Centro Universitário de Brasília
Instituto CEUB de Pesquisa e Desenvolvimento – ICPD

RENAN DE SOUZA RODRIGUES

PROPOSTA DE ELABORAÇÃO DE UMA SOLUÇÃO DE SEGURANÇA PARA REDES DE COMPUTADORES COM USO DE *SOFTWARE* LIVRE

Brasília

2013

RENAN DE SOUZA RODRIGUES

PROPOSTA DE ELABORAÇÃO DE UMA SOLUÇÃO DE SEGURANÇA PARA REDES DE COMPUTADORES COM USO DE *SOFTWARE* LIVRE

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Redes de Computadores com Ênfase em Segurança.

Orientador: Professor Mestre Rafael Sarres

Brasília

2013

RENAN DE SOUZA RODRIGUES

PROPOSTA DE ELABORAÇÃO DE UMA SOLUÇÃO DE SEGURANÇA PARA REDES DE COMPUTADORES COM USO DE *SOFTWARE* LIVRE

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Redes de Computadores com Ênfase em Segurança.

Orientador: Professor Mestre Rafael Sarres

Brasília, ___ de _____ de 2013.

Banca Examinadora

_____

Prof. Dr. Nome completo

_____

Prof. Dr. Nome completo

Dedico este trabalho àquela que

me fez compreender o real

sentido de se trabalhar:

minha eterna amada

Edilene.

AGRADECIMENTOS

Primeiramente, a Deus que nos permite a evolução intelectual e a maravilhosa possibilidade da vida.

À minha amada esposa, meu anjo de luz, Edilene Rossi Lacerda, por aquela noite que me apoiou na decisão de voltar aos estudos, por todas as noites que me apoiou sem me deixar faltar a uma única aula, pelos sonhos em conjunto e também por realizá-los. Esta página não seria escrita sem sua maravilhosa presença em minha vida.

Aos meus amados Luana e Ramon, por todos os carinhos dedicados, pelas noites de saudades suportadas, pelos abraços apertados que nunca faltaram.

À minha querida Mãe, Maria Alexandrina, que abriu as portas de todas as salas de aula de minha vida, em todas me apoiou incondicionalmente e incentivou sem nunca poupar esforços, suor e lágrimas por minha formação moral, por minha vida.

Ao meu inigualável Pai, meu super herói, José Carlos, que tornou este mundo de cabos e fios visível ao meus olhos, que trouxe ao meu alcance a real verdade por trás de um clique, por seu cuidado, por seus exemplos, por minha vida.

Aos meus companheiros de trabalho: Carlos Caribé, Marcelo de Amorin, José Maurício e Alison Barros por todas as calorosas discussões técnicas de elevado nível.

Ao meu orientador, Rafael Sarres, por seu dedicado trabalho, por suas aulas instigantes e orientações precisas.

Aos meus companheiros de curso: Bruno, Ozeti e Leandro por todos os trabalhos executados com primor.

“Talk is **cheap**, show me the **code**.”

Linus Benedict Torvalds, 2006

Resumo

O objetivo de presente trabalho é discutir um método para elaboração de uma solução de segurança para redes de computadores utilizando apenas *softwares* livres. Foi pesquisada uma metodologia para apresentação de muitas funcionalidades, dentre elas: inspeção de pacotes, busca de *strings* interna no pacote, visualização de tráfego em tempo real, acesso VPN, controle de banda, alta disponibilidade, controle de acesso a *sites* e outros. Uma comparação destas funcionalidades com alguns produtos proprietários de mercado foi criada para verificação das suas atuais relevâncias, do seu modo de funcionamento e operação junto a conhecidos equipamentos. Foram comparadas também as formas de licenciamento, os processos de instalação e suporte aos equipamentos, suas documentações e manuseio. Finalmente, ficou demonstrado que, além de possível, é viável a composição de uma confiável solução para segurança da rede de dados com *softwares* livres.

Palavras-chave: *software* livre, segurança, *firewall*.

Abstract

The present project intends to discuss a method for developing a security solution for computer networks only using free software. A methodology was researched for presenting many features, including: deep packet inspection, internal package search for strings, real time traffic display, VPN access, bandwidth control, high availability, access control and other. A comparison of these features with some market proprietary products was created to check its current relevance, its features and operation with a well known equipment. It is also compared the terms of licensing, installation processes and supporting, documentation and handling. Finally, it was shown that, not only possible, but it is viable the composition of a reliable solution for network security data with free software.

Keywords: Open-Source, security, firewall.

Sumário

INTRODUÇÃO....10
1 REFERENCIAL TEÓRICO....13
1.1 Software Livre....13
1.1.1 Desenvolvimento e Comercialização do Software Livre....14
1.1.2 Suporte Técnico e Garantias do Software Livre....15
1.2 Segurança da Informação....17
1.3 Modelo de Camadas TCP/IP....18
1.4 O Protocolo IP....20
1.5 Equipamento de Segurança de Dados: Firewall....21
1.5.1 Tipos de Firewall....22
1.6 Determinação da Rede....23
1.7 Políticas e Normas....25
1.8 Zonas de Segurança....26
1.9 Agregação de Placas de Rede....30
1.10 Virtual LANs....32
1.11 Filtro na Camada de Aplicação....32
1.12 Características Desejadas de um Equipamento de Segurança....33
1.13 Testes de Configuração....36
1.14 Medidas de Desempenho....37
1.15 Principais Softwares Envolvidos....39
1.15.1 GNU/LINUX....39
1.15.2 IPTABLES....40
1.15.3 CONNTRACK-TOOLS....41
1.15.4 TC....41
1.15.5 SQUID-CACHE....42
1.15.6 QLPROXY....42
1.15.7 CLAMAV....44
1.15.8 LOGWATCH....44
1.15.9 NTOP....44
1.15.10 CACTI....44
1.15.11 OPENVPN....45
2 CONDIÇÕES GERAIS DA SOLUÇÃO....46

2.1 Licenciamento dos Softwares Envolvidos....46
2.1.1 Gnu General Public Licence....46
2.2 Documentação....47
2.3 Instalação, Manutenção e Suporte....48
3 IMPLEMENTAÇÃO E TESTES DA SOLUÇÃO....50
3.1 Arquitetura onde o sistema base foi instalado:....50
3.2 Instalação do sistema base CentOS 6.4:....50
3.2.1 Download do sistema:....50
3.2.2 Particionamento:....51
3.2.3 Ajuste dos parâmetros de rede....51
3.2.4 Atualização Inicial do Sistema....54
3.2.5 Adição de Repositórios....54
3.2.6 Instalação dos pacotes que não necessitam compilação:....55
3.2.7 Configuração Squid....55
3.2.8 Instalação e Configuração do QLProxy....57
3.2.9 Instalação e Configuração do C-Icap....60
3.2.10 Instalação e Configuração do SquidClamav....62
3.2.11 Configuração do Clamav....63
3.2.12 Configuração SNMPD....63
3.2.13 Configuração APACHE....63
3.2.14 Configuração MYSQL....64
3.2.15 Configuração do PHP....64
3.2.16 Configuração do CACTI....64
3.2.17 Configuração do NTop....65
3.2.18 Ajustes de e-mail....66
3.2.19 Regras de Firewall....67
3.2.20 Configurações do Traffic Control....71
3.2.21 Configuração do DHCPD....72
3.2.22 Instalação do Conntrackd....73
3.2.23 Ajustes do Conntrackd e KeepAlived....74
3.2.24 Ajuste do SELinux....79
3.2.25 Ajuste de Serviços....80
3.3 Testes....80
3.3.1 Throughput....80

3.3.2 Latência.....82
4 COMPARAÇÃO COM EQUIPAMENTOS PROPRIETÁRIOS DE MERCADO.....84
4.1 Funcionalidades.....85
4.2 Desempenho.....87
4.3 Desvantagens Consideradas.....88
4.4 Outras Propostas de Plataformas de Segurança com Software Livre.....88
5 FUTUROS TRABALHOS.....90
CONCLUSÃO.....91
REFERÊNCIAS.....93

INTRODUÇÃO

Desde uma grande empresa com muitos servidores até os simples pontos de acesso livre, todos devem precaver-se ao usar a *internet*, pois, em ambos os casos, um usuário mal intencionado poderia facilmente tirar vantagem de uma falha na segurança e explorar dados privados.

A área de segurança de redes envolve uma grande responsabilidade por tratar diretamente a informação com a qual uma empresa está trabalhando e, consequentemente, o negócio tratado. Mesmo nos casos em que não há sigilo desta, ainda há necessidade de controle de acesso e respeito de direitos. A segurança da informação deve ser parte integrante de um projeto original e não ser tratada como um adicional, sob o custo de não ter a eficácia desejada (CHESWICK; BELLOVIN; RUBIN, 2005).

Como exemplo, podemos citar o cenário de uma simples rede sem fio de acesso à *internet,* disponibilizada por algum estabelecimento comercial. Esta pode ser usada como caminho para um ataque de maiores proporções em alguma vulnerabilidade descoberta em um sistema de uma grande empresa.

Logo, com o atual crescimento do uso da internet (CASTELLS, 2003), tanto em acessos como em largura de banda disponível na contratação dos circuitos de telecomunicações, não poderemos subestimar a relevância de nossas informações (CHESWICK; BELLOVIN; RUBIN, 2005) nem a relevância de se manter conectado. Com isso, é pressuposto que todo escritório ou estabelecimento tenha alguma responsabilidade sobre o nível de segurança aplicado nessa conexão para que não seja alvo de *softwares* ou pessoas mal intencionadas.

Com esse objetivo é importante criar um processo de Gestão da Segurança da Informação, como sugerido pela NBR ISO/IEC 27001 juntamente com o processo de planejamento, execução, verificação e correção contínua deve ser seguido e mantido dentro das empresas com objetivo de evitar falhas de segurança. A sugestão da norma é aplicável inclusive às pequenas redes de maneira muito prática e rápida desde que o aspecto da segurança das informações seja tratado com relevância e seriedade. E para que tais informações fiquem protegidas, uma

das recomendações é a utilização de equipamentos de segurança na infraestrutura que liga a rede de dados da empresa à *internet.*

A utilização de *softwares* livres para o desenvolvimento de dispositivos de segurança para rede corporativa já é difundido e muito conhecido (HELDENBRAND; CAREY, 2007). Sua flexibilidade de configuração e uso (AROCA; TAVARES; CAURIN, 2007) juntamente com seu licenciamento livre[1] são suas principais vantagens, permitindo o desenvolvimento de dispositivos sob medida para cada caso, de pequenos estabelecimentos até grandes empresas.

Com base nos pontos levantados até aqui, este trabalho vislumbra a apresentação de uma solução de segurança utilizando apenas *softwares* livres e uma comparação de suas principais funcionalidades com soluções e sistemas conhecidos de mercado. Esta solução terá função de *firewall* com diversas interfaces, monitoramento ativo, controle de taxa de transmissão de dados e de conexões ativas, inspeção de pacotes, redundância em vários níveis, acesso VPN, filtro de conteúdo e monitoramento ativo de *hardware*. Algumas funções dos programas implementados serão demonstradas. Porém, este trabalho não visa esgotar as possibilidades e formas de se alcançar os objetivos descritos.

O trabalho está dividido em seis capítulos:

1. Referencial Teórico: descrição dos conceitos utilizados no desenvolvimento do trabalho, bem como os programas auxiliares que compõem a solução proposta, pesquisa sobre implementação de segurança da informação com a utilização de *firewalls*;
2. Condições Gerais: descrição das bases para o desenvolvimento do trabalho, de seu funcionamento e licenciamento, dos suporte e da documentação;
3. Configurações: demostração das configurações dos programas auxiliares e suas funcionalidades;
4. Comparação: comparações entre a solução proposta por este trabalho com demais equipamentos disponíveis no mercado.

1 Disponível em: <http://netfilter.org/licensing.html>. Acesso em: 28 mai. 2012.

5. Trabalhos Futuros: listagem das propostas para futuras implementações e ampliação do presente trabalho contendo áreas ou situações não contempladas devido ao extenso escopo.
6. Conclusão: apresenta os desafios encontrados durante a realização deste trabalho com suas respectivas soluções e sugestões e também fornece as informações adicionais para a continuidade do presente trabalho.

1 REFERENCIAL TEÓRICO

Inicialmente, serão feitas algumas considerações relativas ao ambiente utilizado bem como aos elementos abordados para se alcançar os níveis de segurança desejados. Aqui serão traçados parâmetros para que o ambiente do trabalho possa ser comparado com as demais realidades encontradas por um administrador de redes. Serão consideradas metodologias para testar a performance do produto gerado que poderão ser aplicadas para comparação dos resultados aqui apresentados com as demais soluções de mercado ou em equipamentos de *hardware* diferentes.

Os elementos que compõem este trabalho foram profundamente pesquisados na literatura científica relevante, trazendo em sua composição final o que melhor foi encontrado a época.

1.1 *Software* Livre

De acordo com a definição[2] criada pela *Free Software Foundation*[3], um *software* livre pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. Atualmente, a forma mais comum de um *software* ser distribuído livremente é sendo acompanhado por uma licença de *software* livre (como a GPL ou a BSD), e com a disponibilização do seu código-fonte.

Também definido pela *Free Software Fundation,* existem quatro regras de liberdade que devem ser obedecidas para que um *software* seja considerado livre:

- A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade nº 1). Acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo que você possa ajudar ao

2 Disponível em: <http://www.gnu.org/philosophy/free-sw.pt.html>. Acessado em: 2 fev. 2013.
3 Disponível em: <http://www.fsf.org>. Acessado em: 2 fev. 2013.

seu próximo (liberdade nº 2).

- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

Pode-se notar das regras acima descritas que não há nenhuma referência a custos ou preços. Por este motivo, cobrar ou não pela distribuição ou licença de uso do *software* não está relacionado com o fato deste ser livre ou não. De onde podemos concluir que nada impede que um *software* livre seja copiado e vendido, ou seja, *software* livre não precisa ser gratuito necessariamente. *Software* Livre não significa não-comercial. A forma como será obtido e distribuído o *software* não influencia em sua liberdade e modificar e executar seu código. O desenvolvimento comercial de *software* livre não é incomum; tais *softwares* livres comerciais são muito importantes.

1.1.1 Desenvolvimento e Comercialização do Software Livre

Atualmente, muitas empresas desenvolvedoras de *softwares*, como a *Oracle*[4] e *Red Hat*[5], disponibilizam seus programas como *softwares* livres. Esta vertente, inclusive, está em crescimento (FITZGERALD; 2006) uma vez que estes grandes líderes de mercado encontraram boas formas de negócio mesmo sem o licenciamento fechado de seus produtos.

Ao contrário do que se possa imaginar, o desenvolvimento livre de licenças cresceu evoluindo juntamente com o mercado de produtos licenciados. A linha de pensamento e o ciclo de produção que inicialmente eram baseados em pouco planejamento, alguns rascunhos e grande modularização desenvolvida por muitas pessoas ao mesmo tempo com uma estrutura horizontal foi, aos poucos, redesenhada para uma estrutura mais vertical, planejada e com objetivos bem traçados. Apesar de muitos *softwares* ainda seguirem o formato tradicional, o que não os inviabiliza, muitas evoluções podem ser notadas neste campo.

4 Disponível em http://www.oralce.com. Acessado em: 10 fev. 2013.
5 Disponível em http://www.redhat.com. Acessado em: 10 fev. 2013.

Uma forma de negócio que tem gerado grande lucratividade segue a linha da venda do suporte e do acesso aos repositórios oficiais: o cliente paga por uma licença de suporte onde pode contar com alguns canais de comunicação com a empresa desenvolvedora e também tem acesso às atualizações oficiais homologadas do *softwares* cobertos pela licença. Estes mesmos não tem licença fechada e podem ser copiados entre outros sistemas, entretanto, não terão acesso ao canal de atualização ou ao suporte dos desenvolvedores. Um claro exemplo deste formato de negócio ocorre com a *Red Hat* que vende subscrições de seus sistemas.

Um outro exemplo que também corrobora com essa evolução na forma de desenvolvimento e comercialização do *software* livre é o banco de dados *MySQL*[6]. Este produto livre possui versão gratuita e paga. Como é um banco de grande versatilidade, é largamente instalado em sua versão gratuita, ampliando sua visibilidade e confiabilidade. Assim, para grandes aplicações, muitos administradores preferem evoluir para versão *enterprise,* pois já conhecem o *software* e suas características além de manter a compatibilidade com a versão anterior.

1.1.2 Suporte Técnico e Garantias do Software Livre

Podemos verificar duas linhas principais de suporte técnico e garantidas oferecidas aos *software* livres: uma que não envolve contratos comerciais mas apenas a boa relação entre usuários (LAKHANI, HIPPEL; 2003) e outra que é normatizada pelas regras estabelecidas em contratos firmados entre um cliente e um fornecedor de serviços (FITZGERALD; 2006).

A primeira linha de suporte é oferecida gratuitamente ao usuários baseada em fóruns e listas de discussão de e-mails. Seu principal funcionamento gira em torno de um grupo de usuários avançados e de desenvolvedores, do próprio programa ou não, que se disponibilizam a estudar os casos apresentados pela comunidade e resolvê-los sem cobrar nada por isso. Estas pessoas não são responsáveis diretas pela solução de nenhum caso específico, não existem garantias de que qualquer problema será efetivamente solucionado, nem mesmo

6 Disponível em: http://www.mysql.com. Acessado em: 12 fev. 2013

que o programa em questão terá desenvolvimento continuado. Entretanto, este tipo de atendimento descomprometido é muito comum e funcional dentro da comunidade livre. Tanto os usuários avançados quando os desenvolvedores aproveitam o livre uso de suas criações para testá-las, aprimorá-las e divulgá-las. E se prestam a fazer o atendimento dos casos apresentados por vários motivos:

- grande aprendizado acumulado com o estudo empregado nos casos;
- aproveitamento dos mais variados problemas para ajustar ainda mais seus programas;
- ganho de reputação e respeito frente a comunidade;
- reciprocidade do atendimento, onde ajudando hoje para poder ser ajudado amanhã ou por ter sido ajudado ontem;
- prazer por ajudar em prol de uma filosofia;
- promover o uso do *software* livre.

Apesar de parecer uma forma muito descompromissada de ser oferecer suporte aos usuários, a grande maioria dos *softwares* livres faz uso dessa forma. Como exemplo, podemos citar: Apache Web Server[7], Cacti[8], TORQUE[9], NMIS[10].

A proposta comercial de suporte apareceu mais recentemente juntamente com o crescimento do comércio de *softwares* livres, pois também cresceu a necessidade de treinamento aos usuários finais, suporte técnico especializado, garantias de desenvolvimento e continuidade dos serviços prestados por estes *softwares*. Apesar do método anterior, muitos preferem a contratação de suporte profissional que atribui responsabilidades, deveres e prevê garantia de tempo de atendimento.

Algumas empresas conseguiram perceber este novo e crescente ramo de negócios e formaram alianças com outras empresas para atender esta frente (FITZGERALD; 2006). Esta rede de parceiros engloba desde grupos de

7 Disponível em http://www.apache.org/. Acessado em: 15 fev. 2013.
8 Disponível em http://www.cacti.net. Acessado em: 15 fev. 2013.
9 Disponível em http://www.adaptivecomputing.com/products/open-source/torque/. Acessado em: 17 fev. 2013.
10 Disponível em https://opmantek.com/. Acessado em: 03 mar. 2013.

desenvolvedores, inclusive muitos profissionais começaram a ser **pagos** para criar e manter os programas em que já trabalhavam livremente para comunidade, grupos especializados em *marketing* e propaganda, grupos de vendas e de atendimento ao usuário para suporte técnico. A lógica do funcionamento gira em torno do *software* livre que passa a ter suporte técnico, desenvolvimento e garantias de funcionamento sem deixar de ser livre.

Desta maneira, um usuário é atendido em suas necessidades específicas ao utilizar um *software* livre com grandes possibilidades de personalização mantendo todo o suporte técnico e garantias necessárias. Alguns *softwares* adotam, ainda, uma postura mista: suporte livre e pago. Caso o usuário deseje utilizar o suporte pago, terá as garantias previstas em contrato. Caso contrário, poderá acessar os fóruns e trocar e-mails.

Dentro desta perspectiva, temos este trabalho que traz uma real possibilidade de desenvolvimento de uma ferramenta de segurança. Como é composto apenas por elementos livres, sem garantias ou suportes, caso uma empresa assuma esta responsabilidade (pesquisa contínua para evolução dos conceitos envolvidos, atualização de *softwares* e suporte ao cliente final), poderá vender um serviço utilizando toda a liberdade e possibilidades de ajuste dos *softwares* livres.

1.2 Segurança da Informação

De acordo com a norma NBR ISO/IEC 27002, temos a seguinte definição para o termo **informação**: “é um ativo que, como qualquer outro ativo importante para os negócios, tem um valor para a organização e consequentemente necessita ser adequadamente protegido”.

Este trabalho compartilhará esta definição e importância para toda a informação de uma empresa, sendo ela um poderoso bem que poderá, inclusive, definir posições mercadológicas e estratégicas.

Ainda, de acordo com a norma NBR ISO/IEC 27002, a segurança da informação tem por objetivo garantir a continuidade do negócio, minimizar o risco ao

negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio, visando garantir a integridade, confidencialidade e disponibilidade das informações processadas pela organização.

