UNIVERSIDADE SAGRADO CORAÇÃO

VITOR MUTA BARRETO

SISTEMA DE DETECÇÃO DE INTRUSÃO ASSOCIADO A UM FIREWALL PARA SEGURANÇA DE REDES

BAURU
2011

UNIVERSIDADE SAGRADO CORAÇÃO

VITOR MUTA BARRETO

SISTEMA DE DETECÇÃO DE INTRUSÃO ASSOCIADO A UM FIREWALL PARA SEGURANÇA DE REDES

Trabalho de conclusão de curso apresentado à Universidade Sagrado Coração, para a obtenção do título de bacharel em Ciência da Computação, sob orientação do prof. Esp. Henrique Pachioni Martins.

BAURU
2011

Barreto, Vitor Muta

B273s

Sistema de detecção de intrusão associado a um firewall para segurança de redes / Vitor Muta Barreto -- 2011.

55f.: il.

Orientador: Prof. Esp. Henrique Pachioni Martins

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Sagrado Coração – Bauru – SP.

1. Firewall. 2. Vulnerabilidades. 3. Ataques. 4. Sistemas de detecção de intrusão. I. Martins, Vitor Muta. II. Título.

VITOR MUTA BARRETO

SISTEMA DE DETECÇÃO DE INTRUSÃO ASSOCIADO A UM FIREWALL PARA SEGURANÇA DE REDES

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Esp. Henrique Pachioni Martins.

Banca examinadora:

_____

Prof. Esp. Henrique Pachioni Martins
Universidade Sagrado Coração

_____

Prof. Esp. Andre Luiz Ferraz Castro
Universidade Sagrado Coração

_____

Prof. Dr. Kelton A. Pontara da Costa
Universidade Sagrado Coração

Bauru, 07 de dezembro de 2011.

Dedico este trabalho a toda minha família, pela imensa confiança depositada em mim ao longo dessa jornada, que me apoiaram nos meus estudos e me deram base desde os primeiros contados com a escola.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me iluminou ao longo dessa caminhada, por ter me dado muita coragem, força, sabedoria e paciência.

A todos os meus familiares, amigos e professores que direta ou indiretamente me apoiaram e me incentivaram durante todas as dificuldades enfrentadas para a realização desse trabalho.

Mas em especial gostaria de agradecer aos meus pais, Sidnei e Marcia, pelo esforço, dedicação, compreensão e incentivo, me ajudando a chegar até aqui hoje.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

DMZ - DeMilitarized Zone

DoS - Denial of Service

FTP - File Transfer Protocol

ICMP - Internet Control Message Protocol

IDS - Intrusion Detection System

IP - Internet Protocol

MySQL - My Structured Query Language

PHP - Hypertext Preprocessor

SDI - Sistema de Detecção de Intrusão

TCP - Transfer Control Protocol

UDP - User Datagram Protocol

WWW - World Wide Web

SUMÁRIO

RESUMO

Nos dias atuais, com o crescente uso da Internet como meio global de comunicação, tem-se verificado um significativo aumento com relação ao número de vulnerabilidades e ataques existentes nos sistemas computacionais, podendo esses problemas trazer consigo grandes prejuízos para as organizações, e com isso a segurança se tornou um ponto crucial em todos os sistemas computacionais. Contudo, apesar da grande variedade de ferramentas de segurança, um problema persiste na forma como estas são implementadas e utilizadas. Nesse sentido, o objetivo deste trabalho foi apresentar e implementar, de forma prática e objetiva uma associação entre dois importantes sistemas de segurança: Sistemas de Detecção de Intrusão (*IDS*) e *firewall*, onde foi possível identificar e testar as vulnerabilidades de um *firewall* a partir da implantação do Sistema de Detecção de Intrusão e de testes de simulação de ataque como scanner da rede e ferramentas para varredura de vulnerabilidades, e portanto teve esses ataques detectados e identificados pelo *IDS*. Pois ambas tecnologias são utilizadas separadamente na maioria dos casos, sendo que o propósito desta associação foi se obter uma efetiva solução de segurança em conjunto.

Palavras-chave: *Firewall*. Vulnerabilidades. Ataques. Sistemas de Detecção de Intrusão.

ABSTRACT

Nowadays, with the relevant Internet use as a global way of communication, we have verified a real increasing vulnerable number of the existent attacks in the computer systems and so, these problems may bring big losses to the organizations, with it the security has become the crucial point in all the computer system. However, besides the great variety of security tools, a problem persists in the way they are implemented and used. So, the main objective of this research was to show and implement in a practical and objective way, an association between two important security systems: Intrusion Detection System (IDS) and Firewall, where it was be possible to identify and test a firewall vulnerability from the Intrusion Detection System implantation and from attack simulation tests as net scanner and tools to sweep the vulnerabilities, having these attacks detected and identified by the IDS. The reason being both technologies are used separately in most cases and the aim of this association was to obtain an effective security solution in its total.

Key-words: *Firewall*. Vulnerability. Attacks. Intrusion Detection System.

1 INTRODUÇÃO

Na visão de Rocha Junior (2010), com a crescente evolução tecnológica tornou-se indispensável que as empresas possuam uma estrutura que consiga oferecer a seus funcionários acesso a Internet, como para fornecer acesso externo as suas aplicações para servir seus parceiros e clientes. Nesse contexto, milhares de empresas nos dias atuais possuem uma infraestrutura de acesso a Internet. Com as inúmeras vantagens que este acesso acaba proporcionando como a conectividade entre redes distintas, correio eletrônico, comunicação eficiente e barata entre pessoas, etc., vieram também inúmeros problemas, principalmente no que diz respeito à segurança dos dados.

Rocha Junior (2010) ainda apresentou que a partir do momento que máquinas são ligadas em rede, elas passam a ser alvo de diversos tipos de ataques, vindos tanto da rede externa (Internet) quanto da própria rede interna, e com isso passa a existir a possibilidade das informações serem roubadas ou destruídas. É de responsabilidade do administrador de rede e de especialistas em segurança criar ou utilizar mecanismos para dificultar a ação dos invasores.

Com o aumento dos ataques as empresas, profissionais da área da tecnologia da informação visando proteger informações e dados, buscam utilizar diversas ferramentas que auxilie na segurança como *firewalls*, Antivírus, Sistemas de Detecção de Intrusão, etc., para ter a possibilidade de detectar e bloquear invasões e possíveis danos causados por invasores.

O *firewall* vem a ser uma barreira de proteção, que controla o tráfego de dados entre o computador e a Internet, e seu principal objetivo é permitir somente a transmissão e a recepção de dados autorizados. Com essas características se tornou uma ferramenta de defesa cada vez mais usada, no entanto sua configuração é muito complexa, podendo resultar em erros, assim apresentar vulnerabilidades para ataques.

> Frequentemente existe a necessidade de conhecer o nível de proteção oferecido por um *firewall*, bem como suas deficiências. *Firewalls* não são softwares ou equipamentos que podem simplesmente ser retirados da caixa, conectados na rede e utilizados instantaneamente. Precisam ser adequadamente configurados, geralmente seguindo uma política de segurança da informação corporativa, para que possam atender necessidades específicas de cada rede. Além disso, a configuração é dinâmica e precisa ser revista periodicamente, seja quando novas

> vulnerabilidades são descobertas, quando são efetuadas alterações na arquitetura da rede ou ainda quando a política de segurança da informação corporativa é modificada.(AL-SHAER, 2003, 2004b apud BARBOSA, 2006, p. 2).

Segundo Bernardes (1999), apesar do uso dos diversos esquemas de segurança existentes, ainda existe a possibilidade de ocorrências de falhas nestes esquemas e, portanto, é desejável a existência de sistemas capazes de realizar a detecção de tais falhas e informar o administrador da rede. Tais sistemas são conhecidos como Sistemas de Detecção de Intrusão (SDI).

Assim o conhecimento e uso de ferramentas de detecção e prevenção de intrusão pode ser fundamental para manter a segurança de computadores e o sigilo de informações, pois com a sua implantação é possível verificar os ataques e com isso as vulnerabilidades que o *firewall* possui, podendo essas vulnerabilidades serem sanadas de imediato pelo administrador da rede.

Diante desse panorama, este trabalho tem como objetivo unir um Sistema de Detecção de Intrusão a um *firewall* e possibilitar um gerenciamento mais fácil e eficiente de uma rede, onde possa analisar as vulnerabilidades de um *firewall*, sendo esse um dos problemas enfrentados em ambientes de redes, onde através da implantação da tecnologia de *IDS*, verificar se realmente a mesma auxilia na segurança e acaba por detectar ataques a rede, assim demonstrando as vulnerabilidades de um *firewall*.

1.1 PROBLEMA

Com a implantação de um Sistema de Detecção de Intrusão pode-se verificar a vulnerabilidade de um *firewall*?

2 OBJETIVOS

2.1 OBJETIVO GERAL

Verificar a vulnerabilidade de um *firewall* por meio da implantação de um Sistema de Detecção de Intrusão (SDI).

2.2 OBJETIVOS ESPECÍFICOS

- Implantar um Sistema de Detecção de Intrusão (SDI);

- Avaliar a segurança de um *firewall* por meio de um Sistema de Detecção de Intrusão, analisando as tentativas de invasão a rede oriundos de simulações.

2.3 JUSTIFICATIVA

A crescente utilização da internet pelas empresas acarretou em inúmeros problemas principalmente no que diz respeito à segurança dos dados à medida que o número de ataques e intrusões a sistemas e redes de computadores aumenta de maneira significativa, com isso o *firewall* se tornou um dos mais importantes nesse aspecto, sendo essencial na infraestrutura de qualquer ambiente corporativo.

Com o aumento de ataques, ferramentas e técnicas são criadas com o intuito de barrar invasões, roubo de informações ou qualquer tentativa de acesso que não seja autorizado, mas levando em consideração que ocorram falhas na criação dos métodos e implantações dessas ferramentas e técnicas, faz com que tornem ineficientes. Contudo é de extrema importância o uso correto de metodologias e técnicas de detecção de intrusão associados ao *firewall*, para garantir a integridade das informações que trafegam por toda rede.

> Um *firewall* não deve ser seu principal meio de defesa contra a ação de intrusos. Ele só é apropriado como uma medida de segurança complementar. O emprego de *firewalls* normalmente fornece um falso sentido de segurança. Se ele o induzir a relaxar em outras medidas de segurança, terá um efeito negativo na segurança de sua instalação.(NEMETH; SNYDER; HEIN, 2007, p. 486).

A utilização de uma ferramenta para analisar as tentativas de ataques à rede auxilia na segurança de redes de computadores quanto a verificar as vulnerabilidades que estão sendo exploradas no *firewall*, com isso faz-se necessário à utilização de ferramentas de monitoração dessas redes de forma a se identificar possíveis atividades suspeitas e tomar as medidas cabíveis em tempo hábil. Através desta análise é possível saber de onde está partindo uma invasão podendo assim bloquear a comunicação com a origem, evitando um possível ataque a rede e ao *firewall*.

SDI (Sistema de Detecção de Intrusão) é um sistema onde é possível detectar e notificar as tentativas de intrusão, podendo emitir alarmes ou executar uma ação automática, assim com a sua implantação o administrador da rede consegue tomar alguma atitude em relação ao *firewall*, ou até mesmo outro programa ou processo seja iniciado automaticamente.

Portanto com a crescente complexidade das ameaças digitais tem exigido a associação de várias tecnologias de segurança atualmente disponíveis, sendo que a associação do Sistema de Detecção de Intrusão com o *firewall* pode-se criar uma solução de segurança integrada, onde com a implementação do *IDS* é possível verificar os ataques à rede e com isso identificar as vulnerabilidades que o *firewall* possui, podendo essa vulnerabilidade detectada no *firewall* ser imediatamente solucionada pelo administrador da rede.

3 REVISÃO DE LITERATURA

Neste capítulo serão apresentados os conceitos utilizados para compreensão do modelo proposto. Estes conceitos servirão para o entendimento do conteúdo presente no restante deste documento.

3.1 *FIREWALL*

Firewall é um termo inglês que traduzido para Língua Portuguesa significa parede de fogo, que segundo Zwicky, Cooper e Chapman (2000, p. 104) é “um componente ou conjunto de componentes que restringe o acesso entre uma rede protegida e a Internet ou entre outros conjuntos de rede.”

“Um mecanismo muito utilizado na prática para aumentar a segurança das redes de computadores, protegendo-as de ataques externos, é o *firewall.*” (BERNARDES, 1999, p. 16).

Têm por objetivo básico principal defender a organização de ataques externos, podendo ser utilizado para regular o uso de recursos externos pelos usuários internos. (CAMPELLO; WEBBER, 2001). Para atingir seu objetivo, todo o tráfego entre a Internet e a rede obrigatoriamente terá que passar através do *firewall*, onde, será verificado, assim deverá permitir unicamente a passagem de tráfego autorizado.

Em relação ao *firewall* NIC (2003) relata que um *firewall* bem configurado é um instrumento importante para implantar a política de segurança da sua rede. Ele pode reduzir a informação disponível externamente sobre a sua rede, ou, em alguns casos, até mesmo barrar ataques a vulnerabilidades ainda não divulgadas publicamente. Por outro lado, *firewalls* não são infalíveis. A simples instalação de um *firewall* não garante que sua rede esteja segura contra invasores. Um *firewall* não pode ser a sua única linha de defesa, ele é mais um dentre os diversos mecanismos e procedimentos que aumentam a segurança de uma rede.

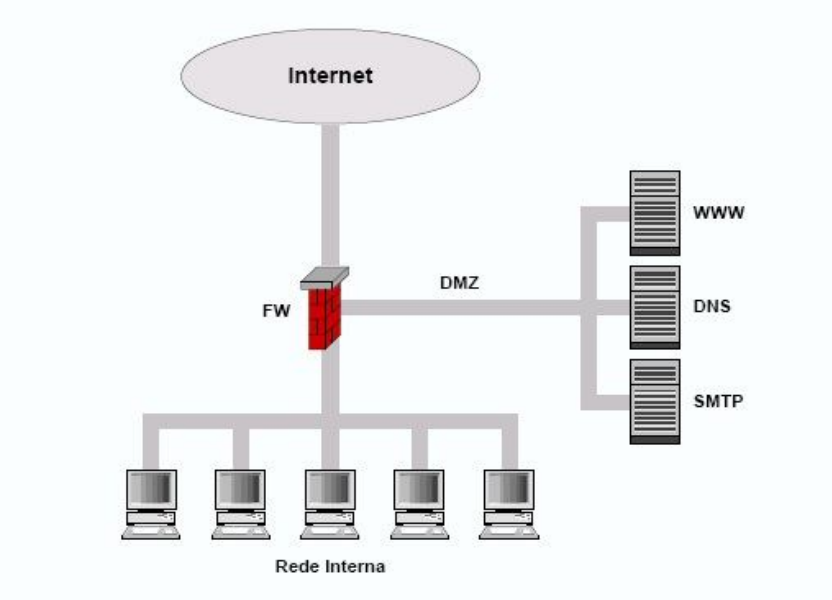

Figura 1 - Exemplo simples de *Firewall.*
Fonte: Nic (2003).

3.1.1 Tipos de *firewalls*

Existem três tipos principais de *firewall*: *firewall* a nível de pacote, que funciona nas camadas de rede e de transporte, *firewall* a nível de pacotes baseado em estados e o *firewall* a nível de aplicação, que funciona nas camadas de aplicação, sessão e transporte.

Cada tipo possui vantagens e desvantagens um em relação ao outro, é muito comum as empresas utilizarem as duas tecnologias, *firewall* a nível de pacote e a nível de aplicação, em conjunto na sua infraestrutura de segurança, proporcionando um nível maior de segurança. (ROCHA JUNIOR, 2010).

3.1.1.1 Firewall a nível de pacotes

Como o nome já diz, esse tipo de *firewall* analisa e filtra pacotes enviados por redes distintas de comunicação. (INOKOSHI, 2007).

Zwicky, Cooper e Chapman (2000, p. 105) define filtragem de pacotes como,

> A ação de um dispositivo para controlar seletivamente o fluxo de dados de e para uma rede. Os filtros de pacotes deixam passar ou bloqueiam pacotes, em geral enquanto estão roteando (encaminhando) os pacotes de uma rede para outra (com maior frequência de uma rede interna e vice-versa). Para

> realizar a filtragem de pacotes, você configura um conjunto de regras que especificam que tipos de pacotes serão permitidos e que tipos deverão ser bloqueados. A filtragem de pacotes pode acontecer em um roteador, em uma ponte ou em um *host* individual. Às vezes ela é conhecida como triagem.

O *firewall* a nível de pacote analisa as informações contidas no cabeçalho dos pacotes, e de acordo com as regras especificadas pelo administrador, determinam se o pacote será aceito ou descartado. Isso torna o *firewall* a nível de pacotes transparente ao usuário e uma outra vantagem é que ele ganha em desempenho se comparado ao *firewall* a nível de aplicação. (ROCHA JUNIOR, 2010).

Na Figura 2 pode-se demonstrar uma rede protegida por um roteador configurado com a capacidade de filtragem de pacotes, também chamado de *screening router*.

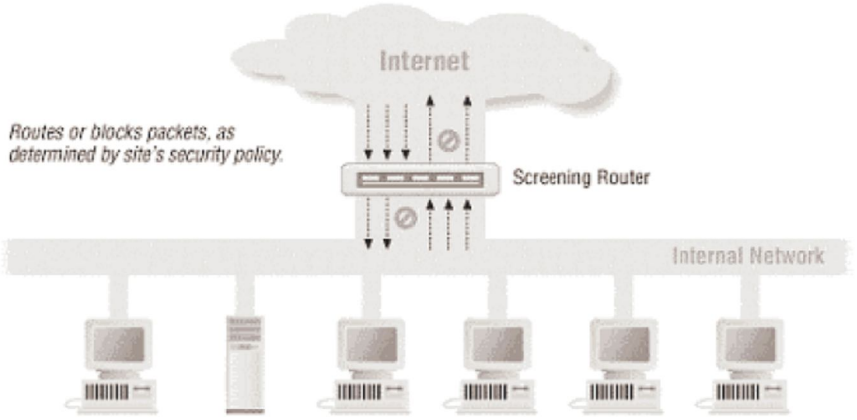

Figura 2 - Exemplo de Filtro de Pacotes.
Fonte: Camy, Silva e Righi (2003).

