UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

ANÁLISE DOS SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES*: SNORT E SURICATA* COMPARANDO COM DADOS DA DARPA

TRABALHO DE CONCLUSÃO DE CURSO

Cléber Taschetto Murini

Santa Maria, RS, Brasil
2014

TCC/REDES DE COMPUTADORES/UFSM, RS MURINI, Cléber Taschetto Tecnólogo 2014

ANÁLISE DOS SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES*: SNORT E SURICATA* COMPARANDO COM DADOS DA DARPA

Cléber Taschetto Murini

Trabalho apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores, Área de concentração em Segurança em Redes, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Tecnólogo em Redes de Computadores**.

Orientador: Prof. Ms. Renato Preigschadt de Azevedo

Santa Maria, RS, Brasil
2014

UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

A Comissão Examinadora, abaixo assinada, aprova o trabalho de conclusão de curso

ANÁLISE DOS SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES*: SNORT E SURICATA* COMPARANDO COM DADOS DA DARPA

elaborado por
Cléber Taschetto Murini

como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores.

COMISSÃO EXAMINADORA

Prof. Ms. Renato Preigschadt de Azevedo
(Presidente/Orientador)

Bruno Augusti Mozzaquatro, Ms. (UFSM)

Claiton Pereira Colvero, Dr. (UFSM)

Santa Maria, 08 de janeiro de 2014.

Dedico este trabalho à minha família, meus pais Vanilda e João, minhas irmãs Lisandra e Gleice, especialmente a minha esposa Silvane pelo apoio incondicional.

AGRADECIMENTO

A Deus pela vida, pela vida e pela oportunidade de vivenciar mais esse momento.

A minha família: meus pais, João Buzatti Murini e Vanilda Taschetto Murini, minhas irmãs: Gleice Taschetto Murini e Lisandra Taschetto Murini pela dedicação e apoio durante todo tempo, mostrar o caminho a seguir, ajudando a alcançar mais esse objetivo.

A minha esposa Silvane Spolaor pela compreensão, atenção, dedicação, companheirismo, força e amor durante mais esta caminhada.

À Universidade Federal de Santa Maria (UFSM) por proporcionar a oportunidade de cursar o ensino superir de forma gratuita e de qualidade.

Aos professores do curso de Redes de Computadores que no começo eram poucos e foram recebendo colegas esses contribuíram com seus saberes incondicionalmente.

Ao professor Renato Preigschadt de Azevedo pelo seu esforço, atenção e dedicação na elaboração deste trabalho.

Aos colegas de curso que, de uma forma ou outra, contribuíram para que eu chegasse até aqui, em especial ao colega Fabrício Bevilaqua Scariotti que demonstrou sua boa vontade em ajudar, não desistindo nas dificuldades encontradas.

Àqueles amigos que disponibilizaram de seu tempo e esforço de uma jornada diária não muito fácil, em especial ao Henrique Sobroza Pedroso pela sua ajuda e experiência na área.

“Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção.”

- Steve Jobs

RESUMO

Monografia
Curso Superior de Tecnologia em Redes De computadores
Universidade Federal de Santa Maria

ANÁLISE DOS SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES*: SNORT E SURICATA* COMPARANDO COM DADOS DA DARPA

AUTOR: CLÉBER TASCHETTO MURINI
ORIENTADOR: Prof. Ms. RENATO PREIGSCHADT DE AZEVEDO

Data e Local da Defesa: Santa Maria, 08 de Janeiro de 2014.

O presente trabalho tem o objetivo de analisar os Sistemas de Detecção de Intrusão: *Snort* e *Suricata,* comparando com dados sintéticos fornecidos pela DARPA (*Defense Advanced Research Projects Agency*). Essa análise foi realizada de forma a comparar o desempenho e as detecções de intrusões por essas duas ferramentas *open-source* (código aberto)*,* as quais serão colocadas em funcionamento fazendo uma análise de dados da DARPA e verificando os ataques detectados, suas vantagens e desvantagens existentes. A pesquisa de modo geral, procedeu-se na análise bibliográfica e estudos de casos de aplicação comparando esses dois sistemas, e a parte prática colocaram-se esses dois Sistemas de Detecção de Intrusão (*Snort* e *Suricata*) em funcionamento rodando os dados sintéticos da DARPA para avaliação dos resultados. *Snort* é uma sistema de detecção com mais de dez anos de surgimento (1998), possui várias regras implementadas, já o *Suricata* lançado há aproximadamente quatro anos (2010), é um NIDS também baseado em assinaturas e compatível com as regras o *Snort,* tirando proveito da atual tecnologia *multithreading* que melhora a capacidade de processamento. Pode-se notar que este trabalho foi viável no quesito custo/benefício, visto que usa ferramentas *Open Source* possibilitando novos estudos e pesquisas sobre o assunto em um futuro próximo.

Palavras-chave: Sistemas de Detecção de Intrusão; *Snort*; *Suricata*, DARPA.

ABSTRACT

Monograph
Course of Technology in Computer Networks
Federal University of Santa Maria

ANALYSIS OF INTRUSION DETECTION SYSTEMS NETWORK: SURICATA SNORT AND COMPARING WITH DATA DARPA

AUTHOR : CLEBER TASCHETTO MURINI
ADVISER: Prof. Ms. RENATO PREIGSCHADT DE AZEVEDO

Date and Venue of Defense: Santa Maria, January 8, 2014.

This study aims to analyze Intrusion Detection Systems: Snort and Suricata, comparing with synthetic data provided by DARPA (Defense Advanced Research Projects Agency). This analysis was performed in order to compare the performance and intrusion detections for these two open-source tools (open source), which will be put into operation by analyzing data from DARPA and verifying detected attacks, their advantages and disadvantages existing. The research generally proceeded in the literature and case studies of application analysis comparing these two systems, and the practice of these two put themselves Intrusion Detection Systems (Snort and Suricata) running synthetic data for DARPA evaluation of results. Snort is a detection system with over ten years of emergence (1998), has several rules implemented since the Suricata launched four years ago (2010), is a signature-based NIDS also compatible with the rules and Snort, taking advantage the current multithreading technology that improves the processing capacity. It may be noted that this work was feasible in the item cost/benefit seen using Open Source tools enabling new studies and research on the subject in the near future.

Key words: *Intrusion Detection Systems; Snort, Suricata, DARPA.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ACIDBASE - *Analysis Console for Intrusion Detection*

DARPA - *Defense Advanced Research Projects Agency*

DoS - *Denial of Service*

DDoS – *Distributed Denial of Service*

FN - Falso Negativo

FP - Falso Positivo

HIDS - *Host Intrusion Detection System*

HTTP - *Hypertext Transmission Protocol*

ICMP - *Internet Control Message Protocol*

IDS - *Intrusion Detection System*

IIS - *Internet Information Services*

IP - *Internet Protocol*

JAI – Jornada Academia Integrada

LOG – *Registro de evento*

MIT - *Massachusetts Institute of Technology*

NIDS - *Network Intrusion Detection Service*

OISF - *Open Information Security Foundation*

SGBD - *Data Base Management System*

SNMP - *Simple Network Management Protocol*

PHP - *Personal Home Page*

TCP - *Transmission Control Protocol*

UDP - *User Datagram Protocol*

UFSM - Universidade Federal de Santa Maria

WEB – *Web pagina*

SUMÁRIO

1 INTRODUÇÃO

Com o crescente uso da internet e sistemas *web* na atualidade necessita-se verificar a importância dos possíveis Ataques de Negação de Serviço (DoS, *Denial Of Service)* entre outros, tornando necessário prover acesso a redes com alta disponibilidade e qualidade. Devido a isto o presente trabalho mostra Sistemas de Detecção de Intrusão em Redes de Computadores (NIDS, *Network Intrusion Detection System*) representados pelas ferramentas *Snort* e o *Suricata.*

Os ataques são uma verdadeira ameaça à Internet e a rede, visando degradar a qualidade, ou tornar completamente sem disponibilidade os serviços oferecidos pelas vítimas. Já os Sistemas de Detecção de Intrusão em Rede são um conjunto de ferramentas de *software* que permitem a análise e detecção de intrusões em redes de dados.

Nesse trabalho desenvolveu-se uma conceituação de redes de computadores e meios de comunicação fazendo uma revisão bibliográfica sobre Detecção de Intrusão (IDS) e Sistemas de Detecção de Intrusão em Redes (NIDS). Também se referencia e analisa-se Sistemas de Detecção de Redes de Computadores baseado-se em regras e anomalias e, após foi feito um estudo e instalação de dois NIDS *opensorce*: *Snort e Suricata*, implementando, comparando e testando as funcionalidades com uma base sintética da segunda semana de 1999 da DARPA contendo ataques.

De acordo com Barford *et al.* (2002 apud AZEVEDO, 2012, p. 17), “as redes de computadores sem análise de tráfego não podem operar eficientemente ou com segurança.” A análise de tráfego é uma atividade essencial para o correto funcionamento de redes. Dentre os ataques em redes de computadores, o DoS é o que ocasiona uma maior perturbação da qualidade de serviço da rede (BADISHI; KEIDAR; SASSON, 2006).

Os NIDS são capazes de detectar diversos tipos de ataques e intrusões, auxiliando na proteção do ambiente, e sua localização é um dos pontos a ser definido com cuidado (NAKAMURA, 2007).

Conforme Mycert (2013), através do seu serviço Cyber999, de abril a junho de 2013, foram notificados um total de 3.093 incidentes representando 23,77%

de aumento em relação ao 1º trimestre de 2013. No 2º trimestre de 2013, incidentes como fraude, códigos maliciosos, intrusão, DoS e *Cyber-assédio* aumentaram, devido a isso, buscou-se desenvolver o projeto para monitoramento através de duas ferramentas com código fonte aberto que são o *Snort e o Suricata*. Esses dois NIDS são multiplataformas e trazem um conjunto de vantagens para detecção de intrusão na rede.

De acordo com Secundado (2012), a Internet no Brasil apresenta crescimentos expressivos, entre 2008 e 2012 mais de 24,5 milhões de internautas novos começaram a navegar na rede, um crescimento de aproximadamente 45% no período. Além disso, existe crescimento no uso da internet Banda Larga em domicílios com acesso a internet que em 2010 esse número chegou a 21% da população (CERT.br, 2012). Devido ao aumento significativo ao acesso e uso da internet tornou-se a mesma mais vulnerável a ataques.

Em frente a essas preocupações e dos propósitos da pesquisa, elaboraram-se os seguintes objetivos, os quais nortearam o desenvolvimento do trabalho:

Objetivo geral:

Analisar os dois Sistemas de Detecção de Intrusão em Redes: *Snort* e *Suricata,* comparando com ataques sintéticos da DARPA.

Objetivos específicos:

- Compreender requisitos, características, modos de atuação e funcionalidades do *Snort e Suricata;*

- Instalação contendo regras desses dois sistemas de detecção propostos;

- Apresentar possíveis correções para que esses ataques não tenham sucesso;

- Verificar como se comporta o *Snort* e o *Suricata* na detecção desses ataques comparando com assinaturas (regras) dessas ferramentas, após instalar e configurar os NIDS utilizou-se os ataques sintéticos da DARPA.

1.1 Organização do Trabalho

Para atingir os objetivos propostos nesse trabalho o mesmo organizou-se por meio das seguintes etapas: Inicialmente no Capítulo 2 será mencionado a Detecção de Intrusão a Rede de Computadores para entendimento de como acontecem ataques, conceituação e o funcionamento do sistema.

No Capítulo 3 desenvolveu-se o conceito de Sistemas de Detecção de Intrusão (IDS) fazendo referências a Ataques de Negação de Serviço (DoS) e demais ataques comuns existentes. O Capítulo 4 demonstra os dois NIDS abordados, *Snort* e *Suricata* com suas instalações e configurações. Esses dois sistemas funcionaram de forma a analisar, e posteriormente verificação de suas funcionalidades de geração de *logs* comparando com a base de dados sintéticos da DARPA (*Defense Advanced Research Projects Agency*).

