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

DESENVOLVIMENTO DE *FIREWALL* COM HARDWARE DE BAIXO DESEMPENHO E FÁCIL CONFIGURAÇÃO EM NÍVEL DE USUÁRIO

TRABALHO DE CONCLUSÃO DE CURSO

Anderson Cunha Petry

Santa Maria, RS, Brasil

2013

TCC/REDES DE COMPUTADORES/UFSM, RS PETRY, Anderson Cunha Tecnólogo 2013

DESENVOLVIMENTO DE *FIREWALL* COM HARDWARE DE BAIXO DESEMPENHO E FÁCIL CONFIGURAÇÃO EM NÍVEL DE USUÁRIO

Anderson Cunha Petry

Trabalho de Conclusão de Curso (TCC) do Curso Superior de Tecnologia em Redes de Computadores, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores

Orientador: Prof. Ms. Walter Priesnitz Filho

Santa Maria, RS, Brasil

2013

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

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de conclusão de Curso

DESENVOLVIMENTO DE *FIREWALL* COM HARDWARE DE BAIXO DESEMPENHO E FÁCIL CONFIGURAÇÃO EM NÍVEL DE USUÁRIO

elaborado por
Anderson Cunha Petry

COMISSÃO EXAMINADORA

Walter Priesnitz Filho, Ms.
(Presidente/orientador)

Guilherme Dhein, Ms. (UFSM)

Rogério Correa Turchetti, Ms. (UFSM)

Santa Maria, 18 de Janeiro de 2013

RESUMO

TRABALHO DE CONCLUSÃO DE CURSO
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES
UNIVERSIDADE FEDERAL DE SANTA MARIA

DESENVOLVIMENTO DE *FIREWALL* COM HARDWARE DE BAIXO DESEMPENHO E FÁCIL CONFIGURAÇÃO EM NÍVEL DE USUÁRIO

AUTOR: ANDERSON CUNHA PETRY
ORIENTADOR: WALTER PRIESNITZ FILHO
Data e Local da Defesa: Santa Maria, 25 de Janeiro de 2013

Este trabalho apresenta o estudo e desenvolvimento de um *firewall* com hardware de baixo desempenho e fácil configuração em nível de usuário, com o intuito de desenvolver uma ferramenta que seja de fácil manuseio por parte dos usuários com pouco conhecimento em informática.

A aplicação é desenvolvida com ferramentas baseadas em *software* livre, e utiliza o *iptables* como ferramenta de segurança, tendo como principal objetivo o controle do acesso aos conteúdos da internet.

Palavras-Chave: Segurança.*Firewall*.IPTables

ABSTRACT

COMPLETION OF COURSE WORK
SUPERIOR COURSE OF TECHNOLOGY IN COMPUTER NETWORKS
FEDERAL UNIVERSITY OF SANTA MARIA

FIREWALL WITH HARDWARE DEVELOPMENT OF LOW PERFORMANCE AND EASY SETUP LEVEL USER

AUTHOR: ANDERSON CUNHA PETRY
ADVISER: WALTER PRIESNITZ FILHO
Defense Place and Date: Santa Maria, January 25, 2013

This paper presents the study and development of a hardware firewall with low performance and easy setup user level, in order to develop a tool that is easy to handle for users with little computer knowledge.

The application is developed with tools based on open source software, and uses iptables as a security tool, with the primary objective of controlling access to content on the internet.

Keywords: Security.Firewall.IPTables

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ACL	-	*Access Control List*
ASPs	-	*Application Service Providers*
DNAT	-	*Dynamic Network Address Translation*
FTP	-	*File Transfer Protocol*
HTML	-	*Hyper Text Markup Language*
HTTP	-	*HyperText Transfer Protocol*
ICMP	-	*Internet Control Message Protocol*
IP	-	*Internet Protocol*
IPSEC	-	*Internet Protocol Security*
IPV6	-	*Internet Protocol version 6*
IRC	-	*Internet Relay Chat*
ISDN	-	*Integrated Services Digital Network*
ISP	-	*Internet Service Provider*
NAT	-	*Network Address Translation*
PGP	-	*Pretty Good Privacy*
PHP	-	*Personal Hypertext Preprocessor*
RADIUS	-	*Remote Authentication Dial In User Service*
RAM	-	*Random Access Memory*
SGBD	-	Sistemas Gerenciadores de Banco de dados
SNAT	-	*Static Network Address Translation*
SO	-	Sistema Oparecional
SQL	-	*Structured Query Language*
SSH	-	*Secure Shell*
SSL	-	*Secure Socket Layer*
TCP	-	*Transmission Control Protocol*
TELNET	-	*Telecommunication Network*
TI	-	Tecnologia da Informação
UDP	-	*User Datagram Protocol*
URL	-	*Uniform Resource Locator*
VPN	-	*Virtual Private Network*
WWW	-	*World Wide Web*

SUMÁRIO

1 INTRODUÇÃO

Um sistema de comunicação é um conjunto de mecanismos que permite que uma informação seja transportada através de um meio desde sua origem até o seu destino (MORAES, 2010). Ou seja, as redes de computadores são caracterizadas como telecomunicações, pois há presença de equipamentos nesta comunicação, em que a origem e o destino das informações são equipamentos da tecnologia da informação, e o canal de comunicação pode ser guiado ou não guiado. As redes de computadores têm crescido muito nas duas últimas décadas, e o que era uma raridade, tornou-se uma parte essencial no cotidiano de todos.

O avanço no desenvolvimento de novas tecnologias para o aprimoramento da utilização de computadores fez com que o investimento em ferramentas de proteção contra invasores acompanhasse este crescimento. Outra necessidade de quem gerencia redes de dados é o monitoramento do conteúdo nela presente, tanto para qualificar o teor do que está sendo acessado, bem como para fazer o melhor uso da infraestrutura disponível.

De acordo com SANGARA (2011), o desenvolvimento da internet proporcionou muitos benefícios para quem usufrui desta tecnologia, em contrapartida, também expõe os usuários com pouco conhecimento nas questões de segurança, ou com certo grau de inocência (e neste quesito, em grande parte, jovens e crianças) à mercê de criminosos, que passaram a utilizar o meio virtual como cenário de delitos. Ainda segundo SANGARA (2011), os crimes de pedofilia vêm tendo um aumento considerável, pois os criminosos, além de interagir com outros infratores de diversas partes do mundo, podem passar um perfil falso para adquirir a confiança de quem está interagindo consigo em salas de bate-papo, sites de redes sociais ou programas de trocas de mensagens instantâneas. Neste contexto, a autora apresenta dados estatísticos sobre crimes de pedofilia na Internet, onde a 4ª Delegacia de Investigações Gerais do estado de São Paulo informa que 8% dos crimes praticados ciberneticamente referem-se à pedofilia.

Ainda abordando o tema sobre o desenvolvimento da internet, é importante salientar que novas tecnologias de software, na maioria das vezes, requerem maior capacidade de hardware. Baseada nesta evolução tecnológica, onde em curtos espaços de tempo novas famílias de processadores, discos rígidos com maior capacidade e dispositivos de memória física volátil, comumente conhecida como memória RAM, são lançados, a obsolescência e

depreciação dos equipamentos de informática ocorrem de forma rápida, gerando desta forma uma grande quantidade de lixo tecnológico. Segundo FLORES (2012), o lixo tecnológico é um dos causadores de diversos problemas que estão afetando o meio ambiente, onde a conscientização das pessoas sobre a reutilização de equipamentos de informática, bem como o descarte em locais apropriados são algumas medidas que podem ser tomadas para prevenir diversos problemas relacionados ao clima que estão ocorrendo atualmente, além de evitar possíveis danos à saúde dos seres humanos e dos animais.

Deste modo, este estudo será guiado pelo objetivo de desenvolver um sistema de *firewall* doméstico que não necessite de um grande desempenho de hardware, mas que seja suficiente para suprir as necessidades de acesso, ou que garanta a restrição de acesso a conteúdos não desejados em ambientes residenciais, com a finalidade de que cada usuário administrador possa definir os conteúdos que poderão ser acessados, ou conteúdos que deverão ser bloqueados, possibilitando desta forma a melhoria da qualidade do acesso aos conteúdos da internet.

Assim, tem-se por objetivos específicos: analisar diferentes Sistemas Operacionais para trabalhar com a versão que exige menos recursos de hardware; trabalhar com ferramentas, tais como *IPTables* e *Cron*, com a finalidade de elaborar um sistema de *Firewall* que satisfaça às necessidades dos usuários; elaborar uma interface que seja amigável, e de fácil configuração por parte do usuário; testar a funcionalidade, tanto da interface com o usuário, quanto da eficiência da ferramenta de *firewall* desenvolvida e analisar se está de acordo com o proposto.

Frente ao exposto, este estudo é de relevância para o desenvolvimento de novas tecnologias de proteção, e o aprimoramento de ferramentas já existentes. Este trabalho está estruturado da seguinte forma: o capítulo 2 trata do assunto Segurança, abordando conteúdos como Política de Segurança, *firewall* e *iptables.* O capítulo 3 irá apresentar a solução proposta, desde o seu planejamento até a sua conclusão, e também serão apresentados os ambientes de testes. O capítulo 4 apresenta alguns resultados obtidos em decorrência dos testes realizados, e no capítulo 5 são apresentadas as considerações deste trabalho.

2 SEGURANÇA

Este capítulo irá abordar os conceitos básicos de segurança em redes de computadores, além de uma breve descrição sobre alguns itens para a elaboração de uma política de segurança, além da abordagem sobre os temas *firewall* e *iptables*.

2.1 Estudo sobre Segurança

As questões de segurança são primordiais quando falamos em redes de computadores e são muito importantes, tanto em sistemas cabeados quanto em redes sem fio (*wireless*). Segundo Tanenbaum (2003), as redes de computadores, no início de sua existência, eram usadas principalmente por pesquisadores universitários, com a finalidade de enviar mensagens de correio eletrônico, e por funcionários de empresas para o compartilhamento de impressoras, e deste modo, a segurança nunca precisou de atenção especial. Mas, com o aumento de usuários comuns de TI, a utilização das redes tomou novos rumos, e algumas de suas principais usabilidades destinaram-se a serviços bancários, transações comerciais, envios de informações restritas e dados confidenciais, fazendo com que a segurança começasse a se tornar uma questão relevante, pois a maioria dos problemas de rede passou a ser gerado de forma intencional por pessoas maliciosas (NAKAMURA E GEUS, 2007).

Thomas (2007) define que elaborar políticas de segurança é o primeiro e mais essencial passo para proteção da rede, pois fornecem a base para definir o que aceitável, e qual o comportamento da rede de uma empresa. Moraes (2010) define segurança da seguinte forma:

- Um sistema é seguro se ele se comporta da forma que você espera que ele o faça;
- Um computador é seguro se você pode depender dele e o software possui o comportamento que você espera dele;
- Segurança de computadores é prevenir ataques com objetivos definidos através de acessos não autorizados, ou usos não autorizados de computadores e redes.

