CENTRO PAULA SOUZA
GOVERNO DO ESTADO SÃO PAULO

Faculdade de Tecnologia de Americana

Curso Segurança da Informação

IPTABLES, UMA FERRAMENTA VERSÁTIL PARA SEGURANÇA

GIOVAN BARREIRA

Americana, SP

2013

Faculdade de Tecnologia de Americana

Curso Segurança da Informação

IPTABLES, UMA FERRAMENTA VERSÁTIL PARA SEGURANÇA

GIOVAN BARREIRA

giovanbarreira@bol.com.br

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Segurança da Informação, sob a orientação do Prof. Me. Gabriel de Souza Fedel.

Área: Segurança de Redes

Americana, SP

2013

BANCA EXAMINADORA

Prof. Me. Gabriel de Souza Fedel (Orientador)

Prof. Me. Alexandre Garcia Aguado

Prof. José William Pinto Gomes

AGRADECIMENTOS

Aos meus pais, que desde cedo, me ensinaram o caminho do trabalho honesto.

À minha esposa e à minha filha, que tiveram paciência e abriram mão da minha companhia por três anos.

Ao meu orientador, Gabriel de Souza Fedel , pela ajuda e confiança.

A todas as pessoas que trabalham ou estudam na Fatec de Americana por colaborarem na minha formação.

Aos meus grandes amigos Romieri e Juliana, que me ajudaram e me mantiveram motivado.

DEDICATÓRIA

À minha família, aos meus amigos e a todos os meus mestres dos Ensinos Fundamental e Médio, que acreditaram e me incentivaram para que continuasse estudando.

RESUMO

Nos últimos vinte anos ocorreram muitas mudanças no mundo da computação. Um exemplo ilustrativo é que nos anos 80 e no início dos anos 90 era comum utilizar-se um computador sem conexão em rede. Atualmente, quase todo computador de uso pessoal ou corporativo vai se conectar em uma rede, que pode ser simples ou complexa, composta de algumas máquinas que compartilham uma impressora ou formadas de redes menores que compartilham recursos dos mais variados tipos. Na medida em que as redes crescem, torna-se difícil o controle e o gerenciamento das conexões entre máquinas e redes, tem-se aqui o ambiente perfeito para ataques que tem como finalidade a exploraração de vulnerabilidades ou no mínimo para o desperdício de recursos como banda disponível. Neste trabalho será apresentado o iptables, ferramenta para a configuração de um muro entre uma máquina e uma rede, utilizado contra os ataques de *phishing* e *ping flood.* Serão mostradas também possíveis configurações de tal ferramenta, visando a economia de banda da rede.

Palavras Chave: iptables; *firewall*; redes de computadores.

ABSTRACT

In the last twenty years many changes happened in the world of computing. One example is that in the 80's and 90's early was common to use a computer without a network connection. Nowadays , almost all personal or corporate computer will connect to a network, which can be simple or complex, composed of some machines that share a printer or formed of smaller networks that share resources of all kinds. Insofar as networks grow, it becomes difficult to control and manage the connections between machines and networks ,this is the perfect environment for attacks that aims to exploit vulnerabilities or at least for the waste of resources such as bandwidth. In this work will be presented iptables, tool for setting up a wall between a host and a network , used against phishing and ping flood attacks, are also shown some possible configurations of this tool , in order to save network bandwidth .

Keywords: *iptables; firewall; network computing.*

SUMÁRIO

LISTA DE FIGURAS E DE TABELAS

1 INTRODUÇÃO

Entre o fim dos anos 80 e início dos anos 90, quando vendia-se cursos de Wordstar (editor de textos) e Samba (planilha eletrônica), os laboratórios de Informática destas escolas eram compostos por computadores independentes, ainda não havia a necessidade de criação de redes de computadores pois os editores de texto e as planilhas eletrônicas rodavam exclusivamente na máquina onde estavam instalados.

Atualmente tem-se uma visão bem diferente do quê deve ser e fazer um computador. As máquinas modernas, necessariamente possuem dispositivos para conexão em rede, no caso dos *laptops* mais de um. Hoje é muito comum um usuário editar um texto, salvá-lo e enviá-lo por email para alguém, ou ainda, editar tal texto direto na ferramenta de edição de textos de seu *email*.

Mas as modernas redes de computadores, sejam elas domésticas ou corporativas, são muito flexíveis, compartilham uma ou mais impressoras, compartilham pastas com arquivos ou vídeos que podem ser executados por um computador ou *videogame* e ainda compartilham uma conexão com a *internet*, que permite a comunicação via *email*, webconferência, entre outros.

Então, a rede local composta por algumas máquinas, na maioria das vezes está conectada na *internet*. Esta é livre, ampla, possui usuários do mundo todo, com interesses diversos, alguns são honestos, outros nem tanto.

Para Urubatan Neto (NETO, 2004), a *internet* é bastante insegura e tal insegurança sempre cresce, na medida em que cresce a quantidade de usuários na rede. Além disso, existe uma infinidade de funcionalidades para computadores em rede e uma possível funcionalidade é concentrar páginas ou programas em um computador para que outros acessem. Tais tipos de páginas podem ser: de pesquisa, institucionais, religiosas, de vendas de produtos, que ensinam como furtar carros, de apologia ao nazismo, de venda de drogas ilícitas, de pedofilia, de venda de serviços de *cyber attack*, entre outras e o mesmo ocorrendo com os programas.

O panorama acima citado faz com que os administradores de redes, desenvolvedores de sistemas e outros profissionais da Computação desenvolvam, constantemente, novas ferramentas indispensáveis de filtragem para o conteúdo que

trafega entre computadores.

Diante de um contexto, onde há a necessidade de estar conectado à alguma rede, existe também a necessidade de utilização de algum dispositivo capaz de filtrar a conexão, de forma que uma máquina acesse (e seja acessada por) apenas a máquina correta.

Neste trabalho, o objetivo é a apresentação do funcionamento da ferramenta iptables, presente no sistema operacional Linux, utilizada para configurar o Netfilter (um *firewall*). Serão explicados e apresentados dois contextos de utilização: um mais básico com usos tradicionais da ferramenta; e o segundo apresentando como utilizá-los na defesa contra dois ataques comuns.

Será mostrado ainda, o tipo de usuário a que o iptables se destina, bem como as vantagens da adoção de um *firewall* Linux sobre outras soluções similares existentes atualmente, muitas delas, proprietárias.

Dentre os testes, será apresentado um ambiente onde o iptables será responsável por passar regras de bloqueio de páginas web arbitrárias para o Netfilter, será demonstrado também o funcionamento do iptables para que a rede “ignore” determinados tipos de requisição de serviços o que pode ser usado contra ataques à rede.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados alguns trabalhos relacionados além da terminologia necessária à compreensão deste trabalho.

2.1 CONCEITUALIZAÇÃO E TERMINOLOGIA

Nesta seção são descritos os conceitos e a terminologia básica necessários ao entendimento deste trabalho.

2.1.1 LINUX

Em 1984, Richard M. Stallman iniciou um projeto chamado GNU[1] (GNU's *Not* UNIX) com o objetivo de reescrever o código do sistema operacional proprietário UNIX de forma que o novo sistema operacional pudesse ser distribuído livremente, tal projeto foi registrado na Fundação de *Software* Livre (FSF – *Free Software Foundation*) e foi distribuído entre dezenas (talvez centenas) de programadores para possibilitar o desenvolvimento do novo sistema operacional, bem como um conjunto de utilitários para tal sistema (STALLMAN, 2010).