O Capítulo 5 demonstra os procedimentos metodológicos utilizados para instalação dessas ferramentas e trabalhos relacionados na área com suas referências; já no Capítulo 6 foi mencionada a avaliação e resultados de *logs* gerados dessas duas ferramentas comparando com dados que contém ataques sintéticos da segunda semana da DARPA, e por fim no capítulo 7 apresenta as conclusões e sugestões de trabalhos futuros para implementação deste e outros trabalhos na área.

1.2 Justificativa

A internet é um espaço virtual de encontros, e ao mesmo tempo de controversas, onde cada usuário busca a sua diversidade cultural e seus anseios, oportunizando fontes maliciosas de explorar possíveis falhas existentes nas aplicações e protocolos de comunicação da mesma (MURINI, 2013).

A existência de diversos tipos de ataques em redes de computadores, como por exemplo: ataques de força bruta, ataques para obtenção de permissões, ataques de negação de serviço (DoS), entre outros, vem a perturbar o comportamento e a disponibilidade da rede, causando prejuízo ao usuário.

Referenciando esses ataques desenvolveu-se este trabalho enfatizando Sistemas Detectores de Intrusão (NIDS), fazendo a análise com ataques sintéticos

fornecidos pela DARPA, mostrando possíveis falhas e ataques que prejudicam a mesma, trazendo algumas soluções para seu melhor funcionamento.

2 DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES (IDS)

Neste capítulo serão apresentadas definições sobre detecção de intrusão, intrusão, intrusos, falsos positivos, falsos negativos e registros de eventos *(logs)* em redes de computadores.

2.1 Conceitos relacionados à IDS

Esse Capítulo traz conceitos relacionados à Detecção de Intrusão em redes de Computadores.

2.1.1 Redes de Computadores

Uma rede de computadores, definida em (TANENBAUM, 2003 apud SILVA COSTA, 2009, p. 16), “é um conjunto de computadores interligados com o objetivo de prover comunicação entre os usuários de acordo com seus interesses.” Um exemplo de rede de computadores é a Internet, onde um computador, quando conectado a essa rede, dispõe de uma conexão entre diversas fontes de informação, que consiste em servidores conectados à rede.

2.1.2 Detecção de Intrusão

O conceito de detecção de intrusão (*Intrusion Detection System* – IDS) pode ser referenciado a uma tentativa com sucesso ou não de acesso, manipulação das informações e/ou tornar um sistema invadido não confiável. Um ataque é caracterizado como uma tentativa de modificar, manipular, ou perturbar o funcionamento de um sistema, bem sucedida ou não (WANG, 2009).

2.1.3 Intrusão em rede

Segundo Silva e Sampaio (2006) intrusão pode ser definida como um conjunto de ações desencadeadas pelo intruso que compromete a estrutura básica da segurança da informação de um sistema: integridade, confidencialidade e

disponibilidade. Devemos ter uma diferenciação entre “Ataque” e “Intrusão”, pois parecem ser a mesma coisa, mas tem algumas particularidades: ataque refere-se à tentativa de perturbação, já intrusão é um ataque realizado que obteve sucesso (foi bem sucedido), pois invadiu a rede.

2.1.4 Intrusos

De forma simplificada pode-se dizer que intruso é quando um usuário tenta invadir um Sistema ou fazer mau uso do mesmo (LAUFER, 2003).

Classificam-se os intrusos em dois tipos:

·*Intrusos Externos:* são usuários que não possuem acesso físico a Rede ou Sistemas que estão atacando.

·*Intrusos Internos:* são usuários autorizados a usar os recursos disponíveis pelo sistema, estes possuem acesso às máquinas e seus recursos, são identificados como usuários legítimos do Sistema, que com seus privilégios para tirarem algum proveito, podendo ser de duas formas: mascarados (passam como usuários legítimos do sistema) e clandestinos (aqueles que têm o poder de direcionar os dados de controle para si próprios), tendo como principal diferença o seu perfil de uso.

Os intrusos clandestinos são mais difíceis a serem detectados, eles aproveitam das condições privilegiadas que possuem para conseguir vantagens não autorizadas, podendo assim direcionar dados para outros caminhos (LAUFER, 2003).

Uma representação de ameaça em um sistema pode ser vista na Figura 1, onde os recursos protegidos são vistos como anéis de controle e anéis de usuários.

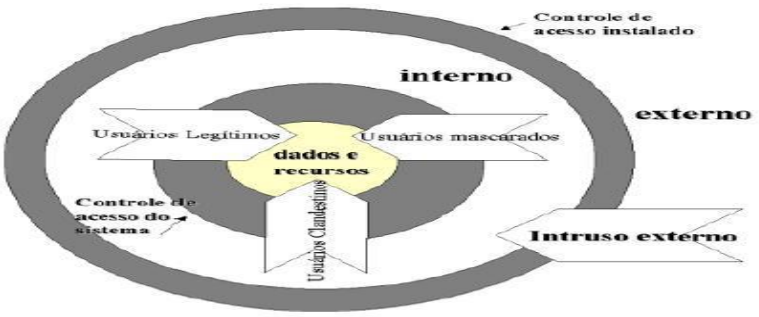

Figura 1: Tipos de intruso. (LAUFER, 2003)

Na figura pode-se verificar que quando o controle de acesso possuir uma falha, os intrusos externos passam pelo mesmo e buscam acesso ao Sistema. Já os intrusos internos quando encontram uma falha ou vulnerabilidade no controle de acesso ao sistema invadem os dados e os recursos se passando por usuários legítimos.

2.1.5 Falsos Positivos e Falsos Negativos

Na atualidade um dos problemas de IDS, levando em conta a questão de eficiência (precisão da detecção) que contabiliza através do número de erros de detecção que ocorrem, podemos verificar os seguintes critérios:

1 - Falsos positivos (FP): ocorrem quando pacotes são identificados pelas ferramentas de segurança como tentativas de ataque, quando na verdade trata-se de ações legítimas e não ataques. Pode ocorrer um número grande de falsos positivos que irá até mesmo atrapalhar a análise de um arquivo de registro de eventos, onde os ataques verdadeiros podem passar despercebidos (CRUZ, 2000).

2 - Falsos negativos (FN): ocorrem quando não são identificados/detectados as tentativas autênticas de ataques. No caso de uma ferramenta de detecção de intrusão usar análise de assinaturas, o requisito seria que as mesmas sejam atualizadas constantemente para verificar que um *log* de ataque que não tenha suas regras, passe despercebido (NED, 1999).

2.1.6 Registros de Eventos *(logs)*

Registros de eventos ou mais comumente chamados de *logs* são os registros de atividades geradas por programas computacionais, geralmente por incidente de segurança detectados pelo *firewalls* ou por NIDS.

Nos NIDS, a geração de *logs* pode se dar tanto para casos de tentativa de invasão sem sucesso, por exemplo, erro de senha por três vezes, ou quando um intruso obtém sucesso na intrusão (BARBOSA, 2006).

Segundo Cert.br (2012), *logs* são essenciais para notificação de incidentes, pois permitem que diversas informações importantes sejam detectadas, como por exemplo: a data e o horário em que uma determinada atividade ocorreu, o fuso horário do *log*, o endereço IP (*Internet Protocol*) de origem da atividade, as portas

envolvidas e o protocolo utilizado no ataque (TCP, UDP, ICMP, etc.), os dados completos que foram enviados para o computador ou rede e o resultado da atividade (se ela ocorreu com sucesso ou não).

2.2 Análise e comportamento do fluxo de dados na rede

Apenas com um monitoramento mais adequado do tráfego de rede é possível determinar certos tipos de ataque. Para isso, existem várias maneiras de obter as informações sobre o tráfego de rede, sendo uma delas através de consultas SNMP (*SNMP-based*) dirigida aos dispositivos conectados à rede. Este tipo de consulta retorna informações a respeito do nível do tráfego, e não fornecem dados suficientes para uma análise de segurança mais detalhada dos sistemas monitorados (BERTHOLDO; ANDREOLI; TAROUCO, 2003).

Um fluxo pode ser identificado unicamente pelas seguintes informações, conforme Amoroso (1999):

- Endereço IP de origem e de destino;
- Porta de origem e destino (camada de transporte);
- Tipo de protocolo (camada de rede);
- Tipo de serviço (cabeçalho TCP);
- *Interface* de entrada do roteador.

Uma vez os dados gerados no *gateway* e armazenados em um servidor, eles são processados por ferramentas, gerando informações em modo gráfico e permitindo consultar os dados armazenados na base de dados de fluxos, através de um conjunto de ferramentas adicionais.

Os dispositivos conectados a rede são capazes de verificar o tráfego, identificando uma sequência de pacotes (mensagens) que são trocadas (AZEVEDO, 2012), assim esse tráfego pode ser definido como:

• Normal: tráfego legítimo, ou seja, não existe a ocorrência de ataques ou anomalias;

• Anômalo: presença de ataques ou anomalias como interrupção de segmentos da rede.

2.3 Localizações de um IDS na rede

Em alguns casos a análise feita pelos NIDS abrange segmentos da rede mais sensíveis, dependendo da política de segurança, devido a este fato normalmente esses sistemas ficam após os *firewalls* e roteadores, conforme é apresentado por Rehman (2013) na Figura 2, o posicionamento mais comum para um Sistema de Detecção de Intrusão:

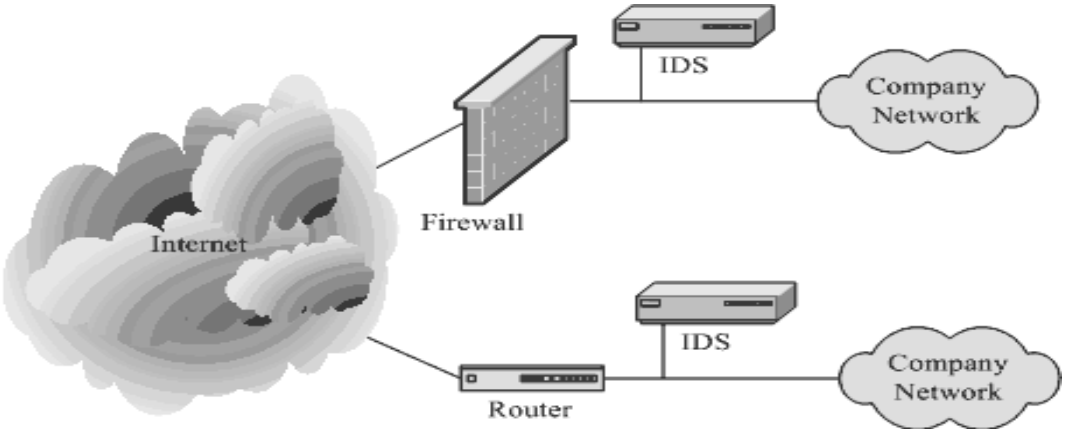

Figura 2. Localização dos Sistemas de Detecção de Intrusão (REHMAN, 2013, p. 10)

O posicionamento dos IDS depende de uma série de fatores, esses fazendo referência à topologia da rede e as atividades de intrusão que se deseja monitorar.

Os ataques têm a sua origem em qualquer ponto da rede, tanto em redes locais ou outras. Muitas vezes, pode não ser suficiente a colocação dos equipamentos de monitoramento apenas nos pontos de entrada na rede (CARDANA, 2006).

2.4 Tipos de detecção de intrusão segundo a arquitetura

A combinação de diferentes tipos de IDS é importante para que a organização esteja adequadamente protegida contra ameaças, principalmente dos ataques realizados internamente e dos ataques vindos da internet (NAKAMURA, 2007), podendo ser classificados em detecção de intrusão baseado em *host* e rede.