3.1.1.2 Firewall a nível de pacotes baseado em estados

De acordo com Rocha Junior (2010) o *firewall* a nível de pacotes baseado em estado, também chamado de *stateful packet filter*, é semelhante ao *firewall* a nível de pacotes sendo que ele possui uma funcionalidade a mais. A tomada de decisão se o pacote será aceito ou descartado é feita com base em dois elementos: Informações do cabeçalho do pacote (assim como o *firewall* a nível de pacotes) e uma tabela de estados que guarda o estado das conexões.

3.1.1.3 Firewall a nível de aplicação

Segundo Rocha Junior (2010) um *firewall* a nível de aplicação, também conhecido como servidor *proxy*, proporciona um nível mais refinado de segurança, ele faz mais do que analisar o cabeçalhos *TCP*, *UDP* e *IP*, ele toma decisões com base em dados da aplicação. Um servidor *proxy* funciona da seguinte maneira: Um cliente, que neste caso pode ser um navegador web, se conecta ao servidor *proxy* e realiza uma requisição de um site qualquer, o servidor então recebe esta requisição e a encaminha para o servidor *web* de destino. O servidor *web* irá responder a requisição ao servidor *proxy* que irá repassar os dados para o cliente que realizou a requisição.

Os serviços de *proxy* de um *firewall* são programas aplicativos ou servidores especializados que tomam as solicitações de usuários de serviços da Internet e os encaminham aos serviços reais. (ZWICKY; COOPER; CHAPMAN, 2000).

A Figura 3 ilustra o funcionamento de um servidor proxy.

> Primeiro o usuário estabelece uma conexão com o servidor *proxy* no firewall (passo 1). O *proxy* junta a informação enviada pelo usuário e verifica se o endereço solicitado é permitido realizando uma busca no Banco de Dados de Autorização (passo2). Caso a chamada seja permitida, ele reenvia adiante a solicitação para o servidor real, mas com o endereço *IP* do *firewall* (passo 3), protegendo assim o verdadeiro endereço de origem da solicitação. Ao receber um pacote de respostas (passo 4), examina seu conteúdo e em seguida verifica se o pacote pertence a uma solicitação interna. Em caso positivo, repassa-o ao solicitante (passo 5). Depois que a sessão é estabelecida (passo 6), o servidor *proxy* atua como uma retransmissora e copia os dados entre o cliente que iniciou a aplicação e o servidor. Desta forma, ele tem controle total sobre a sessão e pode realizar um *logging* tão detalhado quanto desejar.(CAMY; SILVA; RIGHI, 2003, p. 15-16).

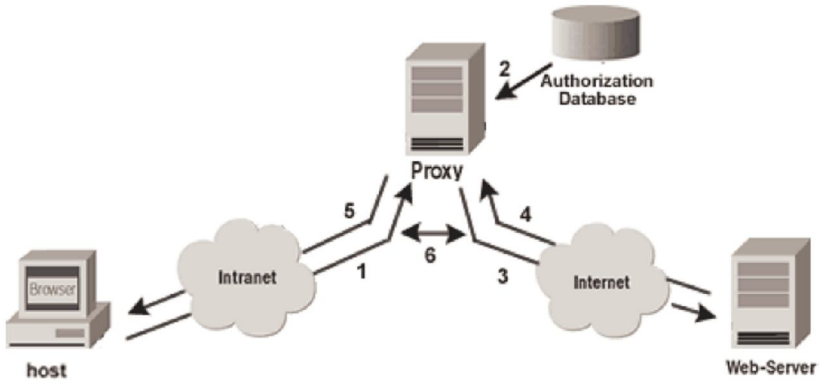

Figura 3 - Modelo de servidor *proxy.*
Fonte: Camy, Silva e Righi (2003).

3.1.1.4 Firewalls bastion host

Os *bastion hosts* são os servidores que possuem instalados serviços a serem oferecidos para a Internet.

Um *firewall* ou um roteador podem ser considerados *bastion hosts*, sendo que provê os recursos permitidos segundo a política de segurança da empresa. Pode ser projetado com a finalidade de ser servidor *web*, servidor *FTP*, entre outras funções que estiverem disponíveis a usuários de fora da proteção interna da rede. (CAMY; SILVA; RIGHI, 2003).

A Figura 4 mostra um exemplo de rede que suporta um *bastion host* simples.

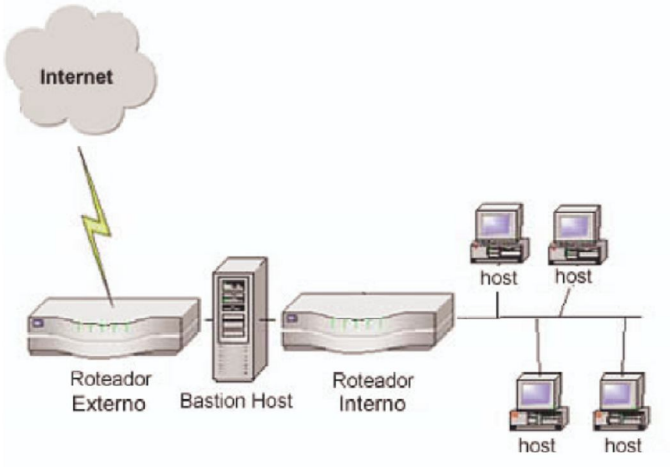

Figura 4 - *Firewall* tipo *Bastion Host.*
Fonte: Camy, Silva e Righi (2003).

3.1.2 Arquiteturas

A posição que o *firewall* será implantado na topologia de rede terá um impacto significativo no nível de segurança da rede da empresa. Todo o dado que trafegue de uma rede para outra deve passar obrigatoriamente pelo *firewall*. (ROCHA JUNIOR, 2010).

Assim a seguir poderão ser apresentadas três possíveis arquiteturas para a implantação de um *firewall*, a saber: *Dual-homed Host*, *Screened Host* e *Screened Subnet*.

3.1.2.1 Dual-homed

Siewert (2008), essa arquitetura normalmente é montada sobre um computador que possui no mínimo duas interfaces de rede, agindo também como um roteador entre as duas redes conectadas as placas de rede, o que permite a retransmissão de pacotes diretamente entre as redes, o que permite ainda que sistemas dentro do *firewall* se comuniquem diretamente com a Internet e com a rede protegida, porém o acesso da Internet para sistemas internos ou para a rede protegida é bloqueado, isso é claro, se for bloqueado esse acesso.

“A arquitetura *dual-homed host* apresenta um *firewall* com duas placas de rede que funciona como um separador, protegendo a rede interna de possíveis ataques vindos da rede externa.” (ROCHA JUNIOR, 2010, p. 47).

A Figura 5 apresenta os elementos que formam esta arquitetura.

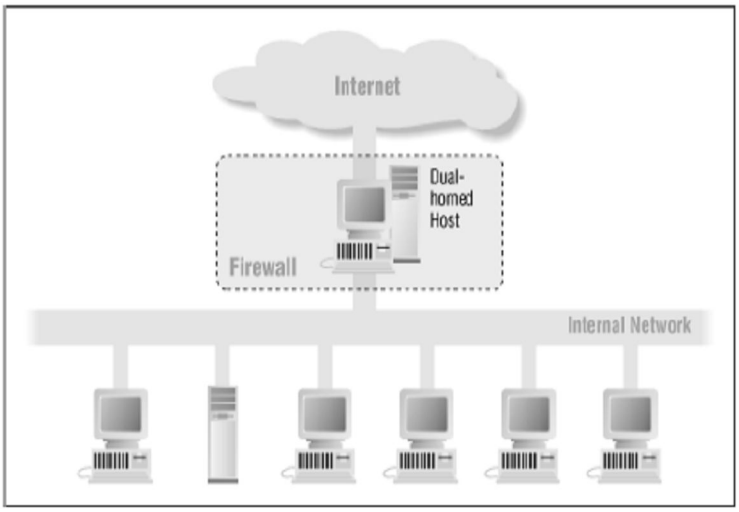

Figura 5 - Exemplo de *Dual-Homed.*
Fonte: Inokoshi (2007).

3.1.2.2 S*creened host*

Siewert (2008) nessa arquitetura existe um roteador separado, normalmente chamado de roteador de borda e um *host* chamado de *bastion host*, que nada mais é que um dispositivo que implementa um alto grau de segurança e um roteador separado para a rede interna. Neste caso a segurança primária é implementada pelo filtro de pacotes que está implementado no roteador de borda e a segurança secundária é implementada pelo *bastion host*, provê serviços somente para a rede interna.

Rocha Junior (2010) a arquitetura *screened host* é formada por um filtro de pacotes e um *bastion host*. Nesta arquitetura o *firewall* é configurado para permitir acesso a rede interna apenas através do *bastion host*, ou seja, os usuários externos que quiserem acessar os sistemas internos deverão se conectar primeiramente com o *bastion host*.

A Figura 6 apresenta as posições na rede dessa arquiterura.

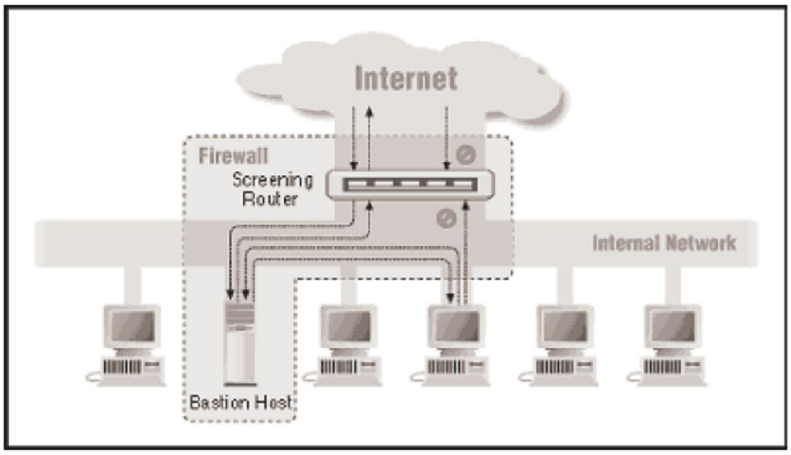

Figura 6 - Exemplo de *Screened Host.*
Fonte: Camy, Silva e Righi (2003).

3.1.2.3 Screened subnet architecture

Rocha Junior (2010) a arquitetura *screened subnet* proporciona um aumento considerável da segurança ao fazer uso de uma *DMZ*, é um melhoramento da arquitetura *screened host*. No modelo anterior se o *bastion host* fosse comprometido o invasor teria total acesso a rede interna, isso não ocorre na arquitetura *screened subnet*. O *bastion host* fica em uma *DMZ*, que é uma zona que fica entre a rede interna e a externa, caso ele seja comprometido o filtro interno ainda protegerá a rede interna.

Siewert (2008) essa arquitetura é um melhoramento da arquitetura *screened host*, ou seja tem uma camada extra de segurança. É implementada uma rede de perímetro com o fim de proteger a rede interna da Internet, e existe também um roteador externo e um roteador interno. Essa rede de perímetro é para evitar ataques à rede interna e *bastion host* fica alocado na rede de perímetro, considerando o mesmo é alvo de ataques, caso um atacante chegue até ele, não passará da rede de perímetro, por causa do filtro de pacotes implementado no roteador interno.

Os componentes da arquitetura *screened subnet* podem ser visualizados na Figura 7:

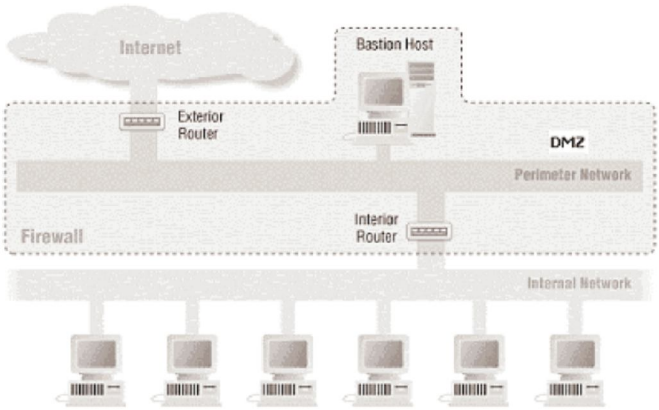

Figura 7 - Exemplo de S*creened Subnet.*
Fonte: Camy, Silva e Righi (2003).

3.1.3 F*irewall iptables*

O *Iptables* é um *firewall* em nível de pacotes e funciona baseado no endereço/porta de origem/destino do pacote, prioridade, etc. Ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar. Em *firewalls* mais restritivos, o pacote é bloqueado e registrado para que o administrador do sistema tenha conhecimento sobre o que está acontecendo em seu sistema. (SILVA, 2005).

Ele também pode ser usado para modificar e monitorar o tráfego da rede, redirecionamento de pacotes, marcação de pacotes, modificar a prioridade de pacotes que chegam/saem do seu sistema, contagem de bytes, dividir tráfego entre máquinas, criar proteções anti-*spoofing*, contra *DoS*, etc. O tráfego vindo de máquinas desconhecidas da rede pode também ser bloqueado/registrado através do uso de simples regras. As possibilidades oferecidas pelos recursos de filtragem *iptables* como todas as ferramentas *UNIX* maduras dependem da imaginação do criador das regras, pois ele garante uma grande flexibilidade na manipulação das regras de acesso ao sistema, precisando apenas conhecer quais interfaces o sistema possui, o que deseja bloquear, o que tem acesso garantido, quais serviços devem estar acessíveis para cada rede, e iniciar a construção de seu *firewall.* (SILVA, 2005).

Como surge várias ameaças novas diariamente, o *firewall IpTables* também está sujeito à falhas e vulnerabilidades, e com isso necessita sempre estar se atualizando. (SIEWERT, 2008).

3.1.3.1 A configuração do iptables

A configuração de um *firewall iptables* é realizada por uma série de comandos (regras) que são interpretados pelo *kernel* do sistema operacional. Tais comandos podem ser inseridos diretamente no *shell* do sistema operacional ou através de arquivos de *scripts* (arquivos texto). (INOKOSHI, 2007).

3.2 VULNERABILIDADES

Vulnerabilidade é definida como uma falha no projeto, implementação ou configuração de um *software* ou sistema operacional que, quando explorada por um atacante, resulta na violação da segurança de um computador.

As vulnerabilidades por si só não causam incidentes, elas apenas são brechas para que uma possível ameaça cause, através de falha existente em um ou mais ativos que pode ou não ser explorada por uma ameaça. (ROCHA JUNIOR, 2010).

Há diversos tipos de ataques que exploram as falhas das vulnerabilidades, a exploração de tais vulnerabilidades pode ser utilizada com as mais diferentes finalidades, algumas inclusive, podem deixar computadores, *firewalls* ou outros equipamentos de rede completamente inutilizáveis. (BARBOSA, 2006).

Na visão de CERT (2006), existem casos onde um *software* ou sistema operacional instalado em um computador pode conter uma vulnerabilidade que permite sua exploração remota, ou seja, através da rede. Portanto, um atacante conectado à Internet, ao explorar tal vulnerabilidade, pode obter acesso não autorizado ao computador vulnerável.

3.3 ATAQUES

Segundo Oliveira (2007), várias técnicas de segurança são usadas contra os ataques proporcionados pela Internet, mas a maioria das técnicas de segurança usadas não são eficazes, talvez porque os administradores de sistema não compreendam a natureza dos ataques, que passam a existir devido a aspectos técnicos, humanos e organizacionais, ou seja devido a uma falha no projeto, na

implementação de um protocolo, aplicação, serviço, sistema, ou ainda devido a erros de configuração e administração de recursos computacionais, sem mencionar novas tecnologias que são criadas e trazem consigo novas vulnerabilidades. Novas formas de ataques estão em constante evolução e cada vez mais sofisticadas.

Todo sistema está sujeito a diferentes tipos de ameaças, sejam elas internas ou externas, acidentais ou maliciosas. Explorando certas vulnerabilidades ou brechas, diversos tipos de ataques são desencadeados atualmente, desde tentativas simples de negação de serviço, até ataques sofisticados que utilizam recursos distribuídos. (CAMPELLO; WEBBER, 2001).

O aumento dos casos de insegurança da informação existe devido aos ataques como, do tipo *port scanning*, as ferramentas para injeção de pacotes, *DOS*(*Denial of Service*), *IP spoofing*, , *sniffing* de pacotes, ferramentas para varredura de vulnerabilidades, sendo elas descritas a seguir.

O *port scanning* é uma técnica que consiste na busca por portas *TCP* abertas por onde pode ser feita uma invasão, sendo o Nmap um dos *port scanning* mais conhecidos e utilizados, com ele o invasor é capaz de descobrir os serviços que estão sendo utilizados por uma determinada máquina, além do sistema operacional que ela roda. (ROCHA JUNIOR, 2010).

As ferramentas para injeção de pacotes permitem que pacotes de protocolos da pilha *TCP/IP* sejam criados e injetados em uma rede. Os tipos de protocolos suportados e os dados que podem ser configurados variam de aplicativo para aplicativo. De modo geral, os protocolos suportados são o *IP, TCP, UDP e ICMP*. (WANNER, 2003).

A técnica *DoS*, são ataques baseados na negação de serviço, no qual consistem em deixar o serviço oferecido por um servidor indisponível, onde o ataque consiste em sobrecarregar um servidor com uma quantidade excessiva de solicitações de serviços. (ROCHA JUNIOR, 2010).

O *IP spoofing* é uma técnica com a qual o atacante mascara o endereço *IP* real que ele utiliza, para se fazer passar por outro computador da rede para conseguir acesso a um sistema. (ROCHA JUNIOR, 2010).