Segundo Negus (2008), no ano de 1991, Linus Torvalds da Universidade de Helsinki, na Finlândia, começou a desenvolver um sistema operacional (*kernel*) livre baseado no sistema operacional Minix, que foi desenvolvido com fins educacionais por Andrew S. Tanenbaum e é baseado no UNIX. O objetivo de Torvalds era de que seu sistema operacional fosse compatível com seu computador doméstico e com os computadores da Universidade (NEGUS, 2008).

O projeto GNU já havia produzido milhares de utilitários, mas ainda não tinha um *kernel*, quando Torvalds tornou público o seu *kernel,* que foi adotado pela equipe GNU pois era a peça que faltava para ter-se um sistema operacional livre, baseado no UNIX (NEGUS, 2008). Em relativamente pouco tempo surgiu um sistema (*kernel*) estável funcionando sem nenhuma linha de código proprietário (PITTS *et al*, 1998).

1 Disponível em: <www.gnu.org>. Acesso em: 05 dez. 2013

Hoje milhões de desenvolvedores espalhados pelo mundo todo colaboram na manutenção e evolução deste sistema operacional .

O *Kernel* é um programa que é carregado no momento em que o computador inicializa, provendo uma interface entre os programas que rodam em nível de usuário e o *hardware* (PITTS *et al*, 1998. p. 80). O senso comum se confunde ao achar que Linux é uma distribuição, como Mandriva ou Fedora, mas na realidade, Linux é apenas o *kernel* e vale ressaltar que muitas distribuições diferentes possuem exatamente o mesmo *kernel* conforme Urubatan Neto (NETO, 2004).

2.1.1.1 DISTRIBUIÇÕES

Uma distribuição Linux é um conjunto de aplicativos e/ou utilitários rodando sobre um *kernel* Linux (NETO, 2004). Em geral uma distribuição Linux vem com alguns programas para tocarem músicas, vídeos, gravarem dvds, um editor de textos, uma planilha eletrônica, algumas ferramentas amigáveis para a instalação de novos programas e para a configuração do sistema.

As distribuições também são chamadas de *distros* ou sabores do Linux e refletem a filosofia (ou necessidade) da comunidade que ajuda na manutenção e evolução de tais distribuições. Só para citar algumas: Fedora, Mandriva, Mageia, Ubuntu, Deft, Slackware, entre outras.

2.1.2 FIREWALL

O *firewall* é um programa tornado autônomo pelo sistema operacional cuja função é disciplinar qualquer tráfego existente entre máquinas e/ou entre redes de máquinas (RASH, 2007); mas, quando o *firewall* é apenas um componente dentre um conjunto de componentes em que se incluem *hardware* e *software* projetados sob medida para comporem uma solução de controle de tráfego, tem-se um *firewall-in-a-box* (NETO, 2004).

Ainda de acordo com Neto (2004), o *firewall* pode ser dividido em duas

categorias: de filtro de pacote e NAT.

A utilização de um *firewall* é indispensável atualmente, tanto que todo sistema operacional moderno já vem com um *firewall*, pode-se tomar como exemplo o sistema operacional Windows 7 ou ainda o Fedora 16; aqui vale salientar que existem *firewalls* que podem ser descarregados gratuitamente da página de seus fabricantes, como o ZoneAlarm Free Firewall 2013[2], o Comodo Firewall 5.14[3], o McAfee Firewall 7.1 for Windows[4] ou pode-se adquirir as versões pagas de tais produtos, que possuem muitas funcionalidades não disponíveis nas versões gratuítas.

Nas redes corporativas, o *firewall* é muito utilizado para limitar o acesso às páginas de relacionamento ou vídeos fora do horário de almoço; outra importante utilização é para que sejam evitadas as conexões vindas de certos endereços (*blacklists*), o que reduz a possibilidade de ataques; o *firewall* pode ser usado ainda para evitar que um computador entregue informações como a versão do seu sitema operacional e evitar também que um computador responda requisições fora do comum, o que pode significar uma tentativa de ataque.

2.1.3 FIREWALL FILTRO DE PACOTES

A função deste tipo de *firewall* é filtrar o tráfego de pacotes[5] destinado ao computador que está rodando o *firewall* ou à rede que tal computador isola. Este tipo de *firewall* também filtra os pacotes emitidos pelo computador que executa o *firewall* ou pela rede que esta máquina isola (NETO, 2004).

Para saber quais pacotes serão barrados e quais serão permitidos, o *firewall* recebe uma lista de regras desenvolvidas pelo administrador da rede, tal lista contém permissões e/ou proibições, o *firewall* faz um extenso trabalho de comparar o cabeçalho de um pacote com as listas de regras, então tal pacote poderá passar

2 Disponível em: <http://www.zonealarm.com/security/en-us/zonealarm-pc-security-free-firewall.htm>. Acesso em: 05 dez. 2013

3 Disponível em: <http://personalfirewall.comodo.com/>. Acesso em: 05 dez. 2013

4 Disponível em: <http://home.mcafee.com/>. Acesso em: 05 dez. 2013

5 Quando uma máquina envia uma mensagem para outra, tal mensagem precisa ser quebrada em partes menores para ser transmitida pela rede e estas partes menores contém, além dos fragmentos que compõem a mensagem original, um cabeçalho que possue o endereço da máquina de destino e o endereço da máquina de origem, dentre outras informações adicionais (KUROSE e ROSS, 2010).

pela filtragem ou ser barrado (NETO, 2004). Este tipo de *firewall* é, sem dúvida, o mais utilizado pelo mercado e está implementado no Linux desde o *kernel* 1.x (NETO, 2004).

2.1.4 FIREWALL NAT

Da Língua Inglesa *Network Address Translation*, é um processo onde modifica-se a parte referente aos endereços de destino ou de origem de um pacote que trafega na rede, com o objetivo de fazer a comunicação entre redes (RASH, 2007). Neste processo ocorre uma tradução de um endereço interno (da rede interna) para um endereço externo (da rede externa) (RASH, 2007).

Um *firewall* de tipo NAT é extremamente versátil, sendo capaz de manipular a rota padrão dos pacotes que passam pelo *kernel* que o executa, fazendo com que os pacotes sofram “tradução de endereçamento” (ou Nat), estes *firewalls* são capazes de manipular os endereços de origem (SNAT ou *source* NAT) e de destino (DNAT ou *destination* NAT) dos pacotes (NETO, 2004).

2.1.5 FIREWALL HÍBRIDO

O *firewall* híbrido é capaz de desempenhar funções de filtragem de pacotes e redirecionamento de rotas de pacotes, em outras palavras, é um somatório das características dos firewalls citados nas sessões anteriores (NETO, 2004).

Muitos dispositivos de rede, como os roteadores domésticos mais modernos por exemplo, são equipados com este tipo de *firewall* pois precisam fazer a tradução de endereçamento além da filtragem de pacotes.

2.1.6 NETFILTER

O Linux (seu *kernel*) possui uma ferramenta cuja função é o monitoramento e

controle do fluxo de dados do próprio sistema (NETO, 2004), tal ferramenta, criada por Marc Boucher, James Morris, Harald Welte e Rusty Russell chama-se Netfilter, que pode ser encarado, de maneira simplificada, como um banco de dados que possui, por padrão, três tabelas : Filter, Nat e Mangle (EYCHENNE, 2012).

Cada tabela possui direções específicas de fluxo de pacotes, tais direções são chamadas tecnicamente de *chains* e podem ser: *INPUT, OUTPUT, FORWARD, PREROUTING* e *POSTROUTING*.

