Segurança de Redes aplicada à Gestão de Logs e Eventos (GLE)

Trabalho de Conclusão de Curso

Engenharia da Computação

Rafael Bezerra Correia da Silva
Orientador: Prof. Edison de Queiroz Albuquerque

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

Rafael Bezerra Correia da Silva

Segurança de Redes aplicada à gestão de Logs e Eventos.

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, junho de 2011.

De acordo

Recife

_____/_____/_____

_____
Orientador da Monografia

A meus Pais,

Pelo incentivo, carinho e amor.

Agradecimentos

Venho a agradecer primeiramente a Deus que me presenteou com essa maravilhosa família e aos meus pais que sempre me apoiaram nessa minha batalha que foi a conquista do meu diploma como Bacharel em Eng. Da Computação pela Escola Politécnica de Pernambuco, aos meus primos, tios e tias que sempre me deram força nesses anos de estudos, e ao meu amigo Raphael Figueiredo que sempre me apoiou e me ajuda até hoje nos estudos e na vida.

Agradeço a todos meus amigos, especial aos mais próximos: João Fausto, Nathalia Maria Temudo, Cristiano Moura, Rodrigo de Ataíde, Andréia Oliveira, Marlon Cavalcanti, Saulo Medeiros, Lorena Tablada, Rômulo Jales, Samuel Martins e Raphael Figueiredo que foram responsáveis direta ou indiretamente pela conquista do meu sucesso.

Ao meu orientador, que me forneceu os materiais necessários para a conclusão desse documento, só tenho o que agradecer.

Agradeço também a minha namorada Lívia Maria Oliveira Sena, que foi uma das pessoas mais importantes na conclusão deste trabalho, me apoiando e ajudando em todos os momentos de dificuldades, a ela só tenho o que agradecer.

Resumo

Com a utilização da Internet as transações eletrônicas vem aumentando de forma considerável nesses últimos anos, bem como os valores agregados às informações. Este trabalho faz a proposta de uma implementação de um servidor de IDS(*intrusion detection system*) para assegurar o sigilo e integridade dos dados. Para tal problema serão utilizados as seguintes ferramentas de código aberto: *Snort* e o *Base*, que farão uma analise dos pacotes trafegados, alertando ao administrador de rede caso haja tentativas de invasão ao sistema via sms(*short message service*), que será implementado no trabalho.

Abstract

The use of the Internet has increased considerably in recent years as long as the aggregate value of the information. This work makes a proposal to implement a IDS server (instrusion detection system) to ensure proper data security. For this problem it will be used the following open source tools: Snort and Base, which will make an analysis of packet traffic, alerting the network administrator of any attempted intrusions into the system by SMS (short message service), which will be implemented in this work.

Sumário

Conteúdo

Índice de Figuras

Tabela de Símbolos e Siglas

S.O - Sistema Opercaional

TCP - Transmission Control Protocol

UDP - Utilization Datagram Protocol

RAM - Random Access Memory

DNS – Domain Name System

WAN – Wide Area Network

LAN – Local Area Network

Dos – Denial of Service

IP – Internet Protocol

IDS- Intrusion Detection System

B.O – Back Orifice

MAC – Media Access Control

NTFS – New Technology File System

CERT – Computer Emergency Response Team

Capítulo 1
Introdução

Antes de começar este trabalho, será apresentado uma simplificação do significado de segurança de rede. Todo tipo de proteção aos dados que estejam sendo transmitidos se encaixa na descrição de segurança de rede como por exemplo, aplicação de criptografia, uso de *firewall*, filtros e outras técnicas que serão mencionadas durante o andamento deste trabalho. Em meados de 1995 a 2000, a internet começou a se popularizar, junto com as ameaças de ataques virtuais, pois antes as máquinas das pessoas não eram interconectadas. Hoje em dia, a maioria das máquinas já estão, de alguma forma, conectadas à rede mundial, ficando sujeita a ataques e invasões. Nos dias atuais, mesmo não estando conectadas na rede, as máquinas não estão mais seguras por causa da invenção dos dispositivos móveis (*flash memory*) os populares *pen drives*. Antigamente o único risco de transmissão de vírus era por meio de disquetes e pela rede. Com o avanço da tecnologia a portabilidade e a facilidade do uso dos *pen drives* provocou um aumento muito significativo nas infecções em máquinas. Isto pode permitir que um *hacker* invada a máquina do usuário com a instalação de algum *trojan*[1]*,* em *background*[2]*,* quando se pluga o *pen drive* nessa máquina na hora do *autorun*[3], liberando o acesso a mesma.

1.1 Motivação

A motivação deste trabalho é devido à constante necessidade de termos segurança dos dados, visto que a cada dia a internet se torna o padrão de comunicação mais comum na sociedade, e é na mesma que trafegam dados importantes como dados pessoais, dados de conta de banco, senhas para

[1] É um programa que é usado em conexão reversa ou inversa, normalmente usado na obtenção de senha ou outras informações.

[2] Segundo plano

[3] Execução automática.

transações *Home Banking*, negociações financeiras e até mesmo informações vitais de uma empresa. Com o aumento de ataques de invasão à empresas e até mesmo em residências, devido a várias brechas de seguranças que os sistemas estão expostos, a utilização de uma ferramenta IDS que detecte a invasão gerando *logs* para evitar futuras invasões na porta que está aberta pode ser uma boa solução para o problema proposto.

Capítulo 2
Conceito de Segurança de Redes

2.1 Conceitos Básicos

O envio e o recebimento de informações sigilosas sempre foi uma necessidade antiga, há varios anos. Com o surgimento da internet no mundo e suas facilidades de transmissão dados de maneira precisa e extremamente ágil, por meio de criptografia torna-se fundamental para permitir que apenas o emissor e o receptor tenham acesso às informações trafegadas.

O que é criptografia [1]? a palavra criptografia surgiu da fusão de 2 palavras gregas "*kryptos*" e "*graphein*", que significam "oculto" e "escrever", respectivamente. Trata-se de um conjunto de conceitos e técnicas que visavam codificar uma informação (dado) de forma que somente o emissor e o receptor possam ter acesso, dificultando o entendimento por outra pessoa que possa interceptá-la. Para que se tenha uma maior segurança da informação, uma série de técnicas são usadas e outras devem surgir no decorrer dos tempos.

2.2 Política de Segurança

Uma política de segurança é definida como um conjunto de regras, normas e procedimentos a serem seguidos pelos usuários para limitar o seu acesso a rede. A política de segurança é baseada num sistema especifico não sendo utilizada para um sistema geral.

As regras estabelecidas em uma política de segurança indicam as restrições que cada componente do sistema (usuário) possui para a utilização na rede. As normas indicam o que cada componente tem permissão de fazer e como deverá ser realizado.

A politica de segurança pode ser divida em três ramos [2]:

- **Política de Segurança Física -** Esse tipo de política se baseia no meio físico onde o sistema opera, criando medidas preventivas contra desastres como incêndios, terremotos, enchentes, curto-circuitos e etc. Esta politica de segurança protege o acesso à pessoas não autorizadas às estações dos provedores do sistema, proibindo a entrada das mesmas nas estações.
- **Política de Segurança Gerencial -** Esse tipo de política se baseia no lado organizacional, definindo processos para seleção de pessoal para fazer parte do grupo da empresa ou instituição, e até mesmo processos para criação e manutenção das próprias políticas de seguranças da empresa.
- **Política de Segurança Lógica –** Esse tipo de politica é a mais praticada correspondendo ao controle de acesso dos usuários ao sistema, e controlando os níveis de privilégios que cada usuário terá no sistema. São aplicados 2 preventivas: a autenticação (onde o usuário irá se identificar ao sistema para poder ter acesso aos recursos que ele tem permissão) e a autorização (onde o usuário tem que provar que tem permissão de acessar os recursos que ele quer acessar).

Capítulo 3
Hackers e Ataques

3.1 Hackers x Crackers x Phreakers

Antes do estudo sobre os principais responsaveis pelas invasões, será feito uma distinção entre eles para melhor entendimento. Existe uma certa hierarquia imposta aos que decidem iniciar sua jornada pelo conhecimento da tecnologia da informação. Eles costumam se enturmar em sociedades secretas chamadas de clãs, e alguns deles agem sozinhos atribuindo suas ações a todo um clã [3].

Mas nem todos desejam ser criminosos, alguns agem por motivações que vão das ações desonestas à ações nobres, passando pela insensatez. Mas tanto os "bons" quanto os "maus" *hackers* são rebeldes e vivem em um mundo que possui seus folclores e até mesmo suas crendices. A disposição deles em camadas é um dos folclores desse meio. Essa divisão varia de clã para clã, em alguns, essa classificação é aceita como regra, e em outros apenas informalmente. Eles se classificam em:[3]

- ***Hacker*** – Essa é uma palavra bastante conhecida mas que possui um entendimento errado. Nos anos de 40 e 50, a palavra *hacker* era usada para categorizar radioamadores e *hobbystas* de mecânica ou eletrônica. Já na década de 60, o nome se popularizou como sinônimo de programador e especialista em computadores, embora fosse comum utilizá-lo para definir qualquer especialista: haviam *hackers* de astronomia, de mecânica de automóveis ou de jardinagem, por exemplo. Devido ao já citado desserviço prestado à comunidade *hacker* pela mídia, atualmente o termo tende a se referir aos criminosos digitais. Eles são especialistas que já dominam diversas técnicas de invasão e conhecem com profundidade pelos menos um S.O e são excelentes programadores e administradores de sistemas, mas diferentemente do que população acredita, eles possuem um

rígido código de ética e nunca usam seus conhecimentos para o mal [3].

- ***Crackers*** – São chamados de "*hackers* do mal", normalmente são especializados em quebrar travas de *softwares* comerciais para poder pirateá-los, mas também usam seus conhecimentos para invadir *sites* e computadores com objetivos ilícitos, como vandalismo ou roubo. Muitas vezes os *crackers* são excelentes programadores e podem criar programas que infectem ou destruam completamente sistemas alheios sem deixar vestígios. Normalmente eles fazem uso de uma grande quantidade de ferramentas para explorar as vulnerabilidades nos sistemas que querem efetuar a invasão. Um *cracker* sabe o que faz e, mesmo sendo um *hacker* "para fins ilícitos", ele tem noções suficientes para "se virar" caso algum imprevisto aconteça [3].
- ***Phreaker*** – É o *cracker* dos sistemas telefônicos. Normalmente eles possuem conhecimentos avançados de eletrônica e telefônia (principalmente sobre sinalização telefônica) e podem fazer chamadas de qualquer local sem pagar por elas. Os métodos de fraude incluem transferir as faturas para outros números (válidos ou não), modificar telefones públicos para conseguir crédito ilimitado ou mesmo enganar a central telefônica [3].

3.2 Quais motivos levam os Hackers a fazer um ataque

Os *hackers* sempre tem um motivo para fazer um ataque, seja ele um *hacker* "bom" ou "mau". Esses motivos podem ser por desonestidade ou por impulso, pode ser por amor ou por ódio, por necessidade ou até mesmo por vingança, ou por outro motivo qualquer, pois sempre há um motivo.

A busca por conhecimento parece ser o principal motivo para os ataques, mas na maioria dos casos é apenas um objetivo intermediário para atingir algo maior.

Existem pessoas que "hackeiam" por motivos políticos, ideológicos ou ambientais. Na China existem vários grupos lutando por uma abertura democrática e

usam a internet para isso. O Greenpeace e grupos neonazistas são outros exemplos [3].

Outras pessoas o fazem por simples vandalismo, ou por objetivos indecentes, tais como pornografia infantil, venda de armas e pirataria com fins econômicos ou não.

Grande parte dos *hackers* que invadem sistemas de terceiros são na maioria adolescentes ou jovens no inicio da fase adulta, que encaram os segredos do submundo digital como uma nova aventura ou até mesmo desafio na busca de conhecimento.

Existem pessoas mais velhas nesse meio mas a grande maioria são de pessoas muito jovens, ingênuas ou com pensamentos mal intencionados.

Podemos até dividir seus motivos por categoria:[3]

- **Espionagem Industrial** - Pode ocorrer de um empresa contratar um *hacker* para que este invada o sistema da concorrência, descubra seus planos e roube seus programas sem deixar rastros.
- **Proveito próprio** – Um *hacker* pode invadir um sistema para roubar dinheiro, transferir bens, cancelar dívidas ou até mesmo passar em concursos e ganhar em sorteios manipulando os números do sorteio. Qualquer ação em que ele seja diretamente beneficiado.
- **Inexperiência** – Um funcionário de uma empresa pode tentar acessar sua estação de trabalho de casa para concluir o trabalho que ele não conseguiu concluir durante o dia. Isto pode ser considerado uma invasão, mesmo que o funcionário não tenha o mínimo conhecimento do problema que seu ato pode causar.
- **Vingança** - Um ex-funcionário, tendo conhecimento do sistema, pode causar vários problemas como acessar os servidores da empresa e roubar dados vitais para ser contratado por um concorrente, ou até reconfigurar a rede para que ela não funcione direito como forma de vingança, raiva e descontentamento por ter sido afastado. Se o gerente de segurança da empresa não bloquear seu acesso e suas senhas

imediatamente após sua saída da empresa, poderá passar por esses possíveis riscos citados acima.

