UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

CLAUS PACHECO LOTTIN

ANÁLISE E COMPARAÇÃO ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO

CRICIÚMA, JULHO DE 2011

CLAUS PACHECO LOTTIN

ANÁLISE E COMPARAÇÃO ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. MSc. Paulo João Martins

CRICIÚMA, JULHO DE 2011

CLAUS PACHECO LOTTIN

Análise e Comparação entre Sistemas de Detecção de Intrusão

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

Prof. MSc. Paulo João Martins (UNESC)
Orientador

Prof. Esp. Sérgio Coral (UNESC)

Prof. MEng. Evânio Ramos Nicoleit (UNESC)

Dedicado aos meus pais!

RESUMO

Atualmente é raro existir uma empresa sem uma rede de computadores. Com o aumento considerável do tráfego de informações nas redes de computadores também há um aumento proporcional na utilização de serviços voltados para Internet, como por exemplo, e-mails, acessos a sites, transmissão e recepção de arquivos. Esses serviços facilitam as atividades dos usuários. Porém, quando utilizados inadequadamente podem colocar a rede em risco, as expondo a possíveis invasões do sistema. Para evitar esses problemas, é necessária uma política de segurança utilizada em conjunto com ferramentas para auxiliar na segurança, como *firewalls*, antivírus e *Intrusion Detection System* (IDS). Ferramentas IDS detectam ataques à rede e reportam-se ao administrador, gerando alertas antes que possam causar danos significativos. As ferramentas de detecção de intrusão estão se tornando cada vez mais utilizados por empresas, pois *firewalls* e antivírus não dão total segurança para a rede devido à variedade de ataques existentes atualmente. Hoje em dia existem várias ferramentas IDS disponíveis, dentre elas destacam-se as *open source*, com facilidade de acesso e com licença de uso livre, e algumas ferramentas proprietárias. Dessas ferramentas foram escolhidas duas, Snort (*open source*) e RealSecure (proprietária), para serem submetidas a testes em um ambiente montado para a análise. Os resultados obtidos são descritos e comparando-se o comportamento das duas ferramentas, informando-se as deficiências e as vantagens de seu uso.

Palavras-chave: IDS; Snort; *RealSecure*; Segurança; Redes.

ABSTRACT

Nowadays is rare to exist a company without a computer network. With the increase in the traffic of information on computer networks there is also a proportional increase in the utilization of Internet services, for example, e-mails, websites, transmission and reception of files. These services facilitate the activities of users. However, when used inappropriately can put the network at risk, exposing the system to possible intrusions. To avoid these problems, requires a security policy used in combination with tools to assist in security, like firewalls, antivirus and Intrusion Detection System (IDS). IDS tools detect network attacks and report to the administrator, generating alerts before they can cause significant damage. The IDS are becoming more used by companies, because firewalls and antivirus do not provide total security to the network due to the variety of attacks available today. There are several IDS tools available, among them are noticeable the open source tools, with easy access and free use license, and some proprietary tools. Among these tools, two were selected, Snort (open source) and RealSecure (proprietary) to be subjected to trials in an environment built for analysis. The results obtained are described, comparing the behavior of tools, reporting the deficiencies and advantages of its use.

Keywords: IDS; Snort; RealSecure; Safety; Network.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ARPA	*Advanced Research Projects Agency*
ARPANET	*Advanced Research Projects Agency Network*
DARPA	*Defense Advanced Research Projects Agency*
DDoS	*Distributed Denial of Service*
IANA	*Internet Assigned Numbers Authority*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
ISO	*International Standart Organization*
MIT	*Massachusetts Institute of Technology*
NPC	*Network Control Protocol*
OSI	*Open System Interconnection*
OSSTMM	*Open Source Secutory Testing Methodology Manual*
SRI	*Stanford Research Institute*
TCP	*Transmission Control Protocol*
UCLA	*University of California, Los Angeles*
UCSB	*University of California, Santa Barbara*
UTAH	*University* of *Utah*
UDP	*User Datagram Protocol*

SUMÁRIO

1 INTRODUÇÃO

A segurança da informação é importante para o sucesso de uma empresa no ambiente virtual de negócios, sendo responsáveis por decisões estratégicas que fazem a diferença em um mercado onde a vulnerabilidade das informações é uma grande ameaça à rentabilidade empresarial e credibilidade no mercado. Por isso é importante proteger as informações contidas nas redes, seja de uma empresa ou em uma rede pessoal, já que é possível perder muito somente pelo fato de algum desconhecido extrair os dados dessa rede (BRANDÃO, 2008).

A segurança é uma das maiores preocupações, se não a maior preocupação, na elaboração de um projeto de rede de computadores pelo fato da importância da informação. A utilização de ferramentas *firewalls* e antivírus nem sempre conseguem proteger totalmente a rede, sendo que esta ainda fica sujeita a ataques de pacotes infectados devido aos variados tipos e formas de ataque existentes. A segurança das redes é um assunto bastante abrangente, e que se preocupa com que pessoas não leiam ou modifiquem mensagens enviadas a outras pessoas ou com o acesso a serviços remotos aos quais não estão autorizadas (TANENBAUM, 2003).

Devido a esses pacotes hostis, cria-se uma necessidade de melhorar a proteção dos dados da rede e servidores, surgindo formas diferentes de aumentar a segurança. Um modo de aumentar a segurança em uma rede é a utilização de um Sistema de Detecção de Intrusão (IDS), que são sistemas que monitoram o tráfego da rede, analisando os pacotes, comparando-os com assinaturas de ataques conhecidas e notificando o administrador caso exista algum risco.

Existem vários IDS, entre eles existe o Snort, que é capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP e, além disso, é uma ferramenta

open source. Outros IDS, como por exemplo, o *RealSecure*, operam de maneira semelhante, porém com configurações, bibliotecas e técnicas diferenciadas.

Apesar das vantagens que um IDS proporciona, existem duas situações indesejáveis que frequentemente surgem num ambiente que utiliza mecanismos de detecção de intrusão. Uma é chamada de falso positivo (ou falso alerta) e a outra é o falso negativo, ou seja, o ataque real, porém não detectado (VAZ, 2004).

Esses alertas podem ser gerados de maneira errônea devido à má configuração ou tratamento das regras utilizadas. Outro grande problema é saber qual dos IDS utilizar justamente devido à sua variedade de tipos e de modos de detecção. Para avaliar o desempenho das ferramentas existem diversas metodologias, as quais serão descritas ao longo deste projeto. Na metodologia de Fagundes, por exemplo, os IDS são testados sem ter a necessidade de conhecer a arquitetura interna das ferramentas, focando no desempenho de cada uma, que é o objetivo maior do projeto. Utilizando o Manual de Metodologia Aberta de Comprovação de Segurança (OSSTMM), que é um dos documentos mais completos e é comumente utilizado em segurança da informação, também é possível realizar comparações entre as ferramentas, mostrando onde sua rede falhou quando sofreu o ataque.

Tendo isso como base, o objetivo deste projeto é descrever as metodologias de comparação e definir alguns cenários para realizar comparações entre ferramentas IDS, de forma que se consiga apresentar uma análise da sua potencialidade e identificar algumas de suas características operacionais, conseguindo assim contribuir com administradores e usuários finais, auxiliando no entendimento do desempenho de cada um, auxiliando na compreensão de seus alertas contra tentativas de ataques e verificando os falsos positivos gerados por cada uma delas.

1.1 OBJETIVO GERAL

Analisar e comparar IDS no que diz respeito à quantidade gerada de falsos positivos, utilizando metodologias de comparação.

1.2 OBJETIVO ESPECÍFICO

Os objetivos específicos da pesquisa são os seguintes:

a) enunciar e utilizar conceitos sobre segurança de redes.

b) compreender e aplicar os conceitos de funcionamento dos IDS.

c) descrever metodologias de comparação entre IDS.

d) analisar e definir quais os IDS a serem utilizadas ao longo do projeto.

1.3 JUSTIFICATIVA

Segundo Borges e Coutinho (2007) atualmente houve um crescimento em relação à utilização da Internet e consequentemente aumentou o número de fraudes nas redes. Informações com conteúdo importante de certa forma ficam vulneráveis a pessoas com más intenções, e tê-las extraídas da rede pode comprometer seriamente a empresa ou a pessoa que teve essa informação roubada.

As intrusões detectadas são geralmente o resultado de operações realizadas por pessoas que desejam causar danos, se concretizando assim, uma guerra de informações existindo jogadores ofensivos e defensivos. Por esse motivo, tudo o que é feito para ter uma segurança melhor é impulsionado pela ameaça, e as vulnerabilidades nos sistemas são os portões por onde ela se manifesta (NORTHCUTT, 2001).

Para que uma rede se encontre com certo grau de segurança é importante a utilização de *firewall*s, antivírus e outras ferramentas que tem como objetivo fornecer essa possível segurança, mas nem sempre isso é o suficiente. Os sistemas de detecção de intrusão (IDS) surgiram como uma forma de cobrir falhas na segurança e alertar o administrador da rede caso haja alguma tentativa de invasão. Porém existem hoje em dia diversos tipos de ferramentas IDS, cada uma com suas características.

Devido a essa variedade de tipos de IDS, o administrador de redes fica com uma dúvida de qual software utilizar. Por isso é necessária uma comparação para demonstrar como comparar e permitir um grau maior de conhecimento aos administradores e usuários finais, e assim tenham como decidir qual delas é mais adequada para a o perfil de sua rede. Para isso, serão utilizadas metodologias de comparação definidas, como a de Fagundes ou a de Pucketza, ajudando a organizar o processo e prover uma base para a avaliação das respostas de falsos positivos e negativos. A metodologia de Fagundes se encaixa melhor no projeto devido ao fato de ser exclusivamente voltada para o teste de desempenho, não se aprofundando na arquitetura interna de cada IDS, porém não é descartada a utilização de outros tipos de metodologias ao longo do projeto, pois cada uma tem uma forma de comparação diferente que pode ser aproveitada.

Segundo Hezog (2009) o Manual de Metodologia Aberta de Comprovação de Segurança (*Open Source Security Testing Methodology Manual* - OSSTMM), como já citado anteriormente, é um dos documentos mais completos e comumente utilizado em segurança da informação, mostrando onde sua rede e quando sofreu o ataque. Os testes realizados com sucesso pelo OSSTMM englobam: controle de dados e informações, níveis de segurança, níveis de controle de fraude, redes de computadores e telecomunicações, dispositivos sem fio, dispositivos portáteis, controles de acesso, segurança de bases militares, entre outros. O OSSTMM é a compilação da evolução de uma série de normas, padrões e boas práticas. Mais

que isso, ela permite ao auditor de segurança medir com precisão quão desprotegida está a sua rede. Utilizando-se de metodologias de teste combinadas com o OSSTMM é possível comparar os IDS e apresentar resultados confiáveis sobre o desempenho de cada um.

1.4 ESTRUTURA DO TRABALHO

Este trabalho foi organizado em sete capítulos. O Capítulo 1 contextualiza a introdução referente ao trabalho, com a definição do objetivo desse estudo e também os objetivos específicos e concluindo então com a justificativa do trabalho realizado.

Os conceitos no Capítulo 2 visam apresentar como a história sobre as redes de computadores e alguns protocolos utilizados para seu funcionamento enquanto no Capítulo 3 é explicada a importância da segurança de rede e citadas algumas ameaças constantes.

O Capítulo 4 apresenta as ferramentas IDS, falando como funcionam, suas vantagens e desvantagens, e também sobre alguns problemas na sua implantação.

No Capítulo 5 são abordadas as ferramentas IDS que participaram da pesquisa, mostrando um pouco sobre suas características, sua história e seu funcionamento.

O Capítulo 6 apresenta alguns trabalhos correlatos na área de redes e segurança de redes.

Ao decorrer do Capítulo 7 são descritas as metodologias de comparação existentes, a metodologia utilizada no projeto, os tipos de ataques que foram aplicados, quais foram as ferramentas utilizadas para a simulação dos mesmos e é descrito como foi montado o ambiente onde foram feitos os testes. Também se encontram os resultados obtidos dos testes realizados com as ferramentas IDS.

As conclusões são apresentadas ao final do trabalho, contendo a síntese de tudo o que foi apresentado ao longo do trabalho.

2. CONCEITO DE REDE

Segundo Tanenbaum (2003) pode-se definir redes de computadores como um conjunto de computadores autônomos interconectados por uma única tecnologia, que podem trocar informações. Os mesmos podem compartilhar o uso de equipamentos que estejam instalados em um destes computadores conectados a rede. O modo que esses computadores se encontram interligados varia muito, indo de fibras ópticas até sinais de satélite, encontrando a melhor forma de se ajustar com a necessidade e o tamanho da rede. Hoje em dia existem redes em todos os lugares: residências, empresas pequenas, grandes empresas, multinacionais, entre outras.

A rede mais usada atualmente é a famosa Internet que é um conjunto de redes diferentes interligadas e utilizando protocolos comuns, fornecendo serviços e fazendo que computadores do mundo inteiro se comuniquem entre si segundo Tanenbaum (2003).

Esse conceito de rede vem de uma grande evolução das redes de computadores ao longo desses anos e por meio de muito estudo e diversas tentativas de montar uma rede que chegamos onde estamos e temos a tecnologia e os padrões que utilizamos atualmente.

2.1 HISTÓRICO

As primeiras redes de computadores foram criadas ainda durante a década de 60, com o objetivo de transferir informações de um computador a outro. Naquela época, o meio mais usado para armazenamento externo e transporte de dados ainda eram os cartões perfurados, que armazenavam poucas dezenas de caracteres cada.

Nessa época, precisamente no ano de 1965 nos EUA, dois cientistas, Lawrence Roberts e Thomas Merril, fizeram o primeiro experimento de conexões de computadores em

rede. A experiência foi realizada por meio de uma linha telefônica discada de baixa velocidade, conectando dois centros de pesquisa nos Estados Unidos entre si, em Massachusetts e na Califórnia.

O começo da utilização de redes de computadores está diretamente ligado a corrida espacial que aconteceu durante a guerra fria. Uma grande parte de aplicações e protocolos para que aconteça a conexão em rede, como por exemplo, os protocolos TCP/IP, estão relacionados ao desenvolvimento da ARPANET (rede que originou a Internet), esta sendo criada por um programa desenvolvido pela Advanced Research Projects Agency (ARPA), que mais tarde foi renomeada para DARPA. A agência nasceu de uma iniciativa do departamento de defesa dos EUA, preocupado com a corrida tecnológica contra os russos. Roberts era um acadêmico do Instituto de Tecnologia de Massachusetts (MIT), um dos integrantes da DARPA e um dos criadores da ARPANET. Ele começou conectando quatro universidades: SRI, UCLA, UCSB e UTAH.

Esta rede inicial foi criada com propósitos de teste, com o desafio de interligar quatro computadores de arquiteturas diferentes, mas a rede cresceu rapidamente e em 1973 já interligava trinta instituições, incluindo universidades, instituições militares e empresas. Para garantir a operação da rede, cada nó era interligado a outros dois nós ou mais, de forma que a rede pudesse continuar funcionando mesmo com a interrupção de várias das conexões.

Em 1974 surgiu o TCP/IP, a princípio chamada *Network Control Protocol* (NPC), que acabou se tornando o protocolo definitivo para uso na ARPANET a partir de 1980 e eventualmente na Internet. Uma rede interligando várias universidades permitiu o livre tráfego de informações, levando ao desenvolvimento de recursos que usamos atualmente, como o e-mail, ou seja, esse livre tráfego de informações permitia aos usuários conectados trocarem informações, acessarem outros computadores remotamente e compartilhar arquivos com computadores que se encontravam em outras universidades.

2.2 MODELOS DE REFERÊNCIA

Esses modelos têm como principal objetivo permitir a definição de padrões para que tecnologias e equipamentos diferentes possam trocar informações e se comunicar livremente, como por exemplo, o TCP/IP e o modelo OSI.

Apesar de possuir o mesmo objetivo, os modelos apresentam características opostas. Protocolos do modelo OSI raramente são utilizados, atualmente apesar das características de cada camada sejam importantes. No TCP/IP os protocolos já possuem um uso geral maior que os do modelo OSI.

2.2.1 Modelo OSI

Modelo desenvolvido pela ISO no começo dos anos 80, servindo para padronizar os protocolos utilizados entre as camadas, fazendo com que haja comunicação em redes distintas.

Este modelo não é considerado uma arquitetura de rede, já que não mostra exatamente quais serviços e protocolos precisam ser usados em cada camada, mas sim diz o que cada camada deve fazer.