A tabela Filter guardará as regras referentes à filtragem dos seguintes pacotes: que se destinam ao computador que está executando o *firewall* (*chain INPUT*), que saem do computador que está executando o *firewall* (*chain OUTPUT*) e que passam pelo computador que está executando o *firewall* mas estão destinados a outro computador ou outra rede (*chain FORWARD*) (EYCHENNE, 2012).

A tabela Nat fica responsável por guardar regras referentes à tradução de endereçamento. Esta tabela trabalha com três situações de pacotes *(chains*)*: PREROUTING* (pacotes que devem ser alterados antes de serem roteados), *OUTPUT* (pacotes emitidos pelo computador que está executando o *firewall*) e *POSTROUTING* (pacotes que necessitam ser alterados após o roteamento) (EYCHENNE, 2012).

A tabela Mangle altera pacotes de forma complexa. Esta tabela trabalha com as *chains PREROUTING* (modifica pacotes antes de serem roteados) e *OUTPUT* (modifica pacotes gerados na máquina que está executando o *firewall*) (EYCHENNE, 2012).

As regras que serão armazenadas nas tabelas precisam ser manipuladas por alguma ferramenta, neste trabalho a ferramenta que será apresentada para tal manipulação é o iptables.

2.1.7 IPTABLES

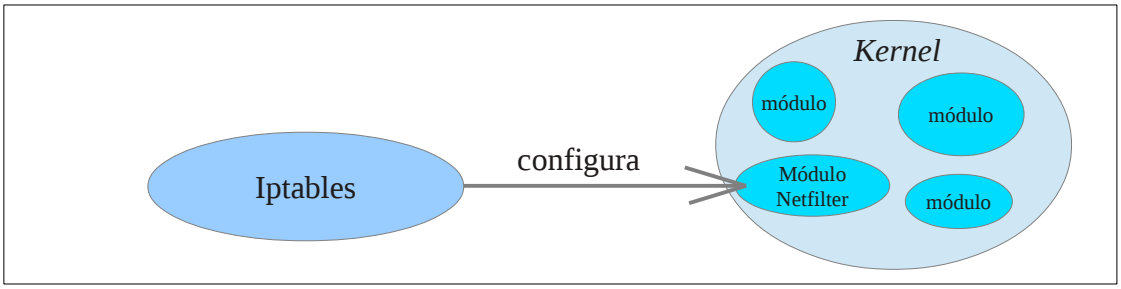
Criado por Rusty Russell, o Iptables, que constantemente é confundido com um *firewall*, na verdade é uma interface de configuração (ou *front-end*) que permite ao usuário manipular as três tabelas do Netfilter que é o verdadeiro *firewall* (NETO,

2004).

O Iptables entrega para o usuário todas as possibilidades de configuração das funcionalidades de um *firewall* híbrido pela configuração e armazenamento de regras nas tabelas Nat e Filter e ainda funcionalidades avançadas, via configuração e armazenamento de regras na tabela Mangle (NETO, 2004) .

É importante ressaltar que via instalação de módulos externos, o iptables pode ser usado para configurar regras compostas, ampliando ainda mais as possibilidades de utilização do Netfilter (NETO, 2004). Na figura 1 tem se a relação entre o *kernel* Linux, o Netfilter e o iptables.

Figura 1 – O iptables e o *kernel*



Fonte: Próprio autor.

2.1.8 AMBIENTE ONDE É NECESSÁRIA A UTILIZAÇÃO DE FIREWALL

Quanto mais pessoas ou empresas estiverem conectadas à *internet*, maior será o valor agregado desta e quanto maior este valor agregado mais atrativo será este ambiente para criminosos (NETO, 2004).

Hoje em dia, praticamente todo computador necessita estar conectado a algum tipo de rede, seja doméstica para compartilhar uma impressora ou conectado à *internet* para executar um aplicativo em nuvem ou acessar uma conta de email, tem-se então um ambiente propício para a ação de criminosos.

Um ataque na rede, pode se dar de forma ativa, quando um atacante consegue se infiltrar em uma rede e acessar um computador, ou de forma passiva, quando um usuário desavisado conecta-se a uma página maliciosa na *internet* e tem seu computador invadido (NETO, 2004).

Nas situações citadas acima a melhor defesa é a filtragem do que se vai acessar e de quem acessará, o mecanismo capaz de fazer tal filtragem é o firewall e a não adoção de tal mecanismo deixará a rede aberta para pacotes não confiáveis o que mais cedo ou mais tarde, retornará em prejuízos (NETO, 2004).

2.2 TRABALHOS RELACIONADOS

O número de redes de computadores continua crescendo na medida em que muitas comunidades ao redor do mundo passam a ter acesso aos sistemas computacionais. E como já foi dito anteriormente, tal crescimento, faz com que a *internet* se torne mais atrativa para criminosos (NETO, 2004), este fato cria um enorme campo de trabalho para as pessoas responsáveis pela segurança dos usuários de sistemas computacionais.

Atualmente há a necessidade de intervenção de profissionais responsáveis pela segurança na rede mundial (*internet*) e também em redes internas, cabe aqui citar o livro de Neto (2004) que introduz o conceito de *firewall*, em seguida descreve o ambiente onde é necessário a adoção de tal ferramenta e aborda de forma clara e concisa a utilização do iptables na configuração do Netfilter, além de abordar a utilização dos módulos extendidos[6]: *limit, state, mac, multiport, string* e *owner*.

Um outro livro interessante neste contexto é o de Rash (2007), que aborda a aplicação do iptables em regras visando a defesa contra vários tipos de abusos, tais como *buffer over flow exploits, sql injection, connection exhaustion, ping flood, phishing*, dentre outros, é interessante ressaltar que Rash (2007) classifica os abusos quanto a camada de rede a que estes pertencem.

Há também um artigo publicado no sítio da IBM[7] onde Deza (2011) mostra a montagem de um conjunto de regras para um servidor e cita que, quando um conjunto de regras está maduro e funcionando bem, tal conjunto se torna complexo, o que impossibilita a recriação de um conjunto de regras igualmente eficiente em um prazo pequeno, portanto é ressaltada a importância de manter um plano de

6 São módulos externos que podem ser descarregados e instalados com a finalidade de ampliar as funcionalidades do iptables (EYCHENNE, 2012) Disponível em: <http://ipset.netfilter.org/iptables-extensions.man.html>. Acesso em 05 dez. 2013.

7 Disponível em: <http://www.ibm.com/developerworks/linux/library/os-iptables/>. Acesso em: 05 dez. 2013.

recuperação de desastres onde são feitos *backups* do conjunto de regras.

Chuvakin (2001) apresenta um artigo, disponível no sítio da Symantec[8], sobre as possibilidades de utilização do módulo extendido *string* na defesa contra *worms* via o monitoramento de pacotes portadores da *string "cmd.exe*" que embora sejam inofensivos para os sistemas Linux, poderiam atingir os sistemas Windows presentes na rede.

Outro ponto importante refere-se aos servidores que disponibilizam os serviços ssh, ftp e telnet, que exigem senhas para acesso. Estes tipos de máquinas podem ser vítimas de tentativas de conexão por parte de um atacante via listas, amplamente compartilhadas em certos meios, com nomes de usuários e senhas (OWENS e MATTHEUS, 2008), este tipo de ataque pode ser classificado como ataque de força bruta baseado em listas (*brute-force attack based on lists)* e pode ser minimizado por regras de restrição configuradas para o Netfilter via iptables nas situações onde ocorrerem sucessivas falhas de *login* originadas em um mesmo endereço (OWENS e MATTHEUS, 2008).

Outra fonte preciosa de informações sobre o iptables e a segurança em redes de máquinas é a obra de Fuller *et al* (2012)[9] que foi desenvolvida com a finalidade de ensinar aos usuários do Linux (a distribuição Fedora 17, para ser mais específico) os processos e práticas de segurança em estações de trabalho e servidores aplicados contra atividades maliciosas e tentativas de intrusão originadas nas redes interna e externa.