- **Status ou aceitação** - Uma invasão difícil pode fazer com que o invasor ganhe um certo status(mérito) junto aos seus colegas. Isso pode acarretar numa competição, numa verdadeira "gincana" dentro do seu clã. A mente humana tem uma constante necessidade de mostrar sua superioridade dentro de grupos.
- **Curiosidade ou aprendizado** - Muitos *hackers* dizem invadir sistemas apenas para aprender como eles funcionam por curiosidade. Este tipo de ataque raramente causa um dano maior ou compromete os serviços atacados, pois os mesmos invadem olham e saem sem causar nenhum risco aos sistemas ou aos arquivos.
- **Busca de novas aventuras** - O ataque a sistemas importantes, onde os esquemas de segurança são muito avançados e difíceis de invasão, podem fazer com que o *hacker* se sinta motivado pelo desafio e pelo perigo de ser pego.
- **Maldade** - Algumas pessoas sentem prazer na destruição. Invadem e destroem arquivos e sistemas de outras pessoas, pelo puro prazer de causar o mal. Raramente são pegos e se vangloriam dos seus atos deixando marcas.

3.3 Ataques e Vulnerabilidades

3.3.1 Ataques

Os ataques, normalmente, são planejados e segue uma estratégia para conseguir o sucesso sobre seu alvo desejado. Uma pessoa experiente em planejamento de ataque sempre traça um roteiro a ser seguido a fim de alcançar o seu objetivo, que seria a invasão do sistema ou maquina alvo. Um exemplo de um planejamento para uma invasão será simulado para passar um melhor entendimento de como os *hacker* pensam, conforme os passos abaixo [4]:

1. Primeiro, escolher um alvo, e localizar o alvo para o futuro ataque;

2. Reconher o máximo de informações possíveis sobre o alvo, utilizando ferramentas de gerenciamento de redes, ou de administração, para capturar dados importantes para o ataque em questão;
3. Fazer uma tentativa de invasão sobre o alvo para realizar testes de exploração das vulnerabilidades do S.O, servidores e serviços oferecidos pela rede;
4. Ter uma certa preocupação em não deixar rastros da invasão para que o alvo não descubra a violação nos arquivos de *logs* gerados pelo S.O;
5. Conseguir a senha de administrador do S.O para poder ter acesso aos recursos mais avançados do sistema, permitindo fazer alterações ou gerar um *bug* no sistema, prejudicando o usuário alvo. O invasor poderia instalar um *software* de obtenção de senha que grava e envia para o *e-mail* do invasor um *log* com todas as senhas digitadas pelo usuário, e o alvo não estaria sabendo. Estas senhas poderiam ser: senhas de correios eletrônicos, banco, acesso remoto do servidor da empresa, orkut, me*ssenger* entre outras;
6. Se preocupar em gerar novas rotas para fazer outra invasão, se prevenindo de que o alvo ou o administrador da rede encontre a "porta aberta" da máquina alvo e a feche para não permitir mais o acesso ao mesmo;
7. E por último, utilizar a máquina alvo para poder invadir outras máquinas dentro da própria rede do alvo.

Mas para um bom *hacker* o necessário, antes de tudo, é saber as vulnerabilidade do sistema para poder fazer o ataque. Os *hackers* possuem comunidades de discussões secretas que compartilham informações sobre vulnerabilidades de vários sistemas, e são dessas informações que eles usam como fonte para reconhecer informações de vulnerabilidades de sistemas comuns [3].

Segue abaixo na Figura 1, um gráfico com as estimativas de ataques de vários tipos, reportados a partir do ano de 1999 à 2010 [5].

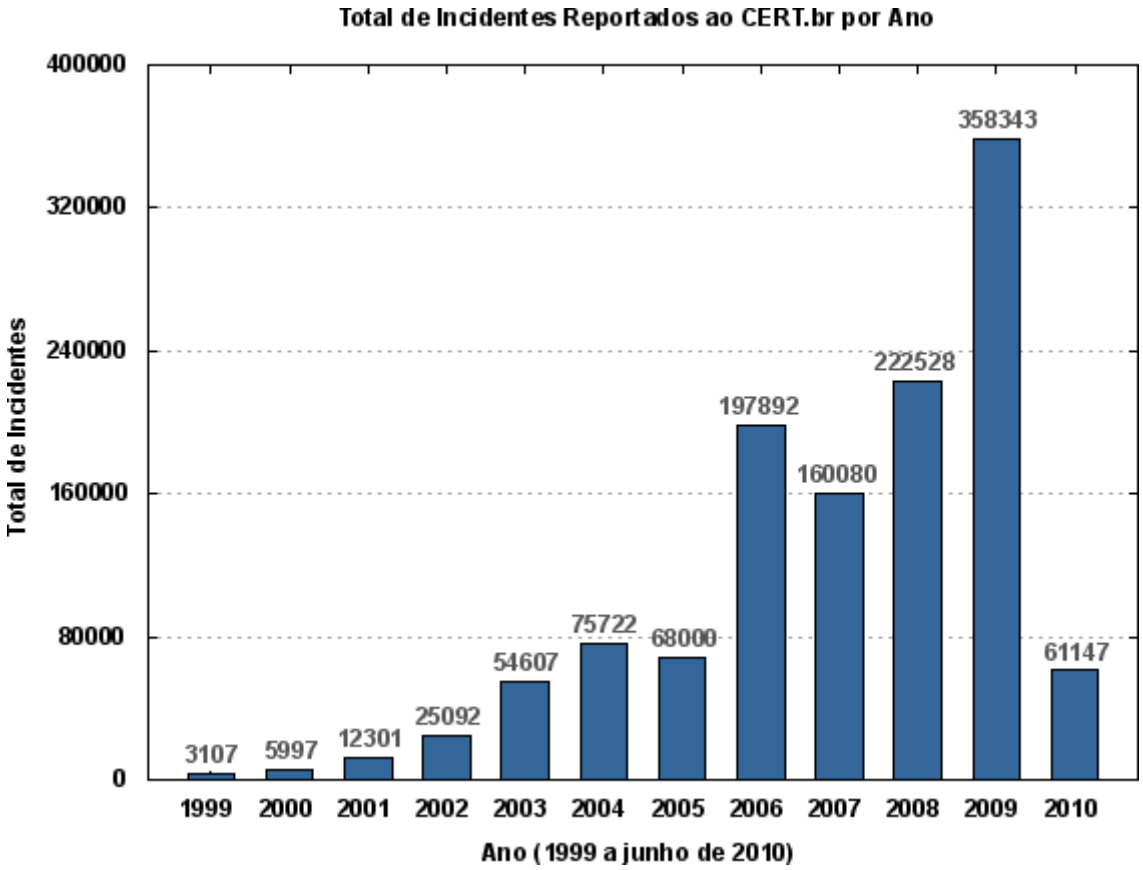


Figura 1. Dados de incidentes entre 1999 à 2010 (CERT.br, 2010)

Dados coletados pelo site CERT sobre os tipos de ataques mais freqüentes entre abril a junho de 2010, são indicados na figura 2 [5].

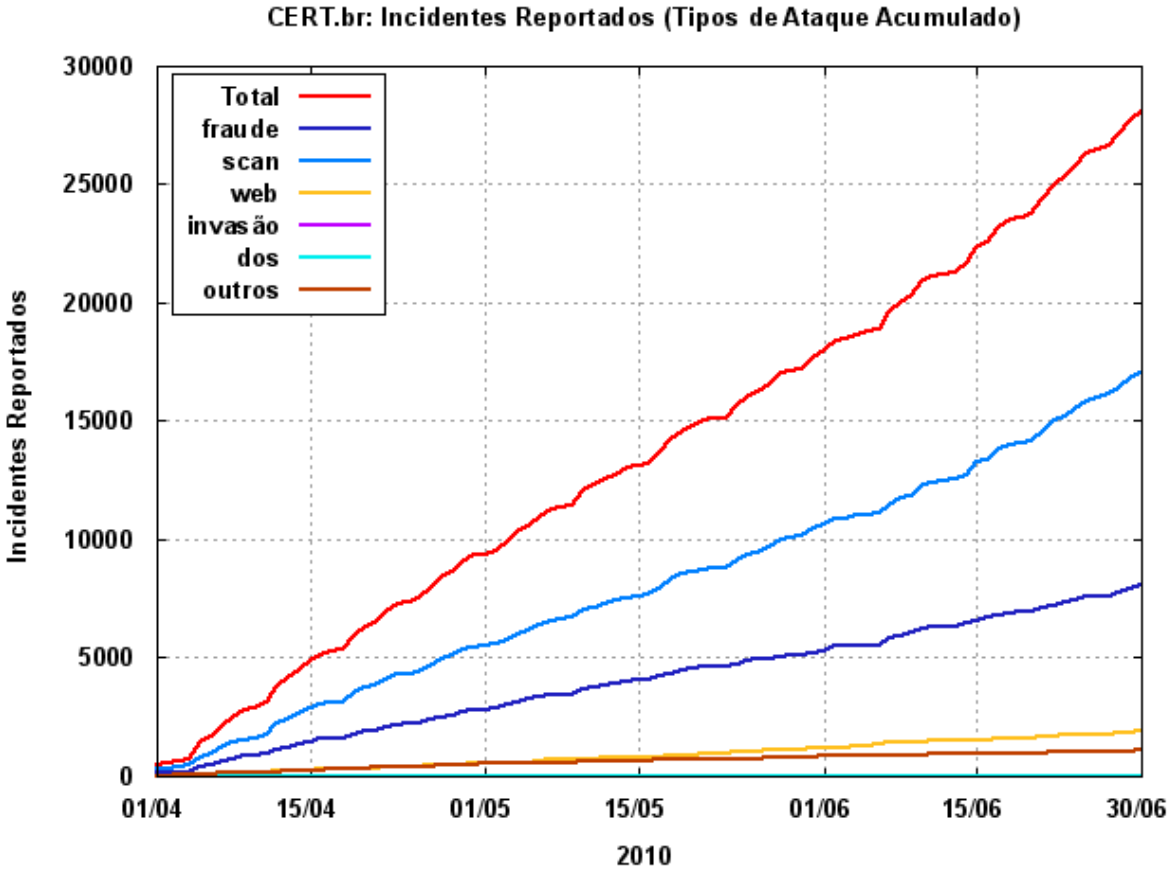


Figura 2. Tipos de ataques entre abril a junho de 2010 (CERT.br, 2010)

Conforme a Figura 3, as portas que tiveram um maior número de ataques foram as portas 22, 23 e 25 usando o protocolo TCP, que seriam os serviços SSH[4], Telnet[5] e SMTP[6] respectivamente[5].

[4] Protocolo de acesso remoto com criptografia(seguro).

[5] Protocolo de acesso remoto sem criptografia (não seguro).

[6] Protoloco de envio de correios eletrônico.

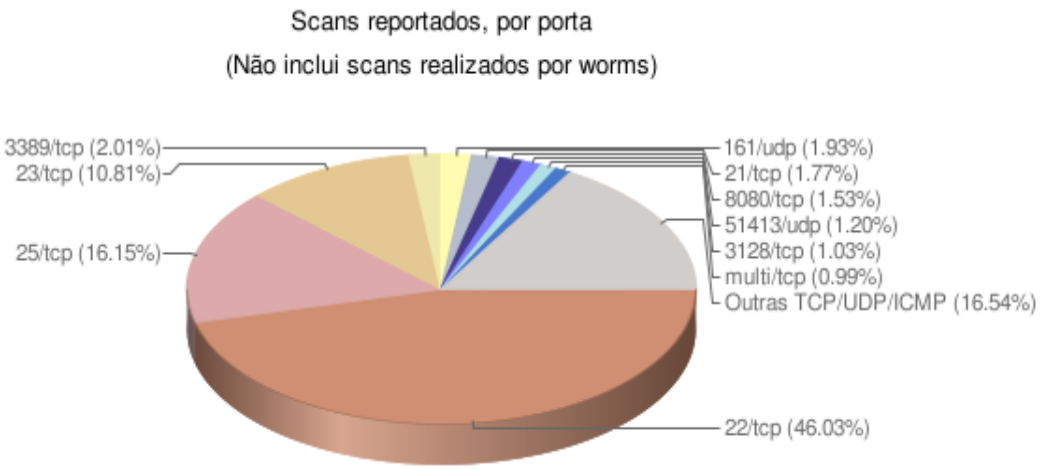


Figura 3. Portas que são frequentemente atacadas (CERT.br, abril-junho 2010)

3.3.2 Vulnerabilidades

A vulnerabilidade em sistemas significa a existência de brechas ou caminhos alternativos, conhecida como *bug*, que provocam a facilitação dos ataques a sistemas. Alguns tipos de vulnerabilidades são muito explorados atualmente pelos *hackers*, como as do tipo *buffer overflow* (transbordamento de dados), que muitas vezes pode dar privilégios de administrador para o invasor, rodar códigos maliciosos remotamente, burlar particularidades de cada sistema, fazer ataques DoS (*Denial of Services*) e acesso irrestrito ao sistema [4].

Existem ferramentas que exploram as vulnerabilidades do sistema, e que são chamadas de *exploits*.