Os Sistemas de Detecção de Intrusão baseados em *Host (*HIDS – *Host Based Intrusion Detection System*) realizam o monitoramento das atividades do sistema e

dos programas que rodam no mesmo, com base em informações de arquivos de *logs* ou de agentes de auditoria.

Os HIDS são capazes de monitorar acessos e alterações em importantes arquivos do sistema, modificações nos privilégios dos usuários, processos do sistema, programas que estão sendo executado, uso da CPU e detecção de *port scanning.* Essas ferramentas são usadas para detectar atividades maliciosas em um único computador (KIZZA, 2005).

A detecção de Intrusão baseados em *host* conforme Kizza (2005) apresenta algumas vantagens:

- Capacidade de verificar sucesso ou falha de um ataque rápido analisando os *logs* do evento, apresentando informações mais precisas e menos falsos positivos;
- Detecção em tempo quase real, e o alertas são enviados ao administrador rapidamente;
- Não necessita de *hardware* adicional para sua instalação assim tendo um custo reduzido;
- Acessa informações antes e após a encriptação de dados;
- Capaz de analisar atividades em baixo nível, como acesso as permissões dos arquivos e tentativas de mudanças de privilégios;

O sistema de detecção de Intrusão baseados em Rede *(Network-Based Intrusion Detection System* - NIDS) monitora o tráfego do segmento da rede, geralmente com a interface de rede atuando em modo *“promíscuo”*.

A detecção é realizada com a captura e análise dos cabeçalhos e conteúdos dos pacotes, que são comparados com assinaturas ou padrões conhecidos. Exemplos de NIDS são o *RealSecure*, o *Suricata*, o NFR, o *Snort,* entre outros que serão especificados e estudados no capítulo 3 deste trabalho.

3 SISTEMAS DETECTORES DE INTRUSÃO EM REDES DE COMPUTADORES

Os Sistemas de Detecção de intrusão em Redes de computadores (NIDS, *Network Intrusion Detection System*) são utilizados para monitoramento do tráfego de dados de uma rede ou de um segmento de rede. A análise é realizada em dados coletados da rede, ou em base de dados disponíveis ao NIDS (AZEVEDO, 2012).

Segundo Nakamura (2007),

> NIDS são componentes essenciais em um ambiente cooperativo. Sua capacidade de detectar diversos ataques e intrusões auxilia na proteção do ambiente, e sua localização é um dos pontos a serem definidos com cuidado (NAKAMURA 2007, p. 264).

Conforme Kizza (2005) as principais vantagens dos NIDS são:

- Detecção e resposta em tempo real: estando em pontos estratégicos da rede, consegue detectar intrusões rapidamente e notificar ao administrador;

- Capacidade de detectar ataques que os HIDS não pegam, porque monitora no nível de transporte da arquitetura da rede. Nesse nível analisa pacotes não apenas por endereços, mas também em números de porta;

- Dificuldade de remover evidências: Os NIDS ficam em uma máquina dedicada e protegida, o que dificulta bastante a remoção de evidências pelo atacante;

- Baixo custo: segundo Wang (2009), é necessário apenas sondas em alguns pontos estratégicos da rede. A monitoração passiva é outro ponto positivo, pois não há tráfego de rede normal resistente a intrusão;

Os NIDS possuem também algumas desvantagens como: não é possível analisar protocolos criptografados, há dificuldade de identificar ataques fragmentados, possui pontos cegos, ou seja, na maioria das vezes os NIDS são colocados nas bordas da rede com isso alguns segmentos não são vistos para análise.

3.1 Componentes do NIDS

De acordo com Nakamura (2007), um sistema de detecção de intrusão funciona de acordo com uma série de funções que, trabalhando de modo integrado,

são capazes de detectar, analisar e responder as atividades suspeitas. A figura 3 apresenta as funções e distribuição de um NIDS abaixo.

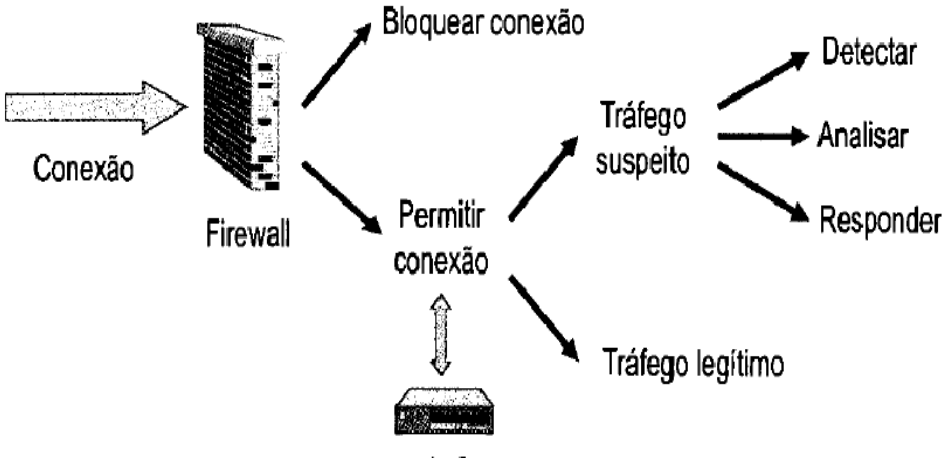

Figura 3: Funções dos NIDS. (NAKAMURA, 2007).

Como se verifica na Figura 3, após o *firewall* liberar as conexões, os Sistemas de Detecção coletam, analisam, armazenam essas informações e respondem às atividades suspeitas, classificando o tráfego como suspeito ou legítimo.

Para que o sistema tenha um bom desempenho na detecção de intrusão necessita-se um conjunto de ferramentas básicas como um coletor (sonda), analisador, Banco de dados (*Mysql*), atuador e monitor, conforme especificado abaixo:

- *Coletor:* O coletor é responsável por capturar descritores do tráfego de rede, e normalmente está conectada em algum ponto de interconexão da rede, como por exemplo: roteador de borda, *bridge*, *firewall*, etc. O desempenho do coletor depende dos equipamentos da rede usados para a coleta, principalmente em redes de grande tráfego. Alguns *firewalls* também atuam como coletor, armazenando informações para os NIDS (NORTHCUTT; NOVAK, 2002 apud PERLIN, 2010).

- *Analisador*: é um componente responsável por verificar os dados coletados buscando por eventos que indiquem uma intrusão ocorrida ou que esteja ocorrendo, tendo diferentes abordagens para análise dos dados, como a baseada em assinaturas ou anomalias descritas no decorrer do trabalho.

- *Banco de dados*: é onde ficam guardadas as informações dos NIDS e os eventos suspeitos, para posteriormente busca dessas informações e análise mais detalhada.

- *Notificador*: é o sistema responsável pelo envio de alertas ao administrador. Os alertas frequentes, com falsos positivos são prejudiciais e deixam o sistema com pouca confiança.

- *Atuador*: é uma ferramenta que possui a capacidade de execução de ações automatizadas quando uma anomalia é detectada pelo IDS. Na maioria das vezes o monitor ou terminal de comando tem o objetivo de fazer ligação entre o administrador do sistema e o sistema de detecção, sendo o monitor usado para configurar, verificar o funcionamento do IDS e até mesmo envio de alertas.

3.2 Classificações dos métodos de detecção:

Os sistemas de detecção de intrusão se utilizam de algumas metodologias para a realização da análise dos dados e, consequente detecção de incidentes. Assim, sendo classificados quanto ao método de análise ou detecção dos dados normalmente de duas formas: baseados em assinaturas e baseados em anomalias. A maioria dos sistemas de detecção existente se utiliza de ambas as metodologias, de forma separada ou integrada, para uma detecção híbrida e mais precisa de possíveis incidentes (SCARFONE; MELL, 2007).

3.2.1 Sistemas de Detecção de Intrusão de Rede baseado em Assinaturas

Um Sistema de Detecção de Intrusão baseados em Assinaturas compara os dados coletados em um ambiente com uma base de dados de ataques conhecidos ou regras pré-definidas por um sistema de detecção. Quando os eventos analisados são compatíveis com alguma assinatura da base de dados um alarme é disparado.

Segundo Roesch (1999), a detecção baseada em assinatura identifica ataques através da análise de assinaturas previamente estabelecidas sobre o comportamento padrão de determinado tipo de ataque, que são geradas por especialistas.

Com o surgimento de novas formas de ataques ou variações são necessárias atualizações constantes na base de ataques. Porém, mesmo com essa base atualizada, os NIDS têm dificuldades de detectar ataques desconhecidos, ataques mutantes ou ataques camuflados. De acordo com cada sistema, ataques são

modelados como padrões de eventos específicos e quando a ocorrência de algum ou parte desses padrões é observada considera-se que há um ataque em curso (DEMIRAY, 2005).

A metodologia baseada em assinaturas é eficiente na detecção de ameaças com comportamento constante e já conhecidas, entretanto, é ineficaz na detecção de ameaças ainda desconhecidas ou na detecção de ameaças já conhecidas e que se utilizam de técnicas de evasão e, consequentemente, tem o comportamento original alterado (DEMIRAY, 2005).

Os NIDS baseados em assinaturas são bastante precisos nas suas detecções, apresentando baixo número de falso positivo, mas devido à dificuldade de detectar de novos ataques, podem apresentar grande número de falsos negativos, onde os ataques não são detectados, sendo uma possível falha de segurança.

Existem diversos NIDS disponíveis para o uso, como o Suricata *(Open Information Security Fundation), Snort* (Roesch, 1999) (*software livre*), *NetRanger* (Cisco, 2011) (*software* proprietário). Estes NIDS são baseados em assinaturas, os quais inspecionam eventos atuais e comparam suas regras (assinaturas) com dados da rede (WANG, 2009).

Na Figura 4 verifica-se que os dados da auditoria são comparados pelo perfil do sistema (regras estabelecidas) e quando houver uma combinação dos dados da rede com as regras estabelecidas há uma detecção de ataques e envio de alertas.

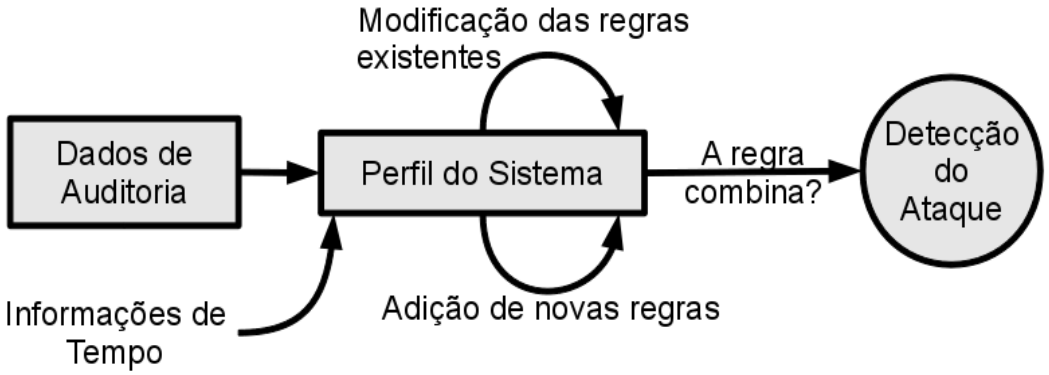

Figura 4: Detecção de Intrusão Baseados em Assinatura. Adaptado de (SUNDARAM, 1996)

Os diferentes tipos de assinaturas utilizados nesses sistemas podem ser (WANG, 2009):

– Assinaturas de redes: informações dos pacotes que podem afetar a execução normal da rede consistem em assinaturas de cabeçalhos ou de campo de

dados (*payload*), os quais verificam ações do usuário, enquanto assinaturas de cabeçalhos verificam pacotes maliciosos que são capazes de identificar, como, por exemplo, ataques de difusão (*broadcast*);

– Assinaturas de *hosts*: utilizam informações de comportamentos que podem afetar na execução normal do sistema. Um exemplo seria três tentativas seguidas de senha errada de determinado usuário. Essas assinaturas podem ser:

• de evento-único: um único comando que e um comportamento suspeito;

• de multi-eventos: formada por grupos de várias assinaturas de evento único;

• de multi-*hosts*: formada por sequências de assinaturas de eventos-únicos originados de diferentes *hosts*;

• compostos: faz a união de assinaturas de rede com assinaturas de *hosts* para identificar certos tipos de ataques que não podem ser identificados somente por um tipo de assinatura.