Segundo Tanenbaum (2003) o modelo é divido em sete camadas e as camadas são:

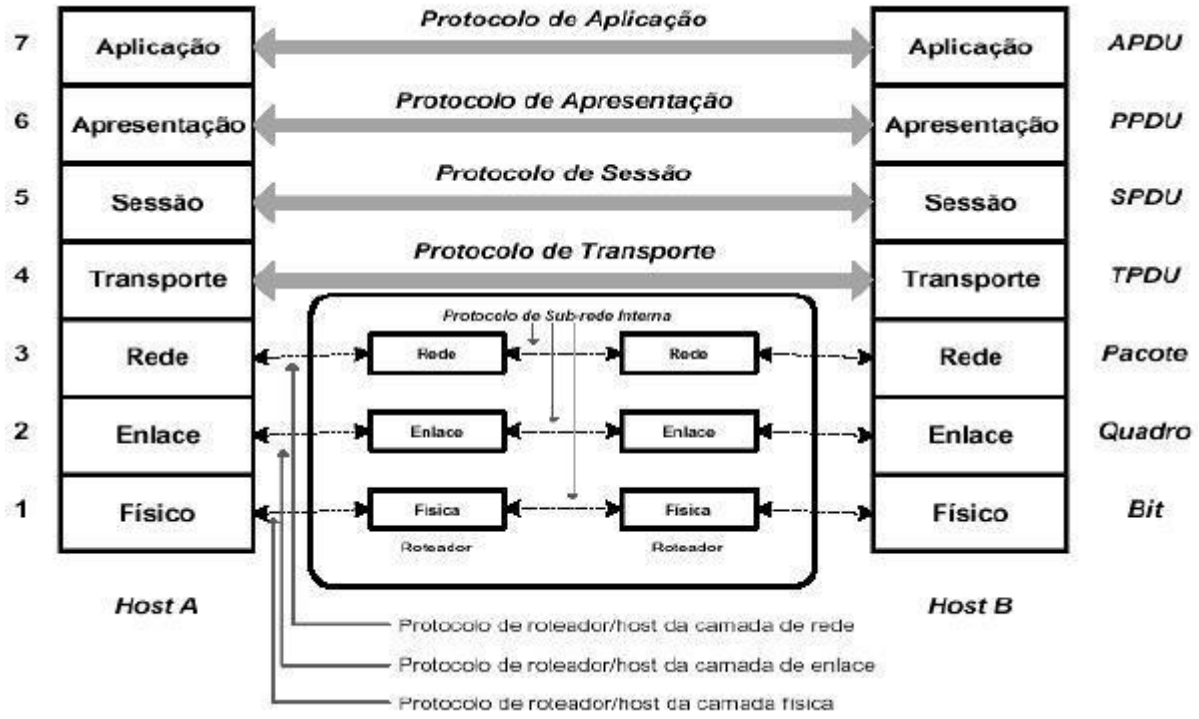

Figura 1. As sete camadas do modelo OSI
Fonte: TANENBAUM, A. (2003)

A camada física tem a função de fazer o tratamento de transmissão dos bits e a necessidade de manter a integridade desses bits. Está ligada com a parte elétrica, mecânica e de sincronização, assim permitindo uma comunicação confiável.

A camada de enlace faz com que um canal de transmissão bruto transforme-se em uma linha que aparenta ser livre de erros, fazendo assim que os dados de entrada sejam divididos em vários quadros e sendo transmitidos em sequência.

A camada de rede trata do endereçamento de pacotes, convertendo endereços lógicos em físicos, permitindo que os pacotes cheguem corretamente ao seu destino e o modo que eles são roteados. Também controla o congestionamento das sub-redes, ou seja, evita os gargalos quando muitos pacotes estão presentes na sub-rede ao mesmo tempo.

A camada de transporte recebe os dados da camada superior e se necessário, os divide em partes menores para que possam ser transmitidos de forma mais eficiente para a camada de rede. É a parte central de toda a hierarquia de protocolos. Sua tarefa é prover o

transporte econômico e confiável de dados, independente da rede física ou das redes atualmente em uso.

A camada de sessão permite que vários usuários estabeleçam sessões entre eles. Também controla e sincroniza o diálogo e gerencia a troca de dados entre entidades da camada de apresentação comunicantes.

A camada de apresentação ligada com a sintaxe e semântica das informações que são transmitidas.

A camada de aplicação é composta de protocolos comumente usados pelos usuários para transferir arquivos.

2.2.2 Modelo TCP/IP

Segundo Scrimger (2002) o TCP/IP não é um único protocolo, mas sim um conjunto de protocolos de comunicação entre computadores em rede e por causa dessa diversidade ele não utiliza diretamente o modelo OSI. Em vez disso, ele utiliza um modelo de quatro camadas para a comunicação: camada de aplicação, de transporte, inter-redes/Internet e *host*/rede.

Cada camada no modelo resolve um grupo de problemas de transição de dados, fornecendo um serviço bem definido para os protocolos das camadas superiores e já que os processos de comunicação são bem definidos e divididos em cada camada, alterações podem ser feitas isoladamente, não precisando rever todo o protocolo.

Segundo Tanenbaum (2003) as quatro camadas que formam o TCP/IP possuem certas características:

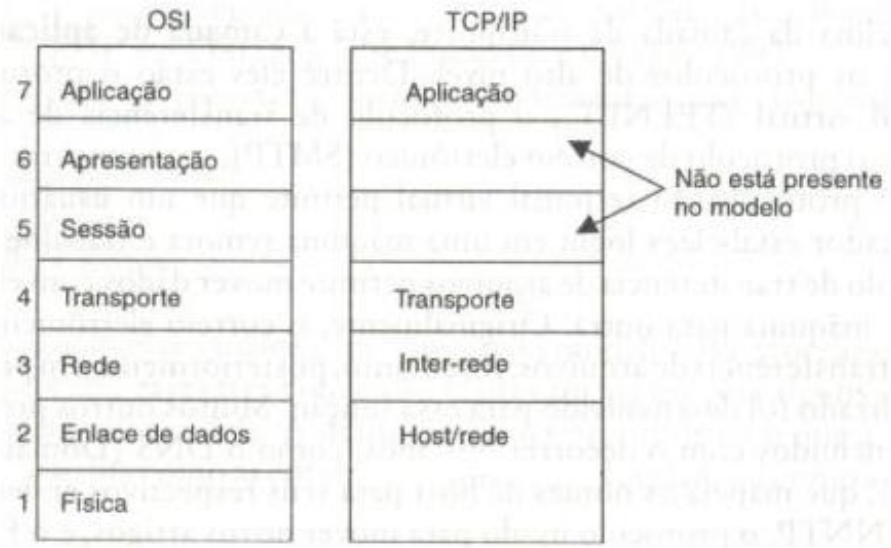

Figura 2. Camadas do modelo TCP/IP
Fonte: TANENBAUM, A. (2003)

Na camada de aplicação é onde estão os protocolos de nível mais alto que garantem o funcionamento correto das aplicações.

A camada de transporte é onde atuam os protocolos TCP e UDP. Essa camada permite que os *hosts* de origem e destino se comuniquem não importando a distância que estejam um do outro.

Camada inter-redes/Internet é a camada onde se encontra o protocolo IP. Essa camada faz com que pacotes transmitidos cheguem ao seu destino com ou sem falhas.

Camada *host*/rede é ligada diretamente com a placa de rede e pode trabalhar em diferentes padrões dependendo do meio onde está funcionando.

2.3 PROTOCOLOS PRINCIPAIS

Segundo Borges e Coutinho (2007) protocolo é uma convenção ou padrão que controla e possibilita uma conexão, comunicação ou transferência de dados entre dois sistemas computacionais. De maneira simples, um protocolo pode ser definido como as regras que controlam a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados pelo hardware, software ou por uma combinação dos dois. Dentre os protocolos que atuam no ambiente de rede alguns se destacam e entre eles se encontram: IP, TCP, UDP e ICMP.

2.3.1 Protocolo IP

Segundo Downes (2000 apud Steffen, 2003) o IP é um protocolo que se encontra na camada de rede e possui informações de endereçamento e controle que permitem o roteamento de pacotes na rede.

De acordo com Tanenbaum (2003) o protocolo IP foi projetado tendo como objetivo a interligação de redes desde o início e sua tarefa é fornecer a melhor forma de transmitir datagramas pela rede independentemente se as máquinas estiverem na mesma rede ou se existirem algumas redes entre a origem e o destino.

Apesar de o IP transportar o datagrama durante toda sua trajetória ele não dá garantia que um pacote chegue ao seu destino.

2.3.1.1 Endereçamento IP

O endereço IP identifica a localização de um *host* na rede, assim como um endereço de uma casa identifica a sua localização.

Cada endereço IP possui uma identificação de rede e uma de *host*. A identificação de rede indica em qual segmentação de rede o *host* esta e qualquer *host* da mesma rede deverá ter a mesma identificação. A identificação de *host* indica um *host* na rede, esse endereço deve ser único.

Um endereço de IP possui 32 bits divididos em quatro octetos de 8 bits, cada octeto é convertido em número de base decimal que abrange de [0-255] e são separados por ponto.

Segundo Melo (1997) para um maior auxílio no gerenciamento do endereçamento IP foram criadas cinco classes, que são divididas em A, B, C, D e E, sendo usadas para definir quantos bits são alocados para endereço de rede e de *host*s, podem ser usadas também para dimensionar o tamanho da rede.

De acordo com a *Internet Assigned Numbers Authority* (IANA) a classe A possui endereços de 1.0.0.0 até 127.0.0.0, o primeiro octeto (8 bits) é endereço de rede, os três últimos octetos (24 bits restantes) são endereços de *host*s, então teremos 126 redes e 16.777.214 *host*s por rede.

A classe B possui endereços de 128.0.0.0 até 191.255.0.0, os dois primeiros octetos (16 bits) são endereços de rede, os dois últimos octetos (16 bits restantes) são endereços de *host*s, então teremos 16.385 redes e 65.534 *host*s por rede.

A classe C possui endereços de 192.0.0.0 até 223.255.255.0, os três primeiros octetos (24 bits) são endereços de rede, o último octeto (8 bits restantes) é endereço de *host*s, então teremos 2.097.152 redes e 254 *host*s por rede.

A classe D possui endereços de 224.0.0.0 até 239.255.255.255, essa classe é usada para protocolos multicast.

A classe E possui endereços de 240.0.0.0 até 255.255.255.255, essa classe é experimental e reservada para uso futuro.

O primeiro endereço da rede e o último endereço da rede não são usados para endereçar *host*s por serem reservados.

2.3.2 Protocolo ICMP

ICMP é um protocolo integrante do IP e utilizado para fornecer relatórios de erros à fonte original. Qualquer *host* que utilize IP precisa aceitar as mensagens ICMP e alterar o seu comportamento de acordo com o erro relatado. Os gateways devem estar programados para enviar mensagens ICMP quando receberem datagramas que provoquem algum erro.

2.3.3 Protocolo TCP

TCP é um protocolo feito para obter conexões confiáveis, permitindo uma entrega de um fluxo de bytes para qualquer *host* na rede. A versatilidade e robustez deste protocolo tornaram-no adequado às redes globais, já que este verifica se os dados são enviados de forma correta, na sequência apropriada e sem erros.

2.3.4 Protocolo UDP

UDP permite que a aplicação escreva um datagrama encapsulado em um pacote e envie ao destino. Mas não há qualquer tipo de garantia que o pacote irá chegar ou não. É um

protocolo muito utilizado em aplicações cliente/servidor, onde a entrega imediata é mais importante que a entrega precisa e garantida.

A diferença básica entre o UDP e o TCP é o fato de que o TCP é um protocolo orientado à conexão e, portanto, inclui vários mecanismos enquanto o UDP é feito para transmitir dados pouco sensíveis, como fluxos de áudio e vídeo, ou para comunicação sem conexão, não necessitando de confiabilidade extrema na entrega de pacotes.

3 SEGURANÇA DA INFORMAÇÃO

A informação é um conjunto de dados que pode-se extrair conhecimento útil, já que um conjunto de dados aleatórios não significa nada se não tem alguém que os interprete (AURÉLIO, 2004).

Atualmente informação é um objeto de grande valor em um empresa e são tratadas como qualquer outro bem da empresa, talvez até com mais cautela e com mais segurança à sua volta.

Segundo Brandão (2008) a segurança da informação é importante para o sucesso de uma empresa no ambiente virtual de negócios, sendo responsáveis por decisões estratégicas que fazem a diferença em um mercado onde a vulnerabilidade das informações é uma grande ameaça à rentabilidade empresarial e credibilidade no mercado.

Com base nisso é importante que haja certa proteção das informações contidas nas redes de empresas ou mesmo redes pessoais já que é possível obter um grande prejuízo caso aconteça de informações valiosas serem roubadas.

3.1 AMEAÇAS

Segundo Borges e Coutinho (2007) ameaças são agentes que conseguem explorar as falhas de segurança do sistema provocando danos, perdas ou extraindo informações vitais de uma empresa, assim desestabilizando-a e provocando prejuízos. Essas ameaças podem ser:

a) naturais: algum fenômeno da natureza ou acidente como terremotos, incêndios que possam causar alguma perda na empresa;
b) intencionais: essas são causadas com a intenção de provocar prejuízo à empresa desde roubos de informações até sabotagens (*hackers*);

c) involuntárias: causadas geralmente pela falta de preparo dos usuários como, por exemplo, uma infecção por vírus, alterações do sistema provocando algum dano ou divulgação de senhas.

Cada vez surgem mais vulnerabilidades com o surgimento de novas tecnologias e um dos principais objetivos da segurança da informação é a correção dessas vulnerabilidades existentes, assim reduzindo os riscos a ameaças.

3.1.1 *Hacker*s

Segundo Aurélio (2004) *hackers* são programadores com gênio para dominar e alterar programas e equipamentos de computação e teleprocessamento, e capaz de invadir à distância outros computadores, utilizando ilegalmente os recursos do modem.

Porém *hackers*, segundo Nakamura e Geus (2003) não invadem para roubar informações ou causar prejuízo, mas sim para provar que conseguem e depois compartilhar com amigos os locais que conseguiram invadir.

Crackers são aqueles que invadem sistemas com a finalidade de roubar informações e causar prejuízos, porém o termo *cracker* não é muito utilizado e por isso os ataques são atribuídos ao termo *hacker*.

Hackers levam uma enorme vantagem contra administradores de sistemas, pois um *hacker* necessita somente de uma pequena abertura na segurança do sistema para conseguir invadir. Já o administrador precisa encontrar todas as vulnerabilidades possíveis do sistema para neutralizar essas aberturas que possibilitam as invasões.

3.1.2 Tipos de Ataques

Segundo Crothers (2000 apud Steffen, 2003) ataques podem ser detidos. Intrusões são ataques que cumpriram com seus objetivos enquanto um ataque é somente uma tentativa de intrusão e para que um ataque aconteça, é necessária uma estratégia para infiltração. *Hackers* seguem certa linha de pensamento que facilitam chegar onde querem:

a) pesquisa no *firewall* em busca de portas para serem atacadas;
b) infiltração no sistema devido a vulnerabilidades;
c) uma vez já infiltrado no servidor o *hacker* explora ainda mais a sua vulnerabilidade da segurança e tenta se tornar administrador da rede;
d) utiliza ferramentas para hackear o que ele deseja;
e) após tudo isso apaga seus rastros para não ser encontrado e identificado.

Coisas simples como utilização de senhas fortes e manter a segurança dos servidores atualizados podem salvar o sistema de ser invadido.

3.1.2.1 Ataques para Obter Informação

Para atacar com perfeição é importante conhecer bem a vulnerabilidade do alvo e planejar seus movimentos. Algumas técnicas conhecidas de ataque são:

a) engenharia social: conseguindo certa aproximação com funcionários da empresa e explorando a confiança deles, é possível conseguir senhas e informações necessárias para o ataque;
b) scanning de vulnerabilidade: realizando testes na rede a procura de falhas na segurança em protocolos ou no sistema operacional.

c) port scanning: mapeia as portas TCP e UDP a procura de informações referentes a serviços acessíveis. Sendo assim o *hacker* ataca diretamente agindo em cima de serviços ativos.

d) packet sniffing: procurar senhas e informações em todos os pacotes que trafegam na rede.

e) trashing: procura informações no lixo de empresas com a finalidade de encontrar desde manuais de sistemas até número de senhas.

f) IP Spoofing: o IP do *hacker* é mascarado. Essa técnica é utilizada para entrar em locais onde a identidade onde o IP é um identificador de acesso.

g) ataque físico: um ataque direto nos sofwares e equipamentos da empresa. Os invasores podem roubar os equipamentos e modificar arquivos.

3.1.2.2 Ataques de Negação de Serviços

É um tipo de ataque que faz o sistema receber várias requisições ao mesmo tempo, fazendo com que ele sobrecarregue e fique sem condições de uso. Dentre eles se destacam:

a) smurf: ataque onde um grande número de pacotes ping é enviado por broadcast para a rede com o endereço do alvo para receber as respostas. Todas as máquinas da rede receberão o pacote e retornarão a resposta para a mesma máquina, impossibilitando o alvo de executar tarefas de rede devido ao grande tráfego de pacotes, assim sofrendo uma negação de serviços;

b) bugs: os desenvolvedores do software deixam brechas no código, tornando o sistema suscetível a ataques.

3.1.2.3 Ataques contra TCP

Os principais ataques ativos contra TCP são os ataques coordenados e sequestros de conexão.

Ataque coordenado é conhecido como um ataque de negação de serviço distribuída, onde vários *host*s podem ser atacados por *hacker*s, sendo coordenados para atacar simultaneamente o mesmo alvo. É um tipo de ataque muito eficiente que deixa a vítima sem defesa e praticamente impossível de identificar de onde está vindo, já que o ataque vem de *host*s intermediários.

Os sequestros de conexão ocorrem quando o atacante se aproveita de uma falha na sincronização entre cliente e servidor, mantendo a conexão entre os dois, mas impossibilitando a sua troca de dados. Assim é possível criar pacotes e enviar dados para as máquinas, sequestrando sua conexão.

3.1.2.4 Ataques de Nível de Aplicação

As ameaças mais comuns são os vírus, worms e trojans e a maioria explora o buffer overflow.

Buffer overflow – Segundo Russell (2002) esse ataque compõe uma coleção de vulnerabilidades existentes e se forem exploradas corretamente, permite que o atacante execute um código malicioso com direito equivalente a um processo original. São considerados ataques de alto risco. Ocorre quando um programa recebe uma quantidade de dados muito maior do que está preparado para armazenar em *buffer*. Nessa quantidade de

dados pode possuir algum código com conteúdo mal-intencionado que será executado pelo sistema. Esse tipo de ataque é difícil de ser descoberto e um dos mais utilizados.

Tojans – Segundo Russell (2002) trojans (cavalos de tróia) são códigos que aparentam serem programas comuns e acabam se comportando de modo malicioso. É um ataque considerado limitado pelo fato de que o usuário precisa ser convencido a executá-lo.