Um *exploit*, em segurança da informação, é um programa, uma sequência de comandos que utilizam as vulnerabilidades de um sistema. São geralmente elaborados por *hackers* como programas de demonstração das vulnerabilidades, para criação de *hotfix* (*software* de reparo de problemas e *bugs* nos sistemas) que corrige as falhas, ou por *crackers* que as usam para ter acesso não autorizado aos sistemas.[6]

Capítulo 4
Tipos de Ataques e Prevenção de Ataques

4.1 Tipos de ataque

4.1.1 Trojan Horse (Cavalo de Tróia)

O cavalo de tróia (*trojan horse*), é um programa capaz de infectar outros computadores, permitindo total acesso do invasor à máquina infectada, onde ele poderá executar qualquer ação na máquina da vitima. Exemplos de programas que serviam como cavalo de tróia são o Netbus e o B.O (*Back* O*rifice*), os quais eram postados em sites informando que a função deles era de invasão, e que utilizando tanto o Netbus ou o B.O poderia invadir a máquina de outra pessoa, enganando as pessoas que baixavam e instalavam os programas. Os trojan Horse abrem as portas que serão utilizadas para o futuro ataque [4].

4.1.2 Flood

O *flood* é uma técnica usada para ocasionar uma lentidão no serviço que está sendo atacado, podendo até derrubar o serviço, dando uma sobrecarga no servidor que o está rodando. O *flood* é o envio constante de disparos de pacotes para o servidor com objetivos de causar um congestionamento no *buffer*[7] do servidor [4].

4.1.3 Nuke

Nuke é o nome dado aos pacotes TCP (Protocolo de controle da camada de transporte), UDP (Protocolo de transporte não orientado a conexão), ICMP (Protocolo de controle de mensagens de erro) e IP(Protocolo de Internet) com alterações, que pode provocar lentidão no servidor, ou até mesmo travar o sistema

[7] É uma região de memória temporária utilizada para escrita e leitura de dados

devido à dificuldade de identificar e tratar , corretamente os pacotes recebidos. Um exemplo mais claro para esta explicação seria, enviar um pacote alterado com o IP de origem e o mesmo valor do IP do servidor que irá receber. Assim o servidor tentará estabelecer uma conexão a si próprio, entrando em uma repetição infinita, causando um travamento e uma possível queda no serviço, caso não seja detectado esse ataque. Os programas nukes utilizam os seguintes princípios para provocar o ataque:

- Define quem serão os clientes e servidores;
- Define o tipo de mensagem a ser enviada;
- Define o host cliente;
- Define o host servidor (alvo do ataque);
- Define as portas a serem “nukadas” no cliente;
- Define as portas a serem “nukadas” no servidor;
- Define o intervalo que cada pacote será enviado.

Após esses procedimentos citados anteriormente, se inicia o envio dos pacotes com mensagens de erro (pacotes alterados). Quanto maior a velocidade do link utilizado para a conexão, mais rápido o alvo será atacado[4].

4.1.4 IP Spoofing

É uma técnica que consiste em mascarar pacotes IP usando endereços de remetentes falsificados.

O reencaminhamento de pacotes é feito usando premissa muito simples: O pacote é enviado para o destinatário (endereço de destino) e nesse envio não haverá verificação do remetente ou melhor, não terá uma validação do endereço IP em relação ao pacote com o roteador anterior. Assim, torna-se fácil falsificar um endereço de origem através do cabeçalho IP.

Vários computadores podem enviar pacotes fazendo-se passar por um determinado endereço de origem, o que poderia provocar uma séria ameaça para os sistemas baseados em autenticação pelo endereço IP[4].

4.1.5 DNS Spoofing

A idéia principal desse tipo de ataque seria fazer que um servidor DNS (*Domain Name System ou* sistema de nomes de domínios) permita que máquinas não autorizadas (não-confiáveis) passem a ser vistos pelo servidor de DNS como *hosts* confiáveis à rede. Mas para este objetivo ser conquistado, o invasor deve conseguir ter o controle do *host* servidor de DNS, e identificar um nome de uma maquina, que tenha na lista de *host* confiáveis no servidor DNS, desta forma, edita-se o registro do DNS que mapeia o endereço IP do *host* confiável para o seu nome, alterando-o para que o servidor DNS tenha o endereço do *host* invasor. A partir disso, o invasor terá acesso aos serviços baseados em autenticação por nome. Grande parte dos sistemas atuais já possuem métodos que identificam este tipo de ataque utilizando a técnica conhecida como *cross-check [4][3].*

4.1.6 Source Routing Attack

Este tipo de ataque utiliza os mecanismos de roteamento existentes na rede e da opção *loose source route* [8] do protocolo IP para induzir que a máquina que será atacada acredite que o ataque tenha sido originado de uma operação legítima de uma outra máquina confiável. A opção de *loose source route* disponibiliza um mecanismo para que a origem de um datagrama possa fornecer informações de roteamento usada pelos gateways, para direcioná-lo ao destino. Desde modo um processo pode começar uma conexão (sessão) TCP fornecendo um caminho explícito para o destino, modificando o processo atual de roteamento.

O protocolo IP especifica quais pacotes contendo informações de roteamento devem retornar à sua origem utilizando-se do caminho reverso de rotas escolhida por ele. Um ataque pode desativar um computador que possui relação de confiança com o computador alvo (utilizando um ataque como *Flood*, *Nuke*, etc., mencionados anteriormente) e definir em seu computador o IP do computador desativado.

Para um melhor entendimento siga o exemplo: Sejam três hosts T,Y,Z. Imagine uma situação em que o intruso em T lança um *source route attack* em Y que confia em Z. Os seguintes passos do ataque serão descritos abaixo:

[8] É uma opção no protocolo IP que pode ser usado para fazer traduções de endereços.

1. O invasor, com algum tipo de ataque desestabiliza Z, ou espera até que Z seja desligada.

2. O invasor configura T para que contenha o endereço IP de Z. Neste momento o invasor está disfarsado de uma máquina confiável.

3. Nesse momento T acessa Y, para isso deve se utilizar de pacotes IP com opção *loose source route* ativada e corretamente configurada, contendo um caminho valido de T para Y.

4. Y aceita as requisições de T pensando que são de Z. A partir deste momento T obtém uma conexão confiável com Y [4].

4.1.7 Vírus

Vírus são programas de computadores, com a diferença de que foram desenvolvidos com um único propósito: prejudicar o usuário que será atacado por ele.

Para ser atacado por um vírus, o usuário alvo, obrigatoriamente, tem que executar uma vez um programa infectado ou acessar os antigos dispositivos de armazenamento, tais como disquetes infectados ou, atualmente, os *pen drives*. Por exemplo, em casos de *e-mails* contento arquivo infectado, caso o usuário não execute o programa em questão o mesmo não será infectado.

Os micros em si não “pegam” vírus nem “ficam” com vírus. Os vírus ficam na verdade alojados ocultamente dentro de programas, alocado na memória RAM em área de boot de um S.O ou disco rígido. Por causa desses motivos, é improvável que exista algum *hardware* de computador infectado, pois a maioria dos *hardware* que possui memória normalmente são voláteis e por ser volátil o vírus não será armazenado no dispositivo em questão.

O funcionamento dos vírus ocorre da seguinte maneira: após a execução de um arquivo infectado, o vírus se aloca na memória RAM do micro, passando a interceptar todas as rotinas de acesso a arquivos e disco no sistema operacional. Sempre que um novo arquivo for executado, o vírus irá adicionar uma cópia de si próprio neste arquivo. Existe outro tipo de vírus que é conhecido como “vírus de

boot[9]". Este vírus se aloca no setor de inicialização de um disco, sejam eles disquetes, *pen drives* ou discos rígidos. Normalmente os vírus criam um cópia de si mesmo em todos os disquetes que forem inseridos na unidade. Atualmente, o uso de disquete foi praticamente extinto, mas a mesma forma de contaminação é usada para infecção de *pen drives*, cartões de memória e memória *flash* [4].

4.1.8 Sniffers

Na maior parte das redes de computadores, pacotes são transmitidos para outros computadores conectados à rede, e cada máquina (*host*) é configurada para somente tratar os pacotes direcionados a ela. No entando, é possível reconfigurar a interface de rede para que ela capture todos os pacotes que circulam pela rede, não importando o destino desse pacote. Este modo que a interface de rede opera é chamada de modo promíscuo e esta técnica é denominada *sniffing.*

Os ataques baseados em monitoria tem um software especifico chamado "*sniffer*". O *sniffer* busca gravar os 128 bytes de cada sessão de login, telnet e sessão FTP executada no mesmo segmento da rede local, comprometendo todo conteúdo trafegado na rede. Os dados capturados incluem o nome do *host* (máquina) destino, a identificação do usuário (*username*) e a senha (*password*). A informação é armazenada num arquivo que futuramente poderá ser recuperado pelo invasor que acessa outras máquinas. Em alguns casos os invasores adquirem acesso inicial aos sistemas utilizando das seguintes técnicas abaixo:

- Obtém o arquivo de senhas via FTP em sistemas mal configurados;
- Obtém acessos ao sistema de arquivos locais usando o sistema NTFS sem restrições;
- Utilizam nome de login (*username*) e senha (*password*) capturados por um *sniffer* rodando em outro sistema.

Normalmente, nas redes de computadores trafegam muitas informações sigilosas como, por exemplo, nomes de usuários e senhas, tornando fácil para um

[9] É o termo em inglês para o processo de iniciação do computador que carrega o sistema operacional quando a máquina é ligada

programa *sniffer* obter estas informações. O volume de dados geralmente são enormes para serem tratados, mas o processo pode ser simplificado através de regras simples de filtragem, e pelo fato de ser fácil detectar o inicio de uma conexão TCP.

A utilização de um *sniffer* na rede é de difícil detecção. Se ele estiver somente coletando dados e não respondendo a nenhuma requisição, a única forma do *sniffer* ser detectado é percorrendo fisicamente todas as conexões da rede. Outra maneira para detectar um *sniffer* em operação é através dos imensos arquivos gerados por ele. O *sniffer* requer recursos que só pode ser acessado com usuários administrativos, ficando inviável a utilização do *sniffer* por usuários normais ou limitados [4].

4.1.9 DoS (Denial of Service)

De acordo com dados coletados do CERT (*Computer Emergency Response Team*), os ataques DoS, também denominados ataques de negação de serviços, consistem em tentativas de impedir usuários legítimos de utlizarem um determinado serviço de uma máquina. Para esse fim, são usadas algumas técnicas que podem: sobrecarregar uma rede a tal ponto em que os verdadeiros usuários dela não consigam usá-la; derrubar uma conexão entre dois ou mais computadores, fazer varias requisições a um site até que este não consiga mais ser acessado por causa da demanda das requisições do ataque, negar o acesso a um sistema ou a determinados usuários.

Fazendo uma analogia para melhor entendimento de como funciona este ataque, admite-se que as pessoas usam um metrô regularmente para irem para o trabalho ou faculdade. Mas, em um determinado dia, uma quantidade enorme de pessoas egoístas desrespeitam a fila e entram no metrô, deixando-o tão lotado que os outros passageiros regulares não conseguem entrar. Ou então, que as pessoas tenham que entrar no metrô, mas este ficou tão cheio que não conseguiu sair do lugar por excesso de peso. Este metrô acabou negando o seu serviço, o de transportá-lo até seus trabalhos ou faculdades, porque recebeu mais solicitações (passageiros) do que o mesmo suportava.

É importante frisar que quando uma maquina (computador) ou um site sofre um ataque DoS, ele não é invadido, mas sim sobrecarregado. Isso indepente do sistema operacional utilizado.

Os ataques do tipo DoS são mais comuns e podem ser feitos devido a algumas características do protocolo TCP/IP, sendo possível ocorrer em qualquer computador que o utlize. Uma das formas de ataque mais conhecidas é a *SYN Flooding*, onde um computador tenta estabelecer uma conexão com um servidor através de um sinal do TCP conhecido por SYN (*Synchronize*). Se o servidor atender o pedido de conexão, enviará ao computador solicitante um sinal chamado de ACK (*acknowledgement*). O problema é que em ataques desse tipo, o servidor não consegue responder a todas as solicitações e então passa a recusar novos pedidos.

Mais uma forma de ataque comum é o *UDP Packet Storm* (Tempestade de pacotes UDP), onde um computador faz solicitações constantes para que uma máquina remota envie pacotes de respostas ao solicitante. A máquina fica tão sobrecarregada que a mesma não consegue executar suas funções [7].

4.2 Tipos de Prevenção Contra Ataques

4.2.1 Firewall

Firewall, que em português significa parede corta fogo, é o nome dado ao dispositivo de uma rede que tem como propósito aplicar uma política de segurança a um determinado ponto de controle da rede para acessos não autorizados. Sua função na rede consiste em analisar o tráfego de dados entre redes distintas e impedir a transmissão ou recepção de pacotes devidos a acessos nocivos a rede ou não autorizados de uma rede para outra. Este conceito se aplica a equipamentos de filtros de pacotes e de *Proxy* (é um servidor que atende a requisições repassando os dados do cliente a outro servidor) de aplicações, normalmente associados a redes TCP/IP.