Ressaltando ainda a importância da política de segurança, Thomas (2007) exibe um quadro com as regras de política de segurança mais comuns seguidas de uma breve descrição, o qual é exibido abaixo:

NOME DA POLÍTICA	DESCRIÇÃO
Criptografia aceitável	Fornece um guia que limita o uso da criptografia àqueles algoritmos que tenham se provado efetivamente funcionais. Adicionalmente, fornece diretrizes que garantam que as leis e regulamentações aplicáveis sejam seguidas.
Uso aceitável	Descreve quem pode usar computadores e equipamentos de rede possuídos pela empresa. Cobre os computadores da empresa localizados dentro da empresa e aqueles dentro das casas dos funcionários.
Linha analógica	Explora o uso aceitável da linha ISDN e analógica e as políticas e procedimentos de aprovação. Regras separadas se aplicam às linhas que sejam conectadas para o simples propósito de fax e recepção e para as linhas que estejam conectadas aos computadores
Provedores de Serviços de Aplicação	Descreve as exigências da empresa de Provedores de Serviço de Aplicação (ASPs – *Application Service Providers*). (ASPs combinam softwares hospedados, tecnologia de hardware e de rede para oferecer um aplicativo baseado em serviços). Refere-se e incorpora as Políticas Padrões ASP separadas.
Padrões ASP	Define os critérios mínimos de segurança que um provedor de serviços de aplicação (ASP) deve atingir para serem usados.
Auditoria	Fornece a autoridade para que membros do time do Departamento de Segurança de Informação para conduzir uma auditoria de segurança em qualquer sistema possuído pela empresa ou que tenha sido instalado com base em suas premissas.
Encaminhamento automático de e-mail	Impede o envio não autorizado ou inadvertido de informações sensíveis da empresa
Credenciais do banco de dados	Determina as exigências para se armazenar e

	recuperar com segurança nomes e senhas de usuários (ou seja, credenciais de banco de dados) para serem usados por um programa que acesse um banco de dados disponível em uma das redes da empresa.
Acesso Discado	Estabelece regras que protegem a informação eletrônica de ser inadvertidamente comprometida por pessoal autorizado usando uma conexão discada.
Extranet	Descreve a política sob a qual organizações externas se conectam as redes da empresa para fins de transações comerciais.
Sensibilidade da informação	Ajuda os funcionários a determinar qual informação pode ser enviada a não-empregados, e a sensibilidade relativa da informação que não deve ser enviada sem autorização adequada.
Segurança interna dos laboratórios	Estabelece exigências de informações de segurança para laboratórios, para garantir que informações e tecnologias confidenciais não sejam comprometidas, e que os serviços de produção e demais interesses sejam protegidos das atividades dos laboratórios.
Antivírus	Estabelece exigências que devem ser cumpridas por todos os computadores conectados às redes da empresa para garantir a detecção e prevenção efetiva contra vírus.
Senha	Estabelecem um padrão para criação de senhas seguras, a proteção destas senhas e a freqüência de alteração.
Acesso remoto	Define os padrões para se conectar a rede da empresa a partir de qualquer ponto. Estes padrões são projetados para se minimizar o potencial de exposição a danos tais como perda de dados confidenciais ou sensíveis da empresa, propriedade intelectual, danos a imagem publica, danos críticos aos sistemas internos e assim por diante.
Levantamento de riscos	Dá poderes ao departamento de segurança de informação em efetuar levantamentos periódicos de risco de segurança com o propósito de determinar

	áreas de vulnerabilidade, e para iniciar a remediação apropriada.
Segurança de roteadores e switches	Descreve a configuração mínima para todos os roteadores e switches conectados à rede de produção ou utilizados em um local de produção.
Segurança do servidor	Estabelece padrões para a configuração base dos equipamentos internos de servidores que são possuídos e/ou operados na empresa ou em localizações de pontos da web.
Rede Privada Virtual	Fornece diretrizes para o acesso remoto as conexões IPSec ou L2TP VPN para a rede corporativa da empresa
Comunicação sem fio	Estabelece padrões para o acesso a rede da empresa via mecanismos de comunicação sem fio (wireless)

Quadro 1 – POLÍTICAS DE SEGURANÇA MAIS COMUNS.
FONTE: THOMAS (2007, P. 44)

Existem soluções, tecnologias e dispositivos que permitem garantir um ambiente seguro, das quais, segundo Moraes (2010) se destacam:

- Criptografia: garante confidencialidade e integridade através da matemática. A encriptação é o processo de transformação de dados claros em uma forma ilegível, mantendo a informação escondida para qualquer usuário que não seja o destinatário.
- Assinatura digital e certificados digitais: é uma tecnologia que permite dar garantia de integridade e autenticidade a arquivos eletrônicos. É um conjunto de operações criptográficas aplicadas a um determinado arquivo, que permite comprovar que a mensagem ou arquivo não foi alterado e que foi assinado pela entidade ou pessoa que possui a chave criptográfica (chave privada) utilizada na assinatura.
- Biometria: é a tecnologia que o individuo é realmente quem diz ser através da análise de características físicas. É o sistema mais seguro, mas em compensação, o mais caro.
- *Firewall*: é um sistema que atua como um ponto único entre a rede privada e a rede pública. Todo tráfego da rede deve passar pelo *firewall*, que pode bloquear e aceitar, além de registrar tudo o que está passando por ele.

Comer (2007) destaca que uma rede de computadores não pode ser classificada simplesmente como segura ou não segura, pois não há uma definição padrão para segurança, e

cada empresa define o seu próprio nível de acesso. Uma política de segurança não especifica como obter proteção, mas sim como os itens serão protegidos. As políticas de segurança são complexas, pois envolvem o comportamento humano, e isto é difícil de prever, e somente poderão ser definidas a partir do momento que se souber o valor que possuem as informações. Para definir uma boa política de segurança, é preciso considerar:

- Integridade dos dados: refere-se à proteção contra mudanças, não podendo ser alterados em seu percurso;
- Disponibilidade de dados: refere-se à proteção contra a interrupção do serviço, permitindo que os dados fiquem acessíveis a maior parte do tempo possível;
- Confidencialidade dos dados: refere-se à proteção contra acesso não autorizado por parte de usuários maliciosos;
- Privacidade: os dados e informações dos usuários permanecem em sigilo.

Ainda segundo Tanenbaum (2003), os problemas de redes de computadores podem ser divididos em áreas interligadas, tais como:

- Sigilo: manter informações longe de usuários não autorizados;
- Autenticação: processo que determina com quem você está se comunicando antes de revelar informações sigilosas ou transmitir transações comerciais;
- Não - repúdio: trata das assinaturas, com a finalidade de garantir a veracidade das atividades realizadas;
- Controle de integridade: garantia de que o conteúdo que trafega na rede não foi alterado por nenhum usuário mal intencionado.

Baseado nestas informações, Comer (2007) informa que uma grande variedade de tecnologias de segurança foram desenvolvidas, destacando-se as seguintes:

- *Intrusion Detection System – IDS*: sistema que monitora os pacotes e notifica o administrador da rede se uma violação de segurança for detectada, podendo detectar desde uma sobrecarga *SYN* (deixando o computador lento) até a um *scanner* de portas (tentativa de acesso através das portas de protocolo TCP).
- *Pretty Good Privacy – PGP*: sistema criptográfico que os aplicativos podem utilizar para criptografar os dados antes da transmissão.
- *Secure Socket Layer – SSL*: tecnologia que usa codificação para oferecer autenticação e confidencialidade, e é usado em uma conexão *web* para transmissão segura de informações financeiras;

- *IP Security – IPSec*: padrão de segurança usado com datagramas IP que usa técnicas criptográficas e permite ao usuário escolher entre autenticação ou confidencialidade;
- *Remote Authentication Dial-In User Service – RADIUS*: protocolo utilizado para fornecer autenticação, autorização e responsabilidade centralizadas.

Já em ambientes domésticos, os riscos com segurança são menores, mas nem por isso, insignificantes. Com o avanço das tecnologias de banda larga, os computadores em ambientes residenciais permanecem conectados à Internet por um maior período de tempo, e desta forma, mais expostos às pragas virtuais, como vírus, e também ataques gerados por criminosos virtuais, além do acesso à conteúdos indevidos (CECÍLIO, 2000).

Assim como em redes corporativas, as redes domésticas também precisam ter os quesitos de confidencialidade e integridade contemplados, pois a presença de arquivos confidenciais também é grande. O uso de *Access Points,* que é uma solução mais barata de compartilhamento de Internet do que uma rede cabeada, com uma configuração inadequada pode facilitar ainda mais a ação de usuários maliciosos, pois a área de cobertura do sinal pode ir muito além do esperado.

Desta forma, assim como em ambientes empresariais, o ambiente doméstico também precisa de uma atenção especial na sua segurança, para que os arquivos, dados pessoais e o próprio equipamento não sejam violados.

2.2 *Firewall*

Firewall é uma combinação de hardware e software que isola a rede interna de uma empresa da Internet em geral, permitindo que alguns pacotes passem e outros sejam bloqueados, possibilitando que o administrador da rede tenha total acesso sobre o sistema que gerencia.

Um *firewall* possui três objetivos principais, que são: ser imune a penetração, controlar todo o tráfego de entrada e de saída e ser o único caminho de entrada/saída dos dados de uma rede (KUROSE, 2010). Segundo Comer (2007), o termo *firewall* faz uma analogia à barreira física a prova de fogo colocada entre duas estruturas para prevenir que o fogo passe de uma para a outra.

Ao falar de *firewall*, sobre sua funcionalidade e importância, parece ser uma ferramenta muito simples, mas na prática, esta facilidade torna-se inversamente proporcional

ao nível de segurança desejada. Na prática, um *firewall* mal configurado pode gerar um efeito tão indesejado quanto o de sua ausência. Sua presença é importante tanto em ambientes residenciais quanto em ambientes corporativos, evitando danos na rede, que podem ser causados por usuários mal intencionados, ou até mesmo por usuários inexperientes. Caso esteja bem configurado, um *firewall* pode evitar ameaças, tais como vírus, *worms*, ataques de negação de serviço, etc (THOMAS, 2007).

Ainda segundo Comer (2007), os *firewalls* baseiam-se no *Stateful Packet Inspection (SPI)*, com a finalidade de inspeção e filtragem dos pacotes, e Moraes, (2010) classifica como um conjunto de recursos de hardware e *software* destinados a garantir a segurança das redes, tendo como principais funções:

- Estabelecer um perímetro de segurança;
- Separar as redes e controlar os acessos;
- Ser um elemento central de controle e aplicação de políticas de segurança;
- Proteger sistemas vulneráveis na rede;
- Aumentar a privacidade;
- Logar e gerar estatísticas do uso da rede e de acessos indevidos. (MORAES, 2010)

Geralmente, o *firewall* está localizado em um roteador de borda, conectando a rede interna a um ISP (*Internet Service Provider*), e é neste nodo que ocorre a filtragem de pacotes. O filtro pode tomar as suas decisões baseadas em aspectos definidos pelo gerente da rede que podem ser: endereço ip de origem e destino do pacote, tipo de protocolo no campo do datagrama (IP, TCP, UDP, ICMP, etc), porta TCP/UDP de origem ou destino, tipo de mensagem ICMP. Também podem possuir diferentes regras para diferentes tipos de pacotes/interfaces (KUROSE, 2010).

A funcionalidade de um *firewall* pode ser simples, como a implementação de um roteador que aplica um filtro de pacotes, ou complexa como um *gateway*, que combina funções de filtros de pacotes e Proxy na camada de aplicação.

As primeiras arquiteturas de *firewall* isolavam as redes em um nível lógico, mas na atualidade existem sistemas de segurança tais como: Filtro de Pacotes, *Statefull Inspection* e *Circuit Level Gateways* ou *Gateways* de Aplicação.

2.2.1 Filtro de Pacotes:

Atua na verificação dos endereços IP e nas portas TCP/UDP, trabalhando com uma lista de controle de acesso, a qual é verificada antes que o pacote seja encaminhado para a rede interna. Possui vantagens como rapidez, eficiência, transparência e facilidade de compreensão, mas possui desvantagens como a aplicação de muitos testes para verificar suas funcionalidades, sintaxe difícil devido a complexidade das listas de acesso e por dificultar a aplicação das políticas e a análise é feita em um pacote por vez.

2.2.2 *Statefull Inspection:*

Cada pacote é individualmente verificado, de acordo com o pacote anterior ou subseqüente, existindo uma verificação de contexto. Os pacotes são examinados usando informações de dados de comunicações passadas. Ainda podem ser criadas sessões de informações virtuais para manter a inspeção sobre protocolos não orientados a conexão de pacotes que possam ter conteúdos ilegais. Os principais critérios de avaliação são: Lista de acesso (ACL – *Access Control List*), regras de autorização, avaliação do cabeçalho, tamanho do cabeçalho IP, avaliação do status de conexão. É importante salientar que este tipo de *firewall* também deve ser capaz de prover serviços de roteamento.

2.2.3 *Proxy*

Servidor que faz a intermediação da comunicação entre equipamentos da rede interna com a rede externa. Possui vantagens como o balanceamento da carga, recursos de cachê e isolamento total entre as redes. As desvantagens são a lentidão, inflexibilidade e alta configuração. Além dos *proxies* tradicionais, existem também os *proxies* na camada de aplicação, que trabalham com dados complexos na camada de aplicação, detectando tentativas de quebras de segurança, além de possuir os atributos de *Proxy*. Por possuírem esta característica extra, são mais lentos. Seus critérios de avaliação são: autenticação de usuário, tabelas de associação, regras de aplicação, auditoria.

Os principais tipos de *Proxy* são os *proxies* de aplicação (WWW, FTP, TELNET, MAIL, SQL, etc); *proxies* de circuito que estejam em rede (endereços IP e protocolos TCP/UDP); *Proxy* reverso (trabalham de forma reversa, permitindo acesso a recursos internos) e *proxy* de cachê (armazena os *sites* mais usados para reúso).

Além do monitoramento do tráfego entre as redes, um *firewall* pode desempenhar outras funções, que são:

- Análise de conteúdo – bloquear determinadas URLs determinadas pelo conteúdo dos *sites*;
- *Gateway* de VPN – realizar conexões criptografadas e tuneladas, utilizando um protocolo como o IPSEC;
- Autenticação de usuários – possibilitar acesso a algum servidor ou aplicação, com o uso de um servidor de autenticação RADIUS.
- Balanceamento de carga – balancear a carga entre servidores, baseando-se em seus tempos de resposta (MORAES, 2010).