Vírus – programas feitos para infectar sistemas e se espalhar em outras máquinas da rede. Segundo Russell (2002) é um fragmento de código que adentra um sistema operacional ou um *host* com a intenção de se propagar. Não tem a capacidade de ser executado independentemente e por isso precisa do *host*.

Worms – um tipo de vírus, cujo único objetivo é copiar a si mesmo e se espalhar o mais rápido possível. Segundo Russell (2002) é como o vírus, mas não se reproduz localmente, se propagando apenas entre sistemas e existindo somente na memória.

4 SISTEMA DE DETECÇÃO DE INTRUSÃO (IDS)

Neste capítulo serão abordados tipos, conceitos e métodos para a detecção utilizados por sistemas IDS.

4.1 DEFINIÇÃO DE IDS

Os IDS começaram a ser implementados por volta de 1980, sendo aprimorados ao longo do tempo se adaptando as novas tecnologias e se tornando eficiente em vários casos.

Esses sistemas são implementados com o objetivo de detectar, desviar e, quando possível, deter tentativas de invasões na rede ou servidor. Eles podem detectar ataques em portas legítimas e que por sua vez não são protegidas por *firewall*s.

Segundo Caswell (2003) IDS é uma ferramenta especializada que lê e interpreta o conteúdo de arquivos log de roteadores, *firewall*s, servidores e outros dispositivos da rede. Possui um banco de dados onde armazena assinaturas de ataques conhecidos e compara com padrões de atividade, tráfego ou comportamento da rede. O IDS compara os logs que monitora com suas assinaturas do banco de dados para detectar um possível ataque. Depois de detectado o ataque o IDS emite alertas para que o usuário decida o que fazer ou então, dependendo da sua configuração, age sozinho por meio de ações automáticas.

Normalmente um IDS é instalado em equipamentos como *firewall*s e roteadores, monitorando suas atividades.

Segundo Borges e Coutinho (2007) estes sistemas podem ser classificados em categorias baseadas em:

a) monitoramento:

- de redes: monitoram o *backbone* (espinha dorsal) da rede,

- de *hosts*: monitoram sistemas operacionais,

- distribuídos: IDS que operam em grupos reportando a um centro de gerenciamento;

b) detecção

-Por assinatura: como já dito anteriormente, compara logs do tráfego de rede com assinaturas de ataques conhecidos e se alguma semelhança é encontrada pode significar um ataque ao sistema;

-Por anomalias: usa algumas regras e conceitos pré-definidos utilizando heurísticas para distinguir anomalias que possam vir a ocorrer.

4.2 SISTEMA DE PREVENÇÃO DE INTRUSÃO (IPS)

Funciona como um *firewall* otimizado, fazendo com que todo tráfego passe por ele primeiramente antes de seguir adiante, sendo assim o sistema de prevenção de intrusão além de detectar pode também prevenir a intrusão, pois os pacotes do ataque não conseguem chegar até a rede ou servidor.

Um *Intrusion Protection System* (IPS) seria uma forma mais inteligente que um *firewall*, já que ele bloqueia os pacotes que contém riscos e deixa o tráfego normal passar enquanto um *firewall* bloqueia somente a comunicação de um endereço IP ou uma porta.

Uma desvantagem do IPS é o fato de, se a sua configuração não for feita adequadamente, pode gerar muitos falsos positivos e bloquear tráfego normal.

4.3 MEIOS DE IMPLEMENTAR IDS

Antes de configurar um IDS, é necessário um estudo sobre qual implementação é mais eficiente para o sistema. Caso a implementação seja feita de uma forma errônea o IDS poderá gerar muitos falsos positivos ou falsos negativos dependendo do modo que foi configurado.

4.3.1 IDS de Rede – *Network Intrusion Detection System* (NIDS)

Possui a função de monitorar a rede como já diz o nome. As placas de rede, em uma grande parte dos computadores, trabalham em modo não promíscuo, ou seja, recebem apenas os pacotes que tem como destino seu endereço de acesso à mídia (MAC) e os outros pacotes são descartados. O NIDS exige que as placas de rede sejam configuradas em modo promíscuo, justamente para que todos os pacotes, incluindo os que não são endereçados ao MAC, passem por ele. Com isso o NIDS consegue detectar um possível ataque. Trabalhar com NIDS traz vantagens e algumas desvantagens para a rede.

Suas principais vantagens são: o fato de não interferirem no desempenho da rede, monitoram a rede inteira, possuem uma grande eficiência contra vários ataques e são praticamente imperceptíveis aos atacantes.

Algumas das suas desvantagens: não conseguem analisar um tráfego com informações criptografadas e se o tráfego for muito alto podem não conseguir processar todos os pacotes.

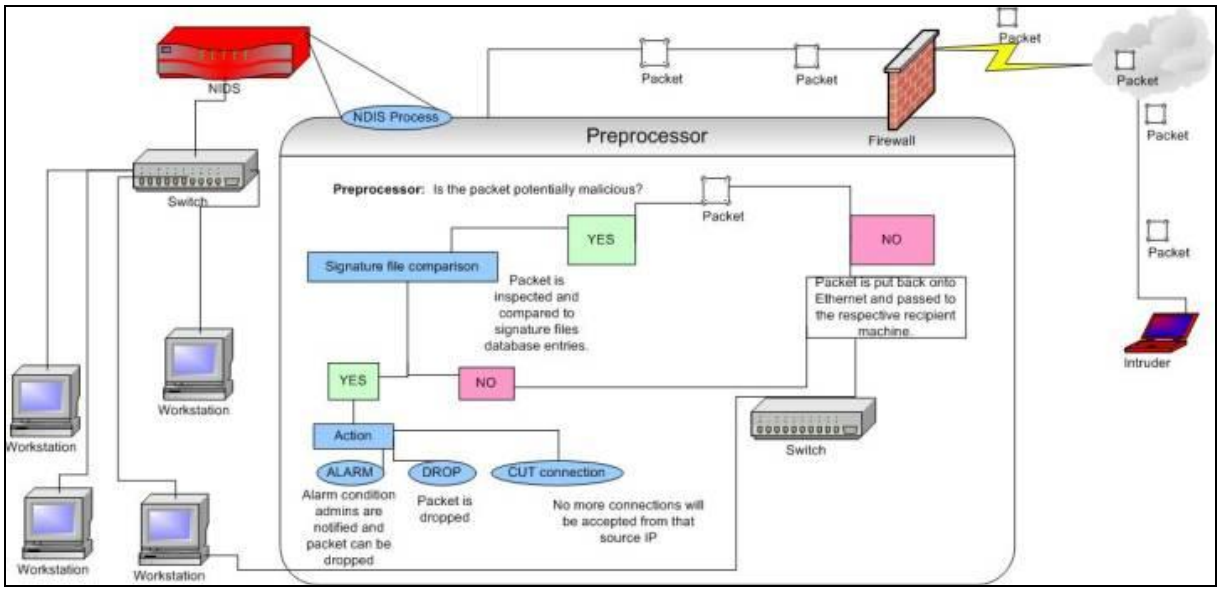

Figura 3. Rede NIDS
Fonte: FAULKNER, M. (2009)

4.3.2 IDS de *Host – Host Intrusion Detection System* (HIDS)

Sua placa de rede trabalha em modo não-promíscuo e o HIDS protege somente os *host*s onde está configurado. Esse *host* que ele está configurado tem todas as suas atividades analisadas com precisão pelo HIDS. Trabalhar com esta configuração também trás outras vantagens:

a) Nem todas as placas de rede conseguem operar no modo promíscuo como a NIDS pede;
b) Não há necessidade de verificar o tráfego de todo segmento de rede, podendo assim ser personalizadas regras para uma necessidade específica, descartando regras desnecessárias, aumentando o desempenho do processador e diminuindo a chance de sobrecarga;
c) Tem a capacidade de detectar ataques que não seriam detectados com NIDS;
d) Podem trabalhar em um ambiente criptografado, pois a informação é analisada antes de ser criptografada.

A sua maior desvantagem é possuir suas informações utilizadas para análise configuradas no próprio *host*, sendo que se um atacante invadir o sistema pode desabilitar esta funcionalidade.

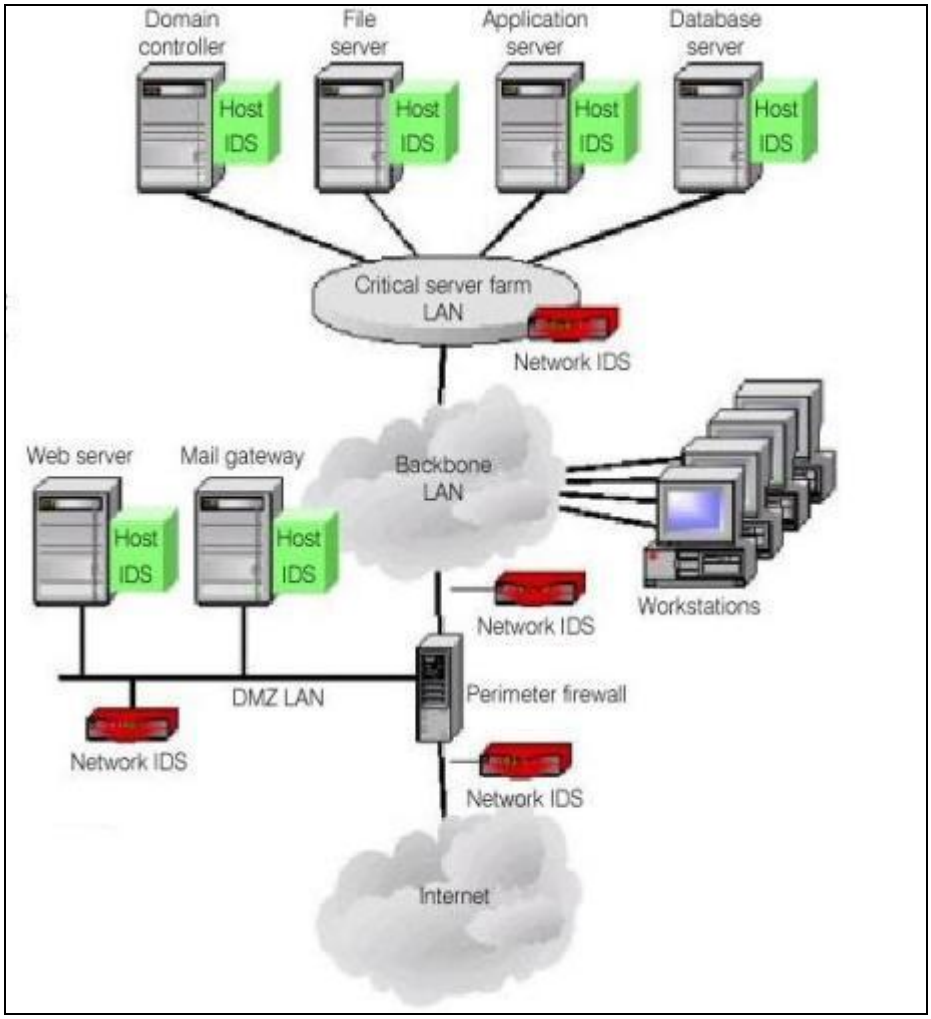

Figura 4. Rede HIDS
Fonte: SOARES, M. (2002)

4.3.3 IDS Distribuído – *Distributed Intrusion Detection System* (DIDS)

Um Sistema de Intrusão de Detecção Distribuído funciona com um tipo de arquitetura onde os sensores de ataques ficam em locais distantes e reportam acontecimentos a uma central de gerenciamento. Periodicamente ocorre o upload dos logs dos sensores para a central de gerenciamento e são armazenados geralmente em bancos de dados.

Para se obter uma melhora na segurança é recomendado o uso de tecnologia VPN, fazendo com que a troca de mensagens entre o gerenciador e os sensores seja por meio de uma rede privada, dificultando assim uma possível invasão e roubo de informações.

Um DIDS é um híbrido entre HIDS e NIDS já que pode ter sensores de ambos e pela grande quantidade de equipamentos e vários tipos de configurações possíveis, ele torna-se muito complexo na instalação e gerenciamento, exigindo um planejamento muito detalhado para que seja utilizado de maneira produtiva.

4.4 ESTRUTURA DO IDS

A estrutura de um IDS, geralmente, possui três tipos de mecanismo, que seriam: mecanismo de captura, de análise da informação e de respostas a ataques.

4.4.1 Fontes para Capturar Dados

O componente do IDS onde são capturadas informações é a parte analisada com o intuito de verificar se há risco de um possível ataque. Os principais modelos são NIDS, HIDS e DIDS, onde seus respectivos locais de coletas de dados são: tráfego da rede, comportamento de computadores e ambos.

Nos sistemas de detecção baseados em *host*s algumas das principais fontes de informação são registros de auditoria e logs do sistema.

As informações destes registros, geralmente são obtidas por ferramentas provenientes do próprio sistema operacional e, em alguns casos, ferramentas externas. As informações que são consideradas confiáveis pelo sistema são descartadas enquanto as outras têm seus registros armazenados para futuras análises.

Os logs são arquivos que possuem informações referentes ao histórico do sistema. São importantes para análise de segurança, já que detectam registram desde um login até o horário que um serviço deixou de funcionar.

A fonte mais comum de obtenção de informações é a captura de tráfego da rede feita por um NIDS, onde a rede é configurada em modo promíscuo e o tráfego é analisado por um mecanismo de análise de dados.

4.4.2 Mecanismo de Análise de Dados

Essa é a fase onde são organizadas e analisadas as informações coletadas sobre atividades no sistema e com isso buscar características que possam detectar uma tentativa de ataque.

Tarefas como pré-processamento, classificação e pós-processamento são funções do analisador. Durante o pré-processamento as informações são organizadas de forma adequada com a estratégia de análise

4.4.2.1 Detecção por Assinatura

Este tipo de detecção compara os dados de entrada do sistema com assinaturas já conhecidas para reconhecer um ataque. É considerada a técnica mais eficaz contra ataques conhecido, mas essa técnica é ineficiente em identificar novos tipos de intrusão ou intrusões que nunca foram registradas.

Existem alguns modelos como é feita a detecção por assinatura. Alguns deles são:

a) production/expert System: é feita na forma if/then com as condições sendo especificadas em if e se as condições forem acionadas, a parte then é

executada. Este tipo de modelo é uma dos primeiros que utiliza o sistema de detecção por assinatura.

b) state transition: utiliza diagramas de transição de estados de alto nível para a detecção de intrusos. Quando um evento acontece, o modelo verifica os diagramas para ver se há uma mudança de estado.

4.4.2.2 Detecção por Anomalias

Este tipo de detecção precisa construir um perfil que representa como seria o comportamento normal dos usuários, do sistema e da rede, tendo como base dados coletados durante um período de funcionamento normal e sem invasões. Esses dados são utilizados na criação de um conjunto de métricas.

Depois deste momento, todas as atividades que ocorrerem no sistema serão comparadas com este perfil, a fim de determinar se tais atividades são algo além do comportamento normal, podendo assim detectar um possível ataque. Seguem a baixo as técnicas mais utilizadas no processo de detecção por anomalias:

a) Análise quantitativa: a técnica mais utilizada em detecção por anomalias. Possui regras expressas em formato de cálculos variados. Informações coletadas neste modelo podem ser usadas como uma base para criação de um modelo de detecção por assinatura.

b) Medidas estatísticas: utiliza métricas para a criação de um perfil comum sendo atualizados periodicamente com o objetivo de mostrar uma possível mudança de comportamento do usuário. Uma grande vantagem deste modelo é a capacidade de detecção de ataques desconhecidos.

c) Modelo baseado em regras: parecido com o modelo de medidas estatísticas, com a diferença que os eventos que são comparados com as regras existentes e esses eventos são armazenados em outro conjunto de regras que contém um histórico do comportamento e atividades no sistema.

Para ser usado um sistema de detecção baseado em anomalias é necessário um complexo treinamento do sistema e ainda assim há um grande número de falsos positivos e negativos, restringindo o seu uso.

4.4.2.3 Modelo de Detecção Alternativo

Modelos alternativos são aqueles que utilizam esquemas híbridos, ou seja, técnicas de mais de um modelo para uma melhora da detecção de intrusões, não se encaixando em outra definição. Alguns desses modelos são:

a) *data mining*: esta técnica constrói modelos de intrusão utilizando mineração de dados, ou seja, procurando por padrões de características do sistema que possam ser úteis para descrever o comportamento de usuários e aplicativos.
b) baseado em sistema imunológico: técnica que vê semelhanças entre sistemas de detecção de intrusão com um sistema imunológico humano. A principal função é a de diferenciar tráfego comum de tráfego suspeito através de análises de dados.
c) algoritmos genéticos: baseados em conceitos de Darwin sobre a evolução. Esta técnica cria vetores de hipótese para os dados de evento que indicam se há intrusão ou não. Essa hipótese passa por testes para ver se é valida. Uma hipótese melhorada (evoluída) é derivada da anterior e passa por testes novamente e este processo é repetido até ser encontrada uma solução.

4.4.3 Mecanismo de Respostas

Esse mecanismo tem como objetivo fazer com que o sistema responda após a detecção de um ataque. Os tipos de respostas são classificados como respostas passivas e respostas ativas.

Na resposta passiva, o mecanismo envia avisos de intrusão e armazenam informações sobre ataques. Na ativa o mecanismo tenta bloquear o ataque de alguns modos, sendo organizados por:

a) resposta contra atacante: causa danos ao próprio atacante sendo mais aconselhável bloquear o endereço do atacante, impedindo a continuação do ataque;
b) correção do sistema: corrigindo as vulnerabilidades do sistema pode evitar ataques futuros, podendo imitar o sistema imunológico humano, como já falado anteriormente, reconhecendo a ameaça, isolando o atacante e respondendo ao ataque achando a melhor alternativa;
c) coletar informação: tem como objetivo atrair ataques para estudá-los a fim de desenvolver assinaturas novas.