Uma das grandes preocupações na área de segurança de redes nos dias de hoje é a vulnerabilidade das máquinas (*hosts*), que podem compremeter as transmissões pelo meio físico da rede a que ela pertence. Para prevenção e proteção dessas vulnerabilidades, um método tem se mostrado bastante eficiente

para impedir que informações indesejadas entrem na rede. Este método não serve como um método absoluto de segurança a um *host* e sim uma forma completamentar de reforçar a segurança do mesmo, que consiste normalmente em analisar a ligação de uma rede interna com a internet (WAN), instalando-se um equipamento que permitirá, ou não a entrada e saída de informação, baseada em uma lista de permissões e restrições, devidamente configuradas para suprir as necessidades básicas de comunicação da rede interna com a internet, e esse procedimento é o que um *firewall* faz. Na Figura 4, segue uma ilustração mostrando um ambiente de como o *firewall* funciona [8].

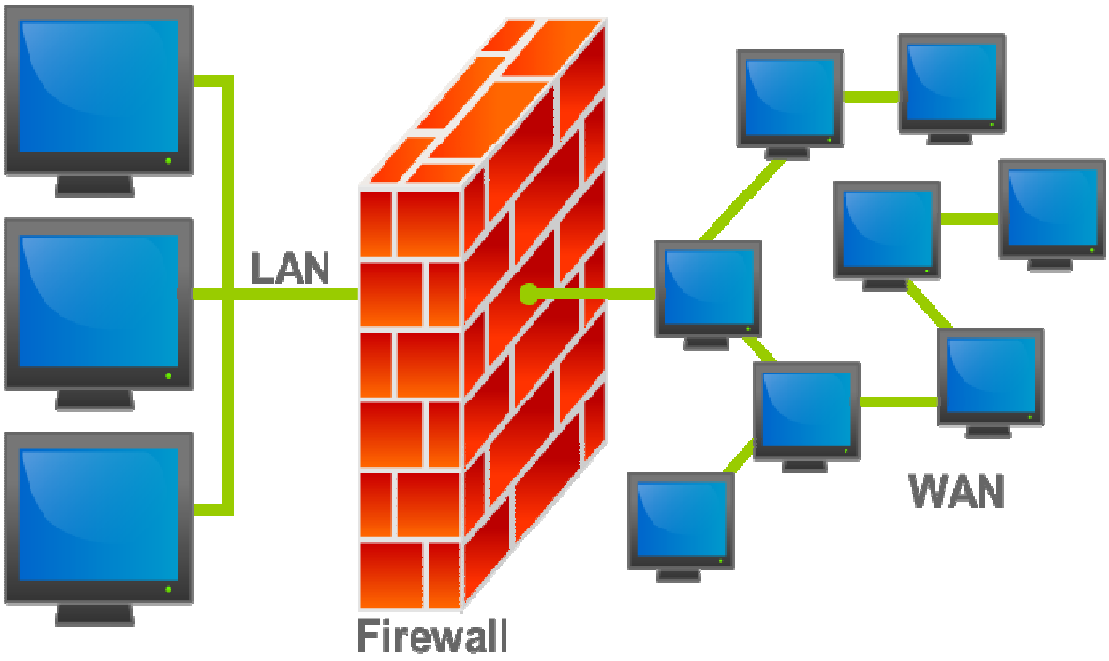


Figura 4. Topologia de um firewall (Wikipédia[10])

4.2.2 Servidores Proxy

Proxy é um servidor que atende as requisições repassando os dados do cliente para o servidor destino. Um usuário (cliente) se conecta ao servidor *proxy*, requisitando algum pedido, como um arquivo, conexão ou acesso a um website, ou outro recurso disponível em outros servidores.

[10] http://pt.wikipedia.org/wiki/Ficheiro:Firewall.png

Um servidor *proxy* opcionalmente, pode alterar as requisições feitas por um cliente ou as respostas dos servidores e, em alguns casos, pode até disponibilizar este recurso sem nem mesmo se conectar ao servidor destino. Ele pode atuar como um servidor *cache* que tem a função de armazenar dados em forma de arquivos *offline* em redes de computadores. São normalmente instalados em máquinas com um grau de processamento e armazenamento elevados em relação as das máquinas clientes. Esses servidores têm uma série de recursos, como filtrar conteúdo numa rede, providenciar anonimato e guardar páginas de servidores *Web*.

Um exemplo de um recurso de um *proxy* seria um *HTTP caching proxy*, que permite o cliente requisitar um documento na *world wide web.*O *proxy* iria procurar pelo documento solicitado em seu cachê e, caso o encontre, o documento solicitado é retornado imediatamente ao *host* solicitante. Caso contrário, o *proxy* iria buscar o documento no servidor destino, entregar o documento ao cliente e salvar uma copia no seu cache para acessos futuros. Isso permite uma diminuição do uso da banda na rede, aumento da velocidade de respostas por não precisar navegar pela internet para ter o retorno do documento necessário, e evita expor ao máximo as maquinas na internet onde poderá ter o risco de captura de pacotes ou de informações, para um possível ataque futuro, fazendo com que as máquinas da rede tenha o mínimo de acesso as redes externas, isso diminui o risco a ataques, pois as máquinas iram se expor menos as redes externas [4].

4.2.3 Políticas de Segurança

As decisões relacionadas à segurança são, em grande parte, determinante para a segurança da rede, mas não se pode tomar boas decisões sem determinar primeiro quais são as metas de segurança que serão adotadas. Até que se determine as metas de segurança, não se pode fazer uso efetivo de qualquer conjunto de ferramentas de segurança, pois não se tem os conhecimentos do que irá se analisar e conferir ou quais restrições irão ser impostas.

As metas serão definidas na maioria das vezes pelos seguintes pontos chaves:

- **Serviços oferecidos X Segurança provida** – Cada serviço ofertado aos demais usuários apresentam um risco de segurança. Para alguns

serviços em si, o risco excede em valor o benefício do serviço e o administrador de rede tem a escolha em optar por eliminar o serviço em lugar de tentar deixá-lo seguro.

- **Custo de segurança X Risco de perda –**Há muitos custos referentes à segurança de redes: monetário (i.e., o custo em investir em equipamentos (*hardwares*) e *softwares* como *firewalls* e geradores de senhas temporárias), desempenho (i.e.,nível de processamento do *hardware*) e facilidade de uso [4]. Tais custos devem ser avaliados frente ao risco relativo à segurança, por não tê-los.

4.2.4 Estouro de Pilha

Qualquer programador, tem por obrigação escrever um código imune a *buffer overflow* (transbordamento de dados). Mesmo sendo inerente às linguagens de programação é possível implementar rotinas de verificação que barram injecção maliciosa de dados.

Os administradores de sistemas ou mesmo usuário domésticos, devem manter seus sistemas sempre atualizados com os últimos *hotfix (patchs)* publicados pelos fabricantes de seu *software*, que consertam certas vulnerabilidades em seus sistemas. Mas a atenção dessas atualizações tem que ser voltada tanto para servidores como para *host* clientes, pois estações de trabalhos também são vulneráveis e podem servir de portas de entrada para sua rede [4].

4.2.5 Sniffing, Spoofing e Hijacking

Primeiramente, se deve colocar nas redes filtros *anti-spoof* e detectores de *sniffers* em todos os pontos de entrada, saída e passagem (roteadores entre sub-redes). IDS (*intrusion detection system*) também se aplica junto destas ferramentas.

Mesmo não sendo impedimento para a ação do *hacker*, as seguintes medidas são agentes complicadores: instalar switches em vez de hubs e fazer segmentação da rede ao máximo. Além do beneficio do desempenho, isso cria uma camada a mais de dificuldade para o invasor. Se possível, dividir a rede em subredes e colocar roteadores com filtros de pacotes (*Access lists*) muito bem estruturados para interligá-las. Dependendo do número de usuários, tempo, custo e tamanho da rede,

é possível configurar estaticamente as tabelas MAC nos switches e bridges em cada uma de suas portas. Com isso, o equipamento fica imune a *ARP spoofing* e *MAC Flooding* (segue o mesmo principio do *DNS spoofing* citado anteriormente)[4].

Capítulo 5
Sistema de detecção de intrusão (IDS)

Um IDS é uma ferramenta utilizada para detectar e alertar o administrador de rede sobre ataques e tentativas de acesso não autorizados na rede. A solução IDS é um conjunto de ferramentas que , aplicadas ao *firewall* proprociona o monitoramento do tráfego tanto de entrada como de saída das informações na rede.

Para facilitar o entendimento, será feita uma comparação de um IDS fazendo uma analogia com o sistema de defesa do corpo Humano [9].

- **Sistema de defesa do corpo humano –** os anticorpos constituem um mecanismo de defesa que o ser humano possui. Eles tem a função importante de proteção ao organismo do indivíduo contra substâncias estranhas no corpo. Quando um vírus ou bactéria, por exemplo, invade o corpo humano, certos glóbulos brancos do sangue denominados linfócitos T, produzem e lançam na corrente sanguinea um tipo especial de proteína capaz de se unir com as moléculas que compõem este vírus ou bactéria, e assim inativando-o. A proteína que o indivíduo produz, em resposta ao ataque do vírus ou bactéria é chamado de anticorpo, e as substâncias estranhas (vírus e bactérias, por exemplo) são chamadas de antígenos. Quando isso ocorre o corpo então se torna imune aquela ameaça, e se caso este vírus vier a invadir novamente o corpo já existirá anti-corpos específicios para eliminá-lo. Caso haja uma nova invasão de um vírus desconhecido pelo corpo, os linfócitos T se encarregarão de encontrar informações sobre esse novo vírus e criar uma vacina (anti-corpos) para eliminar este vírus [10].
- **Sistema de IDS –** O IDS, tem como propósito principal detectar se alguém está tentando invadir o sistema, ou se algum usuário legítimo está tentando fazer um mau uso da rede. Esta ferramenta roda

constantemente em *background* (é recomendável um servidor só para esse serviço) e somente gera uma notificação quando detecta alguma irregulariedade que seja suspeita ou ilegal. Assim sendo, podemos dizer que os linfócitos T corresponde aos sistemas de *firewalls* analogicamente, e que sabem qual o tráfego pertece a rede, de acordo com a configuração da política de rede. As imunidades são os padrões de ataques ou assinaturas, ou seja, são ataques já conhecidos previamente. O sistema também é caracterizado por possuir inteligência para aprender com o comportamento da rede e , com isso, identificar novos padrões ou variações dos padrões existentes.

5.1 Característica de um IDS

5.1.1 Algumas características de um IDS são:

- Gerenciamento centralizado;
- Possibilidade de interação com outros elementos de rede como *firewall*, roteadores e consoles de gerência;
- Possibilidade de construir uma base de conhecimento centralizada de forma a permitir uma visão ampla do nível de segurança da rede e o conhecimento das ameças existentes. Desta forma, quando algum atacante (antígeno) for detectado pelo sistema, torna-se possível as ações de alarme seguintes:

  -Envio de e-mail para o adminstrador,

  -Envio de mensagem via Pager ou SMS,

  -Ativação de alertas nas estações de gerência via SNMP,

  -Reconfiguração de elementos de rede como *firewall* e roteadores, e até mesmo o encerramento da conexão através do envio de pacotes de reset (*flag* RST do TCP) para a máquina atacante e para a máquina atacada, com o propósito de descarregar a pilha TCP.

Isto permite ao administrador da rede adotar contra-medidas (anticorpos) para combater o mal.

5.1.2 Características de um IDS com uma configuração ideal

- Deve rodar continuamente sem interação humana e deve ser seguro o suficiente de forma a permitir sua operação em segundo plano;

- Sua base de conhecimento não deve ser perdida quando o sistema for reinicializado, ou desligado inesperadamente;

- Deve monitorar a si próprio de forma a garantir sua segurança;

- Ter o mínimo de impacto no funcionamento do sistema;

- Poder de detectar mudanças no funcionamento normal;

- Cada sistema possui padrões diferentes e a ferramenta de IDS deve ser adaptada de forma fácil aos diversos padrões.

5.1.3 Vantagem de um IDS

Um IDS tem a possibilidade de monitorar:

- Quais serviços[11] estão sendo atacados;
- Qual a origem dos ataques;
- Portas e protocolos de acesso sendo utilizados para a tentativa de invasão;
- *Software* e *backdoors*[12] os quais o invasor escolhe para utilizar;
- Acesso interno de sua rede para servidores de IRC, ICQ, MSN e Yahoo Messenger;

[11] São serviços de rede em geral, como web, *e-mail*, *proxy*, FTP, *firewall*, dns, etc.

[12] São uma ou mais falhas de segurança para ter acesso ao sistema operacional que podem ser exploradas por um invasor.

5.2 Intrusão

Todas as intrusões devem estar definidas na política de segurança de rede. Enquanto não for escolhido o que realmente é permitido e o que não é permitido no sistema, é difícil entender ao certo o que seria uma intrusão.

Uma intrusão pode ser definida como qualquer conjunto de ações que pode comprometer a integridade do sistema, a confidencialidade e/ou a disponibilidade dos dados ou do sistema.

Uma intrusão pode ser detectada a partir de alguns aspectos do sistema, como a utilização da CPU, número de conexões por minuto, número de processos por usuários entre outros aspectos. Uma variação significativa nestes padrões pode ser um alerta de intrusão. i.e. a exploração das vulnerabilidades de um sistema envolve a utilização indevida ou anormal do sistema.[9]

5.2.1 Como detectar uma intrusão

A maioria das ferramentas de IDS realizam suas operações a partir de análise de padrões (regras) do sistema operacional e da rede, e alguns aspectos para se detectar uma intrusão são:

- Utilização da CPU elevada;
- I/O de disco;
- Uso de memória;
- Atividades dos usuários;
- Números de tentativas de *login*;
- Número de conexões;
- Volume de dados trafegando no segmento de rede;

Estes aspectos citados acima formam uma base de informação sobre a utilização do sistema em vários momentos do tempo, enquanto outras já possuem bases com padrões de ataque previamente montadas (regras criadas) permitindo também a configuração dos valores das bases bem como inclusão de novos parâmetros.