2.2.4 *Iptables*

O *iptables* é um *firewall* que atua em nível de pacotes, tomando decisões baseadas em informações contidas dentro dos pacotes, tais como endereço de destino/origem, porta de destino/origem, conteúdo, estado da conexão, etc. Desta forma, o *iptables* funciona através de comparação, onde as informações contidas nos pacotes são comparadas com uma lista de regras com a finalidade de definir se o pacote pode ou não passar (FERREIRA, 2012).

Ainda segundo Ferreira (2012), o *iptables* é umas das melhores soluções para uma rede de computadores, pois é rápida, estável, eficiente e é de fácil administração, pois suas configurações podem ser realizadas através de *scripts.* Possui as seguintes características:

- Suporte aos protocolos TCP, UDP e ICMP;
- Pode especificar portas de endereço e destino;
- Suporte aos módulos externos, como FTP e IRC;
- Suporta um número ilimitado de regras por *CHAINS* (cadeias);
- Pode se criar regras de proteção contra diversos tipos de ataques;
- Suporte para roteamento de pacotes e redirecionamentos de portas;

- Suporta vários tipos de NAT (*Network Address Translation*), como o SNAT (*Source Network Address Translation*) e DNAT (*Destination Network Address Translation*) e mascaramento;
- Pode priorizar tráfego para determinados tipos de pacotes;
- Tem suporte a IPv6, através do programa 'ip6tables'.

Desta forma, as regras de filtragens são compostas pela seguinte sintaxe:

#iptables [-t **tabela**] [**opção**] [***chain***] [**parâmetros**] -j [**ação**]

As tabelas são os locais utilizados para armazenar as *chains* e as regras com uma determinada característica em comum. Na atualidade, existem três tabelas possíveis de serem usadas. A tabela *MANGLE* é utilizada para alterações especiais nos pacotes e trabalha com as *chains prerouting*, *postrouting*, *input*, *output* e *forward*. A tabela *FILTER* é a tabela padrão, e utiliza as *chains input*, *output* e *forward*.

Caso a tabela não seja informada na regra, as políticas serão acrescentadas na tabela padrão. Já a tabela NAT é usada para dados que geram outra conexão (*masquerading*, *source nat*, *destination nat*, *port forwarding* e *redirect* são alguns exemplos), e possui as *chains prerouting*, *postrouting* e *output* (Filtro de Pacotes com Linux, 2012).

As opções definem o que será feito com a regra, que podem ser inclusão, exclusão, adição, etc. Segue na tabela 2 a descrição das opções disponíveis no *iptables*:

Opção	Descrição da Opção
-P	Define uma política padrão
-A	Adiciona uma nova regra às existentes. Esta tem prioridade sobre –P
-D	Apaga uma regra
-L	Lista as regras existentes de uma determinada tabela
-F	Limpa todas as regras de uma determinada tabela
-I	Insere uma nova regra
-h	Exibe a ajuda
-R	Substitui uma regra
-C	Faz a checagem das regras existentes
-Z	Zera uma regra específica
-N	Cria uma nova *chain* com um nome
-X	Exclui uma regra específica pelo seu nome

Quadro 2: OPÇÕES DO IPTABLES
FONTE: JOHNNY FERREIRA (2012)

As *chains* são os locais onde as regras definidas pelos usuários são armazenadas para a operação do *firewall*. Desta maneira, as *chains* estão relacionadas às tabelas que serão usadas.

Seguem no quadro 3 as *chains* disponíveis para cada tipo de tabela.

Tabela	*Chains* disponíveis para a tabela
MANGLE	*PREROUTING* (quando os pacotes precisam ser modificados antes de ser enviados para o *chain PREROUTING* da tabela nat).
	POSTROUTING (quando os pacotes precisam ser modificados antes de serem enviados para o *chain POSTROUTING* da tabela nat).
	INPUT (quando os pacotes precisam ser modificados antes de serem enviados para o *chain INPUT* da tabela *filter*).
	OUTPUT (quando os pacotes precisam ser modificados antes de serem enviados para o *chain OUTPUT* da tabela nat).
	FORWARD (quando os pacotes precisam ser modificados antes de serem enviados para o *chain FORWARD* da tabela *filter*).
FILTER	*INPUT* (pacotes cujo destino final é a própria máquina *firewall*).
	OUTPUT (pacotes que saem da máquina *firewall*).
	FORWARD (pacote repassado pela máquina *firewall* para uma máquina da rede).
NAT	*PREROUTING* (quando os pacotes entram no *firewall* para sofrer o NAT).
	POSTROUTING (quando os pacotes saem da rede após sofrer o NAT).
	OUTPUT (pacotes gerados na própria máquina e que sofrerão NAT).

Quadro 3: *Chains iptables*
Fonte: Leonardo Damasceno (29/10/2009)

Os parâmetros fazem referência ao local de origem/destino dos dados, ou ao tipo de dado que está trafegando na rede. Seguem, no quadro 4, os parâmetros utilizados para a definição das regras, bem como suas descrições. Ainda é possível utilizar dentro dos parâmetros o atributo [!], que indica uma negação, invertendo a regra.

Parâmetro	Descrição
-p [!]	Define o protocolo que a regra irá tratar.
-s [!]	Define o endereço de origem do pacote que a regra irá tratar.
-d [!]	Define o endereço de origem que a regra irá tratar.
-j [!]	Define uma ação para o pacote.
-i [!]	Define o nome da interface por onde o pacote chegou.
-o [!]	Define o nome da interface por onde o pacote sairá.

Quadro 4: conjunto de parâmetros do *iptables*.
Fonte: Turchetti, (2012)

As ações, que sempre deverão ser acrescidas após o parâmetro "-j", definem o que deverá ser feito com o pacote que se enquadrar em uma das regras definidas pelo administrador da rede, o qual pode ser a aceitação do mesmo, bem como sua exclusão ou redirecionamento. No quadro 5 são descritas as ações que poder ser tomadas utilizando as regras do *iptables*.

Ação	Descrição
ACCEPT	O pacote é aceito pela *chain* e segue em frente.
DROP	O pacote é descartado, sem envio de mensagem ICMP.
REJECT	Semelhante ao *DROP*, porém há envio de mensagem ICMP. Funciona somente para as *chains INPUT*, *OUTPUT* e *FORWARD*.
REDIRECT	Altera o endereço IP de destino do pacote.
LOG	Cria um *log* referente à regra executada.

Quadro 5: Ações *iptables*
Fonte: Turchetti (2012)

Neste capítulo foram abordados os conceitos básicos de segurança em redes de computadores, além de uma breve descrição sobre alguns itens para a elaboração de uma boa política de segurança. Também foi abordado o tema *firewall*, e a utilização do *iptables.*

3 SOLUÇÃO PROPOSTA

Este capítulo irá apresentar a justificativa da elaboração da proposta, bem como a apresentação da ferramenta e a exposição da codificação responsável pela aplicação das regras definidas pelos usuários.

3.1 Apresentação da Proposta

Tendo como base o assunto já exposto, sobre as questões de lixo tecnológico e a segurança em ambientes residenciais, objetivou-se a elaboração de um *firewall* para uso doméstico. O mesmo será acessado através de uma aplicação *web*, onde um usuário sem conhecimentos técnicos em segurança de redes de computadores será capaz de realizar suas próprias configurações, com uma interface intuitiva e amigável. E que ao mesmo tempo faça a utilização de um equipamento com hardware de baixo desempenho, oferecendo uma opção de reutilização para os equipamentos classificados como lixo tecnológico.

3.1.1 Escolha do Sistema Operacional UNIX

Para elaboração desta aplicação foi necessária a pesquisa sobre um sistema operacional, onde o principal quesito para escolha foi o baixo consumo de hardware, mas que ao mesmo tempo seja de fácil utilização. O sistema operacional também precisaria suportar o hardware disponível, possuindo suporte para *drivers* mais antigos, e que fosse estável.

Com base nestes argumentos, foi realizada uma pesquisa, que apontou a distribuição Linux Debian como a mais adequada. Segue um comparativo estatístico que aponta o Debian como distribuição que atende às principais necessidades do projeto.

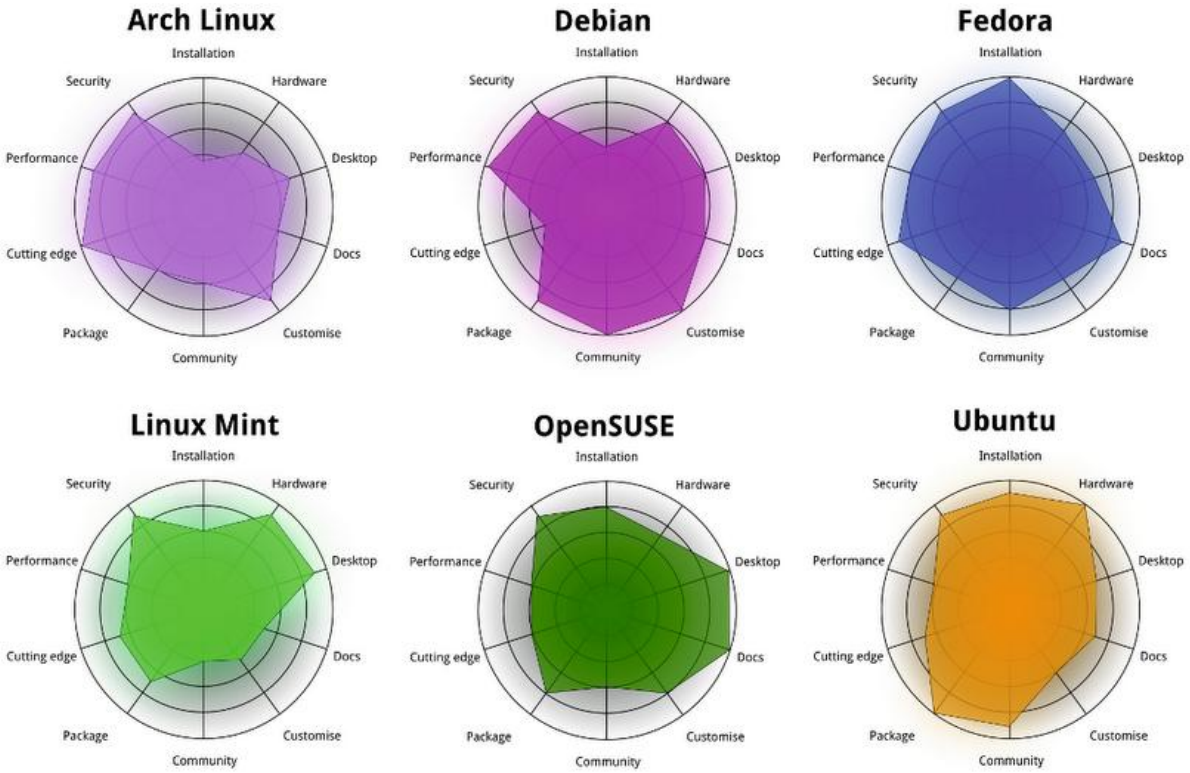

Figura 1: Estatísticas distribuições Linux
Fonte: TuxRadar, 2011

É possível visualizar um estudo em (TuxRadar, 2011), com uma pesquisa abrangente sobre este assunto, onde são demonstrados mais resultados sobre os testes realizados com as distribuições que constam na Figura 1.

O Debian é um sistema operacional livre, elaborado por uma associação de indivíduos, denominada Projeto Debian. É um dos poucos sistemas operacionais livres que ainda suportam equipamentos antigos, pois mantêm em seus arquivos de instalação os *drivers* para este tipo de equipamento (Debian, 2012). O processo de instalação é realizado em ambiente gráfico e de fácil entendimento.

3.1.2 Pesquisa sobre Linguagens de Programação

Depois de instalado e configurado o sistema operacional adequado para aplicação, foi realizado um estudo sobre as linguagens de programação *web* HTML (*Hyper Text Markup Language*) e php (*Personal Hypertext Preprocessor*).

O HTML é uma linguagem de marcação/descrição de páginas que utilizam códigos, denominados TAG's. A codificação deve ser realizada em um editor HTML ou em um editor de texto simples, tais como Notepad (no Windows) ou o Gedit (em ambiente Linux) (Trois, 2012). Mais informações sobre a utilização do HTML podem ser encontradas em Trois (2012).

Outra ferramenta para elaboração do trabalho foi a linguagem php. O php é uma linguagem de *scripts* de uso geral, e é *open source*. É bastante utilizada para o desenvolvimento para aplicações *web* em ambientes HTML. A principal característica da linguagem php é que ela roda diretamente no servidor, e desta forma, o cliente não tem acesso ao seu código fonte. Tem como principais características: velocidade e robustez, orientação à objetos, sintaxe similar a C++ e Pearl (PHP, 2012).