4.5 VULNERABILIDADE DO IDS

Um sistema de detecção de intrusão, como todo software, possui algumas vulnerabilidades. Agentes podem utilizar estas vulnerabilidades para invadir o sistema, sendo assim, as ferramentas IDS devem corrigi-las e minimizá-las antes que seja colocadas em funcionamento.

4.5.1 Falsos Positivos

Um falso positivo é aquele em que o IDS detecta um evento que na realidade não é um ataque. Geralmente isso ocorre devido à uma configuração inadequada do IDS, havendo regras criadas erroneamente ou excessos de regras. Não chega a ser um problema grave isoladamente, porém falsos positivos em grande número podem complicar o funcionamento do IDS, atrapalhando na análise dos resultados.

4.5.2 Falsos Negativos

Um falso negativo é mais grave que um falso positivo, pois neste caso é um ataque que passa despercebido pelo IDS, tendo sucesso na invasão e causando mais problemas ao sistema. Segundo Chroters (2003) um falso negativo pode ser um ataque desconhecido, um sobrecarga no sistema, um erro de configuração ou uma evasão.

4.5.3 Evasão

Segundo Crothers (2000 apud Steffen, 2003) uma evasão é somente um modo de enganar o IDS, fazendo com que um ataque consiga passar despercebido por ele.

A técnica de evasão possui dois objetivos:

- evitar que o ataque seja detectado;

- tentar sobrecarregar o sensor do IDS, fazendo com que a análise das informações seja comprometida, permitindo que um possível ataque seja realizado simultaneamente.

A Figura 5 mostra um exemplo de como a evasão ocorre. Os dados do pacote são modificados para que consigam driblar a ferramenta IDS, não coincidindo com alguma possível assinatura existente na base.

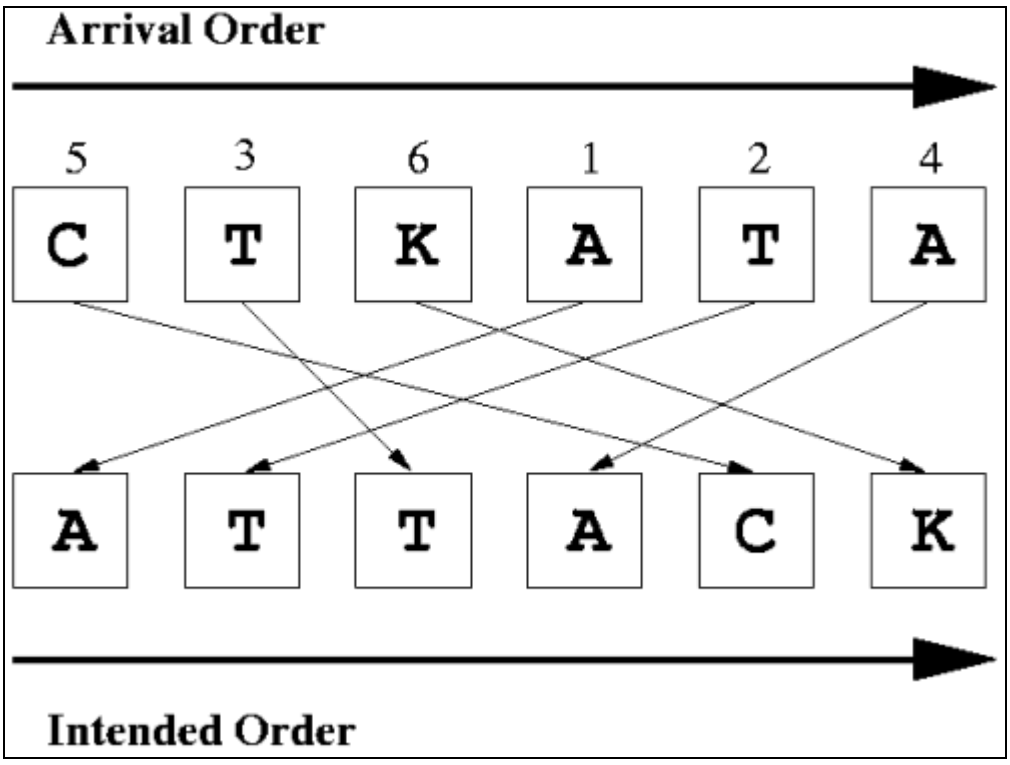

Figura 5. Evasão de IDS
Fonte: Insecure.org (2005)

4.6 PROBLEMAS EM IMPLANTAR O IDS

Pode haver alguns problemas durante a implantação do IDS, que devem ser solucionados para que a ferramenta funcione normalmente.

As ferramentas também possuem certas limitações, sendo que estas podem ser exploradas, fazendo com que a ferramenta fique fragilizada e sua capacidade de interpretar o tráfego corretamente seja comprometida (ALLEN, 2000).

4.6.1 Switches

Um switch direciona as informações somente para o micro de destino, se diferenciando do hub, exigindo assim uma atenção especial na implantação do IDS.

Em switches gerenciáveis esse problema tem como ser contornado, configurando uma porta para onde é direcionado todo o tráfego de rede. Em IDS que não são gerenciáveis esse recurso não é possível.

Outra estratégia contra este problema seria adquirir um switch que já possui IDS embutidos, porém seu preço é maior que um switch comum.

4.6.2 Redes de Alta Velocidade

Conforme a velocidade do tráfego aumenta devido ao crescimento da tecnologia, às vezes o IDS não consegue analisar todos os pacotes, descartando assim algumas informações que podem ser importantes para a descoberta de um ataque.

Existem alguns métodos de amenizar este problema. Uma delas seria o balanceamento do tráfego, que consiste na instalação de sensores inteligentes que analisam segmentos de rede, diminuindo assim a quantidade de informações que passará por cada sensor e balanceando o tráfego.

4.6.3 Redes Criptografadas

Quando os dados de uma rede são preciosos, precisam ser tratados com cautela ao serem transmitidos pela rede. Para isso são utilizadas ferramentas de criptografia para que as informações contidas nestes dados fiquem em sigilo. Essa criptografia acaba atrapalhando na análise de dados do IDS.

Uma alternativa para este problema é a instalação de um HIDS onde o tráfego é criptografado, pois ele analisa antes da informação ser criptografada e depois de ser descriptografada.

5 FERRAMENTAS IDS

Nesse capítulo serão abordadas as ferramentas IDS utilizadas nos testes. As ferramentas escolhidas são duas ferramentas IDS de rede (NIDS): a ferramenta Snort, por ser a ferramenta *open source* mais conhecida e utilizada atualmente, e a ferramenta *RealSecure*, que é um programa comercial distribuído pela IBM.

5.1 SNORT

Snort é um software IDS *open source*, capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP. Dentre suas funções se encontram a execução de análises de protocolo e associação de padrões de conteúdo (assinaturas), sendo utilizado para detectar uma variedade de ataques. Esta ferramenta é suportada em plataformas das mais diversas. Alguns pontos fortes que o Snort possui além da variedade na compatibilidade com sistemas são: o fato de ser leve, ser pequeno e possuir o maior cadastro de assinaturas, o que aperfeiçoa a verificação de anomalias que ocorrem dentro da rede.

Essa ferramenta está em constante desenvolvimento e atualização. Essas atualizações são feitas diariamente, tanto em código como nas regras de detecção.

Por ser uma ferramenta leve, sua utilização é indicada para monitorar redes TCP/IP pequenas, onde não ocorre um tráfego muito grande, onde tem uma eficácia maior, podendo detectar uma variedade maior de tráfego suspeito.

Outro ponto positivo do Snort é o grande número de possibilidades de tratamento dos alertas gerados.

5.1.1 Funcionamento do Snort

O Snort habilita a placa de rede do computador onde foi configurado para modo promíscuo, permitindo assim que todos os pacotes que trafegam pelo segmento de rede daquela máquina sejam capturados e comparando esses pacotes com as assinaturas existentes é possível detectar um ataque e enviar alertas em tempo real.

A arquitetura do Snort prioriza o desempenho, a simplicidade e a flexibilidade e possui três subsistemas básicos que o compõem: farejador de pacotes, mecanismo de detecção, subsistema de alerta e registro.

5.1.2 Farejador de Pacotes

Farejadores de pacotes são dispositivos utilizados para monitorar a rede, podendo transformar os pacotes que trafegam em dados legíveis para seres humanos.

O Snort utiliza uma biblioteca chamada *libpcap* para capturar pacotes de toda a rede. Após capturar os pacotes da placa de rede, ela encaminha os pacotes para os mecanismos de decodificação do Snort, para que possam ser analisados e comparados posteriormente por pré-processadores, que são plug-ins responsáveis por classificar o comportamento dos dados capturados pelo farejador. Após ter o seu comportamento classificado, o pacote é mandado para o mecanismo de detecção.

A biblioteca *libpcap* foi criada inicialmente para sistemas UNIX, porém, atualmente há a sua versão para *Windows*, chamada de *winpcap*.

5.1.3 Mecanismo de Detecção

É a parte onde os dados são comparados com as assinaturas existentes. Caso os dados do pacote sejam semelhantes a uma assinatura conhecida pelo IDS, é enviado um sinal de alerta ao processador.

5.1.4 Subsistema de Alerta e Registro

Caso o IDS dispare um alerta, o mesmo pode ser enviado a um log por uma conexão de rede e também tendo a possibilidade de ser armazenado em um banco de dados.

O Snort também possui a opção de trabalhar juntamente com a ferramenta Aanval, que tem como objetivo gerenciar os alertas gerados, trabalhando em tempo real e fornecendo relatórios sobre o IDS.

5.2 *REALSECURE*

De acordo com o site da IBM ISS, em 1992, Christopher Klaus desenvolveu a primeira versão do *Internet Scanner*, uma tecnologia para proteção que identificasse e corrigisse pontos fracos na segurança da rede. Em 1994, Klaus, em conjunto com Thomas Noonan, fundaram a *Internet Security System* (ISS) com o objetivo de desenvolver um pouco mais o *Internet Scanner* e comercializá-lo. Em 2006 a IBM efetuou a compra da ISS e passou a controlar a venda dos softwares.

5.2.1 Arquitetura do *RealSecure*

A ISS possui vários softwares que tem como objetivos: ajudar no gerenciamento, analisar dados em tempo real e controlar a segurança de redes. Entre esses softwares, se destaca *RealSecure Protection System*. As duas ferramentas utilizadas dessa linha de produtos nesse trabalho são: *RealSecure Network Protection e RealSecure SiteProtector*.

5.2.2 *RealSecure Network Protection*

O *RealSecure Network Protection* é uma ferramenta cujos componentes são sensores de proteção para redes com a capacidade de capturar o tráfego de rede e analisá-lo com a finalidade de detectar ataques. Esse sensor que monitora o tráfego é o *Network Sensor*. Para que o *Network Sensor* consiga analisar todo o tráfego da rede, a placa de rede deve ser configurada em modo promíscuo.

Para que a ferramenta *Network Sensor* conseguisse detectar pacotes suspeitos, ela precisa de políticas de segurança. Estas políticas se encontrar no componente chamado *Network Sensor Policies*. Nele estão contidas diversas políticas pré-definidas e assinaturas de ataques para que o *Network Sensor* consiga realizar uma análise confiável e determinar o que podem ser pacotes com conteúdo malicioso.

Caso algum tipo de anomalia venha ser detectado pelo *Network Sensor*, é enviado um alerta com dados desta. Esses dados podem ser vistos no sistema *RealSecure SiteProtector*.

5.2.3 *RealSecure SiteProtector*

O *RealSecure SiteProtector* é um sistema que gerencia e gera relatórios de segurança para aplicações *RealSecure* de um modo centralizado e escalonável. O sistema de controle e monitoração do *SiteProtector* permite a configuração das ferramentas *RealSecure* de um modo automático para combaterem novas ameaças juntamente com o *firewall*, utilizando as informações das avaliações de vulnerabilidades.

5.2.3.1 Componentes do *SiteProtector*

O *SiteProtector* possui componentes obrigatórios para sua funcionalidade e alguns opcionais. Os componentes essenciais da versão recente são: *Agent Manager*, *Console*, *Database* e *Core*.

O *Agent Manager* gerencia e controla as atividades realizadas pelo *Desktop Protection* e pelo *Express Update Server*, facilitando a transferência de dados dos sensores pro *Event Collector.*

O *Console* é a principal interface e o principal componente do *SiteProtector*. É onde o usuário pode executar as funções do software, dentre elas se encontram monitoramento de eventos, *scan*, configuração de políticas e agentes, geração de relatórios, gerenciamento e configuração vários sensores.

SiteProtector Database é a base de dados do *SiteProtector*. Basicamente é onde são armazenados os dados do agente.

Core é o núcleo do *SiteProtector* e dentre suas funções se encontram: gerenciamento da comunicação entre *Console* e *Database*, controle da funcionalidade dos sensores, proporcionar facilidade nos *updates* e um fácil acesso ao *SiteProtector*.

O usuário do *SiteProtector* também pode editar e criar políticas, fazendo com que ele encontre uma configuração que se adéque ao seu gosto.

O *SiteProtector* não possui um banco de dados próprio, porém a ferramenta traz na sua instalação o MSDE e as configurações do banco já são feitas durante instalação da ferramenta.

6 TRABALHOS CORRELATOS

Este capítulo relaciona alguns trabalhos com o teor semelhante a esta fundamentação teórica.

6.1 SEGURANÇA EXPOSTA EM REDE DE COMPUTADORES

Trabalho feito por Luiz Alexandre Rodrigues Vieira, graduando em Tecnologia em Redes e Ambientes Operacionais pela Unibratec de Recife, PE.

Esse trabalho apresenta os problemas relacionados com os descuidos de softwares não atualizados aos administradores e usuários de rede quanto à necessidade de correções de falha de segurança conhecidas| nos Sistemas Operacionais, e programas que possuam novas versões que corrigem vulnerabilidades contidas em versões anteriores. É demonstrado que aqueles que acreditam que estão seguros por ter instalado sistema de segurança, como antivírus e firewalls atualizados possuem um pensamento equivocado.

6.2 SEGURANÇA DE REDES: SISTEMA DE DETECÇÃO DE INTRUSÃO

Monografia feita por Evelyn Ruth Kler e Gelson Prado do curso de Administração, tendo ênfase em análise de sistemas, pela Faculdade Internacional de Curitiba no ano de 2004.

O trabalho fala sobre os diversos tipos de ataques em rede que as empresas estão suscetíveis atualmente, abordando sistemas de detecção de intrusão como um importante auxiliar na melhoria da segurança em redes, citando um dos mais utilizados hoje em dia, o Snort.

6.3 PRIMESEC: SISTEMA DE DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES SOBRE UMA PLATAFORMA MULTIPROCESSADA

Trabalho de conclusão de curso feito por Guilherme Montez Guindani e Hugo Artur Weber Schmitt, orientados por Fernando Gehm Moraes, Porto Alegre, 2006.

Este trabalho foi feito com a necessidade de assegurar o funcionamento correto de sistemas de segurança, evitando intrusões por meio de co-processadores de uso especifico, onde unidades especializadas de processamento paralelo integram-se aos sistemas tradicionais.

7 ANÁLISE E COMPARAÇÃO ENTRE AS FERRAMENTAS IDS

Este trabalho consistiu em descrever o funcionamento das ferramentas IDS e conceitos sobre segurança de redes, desde os mais básicos até as dificuldades de instalação de ferramentas de IDS. Após a fundamentação teórica, é mostrado um aprofundamento sobre o funcionamento das ferramentas, as metodologias utilizadas para comparações e a realização de testes entre elas. Os resultados dos testes podem ser visualizados nesse capitulo.

7.1 METODOLOGIA

Para os testes entre as ferramentas IDS, foi construído um ambiente isolado, onde as ferramentas serão comparadas com o intuito de fornecer um melhor entendimento sobre as capturas de ataques e seus falsos positivos gerados.

O objetivo principal dos testes foi o de visualizar o comportamento de cada ferramenta durante o funcionamento, utilizando configurações consideradas padrões. Estes foram feitos individualmente e padronizados para que as ferramentas IDS passassem pelos mesmos ataques. Após o termino foi feita a comparação entre as ferramentas, cujo objetivo é demonstrar o desempenho de cada uma.

Para que fossem considerados eficientes perante a comunidade científica, foi necessária uma metodologia que seja aceita pela mesma.

7.1.1 Metodologias de comparação entre sistemas IDS

Metodologias de comparação entre IDS são importantes para possibilitar a análise entre as ferramentas, conseguindo assim, mesmo com variações de tráfego e de algumas características de rede, determinar qual das ferramentas se adapta melhor ao ambiente.

Nesse capítulo serão resumidas algumas metodologias desenvolvidas e seguidas atualmente.

7.1.1.1 Alessandri

Desenvolvida pela IBM em 2000, tendo como objetivo principal fazer testes entre as capacidades do IDS, não levando em conta bases de assinatura e implementacões dos sistemas, analisando como o IDS se comporta contra um ataque novo, cuja assinatura ainda não exista.

Essa metodologia é formada por uma técnica implementada em *prolog*, onde as características dos IDS e as características em ataques variados são descritas em forma de regras. Com o cruzamento de regras é possível identificar o comportamento dos IDS sem a necessidade de realizar testes entre os sistemas.

A descrição de características em forma de regras é feita em duas maneiras. Uma separa as propriedades do IDS entre genéricas e detalhadas de acordo com o nível de detalhe de cada característica. A outra separa as propriedades do IDS de acordo com funções que as características exercem, como por exemplo, técnicas usadas no reconhecimento de padrões e demora de resposta a ataques.

7.1.1.2 Lippmann

Essa metodologia foi desenvolvida em 1998 e 1999, consistindo em classificar os ataques tendo como espelho tais aspectos: tipos de ações feitas pelo usuário, nível de privilégio do usuário e métodos de transição.

Os testes de 98 e 99 tiveram como objetivo identificar ataques detectados e listar os falsos positivos que o IDS testado gerou.