Eychenne (2012)[10] escreveu um manual bastante detalhado para o iptables, com mais de 2400 linhas e definições de termos importantes para a compreensão e utilização da ferramenta, bem como com toda a sintaxe dos comandos e parâmetros nativos, tal manual pode ser encontrado no próprio iptables.

Os trabalhos acima, são muito interessantes para qualquer um que tenha sob sua responsabilidade uma rede de computadores e até para aqueles que, embora não sejam responsáveis por nenhuma rede, possuam conhecimentos técnicos suficientes para implementarem o grau de segurança apresentado aqui em suas

8 Disponível em: <http://www.symantec.com/connect/articles/iptables-linux-firewall-packet-string-matching-support>. Acesso em 05 dez. 2013.

9 Disponível em: <http://docs.fedoraproject.org/en-US/Fedora/17/html/Security_Guide/index.html>. Acesso em 05 dez. 2013.

10 Disponível em: <http://ipset.netfilter.org/iptables.man.html>. Acesso em 05 dez. 2013.

redes domésticas.

3 ATAQUES ÀS REDES

Existem inúmeros tipos de ataques às redes de máquinas (RASH, 2007) e estes ataques podem se originar tanto na rede interna quanto na externa (FULLER *et al,* 2012) e podem ter como alvo: a rede em si, de forma que esta se torne inutilizável (KUMAR, SHARMA, SINGH, 2012); o acesso não autorizado a algum tipo de serviço (OWENS e MATTHEUS, 2008) disponível na rede ou ainda o roubo de informações sigilosas que trafegam na rede (RASH, 2007).

Neste capítulo serão apresentados dois tipos de ataques: um ataque clássico de negação de serviços, conhecido como "*ping flood*", cujo principal objetivo é levar a banda da rede a exaustão ou forçar o desligamento da máquina vítima via exploração de algum tipo de vulnerabilidade e um tipo de ataque bastante atual, conhecido como "*phishing*" que objetiva o roubo de dados sensíveis de usuários da rede através da criação de sítios falsos e propagação de *emails* também falsos.

3.1 PING FLOOD

Em condições normais, uma máquina específica atende requisições de outras máquinas da rede de forma otimizada, sem trabalho excessivo, nestas condições tem-se o cenário para um ataque *DoS (denial-of-service*)*,* que consiste em sobrecarregar uma determinada máquina e/ou a rede com pacotes de forma que ocorra uma negação (indisponibilidade) dos serviços prestados (KUMAR, SHARMA, SINGH, 2012).

Existem várias formas de se executar um ataque de negação de serviços (*DoS),* dependendo do tipo de pacote que será enviado no ataque (*icmp* ou *syn* por exemplo) dentre elas, uma das mais simples se dá pela criação de um fluxo intenso de pacotes do tipo *icmp ping.* O pacote *icmp ping* é utilizado normalmente por dispositivos da rede para enviarem informações suas ou receberem informações sobre outros dispositivos (TANENBAUM, 2011). No ataque utilizando o *icmp ping* é direcionado o fluxo destes pacotes contra a máquina e/ou rede que se deseja sobrecarregar, resultando na indisponibilidade de resposta da máquina às

requisições autênticas, tal ataque é conhecido como *ping flood* (KUMAR, SHARMA, SINGH, 2012).

Outra variação do ataque de negação de serviços anteriormente citado é aquela onde o atacante escolhe uma máquina que possui algum tipo de falha de segurança onde um fluxo de pacotes pode fazer com que a máquina fique *offline* (TAIS, 2007).

Em ambas as técnicas citadas acima (sobrecarregar a rede e/ou a máquina ou fazer com que a máquina vítima se torne *offline*), o ataque pode ser iniciado em apenas uma máquina (*DoS*) ou em mais de uma máquina (*DDoS, distributed denial-of-service*) e em qualquer das técnicas, o ataque pode ser iniciado diretamente por um atacante ou indiretamente via acionamento de algum tipo de *malware* que se apodera de máquinas desprotegidas na rede e as faz atacar outras máquinas (TAIS, 2007).

Para se ter uma melhor noção da gravidade deste tipo de ataque, Tais (2007) cita que em 2004 já ocorriam uma média de 4000 ataques entre *DoS* e *DDoS* semanalmente na *internet* e uma instituição, dependendo de seu porte, pode arcar com prejuízos de mais de 100 milhões de dólares se ficar desconectada da *internet* por 24 horas.

3.2 PHISHING

Há uma forma de ataque chamada de *phishing* que consiste em criar-se um sítio, que deve ser praticamente idêntico ao de uma instituição financeira real e depois hospedar-se tal sítio em um domínio que pode ser: comprado, gratuito ou mesmo montado de maneira ilícita em um sítio possuidor de vulnerabilidades, via invasão e independente do tipo de hospedagem, o domínio fraudulento geralmente possui o nome da instituição financeira em algum ponto de sua *url* (MOORE, CLAYTON, 2007).

Depois de criado e hospedado, o sítio falso passa a ser divulgado via *emails* também falsos, que, como o sítio, devem ser praticamente idênticos aos de entidades financeiras reais (MOORE, CLAYTON, 2007). Estes *emails* por possuírem

um conteúdo bastante persuasivo, geralmente solicitando a troca urgente da senha do cartão de crédito, por motivos de segurança, podem confundir um usuário, fazendo com que este entregue, sem perceber, seus dados bancários (inclusive suas senhas) para uma fonte que não é confiável (RASH, 2007).

Este tipo de ataque, que atinge o usuário através dos programas que ele usa, é considerado um dos mais problemáticos dentre os que ocorrem atualmente via *internet* visto que tal ataque consegue burlar algoritmos de criptografia e sistemas de autenticação (RASH, 2007) e somado a isso, para Moore e Clayton (2007) tem-se a inexperiência de profissionais responsáveis por serviços de hospedagem, que muitas vezes têm seus servidores invadidos, passando a atuar como sítios de *phishing* além da lentidão dos serviços de hospedagem no que se refere ao bloqueio dos sítios fraudulentos que podem permanecer operando por mais de 90 horas. Ainda de acordo com Moore e Clayton (2007), 3,5 milhões de pessoas nos EUA entregam seus dados bancários em um sítio de *phishing* todos os anos.

4 O USO DO IPTABLES PARA A SEGURANÇA

Neste capítulo será apresentado o uso da ferramenta iptables na configuração de regras para o *firewall* Netfilter, bem como vantagens na adoção do iptables em relação a outras soluções existentes no mercado. Serão apresentadas regras de configuração básicas, de uso corriqueiro no bloqueio de acesso a certos endereços e então aplicar-se-á tais regras na defesa contra ataques vindos das redes interna e externa. Para ressaltar a importância da adoção de um *firewall* serão apresentados alguns tipos de abusos que ocorrem em redes e os impactos de tais abusos nos usuários da rede. É importante ressaltar que a utilização do iptables, embora seja composta de comandos simples, com o passar do tempo vai se tornando complexa, na medida que o conjunto de regras adotado vai amadurecendo e vão surgindo situações que exigem novas regras; portanto a utilização de tal ferramenta é recomendada para profissionais com conhecimentos intermediários ou avançados sobre Linux, redes, *firewalls* e soluções de *backup*.

4.1 BENEFÍCIOS DA ADOÇÃO DE UM FIREWALL LINUX

A Segurança da Informação é um mercado muito promissor, que está crescendo. Neste mercado são vendidas soluções de *backup*, programas para proteção contra vermes, vírus e outros tipos de códigos maliciosos, além de sistemas de criptografia, sistemas anti-forense e, dentre muitos outros produtos e serviços, os *firewalls*.

