql, Postgres ou Oracle (bancos de dados SQL);

Smbclient se estiver usando mensagens WinPopup;

Apache ou outro servidor *web*;

PHP ou Perl, se tiver *plug-ins* que os exigem;

SSH para acesso remoto (ou Terminal Server com o Windows);

Apache com recursos de SSL para monitoração (ou IIS para Windows).

7.1.5 Arquitetura

O Snort oferece um conceito de plug-ins que são utilizados para personalizar a implementação de seus componentes, que são: o Farejador, o Pré-processador, o Mecanismo de detecção e a Saída.

Esses *plug-ins* são programas que são escritos para se adaptarem com a API de *plug-in* do Snort.

7.1.6 Farejador

O farejador é basicamente um sensor que ouve todos os pacotes IP que passam pela rede. Esses pacotes podem ser de vários tipos de tráfegos de rede, incluindo TCP, UDP, ICMP, então o farejador pega esses pacotes e os torna em algo legível para o administrador.

7.1.7 Pré-Processador

Algumas das funções de um pré-processador são: a remontagem de pacotes, a decodificação de protocolos e a detecção baseada em anomalias. O pré-processador verifica os pacotes com vários tipos de plug-ins, assim, definido seu comportamento envia para o mecanismo de detecção.

7.1.8 Mecanismo de detecção

O Mecanismo de detecção é o componente mais importante no Snort, pois aí que são configuradas as regras para análise dos dados, e onde se decide se os dados vão ser recebidos, se vai ser gerado um alerta, ou se os dados vão ser descartados.

Uma regra consiste basicamente de duas partes:

O cabeçalho da regra: que é a ação a ser executada (um log ou alerta), é onde está o tipo de pacote (se é UDP, TCP, ICMP e etc), endereços IP e portas de origem e destino;

A opção da regra: é o conteúdo do pacote, que deve corresponder à regra.

Para uma melhor otimização do programa o administrador poderá escrever regras no IDS.

7.1.9 Saída

A saída é um componente que alerta e registra os ataques direcionados à rede. Toda vez que o mecanismo de detecção registra algum ataque, um alerta é gerado. Esses alertas podem ser enviados para um arquivo de eventos, por meio de uma conexão de rede, através de soquetes UNIX ou Windows *Popup* (SMB). Também podem ser enviados a um banco de dados SQL.

Através de arquivos *syslog* (*arquivos de eventos*) ligados a um servidor de Web os alertas podem ser enviados via e-mail para o administrador.

7.1.10 Problemas na execução do Snort

Todos os programas têm seus pontos fracos e o Snort também têm alguns. Os principais são: não pegar todos os pacotes, alertas de falsos positivos e alertas de falsos negativos.

O Snort pode não pegar todos os pacotes devido à velocidade da rede e à velocidade da interface promíscua.

O alerta de falso positivo acontece quando o Snort alerta sobre determinada invasão e ela não existe, ou melhor, trata-se de uma ocorrência normal que o administrador não precisava ser alertado. Isso acontece quando se utiliza um conjunto de regras padrão do Snort. Já o falso negativo é o contrário. É quando não é detectado uma invasão ou comprometimento de um sistema monitorado por um IDS.

7.1.11 Melhorando a segurança do Snort

O Snort está sujeito a todo tipo de ataques e há várias maneiras disso acontecer. O administrador geralmente acessa o sistema de forma remota (SSH), armazena os dados coletados em um banco de dados (MySql ou Postgres) e também utiliza interfaces para visualização dos alertas gerados o que necessita de um servidor Web como o Apache ou o

IIS. Por isso é sempre importante estar atento as novas vulnerabilidades de segurança e nos anúncios de segurança de sistemas operacionais.

Para melhorar a segurança do Snort alguns procedimentos podem ser tomados:

- Desative os serviços que não são necessários para o sistema – por exemplo, FTP, NFS e NIS;
- Mantenha a integridade do sistema – o Tripware protege contra Cavalos de Tróia;
- Use *firewall* ou envelope (tradução) TCP nos serviços – Alguns serviços são vulneráveis como SSH e MySql por isso devem ser traduzidos em TCP ou protegidos com *firewall*, pois estes serviços também têm seus problemas com segurança;
- Utilize criptografia e autenticação de chave pública – Na utilização do Apache para ver eventos, é recomendável que se utilize o Apache-SSL e certificados digitais para autenticação no lado do cliente;
- Atualize aplicativos e sistema operacional – É importante atentar para anúncios referentes aos aplicativos e sistema operacional utilizado independente do sistema.