3.2.2 Sistemas de Detecção de intrusão de Rede baseado em Anomalias

Nos NIDS baseado em Anomalias é analisado o comportamento do tráfego de rede, sendo classificada como anomalia, a ocorrência de um evento que não segue o comportamento esperado, ou semelhante ao suportado pela rede (CHANDOLA; BANERJEE; KUMAR, 2009).

Anomalia é um evento que causa uma alteração em relação ao perfil padrão do sistema (KRUEGEL, 2003). De um modo amplamente analisado uma anomalia na rede pode ocorrer devido a um ataque, falha de equipamentos, problemas de configuração, sobrecarga da mesma ou uso inadequado de algum serviço ou recurso, então a possibilidade com NIDS baseados em anomalias na detecção dessas falhas é grande, se tornando um sistema complexo.

Em função de uma observação prévia das características de um sistema durante um determinado período de tempo torna-se possível definir o perfil de comportamento de seus diversos componentes (rede, usuários, estações, aplicações). Assim, esta metodologia de detecção se baseia na observação de eventos que ocorrem no sistema e sua comparação com um perfil já definido para uma possível identificação de desvios do comportamento normal, ou seja, identificação de anomalias no comportamento do sistema (DEMIRAY, 2005; LINDA, VOLLMER; MILOS, 2009).

A maior vantagem deste tipo de abordagem é a possibilidade de se detectar técnicas de intrusão ainda desconhecidas, pois embora não se conheça o seu comportamento, esse deve se distinguir naturalmente do comportamento esperado e de acordo com o perfil existente.

Além da necessidade de uma base de dados muito extensa, as principais desvantagens dessa metodologia é a impossibilidade de se aprender totalmente o perfil conforme o sistema se torna complexo e o fato que o comportamento dos componentes do sistema monitorado pode mudar com o tempo em consequência da mudança natural de comportamento dos usuários, causando uma taxa elevada de falsos alarmes (LINDA, VOLLMER; MILOS, 2009).

Os sistemas baseados em anomalias baseiam-se na supervisão de comportamentos anômalos do sistema, assumindo que as atividades anormais podem ser tentativas de invasões. Essa técnica constrói primeiro um modelo, geralmente estatístico, capaz de descrever as atividades normais dos usuários, dos *hosts*, ou o tráfego normal da rede, marcando qualquer comportamento que significadamente desvia de um modelo como um ataque.

A Figura 5 mostra quando houver um comportamento diferenciado de um novo perfil diante do que era considerado normal anteriormente é detectado um ataque.

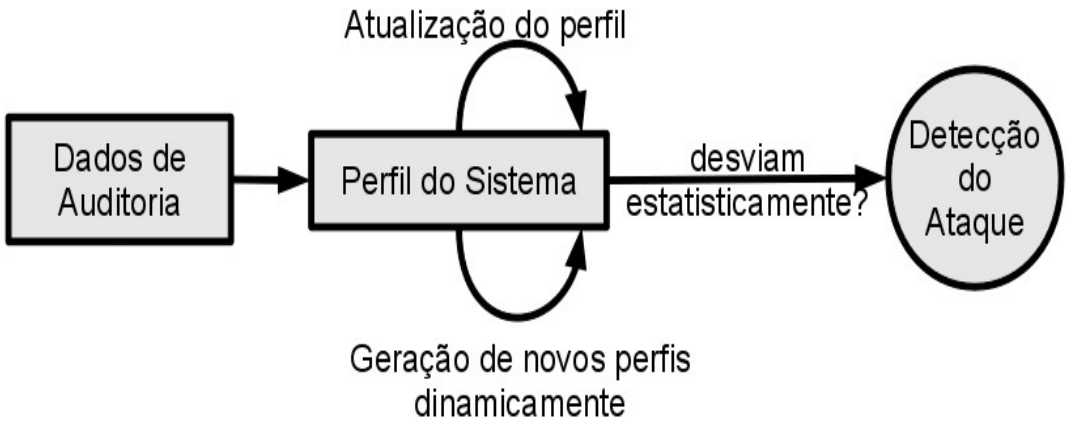

Figura 5: Métodos de Detecção Baseados em Anomalia*. Adaptado de* (SUNDARAM,1996).

Esse tipo de IDS apresenta erros devido à identificação de atividades normais de um usuário como atividades anormais, ou quando deixa de identificar atividades anormais como invasão por parecerem muito com a atividade costumeira de um usuário. Podem-se encontrar os comportamentos (KUMAR e SELVAKUMAR, 2009):

– Intrusivo, mas não anômalo: também chamados de falso negativos. Neste caso, ocorre uma falha na detecção, pois a invasão não provoca atividade anormal, não sendo detectada pelo sistema;
– Não intrusivo, mas anômalo: também chamados de falsos positivos. Neste caso, ocorre uma falha na detecção, que indica erroneamente a anormalidade como uma invasão;
– Não intrusivo e não anômalo: também chamados de verdadeiros negativos (comportamento normal), neste caso a atividade não é anômala e não é evidenciada como invasão;
– Intrusivo e anômalo: também chamados de verdadeiros positivos. Neste caso, ocorre um acerto, pois a atividade é anômala e evidenciada como invasão.

Das diferentes abordagens de algoritmos utilizados, somente duas tem sucesso nas aplicações da última década: modelos estatísticos e baseados em redes neurais (PIETRO, MANCINI, 2008).

Os sistemas orientados a conexão poderiam até utilizar o campo de dados (*payload*) dos pacotes, porém o tempo utilizado para essa tarefa seria grande, o que diminuiria o desempenho do sistema. Na prática, tipicamente obtém-se o número de *bytes* enviados e recebidos, a duração da conexão e o protocolo da quarta camada do modelo de referência TCP/IP que foi utilizado (PIETRO, 2008).

Quanto aos tipos de dados utilizados, têm-se os sistemas que utilizam o cabeçalho e os que utilizam o *payload* do pacote. O primeiro considera somente os cabeçalhos dos pacotes, utilizando os cabeçalhos da terceira camada e, se existir, da quarta camada do modelo de transferência TCP/IP. Já os baseados no *payload* analisam os dados somente da quarta camada do modelo de transferência TCP/IP (aplicação). Existem também sistemas híbridos, que misturam informações coletadas dos cabeçalhos de pacotes e, se houver dos dados do *payload* da quarta camada do TCP/IP (PIETRO, 2008).

3.3 Ataques em Redes de Computadores

Existem diversos ataques em redes de computadores, conforme farei uma descrição dos mais comuns.

3.3.1 Ataques de Negação de Serviço (Dos)

De acordo com CERT (2013), a definição de Ataques de Negação de Serviços (*Denial of Service* - DoS) consiste em tentativas de impedir usuários legítimos de utilizarem um determinado serviço de um computador. Para isso, são usadas técnicas que podem: sobrecarregar uma rede a tal ponto em que os verdadeiros usuários dela não consigam usá-la; derrubar uma conexão entre dois ou mais computadores; fazer múltiplas requisições a um site até que este não consiga mais ser acessado; negar acesso a um sistema ou a determinados usuários.

Para Handley (2006) não é possível distinguir um simples ataque de DoS e/ou uma multidão de *flash,* segundo Azevedo (2012) os ataques *Back, PoD* e *MailBomb* onde são enviadas milhares de requisições a um servidor, ocasionando a negação do serviço aos usuários legítimos. Isso pode ser confundido com tráfego pesado de dados e um ataque malicioso, a defesa está na prevenção de tráfego elevado na rede que venha perturbar o seu comportamento.

O primeiro registro de atividades DDoS (Ataques de Negação de Serviço Distribuído) aconteceu em Julho de 1999 (CERT, 2013), entretanto apenas no primeiro quadrimestre de 2000 tiveram a atenção do público, quando um grande número de *sites* como o *Yahoo, EBay, Amazon e CNN* ficaram inoperantes devido a ataques dessa natureza. No Brasil, no mesmo período, teve-se notícia de ataques contra sites como: UOL, Globo e IG. A primeira versão do verme da *Web* (*Worm Code Red*), desenvolvida em 2001 tinha intenção de elaborar um ataque DoS com todas as máquinas infectadas à Casa Branca, nos Estado Unidos (LEMOS, 2013).

Os DoS são classificados em duas categorias, como pode ser observado na Figura 6 (KUMAR e SELVAKUMAR, 2009): *esgotamento de largura de banda e esgotamento de recursos*.

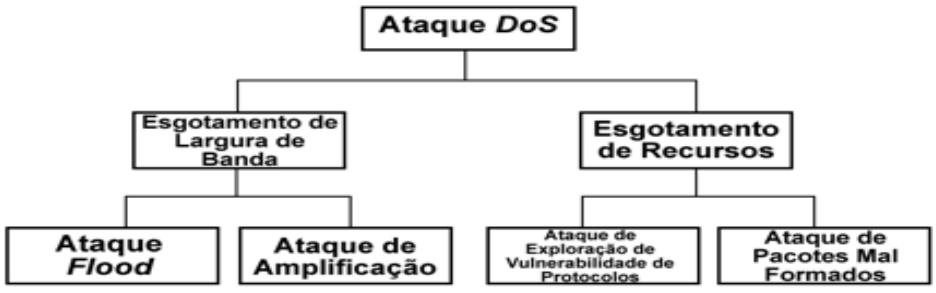

Figura 6 - Modelo de ataque DoS proposto (KUMAR e SELVAKUMAR, 2009)

Ataques DoS do tipo de esgotamento de largura de banda são provocados por atacantes que inundam o *host* destino através de requisições inválidas, com o objetivo de não permitir a comunicação de requisições normais (KUMAR e SELVAKUMAR, 2009).

Existem diversos tipos de ataques de esgotamento de largura de banda como, por exemplo, *flood* UDP (Postel, 1980) e ICMP (Postel, 1981), e ataques de amplificação como, *Smurf* (Specht, 2004) e *Fraggle* (Specht, 2004). Ataques DoS do tipo *flood* inundam o *host* a ser atacado com mais pacotes que o *host* consegue lidar, ocasionando na negação de serviço das aplicações legítimas que funcionam no *host* atacado. Ataques DoS de amplificação multiplicam o número de pacotes enviados ao *host* a ser atacado utilizando-se de *hosts* legítimos que possuam falhas de segurança, tornando o ataque mais efetivo (AZEVEDO, 2012).

Já os ataques DoS da categoria de esgotamento de recursos procuram explorar vulnerabilidades em protocolos e serviços ocasionando a negação de serviço (Kumar e Selvakumar, 2009). Nesta categoria de ataque DoS, são enviados para o *host* a ser atacado pacotes que explorem alguma vulnerabilidade da pilha TCP/IP ou de algum serviço específico, como por exemplo: TCP-SYN (Eddy, 2007), PUSH ACK (Specht ,2004), *apache2* (Apache 2011), IIS (Microsoft, 2013).

3.3.2 Evasão

Segundo Vaz (2004), este tipo de ataque age geralmente para obter informações de configuração em servidores remotos. Seu principal objetivo é “enganar” o IDS através pacotes gerados com perdas de informações para a detecção. Exemplos: *ataques de Case sensitive e Session Splicing.*

3.3.3 Inserção

Este tipo de ataque é comparável ao de evasão, porém utiliza o método de enviar mais informações ao IDS, dificultando a análise e consequentemente a detecção. Exemplos: *Long URL.*

3.3.4. Varredura de portas

Varredura de portas é o método geralmente utilizado nos primeiros passos de um ataque. Ferramentas de varredura, também chamadas de *scanners* de rede, retornam informações sobre os serviços que estão rodando no alvo (VAZ, 2004). Podemos citar como métodos de varredura: *Fingerprint, TCP Connect, Ident, SYN scanning, FIN scanning.*

4 FERRAMENTAS DE DETECÇÃO DE INTRUSÃO

Para desenvolvimento desse trabalho usou-se dois Sistemas de Detecção de Intrusão, sendo o *Snort* e o *Suricata* que serão apresentados abaixo.