Com estes conjuntos de informações a ferramenta de IDS pode identificar as tentativas de intrusão e até mesmo armazenar a técnica usada por ela.

Na Figura 5, temos um (sensor) IDS monitorando o tráfego direto da Internet , e após o *firewall*, temos outro monitorando o tráfego da rede.

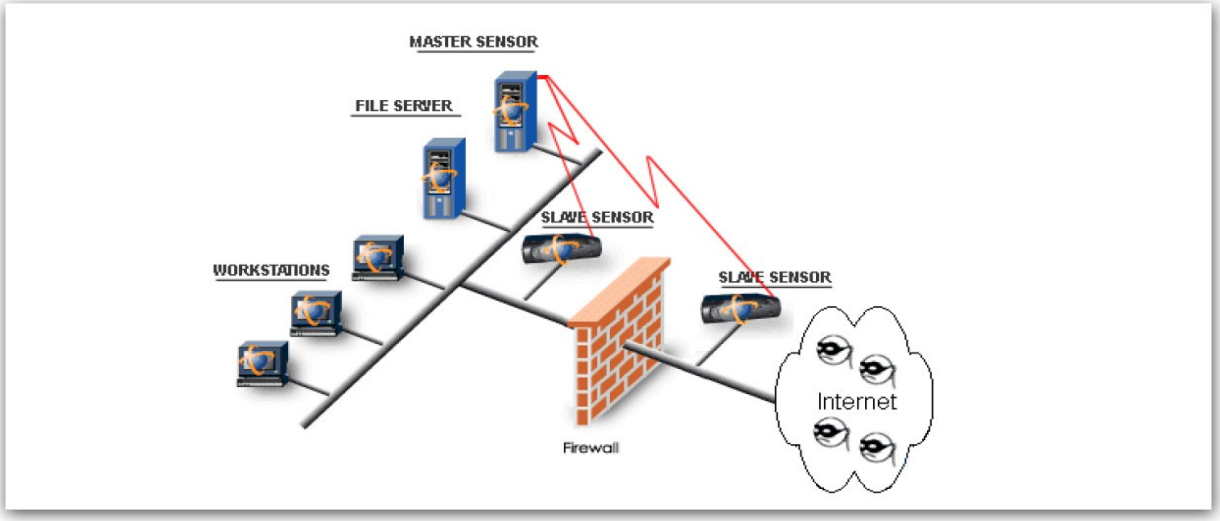


Figura 5. Esquema com 2 servidores IDS[13].

No exemplo da Figura 6, temos um exemplo de servidores IDS monitorando tráfego vindo da internet e o tráfego externo.

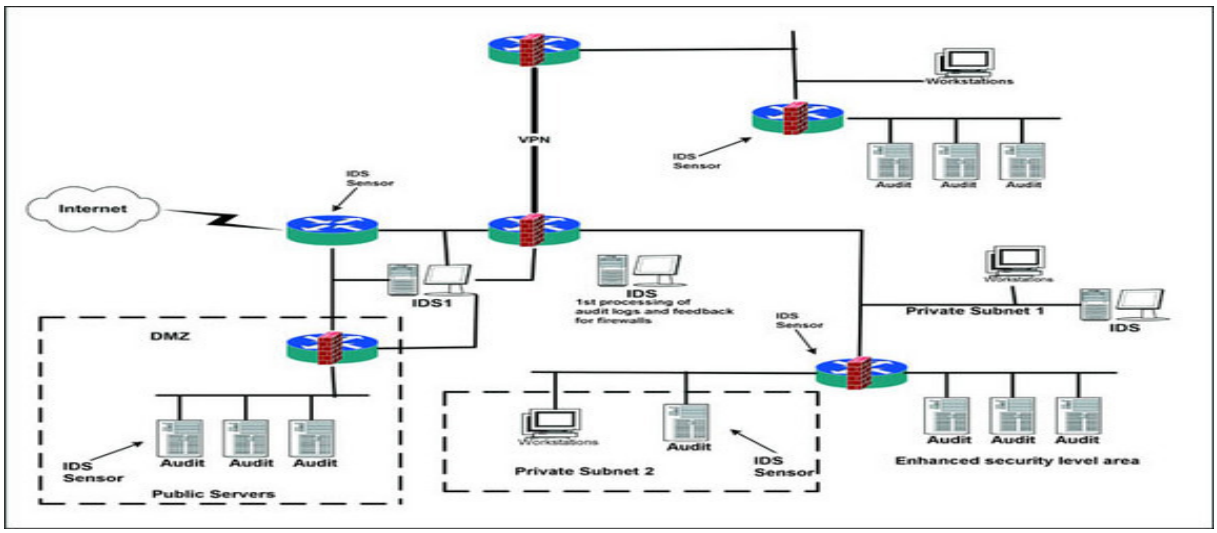


Figura 6. Servidores IDS monitorango o tráfego[14].

[13] http://www.scribd.com/doc/36582402/Ids

Na Figura 7, mostra outro exemplo de um monitoramento de tráfego feito por um IDS, numa tentativa de intrusão.

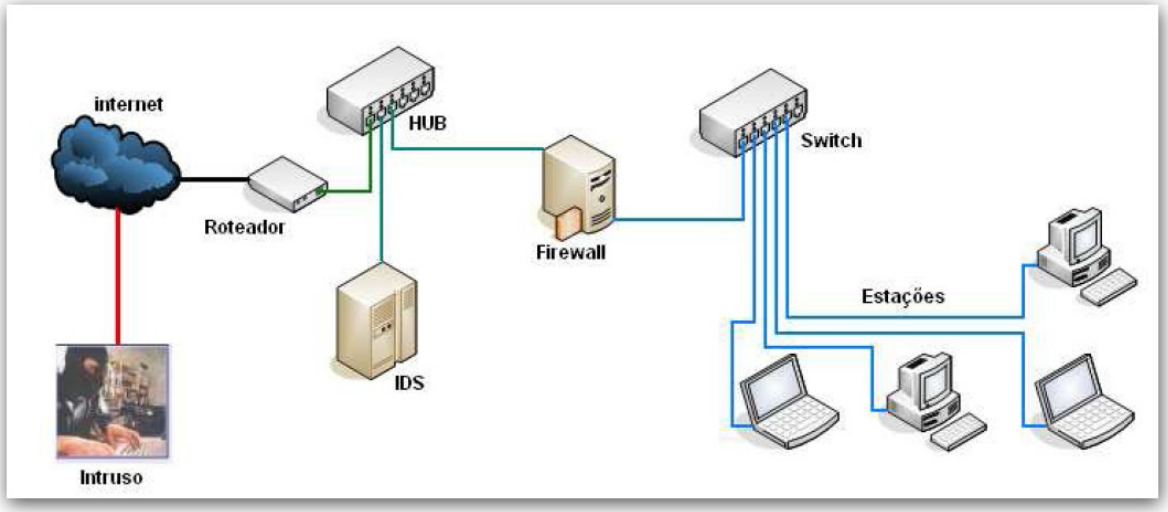


Figura 7. Topologia de um IDS numa tentativa de intrusão[15].

5.3 Ferramenta de IDS SNORT

Será apresentada informações sobre o *Snort*, pois ela será a ferramenta IDS adotada nesse trabalho para manipulação, utilização e aprendizagem para analisar e detectar intrusões através da analise dos seus *logs*.

O S*nort* é uma aplicação de código aberto (*open source*) de detecção de intrusão para rede, capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP. Ele executa uma análise de protocolo, busca, associa padrões de conteúdo e pode ser usado para detectar uma variedade de ataques. Ele suporta as arquiteturas RISC e CISC, e plataformas dos mais diversos tipos, como plataforma Unix, MacOS e Windows[11].

Uma característica forte no *Snort* é a relevante capacidade de gerar alertas em tempo real, que incorpora mecanismos de alerta para o syslog, para arquivo, que pode ser usado por outros *softwares* de segurança de rede.

[14] http://www.windowsecurity.com/img/upl/Miejsce_IDS_Rys41034592917071.jpg

[15] http://www.scribd.com/doc/36582402/Ids

Mas o S*nort* é destinado a monitorar redes TCP/IP pequenas, onde podemos fazer a detecção de uma grande variedade de tráfego suspeito, assim como ataques externos e fornecer argumentos para as decisões de um administrador de rede tratar.

5.3.1 Funcionamento do SNORT

O *Snort*, captura os pacotes e analisa os mesmos segundo assinaturas de reconhecimento de ataques (regras), que seriam padrões de conteúdos dos pacotes que são comparados aos pacotes reais que entram na máquina para tentar reconhecer sua função.

Os conjuntos de pacotes que forem reconhecidos como maliciosos fazem o S*nort* registrar uma tentativa de ataque em seu *log* de alertas. Para tanto o S*nort* consulta regras que decide o que deve ou não ser postado no *log*. Uma regra pega o diagnóstico dado a um pacote e decide como irá registrá-lo. Um *log* é simplesmente um aviso informativo de alguma ocorrência que veio a acontecer, esses avisos ficam normalmente armazenados em arquivos no disco da máquina, ainda que possa ser redirecionado para outra máquina na rede fazendo uso de um banco de dados.

5.3.2 Instalação e configuração do SNORT

A instalação do S*nort* abordado neste trabalho foi realizada num ambiente Linux, com a distribuição Ubuntu versão 10.10. Para a instalação foram necessários os seguintes componentes instalados na máquina:

- Mysql-server
- mysql-query-browser
- php5
- php5-mysql
- libmysqlclient16-dev
- php-gd
- php-image-canvas
- php-image-graph

- libpcap0.8
- libpcap0.8-dev
- pcre-7.0
- libpcre3
- libcre3-dev
- snort-mysql
- apache2
- Base-1.2.7 (Acid_Base)
- Adodb-4.9.3a

Todos esses componentes foram instalados usando o comando "apt-get install" da distribuição Ubuntu, que é baseada em debian, com exceção dos componentes php-image-canvas [16], php-image-graph[17], base-1.2.7 [18], pcre-7.0[19] e o adodb-4.9.3a[20].

Antes de começar a instalação dos componentes é aconselhavel realizar o comando *apt-get update* e logo após o *apt-get upgrade* para atualização do S.O, com as atualizações necessárias para um bom uso do sistema.

Após toda a instalação dos componentes, será feita a configuração do mysql para a gravação dos *logs* do *Snort*. Primeiro no *prompt* de comando usando o comando mysql –u root –p, sendo p a senha definida no momento da instalação do mysql usando o apt-get. Entrando no modo mysql deverá ser criado um database com o nome snort, usando o comando "*create database snort;"* após a criação do

[16] http://archive.ubuntu.schoolnet.lk/ubuntu/pool/universe/p/php-image-canvas/

[17] http://ftp2.kr.freebsd.org/ubuntu/pool/universe/p/php-image-graph/php-image-graph_0.7.1-1_all.deb

[18] http://ufpr.dl.sourceforge.net/sourceforge/secureideas/base-1.2.7.tar.gz

[19] http://ufpr.dl.sourceforge.net/sourceforge/pcre/pcre-7.0.tar.gz

[20] http://ufpr.dl.sourceforge.net/sourceforge/adodb/adodb493a.tgz

database, deverá ser criado também um usuário com o nome *snort* e definir privilégios para o usuário snort poder inserir, consultar, apagar e atualizar nas tabelas criadas no database snort. O usuário precisará ter privilégios de *root*.

Os seguintes comandos deverão ser executados no modo mysql:

mysql> grant INSERT,SELECT on root. * to snort@localhost;

mysql> SET PASSWORD FOR snort@localhost=PASSWORD('seu password escolhido');

mysql> grant CREATE,INSERT,SELECT, DELETE,UPDATE on snort.* to snort@localhost;

mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* snort;

mysql> quit

Após sair da configuração do mysql, deve-se criar as tabelas que o *snort* irá fazer uso para gravação dos *logs* dos pacotes capturados. Acessando a pasta /usr/share/doc/snort-mysql/ como *root*, e executando o comando *zcat create_mysql.gz | mysql –u root –p snort*, serão criadas as tabelas no database snort. Após a criação, deverão existir 16 tabelas, e para verificar a existência das tabelas serão usados os comandos "*use snort";* para selecionar o database em questão e "*show tables;",* que irá mostrar as tabelas existentes.

Após a configuraçao do mysql para comunicação com o Snort, será preciso a instalação do componente adodb. Primeiro será feito uma cópia do arquivo para o diretório /var/www com o seguinte comando *"sudo tar zxf adodb493a.tgz"*, e depois o comando *"sudo rm adodb493a.tgz"* para deletar o arquivo compactado que não será mais necessário. Após as instalações citadas anteriormente, o Base será o ultimo componente a ser instalado, executam-se os seguintes procedimentos: primeiro será criada uma pasta para instalação do base pelo comando "*sudo mkdir /var/www/html".* Após a criação da pasta, será feito uma copia do arquivo base-1.2.7.tar.gz para /var/www/html usando o comando "*sudo cp" ,* com o arquivo na pasta criada, será executada a extração dos arquivos php para execução do Base, pelo comando "*sudo tar zxf base-1.2.7.tar.gz"*.