De acordo com Nemeth, Snyder e Hein, (2007) *Sniffing* de pacotes é um programa ou dispositivo que analisa o tráfego da rede, podendo ver quaisquer informações não codificadas que estejam sendo transmitidas, podendo seguir senhas ou dados.

Ferramentas para varredura de vulnerabilidades, realizam primeiramente uma varredura na estação alvo a fim de verificar os serviços ativos, as portas abertas e o tipo de sistema operacional existente nela. Após a varredura inicial, e com base nos seus resultados obtidos é feita uma série de testes a procura de falhas conhecidas que possam existir nos serviços ou no sistema operacional. Ao final dos testes, é gerado um relatório das características das estações e dos problemas encontrados. Os programas para varredura de vulnerabilidades realizam a maioria dos seus testes no nível de aplicação, procurando falhas em serviços, aplicativos, sistemas operacionais e representam um grande risco à segurança de uma rede quando utilizado por pessoas não autorizadas. Assim, as ferramentas de segurança de rede, como os *IDS*, devem ser capazes de detectar as vulnerabilidades detectadas por tais aplicativos. (WANNER, 2003).

3.4 SISTEMAS DE DETECÇÃO DE INTRUSÃO

Do inglês *Intrusion Detection System*, sendo programa, ou um conjunto de programas, cuja função é detectar possíveis ataques.

Campello e Weber (2001) definem detecção de intrusão como sendo o processo de identificar e responder a atividades maliciosas dirigidas a computadores e recursos de rede e descreve, dentre outras definições, que detecção de intrusão é a tarefa de coletar informações de uma variedade de fontes - sistemas ou redes - e então analisá-las buscando sinais de intrusão e de mau-uso. No primeiro caso, o termo detecção de intrusão é usado tanto no sentido de detecção propriamente dita como na reação a essa atividade. Isso amplia a funcionalidade dos *IDS*s, impondo a eles a difícil tarefa de reagir aos ataques detectados.

Segundo Bernardes (1999) as funcionalidades de um sistema de detecção tornam-se de vital importância na medida em que fornecem meios de inferir sobre o conteúdo das conexões permitidas e detectar as que apresentem um comportamento suspeito ou não condizente com a política de segurança implantada.

Sistemas de Detecção de Intrusos são desenvolvidos para detectar ataques que não foram prevenidos por outras ferramentas de segurança. Conforme Steffen Junior (2003) não se pode usar o *IDS* como única fonte de segurança de uma rede, nem em substituição a um *firewall*, mas sim em conjunto com outros métodos para aumentar a segurança da rede.

3.4.1 Tipos de *IDS*

Existem dois tipos de *IDS*: o baseado em Rede e o baseado em *Host.* Qual modelo escolher depende da estrutura de cada empresa, podendo até ser instalado os dois modelos trabalhando na mesma rede. (STEFFEN JUNIOR, 2003).

Independente da arquitetura ou do método utilizado para a detecção, todo *IDS* possui componentes em comum. Cada um desses componentes desempenha um papel importante na tarefa de identificar ações consideradas danosas ao sistema. (CAMPELLO; WEBBER, 2001). Assim, deve-se conhecer a anatomia de um *IDS* como pode-se ver na Figura 8.

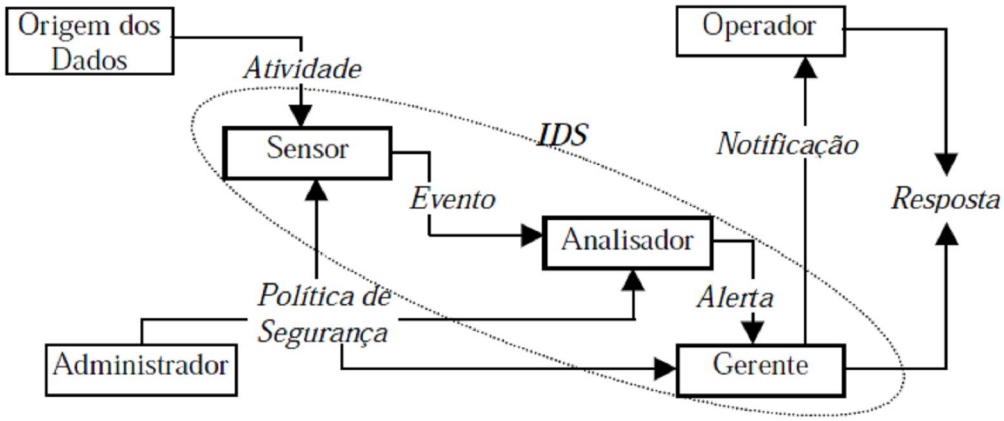

Figura 8 - Componentes de um *IDS.*
Fonte: Campello e Weber (2001).

3.4.1.1 Baseado em rede

"Sistemas baseados em rede examinam os dados que trafegam pela rede através da monitoração *on-line* dos pacotes." (BERNARDES, 1999, p. 21).

De acordo com Barbosa (2000) os sensores apesar de estarem instalados numa máquina específica monitoram todo o tráfego no seguimento de rede, pois a interface com a rede é colocada em modo promíscuo, neste modo o *software* recebe todos os pacotes naquele seguimento de rede, não apenas os destinados ao *IP* da interface, por isso um único sensor tem a capacidade de monitorar atividade de rede em diversas máquinas.

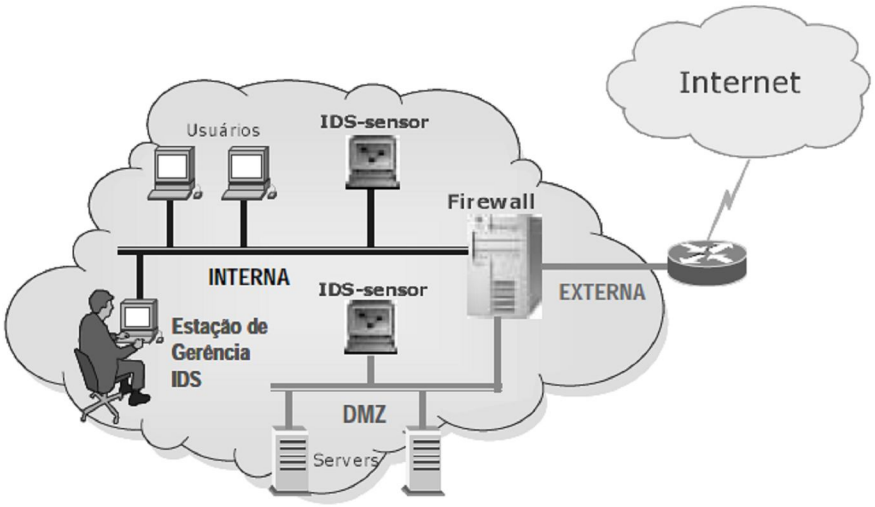

Figura 9 - *IDS* Baseado em rede.
Fonte: Barbosa (2000).

3.4.1.2 Baseado em host

"Sistemas baseados em *host* analisam dados de auditoria recolhidos normalmente pelos sistemas operacionais e buscam de diversas formas pelos padrões de ataque em um único *host.*" (BERNARDES, 1999, p. 21).

Segundo Barbosa (2000) os *IDS* baseados em host analisam sinais de intrusão na máquina nos quais estão instalados, eles frequentemente usam os mecanismos de log do sistema operacional e estão muito ligados aos recursos do sistema. Eles agem procurando por atividades não usuais em: tentativas de *login*, acesso à arquivos, alterações em privilégios do sistema, etc.

"Monitora o tráfego de máquinas individuais, permitindo uma melhor precisão na análise e gerando menos falso positivo." (STEFFEN JUNIOR, 2003, p. 42).

3.4.2 Métodos de detecção de intrusão

Segundo Campello e Weber (2001) os métodos de detecção de intrusão desempenham um dos principais papéis em um *IDS*, sendo responsáveis diretos pela busca de indícios de ações intrusivas, sendo evidente que diferentes ferramentas de detecção de intrusão utilizarão diferentes métodos para analisar os dados.

Assim dentre os métodos de detecção dos *IDS*, os mais comuns podem ser descritos a seguir.

3.4.2.1 Métodos tradicionais

Mesmo não sendo considerados mecanismos de detecção de intrusão propriamente ditos, enquadrados de forma mais correta como técnicas de auditoria de sistemas, a busca por indícios de intrusão utilizando conceitos tradicionais como trilhas de auditoria já são feitas há bastante tempo. Problemas como a dificuldade de manipular manualmente uma grande quantidade de informação, correlacionar diferentes tipos de dados e de realizar essa tarefa em tempo hábil a fim de evitar danos maiores ao sistema, são exemplos das proibições impostas à utilização desses métodos. Por outro lado, em conjunto com ferramentas e técnicas mais avançadas, é possível utilizar esses métodos para rastrear os danos causados em decorrência de um ataque ou, até mesmo, coletar provas que levem à descoberta de um antigo atacante. (CAMPELLO; WEBER, 2001).

Os métodos tradicionais possuem alguns problemas devido a manipulação de uma grande quantidade de informações, grande experiência em redes e baixa eficiência.

3.4.2.2 Baseado em assinaturas

De acordo com Campello e Weber (2001) um dos métodos mais tradicionais de detecção é a utilização de assinaturas de ataques. Nesse conceito, uma base de dados é criada com sequências de ações que seriam consideradas indícios de uma intrusão. Essas sequências são chamadas de assinaturas, identificando a maioria dos ataques conhecidos e servindo como base para a busca de intrusos no sistema.

Na visão de Steffen Junior (2003) as assinaturas são fornecidas pelos desenvolvedores do *IDS*, mas podem ser desenvolvidas pelo próprio administrador da rede ou encontradas em sites especializados em segurança, neste caso, pode ser necessário adaptar a regra para o *IDS* específico.

É possível comparar este método com um antivírus, que trabalha analisando cada pacote e comparando com as regras existentes, assim para criar uma regra se deve analisar as características do pacote que contém o ataque. (BARBOSA, 2000).

3.4.2.3 Baseado em anomalia

Bernardes (1999) define sistemas de detecção de anomalias como um modelo de atividade normal através de perfis de atividade dos usuários e qualquer desvio significante da norma estabelecida é considerado anômalo. Isto é executado pela construção de um modelo estatístico que contém métricas derivadas do sistema operacional e estabelece como intrusivo uma métrica observada que tem um desvio estatístico significante deste modelo.

Segundo Steffen Junior (2003) uma vantagem deste método é que ele pode detectar ataques desconhecidos, já que não trabalha analisando uma regra específica. Como desvantagem cita-se um número alto de falsos positivos.

3.4.4 *IDS*s mais conhecidos

Existem várias ferramentas para detecção de intrusão, onde nesta seção poderá apresentar algumas *IDS*s já existentes.

3.4.4.1 Snort

Um dos *IDSs* mais utilizados no momento, *Snort* combina simplicidade com eficiência. De distribuição livre, essa ferramenta baseia-se em uma arquitetura centralizada, dados coletados na rede e uma análise baseada em assinaturas, podendo ser executada em qualquer sistema *UNIX* e, inclusive, em *Windows.* (CAMPELLO; WEBBER, 2001).

Sua estrutura básica é simples, baseada na captura de pacotes de rede através da biblioteca *libpcap* e em um analisador simples e eficiente que trata tanto informações de cabeçalho quanto a área de dados dos pacotes coletados. Os pacotes que coincidem com alguma das regras da base podem ser simplesmente descartados, armazenados ou podem gerar algum alerta aos responsáveis pelo

sistema. Há ainda a possibilidade de utilizar regras de filtragem durante a coleta dos pacotes (*libpcap*), antes que eles passem pelo analisador, ou conceitos como pré-processadores e processadores de saída, responsáveis respectivamente por analisar os pacotes coletados antes que a base de assinaturas seja avaliada e por fazer a formatação dos resultados gerados. (CAMPELLO; WEBBER, 2001).

Segundo Campello e Weber (2001) uma das principais vantagens do *Snort* e a existência de uma base com milhares de assinaturas de ataques, disponível para *download* na página principal (www.Snort.org). Em sua grande maioria, essa base é fruto de colaborações da própria comunidade de usuários *Snort* espalhados pelo mundo, significando atualizações constantes e respostas praticamente imediatas ao surgimento de novos ataques.

O *Snort* possui ainda a facilidade de, além das regras disponíveis na Internet, permitir a criação de novas regras, no qual comparando o tráfego do segmento de rede com as regras existentes que o *Snort* detecta o código malicioso e gera o alerta ou toma medidas de contra-ataque. (STEFFEN JUNIOR, 2003).

3.4.4.2 Bro

Semelhante à ferramenta anterior, centralizada, de rede e baseada em assinaturas, este *IDS* possui como diferencial o formato de sua base de ataques. Nesse sentido, toda análise é feita utilizando *scripts*, descritos em uma linguagem própria, que representam políticas para cada serviço. *Bro* já conta com implementações em *DecUnix*, *FreeBSD*, *Solaris*, *SunOS* e *Linux*. (CAMPELLO; WEBBER, 2001).

*Br*o utiliza uma biblioteca chamada *libpcap* para fazer a captura de pacotes. Filtros no formato *TCPdump* são aplicados a essa biblioteca para fazer o primeiro nível de redução de dados, agilizando o trabalho das camadas superiores. Depois de capturados, os pacotes são repassados à uma máquina de eventos, que primeiro faz vários testes de integridade com o cabeçalho dos pacotes, descartando aqueles com problemas, e, após esses testes, processa-os na busca por eventos. Esse processamento inclui o tratamento de pacotes de conexão, a manutenção do estado das conexões ativas, bem como o tratamento de protocolos de nível mais alto. O processamento desses pacotes gera eventos para a camada superior, informando o estabelecimento de conexões, a chegada de pacotes *UDP* endereçados a alguma

máquina que já tenha recebido pacotes dessa natureza, considerado um *udp_request*, e outros eventos de nível mais alto. (CAMPELLO; WEBBER, 2001).

De posse desses eventos, o interpretador de *scripts*, escritos em uma linguagem especializada, aplica o código especificamente projetado para tratar de cada um desses eventos, buscando sinais de ataques.

3.4.4.3 Aafid

Desenvolvido pelo CERIAS (Center *for Education and Research in Information Assurance and Security*) da Universidade de Purdue, um dos grupos de maior prestígio na área, essa ferramenta utiliza o conceito de agentes distribuídos para a coleta dos dados e a análise de possíveis intrusões, tanto em redes como em *hosts*. Baseado em uma estrutura hierárquica, ele coleta as informações produzidas pelos agentes de cada *host*, pelo host em si e por conjuntos de *hosts*, reportando qualquer incidente através de uma interface gráfica. (CAMPELLO; WEBBER, 2001).

Segundo Campello e Weber (2001) os métodos de detecção usados podem variar conforme a descrição de cada agente, normalmente escritos para tratar de problemas bem específicos. Por exemplo, para detectar ataques do tipo SYN *flooding* existe um agente especialmente instanciado para analisar as conexões de rede recentemente criadas, a exemplo do que acontece com qualquer outro ataque a ser verificado. Com essa estrutura o *Aafid* agrega as vantagens da distribuição com a flexibilidade de seus mecanismos de detecção.

Por outro lado, sua estrutura hierárquica apresenta alguns problemas quanto à tolerância a falhas do sistema, com pontos únicos de falha bem definidos (monitores e transceivers). Esse problema pode ser minimizado com uma boa distribuição de agentes, mas pode inviabilizar todo o sistema de detecção em caso de ataques dirigidos a esses pontos. (CAMPELLO; WEBBER, 2001).

3.4.4.4 Emerald

Última geração de vários *IDS*s desenvolvidos pela SRI *International*, *Emerald* (*Event Monitoring Enabling Response to Anomalous Live Disturbance*) é uma ferramenta que alia toda a tradição da SRI em métodos de detecção, obtida com

projetos como IDES e NIDES, com o conceito de modularidade e distribuição. O grande objetivo desse projeto é trabalhar com redes de grande escala, abordando todos os problemas relacionados com esse tipo de ambiente, como a dificuldade de monitoramento e de análise. (CAMPELLO; WEBBER, 2001).

Emerald divide a rede em grupos independentemente administrados, chamados domínios. Cada domínio fornece uma gama de serviços de rede, como *ftp* ou *http*, com diferentes políticas de segurança e relações de confiança entre eles. Com essa divisão, uma estrutura hierárquica é adotada, com três níveis de análises feitas por uma arquitetura de monitoramento em três camadas: monitores de serviços, monitores de domínio e monitores de organização. (CAMPELLO; WEBBER, 2001).

Ainda em evolução, *Emerald* é apontado como um dos mais modernos *IDS*s desenvolvidos. Sua arquitetura hierárquica e modular combina a possibilidade de diferentes tipos de análise, possuindo módulos analisadores baseados em comportamento e em assinatura, com a distribuição e a possibilidade de correlacionar alertas oriundos de vários pontos do sistema. Como ponto negativo, resta ainda à preocupação com a tolerância a falhas desses módulos. (CAMPELLO; WEBBER, 2001).

4 METODOLOGIA

Está é uma pesquisa exploratória, a qual segundo Gil (2002) visa proporcionar maior familiaridade com o problema pesquisado, com vistas e torná-lo mais explícito, aprimorar idéias e descobrir intuições. Busca ainda estender o estudo do tema à integração com o problema, de maneira a construir hipóteses ou apenas explicitar os resultados. Envolve levantamento bibliográfico, parecer das pessoas que possuem experiências práticas com o problema pesquisado e análise de exemplos que incitem a compreensão.