Estes princípios básicos devem ser sempre respeitados para que se possa garantir a segurança da informação (ABNT NBR ISO/IEC 27002, 2005):

- Confidencialidade: significa que a informação deve ser protegida contra sua divulgação para pessoas não autorizadas – interna ou externamente. Consiste em proteger a informação contra cópias e distribuição não autorizada. Dessa forma, a informação deve ser confidencial e sua utilização deverá ser feita por pessoas previamente autorizadas.
- Integridade: consiste em garantir que a informação gerada não seja modificada sem a devida autorização da(s) pessoa(s) responsável por ela. Isto implica que não deve ser permitido que a informação original sofra nenhum tipo de violação seja ela escrita, alteração de conteúdo, alteração de status, remoção e criação de informações.
- Autenticidade: o controle de autenticidade está ligado ao fato da informação que esteja sendo trafegada seja de fato originada do proprietário a ela relacionado. Não deve ser permitida a violação da origem da informação.
- Disponibilidade: garantir que a informação esteja disponível às pessoas autorizadas sem nenhum tipo de modificação e sempre que elas necessitarem. Pode ser chamado também de continuidade do serviço.

1.3 Modelo de Camadas TCP/IP

A pilha de protocolos TCP/IP é dividida em quatro camadas conceituais, construídas sobre uma quinta camada (COMER, 2000), que corresponde à camada física. Pode-se observar melhor no figura 1.2.1 abaixo:

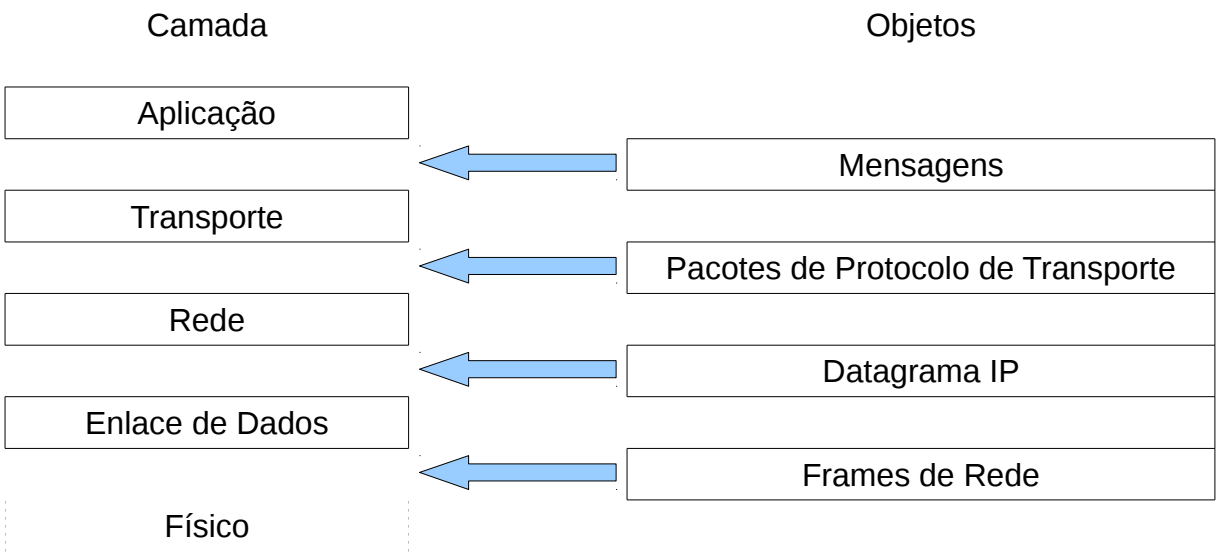

Figura 1.1: Modelo de Camadas TCP/IP

Este trabalho fará referência às camadas estruturadas no modelo TCP/IP baseada neste conceito. Este idealiza uma forma de abstração para os protocolos e serviços da pilha estruturada de dados transmitidos entre os ativos de rede. As camadas são:

1. Acesso a Internet: camada física, compreende as tecnologias usadas para as conexões com a rede; corresponde a camada física e enlace de dados do modelo OSI.
2. Rede: compreende a tecnologia utilizada na rede responsável por sua interconexão, podendo ser IP, MPLS, ATM, Frame-Relay e outros, corresponde a camada de mesmo nome do modelo OSI.
3. Transporte: camada que controla a comunicação entre os ativos da rede, podendo ser TCP, UDP, SCTP e outros, corresponde a camada de mesmo nome do modelo OSI.
4. Aplicação: camada que compreende os protocolos para um serviço específico de comunicação de dados, por exemplo HTTP, DNS, FTP e outros, corresponde às camadas de sessão, apresentação e aplicação do modelo OSI.

1.4 O Protocolo IP

A base para muitos protocolos da pilha TCP/IP é o protocolo IP, *Internet Protocol* (POSTEL, 2002). Este protocolo, através do datagrama IP, provê o transporte para os demais protocolos da pilha.

Pode-se definir o datagrama IP como um bloco de conteúdo com valores de origem e destino bem determinados escritos na forma de *hosts* identificados por endereços lógicos de tamanho fixo. Podemos notar dentro deste bloco, um campo destinado a identificação do fragmento e do seu tamanho permitindo, assim, que grandes volumes possam ser trafegados em pacotes de tamanho determinado. Mesmo quanto há uma rede com um valor diferente de tamanho de pacote, é possível remontar e fragmentar novamente os pacotes, permitindo a comunicação. Desta forma, o protocolo é totalmente preparado para permitir a conexão com origem destino determinados.

Por esta razão, o protocolo também é denominado como serviço de entrega de datagramas não orientado à conexão, pois não contém mecanismos de controle de fluxo, sequenciamento, aumento de confiabilidade. Estas funções normalmente são de responsabilidade dos protocolos superiores.

É também função do protocolo o roteamento do pacote, ou seja, se ele deve ser entregue diretamente ao destino final ou ao *gateway* da rede para que possa ser roteado para o destino final.

Podemos observar uma representação do datagrama na figura 1.3.1:

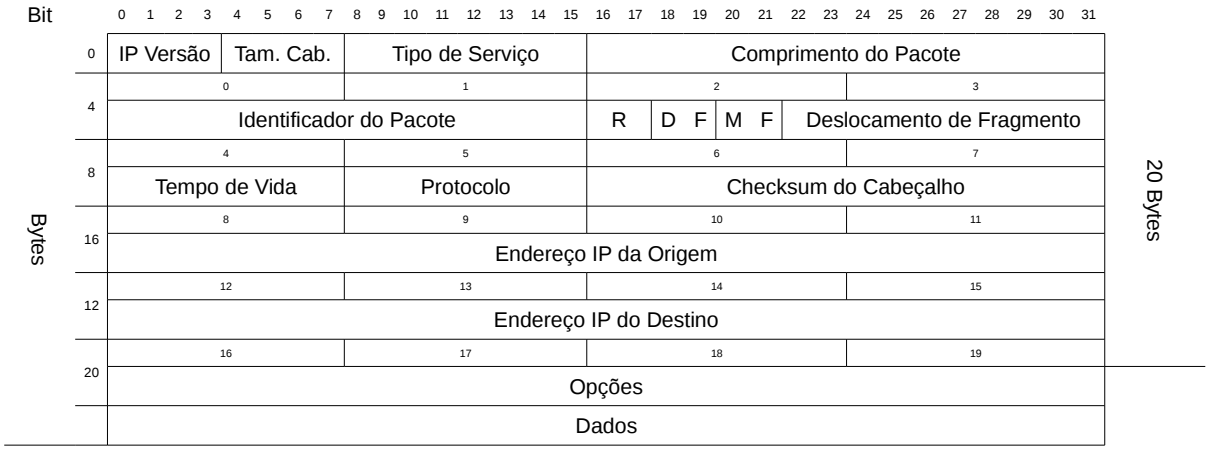

Figura 1.2 – Datagrama IP adaptado, com os campos do cabeçalho IP(POSTEL, 2002)

O campo versão indica o formato do datagrama, para este caso corresponde ao IPv4. O tamanho do cabeçalho pode incluir algumas opções, caso estas ocorram. O tipo de serviço contem parâmetros relacionados a qualidade de serviço desejada, embora a maioria das implementações TCP/IP não suportem esta característica. O comprimento do pacote informa o tamanho total do datagrama, incluindo o cabeçalho IP e os dados propriamente ditos. A identificação do pacote deve identificar univocamente cada datagrama enviado por um host. Os campos R, DF e MF, são argumentos do datagrama, sendo R o bit reservado e que normalmente tem o valor 0, DF o bit responsável por dizer que o datagrama não deve ser fragmentado, e MF o bit responsável por informar que o datagrama está fragmentado e existem mais fragmentos. Descolamento do fragmento indica o posicionamento do fragmento em um dado datagrama a ser remontado. O tempo de vida contém um contador que indica o máximo de roteadores pelos quais um datagrama pode passar. O campo protocolo diz qual é o protocolo utilizado pelo próximo nível, ou seja, pela camada de transporte. E finalmente, os campos de endereço IP de origem de destino contém os endereços lógicos dos *hosts* de origem e destino, respectivamente. O campo de dados, traz a informação líquida do pacote.

1.5 Equipamento de Segurança de Dados: *Firewall*

Com o grande uso da rede mundial de computadores, *internet*, para as mais variadas funções dentro de uma empresa, desde simples buscas de referências até grandes negociações bancárias por meio de conferências e troca de mensagens eletrônicas, a segurança e confiabilidade na infraestrutura que suporta todas essas novas funções agregadas passa a ser encarada como um novo ramo dentro da mesma empresa.

Este novo ramo controla (BERGERON, 1996) a segurança dos dados e informações com que a empresa trabalha, fazendo uso de uma equipe de pessoas e de uma série de equipamentos próprios para que sejam garantidas a confiabilidade, a integridade e a disponibilidade dos dados.

Um desdes equipamentos é conhecido como *firewall,* que atua ativamente na comunicação dos dados entre duas redes de dados distintas permitindo ou não a

passagem de pacotes entre elas. O controle implementado pelo *firewall* é tão bom quanto suas regras são capazes de especificar e tratar precisamente um determinado tráfego de dados. Durante este tratamento, não será permitido perda completa ou parcial de sua capacidade de executar demais filtros ou funções atribuídas. Ou seja, filtrar um determinado fluxo de dados não pode incapacitar o equipamento de executar um roteamento, ou de filtrar outros fluxos de dados, por exemplo (CHESWICK; BELLOVIN; RUBIN, 2005). Assim, um bom *firewall* deve conseguir trabalhar todas as suas capacidades sem, com isso, se prejudicar ou se degradar a ponto de impedir suas funções.

1.5.1 Tipos de *Firewall*

Os *firewalls* são, principalmente, classificados como: filtro de pacotes, filtros de *gateways* de rede e filtros de aplicação *(CHESWICK; BELLOVIN; RUBIN, 2005)*. Estes tipos são assim divididos por atuarem em camadas diferentes do pacote de rede. Entretanto, nos equipamentos mais atuais, existe uma combinação dos três tipos de *firewalls* procurando uma maior proteção aos serviços de acesso disponíveis pela *internet.*

Os filtros de *gateways* de rede funcionam permitindo ou não blocos de rede de se comunicarem entre si. Desta forma, uma rede não muito segura pode ser isolada e não se comunicar com outra onde se deseja garantir maior nível de proteção. São implementados, tipicamente, nos roteadores da rede.

Filtros de pacotes trabalham, essencialmente, descartando pacotes baseados em regras de endereçamento de origem e destino de *host,* protocolo de transporte, opções de cabeçalhos de transporte, porta de origem ou destino, interface onde o pacote foi recebido ou enviado. Inicialmente, estes filtros não eram capazes de executar grandes serviços, pela sua limitação de *hardware* e, ainda, pelo pouco desenvolvimento da *internet* (INGHAM, FORREST; 2002). Não havia suporte ao estado da conexão, nem a tradução de endereçamento, nem suporte para protocolos com portas dinâmicas, como o FTP. Além dessas limitações, apesar de rápida, a escrita das regras era pouco eficiente. Também não era possível qualquer tipo de autenticação de usuário para controle da rede, uma vez que apenas o

endereço IP era visualizado pelos filtros e este poder ser falsificado.

Com o desenvolvimento dos filtros de pacotes, os *firewalls* se tornaram capazes de guardar o estado das conexões, armazenando tabelas com as informações que possibilitaram o rastreio de tudo que foi estabelecido pelos filtros. *IP Filter (IPF)* foi o primeiro a trazer esta funcionalidade[11]. Com esta evolução, também o suporte a NAT foi incluído aos filtros de pacotes.

Ao contrário dos demais tipos de filtro que não analisam os dados do pacote trafegado referente a aplicação chamada, os filtros de aplicação atuam diretamente nestes dados e, por esta razão, são específicos para cada aplicação.

Um exemplo de filtro de aplicação implementada neste trabalho é o *proxy* (INGHAM, FORREST; 2002). Este tipo de filtro funciona partindo o acesso em dois passos: um cliente ao tentar acessar algum endereço remoto tem sua tentativa interceptada pelo equipamento de segurança que analisa a requisição, julgando-a permitida ou não, e, caso seja permitida, a reescreve para que o cliente se conecte a ele e não ao destino remoto. O próprio equipamento irá se conectar ao destino, trazendo toda a informação requisitada, verificando se esta é apropriada, e entregando ao cliente que fez a requisição inicial.

Existem também outras formas menos comuns de implementação de *firewalls:*

- *Firewall* Distribuído: várias máquinas executam várias regras, cada uma ajustada ao próprio caso;
- *Firewalls* Dinâmicos: as regras de segurança podem mudar dependendo do tipo de tráfego que atravessa o equipamento;
- *Firewalls* Normalizadores: o equipamento identifica qualquer ambiguidade nos pacotes, origem ou destinos confusos que não são confiáveis, ou desordem nos fragmentos dos pacotes, filtrando esses casos.

1.6 Determinação da Rede

De acordo com as recomendações da NBR ISO 27001-2006, um estudo

11 Disponível em: http://coombs.anu.edu.au/~avalon/ip-filter.html. Acessado dia 10 fe. 2013.

quantitativo da rede física de dados deverá compor o projeto de desenvolvimento de uma ferramenta de segurança para avaliação do desempenho e capacidade que serão necessários para que o serviço de segurança oferecido atinja os níveis estipulados, inclusive de expansão, evitando qualquer tipo de sobrecarga e consequente perda de produtividade. Assim, para caracterização física do equipamento segundo a NBR, os seguintes pontos principais devem ser estudados:

- quantidade de sub-redes existentes e planejadas;
- quantidade de computadores existentes e planejados;
- quantidade de servidores existentes e planejados;
- quantidade média de conexões estabelecidas por estação de trabalho;
- quantidade de dados média trafegado por estação;
- quantidade de dados média trafegado por servidor;
- expectativa de crescimento da rede de dados e de seus serviços.

Baseado no levantamento de informações acima listado, as características do equipamento pretendido poderão ser norteadas:

- número de interfaces necessárias e disponíveis;
- suas taxas de transmissão exigidas;
- níveis de redundância necessárias;
- velocidade de processamento e taxas de escrita em disco;
- formas e opções de trabalho do equipamento;

Também é necessário um estudo da rede lógica de dados, ou seja, uma determinação das características dos *softwares* instalados no equipamento para que sejam atendidas todas as necessidades pontuadas pela pesquisa:

- quantidade de conexões simultâneas estabelecidas para a *internet;*
- quantidade de conexões simultâneas estabelecidas para a rede interna;

- sistemas operacionais das estações de trabalho;
- existe, ou é planejado, conexão remota entre filiais;
- existe, ou é planejado, conexão remota de clientes e/ou funcionários;
- tolerância de tempo de espera para recuperação;
- o nível de monitoração desejado (tempo real, e-mail, alertas, gráficos);
- eleição dos responsáveis.

Ainda, de acordo com as recomendações da NBR ISO 27001-2006, é necessário refazer essa pesquisa a cada mudança na rede e também periodicamente para garantir que não serão aplicadas mudanças maiores que o equipamento possa suportar, tomando sempre medidas preventiva e controladas. O estabelecimento de boas diretrizes será a garantia de um desenvolvimento preciso e seguro. Idealizado como o ponto central da rede, neste equipamento é muito importante que seja estabelecido um ciclo de verificação contínua, garantindo seu bom funcionamento. Este ciclo é descrito pela metodologia do PDCA - *Plan, Do, Check, Act*. Mesmo em pequenas redes, é importante o desenvolvimento deste processo a fim da melhoria contínua dos negócios.

1.7 Políticas e Normas

Após o estabelecimento do desempenho e das funções do novo equipamento, de acordo com as normas das NBR ISO 27001-2006 e NBR ISO/IEC 17799, uma Política de Segurança deve ser desenvolvida determinando vários pontos (MARTINS, SANTOS, 2005):

- formas de comunicação entre as redes internas e a *internet;*
- formas de comunicação entre as rede internas;
- protocolos e portas permitidas e não permitidas;
- critério temporal para estabelecimento das regras;

- estabelecimento de controle de taxas de transmissão;
- prioridade sobre as transmissões;
- limites de taxa de utilização para atualização dos *links;*
- controle sobre uso de e-*mail;*
- registro de acesso aos sistemas/*internet;*
- uso de antivírus no equipamento;
- uso de antivírus local nas estações;
- rotinas de *backup* de dados e configurações;
- controle de acesso físico ao equipamento de segurança;
- local específico de instalação do equipamento de segurança.

Estes, e possivelmente outros pontos que deverão ser estudados pelo administrador, devem ser verificados e descritos para que haja um documento regulamentar sobre os dados da empresa e seus processos, priorizando a organização e racionalização do uso dos recursos de rede e segurança. Como resultado desse levantamento, algumas funções poderão ser necessárias para o desenvolvimento do equipamento.

1.8 Zonas de Segurança

O funcionamento do equipamento proposto está principalmente baseado em filtros que compõem suas regras de permissão ou negação de tráfego. Estes serão configurados de acordo com as intenções da rede onde será implantado.

Em uma rede de dados, mesmo nas pequenas, existem algumas sub-redes que são criadas a fim de se estruturar um segmentação lógica de dados compatíveis com a segmentação física existente dentro de uma empresa. Seja por hierarquia, pela composição de diferentes áreas de atuação ou por qualquer outro motivo eleito pelo administrador, é comum que tenhamos divisões na rede e, consequentemente, diferentes interfaces no equipamento que a controlará.

Após levantamento das características da rede (KENYON, 2005), podemos citar as seguintes divisões ou zonas dentro de uma determinada instituição:

- *Internet* (abriga a interface de rede que se comunica diretamente com a internet, também conhecida como sub-rede *Untrust*)
- DMZ-Interna (abriga os computadores servidores que não são acessados pela internet, apenas pelos computadores clientes internos)
- DMZ-Externa (abriga os computadores servidores que são acessados pela internet e também pelos computadores clientes internos)
- Redes-Internas-com-Fio (abriga todos os computadores clientes desktops e pode comportar várias sub-redes)
- Redes-Internas-Sem-Fio (abriga todos os computadores clientes com placas de rede sem fio e pode comportar várias sub-redes)

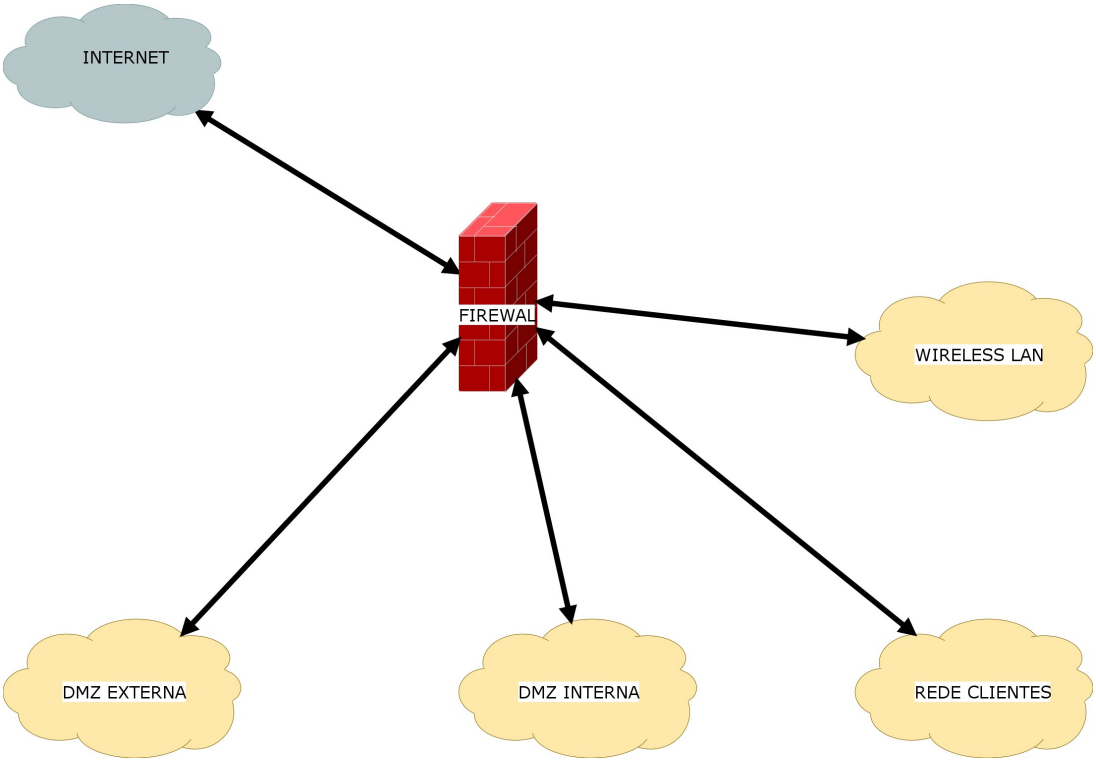

Figura 1.3 : Exemplo de topologia de rede

Dentro da lista de sub-redes acima podemos pensar desde uma pequena rede com acesso apenas para internet compartilhada pela rede sem fio, onde não

seriam utilizadas DMZ's e o equipamento seria mais simples, até uma grande empresa com grande parque de servidores e vasto número de funcionários com diversas redes internas cabeadas.