Mover os arquivos da pasta Base-1.2.7 para pasta base executando o comando "*sudo mv base-1.2.7"*, depois será necessário o acesso ao diretório */var/www/html/base* para gerar uma copia do arquivo de configuração que será usado, usando o comando *"sudo cp base_conf.php.dist base_conf.php"*.

Na parte da configuração, na pasta /etc/snort/ será aberto o arquivo *snort.conf* para configuração do S*nort* que irá permitir o acesso as regras, banco de dados e definir as faixas de rede que o *snort* irá farejar.

Abrindo o arquivo de configuração do *Snort,*o valor do *var HOME_NET* será a faixa da rede interna que no caso é 172.20.1.0/24 e o valor do *var EXTERNAL_NET !$HOME_NET (significa que tudo que não for home_net é considerado externo).* Mais abaixo será descometado a opção de *output database; log, mysql, user=snort password=<senha do banco escolhida> dbname=snort host=localhost,* e mais abaixo descomentar os includes para habilitar as regras existentes na pasta /etc/snort/rules (Onde serão copiadas as regras baixadas do site do Snort[21]) .

Após as configurações necessárias para rodar o S*nort* e para o interesse do trabalho, será preciso configurar o Base para analisar os *logs* que foram inseridos no banco de dados. Para fazer a configuração será preciso acessar o arquivo *base_conf.php* no diretório /var/www/html/base para realização das seguintes alterações:

Em $BASE_urlpath = colocar *'/html/base';* $DBlib_path=*'/var/www/adodb/';* , *$DBtype= 'mysql';* , *$alert_dbname= 'snort';* , *$alert_host='localhost'; , $alert_port=' '; , $alert_user='snort'; , $alert_password= '<senha_escolhida_user snort>'.*

Com essas configuração apresentadas acima, habilitará o uso da ferramenta de detecção de Intrusão[12] .

[21] http://www.snort.org/snort-rules/#rules

5.4 Testes e Resultados

5.4.1 Testes

Para a realização dos testes usando a ferramenta Snort + Base foi necessário instalar os seguintes softwares para os ataques:

- Zenmap
- N-Stalker
- Nessus

O aplicativo Zenmap é responsável pela realização dos teste de Nmap na máquina onde a ferramenta de IDS snort foi instalada e configurada, cujo objetivo é fazer uma varredura de todas as portas abertas na máquina [13].

O aplicativo N-Stalker foi responsável pelo teste de HTTP_inspect que seria um dos pré-processadores do Snort, que normaliza as requisições HTTP para facilitar as regras que analisam os conteúdos dos pacotes HTTP [14].

O aplicativo Nessus também trabalha com ataques usando o pré-processador HTTP_inspect do Snort onde normalmente ele faz ataques usando protocolo *Web* para seus ataques [15].

Segue algumas imagens do uso de cada ferramenta utilizada no momento dos respectivos ataques:

- **Zenmap**

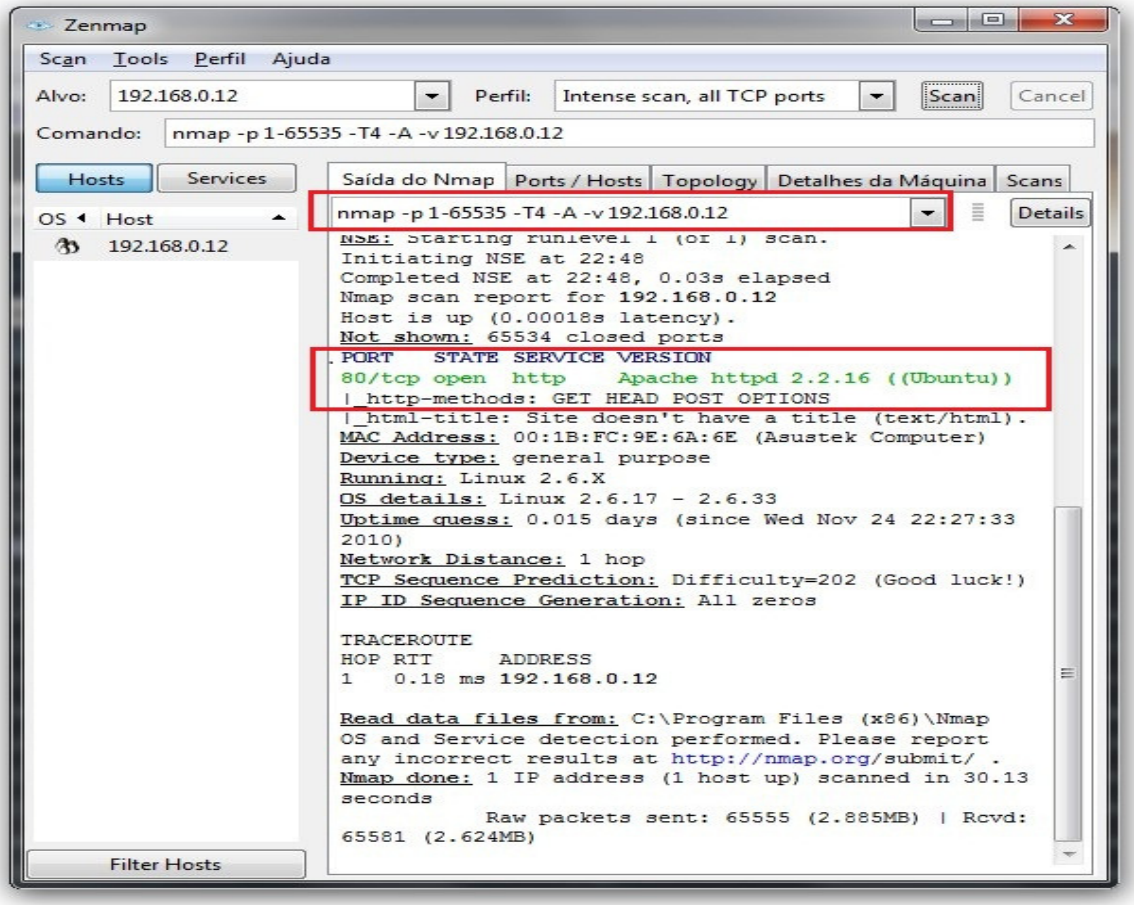


Figura 8. Teste de ataques fazendo uma varredura das portas TCP da maquina alvo[22].

No primeiro item selecionado na Figura 8, o comando nmap -p define as portas para ser scanneada, o qual seria da porta 1 a porta 65535 no exemplo. No segundo item selecionado retorna as portas que estam aberta. No teste da Figura 8, a única porta encontrada foi a porta 80 usando o serviço apache httpd.

[22] http://nmap.org/download.html

- **N-stalker**

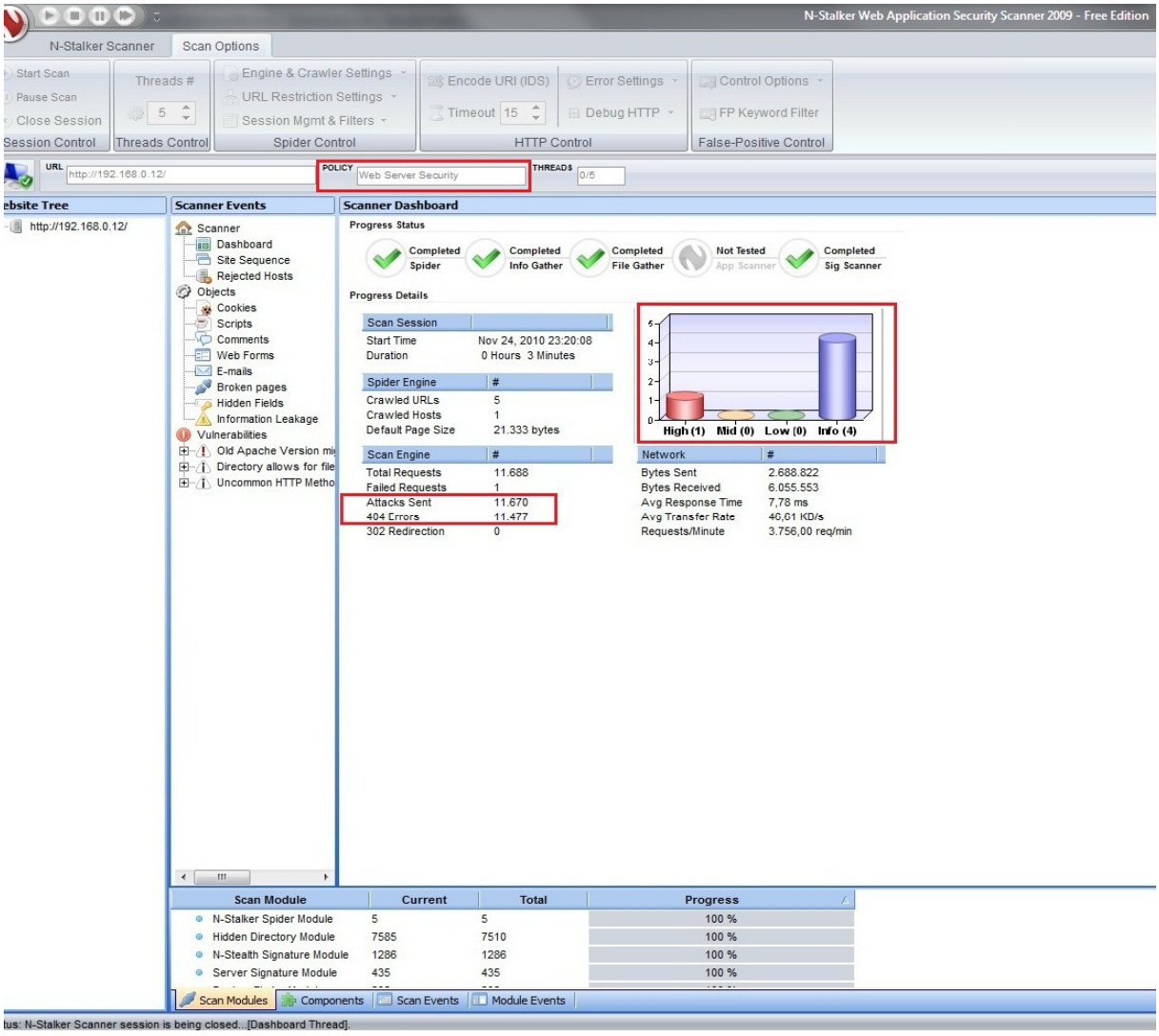


Figura 9. Teste de ataques usando o N-stalker fazendo uma varredura usando protocolo HTTP, e retorno dos resultados do ataque[23].

Na Figura 9, explora a politica de segurança de um servidor web, e no exemplo do teste, são enviados 11.670 pacotes e são bloqueados no teste 11.477, passando 193 pacotes pela segurança do servidor web.

[23] http://www.nstalker.com/

- **Nessus**

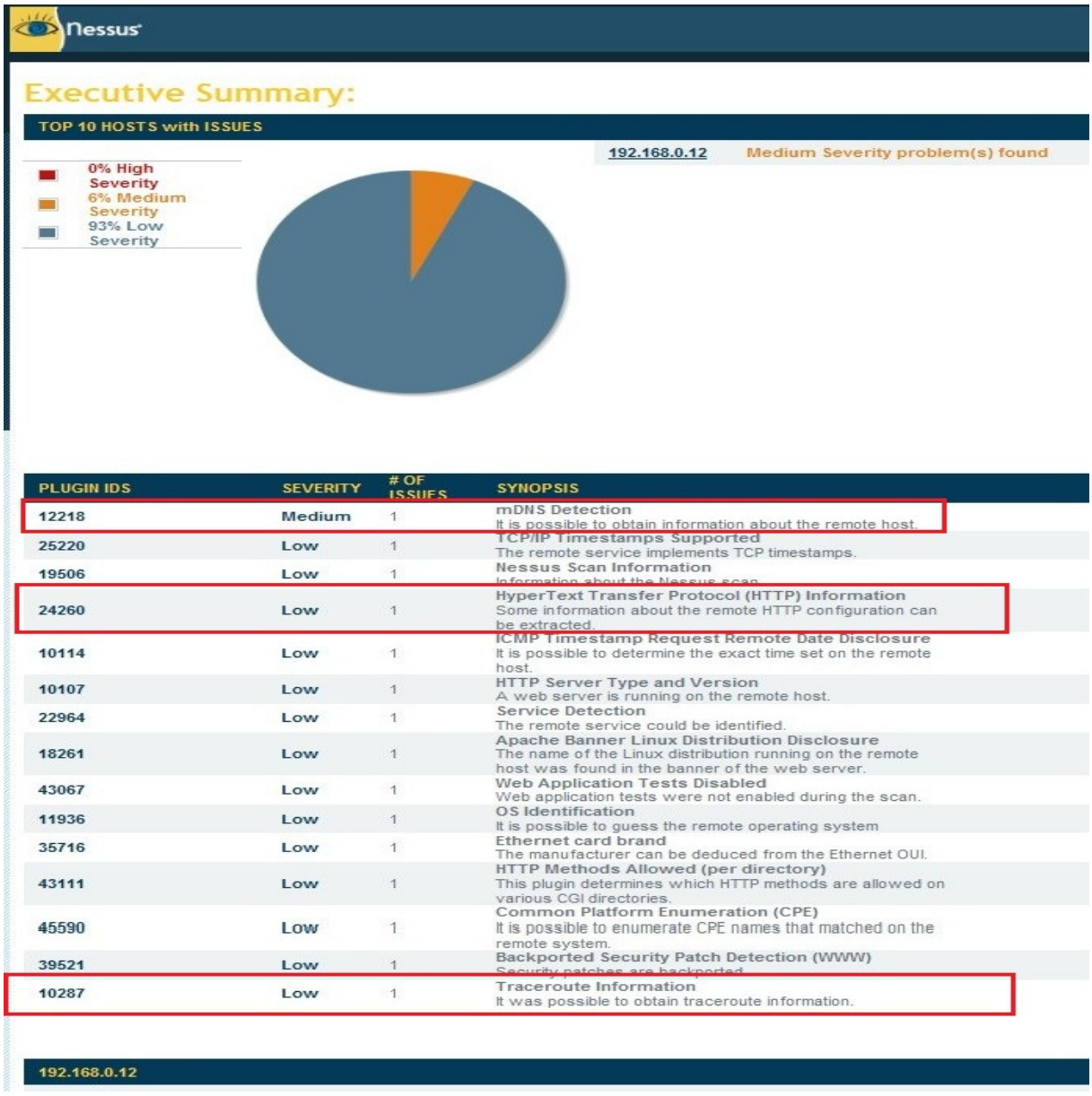


Figura 10. Com uma função de ataques mais elaborada do que o N-stalker, o Nessus faz ataques com mais funcionalidades usando protocolo HTTP[24].