A instalação e aplicação do *firewall* e do Sistema de Detecção de Intrusão foi realizada em diferentes microcomputadores como pode ser visto na Figura 10, sendo a instalação do *firewall* feita em um microcomputador com sistema operacional *Linux* Ubuntu versão 10.04 *Server*, equipado com um processador Intel *Dual Core* de 1.8 GHz, 2 Gb de memória ram, 160 Gb de Disco Rígido. O *IDS* foi instalado a partir do *download* de um ISO através da página http://snorby.org, que contém o *Snorby* que é *IDS* de código fonte aberto Linux sendo baseada no *Snort* e *Snorby*, vem com o sistema operacional *Turnkey Linux* e gerenciados a partir da interface *web Snorby*, sendo configurado em um microcomputador com um processador AMD Sempron de 1.8 GHz, 512 Mb de memória ram, 30 Gb de Disco Rígido. Já em um microcomputador com *Windows* 7 *Ultimate* 32 Bits equipado com processador Intel *Dual Core* de 1.8 GHz, 2 Gb de memória ram, 160 Gb de Disco Rígido foi feito como base para os devidos testes, sendo nesse microcomputador emulado outro sistema operacional, o Ubuntu 11.04 Desktop, que também serviu de base.

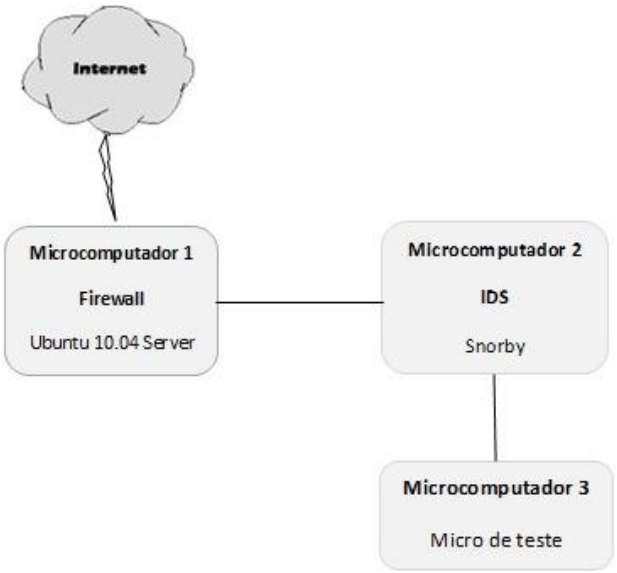

Figura 10 - Topologia do ambiente de testes.

Esta topologia de uma rede simplificada justifica-se pelo fato de que o principal objetivo do teste foi verificar a capacidade da ferramenta de *IDS* em detectar os ataques gerados contra a máquina alvo, assim verificar as vulnerabilidades de um *firewall*, e não de verificar a capacidade de segurança de uma rede como um todo.

As principais ferramentas que foram utilizadas, são *softwares* livres, onde foi utilizado como *IDS* o *Snort* que é um tipo de sistema de detecção de intrusão capaz de executar análise do tráfego da rede em tempo real, e o *Iptables*, como *firewall*, que tem como principal objetivo filtrar os tipos de acesso ao computador assim proibindo ou permitindo o devido acesso.

Para a primeira etapa deste trabalho foi pesquisado na Internet para verificar a existência de *softwares* livres para implantação de um *firewall* e de um sistema de detecção de intrusão, assim realizar a escolha de ambos para utilização no decorrer do trabalho.

Uma segunda etapa foi a instalação dos sistemas operacionais *Linux* em cada máquina, por ser um sistema de código fonte aberto e distribuído sob a licença pública gratuitamente.

Passado período de busca por *softwares* livres, foi realizado a implantação de um *firewall Iptables*, por ser o principal *firewall* para o *Linux*, e de longe o mais utilizado pelos administradores de sistemas.

Após a instalação do *firewall Iptables*, foi instalado e configurado o *software Snort* juntamente com o *Snorby* que vem a ser um moderno *front-end* para o *Snort*, onde com ele foi possível visualizar os alertas gerados pelo *Snort* , como sistema de detecção de intrusão, no qual a escolha do *Snort* para o papel de *IDS* deve-se principalmente pela sua disponibilidade na forma de *software* livre, evitando-se a aquisição de um produto proprietário apenas para a realização dos testes.

Juntamente com o *Snorby* e o *Snort,* vem configurado banco de dados *Mysql* e ao servidor *web* Apache, para que possa realizar o armazenamento e registro dos eventos no banco de dados *Mysql*, e posteriormente demonstrar os resultados através do *Snorby* rodando no servidor *web* Apache, onde os eventos são exibidos graficamente através de uma página *web*.

Com o *Snorby* foi possível ter uma interface de análise, no qual juntamente com o servidor de *web* Apache, e banco de dados *Mysql* conseguiu-se ter um ambiente de análise da base de dados que o *Snort* armazenou, e assim teve uma interpretação mais apurada dos dados, pois esse auxilia na busca por alarmes gerados de ataques a rede.

Depois desta etapa de instalação e configuração tanto do *firewall Iptables* quanto do *Snort* como sistema de detecção de intrusão, foi feita a etapa de teste, fase em que a estrutura implementada foi submetida a testes com o objetivo de verificar o sucesso da solução de segurança baseada em *firewall* e *IDS*. Sendo esses testes feitos com a ferramenta de scaneamento da rede que foi o Nmap, e testes de vulnerabilidades com o uso do Nessus, onde foi submetido a ataques do computador base destinados a máquina que se encontra instalado o *Snort*, a fim de garantir que os sensores *IDS* identificasse os ataques.

O objetivo desta análise foi verificar o funcionamento e o comportamento do *IDS* em um ambiente de rede simulado. Com esta análise foi possível estudar melhor o funcionamento do *IDS* frente a diferentes tipos de ataques que foram realizados.

Por fim foi feito a apresentação de resultados obtidos com a implantação do *IDS* em conjunto com *firewall*, demonstrando que *firewalls* apresentam vulnerabilidades, podendo com o trabalho em conjunto com o *IDS* possa ter uma melhora no quesito segurança de redes.

5 RESULTADOS OBTIDOS

A análise que será apresentada é fruto dos resultados obtidos no decorrer da aplicação, onde foi testado o *firewall* e o Sistema de Detecção de Intrusão, através de testes de scanner e testes de busca de vulnerabilidade na rede, onde as detecções foram identificadas pelo *Snort* e demonstradas através do *Snorby* que consegue demonstra graficamente.

Inicialmente pode-se observar na Figura 11 o arquivo de configuração do *firewall*, sendo que o mesmo está localizado no caminho "/etc/init.d/*firewall*".

```
#!/bin/bash
iniciar(){
    # Abre para a faixa de endereços da rede local:
    iptables -A INPUT -s 10.36.96.0/255.255.255.0 -j ACCEPT
    # Abre uma porta (inclusive para a internet):
    iptables -A INPUT -p tcp --dport 22 -j ACCEPT
    # Ignora pings:
    iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
    # Protege contra IP spoofing:
    echo 1 > /proc/sys/net/ipv4/conf/default/rp_filter
    # Descarta pacotes malformados, protegendo contra ataques diversos
    iptables -A INPUT -m state --state INVALID -j DROP
    # Abre para a interface de loopback. Esta regra é essencial para que
    # o KDE e outros programas gráficos funcionem adequadamente.
    iptables -A INPUT -i lo -j ACCEPT
    # Impede a abertura de novas conexões, efetivamente bloqueando o acesso
    # externo ao seu servidor, com exceção das portas e faixas de endereços
    # manualmente especificadas anteriormente.
    iptables -A INPUT -p tcp --syn -j DROP
    echo "Regras de firewall Ativadas"
}
parar(){
    iptables -F
    iptables -P INPUT ACCEPT
    iptables -P OUTPUT ACCEPT
    echo "Regras de firewall desativadas"
}
case "$1" in
    "start") iniciar ;;
    "stop") parar ;;
    "restart") parar; iniciar ;;
    *) echo "Use os parâmetros start ou stop"
esac
```

Figura 11 - Regra do *firewall*.

É através deste arquivo que o *firewall* pode ser iniciado através do comando "/etc/init.d/*firewall start*" ou pode ser parado através do comando "/etc/init.d/*firewall stop*" no terminal, e como pode-se ver na Figura 12 a regra de *firewall* foi ativado conforme esperado.

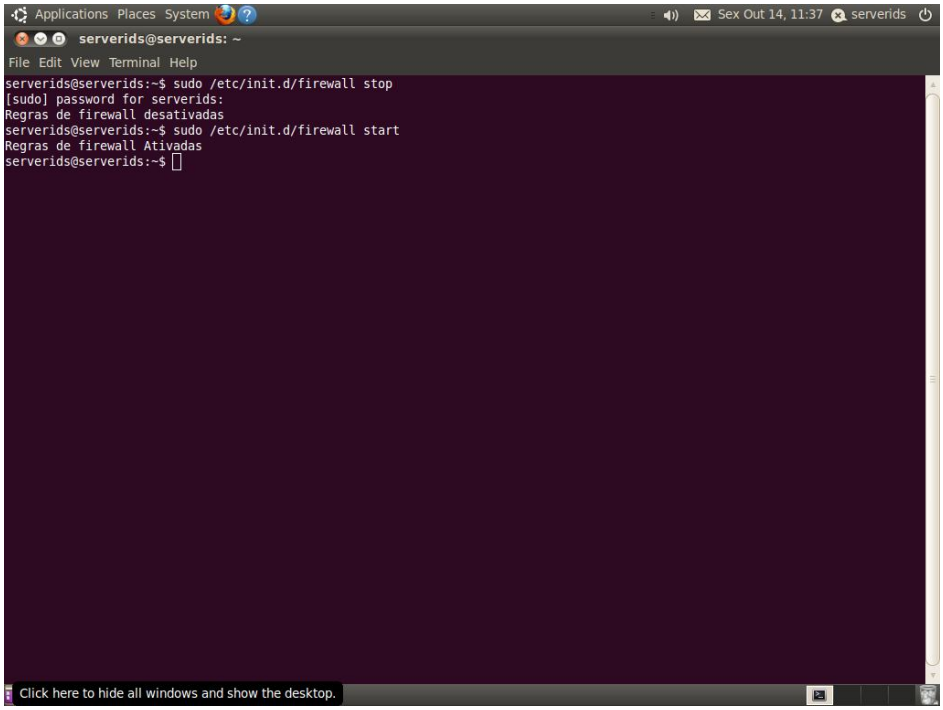

Figura 12 - Regra do *firewall* sendo iniciada via terminal.

Através da Figura 13 pode-se observar a tela de login do *Snorby*, que é a página de visualização dos alertas gerados pelo *Snort*, no qual é acessada pelo endereço http://10.36.96.200 e feita em qualquer navegador de qualquer micro da rede mediante validação de usuário e senha.

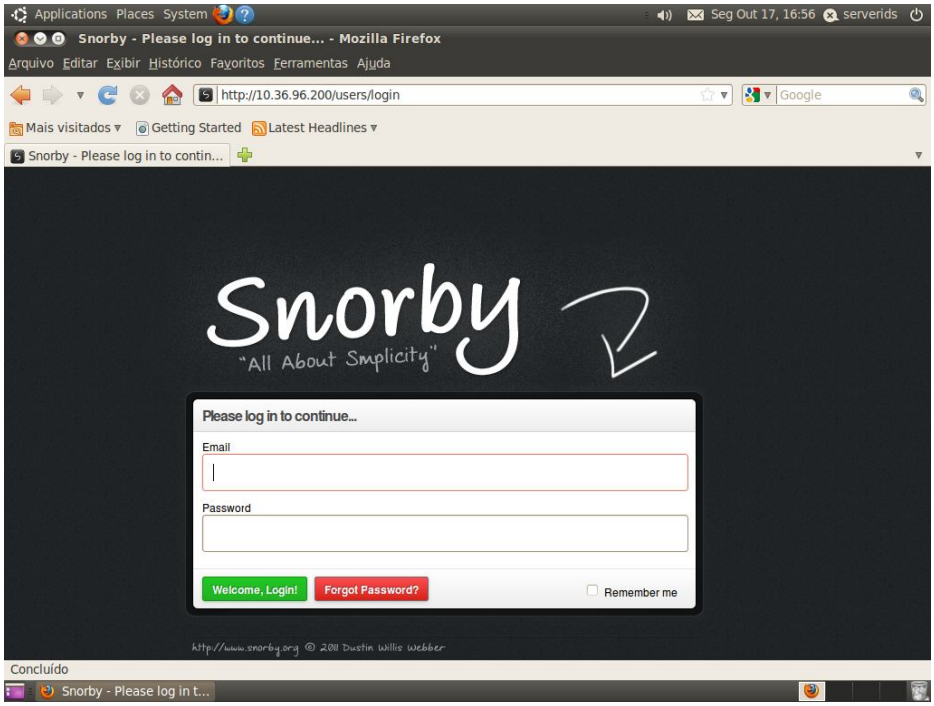

Figura 13 - Tela de login do Snorby.

Na Figura 14, pode-se observar a tela inicial, onde consegue ter a visualização da quantidade de alertas gerados pelo *Snort* e as estatísticas de intrusão no sistema.

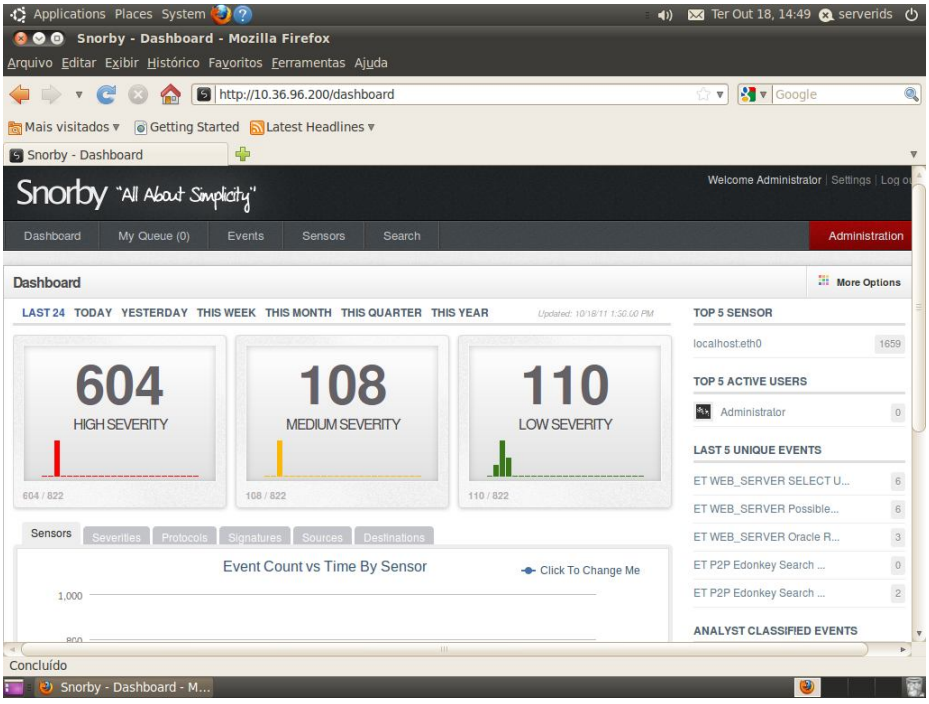

Figura 14 - Tela inicial do Snorby.

Com o *Snorby* é possível ter um grande controle da rede, identificando possíveis ataques e ter a visualização dos eventos, como pode ser visto na Figura 15 que mostra a quantidade de eventos no mês, podendo também ser visualizados somente os eventos detectados nas últimas 24 horas, no dia, no dia anterior, na semana, no trimestre e no ano.

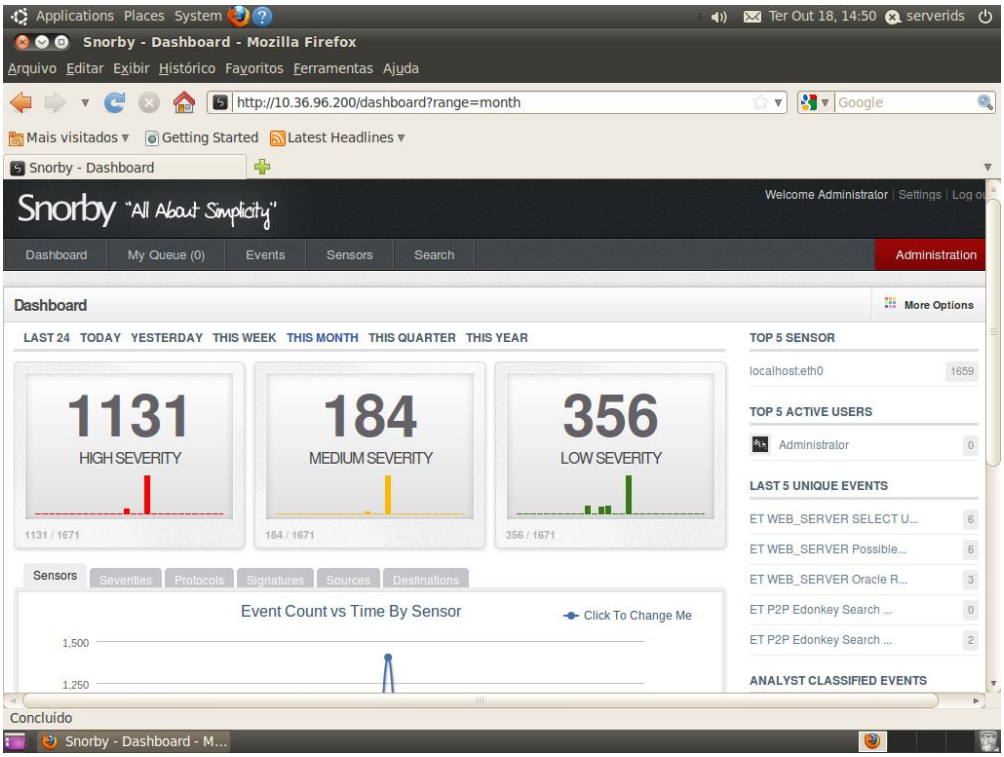

Figura 15 - Visualização dos eventos no mês.

A Figura 16 mostra os eventos considerados de elevada severidade, no qual também consegue visualizar o IP de origem, o IP de destino, o tipo, a hora e o dia dos eventos.