Enquanto um *firewall* proprietário pode custar desde R$ 119,00[11] a R$ 20.613,18[12] e ainda exigir uma grande curva de aprendizado, um *firewall* Linux pode ser adquirido gratuitamente, já que vem, por padrão, no *kernel* de qualquer distribuição moderna.

Vale salientar que a curva de aprendizado para operar um *firewall* linux de

11 Disponível em: <http://www.pandasecurity.com/brazil/homeusers/solutions/antivirus>. Acesso em: 07 abr. 2013

12 Disponível em: <http://www.noteaqui.com.br/loja/produtos/firewall-cisco-asa5520-k8-asa-5520-appliance-with-sw-ha-4ge-partnumber-asa5520-k8.html>. Acesso em: 07 abr. 2013.

maneira eficaz é muito grande, mas isto ocorre devido ao grande número de possibilidades de configuração e ainda devido à possibilidade de instalação de módulos que adicionam funcionalidades ao *firewall* (NETO, 2004).

Tal curva de aprendizado não deve ser encarada como desvantagem, pois, produtos caros, que possuem muitas possibilidades de configuração e muitas funcionalidades também exigem investimento em treinamentos e compra de cursos, portanto, grande investimento em tempo de aprendizado (NETO, 2004).

Mas, do ponto de vista exclusivo da segurança, a grande vantagem de um *firewall* Linux é o fato de este ser parte do sistema operacional, ao contrário de *firewalls* domésticos para sistemas operacionais proprietários, que são utilitários, gerenciados pelo SO como se fossem qualquer outro programa (NETO, 2004).

O grande problema de se ter um *firewall* como sub-sistema, é o fato de que, ainda hoje, os sistemas operacionais proprietários mais utilizados no Brasil, sejam em uma residência ou em uma empresa, possuem ainda dificuldades de gerenciamento de memória e de bibliotecas compartilhadas (NETO, 2004).

Em outras palavras, enquanto o *kernel* Linux gerencia muito bem o fluxo de dados que passa por si utilizando a ferramenta Netfilter, o *kernel* do sistema operacional proprietário dominante no mercado gerencia apenas a memória e muitas vezes de maneira insatisfatória, enquanto o restante do fluxo de sua própria estrutura é gerenciado por sub-sistemas e estes, como um *player* ou uma planilha eletrônica, podem parar de responder (NETO, 2004).

Para deixar mais visível a vantagem do *firewall* no *kernel*, pode-se citar que, além do SO proprietário não gerenciar um fluxo de pacotes em seu *kernel*, fazendo com que a aquisição deste SO obrigue o comprador a adquirir um *firewall* como programa, que via de regra, se possuir funcionalidades avançadas, tal *firewall* será de um fabricante diferente do SO (NETO, 2004).

4.2 UTILIZAÇÃO DO IPTABLES

Além de funcionar de forma veloz, eficaz, segura e econômica, o iptables possui muitas possibilidades de utilização, apenas para citar algumas: filtragem de

pacotes, controle de tráfego, controle de utilização de banda, tradução de endereçamento, redirecionamento, mascaramento de conexão, detecção de fragmentos, bloqueio a diversos tipos de ataques (NETO, 2004).

O iptables tem ainda a possibilidade de utilização de módulos externos que adicionam ainda mais funcionalidades ao mesmo. Por exemplo, com a instalação de um módulo que trata de strings, o iptables pode passar para o Netfilter uma regra onde nenhuma máquina da rede poderá conectar a nenhuma outra máquina que contenha em sua url a palavra *bigbrother, facebook* ou qualquer outra palavra arbitrária.

No iptables, assim como em certos *firewalls* físicos, a configuração é, por padrão, volátil, ou seja, ao desligar ou reiniciar a máquina toda a configuração será perdida. Portanto, deve-se salvar toda a configuração e recarregá-la quando for necessário, esta tarefa pode ser feita pelo **iptables-save** em conjunto com o **iptables-restore** ou através de um *script* que carregue a configuração toda vez que a máquina inicializar (NETO, 2004).

4.3 UTILIZAÇÃO TRIVIAL DO IPTABLES

O iptables, como já foi dito, é a ferramenta necessária para a manipulação das regras de tráfego que estão armazenadas nas tabelas e impõem controle às *chains*.

Em uma empresa ou até mesmo em uma residência, as necessidades mais básicas referentes ao controle de dados na rede é a proibição de acesso a certas páginas na *internet*, ou a proibição de tais páginas em determinados horários.

Abaixo será listado um conjunto de exemplos de utilização do iptables seguido de explicações sobre as *flags* utilizadas. Não custa lembrar que todas as regras que serão listadas podem ser testadas em qualquer máquina que possua o sistema Linux com iptables 1.4.12.2 e que, caso não seja utilizado o iptables-save/iptables-restore, ao reiniciar a máquina, a configuração anterior aos testes será restaurada.

Todos os comandos citados neste trabalho podem ser escritos e executados (necessariamente com privilégios de superusuário) diretamente em um terminal em

modo texto ou em um emulador de terminal, como o LXTerminal, o Terminal ou ainda o Konsole. Abaixo tem-se um resumo da sintaxe utilizada no iptables, extraído e adaptado a partir do manual do iptables[13].

iptables -t tabela {-A ou -C ou -D chain} regra -j alvo
iptables -t tabela {-I ou -R chain} posiçãoDaRegra regra -j alvo
iptables -t tabela {-S ou -D chain} posiçãoDaRegra
iptables -t tabela {-F ou -L ou -Z chain} posiçãoDaRegra opções
iptables -t tabela {-N ou -X chain}
iptables -t tabela {-P chain} -j alvo
iptables -t tabela -E nomeAntigoDaChain nomeNovoDaChain

Onde:

tabela = filter, nat, mangle ou tabelaCriadaPeloUsuário
chain = INPUT, OUTPUT, FORWARD, PREROUTING ou POSTROUTING
regra = regraEspecífica -i interface -p tipoDePacote -s endereçoOrigem -d endereçoDestino
alvo = ACCEPT, DROP, REJECT, QUEUE, RETURN ou nomeDaChainDefinidaPeloUsuário

Primeiramente serão apresentados alguns comandos para a manipulação de *chains*. Referente às *chains*, usa-se **-N** para criar uma nova, **-X** para apagar, **-L** para listar as regras contidas em uma *chain,* **-F** para apagar todas as regras contidas em uma *chain*, **-P** para alterar a política padrão e **-Z** para limpar os contadores de *bytes* e pacotes.

Alguns exemplos:

[root@n4200 Giovan]# iptables -N MINHACHAIN

O comando acima criou a *chain* chamada 'MINHACHAIN'.

[root@n4200 Giovan]# iptables -L

13 Disponível em: <http://ipset.netfilter.org/iptables.man.html>. Acesso em 07 dez. 2013.

O comando acima listou todas as *chains* (inclusive MINHACHAIN) e suas respectivas regras (caso já existam regras).

[root@n4200 Giovan]# iptables -L MINHACHAIN

O comando acima listou todas as regras (caso existam) existentes na *chain* MINHACHAIN.

[root@n4200 Giovan]# iptables -L -t filter

O comando acima listou todas as regras (caso existam) existentes na tabela *filter*.

[root@n4200 Giovan]# iptables -F

O comando acima apagou todas as regras contidas em todas as *chains* (inclusive MINHACHAIN).

[root@n4200 Giovan]# iptables -X MINHACHAIN

O comando acima apagou a *chain* chamada 'MINHACHAIN'.