Nos testes da Figura 10, há possibilidades de obtenção de informações sobre o host remoto, configurações remota HTTP e informações do comando Traceroute.

[24] http://www.nessus.org/download/

5.4.2 Resultados

Os resultados obtidos foram bem satisfatórios para os testes realizados, pois em todos os ataques, a ferramenta de IDS Snort retornou avisos ou melhor alertas sobre os ataques específicos que a máquina estava recebendo naquele exato momento.

Segue abaixo os resultados dos ataques retornados e mostra o poder que essa ferramenta tem em se tratando de um aplicativo de código aberto.

- **Alertas contra ataques Nmap**

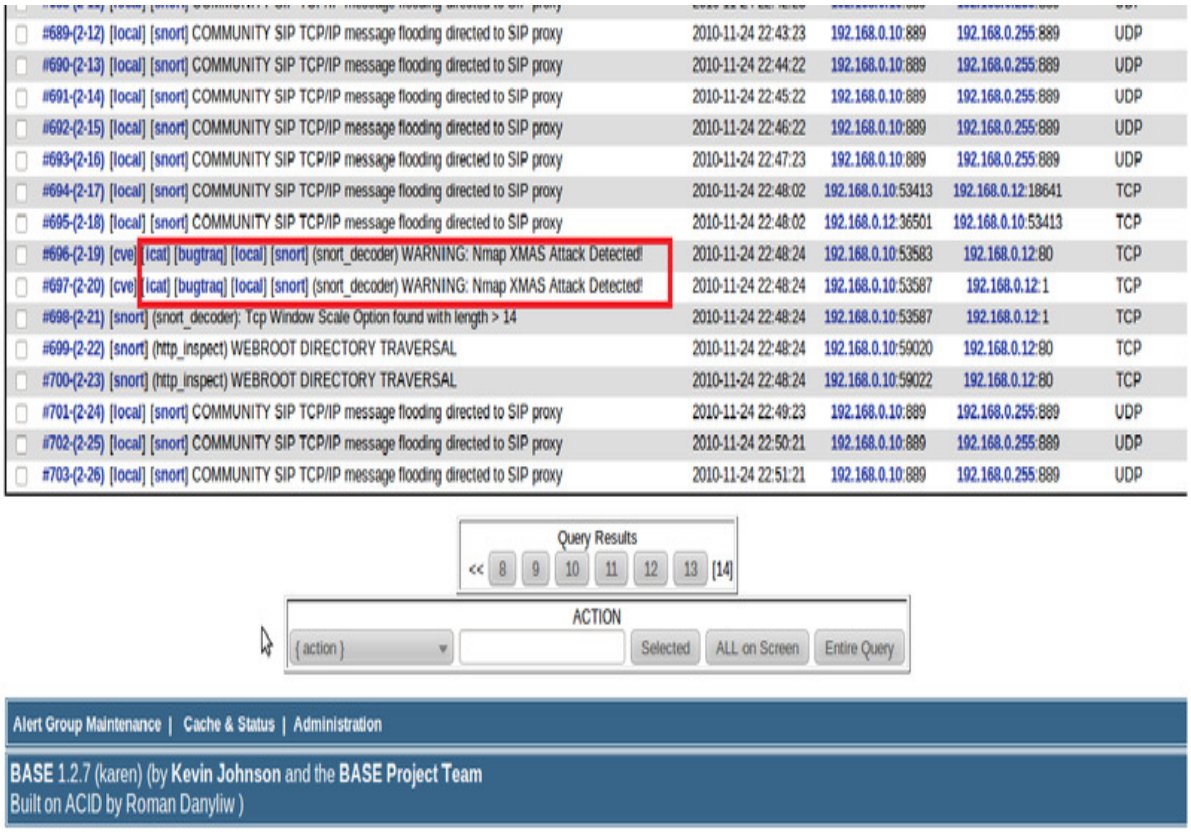
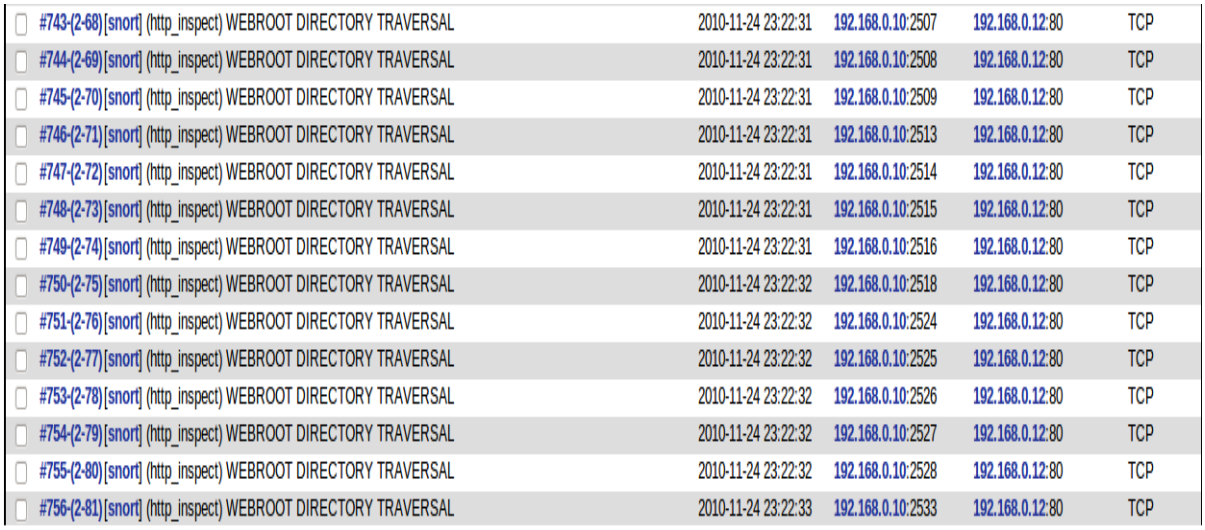


Figura 11. Alerta dos ataques Nmap realizado com a ferramenta Zenmap contra a máquina alvo.

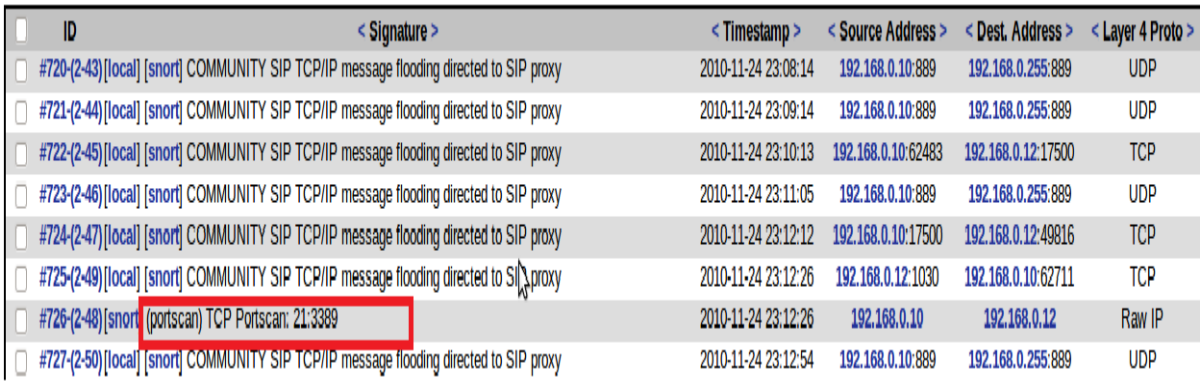
- **Alertas contra ataques usando protocolo web HTTP**



#743-(2-68) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2507	192.168.0.12:80	TCP
#744-(2-69) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2508	192.168.0.12:80	TCP
#745-(2-70) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2509	192.168.0.12:80	TCP
#746-(2-71) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2513	192.168.0.12:80	TCP
#747-(2-72) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2514	192.168.0.12:80	TCP
#748-(2-73) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2515	192.168.0.12:80	TCP
#749-(2-74) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:31	192.168.0.10:2516	192.168.0.12:80	TCP
#750-(2-75) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2518	192.168.0.12:80	TCP
#751-(2-76) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2524	192.168.0.12:80	TCP
#752-(2-77) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2525	192.168.0.12:80	TCP
#753-(2-78) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2526	192.168.0.12:80	TCP
#754-(2-79) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2527	192.168.0.12:80	TCP
#755-(2-80) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:32	192.168.0.10:2528	192.168.0.12:80	TCP
#756-(2-81) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-24 23:22:33	192.168.0.10:2533	192.168.0.12:80	TCP

Figura 12. Alerta de ataque webroot usando o pré-processador do tipo Http_inspect, que alerta ataques do tipo web, usando a ferramenta N-Stalker para o ataque.

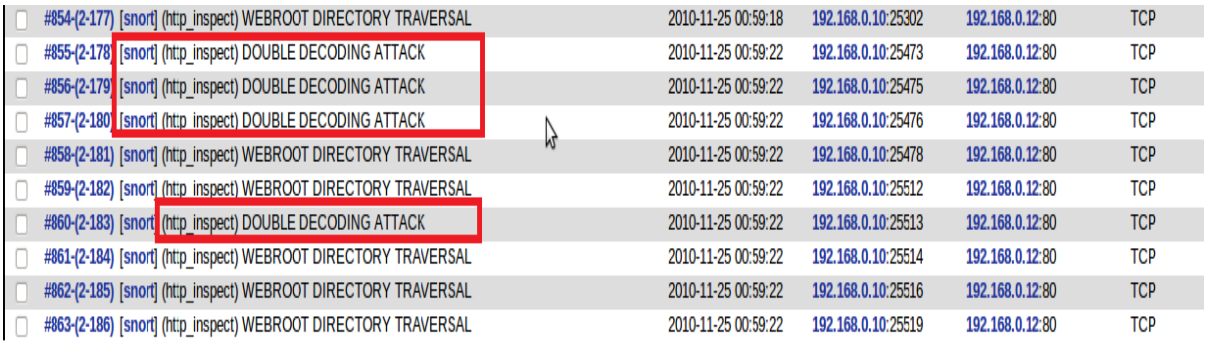
- **Alerta de Ataque de varredura de portas**



ID < Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#720-(2-43) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:08:14	192.168.0.10:889	192.168.0.255:889	UDP
#721-(2-44) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:09:14	192.168.0.10:889	192.168.0.255:889	UDP
#722-(2-45) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:10:13	192.168.0.10:62483	192.168.0.12:17500	TCP
#723-(2-46) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:11:05	192.168.0.10:889	192.168.0.255:889	UDP
#724-(2-47) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:12:12	192.168.0.10:17500	192.168.0.12:49816	TCP
#725-(2-49) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:12:26	192.168.0.12:1030	192.168.0.10:62711	TCP
#726-(2-48) [snort] (portscan) TCP Portscan: 21:3389	2010-11-24 23:12:26	192.168.0.10	192.168.0.12	Raw IP
#727-(2-50) [local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2010-11-24 23:12:54	192.168.0.10:889	192.168.0.255:889	UDP

Figura 13. Alerta de scanneamento das portas da maquina alvo, usando o zenmap.