7.1.1.3 Puketza

A metodologia de Puketza foi a primeira metodologia criada para avaliação de IDS e foi desenvolvida na Universidade da Califórnia. Primeiramente devem-se selecionar os cenários de teste, que podem ser reproduzidos através de scripts, simulando assim, ataques e atividades normais do cotidiano. Depois de selecionados os cenários, é possível o desenvolvimento de scripts que podem simular intrusões variadas.

Os testes foram divididos em três categorias: identificação da intrusão, utilização de recursos e testes de saturação.

Primeiramente são realizados testes para verificar o comportamento do IDS ao detectar ataques concorrentes e ataques seqüenciais. Após a identificação da intrusão são feitos testes para avaliar os recursos computacionais que são utilizados pelo IDS e por último são feitos testes em situações extremas, analisando o comportamento do IDS em situações de “stress”.

7.1.2 Comparação entre as metodologias

Segundo Corrêa (2005) as metodologias de Puketza e Lippmann avaliam principalmente a base de assinaturas do sistema, fazendo com que gere um resultado considerado válido por um curto período de tempo, pois assim que o IDS for atualizado, os testes realizados se tornam, em parte, obsoletos. Já a metodologia de Alessandri testa a capacidade de detecção do IDS não necessitando das assinaturas existentes, o que pode não refletir a real capacidade da ferramenta.

De acordo com Corrêa (2005) a metodologia de Lippmann não descreve como é composto o seu tráfego de fundo, fazendo com que os resulados de falsos positivos sejam possivelmente contestados. Assim como Lippmann, Puketza também apresenta um ambiente de testes muito complexo, podendo assim, inviabilizar a reprodução de testes.

Segundo Corrêa (2005), de modo geral, as metodologias de Lippmann, Puketza e Alessandri não possuem uma proposta voltada para o uso dos IDS nas empresas.

O Quadro 1 mostra um comparativo entre as metodologias citadas anteriormente.

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

Quadro 1. Comparação de metodologias existentes
Fonte: FAGUNDES, L (2002)

Observando o Quadro 1, foi verificado que nenhuma metodologia engloba todas as características metodológicas, portanto foi feita uma síntese demonstrando quais as características utilizadas nesse trabalho, sendo assim, a metodologia utilizada no trabalho é um mesclado de metodologias existentes, como é demonstrado no Quadro 2.

Tipo de avaliação	
Avaliação exaustiva	
Potencialidade de detecção	X
Classificação de ataques	
Descrição técnica do ataque	
Descrição do contexto do ataque	X
Tráfego de fundo	
Tráfego de fundo sintético	X
Tráfego fragmentado	
Tráfego homogêneo	
Ambiente de teste	
Independente	X
Específico	
Resultados apresentados	
Taxa de falsos positivos	X
Tipos de ataques detectados	X

Quadro 2. Metodologia do trabalho

Segundo Fagundes (2002) constatou-se que a maior parte das metodologias apenas avalia a base de assinaturas dos IDS, exceto na metodologia de Alessandri, o que pode ser considerado exaustivo e o resultado gerado só é válido por um pequeno período de tempo, já que assinaturas são desenvolvidas muito rapidamente por seus fabricantes. Por outro lado, a metodologia de Alessandri testa a capacidade de detecção dos IDS e seu experimento só precisará ser refeito quando novos recursos forem implementados às ferramentas.

No que tange à classificação de ataques, a metodologia utilizada nesse projeto usa a descrição do contexto do ataque, a mesma utilizada por Lippmann, pois a descrição técnica do ataque poderia se tornar muito complexa e atrapalhar nos testes.

O tráfego de fundo é um fato que pode interferir diretamente no resultado de alguns testes de acordo com Fagundes (2002). O tráfego de fundo utilizado nos testes é o tráfego de fundo sintético, gerado pela ferramenta *Network Traffic Generator & Monitor* e com ela é gerado um tráfego de fundo para tentar atrapalhar a detecção do ataque com a

finalidade de comprovar que o tráfego de fundo pode dificultar a funcionalidade da ferramenta IDS.

Segundo Fagundes (2002) com exceção da metodologia de Alessandri, todas elas definem um ambiente de testes para realização dos mesmos. Porém as metodologias de Lippmann e Puketza requerem ambientes de teste considerados complexos, com várias estações e diferentes equipamentos de interconectividade, e com isso podem inviabilizar a reprodução dos testes. Por esses motivos, o ambiente de teste utilizado é um modelo simples e é apresentado posteriormente.

Os resultados apresentados no final dos resultados serão os falsos positivos gerados e os tipos ataques detectados, pois ambos são importantes para a análise e comparação das ferramentas.

7.2 ORGANIZAÇÃO DOS TESTES

Durante os testes também foi seguida a norma *Open Source Security Testing Methodology Manual* (OSSTMM), que consiste em várias técnicas sobre testes de segurança. Esta norma, se usada em conjunto com a metodologia escolhida, torna possível um teste completo para a localização de falhas no sistema de segurança da rede e comparações entre os sistemas.

Os testes foram divididos nas seguintes categorias:

a) reconhecimento dos ataques;

b) desempenho;

c) técnicas de evasão.

7.2.1 Reconhecimento dos Ataques

Os ataques que foram gerados contra as ferramentas IDS foram:

7.2.1.1 Exploits e Buffer Overflow

Exploits são programas criados em qualquer linguagem de programação que exploram vulnerabilidades em servidores. Eles executam comandos arbitrários no servidor podendo assim dar acesso root (administrador) àquele que o executa. Muitos deles exploram o Buffer Overflow.

O Buffer Overflow ocorre quando um programa recebe uma quantidade de dados muito maior do que está preparado para armazenar em buffer. Nessa quantidade de dados pode possuir algum código com conteúdo mal-intencionado que será executado pelo sistema. Esse tipo de ataque é difícil de ser descoberto e um dos mais utilizados.

7.2.1.2 Denial of Service (DoS)

É um tipo de ataque que faz o sistema receber várias requisições ao mesmo tempo, fazendo com que ele sobrecarregue e fique sem condições de uso.

7.2.1.3 Port Scan

Ataque que mapeia as portas TCP e UDP a procura de informações referentes a serviços acessíveis. Sendo assim o *hacker* ataca diretamente agindo em cima de serviços ativos.

7.2.1.4 HTTP e FTP

Ataques realizados explorando as vulnerabilidades dos protocolos HTTP e FTP.

7.2.1.5 Ferramentas Utilizadas nos Ataques

Ferramentas de *Exploits* e *Buffer overflow*:

a) Nessus;

b) Metasploit Framework.

Ferramentas de *Denial of Service* (DoS):

a) Nessus;

b) Hpring2.

Ferramentas de *port scan*:

a) Nessus;

b) Nmap;

c) LANguard.

Ferramentas de HTTP e FTP:

a) Nessus;

7.2.1.6 Descrição das Ferramentas

Nessus: ferramenta *open source* muito utilizada para análise de vulnerabilidades da rede. Tem como objetivo principal a verificação de falhas e vulnerabilidades de segurança do sistema. É composto por uma arquitetura cliente/servidor onde, o servidor faz um *port scan* no computador alvo e posteriormente vários scripts são ligados a cada porta aberta com intuito de verificar problemas na segurança.

Nmap: é um software livre que realiza *port scan*. Muito utilizado para avaliar a segurança de redes.

LANguard: software comercial que realiza *port scan*. Possui versões tanto para Linux quanto para Windows.

Hping2: ferramenta utilizada para ataques do tipo *Denial of Service* (*DoS*).

Metasploit Framework: ferramenta *open source* utilizada principalmente para ataques do tipo *Exploit*. Possui versão para Linux e Windows.

7.2.2 Desempenho

Foram feitos testes para a análise do desempenho da ferramenta IDS em detectar ocorrências com e sem tráfego de fundo, sendo realizados os mesmos ataques nas duas condições. Para gerar tráfego de fundo foi utilizada uma ferramenta chamada *Network Traffic Generator & Monitor*.

7.2.3 Testes de Evasão

Para esse tipo de teste foram utilizadas as ferramentas *Fragrouter* e *Whisker*.

Whisker é um *scanner* de vulnerabilidades. Essa ferramenta é implementada com técnicas anti-IDS (evasão e inserção), que dificultam a detecção dos ataques.

A ferramenta *Fragrouter* gera uma fragmentação de dados que tendem a passar pelo roteador com o objetivo de dificultar a detecção de um ataque que aconteça pelo sensor do IDS.

7.3 REALIZAÇÃO DOS TESTES

Todos os testes foram efetuados em condições idênticas com cada ferramenta e depois de testadas foi feito um comparativo entre as ferramentas com os resultados obtidos.

As duas ferramentas IDS foram instaladas em dois computadores diferentes e sujeitas aos mesmos ataques duas vezes, uma sem tráfego de fundo e a outra com tráfego de fundo gerado pela ferramenta *Network Traffic Generator & Monitor*, simulando um tráfego real durante os ataques.

Os testes que da categoria de reconhecimento de ataques foram realizados na ordem a seguir:

a) ataques de *Exploits e Buffer overflow*;

b) ataques *Denial of Service* (DoS);

c) ataques de *port scan*;

d) ataques HTTP;

e) ataques FTP.

As características analisadas durante os testes foram:

a) capacidade de detecção de ataques;

b) capacidade de evasão;

c) quantidade de falsos positivos gerados por cada ferramenta.

7.3.1 Ambiente de Testes

O ambiente de testes mostrado pela figura abaixo foi montado para a realização dos testes.

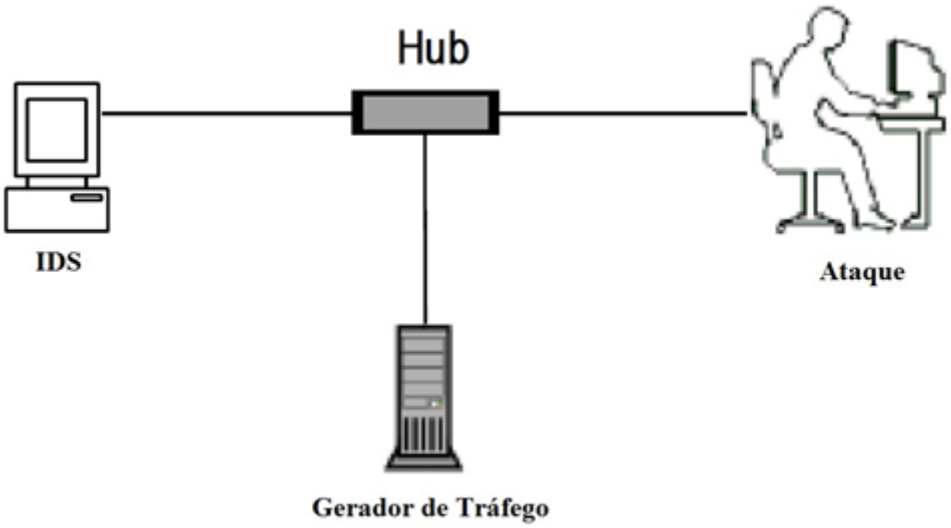

Figura 6. Ambiente de testes

A Figura 6 representa como foi montado o ambiente de testes utilizado e seus componentes. Este ambiente é formado por:

a) gerador de tráfego de fundo: fornecido por uma máquina com a ferramenta *Network Traffic Generator & Monitor* instalada. O tráfego de fundo também é fornecido pelo tráfego comum provido pela Internet;

b) ferramentas IDS: as ferramentas IDS foram instaladas em duas máquinas diferentes, uma ferramenta em cada máquina, com configurações semelhantes. Os alertas gerados são lidos nas próprias máquinas.

c) atacante: máquinas com todos os simuladores de ataque instalados.

Todas as máquinas são interligadas por um *hub*.

7.3.1.1 Softwares Utilizados

Segue a lista de todos os softwares utilizados ao decorrer do trabalho:

a) Snort 2.9.0.5;

b) Oinkmaster 2.0;

c) Avast 5.0.677;

d) Microsoft Security Essentials 1.105.1849.0;

e) WinPcap 4.1.2;

f) Kiwi Syslog Server Console 9.2;

g) Nessus 4;

h) Nmap 5.51;

i) Metasploit Framework 3.7.1;

j) *RealSecure SiteProtector* 2.0 *Service Pack* 8.1;

k) *RealSecure Network Protection* 7.0;

l) Wireshark 1.6.0;

m) Whisker 1.4;

n) Fragrouter 1.6;

o) LANguard 1.0;

p) *Network Traffic Generator & Monitor*;

q) Notepad++ 5.9.

7.4 RESULTADOS DOS TESTES

Para melhor visualização dos resultados obtidos, foram organizadas tabelas com o objetivo de deixar mais clara possível a compreensão das informações. Os testes foram replicados 4 vezes e os resultados apresentados nas tabelas são as médias obtidas. Os Apêndices A e B mostram como as ferramentas foram configuradas.

Foi utilizada a ferramenta *Wireshark* em conjunto aos IDS durantes os ataques onde o tráfego de fundo artificial era presente para verificar se todos os pacotes realmente passavam pelas ferramentas. Foi provado que as ferramentas IDS recebiam todos os pacotes, porém, não conseguiam analisar a todos. Em um teste rápido, sem simulações de ataque, a ferramenta Snort recebeu 658 pacotes da rede, a mesma quantidade que o *Wireshark* detectou, mas analisou 655. O teste foi replicado, em ambas as ferramentas IDS, sendo que os pacotes capturados pelo Snort e pelo Wireshark eram os mesmos, e o resultado foi a análise de quase todos os pacotes, sempre ficando poucos pacotes fora da análise da ferramenta, porém os pacotes analisados eram sempre superiores a 97% do total de pacotes recebidos.

Snort: reconhecimento de ataques e desempenho da ferramenta						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Exploits	7	5	71	7	2	28
Denial of Service	3	2	66	3	2	66
Port Scan	3	3	100	3	2	66
HTTP	6	4	66	6	3	50
FTP	4	2	50	4	2	50
Total	23	16	69	23	11	47

Tabela 1 - Reconhecimento de ataques e desempenho da ferramenta Snort

***RealSecure*: reconhecimento de ataques e desempenho da ferramenta**						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Exploits	7	3	42	7	2	28
Denial of Service	3	2	66	3	2	66
Port Scan	3	3	100	3	3	100
HTTP	6	4	66	6	4	66
FTP	4	4	100	4	3	75
Total	23	16	69	23	14	60

Tabela 2 – Reconhecimento de ataques e desempenho da ferramenta *RealSecure*

As Tabelas 1 e 2 mostram que nenhuma das ferramentas conseguiu detectar todos os ataques efetuados.

Sem a presença de tráfego de fundo artificial gerado pela ferramenta *Network Traffic Generator & Monitor*, as duas ferramentas conseguiram reconhecer a mesma quantidade de ataques, que foram 16 de 23, ou seja, 69% do total. Já quando a ferramenta geradora de tráfego de fundo estava ativa, o *RealSecure* conseguiu reconhecer 14 de 23 (60%), enquanto o Snort conseguiu 11 de 23 (47%). Nesse teste a ferramenta Snort se provou menos eficiente que a *RealSecure*, pois teve sua capacidade de detecção comprometida em situações onde o tráfego de fundo era alto.

Esses resultados podem ser atribuídos também a uma má configuração das ferramentas, pois como foi dito anteriormente, as ferramentas estão operando nas suas configurações padrão. É possível que alterações na configuração possam mudar os resultados dos testes, aumentando a eficiência da ferramenta em situações de stress.

Testes de evasão						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Whisker	1	1	100	1	1	100
Fragrouter	1	1	100	1	1	100

Tabela 3 - Testes de evasão

De acordo com a Tabela 3, ambas as ferramentas se provaram eficientes em testes de evasão, pois detectaram todas as tentativas de ataque.

As Tabelas 4 e 5 mostram os resultados dos falsos positivos gerados pelas ferramentas durante os ataques com e sem tráfego de fundo gerado.

Falsos Positivos Gerados – Snort		
Ataques	**Tráfego de Fundo**	
	Sem Tráfego	**Com Tráfego**
Exploits	0	14
Denial of Service	0	5
Port Scan	6	11
HTTP	10	23
FTP	9	14
Total	25	67

Tabela 4 - Falsos positivos gerados pela ferramenta Snort

Durante o monitoramento da rede pelo Snort, a ferramenta gerou 25 alertas que não possuem relação com os ataques efetuados nos testes sem presença de tráfego de fundo. Já nos testes com tráfego de fundo presente, os alertas falsos quase triplicaram, indo para 67 alertas, mostrando que a eficácia do Snort foi comprometida dependendo da quantidade de tráfego de fundo aplicada na rede.

Falsos Positivos Gerados – *RealSecure*		
Ataques	**Tráfego de Fundo**	
	Sem Tráfego	**Com Tráfego**
Exploits	1	52
Denial of Service	0	11
Port Scan	3	14
HTTP	14	26
FTP	0	51
Total	18	154

Tabela 5 - Falsos positivos gerados pela ferramenta *RealSecure*

A Tabela 5 mostra as ocorrências que a ferramenta *RealSecure* gerou. Sem o tráfego de fundo na rede, foram gerados 18 falsos positivos. Nos testes com a presença de tráfego de fundo, o número de falsos positivos gerados foi 154, quase 9 vezes o número de alertas gerados sem tráfego, mostrando que assim como o Snort, o *RealSecure* também pode ter sua análise comprometida quando há um tráfego de rede elevado.

Entre as duas ferramentas, o *RealSecure* se provou menos eficiente que o Snort, pelo fato de gerar um número muito superior de falsos positivos em relação ao número gerado pelo IDS *open source*.