Para o completo ajuste do equipamento de segurança, muitas regras serão produzidas e inseridas em suas configurações. Para que o equipamento julgue se é ou não permitido um determinado pacote trafegar por suas interfaces, este deve procurar em sua base de regras alguma na qual o pacote se enquadre. Normalmente, essa busca é sequencial e linear, ou seja, se a regra para o específico caso estiver no início da base, pouco esforço computacional é desprendido para encontrá-la. Mas, caso seja uma regra que se encontra no final de suas configurações o esforço computacional para julgá-la será tão maior quanto forem suas bases (FULP, 2005).

Logo, quanto maior a base de regras mais complexa sua administração, mais consumo de recursos e maiores as chances de se cometer erros decorrentes de uma má configuração ou sobrecarregar o processamento. Por sua gravidade e consequências, os erros de configuração em equipamentos de segurança já são tema de estudos por especialistas (WOLL, 2004).

Para facilitar o ajuste e análise das configurações desenvolvidas pelo administrador, bem como diminuir o esforço computacional do equipamento, uma técnica de configuração de **zonas de segurança** foi desenvolvida para que, além dos parâmetros comuns de origem e destino do pacote, seja também considerada a direção de entrada e saída do pacote dentro do equipamento.

Essa nova forma de julgar o tráfego, além das vantagens já descritas acima, também aumenta o nível de segurança que o equipamento pode disponibilizar à rede, pois permite implementações mais efetivas de segurança (WOLL, 2004) como a escrita de regras que se aplicam explicitamente aos pacotes que egressam de uma interface, controle e organização exclusivas sobre as regras aplicadas às sub-redes contidas dentro das zonas, proteção contra ataques to tipo *IP-Spoofing*.

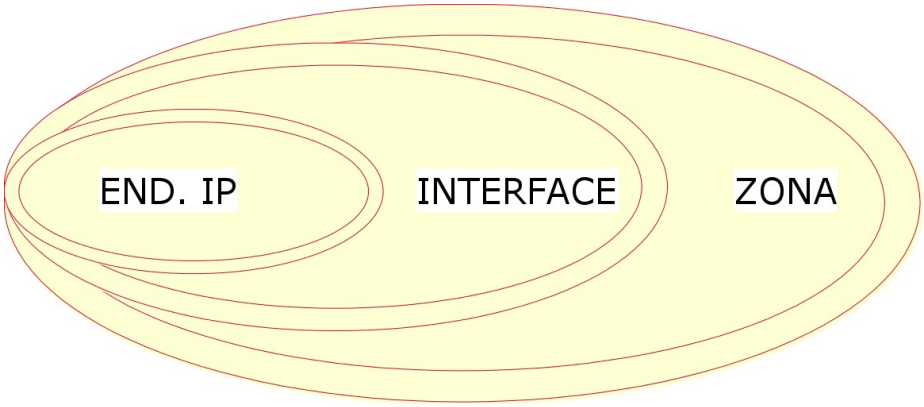

Figura 1.4 : Topologia de zonas de segurança

As zonas de segurança configuradas no equipamento serão os filtros mais amplos dentro das regras escritas e também os primeiros critérios a serem julgados pelo equipamento. Cada zona deverá compreender uma interface de rede e cada uma dessas deverá conter uma ou mais sub-redes, como demostrado na Figura 1.2. Tipicamente, teremos uma relação de uma zona para uma interface para uma sub-rede. Como a configuração do equipamento não será mais absolutamente linear, onde existem muitas regras sobrepostas, e sim separada pela zona de entrada e zona de saída do pacote, sua administração e trabalho serão muito otimizados (MAJ; MAKASIRANONDH; VEAL, 2010). Como regras são divididas em grupos inter zonas especializadas, para cada pacote transitado, apenas um grupo é analisado e não todas as regras configuradas, como descrito na Tabela 1.1.

Zona 1 >>> Zona 2	**Zona 2 >>> Zona 1**
Regra 1	Regra 4
Regra 2	Regra 5
Regra 3	Regra 6

Tabela 1.1: Exemplo de configuração de regras entre zonas de segurança

Caso seja alocado mais de uma sub-rede para uma zona, o administrador deverá entender que as regras para essas sub-redes serão analisadas juntas e, neste caso, será necessário mais cuidado com sua sequência. Este caso normalmente acontece quando existe uma segmentação de rede para endereços com o mesmo nível de acesso dentro das políticas estabelecidas.

1.9 Agregação de Placas de Rede

A metodologia de agregação de placas de rede consiste na utilização de duas ou mais interfaces físicas em conjunto formando uma única interface lógica com um endereçamento IP único. Largamente utilizado em estruturas com *storages* de rede (GUIJARRO; GASPAR, 2008), esta metodologia foi criada por *Donald Becker* para implementação em estruturas de alto desempenho voltadas para cálculo numérico. Atualmente, sua utilização é voltada para estruturas críticas de alto desempenho. Esta técnica permite ao administrador implementar alguns modos de utilização das interfaces, entre elas, balanceamento de carga e tolerância a falhas com e sem dependência necessária de equipamentos secundários.

Visando maior estabilidade e redução dos pontos únicos de falha, isto é, pontos que sozinhos poderiam interromper o funcionamento completo do sistema, serviços críticos fazem uso da agregação de placas de rede (GUIJARRO; GASPAR, 2008).

Os modos de utilização de placas de rede agregadas são[12]:

- Balanceamento *Round-Robin*: transmite os pacotes de maneira sequencial pelas placas de rede físicas disponíveis, desde a primeira até a ultima, um pacote por placa, provendo balanceamento de carga e tolerância a falhas;
- Ativo-Passivo: apenas uma placa física é utilizada e as demais ficam em modo passivo até que a placa ativa apresente problemas, apenas a tolerância a falhas é provido; a transmissão não irá atingir velocidades superiores à das interfaces individuais;
- Balanceamento Exclusivo: transmissão de pacotes baseada em política de *hash*; para garantir que os MAC destino sejam sempre atendidos pelo mesmo MAC de origem; este modo fornece balanceamento de carga e tolerância a falhas; a transmissão não irá atingir velocidades superiores à das interfaces individuais; a política padrão é:

12 *Disponível em:* <http://www.linuxfundation.org>/collaborate/workgroups/networking/bonding>. Acesso em: 13 set. 2012.

[fonte ({endereço MAC origem} XOR {endereço MAC de destino}) modulo de contagem escravo];

- *Broadcast*: transmite todos os pacotes em todas as placas de rede, provendo tolerância a falhas; a velocidade de transmissão não será superior à das interfaces individuais;
- IEEE 802.3ad: cria um grupo agregado de placas de rede de acordo com as regulamentações da IEEE 802.3ad; todas as placas irão compartilhar a mesma velocidade de transmissão e a mesma configuração de duplex; provê balanceamento de carga e tolerância a falhas; a transmissão poderá atingir velocidades superiores à das interfaces individuais;
- Balanceamento de Transmissão Adaptativo: a transmissão dos pacotes de saída usa a da interface que estiver menos carregada, ou seja, de acordo com carga em cada interface a saída de pacotes irá utilizar aquela que estiver mais livre; a entrada de pacotes utilizará a interface corrente, ou seja, a última que foi utilizada para saída de pacotes, teoricamente menos carregada; provê balanceamento de carga na saída dos dados e tolerância a falhas;
- Balanceamento Adaptativo Completo: segue a mesma forma de funcionamento do modo anterior com o acréscimo do balanceamento na entrada de pacotes também, ou seja, a análise de carga das interfaces é feito na saída e na entrada, provê balanceamento de carga completo e tolerância a falhas.

Os modos ativo-passivo, balanceamento de transmissão adaptativa e balanceamento adaptativo completo não demandam configuração específica do *switch*. O modo IEEE 802.3ad requer que o *switch* tenha as portas apropriadas configuradas como uma agregação do tipo 802.3ad. O método preciso usado para esta configuração dependerá do fabricante do equipamento. O modo balanceamento *round-robin*, balanceamento exclusivo *broadcast* geralmente também demandam que o *switch* tenha as portas apropriadas agrupadas.

Neste trabalho, é vislumbrada a possibilidade do administrador usar

interfaces agregadas para aumento do desempenho em transferências internas, possibilitando o uso do equipamento com taxas maiores que 1Gbit/s sem necessidade de utilização de interfaces de 10Gbit/s e, consequentemente, *switches* de mesmo desempenho. Também será vislumbrada a possibilidade de uso da agregação de placas em modo ativo-passivo, funcionando com uma interface *backup,* diminuindo os pontos únicos de falha.

1.10 Virtual LANs[13]

VLANs *Ethernet* estão sendo comumente usadas em redes corporativas para o fins de redução do domínio de *broadcast*. O padrão IEEE 802.1Q VLAN permite que diferentes redes possam ser implementadas em um único meio físico através da inserção de uma marcação de "Virtual LAN".

Hosts e *switches* que suportam VLANs efetivamente permitirão o encaminhamento baseado em *software* dos pacotes trafegados com a reconfiguração dos parâmetros de marcação. Por meio desta nova marca de pacotes, a infraestrutura de rede pode ser alterada logicamente sem alteração fisicamente da fiação entre os segmentos.

Esta metodologia permite que muitas redes lógicas possam utilizar o mesmo meio físico sem o risco do acesso mútuo direto, sem um encaminhamento prévio de um *router* ou equipamento de maior segurança.

1.11 Filtro na Camada de Aplicação

A camada de aplicação, como descrito anteriormente, é a camada do pacote transmitido que contem as informações úteis da aplicação executada, ela carrega a carga líquida. Essa carga identifica efetivamente o dado trafegado contendo informações que podem ser muito relevantes para o administrador da rede, podendo este efetuar classificações que priorizem ou neguem sua passagem. Uma aplicação maliciosa, por exemplo, pela análise desta carga, poderia ser mapeada e bloqueada diretamente pelas regras do *firewall*.

13 Disponível em: <http://xml2rfc.tools.ietf.org/html/rfc4554>. Acessado em 20 jul. 2012.

O funcionamento destes filtros é baseado em busca de expressões regulares dentro da camada de aplicação do pacote e prevê que o administrador possa escolher os programas habilitados a trocar pacotes utilizando as interfaces de seu *firewall* (PEREIRA; RIBEIRO; CARVALHO, 2010).

Dois exemplos de bloqueio de uso mal intencionado são dos vírus NIMDA[14] e CODE_RED[15], ambos muito famosos por seus efeitos desastrosos e pela facilidade com que infectaram muitas máquinas pelo mundo. Pelo uso do filtro de camada de aplicação esse tipo de ataque pode ser evitado assim que o conteúdo do pacote mal intencionado estiver sido identificado e catalogado na base de assinaturas filtradas. Também podemos citar tentativas de invasão como *SQL Injection* (HALFOND; ORSO, 2005) que podem ser filtradas pelo mesmo método uma vez que utilizam alguns tipos de *strings* bem conhecidas.

1.12 Características Desejadas de um *Equipamento de Segurança*

O desenvolvimento de um bom equipamento de segurança presume que este possa executar muito mais que apenas uma simples filtragem de pacotes. A fim de que um administrador possa ter o real controle de sua rede de dados e também atenda às demandas cada vez mais exigentes da segurança da informação, um equipamento centralizado de segurança deve prover muitas funcionalidades.

Baseado na pesquisa dos elementos de quatro[16] equipamentos de segurança recentes de mercado, este trabalho vislumbra um ferramenta que possa prover as seguintes funções:

1. suporte para várias interfaces: para atender a grandes e médias redes, inclusive a arquitetura de rede que é proposto neste, é muito interessante que exista disponibilidade de várias interfaces comportando várias sub-redes diretamente ligadas ao equipamento sem a necessidade de compartilhar recursos físicos;

14 Disponível em: <http://www.cert.org>/advisories/CA-2001-26.html>. Acesso em: 14 set. 2012.
15 Disponível em: <http://www.cert.org>/advisories/CA-2001-19.html>. Acesso em: 14 set. 2012.
16 Os equipamentos pesquisados são: Juniper Networks SSG 550M, Aker UTM Firewall 6.1, CheckPoint 12200 Series e FortGate 600C.

2. suporte para Virtual LAN – 802.1q (IEEE, 1997): com a exigência de recursos e compatibilidade entre equipamentos de grande porte, alguns padrões devem constar no equipamento para que seja possível seu uso de variadas formas; VLAN é usando comumente para segmentação de redes[17], autenticação e garantia de qualidade de serviço;
3. estabelecimento de políticas por zona de segurança: como já descrito no trabalho, é muito vantajoso o uso de zonas de segurança dentro das configurações do equipamento;
4. controle sobre o estado das conexões (ANDREASSON, 2012): o equipamento descrito pelo trabalho poderá controlar as conexões estabelecidas, sejam elas dirigidas ao equipamento ou para terceiros através de suas interfaces;
5. filtragem de conexões de acordo com suas características de origem e destino, portas, horário, tamanho, protocolo e *string* trafegada (ANDREASSON, 2012) e análise da camada de aplicação (PEREIRA; RIBEIRO; CARVALHO, 2010);
6. manipulação controlada das conexões estabelecidas[18]: por meio de desta capacidade, protocolos conhecidos de rede poderão ser mapeados, como FTP, HTTP e DNS, para melhor controle das regras de segurança;
7. possibilidade de uso de servidor de *syslog*[19] interno ou externo (ANDREASSON, 2012): como medida de proteção, os registros gerados pelo equipamento podem ser enviados para outro equipamento, aumentando suas possibilidades de armazenamento, visualização e coesão;
8. filtro de conteúdo web com *cache off-line* e antivírus integrado[20]

17 Disponível em: <http://xml2rfc.tools.ietf.org/html/rfc4554>. Acesso em: 20 jul 2012.
18 Disponível em: <http://conntrack-tools.netfilter.org>/manual.html>. Acesso em: 13 set. 2012.
19 Disponível em: <http://www.rsyslog.com>. Acesso em: 15 de Jun. 2012.
20 Disponível em: <http://www.squid-cache.org>. Acesso em: 11 jun. 2012.

[21]: o equipamento é capaz de categorizar as *URL's* acessadas juntamente com o uso de um sistema que varre os arquivos trafegados tentando garantir a inexistência de códigos maliciosos;

9. garantia de qualidade de serviço definido por endereço, protocolo, porta, horário, origem e destino (ANDREASSON, 2012) (ALMESBERGER, 2012) (GHEORGHE, 2006): o equipamento possibilitará o controle da taxa de uso de seus *links*, permitindo a garantia da taxa de transmissão com que determinadas regras irão trabalhar e do fluxo dos dados categorizados por origem, destino, protocolo, porta;
10. VPN para usuários[22]: o equipamento permite a conexão segura de usuários externos a rede interna mediante prévio cadastro no sistema;
11. alta disponibilidade de *hardware* completo (WELTE, 2002): para garantia de maior tempo sem interrupção, o equipamento conta, opcionalmente, com um segundo equipamento completo e pronto para ser acionado caso necessário; sua ativação é automática;
12. alta disponibilidade de interface individual: para garantia de maior tempo sem interrupção, cada interface do equipamento conta, opcionalmente, com uma segunda porta pronta para ser acionado caso necessário; sua ativação é automática;
13. monitoramento ativo de conexões por interface[23]: o controle dos serviços oferecidos pelo equipamento é fundamental para seu bom funcionamento, permitindo a identificação de problemas e avaliação de sua configuração;
14. histórico gráfico de uso de *links*[24]*:* o uso de gráficos permite a

21 Disponível em: <http://www.clamav.net>. Acesso em: 11 jun. 2012.
22 Disponível em: <http://www.openvpn.net>. Acesso em: 12 jun. 2012.
23 Disponível em: <http://www.ntop.org>. Acesso em: 13 jun. 2012.
24 Disponível em: <http://www.cacti.org>. Acesso em: 13 jun. 2012.

identificação de padrões de uso, tornando-se mais uma ferramenta para identificação de problemas e anormalidades;

15. envio de alarmes por e-mail utilizando serviço interno ou externo: o monitoramento histórico do equipamento pode ser armazenado em e-mail para futuros relatórios;
16. suporte para NAT - *Network Address Translation* - de serviço e de *host* (ANDREASSON, 2012);
17. balanceamento de carga entre *links* (HSUEH; LIN; HUANG, 2006): o equipamento poderá balancear o tráfego entre as interfaces configuradas para tal;
18. *backup* automático das configurações em servidor remoto e/ou por e-mail: todas as configurações do equipamento poderão ser enviadas para um servidor interno e/ou externo a rede ou mesmo anexada dentro de um e-mail;

As características acima listadas, como descrito anteriormente, poderão ser implementadas pelo administrador, uma vez que ele terá a liberdade de escolher exatamente o que precisa e deseja. A seguir, verificaremos como utilizar cada função em um exemplo de equipamento com todas as funções ativadas demonstrando as aplicações e configurações utilizadas.

1.13 Testes de Configuração

Ao configurar uma regra, o administrador objetiva a permissão de passagem controlada de pacotes pelo *firewall,* viabilizando o funcionamento de algum serviço desejado. Entretanto, é possível que a regra configurada permita a passagem de mais informações que o desejado, mascarando um grave erro. Para minimizar a ocorrência deste tipo de erro, as configurações devem ser testadas (MARMORSTEIN; KEARNS, 2006) contra falhas. Existem algumas formas para se testar e verificar o comportamento do *firewall*:

- testes ativos;

- testes passivos;
- análise de regras;

A primeira forma consiste em injetar pacotes de maneira controlada no *firewall* e verificar quais passam através da análise dos registros gerados e dos pacotes de saída. Desta forma, o administrador poderá verificar se suas regras estão ajustadas com as necessidades desejadas. Podemos citar algumas ferramentas como: *nmap*[25], hping[26], *ftester*[27] e *nessus*[28]. Esta forma exige do administrador uma boa dedicação para configuração dos programas e estudo da maneira como trabalham. Também é importante notar que serão exigidos muitos recursos da rede e do *firewall* em questão, logo, os testes terão de ser executados de forma planejada para que não interrompa os diversos serviços que a rede provê.

A verificação passiva (MARMORSTEIN; KEARNS, 2006) consiste na análise da estrutura de todas as regras, estudando cuidadosamente o que está sendo configurado em cada uma delas sem que sejam transmitidos pacotes pela rede. Este tipo de teste, teoricamente, verificará todas as possibilidades existentes de regras e pacotes a serem transmitidas pela rede. Para delimitar a quantidade de dados, o administrador deverá ajustar bem as possibilidades de endereçamento a serem testados, evitando os casos impossíveis. Uma ferramenta de auxílio para este caso é o *ITVal*[29].

A análise das regras é a simples verificação que todo administrador deve fazer sobre suas lógicas de configuração do *firewall*. A sintaxe dos comandos já será testada pelo próprio interpretador. Esta análise poderá ser demorada e passível de erros, dependendo da complexidade da rede.

1.14 Medidas de Desempenho

Para comparação com demais equipamentos, serão oferecidos pelo trabalho alguns testes (HICKMAN; et al., 2003) como demonstração do desempenho

25 Disponível em: <http://nmap.org>. Acesso em: 02 agosto 2012.
26 Disponível em: <http://www.hping.org>. Acesso em: 02 agosto 2012.
27 Disponível em: <http://www.inversepath.com>/ftester.html>. Acesso em: 02 agosto 2012.
28 Disponível em: <http://www.tenable.com>. Acesso em: 02 agosto 2012.
29 Disponível em: <http://itval.sourceforge.net>. Acesso em: 02 agosto 2012.

alcançado pela metodologia sugerida para o desenvolvimento do equipamento de segurança. Mais especificamente, serão encontrados os valores de *throughput* e latência do equipamento testado.

Inicialmente, algumas considerações devem ser previamente feitas para que seja estabelecido um padrão sobre os testes:

- Um computador cliente poderá originar várias conexões simultâneas com o servidor, sendo que esse número deve ser sempre indicado nos testes; desta forma, uma única fonte (computador) pode gerar várias requisições de clientes (conexões);
- Um ou mais clientes poderão se conectar a um ou mais servidores utilizando um determinado protocolo; para isso cada cliente deverá iniciar suas conexões de forma sequencial sobre os servidores, para que as conexões sejam igualmente distribuídas sobre os servidores;
- Como o equipamento é capaz de realizar tradução de endereços de rede (NAT), os testes serão executados com NAT ativado e, posteriormente, desativado para verificação de sua influência no desempenho;
- Serão realizados testes com diversas quantidades de regras de filtro aplicadas ao equipamento; para cada conjunto [cliente:servidor:protocolo], existirá uma regra específica permitindo o tráfego executado sempre ao final da lista completa para verificação do impacto no desempenho com relação ao número de regras aplicadas; sempre será configurado a política padrão de negação em todas as zonas de segurança;
- O protocolo de camada de aplicação utilizado para os testes será sempre especificado no ato de sua realização.

Nos testes propostos, haverá a descrição do ambiente utilizado, da tarefa executada e das variáveis envolvidas. Os resultados e comentários dos testes estarão inclusos no capítulo 3 deste trabalho.

1.15 Principais *Softwares* Envolvidos

Para alcançar os objetivos descritos neste trabalho, muitos serão os elementos utilizados. Estes elementos são representados pelos *softwares* que manipularão os dados garantindo a segurança da informação e o monitoramento sobre todo o processo.