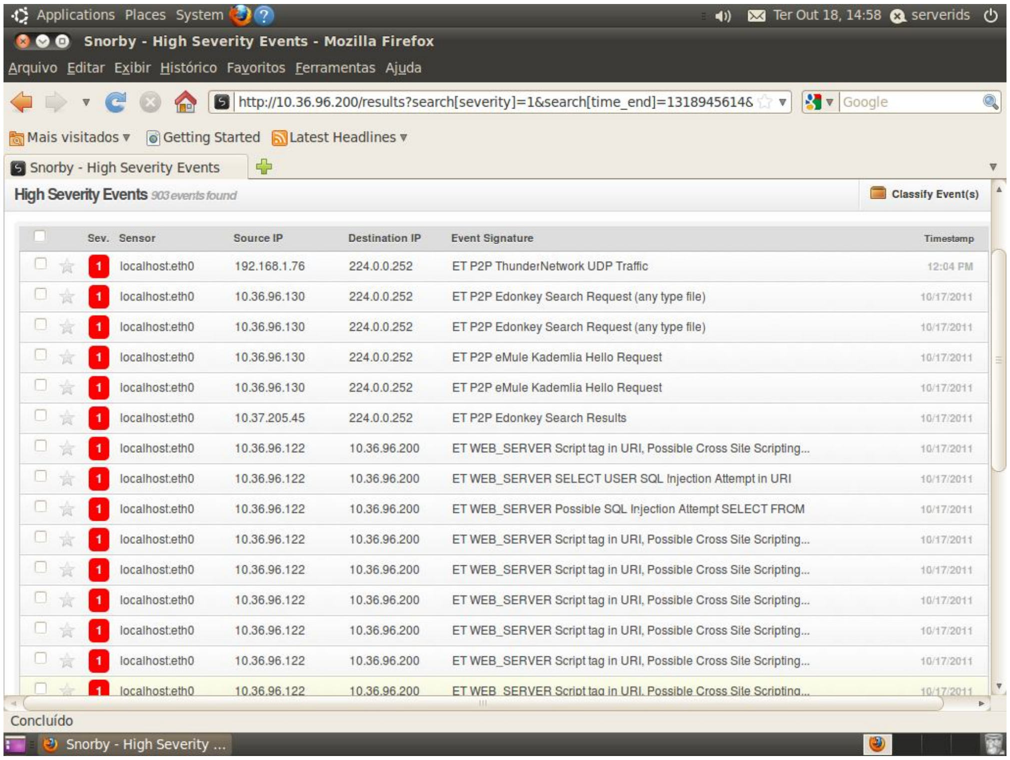

Figura 16 - Eventos considerados como de elevada severidade.

Como na Figura 16 a Figura 17 consegue visualizar o IP de origem, o IP de destino, o tipo, a hora e o dia dos eventos, porém mostra os eventos considerados como de severidade média.

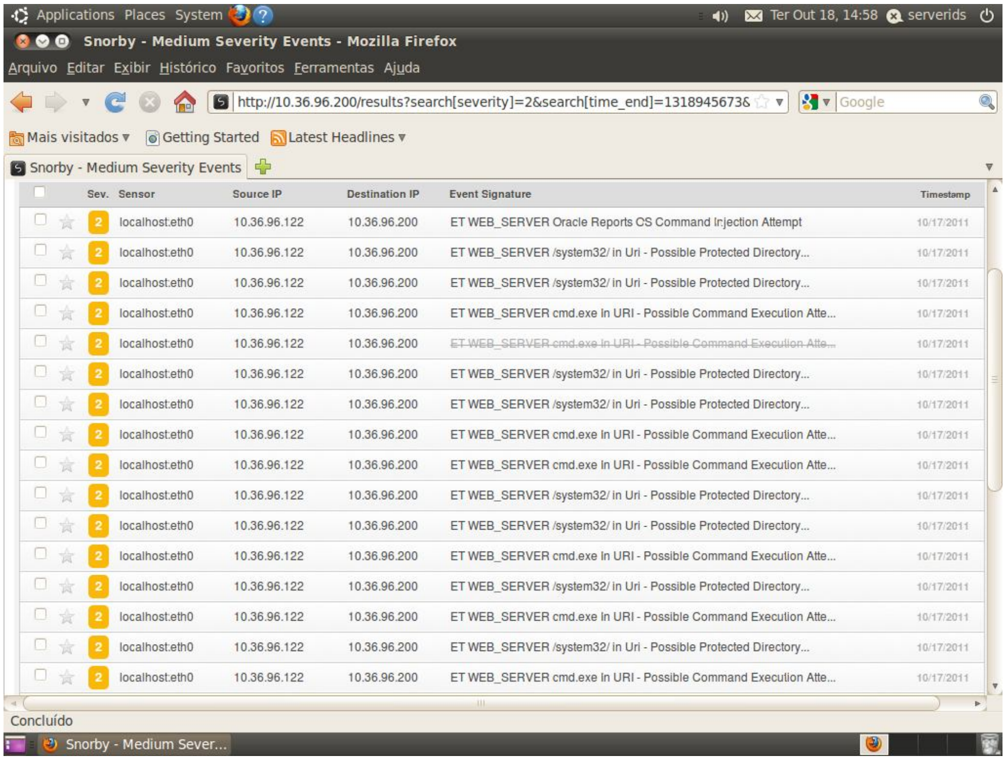

Figura 17 - Eventos considerados como de severidade média.

Bem como a Figura 16 e a Figura 17, a Figura 18 consegue-se ter a visualização do IP de origem, o IP de destino, o tipo, a hora e o dia dos eventos, mas demonstra somente os eventos de baixa severidade.

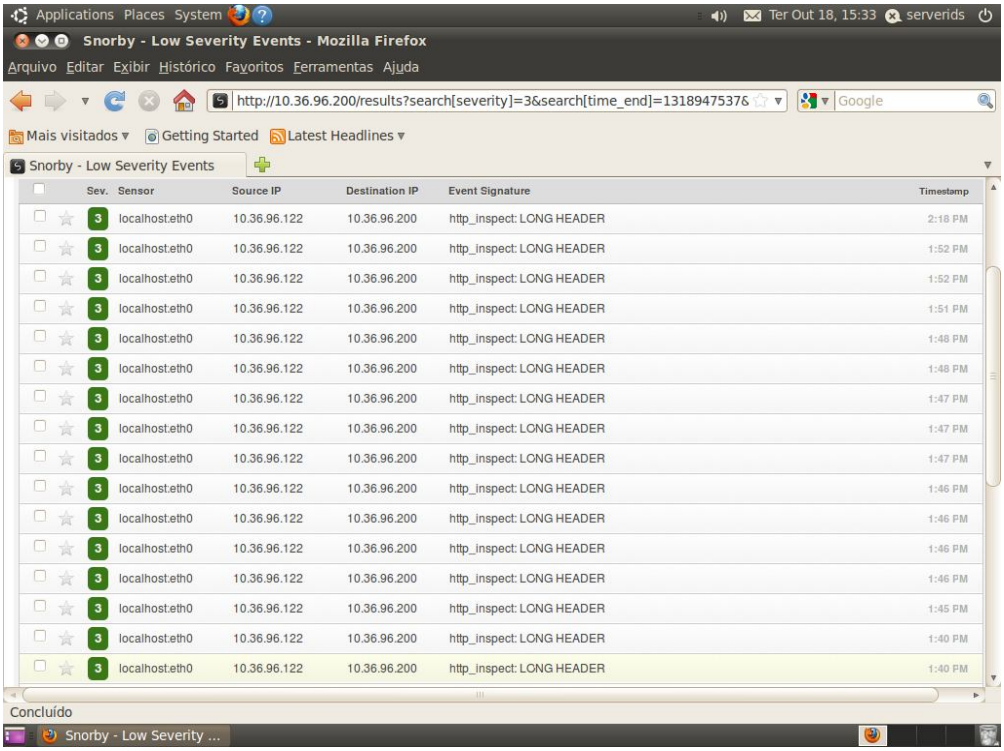

Figura 18 - Eventos considerados de baixa severidade.

Já na Figura 19 consegue ter uma visualização detalhada da configuração do sistema.

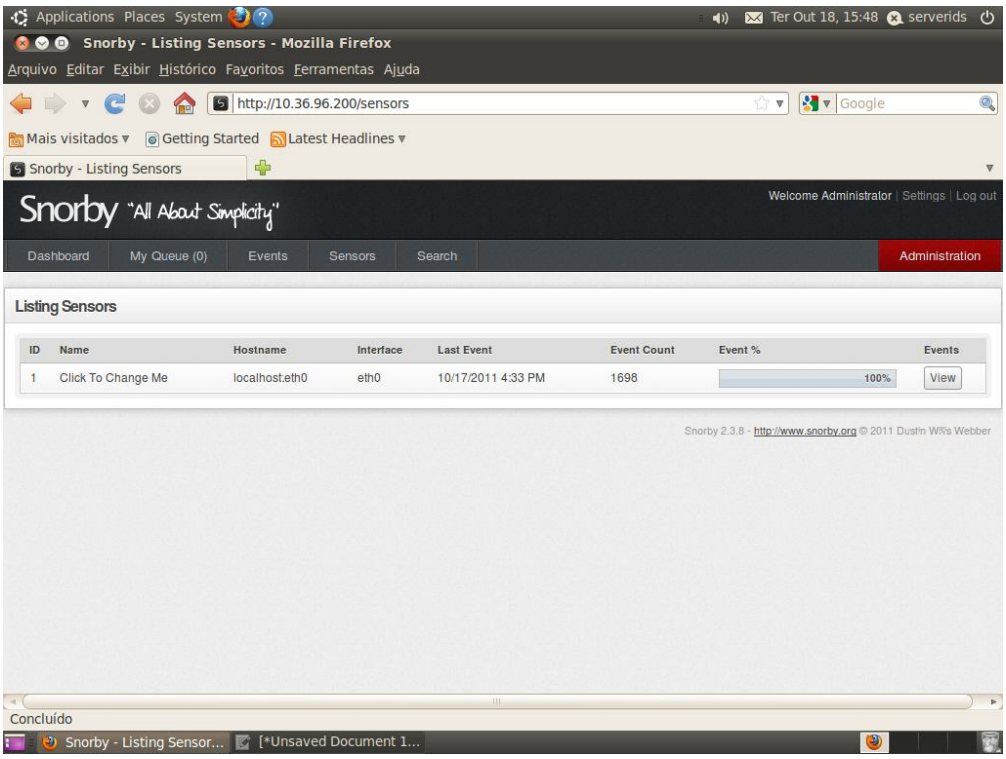

Figura 19 - Listagem dos detalhes do sensor.

Na tela principal do *Snorby* é possível ter acesso as estatísticas de intrusões detectadas como pode ser visto na Figura 20, que mostra através de um gráfico a estatística dos eventos detectados no mês.

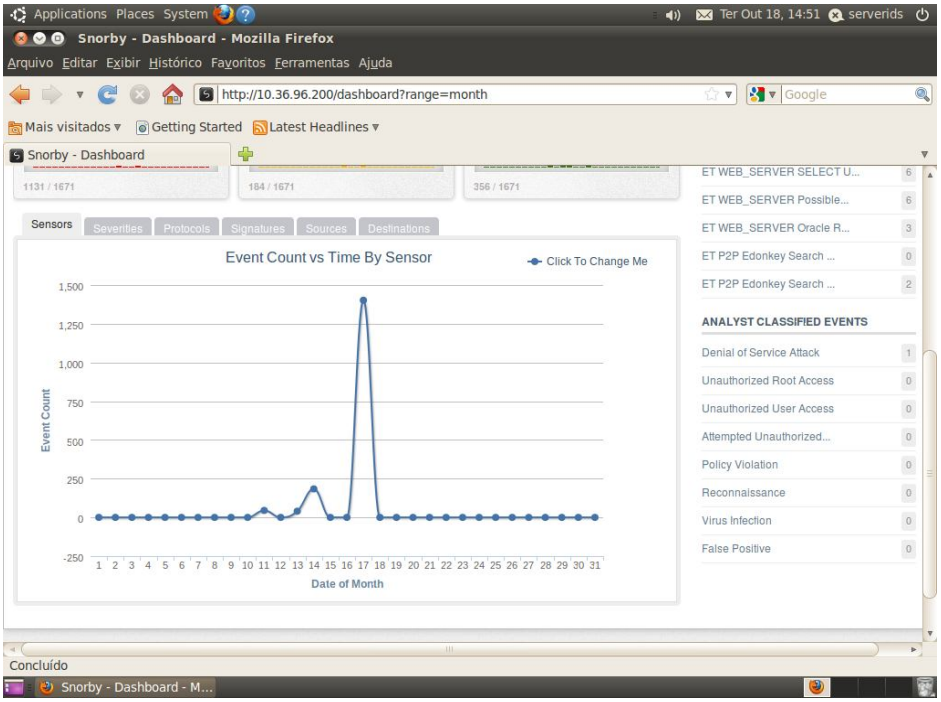

Figura 20 - Estatística de eventos detectados no mês.

Na Figura 21 consegue ter a visualização de um gráfico onde contém a porcentagem de cada tipo de assinatura detectada.

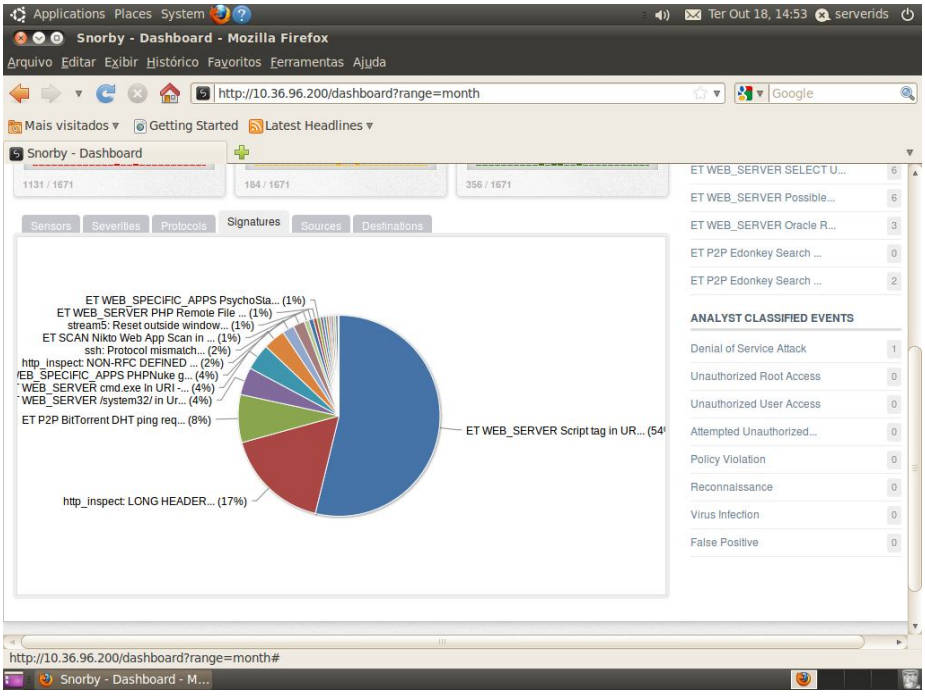

Figura 21 - Estatísticas dos tipos de assinaturas detectadas.

A Figura 22 mostra através de um gráfico a estatística dos IP dos microcomputadores fonte de todas ameaças detectadas.

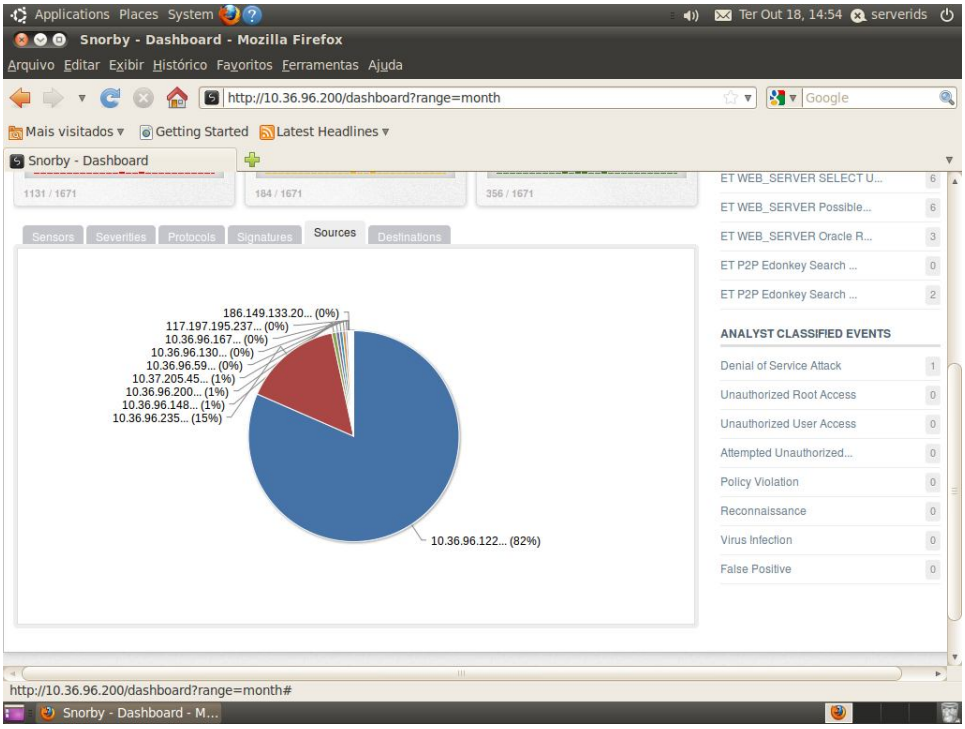

Figura 22 - Estatística de IP fonte das ameaças detectadas.

Já na Figura 23 mostra a estatística dos IP dos microcomputadores que realizaram as ameaças detectadas pelo *Snort*.