7.4.1 Discussão sobre os Testes e Resultados Obtidos

Em relação aos testes sem a ocorrência do tráfego de fundo na rede, as duas ferramentas conseguiram detectar a mesma quantidade de ataques, porém quando houve tráfego de fundo, o *RealSecure* detectou 60% gerados enquanto o Snort conseguiu somente 47% do total, o que demonstra que o tráfego de fundo pode alterar muito na detecção dos

ataques. A ferramenta *RealSecure* apresentou melhor desempenho no reconhecimento de ataques em relação à ferramenta Snort.

A mesma situação aconteceu com os falsos positivos. Os falsos positivos gerados pelas duas ferramentas foram quase da mesma quantidade sem a presença de tráfego de fundo. Já com tráfego de fundo, ambas geraram um número alto de falsos positivos, demonstrando assim que o tráfego de fundo gerado influencia de modo significativo na detecção dos ataques. A ferramenta *RealSecure* gerou mais falsos positivos que a ferramenta Snort, aprensentando um pior desempenho em relação ao IDS *open source*.

Um grande número de falsos positivos pode indicar uma configuração inadequada nos sensores das ferramentas. Sendo assim, verifica-se que é importante que os sensores sejam configurados adequadamente para que sejam minimizados os falsos positivos gerados.

Apesar de alguns erros que possam ocorrer, as duas ferramentas IDS atingiram seu objetivo, detectando ataques e alertando o administrador sobre eles.

CONCLUSÃO

No decorrer deste trabalho constatou-se que ferramentas convencionais em si, já não são o suficiente para garantir a segurança de uma rede, seja ela pequena ou grande. Mesmo com a evolução de *firewalls* e antivírus, enquanto existir um usuário operando algum sistema computacional, esse sistema deixa de ser totalmente seguro.

Um dos melhores métodos de tratar esse problema é com uma política de segurança que se adeque à estrutura da empresa, pois nem sempre se pode contar somente com as ferramentas, trabalhando em conjunto com ferramentas auxiliares, como por exemplo, os IDS.

Devido ao aumento de ameaças e técnicas utilizadas para burlar sistemas de segurança, um *firewall* sozinho, por melhor que seja, pode não defender uma rede, sucumbindo a algum ataque.

As ferramentas IDS têm como objetivo aprimorar a segurança das redes, funcionando em conjunto com as outras ferramentas convencionais.

Para a realização deste trabalho foram escolhidas duas ferramentas IDS (Snort e *RealSecure*), onde ambas foram colocadas a testes para analisar o comportamento de cada uma em situações práticas.

Foi constatado que para a implementação de uma ferramenta IDS é necessário considerar uma série de problemas que possam ocorrer, como por exemplo, as falhas de configuração, redes comutadas, tráfego criptografado, altas taxas de transmissão, sendo que, todos esses são fatores que dificultam a capacidade de detecção de ataques dos IDS, podendo assim, o tornar ineficiente, deixando a rede vulnerável.

Dentre os fatores citados, as falhas de configuração não podem ser muitas, já que podem comprometer a segurança. Tendo isso em mente, a configuração e a operação de um

IDS não podem ser feitas por um usuário desqualificado, pois a sua complexidade em relação às regras de análise é grande e um conhecimento limitado sobre elas pode deixar o sistema ineficiente.

Também foi considerado fora de cogitação utilizar as configurações padrão dos IDS numa rede que requer uma segurança de informação excelente, sendo que as ferramentas se mostram vulneráveis e propensas a falhas quando não personalizadas em relação às necessidades da rede local.

Conclui-se também que, embora haja limitações, os IDS são ferramentas importantes para a segurança de uma rede e assim como os antivírus e *firewalls*, ao longo dos anos devem evoluir, garantindo cada vez mais a segurança de nossas informações.

Ambas as ferramentas apresentaram algumas falhas e vulnerabilidades, pelo fato de estarem operando com suas configurações padrão, mas apesar disso, os IDS detectaram os ataques realizados de modo satisfatório, mesmo sem uma alteração minuciosa em suas configurações.

Devido à constante evolução dos IDS e seus meios complexos de configuração, são sugeridos os seguintes trabalhos futuros:

a) avaliação acerca da evolução das ferramentas IDS ao longo dos anos;
b) testes nas mudanças de configurações nos IDS, analisando as alterações que surgem nas detecções de ataques devido as alterações;
c) implementação de uma interface visual com um modo mais simples de configuração do Snort e suas regras, pois atualmente as mudanças nas configurações ainda são feitas por meio de alterações diretas nos arquivos utilizando o editor de textos;
d) comparação de eficiência de ferramentas IDS com ferramentas IPS, demonstrando a importância do uso de ambas.

REFERÊNCIAS

ALESSANDRI, Dominique. **Using rule-based activity descriptions to evaluate intrusion detection systems**. Switzerland: IBM Research Laboratory Zurich, 2000.

ALLEN, Julia. et al. **State of the Practice of Intrusion Detection Technologies**. Pittsburgh: Software Engineering Institute, 2000.

AURÉLIO, Buarque de Holanda. **Mini Aurélio:** o Dicionário da Língua Portuguesa. Curitiba: Positivo, 2004.

BORGES, Pedro Célio; COUTINHO, Rodrigo Trinck. **Análise de sistemas de detecção de intrusão em redes de computadores**. Trabalho de Conclusão de Curso – Curso de Ciência da Computação , Universidade de Franca, Franca, 2007.

BRANDÃO, A. **Qual a importância da segurança digital para empresas em expansão**. 2008. Disponível em: <http://tinyurl.com/3fqxxnr> Acesso em: 12 nov 2009.

CASWELL, Brian et al. **Snort 2:** Sistema de detecção de intrusão. Rio de Janeiro: Alta Books, 2003.

CORRÊA, Angelita de Cássia. **Metodologia para análise comparativa de Sistemas de Detecção de Intrusão.** 2002. 86 f. Disertação (Mestrado) - Curso de Engenharia da Computação, Instituto de Pesquisas Tecnológicas, São Paulo, 2005.

FAGUNDES, Leonardo Lemes. **Metodologia para avaliação de sistemas de detecção de intrusão**. Monografia – Curso de Informática, Unisinos, São Leopoldo, 2002. Disponível em: < http://tinyurl.com/3bxpakp >. Acesso em 23 nov. 2010

FAULKNER, Matthew J. **Network Based Intrusion Detection System** (NIDS). Disponível em <http://webpages.uah.edu/~faulknmj/660%20extra%20credit.htm> Acesso em: 23 nov 2009.

HERZOG, P. et al. **OSSTMM - Open Source Security Testing Methodology Manual**. Disponível em <http://www.isecom.org/osstmm>. Acesso em: 20 jun 2009.

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício. **Segurança de Redes:** em ambientes cooperativos. 2. ed. São Paulo: Futura, 2003.

LIPPMANN, Richard P. et al. **Evaluating intrusion detection systems:** The 1999 DARPA offline intrusion detection evaluation. Lexingon. Massachusetts: Lincoln Laboratory, Massachusetts Institute of Technology, 2000.

NORTHCUTT, Stephen. et al. **Segurança e Prevenção em Redes**. São Paulo:Berkeley, 2001.

PUCKETZA, Nicholas et al. **A software plataform for testing intrusion detection systems**. IEEE Software vol. 14. no. 5, pp. 43 – 51, 1997.

RUSSELL, Ryan. **Hack Proofing Your Network.** 2. ed. Rio de Janeiro: Elsevier, 2002.

SCRIMGER R. et al. **TCP/IP: a bíblia**. Rio de Janeiro: Elsevier, 2002.

SOARES, M. A .S. **Trabalho sobre IDS**. Disponível em <http://www.gta.ufrj.br/grad/02_2/ids>. Acesso em: 17 nov 2009.

STEFFEN JUNIOR, Julio. **Sistema de detecção de intrusão**. 2003. 95f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário FEEVALE, Novo Hamburgo.

TANENBAUM, Andrew. S., **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003.

VAZ, T. B. et al. **Sistemas de Detecção de Intrusão Livres**: suas limitações e uma arquitetura proposta sobre concentração de mensagens e correlacionamento de eventos. Salvador: UFBA, 2004.

APÊNDICES

APÊNDICE A – Configuração padrão do Snort

A configuração do Snort é feita por meio de edição nos arquivos, ajustando suas características de acordo como necessário.

Para configurar o arquivo snort.conf do Snort, edite as seguintes linhas utilizando o Notepad++:

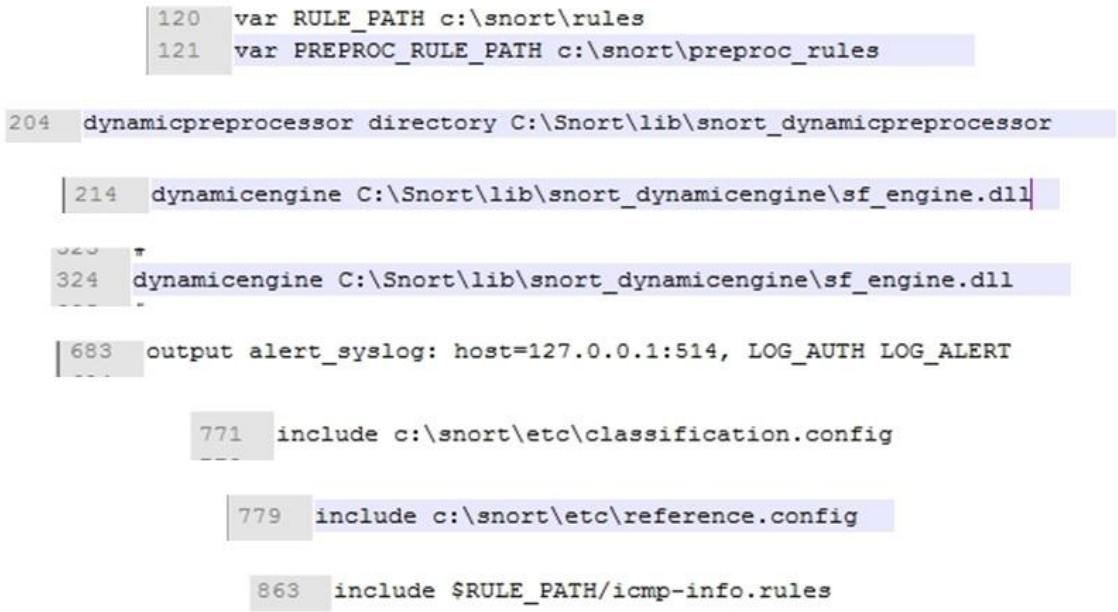

Figura 7. Modificações no snort.conf

Agora salve e feche o arquivo.

Para verificar se o Snort está rodando utilize os seguintes comandos no Prompt:

C:\Snort\bin\Snort –W

Caso apareça na tela dados idênticos a imagem abaixo siga para o próximo passo.

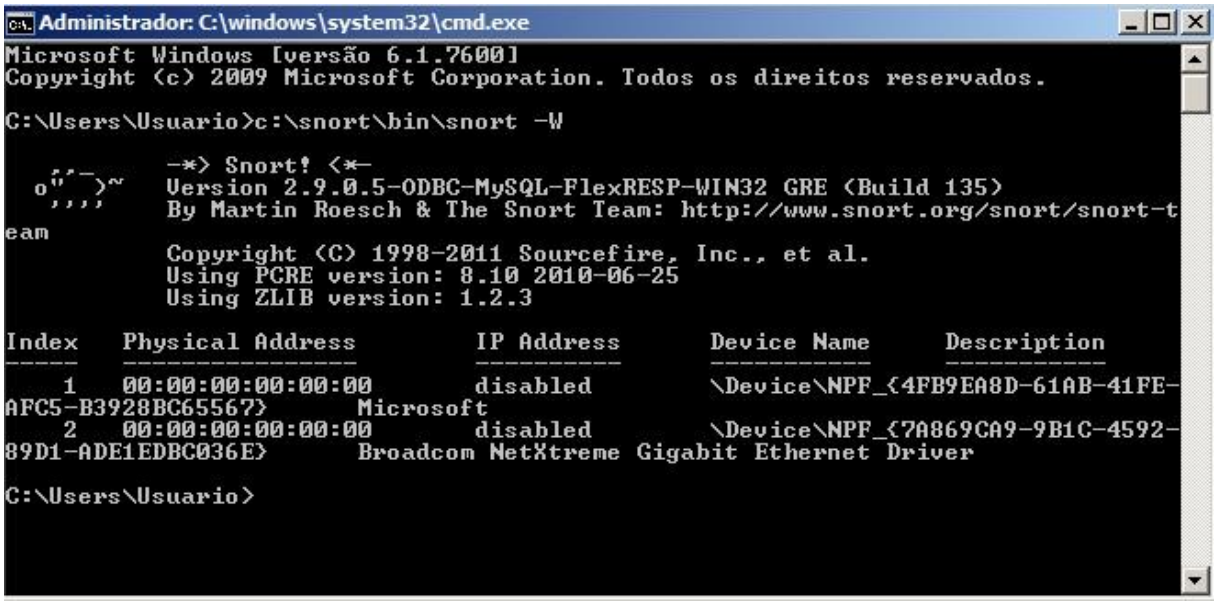

Figura 8. Verificando se o Snort foi instalado com sucesso

Se a tela a cima aparecer, tente rodar o Snort com o comando:

C:\Snort\bin\Snort –v –i2 (no meu caso utiliza-se o numero 2 pois é o “Index” do endereço físico utilizado).

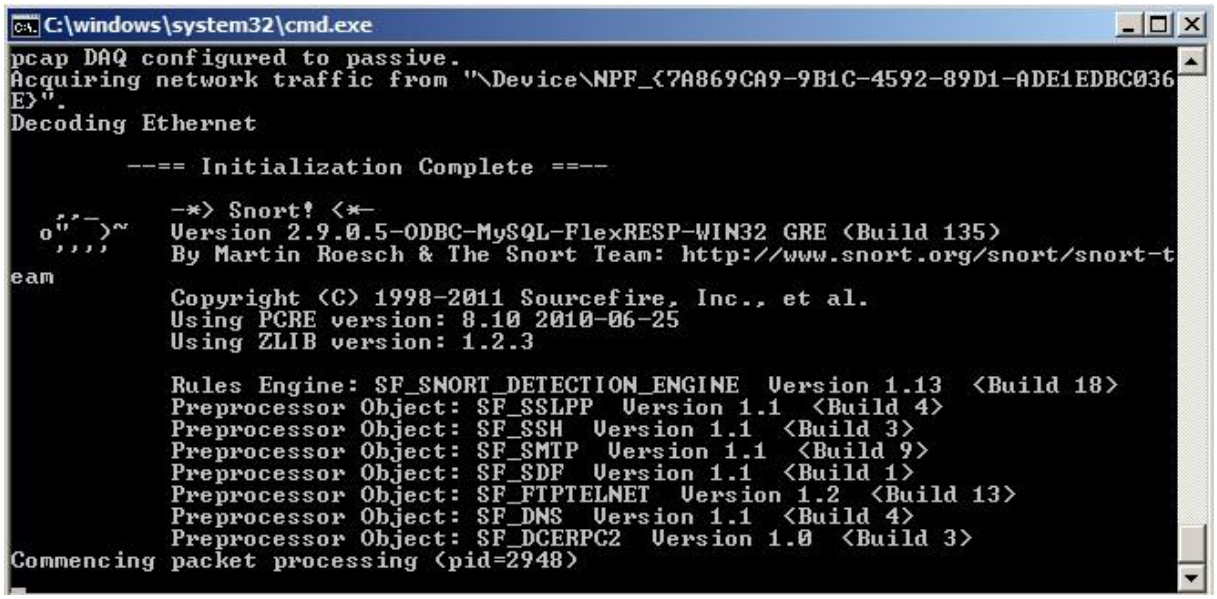

Figura 9. Verificação no funcionamento do Snort

Agora o Snort está rodando. Resta testar se está capturando e analisando pacotes. Aqui o Kiwi Syslog Server Console foi configurado para mostrar os alertas gerados pelo

Snort. Para mais informações sobre como configurar o Snort utilize o arquivo de configuração que o Snort fornece no próprio site pelo link:

http://www.snort.org/assets/135/Installing_Snort_2.8.5.2_on_Windows_7.pdf.