Além dos *softwares* aqui descritos, outros farão parte da solução de forma indireta dentro da instalação e não serão completamente abordados. Sempre que necessário, será detalhado o uso do *software* e estes serão apontados mais a frente, junto a descrição dos procedimentos da implantação do sistema.

Segue um diagrama de funcionamento do equipamento como foi vislumbrado por este trabalho:

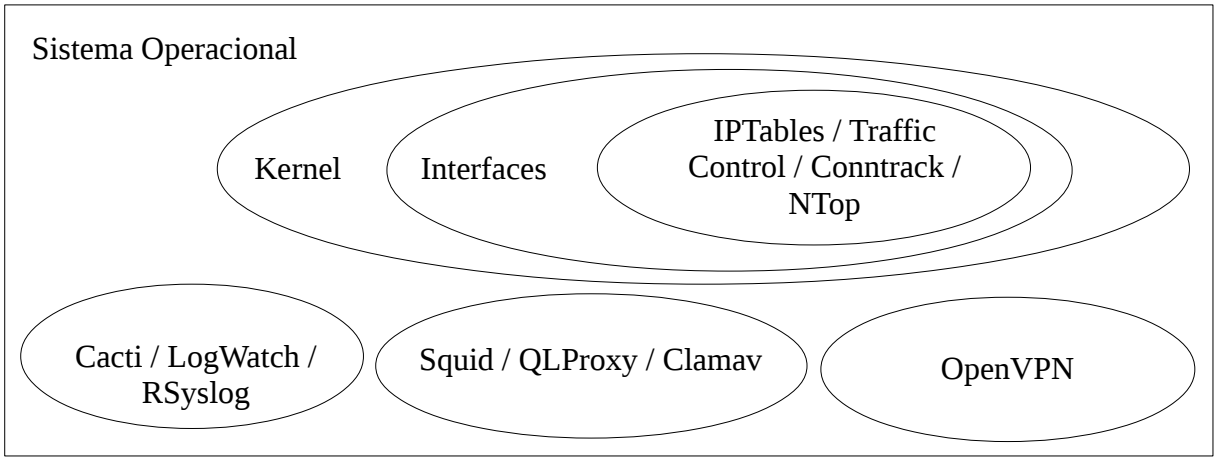

Figura 1.5 : Diagrama de integração dos softwares envolvidos

1.15.1 GNU/LINUX

O presente trabalho será desenvolvido sobre o sistema operacional *GNU/Linux*, mais especificamente o *CentOS 6.4 - Communty Enterprise Operational System*[30]. A escolha deste específico sistema se deve a sua estabilidade, desenvolvimento responsável, à larga disponibilidade de pacotes (inclusive aqueles necessários ao desenvolvimento deste trabalho) em seus repositórios livres e ao seu constante ciclo de atualizações. Também aos seus módulos de *kernel* que contemplam a grande maioria dos dispositivos de rede atuais, permitem a

30 Disponível em: <http://www.centos.org>. Acesso em: 07 jun. 2012.

manipulação dos pacotes de rede trafegados, viabilizando o uso, inclusive, de VLANs, agregação de placas de rede e controle das conexões estabelecidas através de suas interfaces.

1.15.2 IPTABLES

O *IPTables* é um programa desenvolvido para manipulação das tabelas de filtragem de pacotes de rede do *kernel* do sistema em ambiente de usuário. Permite a escrita de regras de filtragem de pacotes orientadas por zona de segurança, porta de destino e origem, IP de origem e destino, protocolo utilizado, horário, tamanho e *string* trafegada. Também permite o controle de alguns protocolos por meio dos *gateways* de aplicação que mapeiam a conexão que usa o protocolo em questão. Os serviços de tradução de endereçamento, *NAT – Network Address Translation*, também serão providos por este. Será o principal agente dentro deste trabalho (ANDREASSON, 2012).

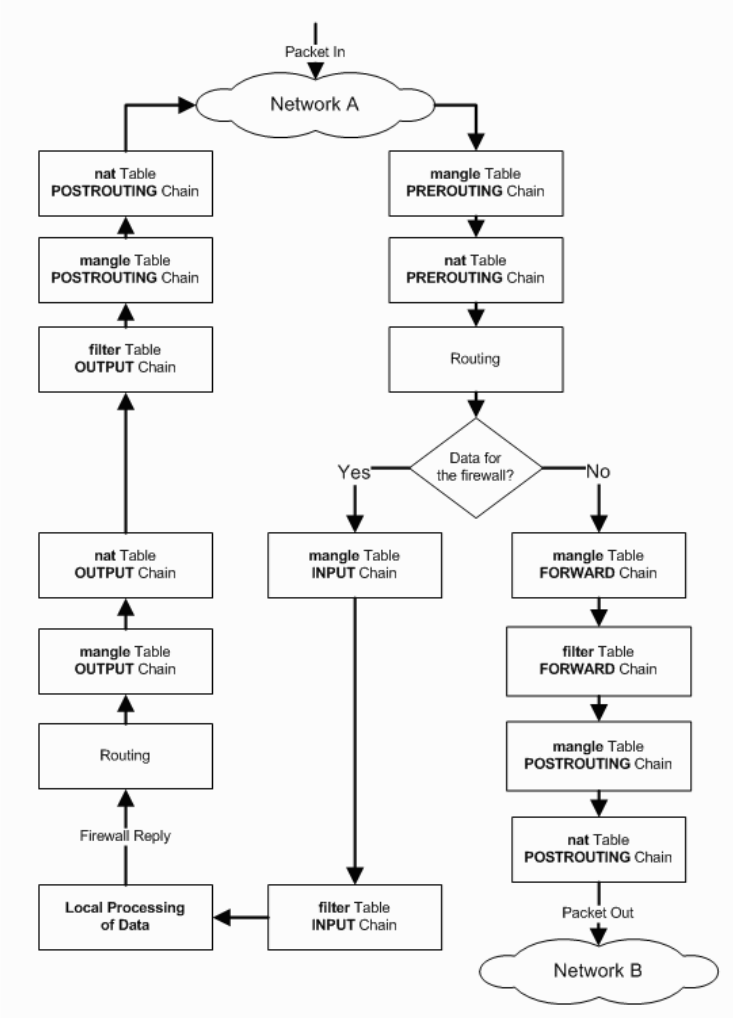

Figura 1.6: Fluxograma de funcionamento interno do IPTABLES. Fonte: www.netfilter.org

1.15.3 CONNTRACK-TOOLS

O *CONNTRACK-TOOLS* é um conjunto de *softwares* livres que permite o administrador interagir com a tabela de conexões estabelecidas do *kernel* do sistema. Com essa função o equipamento poderá copiar sua tabela para um segundo equipamento estabelecendo um redundância ativa das conexões. Também poderá monitorar as conexões ativa gerando estatísticas de uso (WELTE, 2002).

1.15.4 TC

O *TRAFFIC CONTROL* é um conjunto de instruções e mecanismos para tratar os pacotes recebidos e transmitidos pelo equipamento. Assim, são controladas as velocidades de transmissão dos dados pelas interfaces de entrada e saída de

acordo com as configurações estabelecidas pelo administrador. Estas configurações determinarão as filas de disciplina que irão (re)ordenar os pacotes para distribuição posterior (ALMESBERGER, 2012). Essas filas são coordenadas por regras escritas pelo *IPTables*, podendo assim, contar com todos os filtros já descritos.

1.15.5 SQUID-CACHE

O *SQUID*[31] é um programa de código aberto para *proxy* do acesso à internet. Suporta os protocolos mais comuns (HTTP, HTTPS, FTP) e faz uso de *cache* local no equipamento que o executa para melhor uso do *link* de acesso pois armazena as informações de páginas estáticas frequentemente acessadas. Também conta com muitos ajustes para melhor se adequar a qualquer tipo de ambiente ou necessidade, inclusive com a possibilidade de uso de controle sobre a taxa de transferência de seus pacotes.

1.15.6 QLPROXY

QLProxy[32] é um filtro de conteúdo para páginas na *internet*. Através dele é possível o controle do acesso a *sites* de acordo com o seu conteúdo. Desta forma, é possível filtrar o conteúdo acessado pelos clientes através do *firewall*. O programa funciona em conjunto com o *SQUID* e faz, entre outras, análise direta da *URL* acessada pesquisando em suas bases para avaliar a liberação ou bloqueio. Essa base é dividida em categorias que podem ser ajustadas a usuários específicos, criando as políticas de acesso. Suas principais funções são: controle de acesso a *sites* baseado em categorias, bloqueio de conteúdo inapropriado com filtro heurístico, controle de arquivos baseado em tipos, políticas de acesso, remoção de propagandas nas páginas acessadas.

31 Disponível em: <http://www.squid-cache.org>. Acesso em: 11 jun. 2012.
32 Disponível em: <http://www.quintolabs.com>. Acesso em 20 jun 2012.

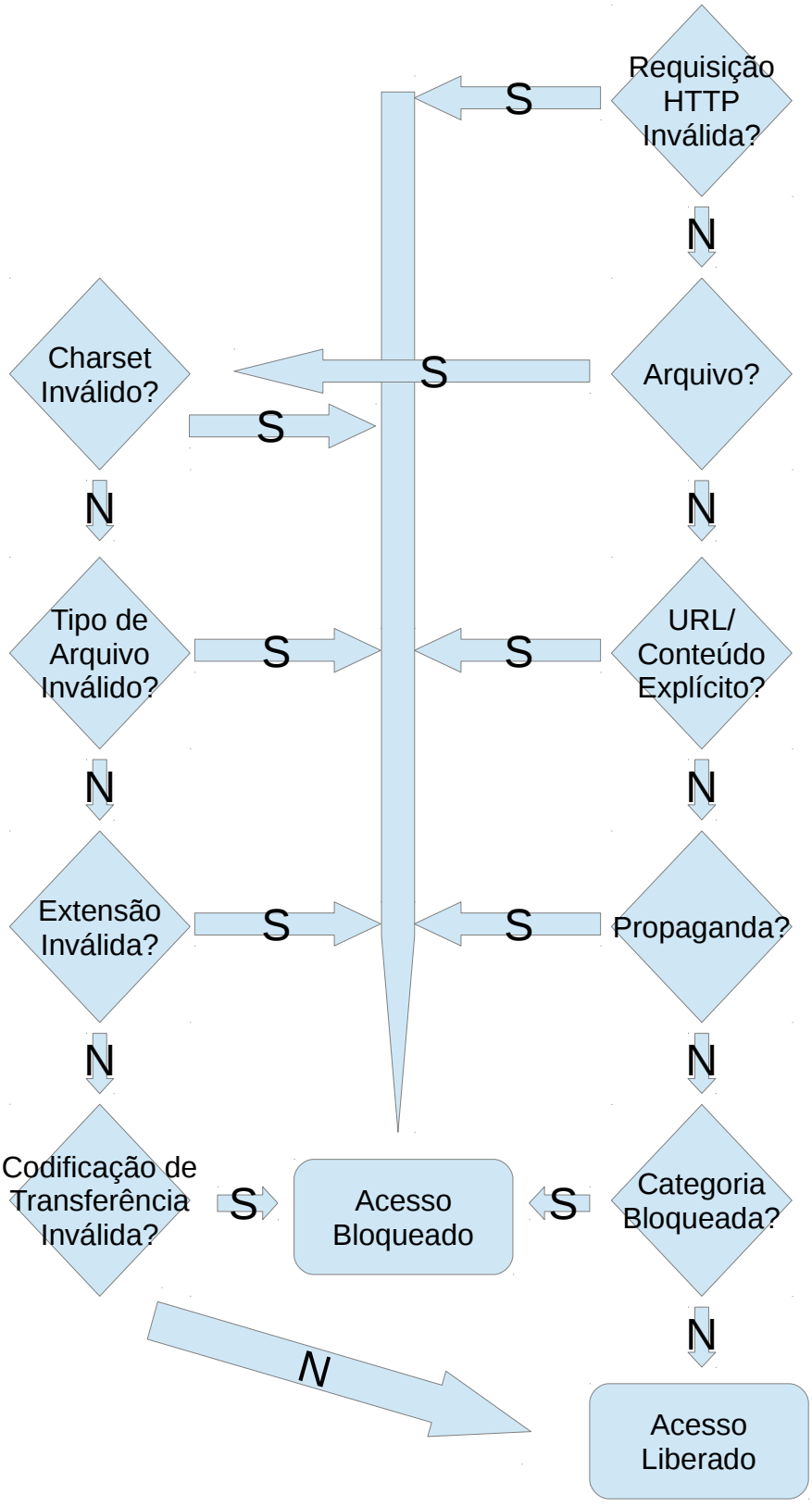

Figura 1.7: Fluxograma de funcionamento interno do QLPROXY

1.15.7 CLAMAV

O *CLAMAV*[33] é um programa de código aberto com função de antivírus para detecção de ameças maliciosas em tempo de conexão. Em conjunto com o *SQUID*, fará verificação dos pacotes por este manipulado oferecendo proteção para os usuários finais.

1.15.8 LOGWATCH

O *LOGWATCH* é um programa de código aberto utilizado para enviar um resumo dos registros do sistema operacional. No caso deste trabalho, será utilizado para resumir e enviar todos os avisos de bloqueio, juntamente com os acessos e avisos gerais do sistema. Esse envio é formatado dentro de um *e-mail* enviado para o administrador.

1.15.9 NTOP

O *NTOP*[34] é um programa de código aberto desenvolvido para monitoramento do tráfego de rede em tempo real. Trabalha diretamente com a placa de rede analisando todos os pacotes transitados. Sua visualização é organizada em resultados formatados por uma página HTML. Este proverá o monitoramento ativo das conexões por interface do equipamento.

1.15.10 CACTI

O *CACTI*[35] é um sistema completo web para monitoramento gráfico utilizando ferramentas RRD - *Round Robin Database* para armazenamento e exibição de dados e coleta via SNMP - *Simple Network Managent Protocol*. Possui uma série de *plugins* que permitem agregação de serviços ao sistema principal. Este sistema proverá o histórico gráfico de utilização dos *links*, bem como sua visualização do uso em tempo real. Também proverá um sistema de alerta para problemas e sobrecargas no tráfego das interfaces.

---

33 Disponível em: <http://www.clamav.net>. Acesso em: 11 jun. 2012.
34 Disponível em: <http://ntop.org>. Acesso em: 13 jun. 2012.
35 Disponível em: <http://www.cacti.net>. Acesso em: 14 jun. 2012.

1.15.11 OPENVPN

O *OPENVPN*[36] é uma ferramenta livre para autenticação e encriptação de usuários para que os dados de uma rede privada protegida pelo equipamento possa ser acessada de maneira segura através da *internet*.

36 Disponível em: <http://www.openvpn.net/>. Acesso em: 14 jun. 2012.

2 CONDIÇÕES GERAIS DA SOLUÇÃO

2.1 Licenciamento dos *Softwares* Envolvidos

Todos os *softwares* sugeridos para este trabalho possuem algum tipo de licenciamento **livre**. Dentre as licenças envolvidas, temos a *GNU General Public Licence*[37] e a *BSD Licence*[38].

2.1.1 Gnu General Public Licence

A *GNU General Public License* é uma licença *copyleft* livre para softwares e outros tipos de obras. Diferente de qualquer outra licença comercial, a GNU GPL pretende garantir sua liberdade de compartilhar e modificar todas as versões de um programa, certificando que o *software* permaneça livre para todos os seus usuários.

Como citado anteriormente, o *software* livre se refere à liberdade de utilização do *software* e manipulação do código fonte, não ao seu valor financeiro. A GNU GPL foi desenvolvida para garantir que o desenvolvedor tenha a liberdade de divulgar e distribuir cópias de seu *software* livre, podendo, inclusive, cobrar por elas caso assim deseje. Quem recebe o *software* poderá alterá-lo ou utilizá-lo em partes em novos *softwares* livres desenvolvidos.

Por este motivo, a GNU GPL também previne que um usuário do código-fonte originalmente licenciado com a GNU GPL possa vir tornar a seu novo programa fechado com uma licença diferenciada: todos os direitos que foram adquiridos com licença GNU GPL devem permanecer quando são utilizados em novos programas. Ou seja, novos programas que utilizem programas livres, devem permanecer livres, mantendo o mesmo licenciamento, garantindo os mesmos direitos originais.

Uma característica importante para proteção dos desenvolvedores e autores da GNU GPL explica claramente que não **há nenhuma garantia para software livre**. A GNU GPL também requer que versões modificadas sejam

37 Disponível em: <http://www.gnu.org/licenses/gpl.html>. Acesso em: 28 agosto 2012.
38 Disponível em: <http://opensource.org/licenses/bsd-license.php>. Acesso em: 30 agosto 2012.

marcadas como modificadas, de modo que os seus problemas não serão atribuídos erroneamente a autores de versões anteriores.

Além destas características, um *software* livre licenciado com GNU GPL deve obedecer sempre as 4 premissas anteriormente citadas neste trabalho.

O licenciamento dos *softwares* utilizados neste trabalho são GNU GPL com exceção do QLProxy, desenvolvido pela QuintoLabs[39]. Este *software* contém partes livres que obedecem todas as regras da GNU GPL, entretanto, seu núcleo tem licenciamento próprio da empresa desenvolvedora. Ainda assim, sua distribuição é gratuita sendo cobrado apelas o suporte ao usuário.

2.2 Documentação

O trabalho descreve uma metodologia para o desenvolvimento de um equipamento de segurança e sugere que, ao ser implantado, o executor crie uma documentação de todos os passo tomados na criação. Este será o **documento de implantação** do equipamento: deverá abordar os pontos e questões discutidos para o funcionamento básico, apontando os arquivos e locais de configuração, porém ainda sem o detalhamento das configurações. O citado documento **será descrito** mais a frente na implementação executada para demonstração do equipamento.

Os *softwares* propostos possuem uma boa documentação e o administrador deverá estudá-los para a melhor aplicação de suas funções. Os manuais necessários estão apontados dentro deste trabalho, especificamente em suas referências. Mais detalhes serão discriminados no capítulo que trata de sua implementação.

Como parte da política de segurança já discutida anteriormente neste trabalho, é fundamental também que todo o empenho nos ajustes do equipamento sejam **documentados** a fim de que sejam sempre reprodutíveis por qualquer outro administrador. Para isso, este trabalho sugere que um segundo documento seja criado e que trate das **alterações e ajustes finos nas configurações** de todos os pacotes implementados objetivando a completa descrição do atual estado do equipamento. Toda alteração deverá constar nas revisões deste documento.

39 Disponível em http://www.quintolabs.com/index.php. Acessado: 4 fev. 2013.

Estes documentos deverão sofrer revisão periódica para que sempre reflitam o real estado do equipamento.

2.3 Instalação, Manutenção e Suporte

Para produtos proprietários poderão ser contratados todos os serviços de instalação, configuração e suporte técnico. Normalmente, são oferecidas garantias e tempos de recuperação, bem como contratos jurídicos para execução de todos os serviços.

Para o equipamento sugerido por este trabalho, toda a instalação, manutenção e suporte ficará por conta do administrador da rede de sua equipe. Assim, todos os passos da instalação dos serviços poderão ser cuidadosamente controlados, configurados e documentados sob o ponto de vista do administrador.

Mesmo dentro de ambientes críticos onde a interrupção dos serviços não é tolerada, a falta de suporte comercial pode ser compensada com algumas ações proativas do administrador.

É muito importante que o *hardware* utilizado para execução dos serviços tenha suporte do fabricante, com garantias sobre suas peças e funcionamento. O equipamento suporta redundância completa de *hardware*, esta função é muito relevante para que não haja interrupção das atividades prestadas. Também deve-se citar que existe uma série de elementos configurados para o monitoramento físico do equipamento, sendo interessante manter uma constante supervisão sobre estes.

Em um equipamento proprietário, por conta de suas cláusulas contratuais, novas adequações físicas no ambiente de rede poderão impedir seu uso, por exemplo uma ampliação do número de redes pode ultrapassar todas as portas de rede disponíveis do equipamento. Para casos extremos, como este, uma nova adequação de *hardware* e/ou *software* poderá acarretar uma nova elaboração completa do contrato comercial por questões de compatibilidade e capacidade da solução anteriormente desenhada. Enquanto que, com o uso do *software* livre, ajustes de *software/hardware* mais simples, poderão sanar o problema e não seriam necessários demais trâmites comerciais.

Tendo em vista a manipulação dos programas e de suas configurações realizadas pelo administrador sobre o *software* livre, os procedimentos deverão ser bem documentados. Com a vasta contribuição de grupos pela rede mundial e documentações compartilhadas de maneira colaborativa (LAKHANI, HIPPEL; 2003), o suporte ao *software* livre poderá ser encontrado gratuitamente. É preciso notar que não existem formas de garantir que seja encontrado o suporte ou mesmo a solução para algum tipo específico de problema. Entretanto, para *softwares* bem estabelecidos, como os que foram sugeridos por este trabalho, é notória a participação e empenho dos grupos de discussão para convergência das soluções ao problemas encontrados pelos usuários.

Atualmente, muitos *softwares* livres são utilizados por empresas privadas com interesse em melhorá-los ou explorá-los internamente para desenvolvimento de uma ferramenta mais adequada para suas necessidades (CAPRA et al., 2009). Este interesse é fundamentado pelos pontos de liberdade e facilidade de manipulação já abordados pelo trabalho. Esse movimento traz um novo rumo para alguns *softwares* por torná-los mais confiantes para o uso geral, criando um ciclo de aprimoramento e uso que fortalece ainda mais a comunidade que os suporta.