- **Alertas dos ataques realizado pelo Nessus**

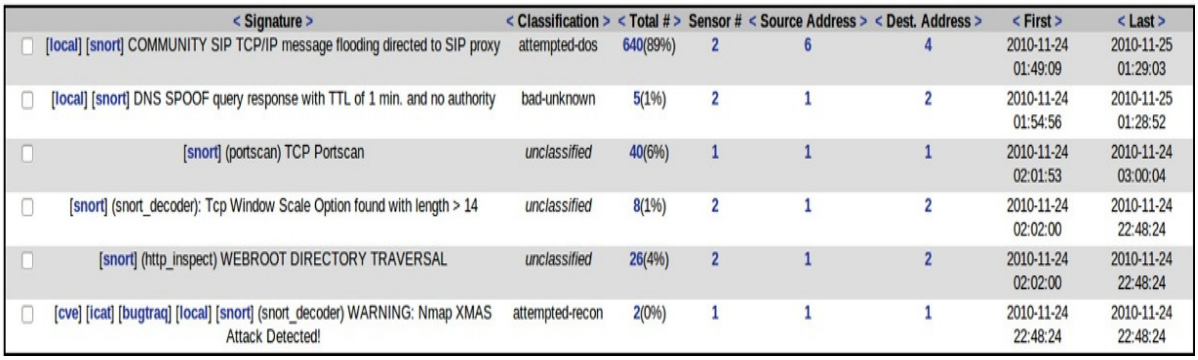


#854-(2-177) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:18	192.168.0.10:25302	192.168.0.12:80	TCP
#855-(2-178) [snort] (http_inspect) DOUBLE DECODING ATTACK	2010-11-25 00:59:22	192.168.0.10:25473	192.168.0.12:80	TCP
#856-(2-179) [snort] (http_inspect) DOUBLE DECODING ATTACK	2010-11-25 00:59:22	192.168.0.10:25475	192.168.0.12:80	TCP
#857-(2-180) [snort] (http_inspect) DOUBLE DECODING ATTACK	2010-11-25 00:59:22	192.168.0.10:25476	192.168.0.12:80	TCP
#858-(2-181) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:22	192.168.0.10:25478	192.168.0.12:80	TCP
#859-(2-182) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:22	192.168.0.10:25512	192.168.0.12:80	TCP
#860-(2-183) [snort] (http_inspect) DOUBLE DECODING ATTACK	2010-11-25 00:59:22	192.168.0.10:25513	192.168.0.12:80	TCP
#861-(2-184) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:22	192.168.0.10:25514	192.168.0.12:80	TCP
#862-(2-185) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:22	192.168.0.10:25516	192.168.0.12:80	TCP
#863-(2-186) [snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	2010-11-25 00:59:22	192.168.0.10:25519	192.168.0.12:80	TCP

O aplicativo Base é uma ferramenta que em combinação com o Snort, permite uma verificação dos logs gerados pelo Snort, apresentando uma interface interativa para o administrador de redes (usuário) analisar.

Fora as análises, o Base cria tabelas dos alertas gerados e ao mesmo tempo também gera uma taxa de porcentagem dos protocolos dos pacotes analisados.

Nas figuras 14 e 15, seguem as tabelas dos alertas gerados na varredura dos pacotes:



< Signature >	< Classification >	< Total # >	Sensor #	< Source Address >	< Dest. Address >	< First >	< Last >
[local] [snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	attempted-dos	640(89%)	2	6	4	2010-11-24 01:49:09	2010-11-25 01:29:03
[local] [snort] DNS SPOOF query response with TTL of 1 min. and no authority	bad-unknown	5(1%)	2	1	2	2010-11-24 01:54:56	2010-11-25 01:28:52
[snort] (portscan) TCP Portscan	*unclassified*	40(6%)	1	1	1	2010-11-24 02:01:53	2010-11-24 03:00:04
[snort] (snort_decoder): Tcp Window Scale Option found with length > 14	*unclassified*	8(1%)	2	1	2	2010-11-24 02:02:00	2010-11-24 22:48:24
[snort] (http_inspect) WEBROOT DIRECTORY TRAVERSAL	*unclassified*	26(4%)	2	1	2	2010-11-24 02:02:00	2010-11-24 22:48:24
[cve] [icat] [bugtraq] [local] [snort] (snort_decoder) WARNING: Nmap XMAS Attack Detected!	attempted-recon	2(0%)	1	1	1	2010-11-24 22:48:24	2010-11-24 22:48:24

Figura 14. Tabela com as porcentagens e o tipo de alertas encontrados no monitoramento dos pacotes.

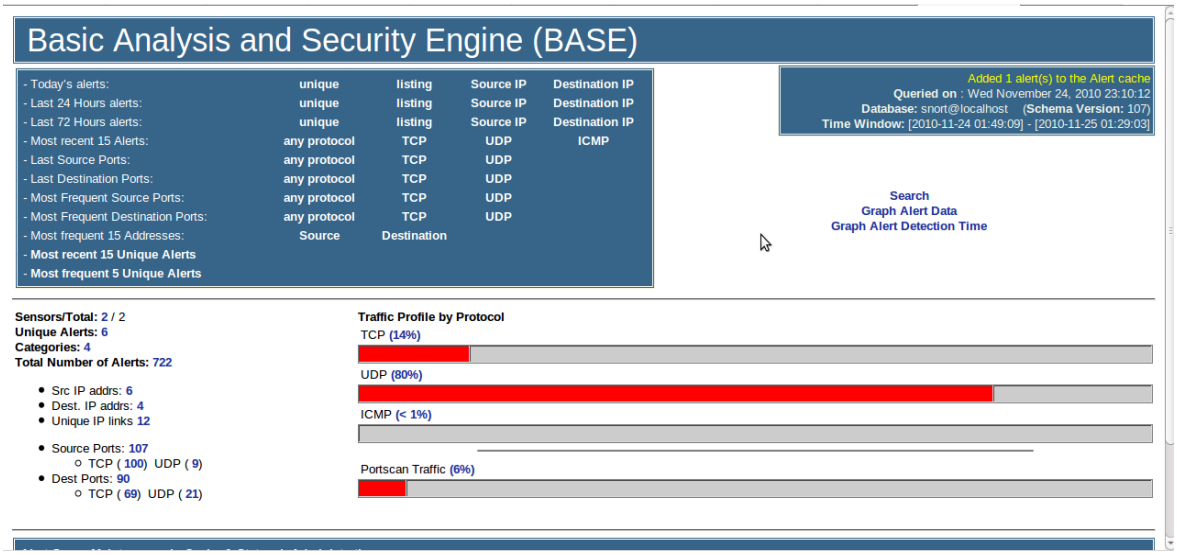


Figura 15. Nível e porcentagens dos protocolos utilizados nos alertas gerados pelo snort.

Capítulo 6
Criação do script para envio SMS de alertas

Para a agregação de envio SMS para avisar ao administrador de rede de ataques o mais rápido possível, foi utilizada a linguagem PHP para envio da mensagem de texto para o celular cadastrado no script.

A classe phpmailer foi utilizada junto com o script que foi criado para a verificação no banco de dados de novas inserções com os alertas mais significativos para o envio do *e-mail*. No mysql uma tabela auxiliar será criada, a tabela sms, na qual serão inseridos os alertas escolhidos do script. O script irá verificar se os alertas possuem o id *signature* e o *timestamp* iguais aos definidos e se dados da tabela sms são diferentes da tabela *event*, caso as condições sejam aceitas, o script insere na tabela sms os novos alertas que foram verificados no banco de dados e após a inserção a função send() criada a partir da classe phpmailer é executada, enviando um *e-mail* para o gateway SMS contratado para o trabalho. O script foi configurado na crontab do ubuntu onde foi definido que a cada 1 minuto o script será executado em background fazendo a verificação de novos alertas para envio ou não do SMS.

O gateway SMS recebe o *e-mail* de um endereço de *e-mail* cadastrado no servidor do gateway e após a verificação do destinatário, a mensagem é encaminhada para o número de celular definido no titulo do *e-mail,* que por exemplo segue o modelo:“16A9BF977AA59778C785F30B1D8BB5DC881C88DB; 2C441d; +55(81)9999-9999; <assinatura do nome para envio do SMS>;”. O envio do SMS pode ser efetuado para todas as operadoras de telefonia celular, sendo que os testes apresentaram um atraso médio de 3 minutos para chegada do SMS.

Capítulo 7
Conclusão e Trabalhos Futuros

As informações trafegadas por empresas, mesmo sendo de vital importância para a mesma, muitas vezes não são tratadas de forma adequada. Alguns resultados dessa falta de preocupação muitas vezes aparecem em jornais e noticiários, quando um ataque em massa ocorre devido a uma falha de segurança de algum aplicativo muito popular utilizado, serviços de e-mail (p.ex. Gmail) ou qualquer aplicativo que provoque a oportunidade de ataque a rede, por um usuário que se aproveitou da brecha (falha) não foi corrigida por um usuario ou adminstrador de rede desatento.

E para esse fim, este trabalho teve como objetivo a apresentação de uma proposta de detecção de invasão, através dos conceitos importantes sobre segurança de redes e da implementação de um servidor IDS. O servidor foi implementado usando os aplicativos Snort, Mysql, o Base e um script php, garantindo a detecção das invasões, armazenamento dos logs, uma leitura facilitada dos logs e o envio de SMS de aviso. Com o servidor em execução foi possível adicionar um nível mais elevado de segurança na rede, no qual se pode detectar as tentativas de invasões.

Além disso, foi desenvolvido um sistema de alerta, para o administrador da rede, via telefonia celular que possui uma grande cobertura, alcançando o administrador rapidamente, onde quer que esteja, afim de que possa adotar as ações necessárias no menor tempo possível, após a ocorrência do ataque.

Logo se pode concluir que com a implementação do servidor IDS não é uma garantia por completa de uma rede 100% segura, levando-se em conta os avanços da tecnologia e das criações de novas técnicas de ataques nos dias atuais, mas que proporciona um nível aceitável de proteção, desde que faça upgrade das ferramentas de forma contínua.

E como mostrado no trabalho, a ferramenta Snort utiliza regras para detecção de intrusão, e por esse motivo, trabalhos futuros podem ser desenvolvidos com a

implementação de redes neurais para aprendizagem e criações de novas regras para os ataques que o Snort não tenha na sua lista de regras, fazendo a própria rede gerar novas regras preventivas para ataques futuros.

Bibliografia

[1] Criptografia-Wikipédia- A enciclopédia Livre. Site: Home Page. Disponível em < http://pt.wikipedia.org/wiki/Criptografia>. Acesso em 10/09/2010.

[2] MELO, E. R.; Redes de Confiança em Sistema de Objetos CORBA; 2003; Disponível em <http://www.das.ufsc.br/~emerson/academico/mello2003-dissertacao.pdf>. Acessado em: 16/09/2010.

[3] Della Valle, J; Ulbrich, H.C. Universidade Hacker: Desvende todos os segredos do submundo dos hackers 6.ed, São Paulo: Digerati Books, 2009 .

[4] Gomes, A. G. A.; Neto, L. T. F. Políticas de Segurança em Redes de Computadores; 2001. Disponível em < http://www.nead.unama.br/site/bibdigital/monografias/Seguranca_redes.pdf>. Acessado em 22/09/2010.

[5] CERT.BR-Centro de Estudo, Resposta e Tratamento de Incidentes de Segurança no Brasil-CGI.br 2010. Site:Núcleo de Informação e Coordenação do Ponto Br- NIC.br. Disponível em: <http://www.cert.br>. Acesso em 28/09/2010.

[6] Exploit –Wikipédia – A enciclopédia Livre. Site: Home Page. Disponível em http://pt.wikipedia.org/wiki/Exploit. Acessado em 10/10/2010.

[7] Infowester.com- Propagando conhecimento. Site:Home Page. Disponível em < http://www.infowester.com/col091004.php>. Acesso em 11/11/210.

[8] Firewall – Wikipédia – A enciclopédia Livre. Site: Home Page. Disponível em < http://pt.wikipedia.org/wiki/Firewall>. Acessado em 15/11/2010.

[9] SCRIBD.com- Site de Compartilhamento de Publicação Social, Documentos e Livros Digitais- IDS. Site: Home Page. Disponível em: < http://www.scribd.com/doc/36582402/Ids>. Acessado em 17/11/2010.

[10] Anticorpos- Toda Biologia-Informações sobre anticorpos. Site: Home Page. Disponível em <http://www.todabiologia.com/genetica/anticorpos.htm>. Acessado em 17/11/2010.

[11] Snort – Wikipédia – A enciclopédia Livre. Site:Home Page. Disponível em < http://pt.wikipedia.org/wiki/Snort>. Acessado em 19/11/2010.

[12] Snort – Ubuntu-Br- Comunidade do Ubuntu no Brasil. Site: Home Page. Disponivel em < http://wiki.ubuntu-br.org/Snort>. Acessado em 22/11/2010.

[13] Zenmap-Nmap.org. Site: Home Page. Disponível em < http://nmap.org/zenmap/> . Acessado em 23/11/2010.

[14] N-Stalker- N-stalker the Web Security Specialists. Site:Home Page. Disponível em < http://www.nstalker.com/>. Acessado em 23/11/2010.

[15] Nessus-Tenable Network Security. Site: Home Page. Disponível em < http://www.nessus.org/nessus/>. Acessado em 24/11/2010.

[16] Charlie, S; Wolfe, P.; Hayes, B. Snort for Dummies 1.ed, Indianapolis: Wiley Publishing, 2004.

[17] Koziol, J. Intrusion Detection with Snort 1.ed, United States of America: Sams Publishing, 2003.