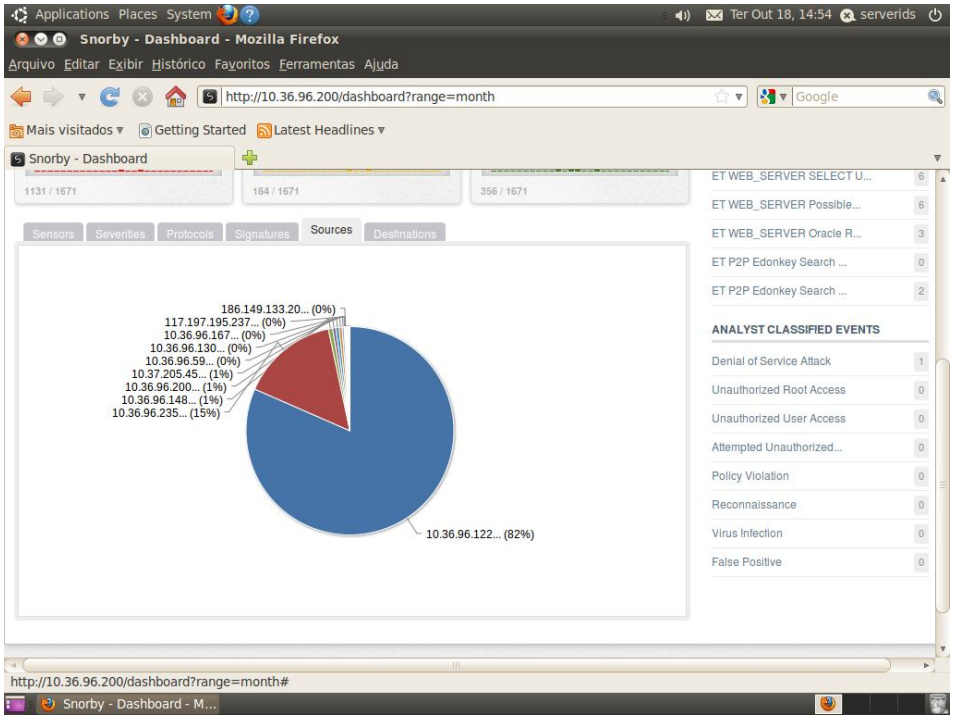

Figura 23 - Estatísticas de IP destino das ameaças detectadas.

Para realizar os testes do *firewall* e do *IDS* foi utilizado uma ferramenta para exploração da rede que utiliza a técnica de *port scanning* conhecida como Nmap, e como pode ser visto na Figura 24 foi instalada via terminal no Sistema Operacional Ubuntu 11.04, e executado o comando direcionado tanto para a máquina alvo que se encontra o *IDS* quanto a que se encontra o *firewall*.

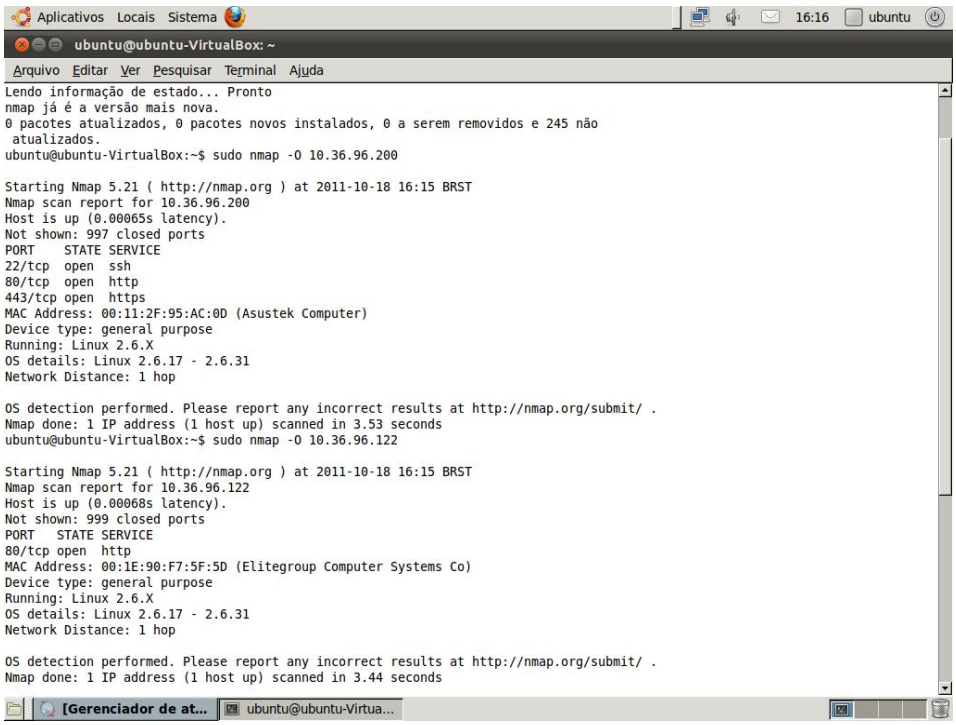

Figura 24 - Instalação Nmap e execução de comandos.

Com a execução do comando do Nmap, o *Snort* imediatamente detectou a ação e foi possível visualizar o evento como é mostrado na Figura 25.

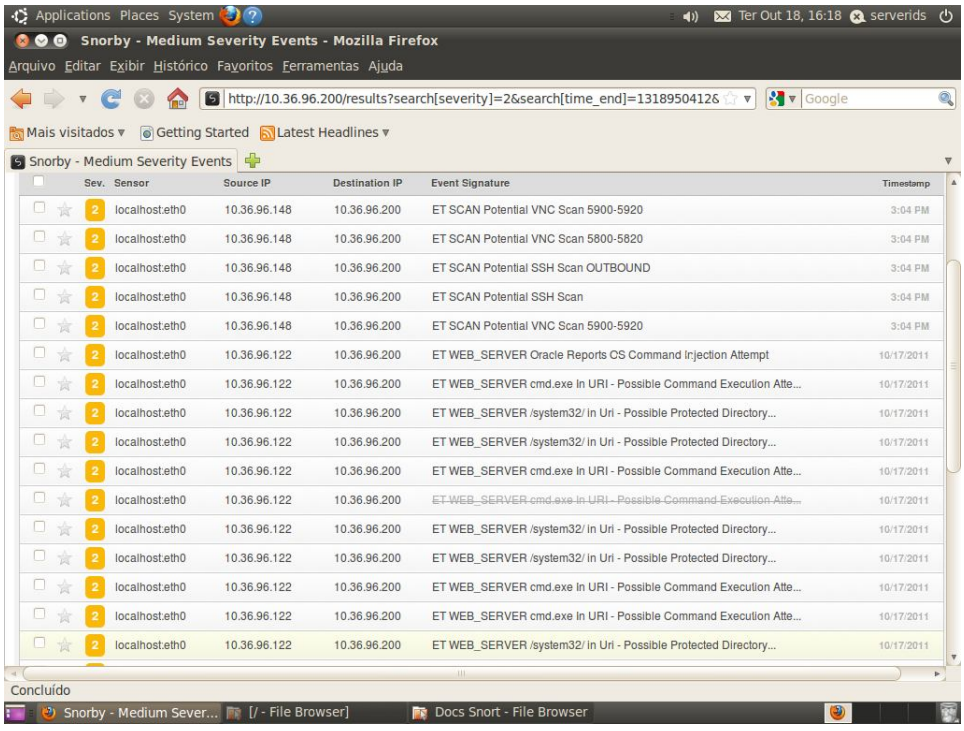

Figura 25 - Detecção dos testes por parte do *Snort*.

Na Figura 26 mostra com um maior detalhe o evento detectado como possível ameaça.

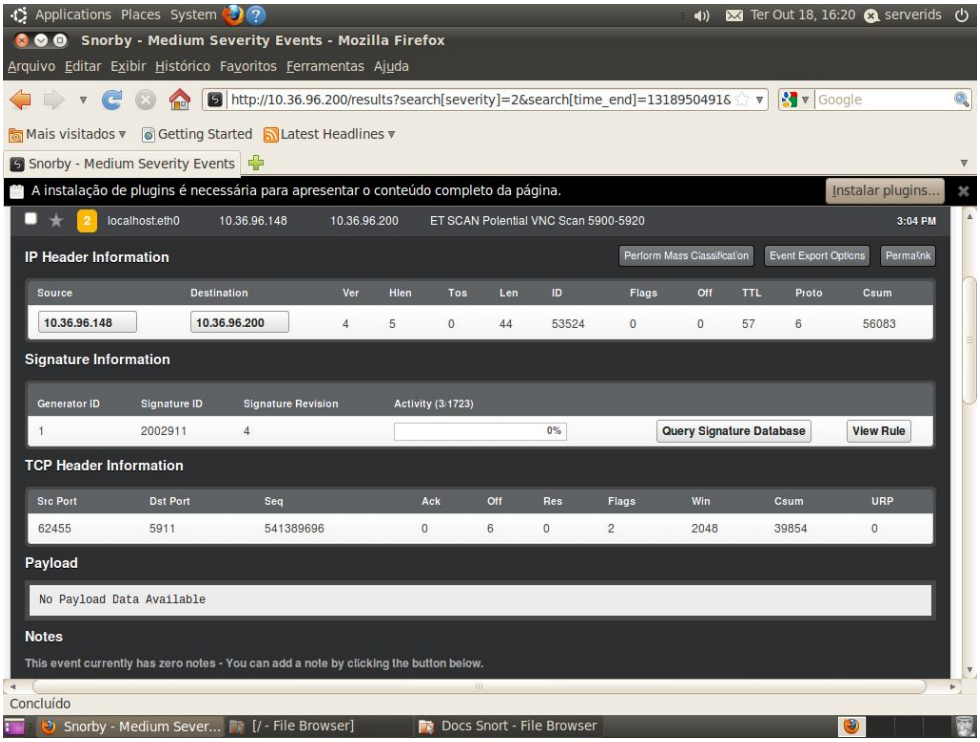

Figura 26 - Detalhes da detecção no *Snort*.

E também foi utilizado para testes o Nessus que fez a verificação de falhas e vulnerabilidades de segurança na rede, sendo que foi instalado o servidor do Nessus em um micro com Sistema Operacional *Windows* 7 *Ultimate* e o cliente podendo ser visto em qualquer outro micro da rede sendo que a verificação dita é feita pelo servidor.

A Figura 27 mostra o servidor Nessus iniciado no microcomputador com Sistema Operacional *Windows* 7.

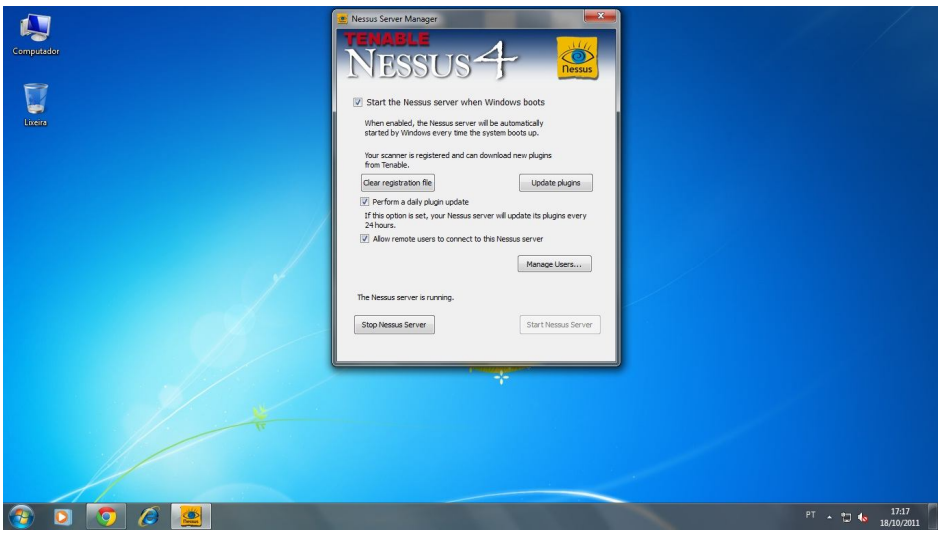

Figura 27 - Servidor Nessus iniciado.

Na Figura 28 consegue ter a visualização da tela de acesso do Nessus, onde pode ser feita em qualquer microcomputador da rede mediante validação de usuário e senha.

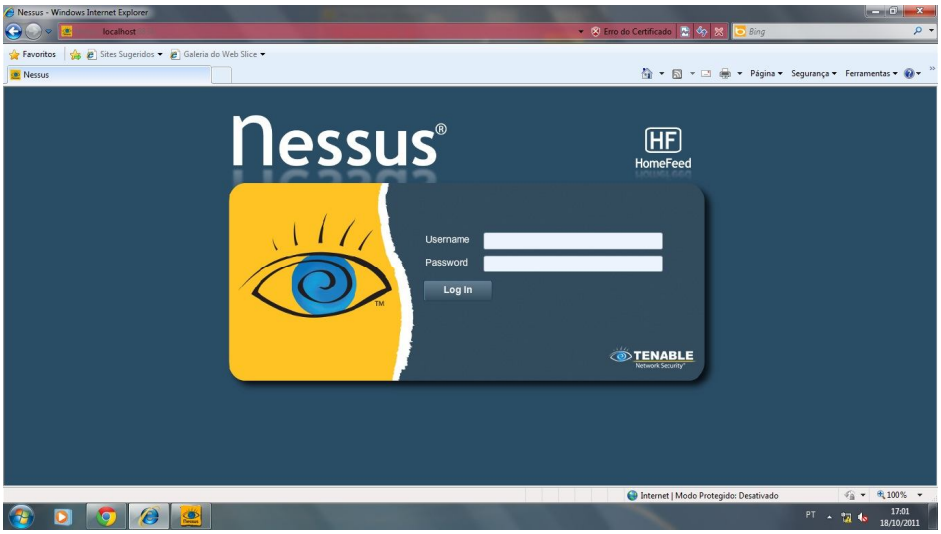

Figura 28 - Tela de login do Nessus.

Com o Nessus iniciado, é possível visualizar na Figura 29 os testes de falhas e busca por vulnerabilidades criados, sendo esses testes destinados tanto para o microcomputador onde se encontra o *firewall*, como também para o que se encontra o IDS.

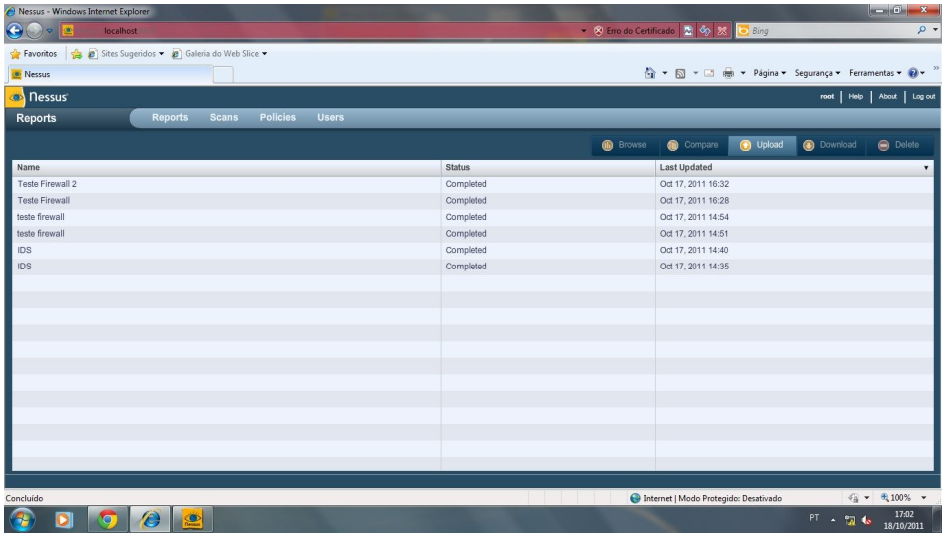

Figura 29 - Testes gerados no Nessus.

Na Figura 30 é possível ter acesso com maior detalhe aos resultados obtidos a partir dos testes gerados no Nessus.

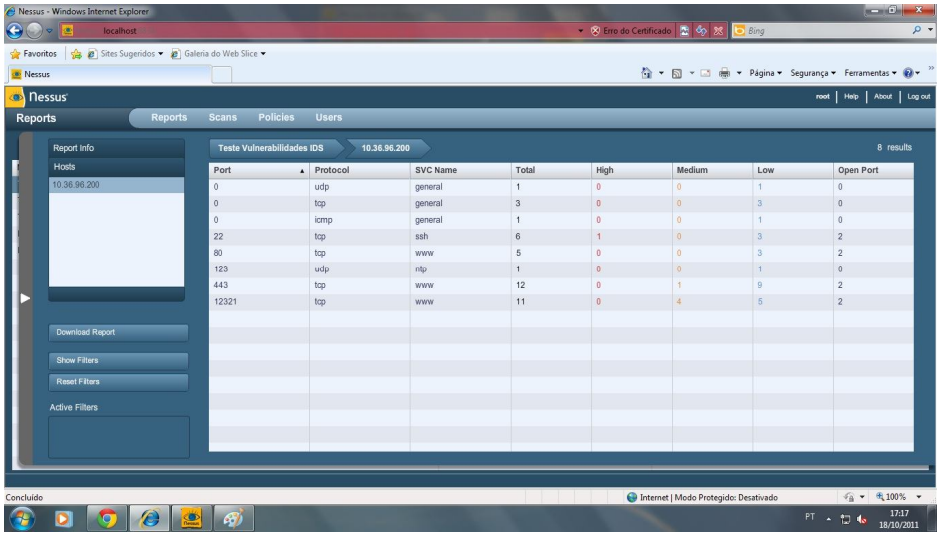

Figura 30 - Detalhes do teste no Nessus.

Já na Figura 31 é possível visualizar a detecção imediata da ação do Nessus pelo *Snort*.