3 IMPLEMENTAÇÃO E TESTES DA SOLUÇÃO

A solução proposta foi implementada em dois servidores para testes. Todas as funcionalidades, descritas no item 1.12 deste trabalho, foram implementadas demonstrando a viabilidade, dificuldades e facilidades do equipamento sugerido.

O *hardware* da implementação teste possui 10 interfaces de rede onde foram configuradas da seguinte forma:

- 5 interfaces de placas agregadas utilizando o modo balanceamento adaptativo completo;
- 4 zonas de segurança: *Untrust, DMZ, LAN, WLAN*; cada zona com uma interface agregada
- Uma interface agregada configurada para *backup* completo de *hardware (fail-over* de máquina)

3.1 Arquitetura onde o sistema base foi instalado:

- processador: Intel(R) Xeon(R) X5550 2.67GHz Dual Core; memória cache: 8192 K;
- 4GB de memória RAM;
- 10 interfaces de rede de 1Gbit/s;
- 80GB de espaço físico em disco.

3.2 Instalação do sistema base CentOS 6.4:

3.2.1 Download do sistema:

O *download* do sistema foi feito a partir da seguinte fonte: http://mirror.hmc.edu/centos/6.4/isos/x86_64/CentOS-6.4-x86_64-minimal.iso

A partir do CD gravado, foi feita a instalação como seguem as descrições abaixo.

3.2.2 Particionamento:

O disco do sistema foi particionado como se segue:

- / com formatação ext4 e tamanho 6144MB;
- swap com tamanho 4096MB;
- /var/log com formatação ext4 e tamanho 10240MB;
- /home com formatação ext4 e com o restante do disco de espaço;

3.2.3 Ajuste dos parâmetros de rede[40]

Inicialmente, as configurações para a agregação de placas de rede. Assim, criaremos o arquivo */etc/modprob.d/bonding.conf* com o seguinte conteúdo:

```
[root@fw01 network-scripts]# cat /etc/modprob.d/bonding.conf
alias bond0 bonding
options bond0 miimon=80 mode=6
alias bond1 bonding
options bond1 miimon=80 mode=6
alias bond2 bonding
options bond2 miimon=80 mode=6
alias bond3 bondig
options bond3 miimon=80 mode=6
```

Essas configurações irão carregar o módulo do kernel para *bond* em modo de balanceamento adaptativo completo (mode=6). Caso o administrador prefira outro modo de atuação do agregador de placas de rede, deverá modificar o modo de inicialização do módulo *bond* no arquivo anterior e observar as demais configurações necessárias.

A seguir, foi necessário configurar as interfaces físicas para que sejam parte das interfaces virtuais agregadas. Os arquivos de configuração estão localizados na pasta */etc/sysconfig/network-scripts/* e tem nome *ifcfg-ethX*, onde o ***X*** é o número da interface. Para este caso, iremos configurar como segue:

```
 [root@fw01 network-scripts]# cat ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond0
```

40 Disponível em: <http://wiki.centos.org/TipsAndTricks/BondingInterfaces>. Acesso em: 30 ago. 2012.

```
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond0
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth2
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond1
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth3
DEVICE=eth3
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond1
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth4
DEVICE=eth4
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond2
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth5
DEVICE=eth5
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond2
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth6
DEVICE=eth6
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond3
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth7
DEVICE=eth7
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond3
SLAVE=yes
```

```
[root@fw01 network-scripts]# cat ifcfg-eth8
DEVICE=eth8
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond4
SLAVE=yes

[root@fw01 network-scripts]# cat ifcfg-eth9
DEVICE=eth9
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
MASTER=bond4
SLAVE=yes
```

Com as interfaces físicas configuradas, os arquivos de configuração das interfaces virtuais agregadas foram criados e seguem abaixo:

```
[root@fw01 network-scripts]# cat ifcfg-bond0
DEVICE=bond0
IPADDR=192.168.199.122
NETMASK=255.255.255.0
NETWORK=192.168.199.0
BROADCAST=192.168.199.255
GATEWAY=192.168.199.1
ONBOOT=yes
BOOTPROTO=none
USERCTL=no

[root@fw01 network-scripts]# cat ifcfg-bond1
DEVICE=bond1
IPADDR=192.168.200.1
NETMASK=255.255.255.0
NETWORK=192.168.200.0
BROADCAST=192.168.200.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no

[root@fw01 network-scripts]# cat ifcfg-bond2
DEVICE=bond2
IPADDR=192.168.201.1
NETMASK=255.255.255.0
NETWORK=192.168.201.0
BROADCAST=192.168.201.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no

[root@fw01 network-scripts]# cat ifcfg-bond3
DEVICE=bond3
IPADDR=192.168.202.1
NETMASK=255.255.255.0
NETWORK=192.168.202.0
BROADCAST=192.168.202.255
```

```
ONBOOT=yes
BOOTPROTO=none
USERCTL=no

[root@fw01 network-scripts]# cat ifcfg-bond4
DEVICE=bond4
IPADDR=192.168.203.122
NETMASK=255.255.255.0
NETWORK=192.168.203.0
BROADCAST=192.168.203.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

O servidor de nomes (DNS) do sistema foi configurado no arquivo /etc/resolv.conf:

```
[root@fw01 ~]# cat /etc/resolv.conf
search network.com.br
nameserver 192.168.199.24
```

3.2.4 Atualização Inicial do Sistema

```
[root@fwl ~]# yum update -y
```

Ao final destas atualizações, o sistema foi reinicializado para que as novas configurações de *kernel* fossem ser aplicadas.

```
[root@fw01 ~]# reboot
```

3.2.5 Adição de Repositórios

Para que o sistema consiga instalar os pacotes desejados, foi necessário a adição de alguns repositórios extras. Seguem os comandos executados:

```
[root@fwl01 ~]# yum install http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
[root@fwl01 ~]# yum install http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm
```

Para adicionar o repositório do *NTOP,* foi necessário escrever um novo arquivo de repositório. Segue o conteúdo:

```
[root@fwl01 ~]# cat /etc/yum.repos.d/ntop.repo
[ntop]
name=ntop packages
baseurl=http://rpm.ntop.org/$releasever/$basearch/
enabled=1
```

3.2.6 Instalação dos pacotes que não necessitam compilação:

A instalação dos programas que não necessitam compilação foi executada como segue:

```
[root@fwl01 ~]# yum install squid net-snmp logwatch cacti clamav clamav-db ntop5 ntopng openvpn ntp mc
man mysql-server phpMyAdmin php-mcrypt net-snmp-utils python-setuptools python-mako crypto-utils
mod_ssl clamd gcc make curl-devel clamav-devel perl-CGI wget keepalived --nogpgcheck -y
```

3.2.7 Configuração Squid

As configurações do *Squid*[41] são muitos amplas e devem ser estudadas com cuidado pelo administrador a fim de se alcançar o melhor desempenho para o caso desejado. Para esta implementação, foram utilizados os seguintes parâmetros:

```
[root@fw01 ~]# cat /etc/squid/squid.conf

#Nome do servidor que será publicado
visible_hostname fw01.local.com.br

#Definição sobre o status do SQUID
acl manager proto cache_object

#Definições de acessos do servidor
acl localhost src 127.0.0.1/32

#Definições de acessos para o localhost
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

#Definições de horário de trabalho
acl worktime1 time M T W H F 08:00-12:00
acl worktime2 time M T W H F 14:00-20:00

#Definição sobre a rede local que será permitida utilizar o servidor
acl localnet src 192.168.1.0/24 192.168.2.0/24

#Definições de portas que serão acessadas através do SQUID
acl SSL_ports port 443
acl Safe_ports port 20 21 80 443 1025-65535

#Definição de uma forma de conexão direta, utilizada para conexões criptografadas
acl CONNECT method CONNECT

#Definição do protocolo de FTP para utilização do SQUID
acl FTP proto FTP

#Definições de permição e negação de acesso às portas e métodos de conexão
http_access allow manager localhost
http_access deny manager
http_access allow CONNECT SSL_ports
http_access allow FTP
http_access deny CONNECT !SSL_ports
```

41 Disponível em: <http://www.squid-cache.org/Doc/>. Acessado em: 15 out. 2012.

```
http_access deny to_localhost
http_access allow localnet
http_access allow localhost
http_access deny all
always_direct allow all
ssl_bump allow all

#Porta e método que será utilizado pelo serviço de proxy
http_port 3128 intercept

#Endereco IP que sera utilizado pelo cluster mais a frente.
tcp_outgoing_address 192.168.199.120

#Lista de palavras que quanto encontradas na URL acessada são tratadas diretamente pelo próprio servidor e não são remetidas para outros servidores de cache que possam existir pela rede; é possível desenvolver uma rede de servidores de cache que irão trabalhar em conjunto
hierarchy_stoplist cgi-bin ?

#Quantidade de memória RAM alocada para o SQUID
cache_mem 1024 MB

#Tamanho máximo do objeto que poderá ficar alocado em memória RAM
maximum_object_size_in_memory 5 MB

#Tamanho do cache alocado em disco, quantidade de pastas e subpastas utilizadas
cache_dir ufs /var/spool/squid 10240 128 256

#Pasta para onde serão remetidas as conexões bloqueadas ou com algum tipo de erro
error_directory /usr/share/squid/errors/pt-br

#Local onde o Squid irá gravar seus arquivos de controle interno
coredump_dir /var/spool/squid

#Controle de tempo para manutenção dos dados arquivados no cache
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320

#Configurações para utilização do Icap do QLProxy e do Clamav
icap_enable on
icap_preview_enable on
icap_preview_size 4096
icap_persistent_connections on
icap_send_client_ip on
icap_send_client_username on
icap_service qlproxy1 reqmod_precache bypass=1 icap://127.0.0.1:1344/reqmod
icap_service qlproxy2 respmod_precache bypass=1 icap://127.0.0.1:1344/respmod
icap_service squidclamav1 reqmod_precache bypass=1 icap://127.0.0.1:1345/squidclamav
icap_service squidclamav2 respmod_precache bypass=1 icap://127.0.0.1:1345/squidclamav
adaptation_service_chain svcRequest qlproxy1 squidclamav1
adaptation_service_chain svcResponse qlproxy2 squidclamav2
adaptation_access svcRequest allow worktime1 all
adaptation_access svcResponse allow worktime1 all
adaptation_access svcRequest allow worktime2 all
adaptation_access svcResponse allow worktime2 all

#Configuração para limitação do uso da taxa de transferência do Squid baseado em horário
```

```
delay_pools 1
delay_class 1 2
delay_access 1 allow worktime1 localnet
delay_access 1 allow worktime2 localnet
delay_access 1 deny all
#50MB / 5MB
delay_parameters 1 6553600/6553600 655360/655360
```

3.2.8 Instalação e Configuração do QLProxy[42]

Para instalação do pacote original, foi executado:

```
[root@fwl01 ~]# yum install http://www.quintolabs.com/qlproxy/binaries/2.0.2/qlproxy-2.0.0-d746b.i386.rpm -y
```

As configurações são feitas no arquivo ***/etc/opt/quintolabs/qlproxy/qlproxy.conf***.

Este *software* é um servidor ICAP (*Internet Content Adaptarion Protocol*) e um reescritor de URL, ou seja, seu mecanismo trabalha em conjunto com o *Squid* para receber as páginas acessadas e verificar seus conteúdos analisando tanto a URL acessada quando o conteúdo das páginas. Dessa forma, ele pode redirecionar o usuário para uma segunda página de bloqueio dependendo do conteúdo que ele tentou acessar.

Todo o manual de configuração do *software* se encontra listado dentro do arquivo de configuração e o administrador deverá estudá-lo. É possível a criação de várias políticas com configurações específicas. Segue a configuração utilizada:

```
[root@fw01 ~]# cat /etc/opt/quintolabs/qlproxy/qlproxyd.conf

#Localização das configurações e pastas utilizadas
ETCDIR = /etc/opt/quintolabs/qlproxy
OPTDIR = /opt/quintolabs/qlproxy
VARDIR = /var/opt/quintolabs/qlproxy

#Usuário que irá executar o processo no sistema
user = qlproxy

#Arquivo de controle de processos
pidfile = $VARDIR/run/qlproxyd.pid

#IP e porta do processo ICAP
icap_address = 127.0.0.1
icap_port    = 1344

#Número de threads utilizados pelo sistema, quanto mais distribuído melhor o desempenho, mas é necessário um ajuste cuidadoso para não comprometer o sistema operacional; o sistema sugere que seja utilizado (2*<número de processadores>+1)
```

42 Disponível em <http://www.quintolabs.com/>. Acesso em: 27 set. 2012.

pool_threads = 5

#Quando um grande arquivo é passado do Squid para o QLProxy para análise, esta pode demorar um pouco em virtude do tamanho do arquivo, fazendo o usuário final esperar muito tempo e passando a impressão de que a conexão pode ter parado de funcionar. Para evitar essa impressão, o QLProxy começa a enviar alguns pacotes de volta para o usuário para que a conexão não fique ociosa até que ele receba o arquivo completo e termine sua análise. Esse processo é chamado *trickling*. Abaixo, é indicado o tamanho mínimo em bits do arquivo para que o QLProxy use o *trickling.* Para conexões mais rápidas, podemos ajustar maiores valores; valor em *bytes;*
# 512Kb
min_trickled_size = 524288

#Tamanho do pacote enviado para o usuário enquanto espera o arquivo completo; valor em *bytes;*
trickle_size = 8192

#Porcentagem máxima do tamanho do arquivo que será enviado para o usuário pelo método descrito acima. É importante notar que não podemos enviar 100% do tamanho do arquivo pois poderemos entregar um arquivo inválido; nem podemos colocar 0% pois esse valor desativaria o processo; 80% é o valor padrão.
trickled_percent = 80

#Valor do tamanho máximo do arquivo que será analisado; cada arquivo analisado precisa sar armazenado em memória RAM, então este valor deve ser ajustado de acordo com as especificações do servidor; valor em *bytes;*
max_trickled_size = 10485760 # default values is 10 Mb

#Ação padrão executada quando é encontrado uma URL na lista de não permitidos. Pode ser: *pass*, *just_log*, *redirect*, *block*.
action = block

#Caso o padrão anterior seja *redirect*, o caminho completo para onde o usuário será redirecionado.
redirect_url = http://www.quintolabs.com/redirect/index.php

#Caso o padrão anterior seja *block*, o caminho completo para onde o usuário será redirecionado.
blocked_page = $VARDIR/www/qlproxy/redirect/blocked.html

#Quando o sistema detecta uma imagem que deve ser bloqueada, normalmente propagandas, a imagem configurada abaixo é usada para preencher o espaço vazio.
blocked_image = $VARDIR/www/qlproxy/redirect/transparent.gif

#Ajuste do nível de *log* do sistema. Valores possíveis: *debug*, *info*, *warning*, *error.*
error_loglevel = info

#Ajuste do caminho do arquivo de saída de *log* de erro, indica as páginas bloqueadas.
error_log = $VARDIR/log/error.log

#Ajuste do caminho do arquivo de saída de *log* de acesso, indica as páginas acessadas.
access_log = $VARDIR/log/access.log

#Ajuste para envio de todas as requisições de acesso para o arquivos de *log;* utilizado apenas para depuração e ajuste fino do servidor, não deve ficar configurado em servidores em produção.
access_log_all = no

#Ajuste para envio de saídas de *log* para cada requisição, resposta e processamento executado pelo sistema; utilizado apenas para depuração e ajuste fino do servidor, não deve ficar configurado em servidores em produção.
debug_dump_enabled = no

#Caminho do arquivo de saída dos *logs* da opção anterior.
debug_dump_dir = $VARDIR/tmp

#Ativação do sistema de bloqueio de propagandas
adblock_enabled = yes

#Ajuste de quais subscrições utilizar; as subscrições são arquivos que apontam os *sites* de propagandas catalogas, estas listas são atualizadas diariamente; caso seja preciso um ajuste pessoal, deverá ser feito um apontamento dentro deste arquivo para indicar o arquivo com a lista de *sites* cadastrados.
adblock_subscriptions = $VARDIR/spool/adblock

# Ativação do sistema de bloqueio de URLs categorizadas
urlblock_enabled = yes

#Diretório raiz que contem os diretórios das categorias
urlblock_categories = $VARDIR/spool/urlblock/blacklists

#Ativação do sistema de bloqueio de páginas baseada em ações, ou seja, é possível bloquear uma determinada ação HTTP de acordo com o que for indicado em outro arquivo de configuração; os métodos podem ser: GET, POST, CONNECT, OPTIONS, HEAD, PUT, DELETE, TRACE; os arquivos de configuração estão localizados dentro das pastas de políticas.

httpblock_enabled = yes

#Ativação do sistema de bloqueio de *download* baseado em tipos de arquivos
contentblock_enabled = yes
contentblock_filetypes = $VARDIR/spool/contentblock/file_types.conf

#Ativação do sistema heurístico de bloqueio de conteúdo inapropriado; este sistema avalia o conteúdo das páginas acessadas e a pontua, dependendo da pontuação atingida, a página será negada dinamicamente.
adultblock_enabled = yes
adultblock_heuristics = $VARDIR/spool/adultblock/heuristics

#Inicialização do verificação de páginas já rotuladas como *Restricted to Adults*; algumas páginas já trazem em seus cabeçalhos uma identificação de que são inapropriadas por conter conteúdo adulto; assim, o sistema as bloqueia diretamente
rta_label_name = RATING
rta_label_value = RTA-5042-1996-1400-1577-RTA

#Lista de tipos páginas que serão inspecionadas
adultblock_inspected_type = text/plain
adultblock_inspected_type = text/html
adultblock_inspected_type = application/json

#Pasta onde está a lista de palavras e as pontuações atribuídas para verificação heurística
adultblock_inspected_content = $VARDIR/spool/adultblock/content

#Tamanho máximo do arquivo verificado pelo sistema de heurística; este valor deve estar em acordo com as especificações do servidor, uma vez que os arquivos verificados são armazenados em memória RAM; valor em *bytes;*
adultblock_inspected_size = 307200

#Políticas utilizadas pelo sistema; uma política é o conjunto de regras e usuários; dentro de cada pasta de políticas existem diversos arquivos para controlar todos os sistema previamente listados aqui; caso o IP ou usuário não esteja listado em nenhuma política, ele é classificado dentro da política *default;*
use_policy = $ETCDIR/policies/default
use_policy = $ETCDIR/policies/strict
use_policy = $ETCDIR/policies/relaxed

3.2.9 Instalação e Configuração do C-Icap[43]

O serviço do *c-icap* foi compilado no servidor, pois não é encontrado nos repositórios oficiais. O código fonte utilizado na época da instalação estava na versão 0.2.5. É de suma importância que o administrador verifique se a versão a ser utilizada possui os mesmos parâmetros de configuração e funcionalidades aqui descritas.

Para começar a instalação, foram executados os seguintes passos:

```
[root@fw01 ~]# wget http://downloads.sourceforge.net/project/c-icap/c-icap/0.2.x/c_icap-0.2.5.tar.gz
[root@fw01 ~]# tar xfv c_icap-0.2.5.tar.gz
[root@fw01 ~]# cd c_icap-0.2.5
[root@fw01 c_icap-0.2.5]# ./configure --enable-large-files
[root@fw01 c_icap-0.2.5]# make
[root@fw01 c_icap-0.2.5]# make install
[root@fw01 c_icap-0.2.5]# echo "/usr/local/c-icap/lib/c_icap" >> /etc/ld.so.conf.d/c-icap-0.2.5
[root@fw01 c_icap-0.2.5]# ldconfig
```

Feita essa compilação, foi necessário uma pequena configuração, além da padrão, em seus arquivos. Um manual contendo todos os parâmetros de configuração do serviço já está contido no arquivo principal do programa. Segue a configuração utilizada com os comentários pertinentes:

```
[root@fw01]# cat /usr/local/etc/c-icap.conf
PidFile /var/run/c-icap/c-icap.pid
CommandsSocket /var/run/c-icap/c-icap.ctl
Timeout 300
MaxKeepAliveRequests 100
KeepAliveTimeout 600
#Quantidade de processos abertos na inicialização
StartServers 10
#Quantidade máxima de processos abertos
MaxServers 20
#Quantidade mínima de threads iniciada
MinSpareThreads     10
#Quantidade máxima de threads iniciada
MaxSpareThreads     20
#Quantidades de clientes atendidos
ThreadsPerChild     10
MaxRequestsPerChild  1000
#Porta Utilizada para o serviço TCP, é necessário mudar a porta padrão para não conflitar com a porta utilizada
pelo serviço do QLProxy.
Port 1345
#Usuário e grupo dono do processo no sistema
User squid
Group squid
#Email para contato
ServerAdmin root@localhost
```

43 Disponível em: <http://c-icap.sourceforge.net/>. Acessado em: 16 out. 2012.