4.1 *Snort*

O *Snort* é um sistema de detecção e prevenção de intrusos de código fonte aberto, baseado em rede (NIDS). Seu criador foi Martin Roesch, que disponibilizou sua primeira versão de testes no ano de 1999, se tornando bastante popular pela flexibilidade nas configurações de regras e atualizações comparadas *c*om outras ferramentas disponíveis (SNORT, 2013).

4.1.1 Apresentação e história do *Snort*

Esse NIDS foi desenvolvido na linguagem C, baseando-se na biblioteca de programação *Libpcap,* que faz a captura dos pacotes de rede. Inicialmente seu criador Roesch queria apenas uma ferramenta *Sniffing* melhor para o sistema operacional *Linux* que as existentes como o *tcpdump* (ferramenta nativa no *Unix* que captura pacotes).

Depois de ter seus objetivos concluídos na captura de pacotes, Roesch começou a desenvolver o *Snort* com função de IDS, pois não possuíam *softwares* livres com essas funções. Em 1999 consegue habilitar o *Snort* para comparar o tráfego de rede com as regras que ele definiu, sendo testada pela Comunidade Científica *Open Source*, passando por diversas atualizações tornando-se a mesma adaptável a vários ambientes (multiplataforma). Hoje a versão mais atual do *Snort* é a 2.9.x que pode ser instalada em vários sistemas operacionais, os quais desempenhando suas funções normalmente (SNORT, 2010).

O logotipo desse sistema de detecção de intrusão é um “*Porco*”, faz essa comparação com o animal, pois fareja e captura todos os pacotes que estão passando na rede, comparando com seu conjunto de regras (banco de dados) existente.

A ferramenta *Snort* é bastante utilizada para detecção de intrusão, possuindo um desempenho bom e uma linguagem flexível para descrição de ataques. Baseia-se na aplicação de regras de filtragem em cada pacote, a fim de procurar assinaturas de ataques conhecidos. Uma limitação dessa ferramenta se refere ao fato de apenas procurar assinaturas de ataques em cada pacote, ou em um fluxo, sem conseguir detectar ataques que necessitem de pacotes de protocolos diversos para serem caracterizados.

4.1.2 Ambiente e mecanismos de detecção

O *Snort* pode ser instalado tanto no sistema operacional *Unix* e suas derivações quanto no Windows. Necessita da instalação da biblioteca *Libcap* no *Linux* e *Winpcap* no sistema operacional Windows, pois para o *Snort* funcionar ele necessita dessa biblioteca, podendo deixar a interface de rede em modo promíscuo, ou seja, capturando pacotes destinados a ela, bem como os pacotes destinados a outros *hosts* da rede.

Um componente importante para o *Snort* são os mecanismos de detecção, todos os dados provenientes dos pré-processadores são verificados através de um conjunto de regras (assinaturas), estas baseadas em um texto que normalmente em uma subdiretoria onde está instalado o *Snort*, constituída por duas partes: o cabeçalho e as opções. Exemplo: ficheiro "*.rules*" é onde ficam todas as regras referentes a ataques DoS.

Existem cinco regras definindo o cabeçalho:

1- *Activation*: Alerta e chama regra do tipo dinâmica "*dynamic";*
2- *Dynamic* – permanece inativa até ser ativado por uma regra *activate*, registrando o tráfego;
3- *Alert* – Gera um alerta usando um método selecionado e então registra os pacotes e os dados;
4- *Pass* – ignora os pacotes;
5- *Log* – Registra e não alerta;

Para essas regras existe uma lista de protocolos suportados pelo *Snort*: protocolo TCP (SNMP, HTTP, FTP), UDP (DNS, SNMP, DHCP, RIP), ICMP (*Traceroute, ping*), IP (ICMP);

4.1.3 Regras

Como os vírus, as intrusões em redes têm determinados tipos de assinatura, essas usadas para criar regras do *Snort.* Podem ser usados *honeypots* (potes de mel), para descobrir o que os intrusos estão a fazer e quais as técnicas a utilizar. Essas assinaturas podem estar presentes em partes do cabeçalho de um pacote ou mesmo nos dados.

As regras do S*nort (rules)* operam na camada de rede (IP) e protocolos da camada de transporte (*TCP/UDP*) também existindo métodos para detectar anomalias nos protocolos da camada de aplicação.

As versões mais recentes do *Snort* fazem análise na camada de aplicação além da camada de rede e de transporte, onde as regras são aplicadas em ordem em todos os pacotes independente do seu tipo e/ou tamanho.

Conforme Caruso (2005) as regras do *Snort* são escritas em formato texto em uma única linha, e constituem-se de duas sessões: sendo cabeçalho e opções ou quando em linhas extensas são quebradas pelo uso de caracteres de concatenação ("\"), essa regras ficam armazenadas no ficheiro "*snort.conf".* Podem ser usadas de diversas formas como: para gerar uma mensagem de alerta ou para fazer um registro de um evento.

Para entendermos melhor as regras deixamos em destaque e enumeradas algumas partes do *logs* gerado por um ataque FTP em uma máquina *Linux* de forma a explicar seu funcionamento:

Exemplo de um cabeçalho de *Log*:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21
```

***alert* -** Este é o formato utilizado para a saída. Os formatos de saída possíveis são: *alert*, *log*, *pass*, *dynamic* e *activate*. O formato de saída é utilizado pelo *Snort* para classificar e separar as regras em cinco categorias principais (cinco *cadeias* de regras).

TCP - Esta parte da sintaxe define o protocolo em uso; neste caso, TCP. Este campo pode aceitar os valores: TCP, UDP, IP e ICMP. Para cada uma das cadeias de saída citadas no item anterior, o Snort cria 4 novas cadeias, uma para cada tipo de protocolo aceito. Existirá, por exemplo, uma árvore de regras TCP para cada uma das cadeias de formato de saída.

$EXTERNAL_NET - Esta parte da sintaxe é o endereço IP de origem.

any. Esta é a porta de origem selecionada como qualquer porta de origem.

=> Esta seta indica a direção do fluxo de dados; neste caso, de $EXTERNAL_NET vindo de qualquer porta para $HOME_NET na porta 21.

$HOME_NET - A variável HOME_NET foi utilizada para representar os endereços de destino abarcados pela regra.

21 - É a porta destino, indicando potenciais ataques à porta 21, que é a porta tipicamente utilizada para atividades FTP.

Opções das regras:

```
(msg: "FTP EXPLOIT wu-ftpd 2.6.0 site exec format string overflow Linux";
flow:to_server, established; content:"|31c031db31c9b046cd8031c031db|";
reference:bugtraq,1187; reference:cve,CAN-2000-0573; reference:
arachnids,287;
classtype: attempted_admin; sid:344; rev:4;
```

msg "FTP EXPLOIT wu-ftpd 2.6.0 site exec format string overflow Linux". Esta é a mensagem exibida pelo alerta.

flow: to_server, established. O *Snort* contém palavras chave que ligam a módulos (ou "*plugins*") de detecção na seção de opções das regras. A opção *flow* é um apontador para os módulos de detecção cliente/servidor. Os módulos cliente-servidor ligam-se ao pré-processador para verificar se o pacote é parte de uma sessão estabelecida.

content "|31 c0 31 bd 31 c9 b0 46 cd 80 31 c0 31 db|". Se o pacote pertence a uma sessão estabelecida, o *Snort* irá tomar o conteúdo indicado e tentará compará-lo com pacote em análise usando um algoritmo de comparação de cadeias de caracteres.

Reference. Esta palavra chave permite incluir referências à informação de identificação de ataques provida por terceiros; por exemplo, *URLs* para "*Bugtraq",* "McAfee", e os códigos de fabricantes ou identificação dos vendedores.

Classtype: attempted_admin. Existe uma classificação dos ataques para permitir aos usuários entenderem rapidamente e priorizar cada ataque. Cada classificação tem uma prioridade, que permite ao usuário priorizar quais eventos ele busca por meio de um apenas um número: 1 para Alto, 2 para Médio e 3 para Baixo, sendo essas classificação não usadas nas definições das regras.

Sid: 344. Este é o identificador único para a regra do *Snort.* Todas as regras do Snort têm um número único de identificação. Informação sobre a regra pode ser

verificada via www.snort.org/snort-db. O SID é também utilizado por programas de relatórios para identificar as regras.

Rev: 4 Esta seção das opções refere-se ao número de versão para a regra. Quando as regras do *Snort* são propostas pela comunidade "*open-source*", as regras passam por um processo de revisão. Ao longo do tempo, este processo permite às regras serem refinadas e a evitar falsos positivos.

4.2 *Suricata*

O Sistema de Detecção de Intrusão *Suricata* é mais recente e menos difundido comparando com o *Snort.* Este sistema foi desenvolvido pelo OISF *(Open Information Security Foundation)***,** também é baseado na detecção por assinaturas, disponibilizada para download em 01 de janeiro de 2010. Este motor incorpora um normalizador *HTTP (Hypertext Transmission Protocol)* e analisador (biblioteca HTTP), que fornece processamento avançado de fluxos *HTTP*, permitindo a compreensão do tráfego na camada 7 do modelo OSI (SURICATA, 2013).

Esse IDS é um *software Open Source* da nova geração de mecanismos de prevenção de intrusão, que não pretende apenas em substituir ou emular as ferramentas existentes no setor, mas sim trazer novas ideias e tecnologias. Utiliza regras baseadas em NIDS, também desenvolvidas externamente definidas para monitorar o tráfego de rede e fornecer alertas para o administrador do sistema quando eventos suspeitos são detectados.

4.2.1 Funcionamento

Projetado para ser compatível com componentes de segurança de rede existente, *Suricata* apresenta uma ferramenta unificada de funcionalidade de saída e opções de bibliotecas conectáveis para aceitar chamadas de outros aplicativos (LIMA, 2012).

Como é um mecanismo de *multithread*, *Suricata* oferece maior velocidade e eficiência na análise de tráfego de rede. Além de aceleração de hardware, o motor é construído para utilizar o poder de processamento oferecido pelos mais recentes processadores *multi-core*.

Ao envolver a comunidade de código aberto, a *OISF* construiu o *software Suricata* para simplificar o processo de manter os níveis de segurança ideais. Através de parcerias estratégicas, *OISF* está aproveitando a experiência de ameaças emergentes e outros recursos de destaque na indústria para fornecer as regras mais atuais e abrangentes.

4.2.2 Vantagens no uso

O sistema *Suricata,* disponível sob a versão 2 da Licença Pública Geral, elimina preocupações com custos de software adicionais, proporcionando uma opção escalável para às arquiteturas de segurança mais complexas da rede.

Conforme OISF (2008) existem três razões para testar a ferramenta *Suricata*:

1. Altamente escalável: *Suricata* é *multi-threaded* (equilibra a carga de processamento em cada processador em um sensor), permitindo que o *hardware commodity* alcance velocidades de até 10 *Gigabit* sobre o tráfego da vida real, sem sacrificar a cobertura do conjunto de regras.

2. Protocolo de Identificação: protocolos mais comuns são automaticamente reconhecidos pelo *Suricata.* Além disso, graças a palavras-chave dedicadas, pode combinar em campos que vão desde protocolo *http* para um identificador de certificado *SSL*.

3. Identificação e extração de arquivos: esse *software* pode identificar milhares de tipos de arquivos ao cruzar uma rede. Pode-se também marcar para a extração, e assim o arquivo será gravado em disco como arquivo de metadados que descreve a situação de captura e fluxo.

O motor fornece ainda as seguintes funcionalidades: opções de idioma (regra), saída unificada permitindo a interação com sistemas externos de registro e gestão, *IPv6*, baseado em regras de reputação de IP, biblioteca ficha-capacidade de interação com outras aplicações e a disponibilização de estatísticas de desempenho.

4.3 Base de Dados DARPA

A DARPA (*Defense Advanced Research Projects Agency*) em 1998 reconheceu a capacidade de realizar avaliações quantitativas em sistema de detecção de intrusão, contratando e financiando a realização dessas avaliações no Centro de Pesquisas *Lincoln Labs do MIT (Massachussetts Institute of Technology).*

Localizado em *Massachusetts,* trabalhou juntamente com o Laboratório de Pesquisas da Força Aérea, em Nova Iorque, com o objetivo de construir um conjunto de dados para futuras avaliações. A construção foi de forma simples, fazendo uma rede de uma base da Força Aérea conectada a internet, produzindo atividades de em *scripts* e ataques injetados em pontos definidos por um período de sete semanas (BRUGGER, 2005).

Os dados de rede foram capturados no período das 8 horas da manhã até às 6 horas da manhã do dia seguinte, ou seja, contendo 22 horas de tráfego capturado em cada dia. Foram coletados os dados durante cinco semanas, onde cada semana contém 5 dias de tráfego registrados, armazenando no formato de dados da ferramenta tcpdump, contendo aproximadamente 9 GB de dados, conforme a Tabela 1 que mostra abaixo os horários.

Tabela 1: **Horário das coletas da segunda semana de 1999.**

Começo da coleta			**Fim da coleta**		
Dia da semana	Data	Horário	Dia da semana	Data	Horário
Segunda	08 março	08:00	Terça	09 março	06:00
Terça	09 março	08:00	Quarta	10 março	02:59
Quarta	10 março	08:00	Quinta	11 março	06:00
Quinta	11 março	08:00	Sexta	12 março	06:00
Sexta	12 março	08:00	Sábado	13 março	06:00

Fonte: DARPA, 2013.

Depois de feitas as avaliações por pesquisadores e membros da comunidade, os mesmos forneceram opiniões e melhorias sobre essas avaliações, surgiu assim uma nova avaliação em 1999, com uso de ataques mais oportunos, incluindo *logs* para o *Windows*, política de segurança para a rede de destino e teste com ataques mais recentes.

As três primeiras semanas de tráfego são destinadas a treinamento, pois possuem documentação dos ataques existentes no período. A primeira e terceira semana não possuem ataques. A partir disso, neste trabalho estabeleceu-se a base de ataques DARPA da segunda semana de 1999 a qual possui ataques definidos na tabela 2, que foi usada para comparar o funcionamento e detecções realizadas pelo *Snort e Suricata*.

A DARPA (2013) disponibiliza esses bancos de dados em arquivos de entrada e saída de cada dia da semana, os quais serão comparados com as assinaturas com as existentes nas configurações dos NIDS selecionados neste trabalho.

Segundo Brugger (2005), a sua análise indicou que o conjunto de dados que detectam ataques do NIDS *Snort* tem problemas para detectar, pois o mesmo pode detectar apenas uma parte das conexões em diversos tipos de ataques. Isso leva a concluir que o conjunto de ataques da DARPA é útil para testar sistemas de detecção, mas não suficiente para demonstrar as capacidades de um IDS avançado.

Nos dados de 1999 temos: a adição de uma estação de trabalho *Windows NT* como uma vítima, a adição de um *tcpdump sniffer* dentro da máquina, entre outros, também ataques furtivos foram adicionados devido à ênfase em atacantes sofisticados que pode criar ataques cuidadosamente para se parecer com o tráfego normal (LIPPMANN et al. 2000).

A Tabela 2 demonstra ataques que ocorrem na segunda semana de dados 1999. A data, a horário e origem de cada ataque é fornecida. Além disso, o nome do ataque é proporcionado como uma fonte de identificação.

Tabela 2: **Descrição de ataques detectados pela DARPA na segunda semana de 1999**.

ID	DATA	HORÁRIO	ORIGEM	NOME
1	03/08/1999	08:01:01	hume.eyrie.af.mil	NTinfoscan
2	03/08/1999	08:50:15	zeno.eyrie.af.mil	pod
3	03/08/1999	09:39:16	marx.eyrie.af.mil	back
4	03/08/1999	12:09:18	pascal.eyrie.af.mil	httptunnel
5	03/08/1999	15:57:15	pascal.eyrie.af.mil	land
6	03/08/1999	17:27:13	marx.eyrie.af.mil	secret
7	03/08/1999	19:09:17	pascal.eyrie.af.mil	Ps attack
8	03/09/1999	08:44:17	marx.eyrie.af.mil	portsweep
9	03/09/1999	09:43:51	pascal.eyrie.af.mil	eject
10	03/09/1999	10:06:43	marx.eyrie.af.mil	back
11	03/09/1999	10:54:19	zeno.eyrie.af.mil	loadmodule
12	03/09/1999	11:49:13	pascal.eyrie.af.mil	secret
13	03/09/1999	4:25:16	pascal.eyrie.af.mil	mailbomb
14	03/09/1999	13:05:10	172.016.112.001-114.254	ipsweep

15	03/09/1999	16:11:15	marx.eyrie.af.mil	phf
16	03/09/1999	18:06:17	pascal.eyrie.af.mil	httptunnel
17	03/10/1999	12:02:13	marx.eyrie.af.mil	satan
18	03/10/1999	13:44:18	pascal.eyrie.af.mil	mailbomb
19	03/10/1999	15:25:18	marx.eyrie.af.mil	perl (Failed)
20	03/10/1999	20:17:10	172.016.112.001-114.254	ipsweep
21	03/10/1999	23:23:00	pascal.eyrie.af.mil	eject (console)
22	03/10/1999	23:56:14	hume.eyrie.af.mil	crashiis
23	03/11/1999	08:04:17	hume.eyrie.af.mil	crashiis
24	03/11/1999	09:33:17	marx.eyrie.af.mil	satan
25	03/11/1999	10:50:11	marx.eyrie.af.mil	portsweep
26	03/11/1999	11:04:16	pigeon.eyrie.af.mil	neptune
27	03/11/1999	12:57:13	marx.eyrie.af.mil	secret
28	03/11/1999	14:25:17	marx.eyrie.af.mil	perl
29	03/11/1999	15:47:15	pascal.eyrie.af.mil	land
30	03/11/1999	16:36:10	172.016.112.001-254	ipsweep
31	03/11/1999	19:16:18	pascal.eyrie.af.mil	ftp-write
32	03/12/1999	08:07:17	marx.eyrie.af.mil	phf
33	03/12/1999	08:10:40	marx.eyrie.af.mil	perl (console)
34	03/12/1999	08:16:46	pascal.eyrie.af.mil	ps (console)
35	03/12/1999	09:18:15	duck.eyrie.af.mil	pod
36	03/12/1999	11:20:15	marx.eyrie.af.mil	neptune
37	03/12/1999	12:40:12	hume.eyrie.af.mil	crashiis
38	03/12/1999	13:12:17	zeno.eyrie.af.mil l	loadmodule
39	03/12/1999	14:06:17	marx.eyrie.af.mil	perl (Failed)
40	03/12/1999	14:24:18	pascal.eyrie.af.mil	ps
41	03/12/1999	15:24:16	pascal.eyrie.af.mil	eject
42	03/12/1999	17:13:10	pascal.eyrie.af.mil	portsweep
43	03/12/1999	17:43:18	pascal.eyrie.af.mil	ftp-write

Fonte: DARPA, 2013.

As descrições dos ataques detectados são fornecidos a seguir e, os nomes fornecidos são aqueles que foram utilizados durante a avaliação e podem não ser os únicos nomes pelos quais um ataque é conhecido (DARPA, 2013).

Back Ataque de negação de serviço contra o servidor *web apache*, onde um cliente solicita uma URL contendo muitas barras invertidas;

Crashiis Um pedido *http* único, faz com que o servidor mal formado falhe;

Eject Buffer overflow através do programa *eject* no *Solaris*. Leva a uma transição usuário *root,* se bem sucedida;

ftp-write Usuário FTP remoto cria arquivo. *Rhost* no mundo gravável diretório FTP anônimo e obtém *login* local;

httptunnel Existem duas fases para este ataque: *Configuração* - um "cliente" web está configurado na máquina que está sendo atacada, o que está configurado, talvez via *crontab*, para fazer periodicamente pedidos de um *"servidor"* rodando em uma porta que não possui privilégios no ataque da máquina. *Ação* - Quando os pedidos são periódico recebido, o servidor encapsula comandos a serem executados pelo "cliente" em um *cookie,* coisas como *"cat /etc/passwd"*, etc;

Ipsweep Faz varredura, vigilância, varredura de porta, ataca com *ping* em vários endereços de host;

Land Negação de serviço em que um host remoto é enviado em um pacote UDP com a mesma origem e destino;

Loadmodule Ataque *loadmodule* não furtivo que redefine IFS para um usuário normal e cria um shell de root;

Mailbomb Um ataque de negação de serviço em que o servidor de correio enviar várias mensagens para a entrega, a fim de retardá-lo, talvez efetivamente travar o funcionamento normal;

Neptune SYN Flood negação de serviço em um ou mais portos;

Ntinfoscan Um processo pelo qual o atacante verifica uma máquina NT para obter informações sobre sua configuração, incluindo os serviços *ftp, telne*t serviços, serviços web, informações sobre a conta do sistema, sistemas de arquivos e permissões;

Phf Script CGI explorável que permite que um cliente para executar comandos arbitrários em uma máquina com um servidor web configurada incorretamente;

Pod Negação de *ping* da morte serviço;

Portsweep Varreduras através de muitas portas para determinar quais serviços são suportados em um único *host;*

Ps	*Ps* se aproveita de uma *racecondition* no comando *ps* em Sol. 2.5, permitindo que um usuário para ter acesso *root;*
Satan	Ferramenta que procura por pontos fracos conhecidos.
Secret	Ataques secretos;
Teardrop	Negação de serviço em que os pacotes UDP fragmentado causar alguns sistemas para reiniciar;

5 PROCEDIMENTOS METODOLÓGICOS

A presente pesquisa classifica-se quanto à natureza como qualitativa e quantitativa. A pesquisa qualitativa de acordo com Malhotra (2012) é uma metodologia de pesquisa não estruturada, fundamentada em amostras pequenas, que adéqua *insights* e o entendimento do contexto do problema. Para Fachin (2006), a pesquisa quantitativa não deve ser realizada sem embasamento, sem uma contagem numérica que identifique os números que proporcionam os resultados finais eficazes, tendo em vista que engloba um sistema lógico a ser observado.

Quanto aos objetivos, a pesquisa caracteriza-se como exploratória e descritiva. A pesquisa exploratória para Gil (2010) é classificada por adequar o máximo de familiaridade com o problema, tem como meta o aprimoramento de ideias, seu planejamento é flexível. Define a pesquisa descritiva como sendo aquela que possui como objetivo a descrição das características de determinada população ou fenômeno, ou aquela que estabelece relações entre variáveis.

No que tange os procedimentos técnicos, este trabalho caracteriza-se como um estudo de caso, conforme Diehl e Tatim (2004), é um estudo aprofundado e exaustivo de poucos objetos descrevendo uma instituição, ou comunidade, isto é, dando ênfase na totalidade e a simplicidade dos procedimentos.

Os ataques foram adquiridos de um banco de dados sintéticos da DARPA da segunda semana de 1999 colocou-se em funcionamento com os dois NIDS: *Snort* e *Suricata* a fim de comparação de assinaturas existentes em suas regras para verificação de resultados de ataques encontrados ou não, assim fazendo a análise e conclusões sobre o trabalho.

5.1 Ataques Sintéticos da DARPA

Para um melhor estudo dos ataques foi usado à avaliação da segunda semana de 1999 da DARPA, conforme trata no Capítulo 4 (4.3 Base de Dados DARPA). A análise baseou-se em uma avaliação *off-line*, pois utilizou os dados e fez-se uma comparação com regras dos sistemas para geração de alertas e por fim comparação com os ataques estabelecidos pela DARPA. A segunda semana de ataques de 1999 traz melhorias comparadas com dados coletados anteriormente em

1998, pois possui mais extensões para melhorar a análise e cobrir mais tipos de ataque.

5.2 Requisitos e instalação do *Snort*

A instalação do *Snort* foi feita no *Linux Ubunbu* versão 12.04 em uma máquina virtualizada com os seguintes complementos: *ACIDBASE* (*Analysis Console for Intrusion Detection*) servindo como interface gráfica para visualização dos ataques, *Apache2*, *Mysql-server* e PHP5. Foi escolhido o *Ubuntu* devido suas funcionalidades e também por ser um *software* livre e gratuito. Essa máquina foi configurada com 2 GB de memória RAM e alocada 40 GB para armazenamentos das informações, após feitas as instalações atualizou-se o sistema.

Para trabalhar-se com o *ACIDBASE* no monitoramento de *logs* torna-se obrigatório o uso de um Sistema de Gerencia de Banco de Dados (SGBD), sendo o *MySQL* mais usado*,* também o uso de um servidor *WEB* como o *Apache* com suporte a PHP, devido os *scripts* do *ACIDBASE* serem escritos em PHP*.*

Comando para rodar arquivos sintéticos de *logs* da DARPA de 1999 para verificação dos ataques comparando com regras do *snort*:

```
Snort –r arquivo.tcpdump –c /etc/snort/snort.conf
```

Após feita as instalações devidas acima, verificamos as configuração das regras para identificar ataques, essas localizadas em um arquivo */etc/snort/snort.conf.* Os alertas gerados (*logs)* de ataques se localizavam no diretório */var/log/snort/alert.*

5.3 Instalação e regras do Suricata

A ferramenta *Suricata* é implementada em uma linguagem completa de assinaturas para coincidir com as ameaças conhecidas, violações de políticas e comportamentos maliciosos, também detecta muitas anomalias no tráfego que inspeciona.

A instalação foi feita de forma simples adicionando as bibliotecas importantes para seu funcionamento como o *libpcap*, igualmente ao *Snort* onde realizou-se a instalação em uma máquina virtualizada usando com *Xubuntu* 12.04.

Foi escolhido o *Xubuntu* devido a possuir já instalado as dependências da ferramenta *Suricata* como o *Libpcap* entre outras, essa máquina foi configurada com 2 GB de memória RAM e alocada 40 GB para armazenamentos das informações e geração de *logs*, após feitas as instalações atualizou-se o sistema.

A configuração e verificação das regras para identificar ataques ficam localizadas no diretório */etc/suricata/suricata.yaml.*

Após a comparação de regras do *suricata* com dados da DARPA pelo comando no terminal: *suricata –r arquivo.tcpdump –c /etc/suricata/suricata.yaml*. Os *logs* de ataques foram gerados, assim criando um diretório */var/log/suricata/fast.log.*

5.4 Trabalhos relacionados

Para realização deste trabalho foram levados em consideração todos os protocolos de rede (*IP, TCP, ICMP, UDP,* entre outros), onde ocorrem detecções devidas a ataques considerados usando uma base sintética da DARPA (*Defense Advanced Research Projects Agency*) que possui ataques e dados de monitoramento da rede durante a segunda semana de 1999.

A implantação da leitura de entrada de dados foi gerada através de informações contidas nas bases de dados da DARPA comparando com as detecções efetuadas pelos NIDS *Snort e Suricata*, esses sendo *softwares* destinados à segurança e avaliação de ataques da rede.

A análise dos resultados com trabalhos relacionados é uma tarefa difícil, devido à utilização de diversas bases de dados, diferentes modos de extração e comparação desses dados nos dois Sistemas de Detecção de Intrusão (*Snort e Suricata*). Por este motivo foram efetuadas comparações com trabalhos que utilizaram essas ferramentas comparando com dados sintéticos de ataques.

Conforme Ribeiro (2005) todo *software* está sujeito a falhas e deve-se ficar constantemente atento aos boletins de segurança, atualizações e vulnerabilidades descobertas.

Para Vaz (2004) há algumas limitações nos IDS livres, tais como as altas taxas de falsos alertas, dificuldade de gerência e a impossibilidade de detecção de alguns ataques mais sofisticados. Diante deste cenário foram propostas outras ferramentas que amenizar tais deficiências agindo essencialmente sobre concentração de mensagens e correlação de eventos distribuídos.

Tanto o *Snort* quanto o *Suricata* são Sistemas de Detecção muito capazes, cada um com pontos fracos e fortes. Testes realizados com dados semelhantes, para fornecer uma recomendação informada ao departamento de Tecnologia de Informação da Naval Escola de Pós-graduação em se usar *Suricata* como uma camada adicional de defesa a Rede de Pesquisa Educacional. Ambas as ferramentas tiveram um bom desempenho durante os testes, tendo alguns falsos positivos e falsos negativos que podem ser atribuídos à debilidade do conjunto de regras utilizados para os testes (ALBIN 2011).

Conforme Brugger (2005) os dados da DARPA são bastante usados e úteis para testar Sistemas de Detecção em Redes de Computadores, em que o desempenho não favorável é uma condição necessária, mas não o suficiente para demonstrar as capacidades de IDS avançadas, não se pode concluir que um NIDS é ineficaz, pois ele faz o que foi projetado para fazer (detecção baseada em assinaturas), precisamos de mais conjuntos de dados recentes de ataques para testar IDS.

6 RESULTADOS E DISCUSSÕES

O *Snort* e *Suricata* são Sistemas de Detecção de Intrusão baseados em assinaturas, ambos gratuitos e multiplataformas, sendo que o *Snort* encontra-se consolidado no mercado há vários anos e com muitas versões contendo atualizações de regras, já o *Suricata* surgiu em 2010 como um novo motor inovador e com a tecnologia *multithreading* (processa várias tarefas ao mesmo tempo), devido a isso que se buscou a comparação dessas duas ferramentas com ataques da DARPA.

Os dados de sintéticos de ataques fornecidos na segunda semana de 1999 pela DARPA (DARPA, 2013), trazem arquivos com ataques, assim podendo validar ou não esses ataques comparando com as regras dessas duas ferramentas.

Durante o desenvolvimento deste trabalho foram encontrados vários desafios que, ao mesmo tempo, tornaram a jornada mais difícil como configurações nas instalações e monitoramento dessas ferramentas de detecção de intrusão. Verifica-se que o NIDS *Snort* como é um sistema, consolidado e bastante usado por empresas e órgãos governamentais, possui bastante material disponível para tirar dúvidas, já o *Suricata* como é um motor lançado mais recentemente possui pouco material e experimentos que no qual foi utilizado.

O entendimento detalhado da parte teórica faz parte do trabalho, e é de fundamental importância, pois sem essa a instalação dos Sistemas não se tornaria possível no desenvolvimento do mesmo.

Quando comparadas as assinaturas (regras) desses dois NIDS: *Snort* e *Suricata* com os dados da segunda semana da DARPA de 1999 buscou-se a detecção dos ataques referenciados nessa base de dados.

Com esses experimentos pode-se verificar que o número de *logs* gerados no *Snort* foi 73.408, já o *Suricata* apresentou 43.428 *logs,* evidenciando um número significativo. Também se verificou que os 43 alertas evidenciados nos dados da DARPA não apareceram nos *logs* gerados pelos Sistemas*.* A partir dessas informações verificou-se que o *Snort* gerou 41% a mais que o *Suricata* de falsos positivos (tráfego normais identificados como ataques), e que nenhum dos dois Sistemas identificou ataques, que possuíam na base de dados da DARPA,

evidenciando assim como falsos negativos (ataques reais que não foram detectados).

A atualização das regras dos dois sistemas é disponibilizada frequentemente devido ao crescente avanço de ataques. Assim, houve dificuldades em comparar dados de 1999 com regras atuais, podendo ser umas das causas da geração de grande número de falsos positivos e falsos negativos.

Os resultados analisados por Brugger (2005), que avalia ataques comparando dados da DARPA com o IDS *Snort,* quando se achava que o mesmo teria resultados positivos, trás resultados negativos, estes podendo ser a uma falha no conjunto de dados para modelar os ataques corretamente, ou devido os ataques da DARPA já serem ultrapassados.

Esses resultados salientam que a utilização do banco de dados da DARPA de 1999, para experimentos de detecção de intrusão são úteis para diferentes modalidades de NIDS, sendo de fundamental importância, referindo-se a suas utilidades e a maneira e comportamento das detecções absorvidas, mas notou-se que esses dados, devido a serem muitos antigos, trazem problemas quando comparados com regras de ataques atuais.

Segundo Brugger (2005), a comunidade de detecção de intrusão precisa de um conjunto de dados mais realista e atuais para avaliação de falsos positivos e falsos negativos. Há ameaças que surgiram recentemente como: *botnets*, *spyware*s, *flash*, vermes entre outros, que IDS, são capazes de detectar e que devem ser testados em relação ao conjunto de dados.

Com isso chega-se à conclusão de que os dados Sintéticos da DARPA de 1999 não possibilitam uma análise formal comparando com assinaturas recentes de Sistemas de Detecção de Intrusão, salientando que os mesmos são ultrapassados e que ataques surgem frequentemente em nosso dia-a-dia, evidenciando regras novas nesses sistemas.

Este estudo possibilitou também afirmar que esses dois NIDS: *Snort e Suricata* funcionam melhor na detecção do tráfego na rede em tempo real analisando os pacotes que passam pela mesma, devido a possuírem regras com assinaturas atualizadas.

7 CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS

A Internet está no caminho de tornar-se o meio de comunicação para todos os tipos de informações, desde a simples transferência de arquivos de computadores até a transmissão de voz, vídeo ou informações interativas em tempo real (AZEVEDO, 2012), dessa maneira as vulnerabilidades em redes e aplicações cresce na mesma proporção. Visto isto, o trabalho percorreu caminhos que envolveram as concepções teóricas e práticas de Sistemas de Detecção de Intrusão em Redes de Computadores analisando duas ferramentas existentes e usadas para esses fins de monitoramento e geração de alertas.

Desse modo, o encaminhamento das fases da pesquisa nos remeteu a análises e levantamentos de questões que envolvem a assuntos estudados. Pesquisar e desenvolver um assunto exige reflexões tanto teóricas, metodológicas e práticas que vão surgindo durante a trajetória da pesquisa, sendo fundamental a persistência e a coerência buscando atender os objetivos propostos.

As características gerais desses dois NIDS são semelhantes sendo que as regras podem ser suportadas em ambos, nesse sentido, pode-se dizer que a detecção de intrusão com uso de dados Sintéticos da DARPA deve trazer os mesmos resultados. Outro aspecto a ser considerado, refere-se ao surgimento desses Sistemas abordados sendo que o *Snort* está há mais tempo no mercado e com várias versões, já o *Suricata* é um sistema mais novo que busca explorar os recursos da máquina.

Ao final desse trabalho tomamos consciência de que a sua elaboração apresentou-se como um processo de crescimento tanto acadêmico quanto pessoal. Verificamos que a parte prática muitas vezes, parece simples e fácil, mas pode englobar problemas que dificultam um bom funcionamento dos Sistemas de Detecção de Intrusão. Desse modo, o fim de uma etapa, torna-se o começo de outra, onde outros trabalhos serão realizados sobre monitoramento em Redes de Computadores com considerações deste estudo.