Para que houvesse a perfeita integração entre as chamadas de sistema para a execução dos comandos do *iptables* através da utilização do comando "*system*" da linguagem php, foi necessário a alteração dos arquivos de configuração do arquivo php.ini, presente no diretório /etc/php5/apache2/. Assim, foram configuradas como "ON" as variáveis globais "register_globals", "allow_url_fopen" e "allow_url_include". Também foi adicionado no arquivo /etc/sudoers as linhas "www-data ALL=NOPASSWD: ALL" e "user ALL=(ALL) ALL", com a finalidade de dar permissão de execução dos comandos como super-usuário do sistema.

3.1.3 Utilização de um Banco de Dados

Para que as configurações realizadas pelo usuário possam ser salvas, foi utilizado o MySQL com a ferramenta de banco de dados. O MySQL é um sistema gerenciador de banco de dados (SGBD) que utiliza a linguagem SQL (*Structured Query Language*) e que possui um código aberto, funcionando em praticamente todos os sistemas operacionais. Possui como principais características a portabilidade, compatibilidade com várias linguagens de programação, excelência em desempenho e estabilidade, pouco consumo de *hardware*, além de ser *software* livre (HEUSER, 1998). Para manipular as bases de dados, foi utilizado o programa phpmyadmin, que é uma ferramenta desenvolvida em php para a administração do MySQL através de uma interface *web* simples, que permite a criação, manipulação e gerenciamento de bases de dados. Oferece suporte à maioria dos recursos do MySQL, tais

como a criação e alteração de bases de dados, consulta, inclusão e exclusão de dados, bem como a exportação ou importação de bancos (phpMyAdmin, 2012).

3.1.4 Agendamento de Tarefas

Outro recurso disponível na aplicação desenvolvida é o agendamento de restrições de acesso, permitindo que usuário agende tarefas em uma determinada data. Este recurso é importante, pois possibilita que sejam realizadas alterações na configuração do *firewall* com agendamento prévio, caso o gerente não esteja presente em determinado momento, ou para que não seja necessário realizar alterações diárias na configuração. Por exemplo, o administrador define que nos dias de semana os sites de redes sociais não estarão acessíveis entre às oito horas da manhã e às dezoito horas da tarde, pois este é o horário destinado aos estudos dos filhos. Desta forma, basta configurar no *link* "Agendamento de restrições de acesso" quando e quais sites serão bloqueados, bem como quando estarão liberados.

Para usufruir do recurso de agendamento de tarefas, foi necessária a utilização do utilitário *cron*. O *cron* é uma ferramenta que permite programar a execução de comandos e processos de acordo com as necessidades de horários dos usuários. As regras podem ser editadas no arquivo crontab, presente no diretório /etc/, onde os parâmetros seguem o seguinte formato: [minutos] [horas] [dias do mês] [mês] [dias da semana] [usuário] [comando]. O preenchimento de cada campo deve ser realizado da seguinte maneira:

- Minutos: informar números de 0 a 59;
- Horas: informar números de 0 a 23;
- Dias do mês: informar números de 1 a 31;
- Mês: informar números de 1 a 12;
- Dias da semana: informar números de 0 a 6 (sendo Domingo = 0);
- Usuário: usuário que vai executar o comando (não é necessário especificá-lo se o arquivo do próprio usuário for usado);
- Comando: a tarefa que deve ser executada.

Para armazenar as informações de configurações do usuário, bem como para trazer informações sobre quais alterações já foram feitas no *firewall*, foram criadas quatro bases de dados, as quais foram identificadas como: "agendacron", "iptables", "login" e "regras".

3.1.5 Apresentação do Desenvolvimento da Proposta

Para que seja possível acessar o sistema, os usuários devem inserir na tela de *login*, ilustrada na Figura 2, o seu nome e sua senha, onde a aplicação fará uma busca na base de dados para verificar a veracidade dos parâmetros informados. Caso esta busca confirme a identificação do usuário, ele terá acesso à aplicação.

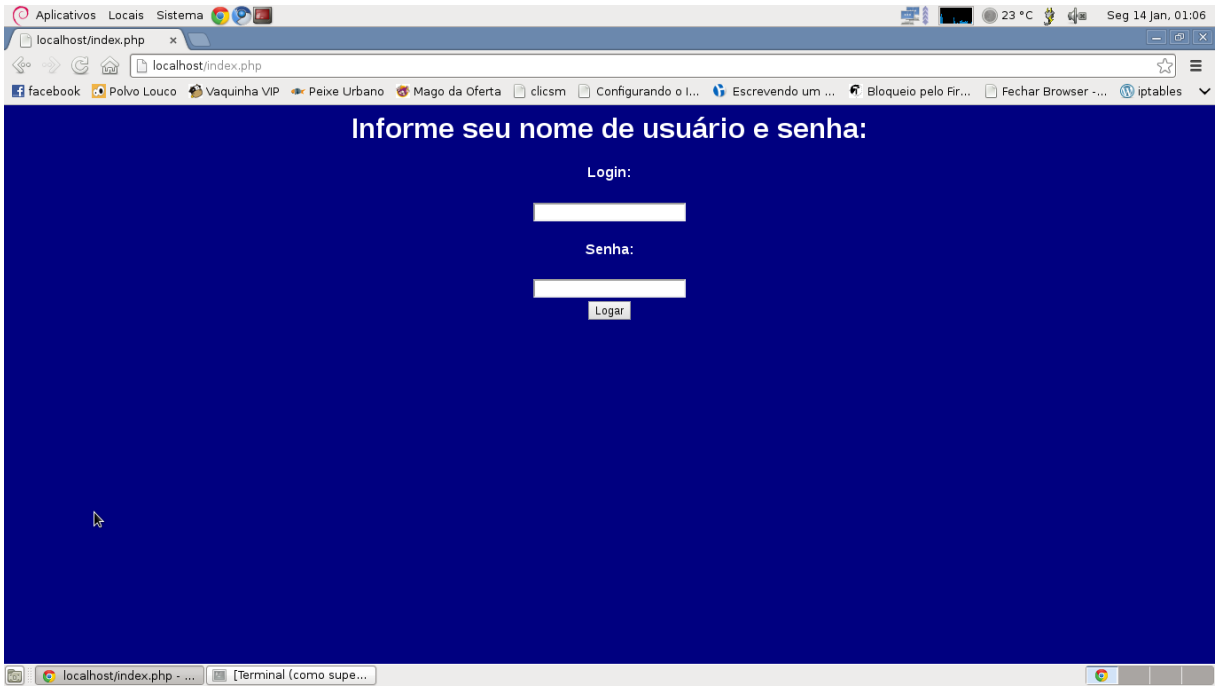

Figura 2: Área de *login* para acesso ao sistema.

Os dados de cadastro dos usuários estão contidos na tabela “login”, e sua estrutura está ilustrada na Figura 3.

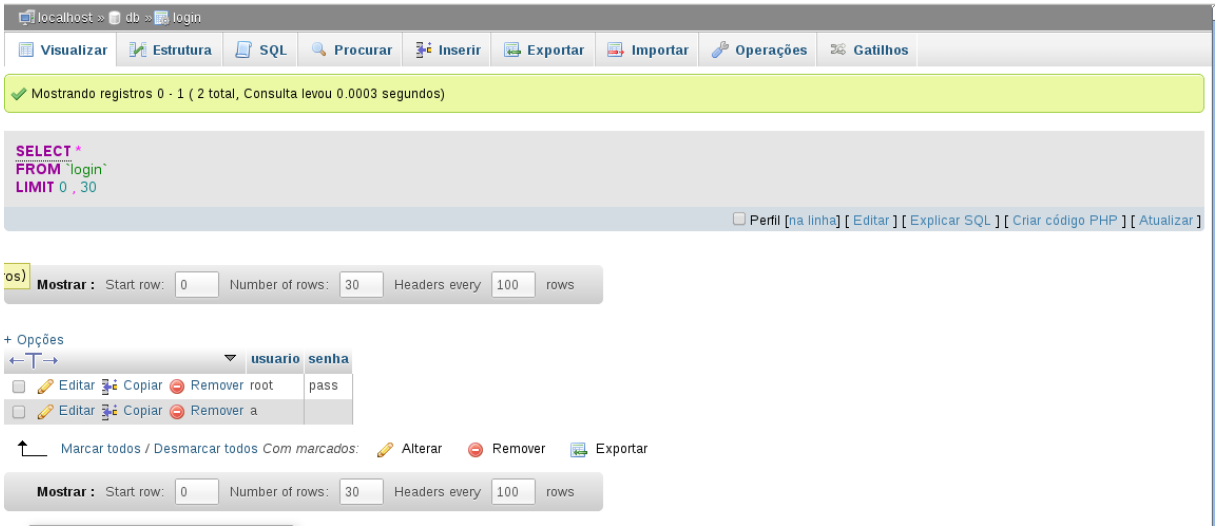

Figura 3: Estrutura da tabela para armazenamento de cadastro de usuários.

Por padrão, o sistema possui cadastrado um usuário administrador denominado “root”, com a senha “pass”. Os administradores podem ainda gerenciar os usuários, incluindo novos, ou excluindo usuários já cadastrados no sistema. O ambiente para gerenciamento de usuários é bastante simples, onde a única exigência do sistema é que seja digitado um nome de usuário e uma senha para efetuar o cadastro de um novo membro. Caso seja informado um usuário já existente na base de dados, ou o *login* for enviado em branco, o aplicativo informará que existe um erro efetuado por quem está realizando o gerenciamento. Não é permitido a criação de usuários sem o cadastro de uma senha.

Para a exclusão de usuários, basta selecionar um registro presente no campo pertinente. A seção de gerenciamento de usuários está ilustrada na Figura 4.

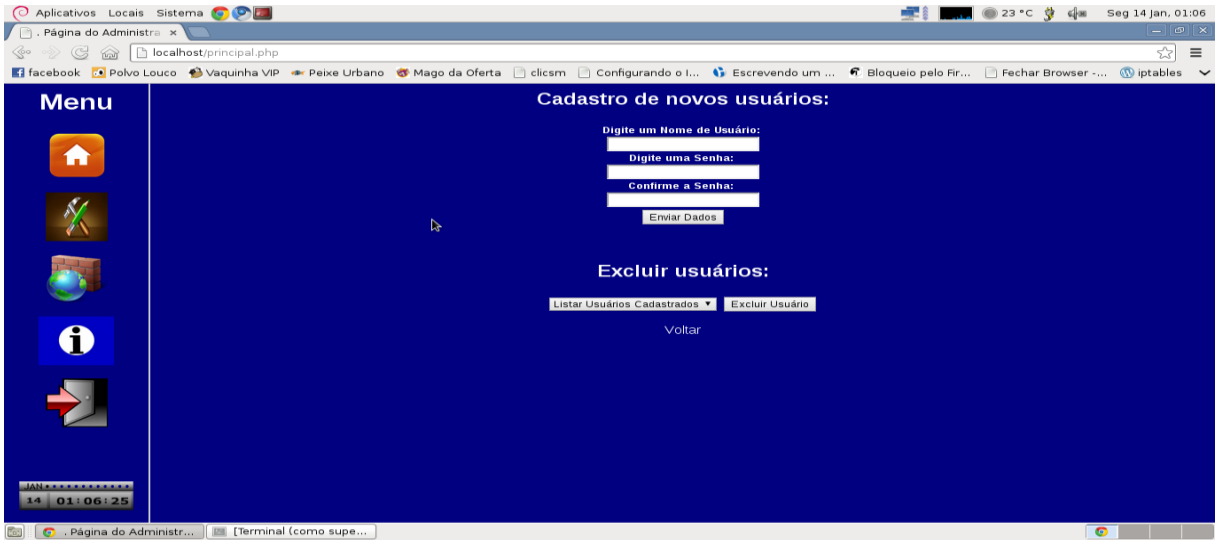

Figura 4: Seção de gerenciamento de usuários do sistema.

Após efetuar o *login* com sucesso, o usuário terá acesso à página principal do sistema, que também é bastante simples. A página possui um menu de navegação no canto esquerdo que possui *links* para as páginas: principal, gerenciamento de usuários, configuração das regras, informações e um botão de *logout*. Estas informações podem ser verificadas na Figura 5.

Figura 5: Página principal do sistema.

Ao acessar a página das regras, é permitido ao usuário fazer o bloqueio de serviços, bem como a liberação dos serviços bloqueados, além do agendamento de tarefas. Também é possível o usuário solicitar informações atuais sobre suas configurações vigentes, clicando no botão responsável pelo serviço (Clique aqui para listar sua configuração atual de *Firewall*).

Estas informações serão retornadas ao usuário de forma amigável, sem linguagens técnicas. Segue na Figura 6 a ilustração sobre o ambiente acima descrito.