Para manipular as regras em uma *chain* tem-se: **-A** para acrescentar uma regra, **-I** para acrescentar uma regra numa dada posição, **-R** para trocar a posição de uma regra e **-D** para apagar uma regra específica.

Um exemplo:

[root@n4200 Giovan]# iptables -A INPUT -p icmp -j DROP

O comando acima bloqueou o recebimento de pacotes do tipo *icmp* na *chain INPUT*.

Para se especificar parâmetros de regra, usa-se: **-s** para o endereço de origem, **-d** para o endereço de destino, **-p** para especificar o protocolo (dentre tcp, udp, icmp, *all)*, **-i** para a interface de entrada dos pacotes, **-o** para a interface de saída dos pacotes, **-f** para fragmentos a partir do segundo pacote, **-j** indicará qual ação será tomada (DROP, ACCEPT, REJECT) se tal regra for utilizada, **-g** para o processo continuar em outra *chain*, **-m** para um módulo extendido.

Quando se utilizar o parâmetro **-p**, cada protocolo aceita um conjunto de extensões, tem-se então para tcp: **-tcp-flags**, **-syn**, **-sport** e **-dport**; para udp: **-sport** e **-dport** e por último tem-se para icmp: **-icmp-type**.

Um exemplo de bloqueio de qualquer domínio que possua a palavra ''futebol'' em sua *url*:

[root@n4200 Giovan]# iptables -A INPUT -m string --string "futebol" --algo bm -j DROP

No comando acima não apareceu o nome da tabela, por padrão tem-se a tabela *filter*, todo pacote que possuir a palavra (*string*) 'futebol' será bloqueado (*DROP*) e 'bm' é o nome do algoritmo de busca por *strings.* É importante notar que o módulo (-m) *string* é sensível às letras maiúsculas e/ou minúsculas, ou seja, o comando acima bloqueará 'futebol', mas não 'Futebol' ou 'fUteBoL'.

4.4 UTILIZAÇÃO DO IPTABLES PARA DEFESA

Devido ao fato do iptables configurar o Netfilter, que é um *firewall* híbrido e às possibilidades de módulos extendidos, têm-se possibilidades de configuração que podem e devem ser utilizadas na manutenção do bom funcionamento da rede (NETO, 2004).

Para Rash (2007), o iptables pode ser usado na defesa contra *phishing*, *sql injection*, *trojans*, *buffer overflow exploits*, *syn floods*, entre outros tipos de abusos, além do fato de que a utilização do iptables pode ser combinada com com a adoção de outras ferramentas, que somadas, formam um conjunto suficiente para a administração de redes de qualquer porte.

Tanto Neto (2004) quanto Rash (2007) afirmam que o iptables pode ser utilizado para bloquear todos os serviços (e até portas) que não são utilizados e que este tipo de política extremamente restritiva dificulta a ação de *malwares* e de outros tipos de abusos na rede.

Como um conjunto de computadores infectados podem se tornar uma rede de máquinas escravas e podem iniciar um ataque de negação de serviços ou ainda se tornarem servidores de *email spam* com um comando do atacante (TAIS, 2007), fato que geralmente ocorre em virtude da inexperiência dos responsáveis pelas configurações de segurança de tais máquinas (MOORE, CLAYTON, 2007), uma política restritiva severa traduzida em um conjunto de regras no iptables iria reduzir drasticamente a possibilidade de se efetivar estes ataques (RASH, 2007).

Em outras palavras, o bom administrador, além de minimizar a possibilidade de invasão à sua rede, diminui também a possibilidade de que suas máquinas sejam utilizadas por criminosos para praticarem outros tipos de abusos e para tanto, estes administradores necessitam do iptables (NETO, 2004) além de outras ferramentas (RASH, 2007).

5 CASOS DE TESTE

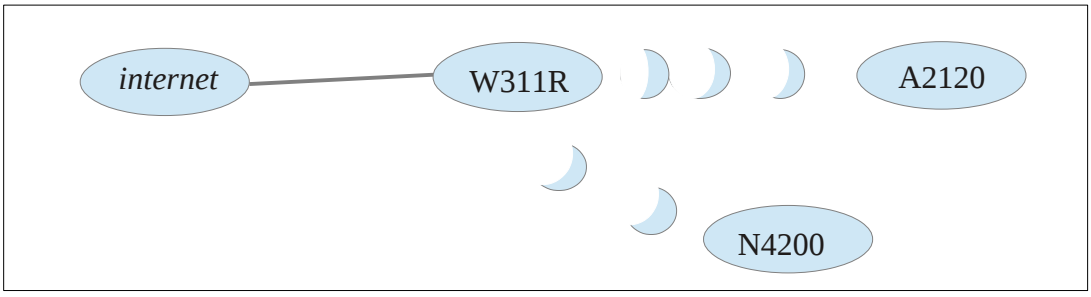
Neste capítulo será feita uma relação entre cada um dos ataques do capítulo 3 com alguns exemplos de configuração do iptables com o objetivo de reduzir o impacto do ataque ou impedí-lo. Quando possível será mostrada mais de uma forma de contenção ou ao menos redução do impacto dos ataques acima citados.

5.1 AMBIENTE DE TESTE

Em todos os exemplos de configuração do iptables citados neste trabalho, o ambiente de teste foi composto por (representado na figura 2):

- um *laptop* Positivo modelo N4200, processador Intel Atom D525 e 4 GB de memória RAM, com o sistema operacional Linux 3.9.10-100;
- um *laptop* Positivo modelo A2120, processador Intel Celeron T3100 e 4 GB de memória RAM, com o sistema operacional Linux 3.9.10-100;
- um roteador *wireless* Tenda modelo W311R, 150 Mbps e padrão 'b/g/n'.

figura 2 – rede *wireless,* ligando as máquinas ao roteador.



Fonte: Próprio autor.

5.2 OS TESTES

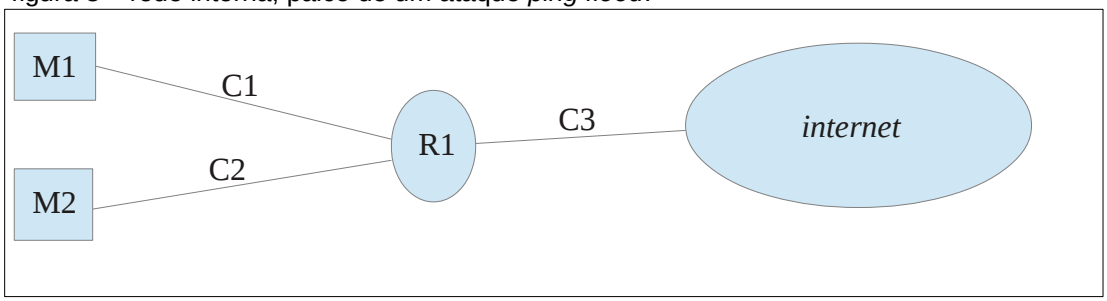
Neste trabalho foram executados testes onde foram abordados dois tipos de impacto na máquina/sistema vítima: no processamento e na utilização da rede. Para mostrar que a adoção de um *firewall* é indispensável (e se este for o Netfilter, a sua

configuração pode ser feita a contento pelo Iptables) foram realizados dois ataques, um do tipo *ping flood* e outro do tipo *phishing*, primeiramente sem nenhuma regra de *firewall* implementada e depois com regras específicas visando a contenção dos ataques. As regras utilizadas para evitar ou ao menos minimizar o impacto dos ataques são explicadas em detalhes, lembrando que toda a sintaxe das regras, em princípio são de uso exclusivo no iptables.