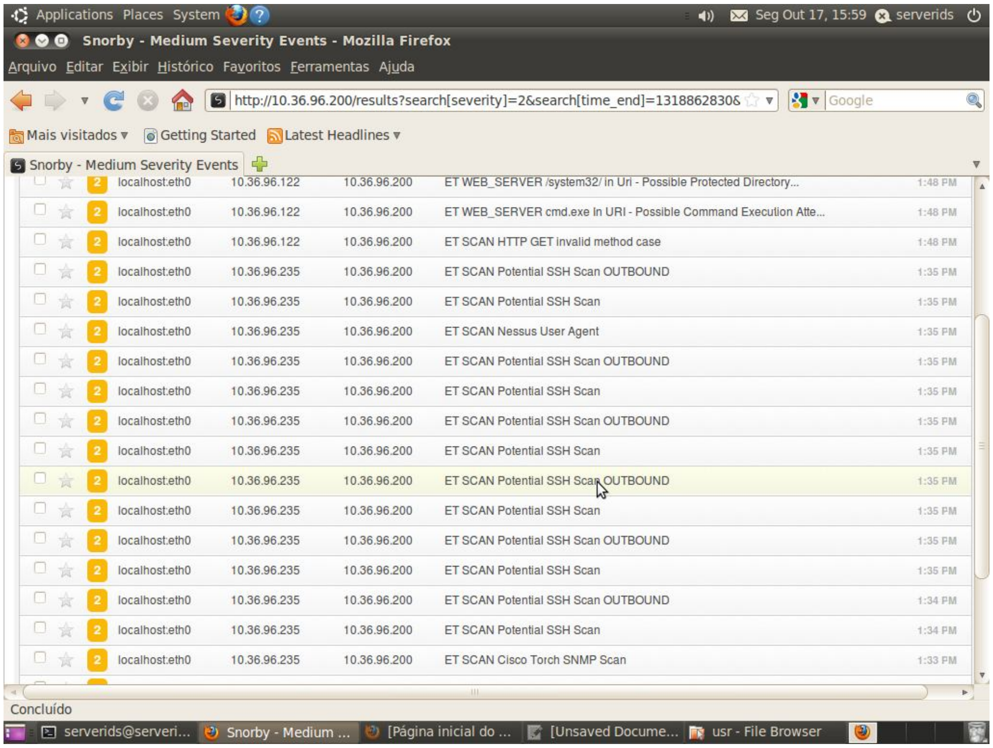

Figura 31 - Detecção dos testes feito no Nessus por parte do *Snort*.

6 CONCLUSÃO

A Internet registra um crescimento contínuo, como também a utilização das redes de computadores, e sua consequente abertura para um mundo externo, os sistemas computacionais acabaram se tornando cada vez mais difíceis de serem protegidos, pois infelizmente com o crescimento da Internet, cresceu também, e em uma escala muito maior, os problemas relacionados a segurança de redes de computadores. Assim, uma das grandes preocupações dos administradores desses sistemas é criar barreiras de proteção contra invasores externos.

Com isso este trabalho teve por objetivo demonstrar, através do estudo teórico e prático sobre *firewall* e também do estudo sobre Sistemas de Detecção de Intrusão (SDI), seus conceitos, métodos e funcionalidades e a importância do uso dessas ferramentas como meio de prover segurança na rede como um todo.

Longe de ser uma aplicação perfeita, o *firewall* quando não bem configurado e atualizado apresenta vulnerabilidades e o *IDS* surge como uma ótima ferramenta de administração e um importante auxiliar na melhoria da segurança, permitindo uma constante monitoração das atividades de tentativas de ataques na rede.

Dessa forma, não basta simplesmente configurar um *firewall* para criar barreiras de proteção contra um mundo externo, pois pode-se verificar nos testes que mesmo utilizando um *firewall* na rede, consegue-se ter facilmente informações da rede com simples ataques, com isso demonstra que *firewall* quando não bem configurados e atualizados, acabam possuindo vulnerabilidades. Com isso Sistemas de Detecção de Intrusão devem estar presentes nas redes de computadores e em constante monitoria, buscando informações que possam identificar ataques a rede.

A instalação de um *IDS* é importante à medida que, a cada ano, o número de ataques cresce de maneira assustadora. É importante detectar o ataque antes que ele possa realizar seu objetivo na estrutura de rede.

Mas o *IDS* não deve substituir o *firewall* na tarefa de proteger as redes de dados, mas sim deve agir em conjunto com ele e tanto como outras ferramentas de segurança de maneira que consiga a garantir a segurança da rede, pois cada ferramenta acaba desempenhando uma tarefa específica.

É importante ressaltar que a segurança total de uma rede de computadores é praticamente impossível, mas os riscos, falhas e vulnerabilidades podem ser controladas, quando o estudo e implementação são utilizados corretamente.

Os resultados obtidos nos testes tanto de verificar as vulnerabilidades do *firewall* e de identificação dos ataques gerados como testes ficaram dentro das expectativas, sendo satisfatórias, pois os testes gerados acabaram sendo detectados pelo *IDS*. A partir destes resultados é possível partir para testes mais complexos.

Contudo, é importante ressaltar que mesmo tendo tanto um dispositivo de *firewall* e um *IDS* configurados de maneira em conjunto quanto de maneira única, não garante que um ataque seja totalmente impedido ou identificado, pois sempre novas formas de ataque ou novos pontos fracos são descobertos, no entanto, com a constante monitoração de tudo o que passa pelo *firewall* e pelo *IDS*, permite ao administrador da rede criar maneiras de dificultar ainda mais a ação dos ataques, seja implementando novos recursos ou criando novas regras.

Por fim deseja-se contribuir para que os administradores de redes possam usar esse material como base para as administrações de redes conectadas a Internet e com isso possa diminuir os riscos de ataques.

REFERÊNCIAS

BARBOSA, Ákio Nogueira. **Um Sistema para análise ativa de comportamento de *Firewall***. São Paulo, 2006.

BARBOSA, André. **Sistemas de Detecção de Intrusão**. Rio de Janeiro, 2000.

BERNARDES, Mauro Cesar. **Avaliação do Uso de Agentes Móveis em Segurança Computacional**. São Carlos, 1999.

CAMY, Alexandre Rosa; SILVA, Evandro R. N.; RIGHI, Rafael. **Seminário de Firewalls**. Florianópolis, 2003.

CAMPELLO, Rafael Saldanha; WEBER, Raul Fernando. **Sistemas de Detecção de Intrusão**. Florianópolis, 2001.

CERT. Centro De Estudos Respostas E Tratamento De Incidentes De Segurança No Brasil. **Cartilha de segurança para internet 3. 1 :** parte I: conceitos de segurança. 2006. Disponível em: <http://cartilha.cert.br/conceitos/>. Acesso em: 4 jun. 2011.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

INOKOSHI, Rodrigo Kiyoshi. **Avaliação de *Firewall* e Sistema de Detecção de Intrusão baseado em software livre**. Jaguariúna, 2007.

NEMETH, Evi; SNYDER, Garth; HEIN, Trent R. **Manual Completo do LINUX:** guia do administrador. 2. ed. São Paulo: Pearson Prentice Hall, 2007.

NIC BR SECURITY OFFICE. Práticas de segurança para administradores de redes internet.**:**versão 1.2. **Cert.br**, 2003. Disponível em: <http://www.cert.br/docs/seg-adm-redes/seg-adm-redes.pdf >. Acesso em: 16 maio 2011.

OLIVEIRA, Sabrina Vitório. **Perícia Forense em Sistemas GNU/Linux**. Vitória, 2007.

ROCHA JUNIOR, Vamberto de Freitas Rocha. **Estudo e implementação de *Firewall* em ambientes corporativos**. João Pessoa, 2010.

SIEWERT, Vanderson C.. ***Firewall* suas características e vulnerabilidades**. Florianópolis, 2008.

SILVA, Glaydson Mazioli da. **Guia Foca GNU Linux:** versão avançada. 2005. Disponível em: <http://www.guiafoca.org>. Acesso em: 20 maio 2011.

STEFFEN JUNIOR, Julio. **Sistema de Detecção de Intrusão**. Novo Hamburgo, 2003.

WANNER, P. C. Herrmann. **Ferramentas de Injeção de Falhas para Avaliação de Segurança**. Porto Alegre, 2003.

ZWICKY, Elizabeth D.; COOPER Simon; CHAPMAN, D. Brent. **Construindo *Firewall*s para Internet**. 2. ed. Rio de Janeiro: Campus, 2000.

Sistema de Detecção de Intrusão associado a um Firewall para Segurança de Redes

Vitor Muta Barreto, Henrique Pachioni Martins, Andre Luiz Ferraz Castro, Kelton A. Pontara da Costa

Centro de Ciências Exatas e Sociais Aplicadas
Universidade Sagrado Coração (USC) – Bauru, SP – Brasil

vitor.barreto@usc.br, henrique.martins@usc.br, andre.castro@usc.br, kelton.costa@usc.br

Resumo. *Nos dias atuais, com o crescente uso da Internet como meio global de comunicação, tem-se verificado um significativo aumento com relação ao número de vulnerabilidades e ataques existentes nos sistemas computacionais, com isso a segurança se tornou um ponto crucial em todos os sistemas computacionais. Nesse sentido, o objetivo deste trabalho foi apresentar e implementar, de forma prática e objetiva uma associação entre dois importantes sistemas de segurança: Sistemas de Detecção de Intrusão (IDS) e firewall, pois ambas tecnologias são utilizadas separadamente na maioria dos casos, sendo que o propósito desta associação foi se obter uma efetiva solução de segurança em conjunto.*

Abstract. *Nowadays, with the relevant Internet use as a global way of communication, we have verified a real increasing vulnerable number of the existent attacks in the computer systems and so, with it the security has become the crucial point in all the computer system. So, the main objective of this research was to show and implement in a practical and objective way, an association between two important security systems: Intrusion Detection System (IDS) and Firewall, the reason being both technologies are used separately in most cases and the aim of this association was to obtain an effective security solution in its total.*

1. Introdução

Na visão de Rocha Junior (2010), com a crescente evolução tecnológica tornou-se indispensável que as empresas possuam uma estrutura que consiga oferecer a seus funcionários acesso a Internet, como para fornecer acesso externo as suas aplicações para servir seus parceiros e clientes. Nesse contexto, milhares de empresas nos dias atuais possuem uma infraestrutura de acesso a Internet. Com as inúmeras vantagens que este acesso acaba proporcionando como a conectividade entre redes distintas, correio eletrônico, comunicação eficiente e barata entre pessoas, etc., vieram também inúmeros problemas, principalmente no que diz respeito à segurança dos dados.

Com o aumento dos ataques as empresas, profissionais da área da tecnologia da informação visando proteger informações e dados, buscam utilizar diversas ferramentas que auxilie na segurança como firewalls, Antivírus, Sistemas de Detecção de Intrusão, etc., para ter a possibilidade de detectar e bloquear invasões e possíveis danos causados por invasores.

O firewall vem a ser uma barreira de proteção, que controla o tráfego de dados entre o computador e a Internet, e seu principal objetivo é permitir somente a

transmissão e a recepção de dados autorizados. Com essas características se tornou uma ferramenta de defesa cada vez mais usada, no entanto sua configuração é muito complexa, podendo resultar em erros, assim apresentar vulnerabilidades para ataques.

Segundo Bernardes (1999), apesar do uso dos diversos esquemas de segurança existentes, ainda existe a possibilidade de ocorrências de falhas nestes esquemas e, portanto, é desejável a existência de sistemas capazes de realizar a detecção de tais falhas e informar o administrador da rede. Tais sistemas são conhecidos como Sistemas de Detecção de Intrusão (SDI).

Diante desse panorama, este trabalho tem como objetivo unir um Sistema de Detecção de Intrusão a um firewall e possibilitar um gerenciamento mais fácil e eficiente de uma rede, onde possa analisar as vulnerabilidades de um firewall, sendo esse um dos problemas enfrentados em ambientes de redes, onde através da implantação da tecnologia de IDS, verificar se realmente a mesma auxilia na segurança e acaba por detectar ataques a rede, assim demonstrando as vulnerabilidades de um firewall.

2. Objetivos

2.1. Objetivo Geral

Verificar a vulnerabilidade de um firewall por meio da implantação de um Sistema de Detecção de Intrusão (SDI).

2.2. Objetivos Específicos

- Implantar um Sistema de Detecção de Intrusão (SDI);

- Avaliar a segurança de um firewall por meio de um Sistema de Detecção de Intrusão, analisando as tentativas de invasão a rede oriundos de simulações.

2.3. Justificativa

A utilização de uma ferramenta para analisar as tentativas de ataques à rede auxilia na segurança de redes de computadores quanto a verificar as vulnerabilidades que estão sendo exploradas no firewall, com isso faz-se necessário à utilização de ferramentas de monitoração dessas redes de forma a se identificar possíveis atividades suspeitas e tomar as medidas cabíveis em tempo hábil. Através desta análise é possível saber de onde está partindo uma invasão podendo assim bloquear a comunicação com a origem, evitando um possível ataque a rede e ao firewall.

SDI (Sistema de Detecção de Intrusão) é um sistema onde é possível detectar e notificar as tentativas de intrusão, podendo emitir alarmes ou executar uma ação automática, assim com a sua implantação o administrador da rede consegue tomar alguma atitude em relação ao firewall, ou até mesmo outro programa ou processo seja iniciado automaticamente.

Portanto com a crescente complexidade das ameaças digitais tem exigido a associação de várias tecnologias de segurança atualmente disponíveis, sendo que a associação do Sistema de Detecção de Intrusão com o firewall pode-se criar uma solução de segurança integrada, onde com a implementação do IDS é possível verificar os ataques à rede e com isso identificar as vulnerabilidades que o firewall possui, podendo essa vulnerabilidade detectada no firewall ser imediatamente solucionada pelo

administrador da rede.

3. Revisão de Literatura

Firewall é um termo inglês que traduzido para Língua Portuguesa significa parede de fogo, que segundo Zwicky, Cooper e Chapman (2000, p. 104) é “um componente ou conjunto de componentes que restringe o acesso entre uma rede protegida e a Internet ou entre outros conjuntos de rede.”

“Um mecanismo muito utilizado na prática para aumentar a segurança das redes de computadores, protegendo-as de ataques externos, é o firewall.” (BERNARDES, 1999, p. 16).

Em relação ao firewall NIC (2003) relata que um firewall bem configurado é um instrumento importante para implantar a política de segurança da sua rede. Ele pode reduzir a informação disponível externamente sobre a sua rede, ou, em alguns casos, até mesmo barrar ataques a vulnerabilidades ainda não divulgadas publicamente. Por outro lado, firewalls não são infalíveis. A simples instalação de um firewall não garante que sua rede esteja segura contra invasores. Um firewall não pode ser a sua única linha de defesa, ele é mais um dentre os diversos mecanismos e procedimentos que aumentam a segurança de uma rede.

3.1 Tipos de firewalls

Existem três tipos principais de firewall: firewall a nível de pacote, que funciona nas camadas de rede e de transporte, firewall a nível de pacotes baseado em estados e o firewall a nível de aplicação, que funciona nas camadas de aplicação, sessão e transporte.

3.1.1 Firewalls bastion host

Os bastion hosts são os servidores que possuem instalados serviços a serem oferecidos para a Internet.

Um firewall ou um roteador podem ser considerados bastion hosts, sendo que provê os recursos permitidos segundo a política de segurança da empresa. Pode ser projetado com a finalidade de ser servidor web, servidor FTP, entre outras funções que estiverem disponíveis a usuários de fora da proteção interna da rede. (CAMY; SILVA; RIGHI, 2003).

3.2 Arquiteturas

A posição que o firewall será implantado na topologia de rede terá um impacto significativo no nível de segurança da rede da empresa. Todo o dado que trafegue de uma rede para outra deve passar obrigatoriamente pelo firewall. (ROCHA JUNIOR, 2010).

Assim a seguir poderão ser apresentadas três possíveis arquiteturas para a implantação de um firewall, a saber: Dual-homed Host, Screened Host e Screened Subnet.

3.3 Firewall iptables

O Iptables é um firewall em nível de pacotes e funciona baseado no endereço/porta de

origem/destino do pacote, prioridade, etc. Ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar. Em firewalls mais restritivos, o pacote é bloqueado e registrado para que o administrador do sistema tenha conhecimento sobre o que está acontecendo em seu sistema. (SILVA, 2005).

Como surge várias ameaças novas diariamente, o firewall IpTables também está sujeito à falhas e vulnerabilidades, e com isso necessita sempre estar se atualizando. (SIEWERT, 2008).

3.3.1 A configuração do iptables

A configuração de um firewall iptables é realizada por uma série de comandos (regras) que são interpretados pelo kernel do sistema operacional. Tais comandos podem ser inseridos diretamente no shell do sistema operacional ou através de arquivos de scripts (arquivos texto). (INOKOSHI, 2007).

3.4 VULNERABILIDADES

Vulnerabilidade é definida como uma falha no projeto, implementação ou configuração de um software ou sistema operacional que, quando explorada por um atacante, resulta na violação da segurança de um computador.

Há diversos tipos de ataques que exploram as falhas das vulnerabilidades, a exploração de tais vulnerabilidades pode ser utilizada com as mais diferentes finalidades, algumas inclusive, podem deixar computadores, firewalls ou outros equipamentos de rede completamente inutilizáveis. (BARBOSA, 2006).

3.5 ATAQUES

Todo sistema está sujeito a diferentes tipos de ameaças, sejam elas internas ou externas, acidentais ou maliciosas. Explorando certas vulnerabilidades ou brechas, diversos tipos de ataques são desencadeados atualmente, desde tentativas simples de negação de serviço, até ataques sofisticados que utilizam recursos distribuídos. (CAMPELLO; WEBBER, 2001).

O aumento dos casos de insegurança da informação existe devido aos ataques como, do tipo port scanning, as ferramentas para injeção de pacotes, DOS(Denial of Service), IP spoofing, sniffing de pacotes, ferramentas para varredura de vulnerabilidades, sendo elas descritas a seguir.

3.6 SISTEMAS DE DETECÇÃO DE INTRUSÃO

Do inglês Intrusion Detection System, sendo programa, ou um conjunto de programas, cuja função é detectar possíveis ataques.