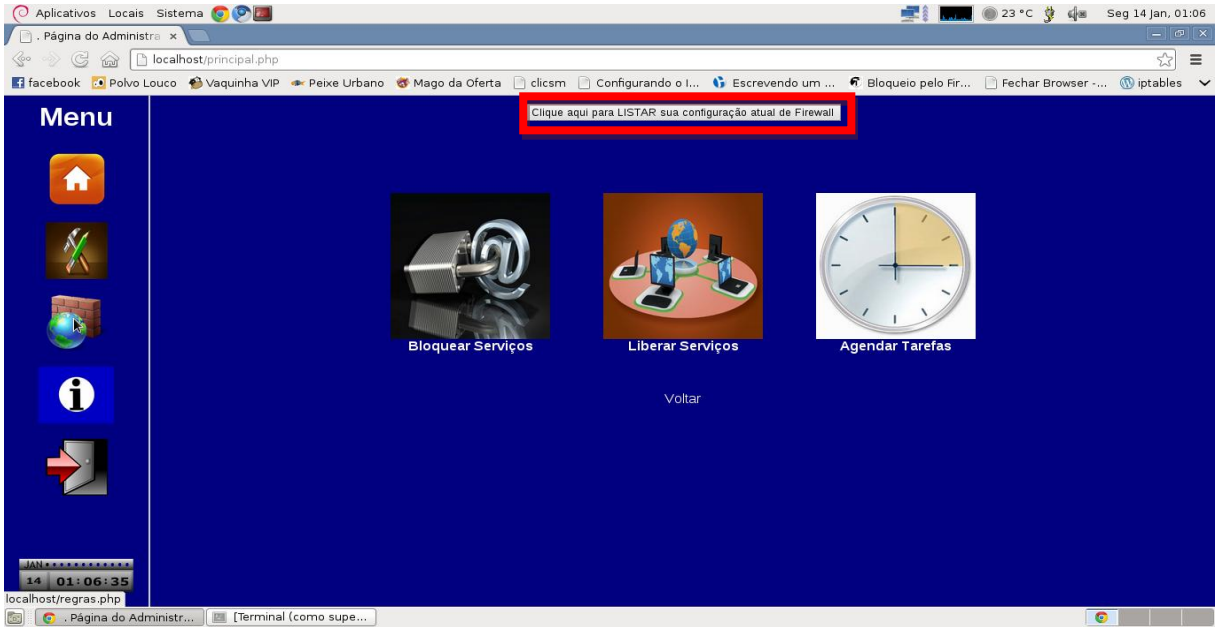

Figura 6: Seção de configuração das regras.

Ao clicar no *link* “Bloquear Serviços”, o usuário dará início à configuração do seu *firewall*. Nesta seção é possível que o usuário faça o bloqueio de sites, bem como de portas específicas na sua rede doméstica.

Partindo do princípio que os usuários não terão conhecimentos sobre o que são as portas de comunicação em redes de computadores, e nem conhecimento sobre qual finalidade de cada uma, foi elaborada uma seção com todas as explicações pertinentes a este assunto. Ainda é possível fazer um bloqueio geral da rede, através de um botão específico, o qual simplifica os comandos “# *iptables* -P *INPUT DROP*” e “# *iptables* -P *FORWARD DROP*”. É possível a familiarização com o acima descrito observando a Figura 7.

Para o bloqueio de portas foram disponibilizadas as mais usuais, conforme avaliação prévia do autor da proposta. É importante salientar que não está sendo considerada a hipótese de os usuários da rede doméstica interna oferecer serviços, sendo apenas usuários comuns.

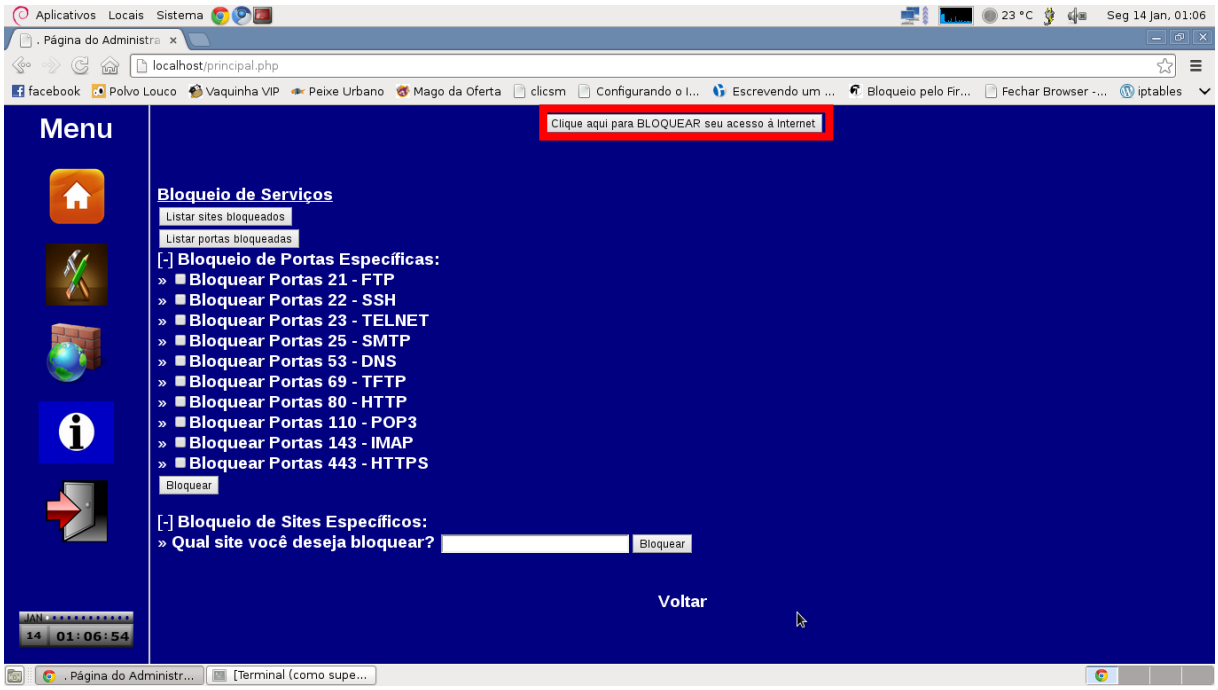

Figura 7: Seção da aplicação responsável pelo bloqueio de serviços.

Para o armazenamento das configurações realizadas pelo usuário, foi criada uma tabela na base de dados para este fim. Esta tabela, que tem sua estrutura ilustrada na Figura 8, foi denominada como "*iptables*". A tabela *iptables* é composta pelas seguintes entidades: "id", "tabela", "ordem", "*chain*", "protocolo", "origem", "destino", "ação", "site" e "portas".

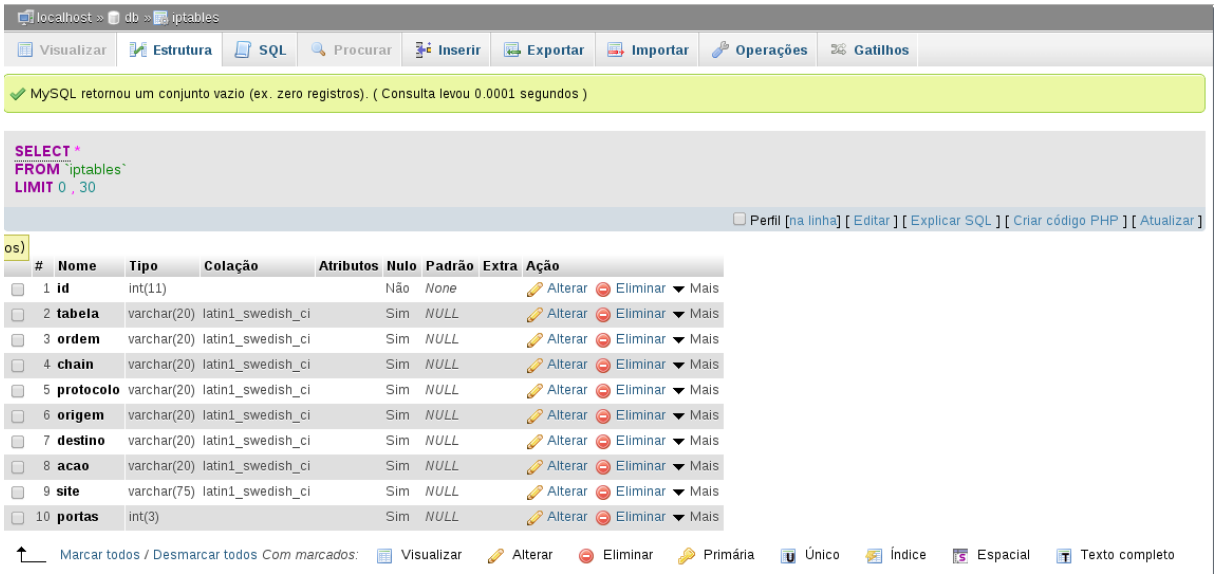

Figura 8: Estrutura da tabela responsável pelo armazenamento das regras do iptables.

A entidade “id” foi gerada com o intuito de criar identificadores para cada tipo de restrição realizada pelo usuário, pois será através da mesma que será realizada a consulta sobre as configurações vigentes do sistema. As entidades “tabela”, “ordem”, “*chain*”, “protocolo”, “origem”, “destino” e “ação” foram geradas exclusivamente para o armazenamento das regras, onde cada informação será armazenada de acordo com seus parâmetros nas regras do *iptables* que serão ativadas através da interface *web*. As entidades “*site*” e “portas” foram criadas com o intuito de facilitar o retorno de pesquisa sobre quais configurações estão vigentes no sistema, bem como informar de forma automatizada quais portas e quais sites estão bloqueados e poderão ser liberados pelo administrador do sistema.

Também de forma simples funciona a sistemática de liberação de serviços. Ao acessar o *link* da página de regras “Liberar Serviços”, o usuário terá informações sobre quais portas e quais *sites* estão bloqueados no momento, e a liberação destes serviços ocorre basicamente em dois cliques: um clique para selecionar o *site*/porta bloqueado, e outro clique para solicitar a liberação.

Ainda é possível que o usuário exclua todas as regras ativas do seu *firewall*, através de um botão específico (Figura 9) para este fim, o qual irá executar os comandos “#*iptables* -F”, “#*iptables* –P *INPUT ACCEPT*” e “#*iptables* –P *FORWARD ACCEPT*”. Da mesma maneira que as regras são incluídas na base de dados ao serem configuradas. Ao selecionar uma regra vigente para ser excluída automaticamente as informações da base de dados referente à regra serão removidas do banco. Todas as ações tomadas pelo gerente do sistema serão imediatamente atualizadas no banco de dados da aplicação. Segue, na Figura 9, a ilustração do ambiente acima descrito.

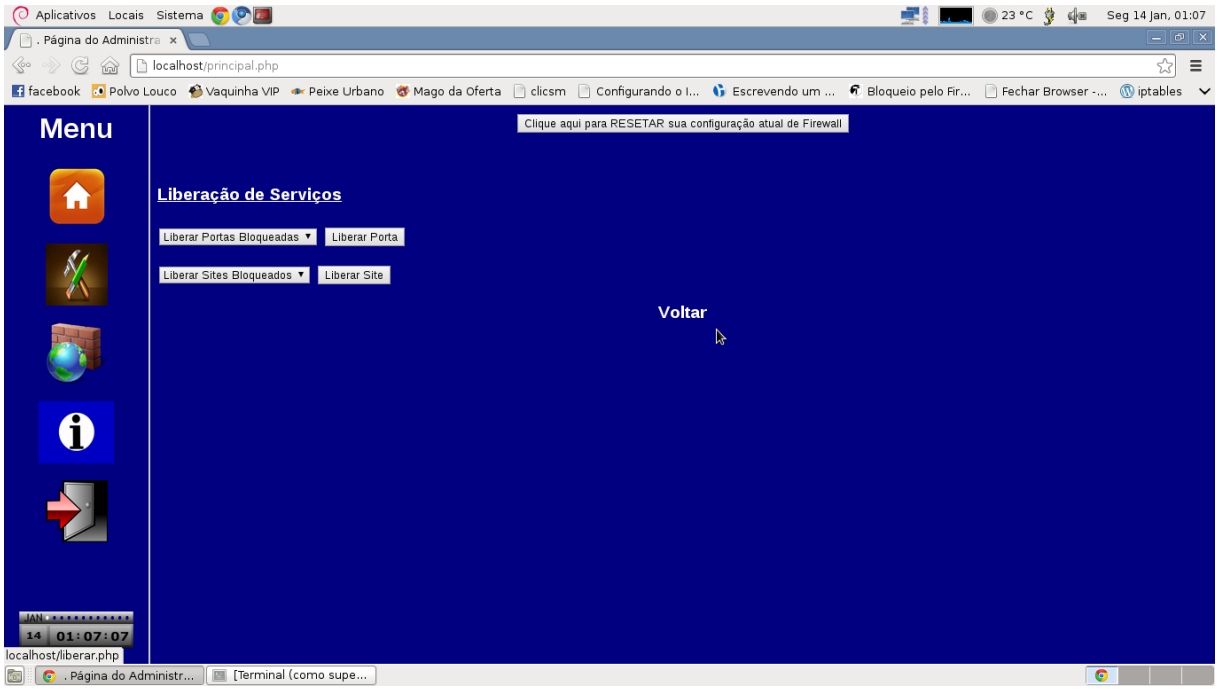

Figura 9: Ilustração da aplicação que permite a liberação de serviços bloqueados.