Caso o Snort gere algum alerta o Kiwi mostrará. Foi feito um ping para demonstração:

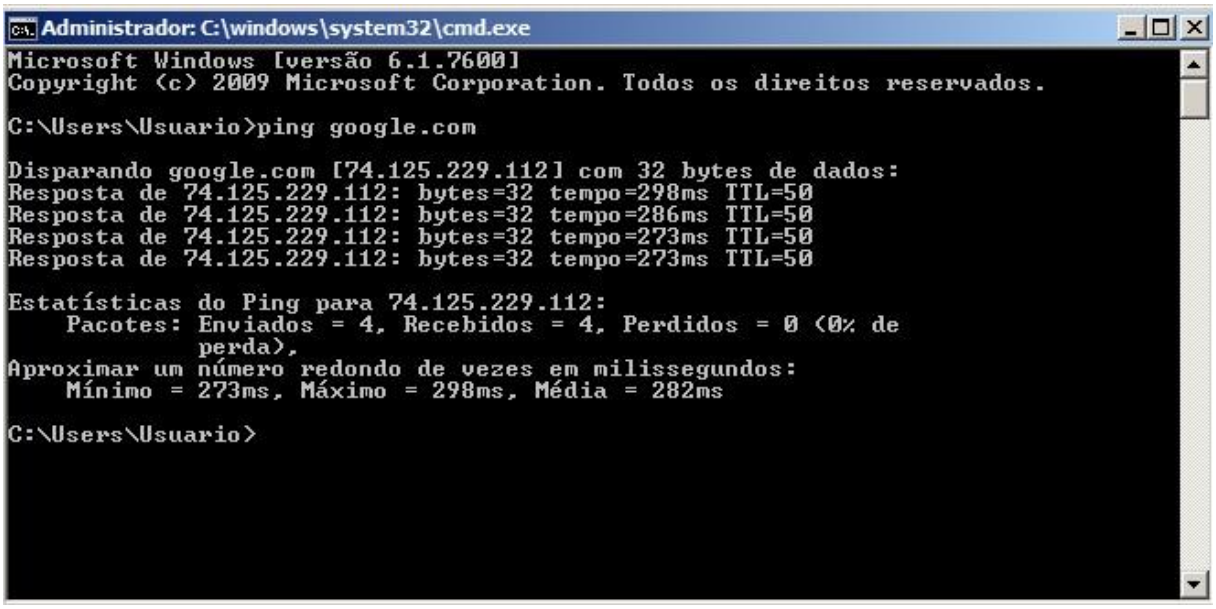

Figura 10. Ping teste

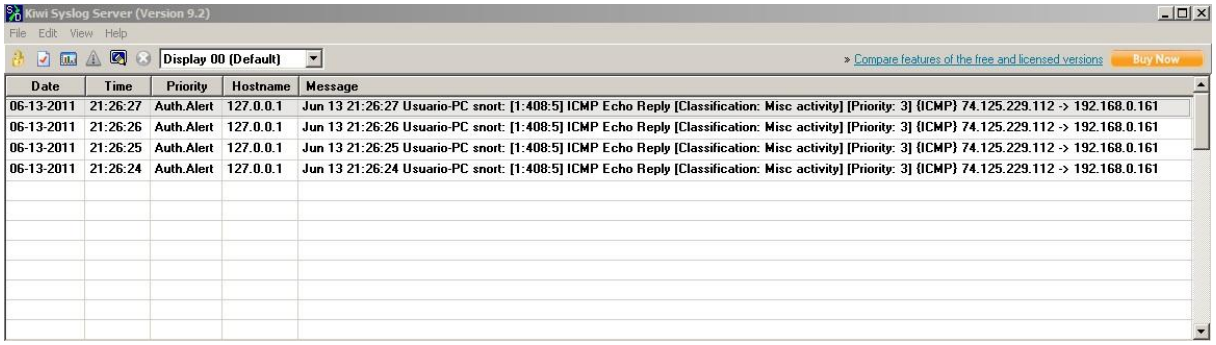

Figura 11. Kiwi mostrando alertas do Snort

O Snort está rodando, capturando e analisando pacotes.

Para atualizar as regras do Snort utilizou-se o Oinkmaster 2.0, uma ferramenta fácil de utilizar, cujos passos também se encontram no link mostrado anteriormente.

APÊNDICE B – Configuração padrão do *RealSecure*

A configuração considerada padrão do *RealSecure* já vem direto da instalação, só é preciso indicar onde estão localizados os arquivos que contém as licenças para que o *Sensor Network* e o *SitePortector* possam receber as informações coletadas no tráfego.

Segue os passos necessários para indicar as licenças:

a) abra o Console do ***RealSecure SiteProtector***;

b) abra **Tools – Licences – Agent/Module**;

c) selecione **Licences**;

d) adicione o caminho arquivos que contém as licenças em **Add**;

e) clique em **OK** e em seguida em **Accept**;

Para confirmar que as licenças foram instaladas com sucesso siga os passos:

a) abra **Tools – Licences – Agent/Module**;

b) clique em **Summary**;

As políticas do *RealSecure* já vêm configuradas e prontas para serem usadas, precisando somente que o administrador da rede escolha quais delas são melhores para aquele segmento de rede. O sistema do *RealSecure* utiliza as assinaturas que se encontram na política chamada *Attack Detector* para capturar os ataques na rede e esta política foi configurada automaticamente durante a instalação do IDS.

Para que haja um melhor desempenho da ferramenta, o mais indicado é que o administrador modifique as políticas existentes de acordo com as suas necessidades e as necessidades da rede, mas nesse trabalho foi utilizado o modo padrão (*default*).

APÊNDICE C – ARTIGO

ANÁLISE E COMPARAÇÃO ENTRE SISTEMAS DE DETECÇÃO DE INTRUSÃO

Claus Pacheco Lottin

Universidade do Extremo Sul Catarinense – Curso de Ciência da Computação
Criciúma – SC – Brasil

klaus1080@hotmail.com

Abstract. Nowadays is rare to exist a company without a computer network. With the increase in the traffic of information on computer networks there is also a proportional increase in the utilization of Internet services, for example, e-mails, websites, transmission and reception of files. These services facilitate the activities of users. However, when used inappropriately can put the network at risk, exposing the system to possible intrusions. To avoid these problems, requires a security policy used in combination with tools to assist in security, like firewalls, antivirus and Intrusion Detection System (IDS). IDS tools detect network attacks and report to the administrator, generating alerts before they can cause significant damage. The IDS are becoming more used by companies, because firewalls and antivirus do not provide total security to the network due to the variety of attacks available today. There are several IDS tools available, among them are noticeable the open source tools, with easy access and free use license, and some proprietary tools. Among these tools, two were selected, Snort (open source) and RealSecure (proprietary) to be subjected to trials in an environment built for analysis. The results obtained are described, comparing the behavior of tools, reporting the deficiencies and advantages of its use.

Resumo. Atualmente é raro existir uma empresa sem uma rede de computadores. Com o aumento considerável do tráfego de informações nas redes de computadores também há um aumento proporcional na utilização de serviços voltados para Internet, como por exemplo, e-mails, acessos a sites, transmissão e recepção de arquivos. Esses serviços facilitam as atividades dos usuários. Porém, quando utilizados inadequadamente podem colocar a rede em risco, as expondo a possíveis invasões do sistema. Para evitar esses problemas, é necessária uma política de segurança utilizada em conjunto com ferramentas para auxiliar na segurança, como *firewalls*, antivírus e *Intrusion Detection System* (IDS). Ferramentas IDS detectam ataques à rede e reportam-se ao administrador, gerando alertas antes que possam causar danos significativos. As ferramentas de detecção de intrusão estão se tornando cada vez mais utilizados por empresas, pois *firewalls* e antivírus não dão total segurança para a rede devido à variedade de ataques existentes atualmente. Hoje em dia existem várias ferramentas IDS disponíveis, dentre elas destacam-se as *open source*, com facilidade de acesso e com licença de uso livre, e algumas ferramentas proprietárias. Dessas ferramentas foram escolhidas duas, Snort (*open source*) e RealSecure (proprietária), para serem submetidas a testes em um ambiente montado para a análise. Os resultados obtidos são descritos e comparando-se o comportamento das duas ferramentas, informando-se as deficiências e as vantagens de seu uso.

Palavras-chave: IDS; Snort; RealSecure; Segurança; Redes.

1. Introdução

Atualmente o uso de redes em ambientes empresariais cresceu notoriamente e em conjunto com esse crescimento também houve o aumento do número de fraudes na Internet, dentre elas invasões a redes com o objetivo de roubar informações, podendo causar prejuízos graves a empresa.

Várias ferramentas de segurança começaram a ser utilizadas pelas empresas para tentar deter as possíveis invasões, como por exemplo, firewalls, antivírus, criptografia de arquivos, protocolos de autenticação e ferramentas IDS. Assim como nos antivírus, também existe uma grande variedade de ferramentas IDS, fazendo com que os usuários fiquem indecisos na utilização de uma em especifico.

Este trabalho visa apresentar uma análise do desempenho de algumas ferramentas de detecção de intrusão de rede (NIDS), demonstrando suas vulnerabilidades e vantagens de uso.

O artigo esta organizado do seguinte modo: a seção 2 descreve conceitos básicos sobre os sistemas de detecção de intrusão (IDS). A seção 3 cita a metodologia utilizada para a análise e comparação dos IDS. A seção 4 descreve as ferramentas IDS utilizadas. A seção 5 apresenta os tipos de ataques utilizados durante os testes. A seção 6 demonstra por meio de tabelas os resultados obtidos dos testes e uma breve comparação entre as ferramentas IDS testadas.

2. Sistemas de Detecção de Intrusão (IDS)

Os sistemas de detecção de intrusão têm como função principal monitorar a rede, ou o host, analisando pacotes e tentando detectar alguma ação que possa ser considerada maliciosa, como tentativas de ataque para obter informações.

Há dois tipos de IDS, os de rede (NIDS) e os de hosts (HIDS). Os IDS de rede ou Network IDS, são geralmente softwares que possuem sensores em certos pontos da rede, que coletam e analisam dados que trafegam por ela com a função de detectar tentativas de invasão. Os IDS de host ou Host IDS, agem em somente um host, ou seja, monitoram todo o tráfego desse host com a mesma função que os IDS de rede, detectar tentativas de ataque.

Utilizou-se somente IDS de rede ao longo desse trabalho.

2.1 Vulnerabilidade do IDS

Existem algumas vulnerabilidades quando se trata de mecanismos IDS, dentre elas os falsos positivos, falsos negativos e a evasão.

Um falso positivo é aquele em que o IDS detecta um evento que na realidade não é um ataque. Geralmente isso ocorre devido à uma configuração inadequada do IDS, havendo regras criadas erroneamente ou excessos de regras. Não chega a ser um problema grave isoladamente, porém falsos positivos em grande número podem complicar o funcionamento do IDS, atrapalhando na análise dos resultados.

Um falso negativo é considerado mais grave que um falso positivo, pois neste caso é um ataque que passa despercebido pelo IDS, tendo sucesso na invasão e causando mais problemas ao sistema. Um falso negativo pode ser um ataque desconhecido, um sobrecarga no sistema, um erro de configuração ou uma evasão.

A evasão é um modo de enganar o IDS, fazendo com que um ataque consiga passar despercebido pela ferramenta. Nesse caso os dados do pacote malicioso são modificados para

que consiga driblar a ferramenta, não coincidindo com alguma possível assinatura de ataques existente na sua base.

3. Metodologia Utilizada

Metodologias de comparação entre IDS são importantes para possibilitar a análise entre as ferramentas, conseguindo assim, mesmo com variações de tráfego e de algumas características de rede, determinar qual das ferramentas se adapta melhor ao ambiente.

Existem várias metodologias para a avaliação de IDS. Dentre elas se destacam as de Pucketza, Lippmann, Alessandri, entre outras. A figura 3.1 mostra uma análise e comparação feita entre as três metodologias citadas.

	Puketza	**Lippmann**	**Alessandri**
Tipo de avaliação			
Avaliação exaustiva	X	X	
Potencialidade de detecção			X
Classificação de ataques			
Descrição técnica do ataque			X
Descrição do contexto do ataque		X	
Tráfego de fundo			
Tráfego de fundo sintético	X	X	
Tráfego fragmentado			
Tráfego homogêneo			
Ambientes de teste			
Independente			X
Específico	X	X	
Resultados apresentados			
Taxa de falsos positivos		X	X
Tipos de ataques detectados	X	X	X

Figura 1. Análise e comparação entre metodologias

Observando a Figura 1, foi verificado que nenhuma metodologia engloba todas as características metodológicas, portanto foi feita uma síntese demonstrando quais as características utilizadas nesse trabalho, sendo assim, a metodologia utilizada no trabalho é um mesclado de metodologias existentes, como é demonstrado na Figura 2.

Tipo de avaliação	
Avaliação exaustiva	
Potencialidade de detecção	X
Classificação de ataques	
Descrição técnica do ataque	
Descrição do contexto do ataque	X
Tráfego de fundo	
Tráfego de fundo sintético	X
Tráfego fragmentado	
Tráfego homogêneo	
Ambiente de teste	
Independente	X
Específico	
Resultados apresentados	
Taxa de falsos positivos	X
Tipos de ataques detectados	X

Figura 2. Metodologia utilizada

Constatou-se que a maior parte das metodologias apenas avalia a base de assinaturas dos IDS, exceto na metodologia de Alessandri, o que pode ser considerado exaustivo e o resultado gerado só é válido por um pequeno período de tempo, já que assinaturas são desenvolvidas muito rapidamente por seus fabricantes. Por outro lado, a metodologia de Alessandri testa a capacidade de detecção dos IDS e seu experimento só precisará ser refeito quando novos recursos forem implementados às ferramentas.

No que tange à classificação de ataques, a metodologia utilizada nesse projeto usa a descrição do contexto do ataque, a mesma utilizada por Lippmann, pois a descrição técnica do ataque poderia se tornar muito complexa e atrapalhar nos testes.

O tráfego de fundo é um fato que pode interferir diretamente no resultado de alguns testes. O tráfego de fundo utilizado nos testes é o tráfego de fundo sintético, gerado pela ferramenta Network Traffic Generator & Monitor e com ela é gerado um tráfego de fundo para tentar atrapalhar a detecção do ataque com a finalidade de comprovar que o tráfego de fundo pode dificultar a funcionalidade da ferramenta IDS.

Com exceção da metodologia de Alessandri, todas elas definem um ambiente de testes para realização dos mesmos. Porém as metodologias de Lippmann e Puketza requerem ambientes de teste considerados complexos, com várias estações e diferentes equipamentos de interconectividade, e com isso podem inviabilizar a reprodução dos testes. Por esses motivos, o ambiente de teste utilizado é um modelo simples e é apresentado posteriormente.

4. Ferramentas IDS utilizadas

Nesta seção serão abordadas as ferramentas IDS utilizadas nos testes. As ferramentas \escolhidas são duas ferramentas IDS de rede (NIDS): a ferramenta Snort, por ser a ferramenta open source mais conhecida e utilizada atualmente, e a ferramenta RealSecure, que é um software comercial distribuído pela IBM.

4.1 Snort

O Snort é um software IDS open source, capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP. Dentre suas funções se encontram a execução de análises de protocolo e associação de padrões de conteúdo (assinaturas), sendo utilizado para detectar uma variedade de ataques. Esta ferramenta é suportada em plataformas das mais diversas. Alguns pontos fortes que o Snort possui além da variedade na compatibilidade com sistemas são: o fato de ser leve, ser pequeno e possuir o maior cadastro de assinaturas, o que aperfeiçoa a verificação de anomalias que ocorrem dentro da rede.

Essa ferramenta está em constante desenvolvimento e atualização. Essas atualizações são feitas diariamente, tanto em código como nas regras de detecção.

Por ser uma ferramenta leve, sua utilização é indicada para monitorar redes TCP/IP pequenas, onde não ocorre um tráfego muito grande, onde tem uma eficácia maior, podendo detectar uma variedade maior de tráfego suspeito.

Outro ponto positivo do Snort é o grande número de possibilidades de tratamento dos alertas gerados.

4.1.1 Funcionamento do Snort

O Snort habilita a placa de rede do computador onde foi configurado para modo promíscuo, permitindo assim que todos os pacotes que trafegam pelo segmento de rede daquela máquina sejam capturados e comparando esses pacotes com as assinaturas existentes é possível detectar um ataque e enviar alertas em tempo real.

A arquitetura do Snort prioriza o desempenho, a simplicidade e a flexibilidade e possui três subsistemas básicos que o compõem: farejador de pacotes, mecanismo de detecção, subsistema de alerta e registro.

4.1.2 Farejador de Pacotes

Farejadores de pacotes são dispositivos utilizados para monitorar a rede, podendo transformar os pacotes que trafegam em dados legíveis para seres humanos.

O Snort utiliza uma biblioteca chamada libpcap para capturar pacotes de toda a rede. Após capturar os pacotes da placa de rede, ela encaminha os pacotes para os mecanismos de decodificação do Snort, para que possam ser analisados e comparados posteriormente por pré-processadores, que são plug-ins responsáveis por classificar o comportamento dos dados capturados pelo farejador. Após ter o seu comportamento classificado, o pacote é mandado para o mecanismo de detecção.

A biblioteca libpcap foi criada inicialmente para sistemas UNIX, porém, atualmente há a sua versão para Windows, chamada de winpcap.

4.1.3 Mecanismo de Detecção

É a parte onde os dados são comparados com as assinaturas existentes. Caso os dados do pacote sejam semelhantes a uma assinatura conhecida pelo IDS, é enviado um sinal de alerta ao processador.

4.1.4 Subsistema de Alerta e Registro

Caso o IDS dispare um alerta, o mesmo pode ser enviado a um log por uma conexão de rede e também tendo a possibilidade de ser armazenado em um banco de dados.

4.2 RealSecure

De acordo com o site da IBM ISS, em 1992, Christopher Klaus desenvolveu a primeira versão do Internet Scanner, uma tecnologia para proteção que identificasse e corrigisse pontos fracos na segurança da rede. Em 1994, Klaus, em conjunto com Thomas Noonan, fundaram a Internet Security System (ISS) com o objetivo de desenvolver um pouco mais o Internet Scanner e comercializá-lo. Em 2006 a IBM efetuou a compra da ISS e passou a controlar a venda dos softwares.

4.2.1 Arquitetura do RealSecure

A ISS possui vários softwares que tem como objetivos: ajudar no gerenciamento, analisar dados em tempo real e controlar a segurança de redes. Entre esses softwares, se destaca RealSecure Protection System. As duas ferramentas utilizadas dessa linha de produtos nesse trabalho são: RealSecure Network Protection e RealSecure SiteProtector.

4.2.2 RealSecure Network Protection

O RealSecure Network Protection é uma ferramenta cujos componentes são sensores de proteção para redes com a capacidade de capturar o tráfego de rede e analisá-lo com a finalidade de detectar ataques. Esse sensor que monitora o tráfego é o Network Sensor. Para que o Network Sensor consiga analisar todo o tráfego da rede, a placa de rede deve ser configurada em modo promíscuo.

Para que a ferramenta Network Sensor conseguisse detectar pacotes suspeitos, ela precisa de políticas de segurança. Estas políticas se encontrar no componente chamado Network Sensor Policies. Nele estão contidas diversas políticas pré-definidas e assinaturas de ataques para que o Network Sensor consiga realizar uma análise confiável e determinar o que podem ser pacotes com conteúdo malicioso.

Caso algum tipo de anomalia venha ser detectado pelo Network Sensor, é enviado um alerta com dados desta. Esses dados podem ser vistos no sistema RealSecure SiteProtector.