7.1.12 Instalando o Snort

Após tanto falar do Snort colocamos neste tópico algumas instruções de como fazer à instalação do Snort e de quais ferramentas podem auxiliar na implementação do programa.

Cabe salientar três programas: Snort + ACID + MySql.

O Acid (*Analisys Console for Intrusion Databases*) é muito utilizado para analisar eventos do Snort e para sua apresentação em uma interface WEB.

Já o MySql será utilizado como banco de dados para armazenar os registros de ataques .

7.1.13 Pré-Requisitos

- Linux;
- Servidor Web Apache;
- Interpretador PHP4;
- MySql.

O download dos pacotes acima poderão ser encontrados em: http://www.linuxpackages.net/

7.1.14 Instalação

Entre no site http://www.snort.org/dl/ e faça o download do arquivo de instalação do SNORT. Descompacte o arquivo:

```
# tar xzvf snort-2.6.0.tar.gz
# cd snort-2.6
```

Compilando o snort com suporte ao MySql.

```
# ./configure  with-mysql=/usr
# make
# make install
```

Se tudo correu bem seu Snort está instalado, mas ainda faltam alguns ajustes. Crie uma pasta chamada “regras” no /etc:

```
# mkdir /etc/regras
```

Para baixar as regras ou *rules*, em inglês, entre no site http://www.snort.org/pub-bin/downloads.cgi Após o download descompacte o arquivo na pasta /etc/rules

```
# tar xzvf snortrules-*.tar.gz
# mv snortrules-*/* /etc/rules/
```

7.1.15 Base de dados do SNORT

Primeiro é necessário criarmos a base de dados que será usada para armazenar o registro dos ataques, e um usuário que terá permissão para adicionar esses registros na base de dados.

```
# mysql  -u root  -p

mysql> create database snort;
mysql> grant all privileges on snort.*  to snort@localhost identified by  12345;
```

mysql> quit

Agora iremos criar as tabelas na database para o funcionamento do SNORT. Entre no diretório onde o snort foi descompactado e execute o comando abaixo:

mysql -u root -p snort < schemas/create_mysql

7.1.16 Configuração do SNORT

O arquivo de configuração do SNORT encontra-se na pasta /etc. Você deverá editá-lo de acordo com suas necessidades. Alguns campos devem ser alterados obrigatoriamente, para isso siga as instruções abaixo:

vi snort.conf

Agora altere o campo var HOME_NET para o IP da máquina ou rede que irá monitorar, e também o campo var DNS_SERVERS para o endereço do seu DNS, evitando assim que o SNORT crie alerta com os acessos vindos do mesmo.

Procure a linha abaixo e deixe-a descomentada.

output database: log, mysql, user=snort password=12345 dbname=snort host=localhost

Essa linha faz com que todos os logs do SNORT sejam gravados no banco de dados para que posteriormente sejam mostrados pelo ACID.

Feito isso salve e saia do arquivo (:wq!).

7.1.17 Configuração do ACID

Você pode encontrar o ACID para download no site http://www.cert.org/kb/acid.

Também é necessário baixar o ADODB para o perfeito funcionamento do ACID. O download pode ser feito no link:
http://prdownloads.sourceforge.net/adodb/adodb491.tgz?use_mirror=ufpr

Após o download do arquivo descompacte-o na pasta /var/www/htdocs

```
# tar xzvf acid-0.9.6b23.tar.gz -C /var/www/htdocs
```

Descompacte o ADODB dentro da pasta acid.

```
# tar xzvf adodb491.tgz -C /var/www/htdocs/acid
```

Agora entre na pasta criada e edite o arquivo acid_conf.php

```
# cd /var/www/htdocs/acid/
# vi acid_conf.php
```

Então altere os campos abaixo:

```
$DBlib_path = "/var/www/htdocs/acid/adodb";
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snort";
$alert_password = "12345";
```

O próximo passo da configuração será feito pelo seu navegador. Abra um navegador de sua preferência e entre no endereço http://localhost/acid

Depois de abrir o endereço acima, clique no *link Setup Page*, depois disso ele estará pronto para o uso.

7.1.18 Iniciando e Testando o Snort

Para iniciar o Snort digite a linha de comando abaixo:

```
# /usr/local/bin/snort -D
```

Para testa-lo, entre em algum outro computador e tente varrer as portas abertas no servidor onde está o SNORT.