Ainda no ambiente das regras, é possível que o usuário faça o agendamento de restrição de acesso a sites específicos. Esta área da aplicação foi elaborada com a intenção de que seja possível restringir sites em determinados períodos onde a presença do gerente não seja possível.

A interface apresentada ao usuário também é de fácil entendimento, e em um único ambiente o administrador faz a programação, tanto da hora do agendamento, quanto do momento em que este agendamento deverá ser cancelado. Como pode ser visualizado na Figura 10, é necessário ser informado a hora, o minuto e o dia da semana em que a restrição deverá entrar em vigor, bem com a URL que deverá ser bloqueada. Para que não existam problemas em decorrência do esquecimento do agendamento da liberação do acesso, estas informações também são exigidas no momento de configuração de bloqueio de acesso. Caso o usuário deseje excluir alguma configuração, existe a opção de apagar as regras incluídas através da escolha do site cadastrado. Todas as informações provenientes do agendamento de restrição de acesso serão armazenadas em uma tabela no banco de dados denominada “agendacron”. Esta tabela é composta pelos campos “hora”, “horafim”, “minuto”, “minutofim”, “diasemana”, “diasemanafim”, “site” e “status”, as quais receberão os parâmetros desejados pelo gerente no momento da configuração.

Figura 10: Ambiente configuração de restrição de acesso.

As entidades “hora”, “minuto” e “diasemana” são responsáveis por conter as informações de bloqueio, onde o *cron* irá monitorar em intervalos de cinco minutos a tabela de dados. Quando alguma entrada na tabela corresponder à hora, ao minuto e ao dia da semana atual, a regra será colocada em vigor. Seguindo o mesmo raciocínio, os campos “horafim”, “minutofim” e “diasemanafim” receberão os parâmetros contendo os dados de término da restrição.

Da mesma forma, o *cron* irá monitorar a tabela de dados, e quando os dados se enquadrarem com o período desejado, a regra será desativada. A tabela, que pode ser visualizada na Figura 11, ainda possui os campos “*site*”, que armazenará o endereço que deverá ser bloqueado/desbloqueado, e o campo “status”, que armazenará o status da regra. Caso a regra ainda não esteja em vigor, o campo da tabela estará com a informação “Agendado”.

Quando o agendamento chegar ao seu horário de ativação, o campo será automaticamente atualizado para “ativo”. E por fim, quando o agendamento chegar a seu horário de ser desativado, o acesso ao conteúdo da *web* será liberado.

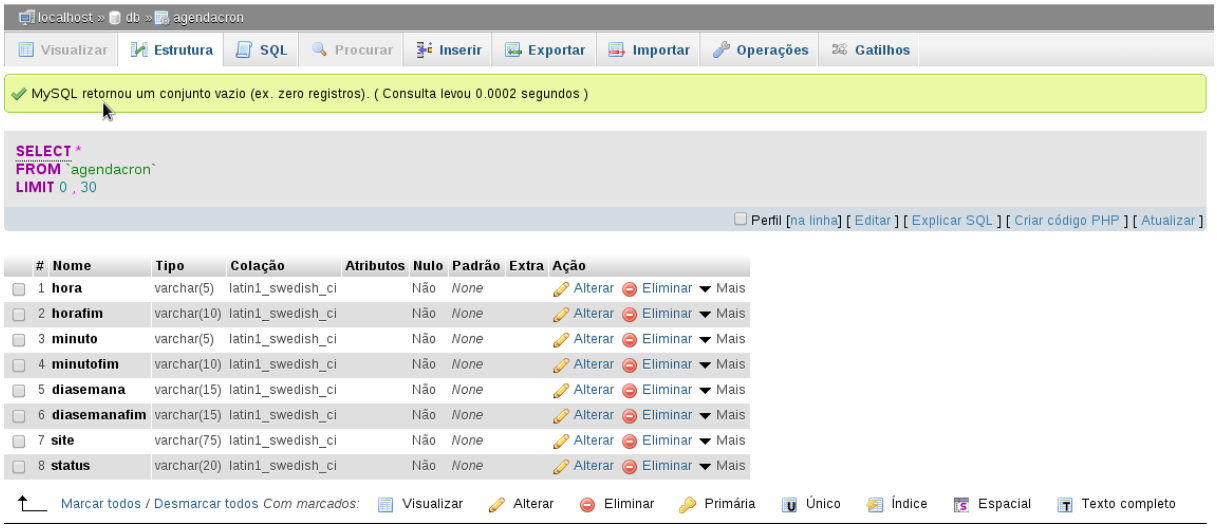

Figura 11: Ambiente configuração de restrição de acesso.

Ainda foi criada a tabela “regras”, com a finalidade de armazenar as informações referentes à quais portas e quais sites foram gerenciados, bem como o status da configuração das regras. Esta tabela recebe os dados juntamente com a tabela “*iptables*” assim que o usuário do sistema definir uma regra. Este mecanismo é útil quando o usuário deseja saber sobre suas informações gerais de configuração, pois esta consulta trará um retorno semelhante ao uso do comando “*#iptables* –L” no terminal do Linux, com a diferença de retornar informações em formato mais amigáveil aos usuários sem conhecimento técnico específico para interpretar a linguagem retornada nas informações. Segue, na Figura 12, a ilustração sobre o ambiente acima descrito.

Figura 12: Estrutura da tabela Regras.

Partindo do princípio que esta aplicação será utilizada por usuários sem conhecimentos específicos em informática, mais precisamente, em segurança de redes de computadores, julgou-se necessário a presença de uma área da aplicação que explique ao gerente da aplicação sobre os conteúdos que a proposta apresentada faz uso. Desta forma, é possível encontrar na seção Informações alguns conceitos, tais como o que é *firewall*, e também o que é o *iptables*, bem como uma listagem contendo as principais portas passíveis de configuração pelo programa, com uma breve descrição de cada uma e contendo como e onde são utilizadas. Este ambiente está sendo ilustrado na Figura 13.

Figura 13: Seção da aplicação responsável pelas informações aos usuários.

Após tomar conhecimento destas seções, o administrador tem total capacidade de gerenciar o seu *firewall*, podendo adequar a sua rede doméstica de acordo com suas necessidades.

Para que este aplicativo funcione corretamente, uma série de comandos foi utilizada, traduzindo a linguagem técnica do *iptables* em simples cliques do *mouse*, e isto foi possível através das linguagens php e HTML. Como exemplos, serão relacionados alguns trechos da codificação utilizada, juntamente com seu comando correspondente no ambiente Linux, bem como com a seção da aplicação responsável por este retorno. As regras foram aplicadas sobre

as *chains INPUT* e *FORWARD* para que não seja possível o acesso interno, como portas, bem como acesso externo, à sites.

Para um gerente de redes, uma atividade importante é obter informações sobre as configurações atuais de suas regras, e esta tarefa é realizada através do comando # *iptables* –L, no terminal do Linux. Para o utilizador da aplicação proposta, basta clicar no botão "Listar Regas", e as informações serão listadas em uma linguagem amigável. Esta interação entre o ambiente gráfico e os comandos utilizados no terminal é possível através da codificação:

```
$var=$_GET["listar"]; //Listar Regras
if($var == "on"){
      $listar = mysql_query("select * from regras WHERE porta is not
null");
      echo "<TABLE>";
      echo "<center><table border=1>";
      echo "<TH><h3>Portas Bloqueadas</TH>";
      echo "<TD><h3>Sites Bloqueados</TD>";
      while($porta=mysql_fetch_array($listar)){
            echo "<tr><th><h4>".$porta['porta']."</th>";
            echo "<td><h4>".$site['site']."</td></tr>";
      }
      echo "</TABLE></center>";
}
```

Ao clicar no botão "Listar Regras", a variável $_GET["listar"] é ativada, fazendo com que o laço seja acionado e seu conteúdo executado.

Caso o usuário deseje limpar todas as regras vigentes, utilizando o terminal do Linux é necessário alguns comandos, tanto para limpar as regras quanto para liberar o acesso nas *chains* correspondentes. Para o usuário da aplicação, basta clicar no botão "Limpar Regras", e esta simples ação será convertida para os comandos:

```
$var=$_GET["liberar"]; //Limpar regras
if($var == "on"){
      system ("sudo iptables -F");
      system ("sudo iptables -P INPUT ACCEPT");
      system ("sudo iptables -P OUTPUT ACCEPT");
      system ("sudo iptables -P FORWARD ACCEPT");
      $sqlEnviando = mysql_query("DELETE FROM iptables") or
die(mysql_error());
}
```

Ao mesmo tempo em que limpa todas as regras vigentes, as informações contidas no banco de dados serão excluídas.

Para bloquear uma porta específica é necessária a utilização de regras que um usuário comum não teria condições de realizar em um ambiente Linux. Utilizando a aplicação, basta selecionar qual porta deverá ser bloqueada, e a aplicação traduzirá para a linguagem técnica pertinente. Como exemplo, será assumido que o usuário desejou bloquear a porta 80, responsável pelas requisições HTTP. É importante salientar que as regras foram elaboradas para permitir requisições locais e da rede interna, pois se não fosse assim, seria impossível manipular a aplicação com esta porta bloqueada. A composição completa dos comandos utilizados está relacionada no Anexo B deste documento. Segue alguns comandos utilizados para esta requisição:

```
system ("sudo iptables -A INPUT -i eth0 -j ACCEPT");
system ("sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED,NEW -i
lo -j ACCEPT");
system ("sudo iptables -A INPUT -p tcp --dport 80 -j DROP");
system ("sudo iptables -A FORWARD -p tcp --dport 80 -j DROP");
```

De forma semelhante, todas as outras configurações de bloqueio de portas foram elaboradas.

Para que a porta seja liberada, o processo realizado é o inverso do processo de bloqueio de portas. As regras são excluídas, tanto das regras vigentes do *iptables*, quanto da base de dados que armazena tais informações. A codificação completa é apresentada no Anexo C deste documento. Segue o laço do código responsável por esta atividade:

```
if ($porta == 80){
      system ("sudo iptables -D INPUT -i eth0 -j ACCEPT");
      system ("sudo iptables -D INPUT -m state -state
RELATED,ESTABLISHED,NEW -i lo -j ACCEPT");
      system ("sudo iptables -D INPUT -p tcp --dport 80 -j DROP");
      system ("sudo iptables -D FORWARD -p tcp --dport 80 -j DROP");
}
else{
      system ("sudo iptables -D INPUT -p tcp --dport $var -j DROP");
      system ("sudo iptables -D FORWARD -p tcp --dport $var -j DROP");
}
```

Além das portas, também podem ser bloqueados sites que o gerente do sistema desejar. Para tal, basta que seja digitada a URL no ambiente correspondente e realizar a requisição. A codificação completa está exibida no Anexo D deste documento. A solicitação será traduzida para os seguintes comandos:

```
system ("sudo iptables -A INPUT -s $var -j DROP");
system ("sudo iptables -A FORWARD -s $var -j DROP");
```

Com uma lógica semelhante, a seção do código responsável pela liberação dos sites bloqueados foi desenvolvida conforme Anexo E deste documento. Os comandos responsáveis pela execução da solicitação estão descritos a seguir:

```
system ("sudo iptables -D INPUT -s $var -j DROP");
system ("sudo iptables -D FORWARD -s $var -j DROP");
```

Para o armazenamento das informações sobre o agendamento de tarefas na base de dados, foram elaboradas duas situações: a primeira onde o usuário deseja agendar uma restrição de acesso, e a segunda, onde o usuário deseja excluir uma regra de agendamento. Ambas as codificações estão relacionadas nos Anexos E e F, respectivamente. Na sequencia, linhas dos códigos responsáveis por tais tarefas.:

```
//agenda bloqueio de sites
$envia1 = mysql_query("INSERT INTO agendacron
(minuto,hora,diasemana,minutofim,horafim,diasemanafim, site, status) VALUES
('$minuto1','$hora1','$diasemana1','$fimminuto1','$fimhora1','$fimdiasemana
1','$agendasite1','Agendado')") or die(mysql_error());

//exclusão de regras agendadas
$sqlEnviando1 = mysql_query("DELETE FROM agendacron WHERE site = '$var'")
or die(mysql_error());
system ("sudo iptables -D INPUT -s $var -j DROP");
system ("sudo iptables -D FORWARD -d $var -j DROP");
```

Para que o agendamento de restrição de acesso seja ativado, a ferramenta *cron* foi configurada executar um *script* em um intervalo de cinco minutos. Este *script* é responsável por verificar se os atributos de hora, minuto e dia da semana se enquadram no período de tempo atual. Caso estas comparações coincidam, o *script* realizará o acionamento ou

desativação da restrição do agendamento. A codificação completa pode ser observana nos Anexos H e I, respectivamente, e na sequencia, a codificação responsável por tais funções:

```
//Início do agendamento da restrição de acesso:
system ("sudo iptables -A INPUT -s $site -j DROP");
system ("sudo iptables -A FORWARD -d $site -j DROP");

//Final do agendamento da restrição de acesso:
system ("sudo iptables -D INPUT -s $site -j DROP");
system ("sudo iptables -D FORWARD -d $site -j DROP");
```

Conforme descrito no capítulo 2, este sistema de *firewall* foi desenvolvido baseado nos conceitos do sistema de segurança Filtro de Pacotes, pois ele trabalha com a verificação das portas de acesso ou nos endereços de destino (no caso, endereços de páginas *web*). Como política de segurança da aplicação foi utilizado o sistema de *login*, onde somente pessoas autorizadas, mediante a utilização de um nome de usuário e senha, podem acessar ao sistema e realizar suas alterações nas configurações.