Enfim, esperamos que este trabalho contribua para o desenvolvimento de estudos nessa área, que são importantes para os usuários finais os quais, muitas vezes, não tem conhecimento das vulnerabilidades e ataques que está sofrendo.

7.1 Sugestões para trabalhos futuros

A Detecção de Intrusão em Redes de Computadores é uma área de pesquisa que vem a crescer, pois as tentativas de comprometer as informações se tornam cada vez mais e mais complexas. Os Sistemas de Detecção em Redes vem a contribuir para análise e monitoramento da mesma, assim possibilitando desenvolvimento de vários trabalhos na área como:

- A análise de monitoramento de redes com tráfego significante de dados;
- Comparação de diferentes métodos de detecção de assinaturas trazendo suas eficiências;
- Incorporação de técnicas baseadas a anomalias na Detecção de Intrusão ainda desconhecidas;
- O uso dos dados sintéticos da DARPA comparando com outros Sistemas de Detecção em Redes de Computadores;

BIBLIOGRAFIAS

AMOROSO, E.G. **Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace-Back and Response.** Intrusion.Net Books, 1999.

ALBIN, Eugene. **A comparative analysis of the snort and Suricata intrusion-detection systems.** 2011.

APACHE 2011. **The Apache HTTP Server Project.** Disponível em: <http://httpd.apache.org>. Acesso em: 15 jun. 2013.

AZEVEDO, R. P. **Detecção de ataques de negação de serviço em redes de computadores através da transformada Wavelet 2D**. Dissertação de mestrado. UFSM, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS. 2012.

BADISHI, G.; KEIDAR, I., SASSON, A. **Exposing and eliminating vulnerabilities to denial of service attacks in secure gossip-based multicast. IEEE, Transactions on Dependable and Secure Computing**, 3:45–61, 2006.

BARBOSA A. **Sistemas de detecção de intrusão.** Disponível em: <http://www.lockabit.coppe.ufrj.br/downloads/academicos/IDS.pdf >. Acesso em: 19 nov. 2013.

BARFORD, P., KLINE, J., PLONKA, D., RON, A. **A signal analysis of network traffic anomalies.** In IMW ’02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, pages 71–82, New York, NY, USA. ACM. ,2002.

BERTHOLDO, Leandro Márcio; ANDREOLI, Andrey Vedana; TAROUCO, Liane M. R. **Gerência de Segurança Através do Uso de Netflow.** Disponível em: <http://www.rnp.br/_arquivo/wrnp2/2003/gsaun01a.pdf>. Acesso em: 18 dez. 2013.

BRUGGER S. T. **Uma avaliação da DARPA IDS Avaliação Data DataSet usando Snort (**An Assessment of the DARPA IDS Evaluation Dataset Using Snort), 2005.

CARDANA J. M. **Analisador comportamental de rede.** Dissertação de Mestrado em Informática pela Faculdade de Ciências de Lisboa Departamento de Informática. 2006.

CARUSO, L. C. M. **Proposta de arquitetura para NIDS acelerado por Hardwarwe.** Dissertação de mestrado da Pontifica Universidade Católica do RS, Porto Alegre 2005, p. 25.

CERT (Computer Emergency Response Team), **Denial of Service Attacks**. Disponível em: http://www.cert.org/tech_tips/denial_of_service.html. Acesso em: 17 out. 2013.

CERT.br: **Cartilha de segurança para Internet**, versão 4.0 /– São Paulo: Comit ˜ e Gestor da Internet no Brasil, 2012.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. **Anomaly detection: a survey**. ACM Computing Surveys, v.41, n.3, p.1–58, 2009.

CISCO, **Cisco. Netranger documentation**. Disponível em: <http://www.cisco.com/ univercd/cc/td/doc/product/iaabu/netrangr/> Acesso em: 10 out. 2011.

CRUZ, Frank. **Entendendo um sistema de detecção de invasões.** Disponível em: <http://www.timaster.com.br/revista/artigos/main_artigo.asp?codigo=100>. Acesso em: 28 out. 2013.

DARPA. **Defense Advanced Research Projets Agency**. Disponível em: <http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/>. Acesso em: 15 dez. 2013.

DIEHL, Astor Antonio; TATIM, Denise Carvalho. **Pesquisa em ciências sociais aplicadas: métodos e técnicas**. São Paulo: Prentice Hall, 2004.

DEMIRAY, Sadettin. **Improving misuse detection with neural networks**, Dissertação de Mestrado, Izmir Institute of Technology, Izmir - Turquia. 2005.

EDDY, W. **TCP SYN Flooding Attacks and Common Mitigations**. Disponível em: <http://tools.ietf.org/html/rfc4987> Acesso em: Jun. 2013.

FACHIN, Odília. **Fundamentos de metodologia.** 6. ed. São Paulo: Saraiva, 2006.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

HANDLEY. M. Internet Denial-of-Service Considerations **RFC 4732.** Disponível em: < http://tools.ietf.org/html/rfc4732>. Acesso em: 18 nov. 2013.

KIZZA, J. M. ***Guide to Computer Network Security*.** New York, NY: Springer, 2005.

KRUEGEL, C.; VIGNA, G. **Anomaly detection of web-based attacks**. In.: 10th ACM conference on Computer and communications security. *Proceedings*, New York, NY, USA: ACM, p. 251-261, 2003.

KUMAR, P. ;SELVAKUMAR, S. **Distributed denial-of-service (DDoS) threat in collaborative environment - a survey on DDoS attack tools and traceback mechanisms. Advance Computing Conference**. IACC. IEEE International, 2009.

LAUFER; Rafael P. **Introdução a sistemas de detecção de intrusão.** Rio de Janeiro, 2003. Disponível em: http://www.gta.ufrj.br/grad/03_1/sdi/index.htm. Acesso em: 18 nov. 2013.

LEMOS. R. **Web worm targets white house**. CNET News.com, Julho 2001. Disponível em: http://www.news.com/2100-1001-270272.html. Acesso em: 28 out. 2013.

LIMA, F. A. **Estudo do sistema de detecção de intrusão.** Curso de Especialização em Redes e Segurança de Sistemas Universidade Católica do Paraná. 2012.

LINDA, Ondrej; VOLLMER, Todd; MILOS. **Manic Neural network based intrusion detection system for critical infrastructures, *em* 'IJCNN'09, Int. Joint INNS-IEEE Conf. on Neural Networks'**, Atlanta, Georgia, USA, pp. 1827–1834. 2009.

LIPPMANN,R., HAINES J. W., FRIED D. J., KORBA J., **The 1999 DARPA Off-Line Intrusion Detection Evaluation.** 2000. Disponível em: <http://link.springer.com/chapter/10.1007%2F3-540-39945-3_11#page-1>. Acesso em: 15 dez. 2013.

MALHOTRA, N. **Pesquisa de marketing: uma orientação aplicada**. 6. ed. Porto Alegre: Bookman, 2012.

MICROSOFT. **The Oficial Microsoft IIS Site**. Disponível em: http://www.iis.net/. Acesso em: 15 out 2013.

MURINI, C. T. **Análise de Sistemas de Detecção de Intrusão em Redes de Computadores.** JAI UFSM (Jornada Acadêmica Integrada da Universidade Federal de Santa Maria). 2013.

MYCERT. **Malaysia Computer Emergency Response Team**. Disponível em: <http://www.mycert.org.my/en/services/advisories/mycert/2013/main/detail/931/index.html>. Acesso em: 08 set. 2013.

NAKAMURA, E. **Segurança em de redes em cooperativos**. São Paulo: Novatec, 2007.

NED, Frank. **Ferramentas de IDS.** Disponível em: <http://www.rnp.br/newsgen/9909/ids.html>. Acesso em: 28 out. 2013.

OISF. **Open Information Security Foundation. 2008.** Disponível em: <http://www.openinfosecfoundation.org/>. Acesso em: 07 dez. 2013.

PERLIN, T. **Um detector de anomalias de tráfego de rede baseado em Walvelets.** Dissertação de mestrado. UFSM, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS. 2010, p. 21.

PIETRO, Roberto D, MANCINI, Luigi V. **Intrusion Detection Systems.** Springer Publishing Company: Nova Iorque. 1st ed. 2008.

POSTEL, J. (1980). **RFC 768 - User Datagram Protocol**. Disponível em: <http://www.faqs.org/rfcs/rfc768.html> Acesso em: 03 dez. 2013.

POSTEL, J. (1981). **RFC 792 - Internet Control Message Protocol.** Disponível em: <http://www.faqs.org/rfcs/rfc792.html> Acesso em: 04 dez. 2013.

REHMAN, U. R. (2013). ***Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID.*** Disponível em: <http://ptgmedia.pearsoncmg.com/images/0131407333/downloads/0131407333.pdf> Acesso em: 11 nov. 2013.

RIBEIRO, B. **Detecção de intrusos usando o snort**. Monografia de Pós-graduação. Universidade Federal de Lavras – MS 2005.

ROESCH, M. **Snort - lightweight intrusion detection for networks**. In: LISA '99: Proceedings of the 13th USENIX conference on System administration, p. 229–238, Berkeley, CA, USA. USENIX Association, 1999.

SCARFONE, Karen; MELL, Peter. **Guide to intrusion detection and prevention systems (IDS)**, Relatório técnico, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2007

SECUNDADO, **Dados sobre internet no Brasil**, Disponível em: <http://www.secundados.com.br>. Acesso em: 11 Set. 2013.

SILVA COSTA. Guilherme., **Detecção de Intrusão em Redes de Computadores: Algoritmo Imunoinspirado Baseado na Teoria do Perigo e Células Dendríticas.** Universidade Federal de Minas Gerais,Dissertação de mestrado. 2009.

SILVA M. P.; SAMPAIO N. S. **Estudos de sistemas de detecção e prevenção de intrusões uma abordagem open Source.** 2006.

SNORT. **Snort Uses Manual**. [S. l.]. 2010. Disponível em: <http://www.snort.org/>. Acesso em: 22 nov. 2013.

SPECHT, S. M. (2004). **Distributed denial of service: taxonomies of attacks, tools and countermeasures.** Proceedings of the International Workshop on Security in Parallel and Distributed Systems, 2004, p. 543–550.

SUNDARAM, Aurobindo**. An introduction to intrusion detection**. Crossroads: The ACM Student Magazine, 2 ed. Abril, 1996.

SURICATA **Suricata-Vs_Bufo.** Disponível em: <http://www.aldeid.com/wiki/Suricata-vs-snort>. Acesso em: 25 nov. 2013.

VAZ T. B, **Sistemas de Detecção de Intrusão Livres: suas limitações e uma arquitetura proposta sobre concentração de mensagens e correlacionamento de eventos,** Universidade Federal da Bahia. 2004. Disponível em: <http://www.uefs.br/erbase2004/documentos/wticgbase/Wticgbase2004ArtigoIC005.pdf>. Acesso em: 12 dez. 2013.

WANG, J. **Computer network security: theory and practice**. Higher Education Press, 2009.

ANEXO 1 TUTORIAL DE INSTALAÇÃO DO *SNORT*

A ideia desse tutorial é demonstrar os comandos necessários para instalação e configuração do *Snort no Ubunbu* 12.04:

```
sudo su
```
(logar como super usuário ou usuário *root*)

```
apt-get install apache2 php5 mysql-server snort –y
apt-get install snort-mysql
```

Configurando o mysql:

```
mysql -u root –p(admin)
create database snort;
grant all privileges on snort.* to ‘snort’@’localhost’ identified by ‘admin’;
quit
```
(Comando para sair do Banco de Dados)

```
cd /usr/share/doc/snort-mysql
zcat create_mysql.gz | mysql -u snort –p(admin) snort
rm /etc/snort/db-pending-config
```

Configurando o Snort-mysql:

```
dpkg-reconfigure snort-mysql
```

Instalação do *Acidbase* para acesso de informações pelo navegador:

```
apt-get install acidbase
```
#no navegador digitar (localhost/acidbase)

Reinicializar *snort*:

```
/etc/init.d/snort start
```