Supondo que a máquina onde o SNORT está rodando tenha o IP 10.21.0.4, vou usar a máquina 10.21.0.5 para varrer as portas:

nmap 10.21.0.4

Agora na máquina 10.21.0.4 entre no seu navegador e entre no endereço http://localhost/acid

Se durante a instalação tudo correu bem e o snort foi iniciado sem erros, ele irá apresentar as tentativas de invasão [12].

7.2 Ossec HIDS

Como já foi dito um *host* IDS faz a monitoração de eventos de servidores e hosts, detectando atividades suspeitas. Um IDS por intermédio de suas assinaturas analisa os eventos e seus comportamentos, podendo tomar decisões até mesmo de interromper todo o tráfego direcionado ao *host* que está sofrendo um ataque.

O programa OSSEC HIDS é Opens Source e foi criado por Daniel Cid. É utilizado para fazer a análise de eventos, gerar alertas e trazer respostas pró-ativas que são atitudes já de IPS, mas que o programa também realiza.

Este programa pode ser baixado em http://www.ossec.net/files/ossec-hids-1.1.tar.gz

7.3 Appliance

Appliance é todo equipamento que foi projetado e configurado para realizar instruções específicas em um sistema. Esse equipamento é baseado a partir de um software genérico e otimizado para atender somente às atividades principais. Pode-se dizer que é uma junção bem conveniente entre hardware e software para determinado fim. Existem tipos de Appliances para diversas áreas tais como: informática, biologia, engenharia e matemática. Os Appliances são customizados para o fim a que se destina e são imutáveis.

Nas figuras 17 e 18 são mostrados dois modelos de appliance:

Figura 17 – Modelo de appliance Fortigate – 60

Características:

- Fabricante: Fortinet;
- Especificações: *Firewall*, antivírus, anti-spam, VPN.
- Vantagens: Duas portas 10/100 mantêm a conexão redundante com a Internet;
- Desvantagem: Na configuração padrão não guarda em arquivos de eventos as varreduras em portas feitas por possíveis invasores;
- Conclusão: Modelo adequado para ambientes de médio porte. [13]

Figura 18: Modelo de appliance Safe@Office 425 w

Características:

- Fabricante: CheckPoint;
- Especificações: Roteador, VPN, *firewall*, ponto de acesso sem fio;
- Vantagem: Possui porta específica para servidores fora do *firewall* ou segundo link de longa distância.
- Desvantagem: Funções adicionais, como antivírus e DNS, devem ser adicionadas separadamente;
- Conclusão: Equipamento versátil, adequado para empresas de médio porte. [13]

7.3.1 Appliances McAfee IntruShield Network IPS

É uma proteção pró-ativa de infra-estruturas de rede e dos terminais contra ataques desconhecidos, de DoS, spywares, ataques à VoIP, botnets, programas mal-intencionados, phishing e ataques criptografados, utilizando uma prevenção contra intrusões, de grande porte e que *reconhece riscos*, disponível no

Appliance McAfee IntruShield Network IPS. [14]

Este equipamento permite através de seu sistema de prevenção de intrusos que empresas e provedores de serviços aumentem sua segurança. Ele é baseado em um Circuito Integrado de Aplicação Específica que protege de forma pró-ativa a rede contra ataques conhecidos e desconhecidos, ataques de negação de serviço e criptografados, vulnerabilidades de VoIP, cavalos de Tróia etc.

Este equipamento é uma solução de IPS com reconhecimento de riscos que identifica e bloqueia as ameaças e ataques direcionados aos recursos da rede. Este bloqueio é realizado antes mesmo que os alvos sejam atingidos. A plataforma é de fácil gerenciamento o que permite uma proteção maior dos recursos da rede.

7.4 HLBR

O HLBR - Hogwash Light BR é um projeto brasileiro, criado em novembro de 2005, derivado do Hogwash (desenvolvido em 1996) por Jason Larsen. Este projeto é destinado à segurança em rede de computadores.

O HLBR é um IPS capaz de filtrar pacotes diretamente na camada dois do modelo OSI (não necessita de endereço IP na máquina). A detecção de tráfego malicioso é baseada em regras simples (o próprio usuário poderá confeccionar novas regras). É bastante eficiente e versátil, podendo ser usado até mesmo como *bridge* para *honeypots* e *honeynets*. Como não usa a pilha TCP/IP do sistema operacional, ele é "invisível" a outras máquinas na rede e atacantes.