3.1.6 TESTES

Após a implementação da ferramenta proposta, foram realizados dois tipos de testes com a aplicação. O primeiro foi realizado por formandos do Curso Superior de Tecnologia em Redes de Computadores da Universidade federal de Santa Maria, e teve como finalidade a certificação de que o programa estaria oferecendo, livre de erros, todos os recursos que foram descritos ao longo deste documento. O segundo teste foi realizado em um ambiente residencial, sob responsabilidade de um usuário comum, sem conhecimentos avançados em informática.

3.1.6.1 Ambiente de testes 1

Foi utilizado um *notebook* como servidor da aplicação. O sistema operacional e a versão do SGBD utilizados para este teste foram os mesmos que os utilizados para a instalação do *firewall* no equipamento com hardware de baixo desempenho, e os testes foram realizados nas dependências do Colégio Técnico Industrial de Santa Maria.

O equipamento servidor foi conectado à rede através da interface local eth0, e para compartilhamento de Internet foi criada uma interface virtual eth0:1. Para que houvesse o compartilhamento da rede, foi executado o comando "#echo 1 > /proc/sys/net/ipv4/ip_forward". Já para o mascaramento de IP, foi realizado o comando "*#iptables* –t nat –A *POSTROUTING* –o eth0 –j *MASQUERADE".* A interface eth0 estava configurada com um endereço de IP interno da UFSM, e para a interface eth0:1, foi configurado endereço endereço 10.1.1.1/24. Como clientes, foram utilizados dois computadores de mesa, e um *notebook*. Os computadores de mesa receberam os endereços 10.1.1.5/24 e 10.1.1.6/24, e o *notebook* foi configurado com o endereço 10.1.1.7/24. Também foi necessário adicionar o equipamento servidor da aplicação como *gateway default* dos clientes, onde os mesmos precisaram executar o comando *#route add default gw 10.1.1.1*.

Com a rede descrita funcionando, foram realizados diversos testes de comunicação, tais como SSH, FTP, TELNET, PING e TRACEROUTE (para verificar se os pacotes estavam sendo encaminhados pelo *gateway*). Todos os testes obtiveram êxito, então foi concluído que a comunicação entre todas as máquinas estava funcionando corretamente. Também foi utilizada a ferramenta *Wireshark* para fazer captura dos pacotes e garantir que as requisições estavam sendo enviadas corretamente para o servidor.

3.1.6.2 Ambiente de Testes 2

Depois da realização de todos os testes possíveis por parte dos acadêmicos, a aplicação foi configurada em seu local original de destino, o "lixo tecnológico". Este equipamento, que possui um processador Intel Pentium III, HDD (*Hard Disk Drive)* com capacidade de armazenamento de 20 Gigabytes e uma memória RAM com capacidade de 128 Megabytes foi instalado em um ambiente residencial. Todas as configurações necessárias para o funcionamento do equipamento foram realizadas, e a configuração do *cron* para execução do *script* de agendamento de tarefas.

Da mesma maneira que para o primeiro teste, as configurações de endereço de rede foram realizadas, mas com a diferença de que o equipamento possui duas interfaces físicas de rede, e que o ambiente residencial possuía apenas *notebooks*, com sistema operacional proprietário. O ambiente da ferramenta foi apresentado para o usuário, seguida de uma breve explicação sobre a finalidade da ferramenta. A ferramenta ficou disponível para testes no ambiente residencial por um período de dois dias, e ao término deste prazo, foi obtido um *feedback* (anexo A) sobre a utilização da ferramenta. É importante salientar que não foi exposta ao usuário nenhuma explicação sobre a utilização da ferramenta, mas apenas sobre a sua finalidade.

Este capítulo apresentou a exposição da proposta do trabalho, desde seu planejamento até a sua conclusão. Também foram apresentados os ambientes dos testes realizados.

4 RESULTADOS

Este capítulo apresentará uma análise dos resultados obtidos sobre a elaboração da aplicação, bem como sobre sua utilização por um usuário final.

4.1 Análise dos Resultados

Após o término da elaboração da ferramenta e sua aplicação em um ambiente residencial, foi possível fazer uma análise sobre a execução da proposta, desde sua projeção até a sua conclusão, bem como sobre os resultados obtidos com a atividade. Conforme relatado no capítulo anterior, dois tipos de testes foram realizados para aferir a funcionalidade da aplicação: o teste técnico, realizado por pessoas com conhecimento em informática, com a finalidade de encontrar problemas provenientes da construção do projeto, e também o teste prático, inserindo o sistema final em um ambiente residencial para testar a sua funcionalidade.

Logo após o ambiente para o primeiro teste preparado, deu-se inicio aos testes técnicos da aplicação. Foram testados todos os serviços oferecidos pela aplicação, e encontrado apenas um erro, onde ao bloquear a porta 80, de acesso HTTP, as estações perdiam a conexão com a aplicação. O erro foi identificado, corrigido e novos testes foram realizados, desta vez sem resultar na identificação de qualquer falha no sistema.

Posteriormente ao término dos testes técnicos, o equipamento apropriado para a aplicação - o lixo eletrônico - foi preparado e instalado em um ambiente residencial. Ao final do período de teste, foi aplicado um breve questionário, para análise qualitativa, ao usuário contendo questões que contemplavam a utilização da ferramenta. A pesquisa qualitativa, que é definida como “modalidade de pesquisa na qual os dados são coletados através de interações sociais e analisados subjetivamente pelo pesquisador” (APPOLINÁRIO, 2004, p. 154).

Foi relatado pelo usuário que a familiarização com o sistema ocorreu de forma rápida, pois todas as funcionalidades são apresentadas de maneira simples e clara. Também foi apontado pelo usuário que todos os testes realizados parecem ter ocorrido com sucesso, pois a ferramenta realizava as restrições e agendamentos solicitados, e a liberação dos serviços bloqueados também foi realizada com facilidade e sucesso.

A única dúvida apontada foi referente à utilização do bloqueio das portas. Ao utilizar este serviço, a única diferença percebida pelo usuário foi a ausência de Internet ao bloquear a porta 80. Ao bloquear as demais portas, não foi percebida nenhuma diferença ao navegar pela Internet.

Analisando o acima relatado, é possível afirmar que os objetivos deste trabalho foram atingidos, pois a ferramenta *web* proposta foi desenvolvida, apresentando uma interface simples. Além do mais, os testes sobre a eficiência da ferramenta foi realizado com sucesso, apontando poucos erros, que foram corrigidos tão logo foram identificados. A aplicação da ferramenta em um ambiente residencial também foi realizada, funcionando a contento e realizando todas as requisições solicitadas pelo usuário.

5 CONSIDERAÇÕES FINAIS

As redes de computadores estão se tornando cada vez mais indispensáveis para organizações e para ambientes residenciais, devido a facilidade encontrada em obter informações, localizar pessoas e recursos, com um tempo de resposta baixo. Mas, da mesma maneira que este recurso trás benefícios pode ser de extremo perigo, caso utilizado de maneira errônea.

Através da ferramenta proposta, foi demonstrado que as ações de segurança podem facilmente ser tratadas por usuários que não tem um conhecimento avançado em informática, mas que tem à disposição material que torne esta tarefa possível. Desta forma, fica evidenciado que a proposta de trabalho foi realizada de forma positiva, e que todos os objetivos foram alcançados, desde a implementação do sistema, até o desempenho em um ambiente cabível de utilização.

Para trabalhos futuros, poderá ser analisada a possibilidade de implementar novas funcionalidades ao projeto atual, bem como a utilização de novas tecnologias de hardware, com maior poder de processamento e com proporções menores. Para esta análise, poderá ser realizada uma pesquisa sobre os equipamentos raspberry pi, placas ITX, ou alguma outra placa que se encaixe dentro das necessidades do projeto, bem como desenvolver um serviço que funcione para um domínio maior, com serviços ativos e um sistema mais incrementado de segurança.

6 REFERENCIAS BIBLIOGRÁFICAS

APPOLINÁRIO, Fabio. Dicionário de pesquisa científica. São Paulo: Atlas, 2004.

CECÍLIO, E. L.; **Acesso Residencial em Banda Larga.** 2000. 33f. Tese de Mestrado em Informática. Universidade Federal do Rio de janeiro, Rio de Janeiro, 2000.

COMER, D. E. **Redes de Computadores e a Internet**. 4. Ed. Porto Alegre: Bookman, 2007.

Debian.org. 2012. Disponível em: <http://www.debian.org/intro/about, acessado em 29/12/2012>. Acessado em 07/09/2012.

FERREIRA, J. **Conhecendo o Iptables.** Disponível em: <http://johnnyroot.wordpress.com>. Acessado em 13/12/2012.

FLORES, P. L.; **Lixo Tecnológico.** 2012. Disponível em: http://www-usr.inf.ufsm.br/~pablo/e-lixo>. Acessado em 27/11/2012

KUROSE, J. F.; Ross, K. W. **Redes de Computadores e a Internet:** uma abordagem to*p-d*own. 5. Ed. São Paulo: Addison Wesley, 2010.

DAMASCENO, L. **Segurança com iptables.** 29/10/2019. Disponível em: <http://www.vivaolinux.com.br/artigo/Seguranca-com-iptables-1?pagina=1>. Acessado em 13/11/2012

HEUSER, C. A.; **Projeto de Banco De dados.** 4. Ed. Porto Alegre: Sagra Luzzatto, 1998.

MORAES, A. F. de.; **Redes de Computadores:** fundamentos. 7. Ed. São Paulo: Érica, 2010.

NAKAMURA, E. T.; GEUS, P. L de. **Segurança de redes em ambientes cooperativos**. São Paulo: Novatec Editora, 2007.

PHP: *Personal Hypertext Preprocessor. 2012. Disponível em:* <http://www.php.net>. Acessado em 12/09/2012

PhpMyAdmin. 2012. Disponível em: <http://www.phpmyadmin.net/home_page/index.php>. Acessado em 11/11/2012.

SANGARA, M.S.; **Crime de pedofilia na internet:** falta de punibilidade ou exposição ilimitada das possíveis vítimas**.** 2011. 101f. Monografia (Tecnólogo em Informática para gestão de negócios) - FACULDADE DE TECNOLOGIA DA ZONA LESTE, São Paulo, 2011

TANENBAUM, A. S. **Redes de Computadores** 4. Ed. Rio de Janeiro: Elsevier, 2003.

THOMAS, T. **Segurança de Redes:** Primeiros passos. Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

TURCHETTI, R. 2012. **Filtro de Pacotes com Linux**. Disponível em: <http://ead06.proj.ufsm.br/moodle/mod/resource/view.php?id=35990>. Acessado em 13/11/2012

TROIS, C. 2012. **Tutorial Básico HTML**. Disponível em: <www.inf.ufsm.br/~trois/redes2 01/01/2013>. Acessado em 11/09/2012

TROIS, C. 2012. **Basic HTML Elements:** Quick Reference. Disponível em: <www.inf.ufsm.br/~trois/redes2, acessado em 02/01/2013>. Acessado em 02/09/2012

TuxRadar Linux. **The Best Linux distro of 2011**. Agosto, 2011. Disponível em: <http://tuxradar.com/content/best-distro-2011>. Acessado em 27/10/2012

7 ANEXOS

7.1 Anexo A – Questionário Aplicado ao Usuário que Efetuou Testes na aplicação:

Pergunta 1: Qual sua opinião sobre a relevância deste trabalho?