```
#Nome do host
ServerName fw01.local.com.br
TmpDir /var/tmp
MaxMemObject
DebugLevel 0
ModulesDir /usr/local/c-icap/lib/c_icap
ServicesDir /usr/local/c-icap/lib/c_icap
TemplateDir
TemplateDefaultLanguage en
LoadMagicFile /usr/local/c-icap/etc/c-icap.magic
RemoteProxyUsers off
RemoteProxyUserHeader X-Authenticated-User
RemoteProxyUserHeaderEncoded on
ServerLog /var/log/c-icap.log
AccessLog /var/log/c-icap-access.log
#Serviço do SquidClamav
Service squidclamav squidclamav.so
Service echo srv_echo.so
```

Nesta configuração, existem dois pontos muito importantes a serem ajustados pelo administrador: a quantidade de processos abertos e o serviço do antivírus. O primeiro está intimamente ligado ao desempenho do serviço. O segundo é essencial para o funcionamento do *i-cap* com o antivírus.

Seguindo a instalação, foi necessário criar alguns diretórios utilizados e ajustar suas permissões no sistema. Para isso, foram utilizados os comandos:

```
[root@fwl01 ~]# mkdir /var/run/c-icap/
[root@fwl01 ~]# chown squid:squid /var/run/c-icap/ -R
[root@fwl01 ~]# touch /var/log/c-icap.log
[root@fwl01 ~]# touch /var/log/c-icap-access.log
[root@fwl01 ~]# chown squid:squid /var/log/c-icap*
```

Por ser um pacote compilado, não há um *script* de inicialização do serviço. Entretanto, como este serviço deverá sempre se inicializado com o *Squid*, uma simples alteração na inicialização deste será suficiente para suprir essa necessidade. No arquivo /etc/init.d/squid, foram adicionadas algumas linhas como se segue:

```
[root@fwl ~]# cat /etc/init.d/squid
…...
case "$1" in
start)
        start;
        /usr/local/bin/c-icap;
        ;;

stop)
        stop;
        kill -9 $(cat /var/run/c-icap/c-icap.pid);
```

;;
….....

3.2.10 Instalação e Configuração do SquidClamav

O serviço do *SquidClamav* irá integrar o antivírus com o *proxy* utilizando o *C-Icap* como intermediador. Ele foi compilado no servidor, pois este pacote não é está disponível nos repositórios oficiais ou extras. O código fonte utilizado na época da instalação estava na versão 6.10. É de suma importância que o administrador verifique se a versão a ser utilizada possui os mesmos parâmetros de configuração e funcionalidades aqui descritas.

Para começar a instalação, foram executados os seguintes passos:

```
[root@fw01 ~]# wget http://downloads.sourceforge.net/project/squidclamav/squidclamav/6.10/squidclamav-6.10.tar.gz
[root@fw01 ~]# tar xfv squidclamav-6.10.tar.gz
[root@fw01 ~]# cd squidclamav-6.10
[root@fw01 c_icap-0.2.5]# ./configure --with-c-icap=/usr/local/
[root@fw01 c_icap-0.2.5]# make
[root@fw01 c_icap-0.2.5]# make install
```

Feita essa compilação, foi necessária uma pequena configuração, além da padrão, em seus arquivos. Um manual contendo todos os parâmetros de configuração do serviço já está contido no arquivo principal do programa. Segue a configuração utilizada com os comentários pertinentes:

```
[root@fw01 ~]# cat /etc/squidclamav.conf
# Tamanho maximo do aquivo verificado
maxsize 5000000

# Endereco URL para onde o usuario sera redirecionado em caso positivo de virus
redirect http://fw01.local.com.br/cgi-bin/clwarn.cgi

# Endereco e porta onde o processo do clamd estara escutando. E possível configurar mais de um servidor de antivirus.
clamd_ip 127.0.0.1
clamd_port 3310

# Valor em segundos que o squidclamav ira esperar pelo antivirus
timeout 1

#Registrar todos os casos de virus positivos
logredir 1

# Habilitar a verificacao de nome do cliente
dnslookup 1
```

3.2.11 Configuração do Clamav

Para que o antivírus se integre com os programas anteriormente configurados, é importante ajustá-lo para que responda as requisições na porta correta. Para isso, dentro de seu arquivo de configuração, foi preciso ajustar os dois parâmetro listados:

```
[root@fwl01 ~]# cat /etc/clamd.conf
…
TCPSocket 3310
TCPAddr 127.0.0.1
…
```

Os demais parâmetros devem ser estudados e ajustados para o equipamento desejado.

3.2.12 Configuração SNMPD

Para o serviço de monitoramento do equipamento, foi preciso configurar o serviço de SNMP – *Simple Network Management Protocol.* Este serviço é padrão do sistema e foi instalado anteriormente.

Para configurá-lo, seguem os comandos utilizados:

```
[root@fwl01 ~]# echo “rocommunity public” > /etc/snmp/snmpd.conf
[root@fwl01 ~]# echo “OPTIONS="-LS 0-4 d -Lf /dev/null -p /var/run/snmpd.pid" > /etc/sysconfig/snmpd
```

3.2.13 Configuração APACHE

Alguns elementos do equipamento farão uso do serviço de página do sistema como interface para o administrador. Algumas poucas configurações foram necessárias para liberar o acesso apenas local e dos endereços administrativos determinados. Para isso, executar:

```
[root@fw01 ~]# cd /etc/httpd/conf/
[root@fw01 conf]# vi httpd.conf
```

Neste arquivo, foi ajustado o nome do equipamento nas configurações do *apache*. Procurar pela chamada *ServerName* e trocar pelo nome do servidor:

```
ServerName fw01.local.com.br:80
```

Também, ajustar os IPs que terão permissão para acessar a página

principal do equipamento. Para isso, substituir todas as entradas ***Allow from all*** por ***Allow from 127.0.0.1 <IPs Administrativos separados por espaços simples>***.

Também é preciso analisar todas as configurações da pasta /etc/httpd/conf.d/, fazendo o mesmo tipo de ajuste: adicionar os endereços administrativos para acesso ao sistemas do equipamento.

3.2.14 Configuração MYSQL

Com o serviço de banco de dados MySQL instalado, foi preciso apenas definir uma senha para o usuário administrador que terá permissões completas sobre o sistema, seguem os comandos utlilzados:

```
[root@fw01 ~]# /etc/init.d/mysqld start
[root@fw01 ~]# /usr/bin/mysqladmin -u root password 'fw01password'
```

3.2.15 Configuração do PHP

Para ajustar o *PHP,* foi necessário editar o seu arquivo de inicialização para configurar apenas o *timezone*. Para isso, no arquivo /etc/php.ini, a palavra *date.timezone* foi localizada e adicionado *America/Sao_Paulo.* Segue exemplo:

```
[root@fw01 ~]# vi /etc/php.ini
date.timezone = “America/Sao_Paulo”
```

3.2.16 Configuração do CACTI

Inicialmente, foi necessário criar um usuário e uma base para o sistema *CACTI* para a importação da sua estrutura de dados. Seguem os comandos:

```
[root@fw01 ~]# mysqladmin --user=root create cacti #Criação da base de dados
[root@fw01 ~]# mysql cacti <  /usr/share/doc/cacti-0.8.8a/cacti.sql #Importação da estrutura para base de dados
[root@fw01 ~]# mysql -u root -p #Acesso a console do mysql para criação do usuário
mysql> GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY 'somepassword';
mysql> flush privileges;
mysql> quit;
```

Assim que todas as prévias configurações estavam prontas, foi necessário ajustar o próprio *CACTI* para utilizá-las. Para isso foi preciso editar o arquivo */usr/share/cacti/include/config.php.* Segue o comando e as configurações ajustadas:

```
[root@fw01 ~]# vi /usr/share/cacti/include/config.php
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
```

```
$database_username = "cactiuser";
$database_password = "cactipassword";
```

Para completar as configurações, foi necessário ativar a execução periódica no sistema *CRON*. Para tanto, foi editado o arquivo /etc/cron.d/cacti e habilitado sua execução, bastou apagar o “#” inicial da única linha do arquivo citado. Ao final, o arquivo ficou da seguinte maneira:

```
[root@fw01 ~]# cat /etc/cron.d/cacti
*/5 * * * *     cacti   /usr/bin/php /usr/share/cacti/poller.php > /dev/null 2>&1
```

Após este ultimo ajuste de configuração, restou apenas acessar o sistema para criação dos gráficos desejados. Para tal, foi necessário ter algum endereço administrativo já configurado no sistema para acessar a interface *web* do *CACTI:*

http://<endereço configurado no equipamento>/cacti/

Neste momento, foi acessado a página de instalação do sistema, uma simples leitura das instruções foi o bastante para seguir até o final do procedimento. Ao término, foi exibida a entrada do sistema, exigindo usuário e senha, respectivamente, *admin* e *admin.* Neste momento, foi exigido a troca da senha.

3.2.17 Configuração do NTop

O pacote do *Ntop* instalado no sistema foi inicializado a partir da linha de comando. Será proposta futura deste trabalho a criação de um *script* de inicialização automática.

Antes da sua inicialização, foram criadas pastas armazenamento de dados para cada instância executada. Para isso, foi executado:

```
[root@fw01 ~]# mkdir /var/lib/ntop[0123]
[root@fw01 ~]# chown ntop:ntop /var/lib/ntop*
```

Então, para sua inicialização, foram elaboradas as seguintes linhas de comando:

```
[root@fw01 ~]# ntop --user ntop --use-syslog=daemon --db-file-path /var/lib/ntop0 --trace-level 3 --http-server
3000 --interface bond0 --numeric-ip-address –daemon &

[root@fw01 ~]# ntop --user ntop --use-syslog=daemon --db-file-path /var/lib/ntop1 --trace-level 3 --http-server
3001 --interface bond1 --numeric-ip-address –daemon &

[root@fw01 ~]# ntop --user ntop --use-syslog=daemon --db-file-path /var/lib/ntop2 --trace-level 3 --http-server
```

```
3002 --interface bond2 --numeric-ip-address –daemon &

[root@fw01 ~]# ntop --user ntop --use-syslog=daemon --db-file-path /var/lib/ntop3 --trace-level 3 --http-server
3003 --interface bond3 --numeric-ip-address –daemon &
```

Na primeira execução do comando foi necessário definir uma senha para administração. Cada comando listado acima executa o serviço do *Ntop* utilizando uma interface específica, respectivamente *bond0* à *bond3,* e escuta em uma porta específica, respectivamente, de 3000 à 3002.

3.2.18 Ajustes de e-mail

Para que os alertas e avisos do sistema sejam enviados para os devidos e-mails administrativos, foi necessário configurá-los na lista do serviço de *e-mail* do sistema. Então, foi necessário editar o arquivo */etc/aliases* para adicionar o redirecionamento do usuário administrativo, ficando da seguinte forma:

```
[root@fwl01 ~]# cat /etc/aliases
# Basic system aliases -- these MUST be present.
mailer-daemon:  postmaster
postmaster:     root

# General redirections for pseudo accounts.
bin:            root
…
support:        postmaster

# trap decode to catch security attacks
decode:         root

# Person who should get root's mail
root:           fulano@email.com, beltrano@email2.com
```

Após qualquer alteração nesta lista, é preciso executar o comando para renovação das bases em arquivo, segue comando executado*:*

```
[root@fwl01 ~]# newaliases
```

Todos os e-mails dos serviços executados no sistema devem ser direcionados diretamente para a conta de *root* ou outro redirecionamento criado dentro deste arquivo para que todo o processo de envio do e-mail seja registrado nos arquivos de controle.

3.2.19 Regras de Firewall

As regras de filtragem de pacotes são configuradas no arquivo */etc/sysconfig/iptables.* Neste arquivo, foram escritas todas as regras de segurança do equipamento, a separação das zonas, as traduções de endereçamento (*NATs*) e as marcações de pacotes para aplicação do *traffic shaping*.

Como o objetivo deste trabalho não é esgotar as configurações de cada um de seus elementos, é necessário ao administrador familiaridade com o comando *iptables* para que possa configurar este arquivo com maior precisão.

Segue um exemplo de configuração contendo *NATs* de entrada, de saída, controle de tráfego :

```
#Configuracao de Traducao de Enderacamento##########################################
*nat
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

#NATs de saida das redes internas
-A POSTROUTING -o bond0 -s 192.168.168.0/24 -j SNAT --to-source 123.456.195.81
-A POSTROUTING -o bond0 -s 192.168.200.51 -j SNAT --to-source 123.456.195.86
-A POSTROUTING -o bond0 -s 192.168.200.0/25 -j SNAT --to-source 123.456.195.82
-A POSTROUTING -o bond0 -s 192.168.200.128/25 -j SNAT --to-source 123.456.195.83
-A POSTROUTING -o bond0 -s 192.168.201.0/24 -j SNAT --to-source 123.456.195.83
-A POSTROUTING -o bond0 -s 192.168.202.0/24 -j SNAT --to-source 123.456.195.83
-A POSTROUTING -o bond0 -s 192.168.0.2/32 -j SNAT --to-source 123.456.195.84

#NATs de entrada pelo canal da Internet
-A PREROUTING -i bond0 -d 123.456.195.81 -p tcp --dport 80 -j DNAT --to 192.168.168.106:80
-A PREROUTING -i bond0 -d 123.456.195.81 -p tcp --dport 443 -j DNAT --to 192.168.168.106:443
-A PREROUTING -i bond0 -d 123.456.195.84 -p tcp --dport 4000 -j DNAT --to 192.168.0.2:4000
-A PREROUTING -i bond0 -d 123.456.195.84 -p tcp --dport 7000 -j DNAT --to 192.168.0.2:7000
-A PREROUTING -i bond0 -d 123.456.195.84 -p tcp --dport 2000 -j DNAT --to 192.168.0.2:2000
-A PREROUTING -i bond0 -d 123.456.195.84 -p tcp --dport 8000 -j DNAT --to 192.168.0.2:8000
-A PREROUTING -i bond0 -d 123.456.195.84 -p tcp --dport 4322 -j DNAT --to 192.168.0.2:4322

#NATs internos
-A PREROUTING -i bond3 -s 192.168.200.0/24 -d 192.168.200.1 -p tcp --dport 20 -j DNAT --to
192.168.168.3:20
-A PREROUTING -i bond3 -s 192.168.200.0/24 -d 192.168.200.1 -p tcp --dport 21 -j DNAT --to
192.168.168.3:21

#Direcionamentos paginas WEB internas que nao sao acessadas via SQUID
-A PREROUTING -i bond3 -s 192.168.200.0/24 -d 192.168.168.6 -p tcp --dport 80 -j ACCEPT
-A PREROUTING -i bond3 -s 192.168.200.0/24 -d 192.168.168.6 -p tcp --dport 3306 -j ACCEPT
-A PREROUTING -i bond3 -s 192.168.201.0/24 -d 192.168.168.6 -p tcp --dport 80 -j ACCEPT
-A PREROUTING -i bond3 -s 192.168.201.0/24 -d 192.168.168.2 -p tcp --dport 80 -j ACCEPT
-A PREROUTING -i bond3 -s 192.168.200.0/24 -d 192.168.168.2 -p tcp --dport 80 -j ACCEPT
```

```
#Redirecionamento das redes internas para SQUID
-A PREROUTING -i bond3 -s 192.168.200.0/24 -p tcp --dport 80 -j DNAT --to 192.168.168.2:3128
-A PREROUTING -i bond3 -s 192.168.201.0/24 -p tcp --dport 80 -j DNAT --to 192.168.168.2:3128
-A PREROUTING -i bond3 -s 192.168.202.0/24 -p tcp --dport 80 -j DNAT --to 192.168.168.2:3128

COMMIT

#Fim das regras de NAT
#############################################################################################
#Regras de marcacao de pacotes para traffic shaping
*mangle

## Regras de UNTRUST para DMZ ###############################################################
-A FORWARD -i bond0 -o bond2 -d 192.168.168.2/32 -j CLASSIFY --set-class 1:11
-A FORWARD -i bond0 -o bond2 -d 192.168.168.4/32 -j CLASSIFY --set-class 1:11
-A FORWARD -i bond0 -o bond2 -d 192.168.168.7/32 -j CLASSIFY --set-class 1:11
-A FORWARD -i bond0 -o bond2 -d 192.168.168.3/32 -j CLASSIFY --set-class 1:12
-A FORWARD -i bond0 -o bond2 -d 192.168.168.5/32 -j CLASSIFY --set-class 1:12
-A FORWARD -i bond0 -o bond2 -d 192.168.168.6/32 -j CLASSIFY --set-class 1:13
-A FORWARD -i bond0 -o bond2 -d 192.168.168.106/32 -j CLASSIFY --set-class 1:13

## Regras de UNTRUST para LAN ##############################################################
-A FORWARD -i bond0 -o bond3 -s 200.175.254.141 -d 192.168.200.0/24 -j CLASSIFY --set-class 1:11
-A FORWARD -i bond0 -o bond3 -d 192.168.200.0/24 -p tcp -m multiport --dports 25,110,143,995 -j
CLASSIFY --set-class 1:12
-A FORWARD -i bond0 -o bond3 -d 192.168.200.0/24 -p tcp --dport 443 -j CLASSIFY --set-class 1:13
-A FORWARD -i bond0 -o bond3 -d 192.168.200.0/24 -j CLASSIFY --set-class 1:14

COMMIT

#############################################################################################
*filter

# CRIACAO DAS ZONAS ########################################################################
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
:FASE2 - [0:0]
:FASE3 - [0:0]
:FASE4 - [0:0]
:UNTRUST-DMZ - [0:0]
:UNTRUST-LAN - [0:0]
:UNTRUST-WLAN - [0:0]
:DMZ-UNTRUST - [0:0]
:DMZ-LAN - [0:0]
:DMZ-WLAN - [0:0]
:LAN-UNTRUST - [0:0]
:LAN-DMZ - [0:0]
:LAN-WLAN - [0:0]
:WLAN-UNTRUST - [0:0]
:WLAN-DMZ - [0:0]
:WLAN-LAN - [0:0]

# REGRAS GLOBAIS ###########################################################################
-A FORWARD -p icmp -m limit --icmp-type echo-request --limit 1/s -j ACCEPT
-A FORWARD -p icmp -m limit --icmp-type echo-reply --limit 1/s -j ACCEPT
-A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# DHCPD NO FIREWALL ##################################################################
-A INPUT -i bond3 -p udp --dport 67 -m state --state NEW -j ACCEPT
-A INPUT -i bond3 -p udp --dport 68 -m state --state NEW -j ACCEPT

# ACESSO AO FIREWALL COM CHAVE SEQUENCIAL
#############################################
-A FASE2 -i bond0 -m recent --name FASE1 --remove
-A FASE2 -i bond0 -m recent --name FASE2 --set
-A FASE3 -i bond0 -m recent --name FASE2 --remove
-A FASE3 -i bond0 -m recent --name FASE3 --set
-A FASE4 -i bond0 -m recent --name FASE3 --remove
-A FASE4 -i bond0 -m recent --name FASE4 --set
-A INPUT -i bond0 -p tcp -m recent --dport 5000 --set --name FASE1
-A INPUT -i bond0 -p tcp -m recent --rcheck --seconds 10 --name FASE1 --dport 7000 -j FASE2
-A INPUT -i bond0 -p tcp -m recent --rcheck --seconds 10 --name FASE2 --dport 9000 -j FASE3
-A INPUT -i bond0 -p tcp -m recent --rcheck --seconds 10 --name FASE3 --dport 8000 -j FASE4
-A INPUT -i bond0 -p tcp -m recent --rcheck --seconds 15 --name FASE4 --dport 4422 -j ACCEPT

# ACESSO AO FIREWALL ################################################################
-A INPUT -i lo -j ACCEPT
-A INPUT -i bond0 -p tcp -m multiport --sports 25,80,443,4322 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond0 -p udp --sport 123 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond0 -p tcp --dport 4422 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond0 -s 192.168.0.1 -p tcp --dport 4422 -j ACCEPT
-A INPUT -i bond1 -p icmp -m limit --icmp-type echo-request --limit 1/s -j ACCEPT
-A INPUT -i bond2 -p icmp -m limit --icmp-type echo-request --limit 1/s -j ACCEPT
-A INPUT -i bond3 -p icmp -m limit --icmp-type echo-request --limit 1/s -j ACCEPT
-A INPUT -p icmp -m limit --icmp-type echo-reply --limit 1/s -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.2/32 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.2/32 -p tcp --dport 4422 -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.2/32 -p udp --dport 161 -m state --state NEW,ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.2/32 -p tcp --sport 389 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.0/24 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.0/24 -p tcp --sport 3389 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -i bond2 -s 192.168.168.2/32 -p udp -m multiport --sports 53,123,389 -m state --state ESTABLISHED
-j ACCEPT
-A INPUT -i bond0 -j LOG --log-prefix "BLOQ-UNTRUST-FW: "
-A INPUT -i bond1 -j LOG --log-prefix "BLOQ-WLAN-FW: "
-A INPUT -i bond2 -j LOG --log-prefix "BLOQ-DMZ-FW: "
-A INPUT -i bond3 -j LOG --log-prefix "BLOQ-LAN-FW: "
-A INPUT -j REJECT --reject-with icmp-host-prohibited

# DIRECIONAMENTO DOS PACOTES PARA SUAS ZONAS
########################################
-A FORWARD -i bond0 -o bond1 -j UNTRUST-WLAN
-A FORWARD -i bond0 -o bond2 -j UNTRUST-DMZ
-A FORWARD -i bond0 -o bond3 -j UNTRUST-LAN
-A FORWARD -i bond1 -o bond0 -j WLAN-UNTRUST
-A FORWARD -i bond1 -o bond2 -j WLAN-DMZ
-A FORWARD -i bond1 -o bond3 -j WLAN-LAN
-A FORWARD -i bond2 -o bond0 -j DMZ-UNTRUST
-A FORWARD -i bond2 -o bond1 -j DMZ-WLAN
-A FORWARD -i bond2 -o bond3 -j DMZ-LAN
-A FORWARD -i bond3 -o bond0 -j LAN-UNTRUST
-A FORWARD -i bond3 -o bond1 -j LAN-WLAN
-A FORWARD -i bond3 -o bond2 -j LAN-DMZ