5.2.1 PING FLOOD

Suponha uma rede simples com duas máquinas M1 e M2 conectadas (conexões C1 e C2) em um roteador R1 e este conectando esta pequena rede à *internet*, pela conexão C3 conforme a figura 3.

figura 3 – rede interna, palco de um ataque *ping flood*.



Fonte: próprio autor.

Um ataque pode partir da rede externa (*internet* por C3) ou pode pertencer à rede interna, no caso M1 ou M2 (e atacar por C1 ou C2). Em qualquer caso, um ataque com êxito pode saturar a rede ou fazer com que alguma máquina conectada à rede se torne *offline* se esta possuir alguma vulnerabilidade específica (RASH,2007).

5.2.1.1 COMANDOS NECESSÁRIOS

Primeiramente, suponha que seja necessário o monitoramento do fluxo de

dados em uma rede, para tal tarefa, neste trabalho será utilizado o comando:

[root@n4200 Giovan]# iptraf-ng

Para verificar-se a configuração do *firewall* em uma máquina, utiliza-se os comandos:

[root@localhost liveuser]# iptables -L INPUT
[root@localhost liveuser]# iptables -L OUTPUT

Onde o **-L** significa que serão listadas as regras da *chain INPUT* na primeira linha de comando e OUTPUT na segunda linha. Estes comandos retornaram que a *chain INPUT* aceita pacotes do tipo *icmp* (*ping*) de qualquer endereço e a *chain OUTPUT* aceita o envio de qualquer tipo de pacote pois está vazia.

Alternativamente, pode-se listar as regras de todas as *chains* de uma só vez:

[root@localhost liveuser]# iptables -L

Caso as *chains* contenham regras, tais regras podem ser apagadas de todas as *chains* simultaneamente com o comando:

[root@localhost liveuser]# iptables -F

Ou, pode-se apagar regras de apenas uma *chain*, por exemplo a *forward*:

[root@localhost liveuser]# iptables -F FORWARD

Agora será criada uma regra na tabela *filter*, *chain output*, que recusará 60 pacotes *icmp* por segundo:

[root@n4200 Giovan]# iptables -t filter -A OUTPUT -p icmp -m limit --limit 60/s -j DROP

No comando acima, após o **-A** deve vir o nome da *chain* e após o **-p** deve vir o tipo de pacote, se não for digitado o **-p** e um tipo de pacote, a regra valerá para todos os tipos de pacotes e por último, após o **-j** deverá vir a opção aceitar (*ACCEPT*) ou bloquear (*DROP*).

5.2.1.2 IMPACTO

Neste ambiente, com ambas as máquinas executando apenas os processos naturais do SO, M1 desferirá um ataque de *ping flood* contra M2, que inicialmente não possui qualquer regra de restrição aos pacotes do tipo *ping*, será um ataque originado na rede interna (se M2 possuísse um sistema operacional vulnerável a tal ataque, esta reiniciaria ao receber o fluxo).

O comando em M1 que inicia o ataque:

[root@n4200 Giovan]# ping -f -s 65000 192.168.0.103

Onde “-f” inicia um fluxo de pacotes do tipo *ping* com tamanho (-s) de 65000 bytes endereçado à M2 que está em 192.168.0.103.

Após o término do fluxo, o comando *ping* (em M1) registrou que em 32 segundos foram enviados 331 pacotes de 65000 bytes e recebidos 21 pacotes de 65000 bytes, o que resultou em um fluxo médio de 715.000 bytes/segundo.

Na tabela 1, gerada a partir do comando *top,* tem-se a utilização máxima do processamento (extraída a maior leitura dentre 12 leituras).

Tabela 1 – Utilização do processamento sob ataque

Quem estava utilizando o processamento:	M2 em repouso	M2 recebendo o fluxo sem regra de bloqueio no *firewall*	M2 recebendo o fluxo com regra de bloqueio de pacotes *ping* em M2	M2 recebendo o fluxo com regra de bloqueio de pacotes *ping* em M1
usuário	0,3 %	0,4 %	0,5 %	0,3 %
sistema	0,3 %	0,3 %	0,3 %	0,3 %

Fonte: próprio autor.

Quando foram estabelecidas regras no iptables em M2 para recusar os pacotes do tipo *ping* nas *chains input* e *output*, em 38 segundos foram enviados 785 pacotes de 65000 bytes e não foi recebido nenhum (devido às regras de restrição em M2), resultando em um fluxo médio de 1.342.763 bytes/segundo.

Conforme os dados mostrados na tabela 1, vale ressaltar que a máquina M2 utilizou a menor taxa de processamento estando em repouso ou quando o ataque originado na máquina M1 foi contido na própria máquina M1; outro fato importante é que a máquina M2 utiliza menos processamento recebendo o fluxo sem a implementação de regras do que quando possui regras implementadas referentes ao fluxo, mas aqui, embora as regras consumam pequena parte do processamento da máquina, está possui maior proteção contra alguma vulnerabilidade desconhecida que possa existir e possa fazer com que a máquina reinicie (NETO, 2004). Conclui-se então, que a melhor ação a ser tomada, referente apenas a maior economia de processamento, é a adoção de regras para coibirem o ataque na máquina M1, mas a opção mais segura é a adoção de regras em ambas as máquinas.

5.2.1.3 DEFESA

De acordo com os dados apresentados acima, levando-se em conta apenas a saturação da rede, a tentativa de contenção do ataque ou pelo menos a redução do impacto não obteve sucesso quando as medidas foram tomadas na máquina M2. A melhor forma de evitar o esgotamento da banda da rede é evitando que se inicie um ataque dentro da rede interna, tarefa facilmente executada via regras de bloqueio de envio de pacotes *icmp* aplicadas no netfilter via iptables.

Um possível comando para tal tarefa é (em M1, evitando que esta inicie um ataque):

[root@n4200 Giovan]# iptables -t filter -A OUTPUT -p icmp -j DROP

Mas a solução mais segura é a adoção de regras de bloqueio em ambas as máquinas nos sentidos de entrada (*chain input*) e saída (*chain output*) dos dados; tal

medida além de dificultar o esgotamento dos recursos da rede, dificultaria também que alguma das máquinas reiniciasse (mesmo possuindo alguma vulnerabilidade) enquanto estivesse recebendo o fluxo.

Por último a situação acima, onde o ataque vem de dentro da própria rede não pode ser ignorada pelos administradores pois boa parte dos ataques bem sucedidos inicia-se dentro da própria rede (NETO, 2004).

5.2.2 PHISHING

Este tipo de ataque só é possível mediante a existência de um sítio falso hospedado em algum domínio na *internet* em conjunto com um *email* também falso enviado para um usuário mal informado. Primeiramente, do ponto de vista das máquinas na *internet,* a adoção de um conjunto de regras do iptables bem aplicadas em servidores *web*, reduziria o número de máquinas infectadas, reduzindo também o número de ataques (MOORE, CLAYTON, 2007); agora, do ponto de vista das máquinas que estão na rede interna, um computador que possua determinado conjunto de regras pode estar melhor protegido do que um computador que não possui um *firewall* ou possui, mas, mal configurado (NETO, 2004).

Para Moore e Clayton (2007) a esmagadora maioria dos ataques de *phishing* se utilizam de uma *url* que contenha o nome de uma instituição real no final, mas tal ataque parte de um servidor que não faz parte do conjunto de servidores da instiuição real, portanto, um conjunto de regras do iptables que permita a conexão nos servidores da instituição real (a partir de seu endereço *ip)* e bloqueie qualquer tentativa de conexão em um domínio que possua o nome da instituição (através do módulo *string*) funcionará muito bem. Um exemplo de tal conjunto de regras é apresentado mais adiante neste trabalho.