Algumas características:

- Encontra-se na versão 1.1 sua 6ª versão;
- É instalado na camada dois do modelo OSI porém, atua também nas camadas 3, 4, 5 e 7;
- É um software livre;
- Esta disponível em http://hlbr.sourceforge.net.

8 Conclusão

Implantar um IDS ou um IPS não é tão fácil como parece. Deve-se levar em conta os sistemas de criptografia, a topologia da rede e os equipamentos, a velocidade da rede etc.

De maneira geral parece que os sistemas baseados em host atendem na maioria dos casos. Os fabricantes desses tipos de produtos já estão com a preocupação para soluções de monitoração tanto do tráfego enviado para a rede, como das atividades internas.

Cabe salientar que, em se tratando de IDS e IPS uma solução única não é adequada, pois existem vários tipos de situações em um mesmo ambiente.

Com relação à implementação do conceito de Entropia na detecção de anomalias ou intrusos, vimos que o conceito introduzido de Shannon para Entropia pode-nos ajudar bastante para construção de sistemas capazes de fazer a detecção de intrusos, porém vimos também que a Entropia Não – Extensiva de Tsallis permite uma melhor avaliação e variação na percepção do nível de caoticidade de um sistema.

Por fim vimos alguns programas de IDS e ferramentas de IPS que estão disponíveis no mercado e que devemos atentar na hora de escolher qual é a mais adequada para uma possível implementação.

Referências

[1] IDS – Conceito. Disponível em: <http://pt.wikipedia.org/>;

[2]CAUDLE Rodney. Assumptions in Intrusion Analysis Gap Analysis. Disponível em: <http://www.sans.org/reading_room/papers/download.php?id=1751>.

[3] Segurança de Redes IDS e IPS. Disponível em: <http://www.dei.unicap.br/~almir/seminarios/2004.2/ts04/ipsids/index.html>.

[4] Detecção de Intrusos com Snort. Disponível em: <http://www.4linux.com.br/whitepaper/snort_418.php;

[5] ANTUNES Chen Leonardo. Diferenças entre IDS e IPS. Disponível em: <**http://www.mettasecurity.com.br/artigo003.html>;**

[6] SILVEIRA Klaubert Herr da. Desafios para os sistemas de Detecção de Intrusos (IDS). Disponível em: <http://www.rnp.br/newsgen/0011/ids.html>;

[7] SOLHA Liliana Esther Velásquez Alegre; TEIXEIRA Renata Cicilini; PICCOLINI Jacomo Dimmit Boca. Tudo que você precisa saber sobre os ataques DDoS. Disponível em: <http://www.rnp.br/newsgen/0003/ddos.html>;

[8] NOGUEIRA Toniclay Andrade BATISTA Othon Marcelo Nunes. Negação de Serviço: Implementações, Defesas e Repercussões. Disponível em: <http://wwwlinhadecodigo.com.br>;

[9] GLEISER Marcelo.Tempo, Vida e Entropia. Disponível em: <http://www.fisicabrasil.hpg.ig.com.br/tempo_entropia.html>;

[10] Teoria da Informação – Conceito. Disponível em: <http://pt.wikipedia.org/>;

[11] SHANNON, C. E. – 1948. A mathematical theory of communication. The Bell System Technical Journal;

[12] OLIVEIRA Fred I. de. Snort + ACID + MySQL. Porto Velho/RO. Disponível em: <http://www.dicas-l.com.br>.

[13]Info Online / Guia de Produtos / Produtos de Segurança. Disponível em: <http://info.abril.com.br/produtos/detalhe>;

[14] Appliances McAfee IntruShield Network IPS. Disponível em: <http://www.mcafee.com/br/enterprise/products/network_intrusion_prevention

[15] STANGER James; LANE T.Patrick; DANIELYAN Edgar. Rede Segura Linux. Editora Alta Books.

[16] RAMALHO, NICOLAU, TOLEDO. Os Fundamentos da Física 2. Editora Moderna.

[17] CASTWELL Brian, editor técnico da SNORT.org; BEALE Jay; FOSTER C. James; POSLUNS Jeffrey, Consultor Técnico. Snort 2 Sistema de Detecção de Intruso – Open Source. Editora Alta Books.

[18] MONSORES Marcelo Luís; ZIVIANE Artur; Rodriguez Paulo Sérgio Silva. Detecção de Anomalias de Tráfego usando Entropia Não-Extensiva. Laboratório Nacional de Computação Científica (LNCC/MCT).Trabalho com participação de aluno.