Resposta 1: Na minha opinião, este trabalho é muito útil. Normalmente a gente não ouve muito sobre o assunto de segurança em computadores, e até nem sabe do que se trata. Também é impossível saber o que as pessoas fazem e o que acessam em seus computadores pessoais. Com a ferramenta, foi possível bloquear o acesso a alguns sites em todos computadores da minha casa, e acho isso bom, principalmente quando existem crianças, pois a gente pode bloquear algum site que não gostaria que elas acessassem.

Pergunta 2: O que você achou do ambiente de navegação do sistema?

Resposta 2: O programa é bem simples, e não é difícil aprender a mexer nele. Foi possível entender rápido o que eu podia fazer.

Pergunta 3: Quais as dificuldades encontradas na utilização do sistema?

Resposta 3: Eu não tive muitas dificuldades, mas não entendi muito bem para que serve aquela parte de bloqueio de portas. Eu sei que quando cliquei em algumas, não consegui mais acessar nada na internet. Até li nas informações que tem no site, mas não entendi muito bem.

Pergunta 4: Qual sua consideração sobre a seção “Bloqueio de Serviços”?

Resposta 4: como já relatei anteriormente, eu não entendi muito bem sobre as portas, mas quando eu mandava bloquear algum site, o programa bloqueava mesmo. Talvez se eu soubesse como testar as portas, poderia dizer também que funcionou bem.

Pergunta 5: Você utilizou a seção “Agendamento de Tarefas”? Se sim, qual sua opinião sobre este serviço?

Resposta 5: Sim, utilizei. Achei esse serviço legal, por que a gente pode programas algum bloqueio, caso a gente não esteja em casa. A gente pode definir uma faixa de horário para que algum site fique inacessível.

Pergunta 6: De modo geral, qual sua opinião sobre esta ferramenta?

Resposta 6: Eu achei a ferramenta legal, foi boa a iniciativa. É fácil de usar.

7.2 Anexo B – Codificação utilizada para elaborar o serviço de bloqueio de portas.

```
$var=$_POST["bloqporta80"]; //bloquear porta
if($var == "on"){
	$analisar = "SELECT * FROM iptables WHERE id = '8'";
	$resultado = mysql_query($analisar);
	$iniciar = mysql_num_rows($resultado);
	if ($iniciar == 6){
		echo "<center>Estas regras j&aacute; foram aplicadas</center>";
	}
	else{
		system ("sudo iptables -A INPUT -i eth0 -j ACCEPT");
		$sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, origem, acao, portas) VALUES ('8','A','INPUT','eth0','ACCEPT','80'))
or die(mysql_error());
		system ("sudo iptables -A INPUT -m state --state
RELATED,ESTABLISHED,NEW -i lo -j ACCEPT");
		$sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, origem, protocolo, acao, portas) VALUES
('8','A','INPUT','local','NEW’, ‘RELATED’, ‘ESTABLHIS.','ACCEPT','80')") or
die(mysql_error());
		$regrasEnviando = mysql_query("INSERT INTO regras (porta,
status) VALUES ('80','bloqueada')") or die(mysql_error());
		system ("sudo iptables -A INPUT -p tcp --dport 80 -j DROP");
```

```
            $sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, protocolo, origem, acao) VALUES ('8','A','INPUT','tcp','80','DROP'))
or die(mysql_error());
            system ("sudo iptables -A FORWARD -p tcp --dport 80 -j DROP");
            $sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, protocolo, destino, acao) VALUES
('8','A','FORWARD','tcp','80','DROP')") or die(mysql_error());
      }
}
```

7.3 Anexo C – Codificação utilizada para elaborar o serviço de liberação de portas.

```
$var=$_POST["libporta"]; //liberar Portas
if($var != ""){
      $analisar = "SELECT * FROM iptables WHERE portas is not null";
      $resultado = mysql_query($analisar);
      $iniciar = mysql_num_rows($resultado);
      if ($iniciar == 0){
            echo "<center><h5>Nenhuma porta est&aacute;
bloqueada!!</h5></center>";
      }
      else{
            $analisar1 = "SELECT * FROM iptables WHERE portas = '$var' and
protocolo is not null";
            $resultado1 = mysql_query($analisar1);
            $iniciar1 = mysql_num_rows($resultado1);
            if ($iniciar1 == 0) {
                  echo "<center><h5>A porta desejada n&atilde;o est&aacute;
bloqueada!!</h5></center>";
            }
            else{
                  if($var==80){
                        system ("sudo iptables -D INPUT -i eth0 -j
ACCEPT");
                        system ("sudo iptables -D INPUT -m state --state
RELATED,ESTABLISHED,NEW -i lo -j ACCEPT");
                        system ("sudo iptables -D INPUT -p tcp --dport 80 -
j DROP");
                        system ("sudo iptables -D FORWARD -p tcp --dport 80
-j DROP");
```

```
                        $sqlEnviando = mysql_query("DELETE FROM iptables
WHERE id = '8'") or die(mysql_error());
                        $regrasEnviando = mysql_query("DELETE FROM regras
where porta = '80'") or die(mysql_error());
                  }
                  else{
                        system ("sudo iptables -D INPUT -p tcp --dport $var
-j DROP");
                        system ("sudo iptables -D FORWARD -p tcp --dport
$var -j DROP");
                        $sqlEnviando = mysql_query("DELETE FROM iptables
WHERE portas = '$var'") or die(mysql_error());
                        $regrasEnviando = mysql_query("DELETE FROM regras
where porta = '$var'") or die(mysql_error());
                  }
            }
      }
}
```

7.4 Anexo D – Codificação utilizada para elaborar o serviço de bloqueio de sites.

```
$var=$_POST["bloqsite"]; //bloquear Sites
if($var != ""){
      $analisar = "SELECT * FROM iptables WHERE id = '12'";
      $resultado = mysql_query($analisar);
      $iniciar = mysql_num_rows($resultado);
      if ($iniciar == 0){
            system ("sudo iptables -A INPUT -s $var -j DROP");
            $sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, origem, acao, site) VALUES ('12','A','INPUT','$var','DROP','$var')")
or die(mysql_error());
            $regrasEnviando = mysql_query("INSERT INTO regras (site,
status) VALUES ('$var','bloqueado')") or die(mysql_error());
            system ("sudo iptables -A FORWARD -s $var -j DROP");
            $sqlEnviando = mysql_query("INSERT INTO iptables (id, ordem,
chain, origem, acao, site) VALUES
('12','A','FORWARD','$var','DROP','$var')") or die(mysql_error());
            echo "<center><h5>Regras aplicadas com sucesso!</h5></center>";
      }
```

```
	else{
		$analisar1 = "SELECT * FROM iptables WHERE origem = '$var'";
		$resultado1 = mysql_query($analisar1);
		$iniciar1 = mysql_num_rows($resultado1);
		if ($iniciar1 == 0) {
			system ("sudo iptables -A INPUT -s $var -j DROP");
			$sqlEnviando = mysql_query("INSERT INTO iptables (id,
ordem, chain, origem, acao, site) VALUES
('12','A','INPUT','$var','DROP','$var')") or die(mysql_error());
			$regrasEnviando = mysql_query("INSERT INTO regras (site,
status) VALUES ('$var','bloqueado')") or die(mysql_error());
			system ("sudo iptables -A FORWARD -s $var -j DROP");
			$sqlEnviando = mysql_query("INSERT INTO iptables (id,
ordem, chain, origem, acao, site) VALUES
('12','A','FORWARD','$var','DROP','$var')") or die(mysql_error());
			echo "<center><h5>Regras aplicadas com
sucesso!!</h5></center>";
		}
		else{
			echo "<center><h5>O site desejado j&aacute; est&aacute;
bloqueado!!</h5></center>";
		}
	}
}
```

7.5 Anexo E – Codificação utilizada para elaborar o serviço de liberação de sites.

```
$var=$_POST["libsite"]; //liberar Sites
if($var != ""){
	$analisar = "SELECT * FROM iptables WHERE id = '12'";
	$resultado = mysql_query($analisar);
	$iniciar = mysql_num_rows($resultado);
	if ($iniciar == 0){
		echo "<center><h5>Nenhum site est&aacute; bloqueado</center>";
	}
	else{
		$analisar1 = "SELECT * FROM iptables WHERE origem = '$var'";
		$resultado1 = mysql_query($analisar1);
		$iniciar1 = mysql_num_rows($resultado1);
```

```
            if ($iniciar1 == 0) {
                  echo "<center><h5>O site desejado n&atilde;o est&aacute;
bloqueado!!</h5></center>";
            }
            else{
                  system ("sudo iptables -D INPUT -s $var -j DROP");
                  system ("sudo iptables -D FORWARD -s $var -j DROP");
                  system ("sudo iptables -D INPUT -s $var -j DROP");
                  system ("sudo iptables -D FORWARD -s $var -j DROP");
                  $sqlEnviando = mysql_query("DELETE FROM iptables WHERE id
= '12' and site = '$var'") or die(mysql_error());
                  $regrasEnviando = mysql_query("DELETE FROM regras where
site = '$var'") or die(mysql_error());
                  echo "<center>Regras aplicadas com sucesso!!</center>";
            }
      }
}
```

7.6 Anexo F – Codificação para o serviço de agendamento de restrição de acesso

```
$var=$_POST["restringeacesso"]; //agenda bloqueio de sites
if (($var != "")&&($delregra=="")){
      $analisar = "SELECT site FROM iptables where site = '$var'";
      $enviasql = mysql_query($analisar);
      $inicia = mysql_num_rows($enviasql);
      if($inicia != "0"){
            echo "<h5><center>Este site j&aacute; est&aacute; bloqueado.
Para agendar uma restri&ccedil;&atilde;o de acesso, v&aacute; at&eacute; a
se&ccedil;&atilde;o \" Liberar Servi&ccedil;os\" e exclua a
restri&ccedil;&atilde;o configurada!";
      }
      else{
            if(($hora=="*") || ($fimhora == "*") || ($minuto=="*") ||
($fimminuto=="*") || ($diasemana=="*") || ($fimdiasemana=="*")){
                  echo "<h5><center>Todos par&acirc;metros de
configura&ccedil;&atilde;o devem ser informados!!</h5></center>";
            }
            else{
```

```
                $envia1 = mysql_query("INSERT INTO agendacron
(minuto,hora,diasemana,minutofim,horafim,diasemanafim, site, status) VALUES
('$minuto1','$hora1','$diasemana1','$fimminuto1','$fimhora1','$fimdiasemana
1','$agendasite1','Agendado')") or die(mysql_error());
                echo "<center>Regras aplicadas com sucesso!!</center>";
            }
      }
}
```

7.7 Anexo G - Codificação para o serviço de exclusão de restrições de acesso.

```
$var=$_POST["delregra"]; //excluir regras agendadas
if($var != ""){
      $analisar1 = "SELECT * FROM agendacron WHERE site = '$var'";
      $resultado1 = mysql_query($analisar1);
      $iniciar1 = mysql_num_rows($resultado1);
      if($iniciar1!=0){
            $sqlEnviando1 = mysql_query("DELETE FROM agendacron WHERE site
= '$var'") or die(mysql_error());
            system ("sudo iptables -D INPUT -s $var -j DROP");
            system ("sudo iptables -D FORWARD -d $var -j DROP");
            echo "<h5><center>Regras aplicadas com sucesso!!</center>";
      }
 }
```

7.8 Anexo H – Codificação desenvolvida para verificação de início de restrição de acesso.

```
if ($iniciarcron != "0"){ //verifica hora, minuto e dia da semana de inicio
de agendamento
      while ($tarefa=mysql_fetch_array($horacron)){
            $hora1 = $tarefa["hora"];
            if($hora1 == date("H")){
                  $minuto1 = $tarefa["minuto"];
                  if($minuto1 == date("i")){
                        $diasemana1 = $tarefa["diasemana"];
                        if($diasemana1 == date("w")){
                              $site1 = $tarefa["site"];
```

```
                                    system ("sudo iptables -A INPUT -s $site1 -j
DROP");
                                    system ("sudo iptables -A FORWARD -d $site1 -j
DROP");
                              }
                        }
                  }
            }
```

7.9 Anexo I – Codificação desenvolvida para verificação de fim de restrição de acesso.

```
if ($iniciarcron1 != "0"){ //verifica hora, minuto e dia da semana de fim
de agendamento
      while ($tarefa1=mysql_fetch_array($horacron1)){
            $hora2 = $tarefa1["horafim"];
            if($hora2 == date("H")){
                  $minuto2 = $tarefa1["minutofim"];
                  if($minuto2 == date("i")){
                        $diasemana2 = $tarefa1["diasemanafim"];
                        if($diasemana2 == date("w")){
                              $site2 = $tarefa1["site"];
                              system ("sudo iptables -D INPUT -s $site2 -j
DROP");
                              system ("sudo iptables -D FORWARD -d $site2 -j
DROP");
                        }
                  }
            }
      }
}
```