5.2.2.1 COMANDOS NECESSÁRIOS

Para configurar regras de *firewall* que combinam endereços *ip* e *strings* é

necessário conhecer todos os endereços *ip* de tal sítio, para tanto utiliza-se o comando "*dig*" seguido da *url* do sítio em questão ou o comando “iptraf-ng”. Exemplo:

[root@n4200 Giovan]# dig www.itau.com.br

O comando acima retorna o endereço *ip* do sítio do Banco Itaú.

Alternativamente:

[root@n4200 Giovan]# iptraf-ng

Então acessa-se o domínio www.itau.com.br e a ferramenta iptraf-ng exibirá todos os endereços *ip* do sítio do Banco Itaú.

5.2.2.2 IMPACTO

Diferentemente do ataque do tipo *ping flood*, que atinge a rede, no *phishing*, o alvo é a privacidade do usuário, que inadvertidamente pode entregar seus dados bancários para um criminoso, resultando também em um impacto nas finanças do usuário (RASH, 2007). Estima-se que, só nos EUA, milhões de usuários são afetados por este tipo de ataque anualmente e cada usuário perde mais de 500 dólares por ataque (MOORE, CLAYTON, 2007).

5.2.2.3 DEFESA

Criar um conjunto de regras de maneira que se garanta o acesso a uma determinada página partindo de uma combinação de uma palavra em sua *url* com o seu endereço *ip*, evitando assim o *phishing.*

Exemplo:

[root@n4200 Giovan]# iptables -A OUTPUT -d 23.11.111.170 -j ACCEPT

[root@n4200 Giovan]# iptables -A OUTPUT -d 165.254.158.57 -j ACCEPT

[root@n4200 Giovan]# iptables -A OUTPUT -d 74.125.234.36 -j ACCEPT

[root@n4200 Giovan]# iptables -A OUTPUT -d 173.194.39.126 -j ACCEPT

[root@n4200 Giovan]# iptables -A OUTPUT -d 74.125.28.95 -j ACCEPT

[root@n4200 Giovan]# iptables -A OUTPUT -m string --string "itau" --algo bm -j DROP

No iptables, as regras são impostas de cima para baixo, ou seja, qualquer pacote que não tenha como destino algum dos *ips* acima será bloqueado se possuir a palavra "itau" na *url* para onde tal pacote se destina. Na data de 14/10/2013 o Banco Itaú possuía apenas os *ips* acima, portanto este conjunto de regras bloqueou os sítios argentinos do Itaú e deixou passar o domínio brasileiro, em outras palavras este conjunto de regras bloqueou os domínios que possuíam "itau" em sua url, menos o domínio brasileiro (dono dos *ips* acima). Mas é muito importante salientar que, caso ocorressem mudanças nos *ips* do Itaú, as regras necessitariam ser atualizadas e para tanto, o responsável pela rede poderia escrever um programa (via *shell script* ou outra linguagem) que atualizasse automaticamente as regras ou gerasse um aviso de que as regras precisariam ser atualizadas.

6 CONCLUSÕES

Com a evolução dos computadores, as redes se tornaram comuns e versáteis pois possibilitam o compartilhamento de recursos (como impressoras ou pastas por exemplo) e a conexão com outras redes, como a *internet,* mas esta, além de ser bastante insegura, pode conter sítios com conteúdos ilegais. Tem-se então o *firewall* como um elemento imprescindível diante da necessidade de se filtrar a conexão entre máquinas, seja com o objetivo de aumentar a segurança ou visando evitar o desperdício da banda da rede e neste caso, quando tem-se um sistema Linux no papel de *firewall*, a configuração deste é muito flexível e versátil se ocorrer via iptables.

As possibilidades, quando se usa o iptables vão das mais simples até as mais complexas; neste trabalho, foi mostrado que o iptables é uma ferramenta poderosa, que pode ser utilizada com sucesso contra os ataques de *phishing* e *ping flood,* além de alguns exemplos de utilização básica e algumas vantagens, como o baixo custo, se comparado às soluções proprietárias, que além de gastos na aquisição, exigem gastos em treinamentos e, algumas vezes, em atualizações.

Embora existam ferramentas de configuração de *firewalls* mais fáceis de operar em situações básicas, o iptables é amplamente utilizado em condições onde se possui redes maiores e mais complexas, que requerem mecanismos de filtragem que bloqueiem não apenas sítios, mas serviços e até portas do sistema, exigindo para tanto, um usuário com razoável nível de experiência. Portanto, de acordo com a experiência e necessidade do administrador, o iptables pode ser utilizado de forma complexa e em conjunto com outras ferramentas visando uma maior proteção e controle da rede.

Vale citar três pontos importantes, neste contexto, que não foram explorados e poderiam ser pesquisados como trabalhos futuros: a utilização do iptables no balanceamento de carga quando se liga uma ou mais redes internas à *internet* por mais de uma conexão; a utilização da ferramenta como auxiliar na execução de ataques contra uma rede com a finalidade de encontrar vulnerabilidades e um estudo sobre as diferenças fundamentais no funcionamento (e na utilização para defesa) do iptables em contraste com o ip6tables.

REFERÊNCIAS

EYCHENNE, Herve. *Iptables 1.4.14 manual page. @PACKAGE_VERSION@.* 2012.

FULLER, Johnray. *et al.* ***Fedora 17 - Security Guide****,* 2012.

IBM. Portal developerWorks. Biblioteca Técnica, divisão de Software Livre. Disponível em: <http://www.ibm.com/developerworks/br/library/os-iptables/>. Acesso em: 17 out. 2013.

KRAWETZ, Neal. ***Introduction to Network Security.*** Charles River Media. Boston. United States of America. 2007.

KUMA, Ashish; SHARMA, Ajay K.; SINGH, Arun. *Performance evaluation of centralized multicasting network over ICMP ping flood for DDoS.* ***International Journal of Computer Applications*** (0975 – 8887) . Volume 37, No.10, January 2012 .

KUROSE, James F.; ROSS, Keith W. ***Computer Networking – a top-down approach.*** Boston: Addison Wesley Higher Education, 2010.

MOORE, Tyler e CLAYTON, Richard. ***Examining the Impact of Website Take-down on Phishing.*** University of Cambridge. Cambridge, United Kingdom. 2007.

NEGUS, Christopher. ***Linux Bible.*** Indianapolis: Wiley Publishing, 2008.

NETFILTER. Disponível em: <http://www.netfilter.org/index.html>. Acesso em: 27 out. 2013.

NETO, Urubatan. **Dominando Linux Firewall Iptables**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.

OWENS, Jim e MATTHEWS, Jeanna. ***A Study of Passwords and Methods Used in Brute-Force Ssh Attacks.*** Clarkson University. Potsdam, United States of America, 2008.

RASH, Michael. ***Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort*****.** San Francisco: No Starch Press, 2007.

STALLMAN, Richard. ***The GNU Project.*** Disponível em: <http://www.gnu.org/gnu/thegnuproject.html>, Acesso em: 05 dez. 2013.

SYMANTEC, Portal connect. *Security Community*, *Articles*, Disponível em: < http://www.symantec.com/connect/articles/iptables-linux-firewall-packet-string-matching-support>, Acesso em: 17 out. 2013.

TANENBAUM, Andrew S. **Redes de computadores.** 5a Ed. Rio de Janeiro: Editora Campus, 2011.

TAIS, Ciprian Andrei. ***General analysis of the Economy behind ddos attacks*****.** Polytechnic University of Bucharest. Bucharest, Romania, 2007.