Campello e Weber (2001) definem detecção de intrusão como sendo o processo de identificar e responder a atividades maliciosas dirigidas a computadores e recursos de rede e descreve, dentre outras definições, que detecção de intrusão é a tarefa de coletar informações de uma variedade de fontes - sistemas ou redes - e então analisá-las buscando sinais de intrusão e de mau-uso. No primeiro caso, o termo detecção de intrusão é usado tanto no sentido de detecção propriamente dita como na reação a essa atividade. Isso amplia a funcionalidade dos IDSs, impondo a eles a difícil tarefa de reagir aos ataques detectados.

Segundo Bernardes (1999) as funcionalidades de um sistema de detecção tornam-se de vital importância na medida em que fornecem meios de inferir sobre o conteúdo das conexões permitidas e detectar as que apresentem um comportamento suspeito ou não condizente com a política de segurança implantada.

3.6.1 Tipos de IDS

Existem dois tipos de IDS: o baseado em Rede e o baseado em Host. Qual modelo escolher depende da estrutura de cada empresa, podendo até ser instalado os dois modelos trabalhando na mesma rede. (STEFFEN JUNIOR, 2003).

O baseado em rede de acordo Bernardes (1999) são "Sistemas baseados em rede examinam os dados que trafegam pela rede através da monitoração on-line dos pacotes." Já segundo Barbosa (2000) os IDS baseados em host analisam sinais de intrusão na máquina nos quais estão instalados, eles frequentemente usam os mecanismos de log do sistema operacional e estão muito ligados aos recursos do sistema. Eles agem procurando por atividades não usuais em: tentativas de login, acesso à arquivos, alterações em privilégios do sistema, etc.

Independente da arquitetura ou do método utilizado para a detecção, todo IDS possui componentes em comum. Cada um desses componentes desempenha um papel importante na tarefa de identificar ações consideradas danosas ao sistema. (CAMPELLO; WEBBER, 2001). Assim, deve-se conhecer a anatomia de um IDS como pode-se ver na Figura 1.

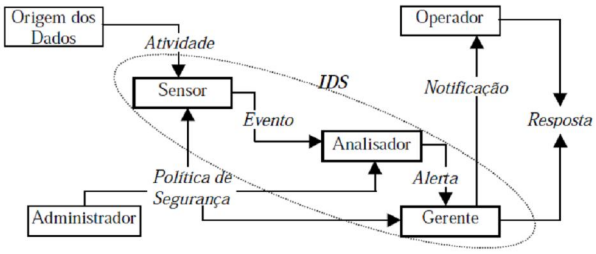

Figura 32 - Componentes de um IDS.

Fonte: Campello e Weber (2001).

3.6.2 Métodos de detecção de intrusão

Segundo Campello e Weber (2001) os métodos de detecção de intrusão desempenham um dos principais papéis em um IDS, sendo responsáveis diretos pela busca de indícios de ações intrusivas, sendo evidente que diferentes ferramentas de detecção de intrusão utilizarão diferentes métodos para analisar os dados.

Assim dentre os métodos de detecção dos IDS, os mais comuns podem ser métodos tradicionais, baseado em assinaturas e baseado em anomalia.

3.6.3 IDSs mais conhecidos

Existem várias ferramentas para detecção de intrusão, onde nesta seção poderá apresentar algumas IDSs já existentes.

Um dos IDSs mais utilizados no momento, Snort combina simplicidade com eficiência. De distribuição livre, essa ferramenta baseia-se em uma arquitetura

centralizada, dados coletados na rede e uma análise baseada em assinaturas, podendo ser executada em qualquer sistema UNIX e, inclusive, em Windows. (CAMPELLO; WEBBER, 2001).

Segundo Campello e Weber (2001) uma das principais vantagens do Snort e a existência de uma base com milhares de assinaturas de ataques, disponível para download na página principal (www.Snort.org). Em sua grande maioria, essa base é fruto de colaborações da própria comunidade de usuários Snort espalhados pelo mundo, significando atualizações constantes e respostas praticamente imediatas ao surgimento de novos ataques.

O Snort possui ainda a facilidade de, além das regras disponíveis na Internet, permitir a criação de novas regras, no qual comparando o tráfego do segmento de rede com as regras existentes que o Snort detecta o código malicioso e gera o alerta ou toma medidas de contra-ataque. (STEFFEN JUNIOR, 2003).

Já o Bro é semelhante à ferramenta anterior, centralizada, de rede e baseada em assinaturas, este IDS possui como diferencial o formato de sua base de ataques. Nesse sentido, toda análise é feita utilizando scripts, descritos em uma linguagem própria, que representam políticas para cada serviço. Bro já conta com implementações em DecUnix, FreeBSD, Solaris, SunOS e Linux. (CAMPELLO; WEBBER, 2001).

Aafid foi desenvolvido pelo CERIAS (Center for Education and Research in Information Assurance and Security) da Universidade de Purdue, um dos grupos de maior prestígio na área, essa ferramenta utiliza o conceito de agentes distribuídos para a coleta dos dados e a análise de possíveis intrusões, tanto em redes como em hosts. Baseado em uma estrutura hierárquica, ele coleta as informações produzidas pelos agentes de cada host, pelo host em si e por conjuntos de hosts, reportando qualquer incidente através de uma interface gráfica. (CAMPELLO; WEBBER, 2001).

Ainda em evolução, Emerald é apontado como um dos mais modernos IDSs desenvolvidos. Sua arquitetura hierárquica e modular combina a possibilidade de diferentes tipos de análise, possuindo módulos analisadores baseados em comportamento e em assinatura, com a distribuição e a possibilidade de correlacionar alertas oriundos de vários pontos do sistema. Como ponto negativo, resta ainda à preocupação com a tolerância a falhas desses módulos. (CAMPELLO; WEBBER, 2001).

4. Metodologia

Está é uma pesquisa exploratória, a qual segundo Gil (2002) visa proporcionar maior familiaridade com o problema pesquisado, com vistas e torná-lo mais explícito, aprimorar idéias e descobrir intuições. Busca ainda estender o estudo do tema à integração com o problema, de maneira a construir hipóteses ou apenas explicitar os resultados. Envolve levantamento bibliográfico, parecer das pessoas que possuem experiências práticas com o problema pesquisado e análise de exemplos que incitem a compreensão.

A instalação e aplicação do firewall e do Sistema de Detecção de Intrusão foi realizada em diferentes microcomputadores como pode ser visto na Figura 2, sendo a instalação do firewall feita em um microcomputador com sistema operacional Linux Ubuntu versão 10.04 Server. O IDS foi instalado a partir do download de um ISO

através da página http://snorby.org, que contém o Snorby que é IDS de código fonte aberto Linux sendo baseada no Snort e Snorby, vem com o sistema operacional Turnkey Linux e gerenciados a partir da interface web Snorby. Já em um microcomputador com Windows 7 Ultimate 32 Bits foi feito como base para os devidos testes, sendo nesse microcomputador emulado outro sistema operacional, o Ubuntu 11.04 Desktop, que também serviu de base.

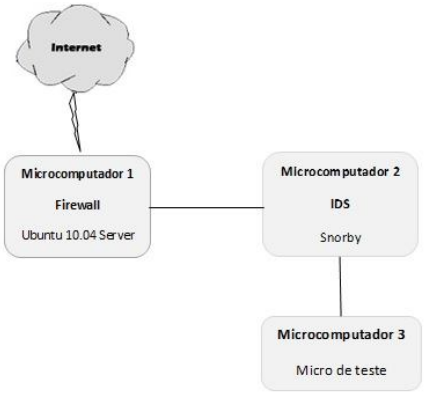

Figura 2 - Topologia do ambiente de testes.

Esta topologia de uma rede simplificada justifica-se pelo fato de que o principal objetivo do teste foi verificar a capacidade da ferramenta de IDS em detectar os ataques gerados contra a máquina alvo, assim verificar as vulnerabilidades de um firewall, e não de verificar a capacidade de segurança de uma rede como um todo.

As principais ferramentas que foram utilizadas e instaladas, são softwares livres, onde foi utilizado como IDS o Snort que é um tipo de sistema de detecção de intrusão capaz de executar análise do tráfego da rede em tempo real, e o Iptables, como firewall, que tem como principal objetivo filtrar os tipos de acesso ao computador assim proibindo ou permitindo o devido acesso.

Após a instalação do firewall Iptables, foi instalado e configurado o software Snort juntamente com o Snorby que vem a ser um moderno front-end para o Snort, onde com ele foi possível visualizar os alertas gerados pelo Snort e é possível ter uma interface de análise, como sistema de detecção de intrusão, no qual a escolha do Snort para o papel de IDS deve-se principalmente pela sua disponibilidade na forma de software livre, evitando-se a aquisição de um produto proprietário apenas para a realização dos testes.

Depois desta etapa de instalação e configuração tanto do firewall Iptables quanto do Snort como sistema de detecção de intrusão, foi feita a etapa de teste, fase em que a estrutura implementada foi submetida a testes com o objetivo de verificar o sucesso da solução de segurança baseada em firewall e IDS. Sendo esses testes feitos com a ferramenta de scaneamento da rede que foi o Nmap, e testes de vulnerabilidades com o uso do Nessus, onde foi submetido a ataques do computador base destinados a máquina que se encontra instalado o Snort, a fim de garantir que os sensores IDS identificasse os ataques.

Por fim foi feito a apresentação de resultados obtidos com a implantação do IDS em conjunto com firewall, demonstrando que firewalls apresentam vulnerabilidades,

podendo com o trabalho em conjunto com o IDS possa ter uma melhora no quesito segurança de redes.

5. Resultados Obtidos

A análise que será apresentada é fruto dos resultados obtidos no decorrer da aplicação, onde foi testado o firewall e o Sistema de Detecção de Intrusão, através de testes de scanner e testes de busca de vulnerabilidade na rede, onde as detecções foram identificadas pelo Snort e demonstradas através do Snorby que consegue demonstra graficamente.

Inicialmente pode-se observar na Figura 3 que o firewall pode ser iniciado através do comando "/etc/init.d/firewall start" ou pode ser parado através do comando "/etc/init.d/firewall stop" no terminal, e como pode-se ver a regra de firewall foi ativado conforme esperado.

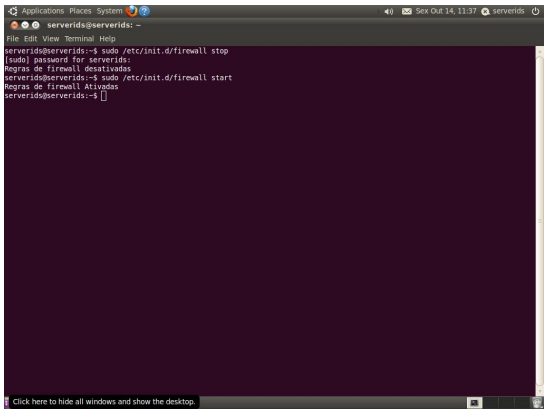

Figura 3 - Regra do firewall sendo iniciada via terminal.

Na Figura 4, pode-se observar a tela inicial, onde consegue ter a visualização da quantidade de alertas gerados pelo Snort e as estatísticas de intrusão no sistema.

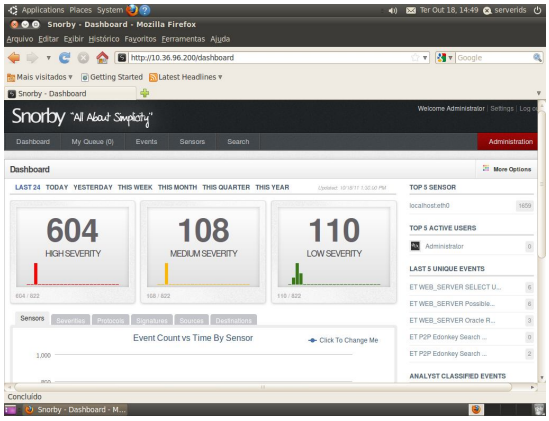

Figura 4 - Tela inicial do Snorby.

Com o Snorby é possível ter um grande controle da rede, identificando possíveis ataques e ter a visualização dos eventos, demonstrando a quantidade de eventos no mês, podendo também ser visualizados somente os eventos detectados nas últimas 24 horas,

no dia, no dia anterior, na semana, no trimestre e no ano.

Através do Snorby é possível ter acesso aos eventos detectados informando o IP de origem, o IP de destino, o tipo dos eventos e a hora do mesmo, sendo considerado como de elevado, médio ou de baixa severidade, bem como também as estatísticas de intrusões detectadas.

Para realizar os testes do firewall e do IDS foi utilizado uma ferramenta para exploração da rede que utiliza a técnica de port scanning conhecida como nmap, foi instalada via terminal no Sistema Operacional Ubuntu 11.04, e executado o comando direcionado tanto para a máquina alvo que se encontra o IDS quanto a que se encontra o firewall. E também foi utilizado para testes o Nessus que fez a verificação de falhas e vulnerabilidades de segurança na rede, sendo que foi instalado o servidor do Nessus em um micro com Sistema Operacional Windows 7 Ultimate e o cliente podendo ser visto em qualquer outro micro da rede sendo que a verificação dita é feita pelo servidor.

Com a execução do comando do nmap e do Nessus, o Snort imediatamente detectou a ação e foi possível visualizar o evento como é mostrado na Figura 5.

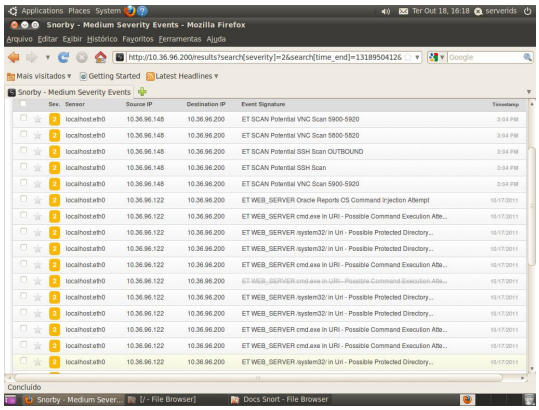

Figura 5 - Detecção dos testes por parte do Snort.

6. Conclusão

Este trabalho teve por objetivo demonstrar, através do estudo teórico e prático sobre firewall e também do estudo sobre Sistemas de Detecção de Intrusão (SDI), seus conceitos, métodos e funcionalidades e a importância do uso dessas ferramentas como meio de prover segurança na rede como um todo.

Longe de ser uma aplicação perfeita, o firewall quando não bem configurado e atualizado apresenta vulnerabilidades e o IDS surge como uma ótima ferramenta de administração e um importante auxiliar na melhoria da segurança, permitindo uma constante monitoração das atividades de tentativas de ataques na rede.

Dessa forma, não basta simplesmente configurar um firewall para criar barreiras de proteção contra um mundo externo, pois pode-se verificar nos testes que mesmo utilizando um firewall na rede, consegue-se ter facilmente informações da rede com simples ataques, com isso demonstra que firewall quando não bem configurados e atualizados, acabam possuindo vulnerabilidades. Com isso Sistemas de Detecção de Intrusão devem estar presentes nas redes de computadores e em constante monitoria, buscando informações que possam identificar ataques a rede.

Mas o IDS não deve substituir o firewall na tarefa de proteger as redes de dados, mas sim deve agir em conjunto com ele e tanto como outras ferramentas de segurança de maneira que consiga a garantir a segurança da rede, pois cada ferramenta acaba desempenhando uma tarefa específica.

Os resultados obtidos nos testes tanto de verificar as vulnerabilidades do firewall e de identificação dos ataques gerados como testes ficaram dentro das expectativas, sendo satisfatórias, pois os testes gerados acabaram sendo detectados pelo IDS. A partir destes resultados é possível partir para testes mais complexos.

Por fim, é importante ressaltar que mesmo tendo tanto um dispositivo de firewall e um IDS configurados de maneira em conjunto quanto de maneira única, não garante que um ataque seja totalmente impedido ou identificado, pois sempre novas formas de ataque ou novos pontos fracos são descobertos, no entanto, com a constante monitoração de tudo o que passa pelo firewall e pelo IDS, permite ao administrador da rede criar maneiras de dificultar ainda mais a ação dos ataques, seja implementando novos recursos ou criando novas regras.

Referências

BARBOSA, Ákio Nogueira. Um Sistema para análise ativa de comportamento de Firewall. São Paulo, 2006.

BARBOSA, André. Sistemas de Detecção de Intrusão. Rio de Janeiro, 2000.

BERNARDES, Mauro Cesar. Avaliação do Uso de Agentes Móveis em Segurança Computacional. São Carlos, 1999.

CAMY, Alexandre Rosa; SILVA, Evandro R. N.; RIGHI, Rafael. Seminário de Firewalls. Florianópolis, 2003.

CAMPELLO, Rafael Saldanha; WEBER, Raul Fernando. Sistemas de Detecção de Intrusão. Florianópolis, 2001.

GIL, Antônio Carlos. Como elaborar projetos de pesquisa. São Paulo: Atlas, 2002.

INOKOSHI, Rodrigo Kiyoshi. Avaliação de Firewall e Sistema de Detecção de Intrusão baseado em software livre. Jaguariúna, 2007.

NIC BR SECURITY OFFICE. Práticas de segurança para administradores de redes internet.:versão 1.2. Cert.br, 2003. Disponível em: <http://www.cert.br/docs/seg-adm-redes/seg-adm-redes.pdf >. Acesso em: 16 maio 2011.

ROCHA JUNIOR, Vamberto de Freitas Rocha. Estudo e implementação de Firewall em ambientes corporativos. João Pessoa, 2010.

SIEWERT, Vanderson C.. Firewall suas características e vulnerabilidades. Florianópolis, 2008.

SILVA, Glaydson Mazioli da. Guia Foca GNU Linux: versão avançada. 2005. Disponível em: <http://www.guiafoca.org>. Acesso em: 20 maio 2011.

STEFFEN JUNIOR, Julio. Sistema de Detecção de Intrusão. Novo Hamburgo, 2003.

ZWICKY, Elizabeth D.; COOPER Simon; CHAPMAN, D. Brent. Construindo Firewalls para Internet. 2. ed. Rio de Janeiro: Campus, 2000.