4.2.3 RealSecure SiteProtector

O RealSecure SiteProtector é um sistema que gerencia e gera relatórios de segurança para aplicações RealSecure de um modo centralizado e escalonável. O sistema de controle e monitoração do SiteProtector permite a configuração das ferramentas RealSecure de um modo automático para combaterem novas ameaças juntamente com o firewall, utilizando as informações das avaliações de vulnerabilidades.

4.2.3.1 Componentes do SiteProtector

O SiteProtector possui componentes obrigatórios para sua funcionalidade e alguns opcionais. Os componentes essenciais da versão recente são: Agent Manager, Console, Database e Core.

O Agent Manager gerencia e controla as atividades realizadas pelo Desktop Protection e pelo Express Update Server, facilitando a transferência de dados dos sensores pro Event Collector.

O Console é a principal interface e o principal componente do SiteProtector. É onde o usuário pode executar as funções do software, dentre elas se encontram monitoramento de eventos, scan, configuração de políticas e agentes, geração de relatórios, gerenciamento e configuração vários sensores.

SiteProtector Database é a base de dados do SiteProtector. Basicamente é onde são armazenados os dados do agente.

Core é o núcleo do SiteProtector e dentre suas funções se encontram: gerenciamento da comunicação entre Console e Database, controle da funcionalidade dos sensores, proporcionar facilidade nos updates e um fácil acesso ao SiteProtector.

O usuário do SiteProtector também pode editar e criar políticas, fazendo com que ele encontre uma configuração que se adéque ao seu gosto.

O SiteProtector não possui um banco de dados próprio, porém a ferramenta traz na sua instalação o MSDE e as configurações do banco já são feitas durante instalação da ferramenta.

5. Seleção de Ataques

Os ataques que foram gerados contra as ferramentas IDS foram:

5.1 Exploits e Buffer Overflow

Exploits são programas criados em qualquer linguagem de programação que exploram vulnerabilidades em servidores. Eles executam comandos arbitrários no servidor podendo assim dar acesso root (administrador) àquele que o executa. Muitos deles exploram o Buffer Overflow.

O Buffer Overflow ocorre quando um programa recebe uma quantidade de dados muito maior do que está preparado para armazenar em buffer. Nessa quantidade de dados pode possuir algum código com conteúdo mal-intencionado que será executado pelo sistema. Esse tipo de ataque é difícil de ser descoberto e um dos mais utilizados.

5.2 Denial of Service (DoS)

É um tipo de ataque que faz o sistema receber várias requisições ao mesmo tempo, fazendo com que ele sobrecarregue e fique sem condições de uso.

5.3 Port Scan

Ataque que mapeia as portas TCP e UDP a procura de informações referentes a serviços acessíveis. Sendo assim o *hacker* ataca diretamente agindo em cima de serviços ativos.

5.4 HTTP e FTP

Ataques realizados explorando as vulnerabilidades desses protocolos.

5.5 Ferramentas Utilizadas nos Ataques

Ferramentas de *Exploits* e *Buffer overflow*: Nessus e Metasploit Framework.

Ferramentas de *Denial of Service* (DoS): Nessus e Hpring2.

Ferramentas de *port scan*: Nessus, Nmap e LANguard.

Ferramenta de HTTP e FTP: Nessus.

Ferramentas de Evasão: *Fragrouter* e *Whisker*.

6. Testes

Foram feitos testes para a análise do desempenho da ferramenta IDS em detectar ocorrências com e sem tráfego de fundo, sendo realizados os mesmos ataques nas duas condições. Para gerar tráfego de fundo foi utilizada uma ferramenta chamada *Network Traffic Generator & Monitor*.

O ambiente de testes mostrado pela Figura 3 foi montado para a realização dos testes.

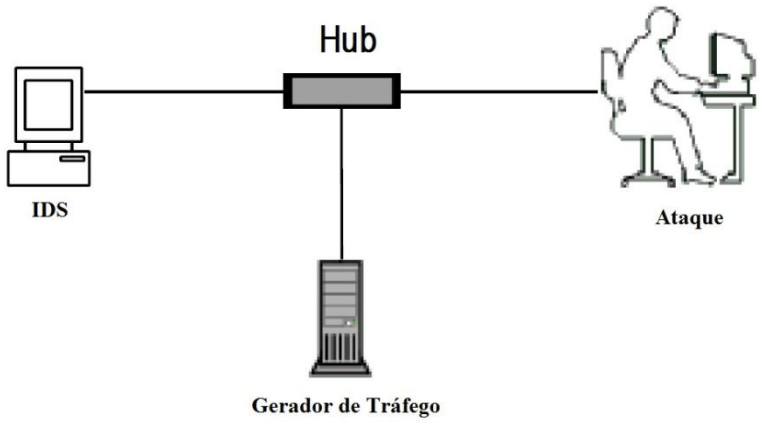

Figura 3. Ambiente de testes

Este ambiente é formado por:

-Gerador de tráfego de fundo: fornecido por uma máquina com a ferramenta *Network Traffic Generator & Monitor* instalada. O tráfego de fundo também é fornecido pelo tráfego comum provido pela Internet;

-Ferramentas IDS: as ferramentas IDS foram instaladas em duas máquinas diferentes, uma ferramente em cada máquina, com configurações semelhantes. Os alertas gerados são lidos nas próprias máquinas.

-Atacante: máquinas com todos os simuladores de ataque instalados.

6.1 Resultados dos Testes

Para melhor visualização dos resultados obtidos, foram organizadas tabelas com o objetivo de deixar mais clara possível a compreensão das informações. Os testes foram replicados 4 vezes e os resultados apresentados nas tabelas são as médias obtidas.

Foi utilizada a ferramenta *Wireshark* em conjunto aos IDS durantes os ataques onde o tráfego de fundo artificial era presente para verificar se todos os pacotes realmente passavam pelas ferramentas. Foi provado que as ferramentas IDS recebiam todos os pacotes, porém, não conseguiam analisar a todos. Em um teste rápido, sem simulações de ataque, a ferramenta Snort recebeu 658 pacotes da rede, a mesma quantidade que o *Wireshark* detectou, mas

analisou 655. O teste foi replicado, em ambas as ferramentas IDS, e o resultado foi a análise de quase todos os pacotes, sempre ficando poucos pacotes fora da análise da ferramenta, porém os pacotes analisados eram sempre superiores a 97% do total de pacotes recebidos.

Snort: reconhecimento de ataques e desempenho da ferramenta						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Exploits	7	5	71	7	2	28
Denial of Service	3	2	66	3	2	66
Port Scan	3	3	100	3	2	66
HTTP	6	4	66	6	3	50
FTP	4	2	50	4	2	50
Total	23	16	69	23	11	47

Tabela 1. Reconhecimento de ataques e desempenho da ferramenta Snort

***RealSecure*: reconhecimento de ataques e desempenho da ferramenta**						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Exploits	7	3	42	7	2	28
Denial of Service	3	2	66	3	2	66
Port Scan	3	3	100	3	3	100
HTTP	6	4	66	6	4	66
FTP	4	4	100	4	3	75
Total	23	16	69	23	14	60

Tabela 2. Reconhecimento de ataques e desempenho da ferramenta RealSecure

As Tabelas 1 e 2 mostram que nenhuma das ferramentas conseguiu detectar todos os ataques efetuados.

Sem a presença de tráfego de fundo artificial gerado pela ferramenta *Network Traffic Generator & Monitor*, as duas ferramentas conseguiram reconhecer a mesma quantidade de ataques, que foram 16 de 23, ou seja, 69% do total. Já quando a ferramenta geradora de tráfego de fundo estava ativa, o *RealSecure* conseguiu reconhecer 14 de 23 (60%), enquanto o Snort conseguiu 11 de 23 (47%). Nesse teste a ferramenta Snort se provou menos eficiente que a *RealSecure*, pois teve sua capacidade de detecção comprometida em situações onde o tráfego de fundo era alto.

Esses resultados podem ser atribuídos também a uma má configuração das ferramentas, pois como foi dito anteriormente, as ferramentas estão operando nas suas configurações padrão. É possível que alterações na configuração possam mudar os resultados dos testes, aumentando a eficiência da ferramenta em situações de stress.

Testes de evasão						
Ataques	**Tráfego de Fundo**					
	Sem Tráfego			**Com Tráfego**		
	Gerados	**Detectados**	**%**	**Gerados**	**Detectados**	**%**
Whisker	1	1	100	1	1	100
Fragrouter	1	1	100	1	1	100

Tabela 3. Testes de evasão

De acordo com a Tabela 3, ambas as ferramentas se provaram eficientes em testes de evasão, pois detectaram todas as tentativas de ataque.

As Tabelas 4 e 5 mostram os resultados dos falsos positivos gerados pelas ferramentas durante os ataques com e sem tráfego de fundo gerado.

Falsos Positivos Gerados – Snort		
Ataques	**Tráfego de Fundo**	
	Sem Tráfego	**Com Tráfego**
Exploits	0	14
Denial of Service	0	5
Port Scan	6	11
HTTP	10	23
FTP	9	14
Total	25	67

Tabela 4. Falsos positivos gerados pela ferramenta Snort

Durante o monitoramento da rede pelo Snort, a ferramenta gerou 25 alertas que não possuem relação com os ataques efetuados nos testes sem presença de tráfego de fundo. Já nos testes com tráfego de fundo presente, os alertas falsos quase triplicaram, indo para 67 alertas, mostrando que a eficácia do Snort foi comprometida dependendo da quantidade de tráfego de fundo aplicada na rede.

Falsos Positivos Gerados – *RealSecure*		
Ataques	**Tráfego de Fundo**	
	Sem Tráfego	**Com Tráfego**
Exploits	1	52
Denial of Service	0	11
Port Scan	3	14
HTTP	14	26
FTP	0	51
Total	18	154

Tabela 5. Falsos positivos gerados pela ferramenta *RealSecure*

A Tabela 5 mostra as ocorrências que a ferramenta *RealSecure* gerou. Sem o tráfego de fundo na rede, foram gerados 18 falsos positivos. Nos testes com a presença de tráfego de fundo, o número de falsos positivos gerados foi 154, quase 9 vezes o número de alertas gerados sem

tráfego, mostrando que assim como o Snort, o *RealSecure* também pode ter sua análise comprometida quando há um tráfego de rede elevado.

Entre as duas ferramentas, o *RealSecure* se provou menos eficiente que o Snort, pelo fato de gerar um número muito superior de falsos positivos em relação ao número gerado pelo IDS *open source*.

6.2 Discussão sobre os Testes e Resultados Obtidos

Em relação aos testes sem a ocorrência do tráfego de fundo na rede, as duas ferramentas conseguiram detectar a mesma quantidade de ataques, porém quando houve tráfego de fundo, o *RealSecure* detectou 60% gerados enquanto o Snort conseguiu somente 47% do total, o que demonstra que o tráfego de fundo pode alterar muito na detecção dos ataques. A ferramenta *RealSecure* apresentou melhor desempenho no reconhecimento de ataques em relação à ferramenta Snort.

A mesma situação aconteceu com os falsos positivos. Os falsos positivos gerados pelas duas ferramentas foram quase da mesma quantidade sem a presença de tráfego de fundo. Já com tráfego de fundo, ambas geraram um número alto de falsos positivos, demonstrando assim que o tráfego de fundo gerado influencia de modo significativo na detecção dos ataques. A ferramenta *RealSecure* gerou mais falsos positivos que a ferramenta Snort, aprensentando um pior desempenho em relação ao IDS *open source*.

Um grande número de falsos positivos pode indicar uma configuração inadequada nos sensores das ferramentas. Sendo assim, verifica-se que é importante que os sensores sejam configurados adequadamente para que sejam minimizados os falsos positivos gerados.

Apesar de alguns erros que possam ocorrer, as duas ferramentas IDS atingiram seu objetivo, detectando ataques e alertando o administrador sobre eles.

7. CONCLUSÃO

No decorrer deste trabalho constatou-se que ferramentas convencionais em si, já não são o suficiente para garantir a segurança de uma rede, seja ela pequena ou grande. Mesmo com a evolução de *firewalls* e antivírus, enquanto existir um usuário operando algum sistema computacional, esse sistema deixa de ser totalmente seguro.
Um dos melhores métodos de tratar esse problema é com uma política de segurança que se adeque à estrutura da empresa, pois nem sempre se pode contar somente com as ferramentas, trabalhando em conjunto com ferramentas auxiliares, como por exemplo, os IDS.

Devido ao aumento de ameaças e técnicas utilizadas para burlar sistemas de segurança, um *firewall* sozinho, por melhor que seja, pode não defender uma rede, sucumbindo a algum ataque.

As ferramentas IDS têm como objetivo aprimorar a segurança das redes, funcionando em conjunto com as outras ferramentas convencionais.
Para a realização deste trabalho foram escolhidas duas ferramentas IDS (Snort e *RealSecure*), onde ambas foram colocadas a testes para analisar o comportamento de cada uma em situações práticas.

Foi constatado que para a implementação de uma ferramenta IDS é necessário considerar uma série de problemas que possam ocorrer, como por exemplo, as falhas de configuração, redes comutadas, tráfego criptografado, altas taxas de transmissão, sendo que, todos esses são

fatores que dificultam a capacidade de detecção de ataques dos IDS, podendo assim, o tornar ineficiente, deixando a rede vulnerável.

Dentre os fatores citados, as falhas de configuração não podem ser muitas, já que podem comprometer a segurança. Tendo isso em mente, a configuração e a operação de um IDS não podem ser feitas por um usuário desqualificado, pois a sua complexidade em relação às regras de análise é grande e um conhecimento limitado sobre elas pode deixar o sistema ineficiente.

Também foi considerado fora de cogitação utilizar as configurações padrão dos IDS numa rede que requer uma segurança de informação excelente, sendo que as ferramentas se mostram vulneráveis e propensas a falhas quando não personalizadas em relação às necessidades da rede local.

Conclui-se também que, embora haja limitações, os IDS são ferramentas importantes para a segurança de uma rede e assim como os antivírus e *firewalls*, ao longo dos anos devem evoluir, garantindo cada vez mais a segurança de nossas informações.
Ambas as ferramentas apresentaram algumas falhas e vulnerabilidades, pelo fato de estarem operando com suas configurações padrão, mas apesar disso, os IDS detectaram os ataques realizados de modo satisfatório mesmo sem uma alteração minuciosa em suas configurações.

8. REFERÊNCIAS

ALESSANDRI, Dominique. **Using rule-based activity descriptions to evaluate intrusion detection systems**. Switzerland: IBM Research Laboratory Zurich, 2000.

ALLEN, Julia. et al. **State of the Practice of Intrusion Detection Technologies**. Pittsburgh: Software Engineering Institute, 2000.

AURÉLIO, Buarque de Holanda. **Mini Aurélio:** o Dicionário da Língua Portuguesa. Curitiba: Positivo, 2004.

BORGES, Pedro Célio; COUTINHO, Rodrigo Trinck. **Análise de sistemas de detecção de intrusão em redes de computadores**. Trabalho de Conclusão de Curso – Curso de Ciência da Computação , Universidade de Franca, Franca, 2007.

BRANDÃO, A. **Qual a importância da segurança digital para empresas em expansão**. 2008. Disponível em: <http://tinyurl.com/3fqxxnr> Acesso em: 12 nov 2009.

CASWELL, Brian et al. **Snort 2:** Sistema de detecção de intrusão. Rio de Janeiro: Alta Books, 2003.

CORRÊA, Angelita de Cássia. **Metodologia para análise comparativa de Sistemas de Detecção de Intrusão.** 2002. 86 f. Disertação (Mestrado) - Curso de Engenharia da Computação, Instituto de Pesquisas Tecnológicas, São Paulo, 2005.

FAGUNDES, Leonardo Lemes. **Metodologia para avaliação de sistemas de detecção de intrusão**. Monografia – Curso de Informática, Unisinos, São Leopoldo, 2002. Disponível em: < http://tinyurl.com/3bxpakp >. Acesso em 23 nov. 2010

FAULKNER, Matthew J. **Network Based Intrusion Detection System** (NIDS). Disponível em <http://webpages.uah.edu/~faulknmj/660%20extra%20credit.htm> Acesso em: 23 nov 2009.

HERZOG, P. et al. **OSSTMM - Open Source Security Testing Methodology Manual**. Disponível em <http://www.isecom.org/osstmm>. Acesso em: 20 jun 2009.

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício. **Segurança de Redes:** em ambientes cooperativos. 2. ed. São Paulo: Futura, 2003.

LIPPMANN, Richard P. et al. **Evaluating intrusion detection systems:** The 1999 DARPA offline intrusion detection evaluation. Lexingon. Massachusetts: Lincoln Laboratory, Massachusetts Institute of Technology, 2000.

NORTHCUTT, Stephen. et al. **Segurança e Prevenção em Redes**. São Paulo:Berkeley, 2001.

PUCKETZA, Nicholas et al. **A software plataform for testing intrusion detection systems**. IEEE Software vol. 14. no. 5, pp. 43 – 51, 1997.

RUSSELL, Ryan. **Hack Proofing Your Network.** 2. ed. Rio de Janeiro: Elsevier, 2002.

SCRIMGER R. et al. **TCP/IP: a bíblia**. Rio de Janeiro: Elsevier, 2002.

SOARES, M. A .S. **Trabalho sobre IDS**. Disponível em <http://www.gta.ufrj.br/grad/02_2/ids>. Acesso em: 17 nov 2009.

STEFFEN JUNIOR, Julio. **Sistema de detecção de intrusão**. 2003. 95f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário FEEVALE, Novo Hamburgo.

TANENBAUM, Andrew. S., **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003.

VAZ, T. B. et al. **Sistemas de Detecção de Intrusão Livres**: suas limitações e uma arquitetura proposta sobre concentração de mensagens e correlacionamento de eventos. Salvador: UFBA, 2004.