## IN: UNTRUST # OUT: DMZ ############################################################
```

```
-A UNTRUST-DMZ -d 192.168.168.106 -p tcp --dport 80 -j ACCEPT
-A UNTRUST-DMZ -d 192.168.168.106 -p tcp --dport 443 -j ACCEPT
-A UNTRUST-DMZ -j LOG --log-prefix "BLOQ-UNTRUST-DMZ: "

## IN: UNTRUST # OUT: WLAN ############################################################
-A UNTRUST-WLAN -j LOG --log-prefix "BLOQ-UNTRUST-WLAN: "

## IN: UNTRUST # OUT: LAN #############################################################
-A UNTRUST-LAN -j LOG --log-prefix "BLOQ-UNTRUST-LAN: "

## IN: DMZ # OUT: UNTRUST #############################################################
-A DMZ-UNTRUST -s 192.168.168.0/24 -m state --state NEW -j ACCEPT
-A DMZ-UNTRUST -j LOG --log-prefix "BLOQ-DMZ-UNTRUST: "

## IN: DMZ # OUT: LAN #################################################################
-A DMZ-LAN -s 192.168.168.2/32 -d 192.168.200.0/24 -p udp --dport 161 -m state --state NEW -j ACCEPT
-A DMZ-LAN -s 192.168.168.2/32 -d 192.168.200.0/24 -p tcp --dport 22 -m state --state NEW -j ACCEPT
-A DMZ-LAN -s 192.168.168.2/32 -d 192.168.200.0/24 -p tcp --dport 5900 -m state --state NEW -j ACCEPT
-A DMZ-LAN -s 192.168.168.3/32 -d 192.168.200.0/24 -p tcp --sport 20 -m state --state NEW -j ACCEPT
-A DMZ-LAN -j LOG --log-prefix "BLOQ-DMZ-LAN: "

## IN: DMZ # OUT: WLAN ################################################################
-A DMZ-WLAN -s 192.168.168.2/32 -d 192.168.201.0/24 -p tcp --dport 22 -m state --state NEW -j ACCEPT
-A DMZ-WLAN -s 192.168.168.2/32 -d 192.168.202.0/24 -p tcp --dport 22 -m state --state NEW -j ACCEPT
-A DMZ-WLAN -s 192.168.168.2/32 -d 192.168.201.0/24 -p tcp --dport 5900 -m state --state NEW -j ACCEPT
-A DMZ-WLAN -s 192.168.168.2/32 -d 192.168.202.0/24 -p tcp --dport 5900 -m state --state NEW -j ACCEPT
-A DMZ-WLAN -j LOG --log-prefix "BLOQ-DMZ-WLAN: "

## IN: LAN # OUT: UNTRUST #############################################################
-A LAN-UNTRUST -s 192.168.200.0/24 -p tcp --dport 1935 -j DROP
-A LAN-UNTRUST -s 192.168.200.51 -j ACCEPT
-A LAN-UNTRUST -s 192.168.200.0/24 -p tcp -m state -m connlimit --connlimit-above 50 --connlimit-mask
32 ! --state RELATED -j LOG --log-prefix "BLOQ-CON: "
-A LAN-UNTRUST -s 192.168.200.0/24 -p tcp -m state -m connlimit --connlimit-above 50 --connlimit-mask
32 ! --state RELATED -j DROP
-A LAN-UNTRUST -s 192.168.200.0/24 -p tcp -m limit --tcp-flags SYN,ACK,FIN,RST RST --limit 1/s -j
ACCEPT
-A LAN-UNTRUST -s 192.168.200.0/24 -m state --state NEW -j ACCEPT
-A LAN-UNTRUST -j LOG --log-prefix "BLOQ-LAN-UNTRUST: "

## IN: LAN # OUT: DMZ #################################################################
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.6/32 -p tcp --dport 80 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.6/32 -p tcp --dport 3306 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.106/32 -p tcp --dport 80 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.106/32 -p tcp --dport 443 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.106/32 -p tcp --dport 3306 -m state --state NEW -j ACCEPT

-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.2/32 -p tcp -m multiport –dports
514,80,443,5222,7777,389,22,53,3128,3130 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.2/32 -p udp -m multiport --dports 389,53,123,514 -m state
--state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.3/32 -p tcp -m multiport --dports
20,21,22,111,662,875,892,2049,32803 -m state --state NEW -j ACCEPT
-A LAN-DMZ -s 192.168.200.0/24 -d 192.168.168.3/32 -p udp -m multiport --dports
111,662,875,892,2049,8923,32769 -m state --state NEW -j ACCEPT
-A LAN-DMZ -j LOG --log-prefix "BLOQ-LAN-DMZ: "

## IN: LAN # OUT: WLAN ################################################################
```

```
-A LAN-WLAN -j LOG --log-prefix "BLOQ-LAN-WLAN: "

## IN: WLAN # OUT: UNTRUST ###########################################################
-A WLAN-UNTRUST -s 192.168.201.0/24 -p tcp -m state -m connlimit ! --state RELATED --connlimit-above
20 --connlimit-mask 32 -j LOG --log-prefix "BLOQ-CON: "
-A WLAN-UNTRUST -s 192.168.201.0/24 -p tcp -m state -m connlimit ! --state RELATED --connlimit-above
20 --connlimit-mask 32 -j DROP
-A WLAN-UNTRUST -s 192.168.201.0/24 -p tcp -m limit --tcp-flags SYN,ACK,FIN,RST RST --limit 1/s -j
ACCEPT
-A WLAN-UNTRUST -s 192.168.201.0/24 -p tcp -m multiport --dports 443,20,21 -m state --state NEW -j
ACCEPT
-A WLAN-UNTRUST -s 192.168.202.0/24 -p tcp -m state -m connlimit ! --state RELATED --connlimit-above
20 --connlimit-mask 32 -j LOG --log-prefix "BLOQ-CON: "
-A WLAN-UNTRUST -s 192.168.202.0/24 -p tcp -m state -m connlimit ! --state RELATED --connlimit-above
20 --connlimit-mask 32 -j DROP
-A WLAN-UNTRUST -s 192.168.202.0/24 -p tcp -m limit --tcp-flags SYN,ACK,FIN,RST RST --limit 1/s -j
ACCEPT
-A WLAN-UNTRUST -s 192.168.202.0/24 -p tcp -m multiport --dports 443,20,21 -m state --state NEW -j
ACCEPT
-A WLAN-UNTRUST -j LOG --log-prefix "BLOQ-WLAN-UNTRUST: "

## IN: WLAN # OUT: DMZ ###############################################################
-A WLAN-DMZ -s 192.168.201.0/24 -d 192.168.168.2/32 -p udp -m state --dport 53 --state NEW -j ACCEPT
-A WLAN-DMZ -s 192.168.201.0/24 -d 192.168.168.2/32 -p tcp -m state --dport 3128 --state NEW -j ACCEPT
-A WLAN-DMZ -s 192.168.201.0/24 -d 192.168.168.2/32 -p tcp -m state --dport 80 --state NEW -j ACCEPT
-A WLAN-DMZ -s 192.168.202.0/24 -d 192.168.168.2/32 -p udp -m state --dport 53 --state NEW -j ACCEPT
-A WLAN-DMZ -s 192.168.202.0/24 -d 192.168.168.2/32 -p tcp -m state --dport 3128 --state NEW -j ACCEPT
-A WLAN-DMZ -j LOG --log-prefix "BLOQ-WLAN-DMZ: "

## IN: WLAN # OUT: LAN ###############################################################
-A WLAN-LAN -j LOG --log-prefix "BLOQ-WLAN-LAN: "

COMMIT
```

3.2.20 Configurações do Traffic Control

Para o completo gerenciamento das filas de prioridades da entrega das pacotes pelas placas de rede, é necessário ao administrador algum estudo do comando *tc* (*Traffic Control*).

Neste trabalho, foi desenvolvido um *script* para configuração das filas como segue abaixo. Este foi salvo na pasta */usr/loca/sbin/.*

```
#!/bin/bash

TC=/sbin/tc

function stop {

for INT in bond1 bond2 bond3; do
   echo $INT
   $ TC qdisc del dev $INT root
done
```

```
}

function start {

INT=bond2
$TC qdisc add dev $INT root handle 1:0 htb default 1
$TC class add dev $INT parent 1:0 classid 1:1 htb rate 1024mbit prio 0                # BANDA PARA A PLACA
$TC class add dev $INT parent 1:1 classid 1:10 htb rate 50mbit prio 1                 # BANDA PARA INTERNET
$TC class add dev $INT parent 1:10 classid 1:11 htb rate 10mbit ceil 15mbit prio 2      # BANDA PARA HTTP
$TC class add dev $INT parent 1:10 classid 1:12 htb rate 5mbit ceil 7.5mbit prio 3      # BANDA PARA EMAIL
$TC class add dev $INT parent 1:10 classid 1:13 htb rate 10mbit ceil 15mbit prio 4      # BANDA PARA HTTPS
$TC class add dev $INT parent 1:10 classid 1:14 htb rate 1.5mbit ceil 2.0mbit prio 7    # BANDA PARA
TORRENT (UDP PORTAS ALTAS)
$TC class add dev $INT parent 1:10 classid 1:15 htb rate 15mbit ceil 20mbit prio 4      # BANDA PARA
DEMAIS PROTOCOLOS

#INT=bond3
#$TC qdisc add dev $INT root handle 1:0 htb default 1
#$TC class add dev $INT parent 1:0 classid 1:1 htb rate 1024mbit prio 0                # BANDA PARA A PLACA
#$TC class add dev $INT parent 1:1 classid 1:10 htb rate 3800kbit prio 1               # BANDA PARA
INTERNET
#$TC class add dev $INT parent 1:10 classid 1:11 htb rate 750kbit ceil 1000kbit prio 2   # BANDA PARA HTTP
#$TC class add dev $INT parent 1:10 classid 1:12 htb rate 750kbit ceil 900kbit prio 4    # BANDA PARA
EMAIL
#$TC class add dev $INT parent 1:10 classid 1:13 htb rate 750kbit ceil 900kbit prio 5    # BANDA PARA
HTTPS
#$TC class add dev $INT parent 1:10 classid 1:14 htb rate 750kbit ceil 900kbit prio 3    # BANDA PARA
DEMAIS PROTOCOLOS

}

function list {

for INT in bond1 bond2 bond3; do
   echo $INT
   for VAR in qdisc class; do
      $TC -d -s $VAR show dev $INT
   done
done

}

case $1 in
   start)      start;;
   stop)       stop;;
   list)       list;;
   restart)    stop; start;;
   *)          echo "Usage: tc start|stop|restart|list";;
esac
```

3.2.21 Configuração do DHCPD

Para configuração do serviço de entrega de endereços de rede para redes internas foi utilizada a seguinte configuração:

```
[root@fwl01 ~]# cat /etc/dhcp/dhcpd.conf
```

```
option domain-name "fwl01.com.br";
option domain-name-servers 192.168.1.1;
option ntp-servers 192.168.1.1;
option wpad code 252 = text;
option wpad "http://fwl01.local.com.br/wpad.dat\n";
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {

  range 192.168.1.31 192.168.1.60;
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.255;

  host host2 {
    hardware ethernet 08:60:6e:d1:87:97;
    fixed-address 192.168.1.2;
  }

  host host3 {
    hardware ethernet 5c:f9:dd:ee:07:ce;
    fixed-address 192.168.1.3;
  }
}
```

Nesta configuração foi utilizado um arquivo secundário de configuração do serviço de *proxy* para rede interna. Este arquivo contem:

```
[root@fwl01 ~]# cat /var/www/html/wpad.dat
function FindProxyForURL(url, host) {

if ( isPlainHostName( host ) ) {
        return "DIRECT";
}

if( dnsDomainIs ( host, ".local.com.br") ) {
        return "DIRECT";
}
return "PROXY fwl01.local.com.br:3128";
}
```

O objetivo desta configuração é passar aos computadores clientes um apontamento para o servidor *proxy*.

3.2.22 Instalação do Conntrackd

Para instalação do pacote que habilita a elaboração do cluster ativo-passivo, foi necessário instalar o *conntrack-tools.* Para isso, foi instalado um novo repositório que foi mantido **desabilitado** para todos os demais passos, com exceção desta instalação. Seguem os passos executados:

```
[root@fwl01 ~]# yum install http://centos.alt.ru/pub/repository/centos/6/x86_64/centalt-release-6-1.noarch.rpm
[root@fwl01 ~]# yum install conntrack-tools libnetfilter_conntrack libnfnetlink
[root@fwl01 ~]# vi /etc/yum.repos.d/centalt
…
enable=0
...
```

3.2.23 Ajustes do Conntrackd e KeepAlived

Para ajuste do *conntrack-tools*, segue a configuração utilizada com os comentários pertinentes:

```
[root@fw01 ~]# cat /etc/conntrackd/conntrackd.conf
Sync {
      Mode FTFW {
         DisableExternalCache Off
         CommitTimeout 1800
         PurgeTimeout 5
      }

      UDP {
# Endereco da interface por onde serao enviados os dados das conexoes
            IPv4_address 192.168.1.1
# Endereco do segundo equipamento que irá receber as conexoes
            IPv4_Destination_Address 192.168.1.2
            Port 3780
            Interface eth1
            SndSocketBuffer 1249280
            RcvSocketBuffer 1249280
            Checksum on
      }
}

General {
      Nice -20
      HashSize 32768
      HashLimit 131072
      LogFile on
      Syslog on
      LockFile /var/lock/conntrack.lock
      UNIX {
            Path /var/run/conntrackd.ctl
            Backlog 20
      }
      NetlinkBufferSize 2097152
      NetlinkBufferSizeMaxGrowth 8388608
      Filter From Userspace {
            Protocol Accept {
# Protocolos que serao exportados
                  TCP
                  UDP
                  ICMP
            }
            Address Ignore {
# Endereco cujas conexoes serao ignoradas neste processo
                  IPv4_address 127.0.0.1
```

```
                IPv4_address 192.168.1.1
                IPv4_address 192.168.199.123
            }
        }
}
```

Para ajuste do *keepalived*, segue a configuração comentada do nó principal:

```
[root@fw01 ~]# cat /etc/keepalived/keepalived.conf
vrrp_sync_group FW-INT {
   group {
#Para este caso, exite uma configuracao de grupo unindo as duas interfaces eth0 e eth1; assim, caso uma delas seja trocada de ativa para backup, a outra tambem sera trocada. Neste grupo devem estar todas as interfaces do equipamento.
   fw-cluster-eth0
   fw-cluster-eth1
   }
}

#Comando executados quando os equipamentos trocarem de posicao master-backup.
notify_master   "/etc/conntrackd/primary-backup.sh primary"
notify_backup   "/etc/conntrackd/primary-backup.sh backup"
notify_fault    "/etc/conntrackd/primary-backup.sh fault"

vrrp_instance fw-cluster-eth0 {
#Definicao do estado inicial como master, caso seja instalado no equipamento secundário trocar para BACKUP
   state MASTER
   interface eth0
   virtual_router_id 20
#Definicao da prioridade inicial do no, este valor deve ser o maior de todos para que entre como MASTER, caso seja instalado no equipamento secundário trocar para um valor menor
   priority 100
   advert_int 1
   authentication {
      auth_type PASS
      auth_pass 1111
   }

# IP virtual do cluster
   virtual_ipaddress {
      192.168.199.120/24 brd 192.168.199.255 dev eth0
   }
   nopreempt
   garp_master_delay 1
}

vrrp_instance fw-cluster-eth1 {
#Definicao do estado inicial como master, caso seja instalado no equipamento secundário trocar para BACKUP
   state MASTER
   interface eth1
   virtual_router_id 30
#Definicao da prioridade inicial do no, este valor deve ser o maior de todos para que entre como MASTER, caso seja instalado no equipamento secundário trocar para um valor menor
   priority 100
   advert_int 1
   authentication {
      auth_type PASS
```

```
      auth_pass 1111
   }
# IP virtual do cluster
   virtual_ipaddress {
      192.168.1.10/24 brd 192.168.1.255 dev eth1
   }
   nopreempt
   garp_master_delay 1
}
```

Esta configuração foi replicada no equipamento secundário, com poucas alterações que já estão comentadas.

Segue abaixo *script* de sinalização do *conntrackd.* Este foi salvo mesma pasta aqui utilizada para exibição, */etc/conntrackd/*:

```
[root@fw01 ~]# cat /etc/conntrackd/primary-backup.sh
#!/bin/sh
#
# (C) 2006-2011 by Pablo Neira Ayuso <pablo@netfilter.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# Description:
#
# This is the script for primary-backup setups for keepalived
# (http://www.keepalived.org). You may adapt it to make it work with other
# high-availability managers.
#
# Do not forget to include the required modifications to your keepalived.conf
# file to invoke this script during keepalived's state transitions.
#
# Contributions to improve this script are welcome :).
#

CONNTRACKD_BIN=/usr/sbin/conntrackd
CONNTRACKD_LOCK=/var/lock/conntrack.lock
CONNTRACKD_CONFIG=/etc/conntrackd/conntrackd.conf

case "$1" in
  primary)
    #
    # commit the external cache into the kernel table
    #
    $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -c
    if [ $? -eq 1 ]
    then
        logger "ERROR: failed to invoke conntrackd -c"
    fi

    #
    # flush the internal and the external caches
    #
    $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -f
```

```
  if [ $? -eq 1 ]
  then
     logger "ERROR: failed to invoke conntrackd -f"
  fi

  #
  # resynchronize my internal cache to the kernel table
  #
  $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -R
  if [ $? -eq 1 ]
  then
     logger "ERROR: failed to invoke conntrackd -R"
  fi

  #
  # send a bulk update to backups
  #
  $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -B
  if [ $? -eq 1 ]
  then
     logger "ERROR: failed to invoke conntrackd -B"
  fi
  ;;
backup)
  #
  # is conntrackd running? request some statistics to check it
  #
  $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -s
  if [ $? -eq 1 ]
  then
     #
     # something's wrong, do we have a lock file?
     #
     if [ -f $CONNTRACKD_LOCK ]
     then
        logger "WARNING: conntrackd was not cleanly stopped."
        logger "If you suspect that it has crashed:"
        logger "1) Enable coredumps"
        logger "2) Try to reproduce the problem"
        logger "3) Post the coredump to netfilter-devel@vger.kernel.org"
        rm -f $CONNTRACKD_LOCK
     fi
     $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -d
     if [ $? -eq 1 ]
     then
        logger "ERROR: cannot launch conntrackd"
        exit 1
     fi
  fi
  #
  # shorten kernel conntrack timers to remove the zombie entries.
  #
  $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -t
  if [ $? -eq 1 ]
  then
     logger "ERROR: failed to invoke conntrackd -t"
  fi
```

```
    #
    # request resynchronization with master firewall replica (if any)
    # Note: this does nothing in the alarm approach.
    #
    $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -n
    if [ $? -eq 1 ]
    then
        logger "ERROR: failed to invoke conntrackd -n"
    fi
    ;;
  fault)
    #
    # shorten kernel conntrack timers to remove the zombie entries.
    #
    $CONNTRACKD_BIN -C $CONNTRACKD_CONFIG -t
    if [ $? -eq 1 ]
    then
        logger "ERROR: failed to invoke conntrackd -t"
    fi
    ;;
  *)
    logger "ERROR: unknown state transition"
    echo "Usage: primary-backup.sh {primary|backup|fault}"
    exit 1
    ;;
esac

exit 0

[root@fw01 ~]# cat /etc/init.d/conntrackd #script de inicializacao do conntrackd#
#!/bin/bash
#
# chkconfig:    345 31 69
# description:  Connection tracking daemon
#
### BEGIN INIT INFO
# Provides:          conntrackd
# Required-Start:    $syslog $network $remote_fs
# Required-Stop:     $syslog $network $remote_fs
# Default-Start:     3 4 5
# Default-Stop:      0 1 2 6
# Short-Description: Connection tracking daemon
# Description:       conntrackd: the connection tracking userspace daemon that can be used to
#       deploy highly available GNU/Linux firewalls and collect
#       statistics of the firewall use.
### END INIT INFO

CONNTRACKD_BIN=/usr/sbin/conntrackd

if [ -r /etc/rc.status ]
then
  source /etc/rc.status
  START="/sbin/startproc"
  STATUS="/sbin/checkproc"
  SUCCESS="echo \$rc_done"
  FAILURE="echo \$rc_failed"
else
  source /etc/rc.d/init.d/functions
```

```
  START="daemon"
  STATUS="status"
  SUCCESS="success; echo"
  FAILURE="failure; echo"
fi

RETVAL=0

case "$1" in
  start)
    echo -n "Starting conntrackd: "
    ${START} ${CONNTRACKD_BIN} -d > /dev/null 2>&1
    RETVAL=$?
    ;;
  stop)
    echo -n "Stopping conntrackd: "
    ${CONNTRACKD_BIN} -k > /dev/null 2>&1
    RETVAL=$?
    ;;
  restart)
    $0 stop
    $0 start
    exit $?
    ;;
  reload)
    echo -n "Reloading configuration of conntrackd: "
    killproc -HUP ${CONNTRACKD_BIN} > /dev/null 2>&1
    RETVAL=$?
    ;;
  status)
    echo -n "Status of conntrackd: "
    ${STATUS} ${CONNTRACKD_BIN} > /dev/null 2>&1
    RETVAL=$?
    ;;
  *)
    echo "Usage: $0 {start|stop|status|restart|reload}"
    exit 1
    ;;
esac
[ ${RETVAL} -eq 0 ] && eval "${SUCCESS}" || eval "${FAILURE}"
exit ${RETVAL}
```

3.2.24 Ajuste do SELinux

Para este trabalho, não foram consideradas as proteções de segurança do *SELinux.* Entretanto, é sabido que este nível de proteção traz grandes vantagens para o equipamento. Será feito um estudo futuro para que este recurso seja utilizado.

Para desativar o *SELinux*, foi editado o arquivo */etc/sysconfig/selinux* e alterado a entrada *enforcing* para *disabled.*

3.2.25 Ajuste de Serviços

Para que todos os serviços executados pelo equipamento sejam inicializados junto com o sistema operacional, foi preciso ativá-los, para tanto, seguem os comandos de ativação:

```
[root@fwl01 ~]# chkconfig mysqld on
[root@fwl01 ~]# chkconfig httpd on
[root@fwl01 ~]# chkconfig iptables on
[root@fwl01 ~]# chkconfig ip6tables off
[root@fwl01 ~]# chkconfig ntpd on
[root@fwl01 ~]# chkconfig squid on
[root@fwl01 ~]# chkconfig snmpd on
[root@fwl01 ~]# chkconfig qlproxy on
[root@fwl01 ~]# chkconfig postfix on
```

3.3 Testes

Através de estudos sobre procedimentos de testes e avaliação de performance de equipamentos de segurança (KADLECSIK; 2005)(SHETH; 2001) (HICKMAN; et al., 2003), este trabalho irá apresentar algumas métricas para futuras comparações (BRADNER; 1991). São elas:

- *Throughput*: máxima taxa de transferência de pacotes entre dois equipamentos sem que nenhum deles seja descartado ou perdido;
- Latência: intervalo de tempo medido entre o pacote entrar completamente e sair completamente do equipamento onde está sendo medido.

3.3.1 Throughput

- Ambiente: um servidor executando *iperf* configurado em um zona e 1 estação configurada dentro de outra zona executando *iperf* em modo cliente; o equipamento de segurança deverá fazer a ligação entre os dois; servidor e cliente estão ligados diretamente nas portas do equipamento.
- Teste:
  - No servidor foi executado o seguinte comando: *iperf -s* (o parâmetro -s indica que será o servidor)
  - No cliente foi executado o seguinte comando: *iperf -c IP_SERVIDOR -t 20*

(o parâmetro -c indica que será um cliente e é seguido do endereço do servidor, o parâmetro -t indica por quanto tempo o teste deverá ser executado)

- Variáveis: foram instalados variadas quantidades de regras no equipamento e propositalmente forçado o tráfego a percorrer esta lista inteira até que a última regra permitisse sua passagem; o número de regras instaladas foi variado entre 0 e 5000; foi feito o mesmo teste com tradução de endereços ativado.
- Propósito: a razão deste teste é encontrar o valor máximo de *throughput* para o equipamento e a verificação de como este valor pode variar em virtude da quantidade de regras instaladas e do uso de tradução de endereços IP.
- Resultados:

Para comparação, na ausência do equipamento, o valor medido foi de 820Mbit/s. Com o equipamento executando a ligação entre cliente e servidor, os seguintes valores foram obtidos:

NAT Ativo (1=Ativo 0=Inativo)	**Quant. Regras Configuradas**	**Valor Obtido (Mbit/s)**
0	0	702
0	1000	685
0	2000	669
0	3000	685
0	4000	633
0	5000	465
1	0	687
1	1000	673
1	2000	669
1	3000	670
1	4000	622
1	5000	464

3.3.2 Latência

- Ambiente: um computador configurado em um zona e outro configurado dentro de outra zona, o equipamento de segurança deverá fazer a ligação entre os dois; ambos estão ligados diretamente nas portas do equipamento.
- Teste: um computador deverá enviar pacotes de requisição ICMP para o outro esperando sua resposta ICMP; o comando utilizado será:
  - *ping -f -c 50000 192.168.1.2* (o parâmetro -f indica para o comando que execute um *flood*, ou seja, dispare quantos pings forem possíveis realizando uma inundação; o parâmetro -c indica que ele deverá executar até que a contagem de pings atinja 50000 requisições)
- Variáveis: foram instaladas variadas quantidades de regras no equipamento e propositalmente forçado o tráfego a percorrer esta lista inteira até que a última regra permitisse sua passagem; o número de regras instaladas foi variado entre 0 e 5000; foi feito o mesmo teste com tradução de endereços ativado.
- Propósito: a razão deste teste é analisar o tempo mínimo necessário para que a requisição ICMP atinja seu destino final atravessando o equipamento e a verificação de como este valor varia em relação ao número de regras configuradas no equipamento e ao uso de tradução de endereçamento IP.
- Resultados:

Como resultado, será avaliado o tempo médio de resposta do comando executado. Para comparação, será utilizado o tempo médio de resposta quando não temos o equipamento de segurança intermediando a conexão: 0,082ms. Os seguintes valores médios foram encontrados com equipamento executando a ligação entre cliente e servidor:

NAT Ativo (1=Ativo 0=Inativo)	**Quant. Regras Configuradas**	**Valor Obtido (Mbit/s)**
0	0	0,262ms
0	1000	0,287ms

0	2000	0,304ms
0	3000	0,349ms
0	4000	0,413ms
0	5000	0,489ms
1	0	0,264ms
1	1000	0,306ms
1	2000	0,367ms
1	3000	0,350ms
1	4000	0,426ms
1	5000	0,494ms

4 COMPARAÇÃO COM EQUIPAMENTOS PROPRIETÁRIOS DE MERCADO

O uso dos *softwares* livres pelos vários centros de informática vem crescendo com o passar dos anos. Os exemplos de uso aparecem em muitos casos: universidades, centros de pesquisas, governo federal do Brasil[44], Banco do Brasil[45] e outros.

O conceito de *software* livre[46] descreve uma implementação livre de apropriação material, ou seja, algo programado e criado para ser livre para o uso, cópia, modificação e distribuição mantida, apenas, a propriedade intelectual. Dessa forma, o nome do criador, individual ou coletivo, sempre deve ser mantido, nunca suprimido, e a cada recriação os novos contribuidores deverão ser colocados juntos aos criadores originais.

Pelo conceito de ser aberto para modificações e adaptações, o *software* livre não mantém seus usuários limitados ao programa, limitados às características que este apresenta. Ele os permite controlá-lo, ajustá-lo, acrescê-lo sempre, buscando, não apenas um melhor, mas, o perfeito ajuste às expectativas e demandas necessárias para um programa. Ao contrário dos licenciamentos proprietários, onde o conceito limita completamente o usuário ao produto final, ficando este dominado pelo instrumento, o *software* livre permite-se ser dominado por aquele que o usa.

Este trabalho, como mais um desses casos, vem demonstrar a viabilidade do uso do *software* livre aplicado à segurança da informação. Entretanto, para fiel demonstração deste objetivo, uma comparação em diversos aspectos será descrita. Para comparação de alguns pontos, serão utilizadas características de equipamentos proprietários para demonstração das possibilidades de implementação destas com uso de *software* livre como uma ferramenta de segurança.

---

44 Disponível em: <http://www.governoeletronico.gov.br/acoes-e-projetos/software-livre>. Acesso em: 10 fev. 2013

45 Disponível em: <http://softwarelivre.org/portal/geral/atm-linux-do-banco-do-brasil>. Acesso em 10 fev. 2013.

46 Disponível em: <http://www.gnu.org>/philosophy/free-sw.html>. Acesso em: 11 ago. 2012.

4.1 Funcionalidades

A lista de funcionalidades oferecidas pelo equipamento de segurança configurado com *software* livre deste trabalho é bastante abrangente graças aos vários programas que podem ser configurados em conjunto e a possibilidade que o administrador tem de pesquisá-los e manipulá-los.

Utilizando alguns equipamentos de mercado, uma lista de funcionalidades foi criada para comparação com o equipamento desenvolvido neste trabalho. Estas funcionalidades estão presentes nos atuais equipamentos e por este exato motivo, foram eleitas para compor este trabalho. O quadro comparativo da tabela 4.1 abaixo traz um simples panorama para ilustração e é composto pelos seguintes equipamentos:

1. **Juniper SSG 550M**[47]
2. **Aker UTM Firewall 6.1**[48]
3. **CheckPoint 12200 Series**[49]
4. **FortiGate 600C**[50]
5. **Proposta do trabalho**

47 Disponível em: <http://www.juniper.net/us/en/products-services/security/ssg-series/>. Acesso em: 20 set. 2012.
48 Disponível em: <http://www.aker.com.br/sites/default/fils/dt-akerFirewallUTM_hardwareEnterprise.pdf>. Acesso em: 22 set. 2012.
49 Disponível em: <http://www.checkpoint.com/products/downloads/datasheets/12200-appliance-datasheet.pdf>. Acesso em: 22 set. 2012.
50 Disponível em: <http://www.fotinet.com/sites/default/files/productdatasheets/FortiGate-600C.pdf>. Acesso em: 27 set. 2012.

EQUIPAMENTO / FUNCIONALIDADE	1	2	3	4	5
Filtro de Conteúdo *WEB*	SIM	SIM	SIM	SIM	SIM
Inspeção de pacote por *STRING*	SIM	SIM	SIM	SIM	SIM
Regras baseadas em Usuários Autenticados	SIM	SIM	SIM	SIM	NÃO
Inspeção de pacote na camada de aplicação	SIM	SIM	SIM	SIM	SIM
VPN para terceiros	SIM	SIM	SIM	SIM	SIM
Alta Disponibilidade de Hardware	SIM	SIM	SIM	SIM	SIM
Alto Desempenho e Disponibilidade de Porta (IEEE 802.3ad)	SIM	NÃO	SIM	SIM	SIM
Balanceamento de Carga	SIM	SIM	SIM	SIM	SIM
Análise Gráfica de Tráfego em tempo Real	SIM	SIM	SIM	SIM	SIM
Anti-Vírus Integrado em Tempo Real	SIM	SIM	SIM	SIM	SIM
Interface de Administração Unificada	SIM	SIM	SIM	SIM	SIM
Aplicação de Politicas por Zona de Segurança	SIM	NÃO	SIM	SIM	SIM
Conexão com Servidor de LOG	SIM	SIM	SIM	SIM	SIM

Tabela 4.1

Pela tabela acima relacionada, podemos perceber que os recursos mais explorados pelos equipamentos de mercado poderão ser disponibilizados pelo conjunto de *softwares* livres propostos por este trabalho.

A interface de administração do equipamento sugerido pelo trabalho é uma simples forma de configurar todos os programas a partir de um mesmo ponto de acesso, entretanto ainda é necessária a configuração isolada de cada programa para seu perfeito funcionamento.

Muitas funcionalidades dos equipamentos acima listados possuem licenciamento a parte do equipamento, ou seja, para usufruir de todas as funções listadas, mais de uma licença deve ser adquirida. É o caso da base de assinaturas de antivírus e da base de filtros de conteúdo *WEB*.

Uma funcionalidade diferenciada dos demais equipamentos é a

possibilidade de criações de regras específicas para usuários autenticados no equipamento. Essa autenticação pode se dar de algumas formas, como *captcha*, e permite ao administrador a criação e regras específicas por usuário sem a necessidade de associá-lo a alguma das demais características comuns às demais regras (endereço de origem e destino, porta, protocolo, horário). Entretanto, apesar de ainda não estar disponível, existe um grupo de pesquisa desenvolvendo esta funcionalidade para a comunidade[51] que, inclusive, pode ser instalada com a ressalva de não ser completamente estável. Este tipo de autenticação pode ser criada diretamente nas regras de filtragem de pacotes, diferente da autenticação para navegação em ambiente de *browser*, que é uma funcionalidade do *proxy* de acesso web e pode ser utilizada no equipamento proposto.

Este trabalho prevê que as funcionalidades descritas possam ser instaladas de maneira granular, ou seja, não é necessário configurá-las todas e algumas poderão, inclusive, ser instaladas em *hardwares* separados, prevendo usos mais específicos que poderão utilizar o parque computacional disponível de forma mais racional. As funcionalidades que poderão ser instaladas separadas são:

- o controle de logs do sistema em servidor de *syslog* externo;
- filtro de conteúdo web com cache *off-line* e antivírus integrado;
- VPN para usuários;
- histórico gráfico de uso de *links*;
- envio de alarmes por e-mail utilizando serviço interno ou externo;
- serviço de DHCP.

4.2 Desempenho

Os valores de desempenho medidos para o equipamento desenvolvido não puderam ser comparados aos equipamentos proprietários descritos por falta de acesso aos mesmos. Futuramente, estes números poderão ser comparados, quando os mesmos testes puderem ser desenvolvidos com todos os equipamentos.

51 Disponível em: <http://www.ufwi.org>. Acesso em: 16 set. 2012.

4.3 Desvantagens Consideradas

Assim como este trabalho vem apresentar uma série de vantagens do uso do *software* livre para elaboração do equipamento de segurança, também serão apresentadas algumas desvantagens que devem ser consideradas.

A questão do suporte ao usuário no que tange às aplicações é uma delas. Apesar de documentadas, as aplicações não tem garantias absolutas de funcionamentos, o que é uma impossibilidade matemática por não ser possível testar todas as configurações e ajustes existentes, ou mesmo de seu desenvolvimento contínuo. Caso o administrador encontre alguma falha de implementação, será preciso algum esforço para que esta seja levada até seus desenvolvedores e ainda esperar que estes a corrijam, sabendo que não há garantias. Também é preciso considerar que o fato de termos muitos *softwares* envolvidos, então maior será o cuidado necessário para atualização de cada um deles, visando que todos continuem funcionando perfeitamente integrados. Este trabalho sugere um ambiente de testes para que possa ser feita uma prévia verificação das atualizações.

Outra caraterística deste tipo de implementação é a necessidade de uma equipe bem qualificada para administração do equipamento. A equipe deverá se dividir no monitoramento do equipamento, da rede e dos usuários. Além de, constantemente, ter de estudar os *softwares* envolvidos, suas atualizações e novos elementos que venham a surgir. Logo, a empresa deverá estar pronta para manter um corpo técnico de grande nível.

4.4 Outras Propostas de Plataformas de Segurança com Software Livre

Atualmente, no mercado de equipamentos e *softwares* de segurança, já existem algumas propostas que foram desenvolvidas com *software* livre e são comercializadas com contratos de instalação e manutenção.

Estes foram desenvolvidos e são mantidos pelo esforço de uma equipe que trabalha para a comunidade aberta, pois disponibilizam produtos com licenciamento livre, com opção de contratação para desenvolvimentos específicos e suporte

técnico. Essa opção é exatamente uma boa tentativa de cobrir a desvantagem de não se ter garantias sobre os *softwares* e também supre a necessidade de se manter um corpo técnico local.

Podemos citar algumas empresas como *Vyatta*[52], a *Smoothwall*[53] e a *pfSense*[54] que oferecem produtos baseados em *softwares* livres.

Testes de eficiência desenvolvidos por outros trabalhos (BIANCO et al., 2005), demonstraram que o desempenho do encaminhamento de pacotes apresentado pelo equipamento configurado com *Vyatta* é muito bom comparado com os equipamentos proprietários de mercado. Testes de uso de recursos de classificação e filtragem de pacotes utilizando características da camada de aplicação (PEREIRA; RIBEIRO; CARVALHO, 2009) com *pfSense* demonstraram como o recurso pode ser explorada com sucesso pelos administradores utilizando uma interface de configuração de fácil manipulação.

52 Disponível em: <http://www.vyatta.com>. Acesso em: 10 set. 2012.
53 Disponível em: <http://www.smoothwall.org/about/>. Acesso em: 10 set. 2012.
54 Disponível em: <http://www.pfsense.org>. Acesso em: 12 set. 2012.

5 FUTUROS TRABALHOS

Futuramente, novos estudos poderão ser feitos visando os seguintes pontos:

- comparação dos custos de implementação dos produtos comerciais e os custos da implementação da solução livre;
- testes de eficiência da implementação das configurações, chegando aos números limites de capacidade do equipamento para um determinado *hardware;*
- desenvolvimento de uma interface única de configuração do equipamento, vislumbrando todos os elementos instalados;
- desenvolvimento de uma interface de monitoramento possibilitando um ponto único de visualização de todos os processos monitores;
- desenvolvimento de um manual mais específico a cerca da instalação e configuração dos elementos componentes do equipamento;
- desenvolvimento de uma linha de testes mais específicos, com objetivo de homologar todos os elementos envolvidos, ou seja, submeter o equipamento a testes ainda mais estressantes e que englobem, possivelmente, todos os *softwares* instalados;
- execução dos testes de desempenho com os equipamentos proprietários para comparação dos valores encontrados;
- criação de uma *script* de inicialização automatizada do *software Ntop;*
- estudo aprofundado do uso do *SELinux* dentro do equipamento;
- estudo para adaptação do equipamento para redes parão *IPV6.*

CONCLUSÃO

Este trabalho propõe uma forma de elaboração de um completo sistema de segurança de redes configurado unicamente com o uso de *softwares* livres. Em comparação com os equipamentos proprietários, quase todas as funcionalidades puderam ser implementadas com a associação de elementos de licenciamento livre dentro da solução, indicando que o *software* livre pode trazer soluções para as atuais necessidades que o mercado aponta para a segurança da informação. A única exceção é o uso de regras de filtragem com autenticação de usuário que está em desenvolvimento, como apontado nas pesquisas.

A grande quantidade de *softwares* utilizada para atender as necessidades não é uma exclusividade do mundo livre. Equipamentos proprietários também fazem uso de muitos *softwares* internos que são administrados via uma única interface que, inclusive, para algumas propostas, como da *Juniper Networks*[55] e da Checkpoint, pode ser licenciada separadamente do equipamento principal. Cabe aqui tornar claro que a administração dos *softwares* livres, componentes da solução proposta, terá de ser feita pontualmente pelo administrador para que o conjunto inteiro funcione de forma coerente. É sugerido um futuro desenvolvimento de uma interface centralizada de configurações.

Nesse estudo, foi testado um equipamento com todas as funcionalidades instaladas e configuradas de forma precisa, garantindo que é possível configuração para perfeita harmonia do produto final. Inclusive, é fundamental citar que, no período em que este trabalho foi desenvolvido, esta mesma solução foi implementada em dois escritórios particulares. Um deles possui uma rede de dados com cerca de 70 usuários, um servidor de FTP, um servidor de aplicação interna, um servidor HTTP, um servidor de autenticação e uma rede sem fio. Nesta implementação, todas as funcionalidades foram instaladas, com exceção do *backup* de *hardware* que não foi aprovada pelo cliente em questão.

Apesar de não terem sido abordados custos de implementação das soluções proprietárias, o que poderá ser feito em futuros trabalhos, pela dificuldade

55 Disponível em: <https://www.juniper.net/generate_license/>. Acesso em: 15 set. 2012.

de precificação em território nacional dos produtos importados, podemos inferir que parte do valor agregado dos produtos proprietários deverá ser convertida para a equipe que desenvolverá e manterá a solução livre. Além das notórias configurações iniciais exigidas para composição, também serão necessárias verificações de atualizações e possíveis ajustes com o decorrer do tempo. Cabe citar que este produto será mantido pelos investimentos intelectuais no corpo profissional e pelo esforço da comunidade que desenvolve os programas utilizados. Uma alternativa levantada pelo trabalho é a contratação de uma empresa de consultoria para o desenvolvimento do um produto específico às necessidades desejadas com contratação dos serviços de atualização e garantias de funcionamento. Com essa terceirização do negócio, que é a forma mais atual de utilização de *softwares* livres, também é possível a criação da ferramenta sem necessidade de corpo técnico próprio.

Finalmente, com os pontos levantados até aqui, este trabalho corrobora a viabilidade do uso de *software* livre para o desenvolvimento de equipamentos de segurança de redes de dados com alta performance e confiabilidade.

REFERÊNCIAS

ALMESBERGER, W. **Linux Network Traffic Control - Overview Implementation**; Escola Politécnica Federal de Lausana, Suíça; 23 de Abril de 1999; Disponível em: <http://www.almesberger.net/cv/papers/tcio8.pdf>. Acesso em: 11 jul. 2012.

ANDREASSON, O. **Iptables Tutorial** 1.2.2; Disponível em: <http://www.frozentux.net/iptables-tutorial/iptables-tutorial.html>. Acesso em: 07 jun. 2012.

AROCA, R. V.; TAVARES, D. M.; CAURIN, G.; **Scara Robot Controller Using Real Time Linux**; IEEE/ASME Internetional Conference on Avanced Intelligent Mechatronics; Zurique, 2007

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27001: Tecnologia da informação - Técnicas de segurança - Sistemas de gestão de segurança da informação - Requisitos.** 2006.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27002: Tecnologia da informação - Técnicas de segurança - Código de prática para a gestão de segurança da informação.** 2005.

BERGERON, P. Information resources management. **Annual Review of Information Science and Technology**, v. 31, p. 263-300, 1996.

BIANCO, A.; et al. **Open-Source Pc-Based *Software* Routers: A Viable Approach To High-Performance Packget Switching**; Lectures Notes in Computer Science, v. 3375, 2005.

BRADNER, S.; **RFC 1242 - Benchmarking Terminology for Network Interconnection Devices**; *Harvard University.* Julho; 1991

CAPRA, E.; FRANCALANCI, C.; MERLO, F.; LAMASTRA, C. **A Survey On Firm's Participation In Open Source Community Projects**; Computer Science, jun. 2009.

CASTELLS, M. **The Internet Galay Relfetions On The Internet, Business And Society**; tradução Maria Luiza X de A. Borges; Rio de Janeiro: Editora Jorge Zahar; 2003

CHESWICK, W. R.; BELLOVIN, S. M.; RUBIN, A. D. **Firewalls E Segurança Na Internet: Repelindo O Hacker Ardilosos**; tradução Edson Furmankiewicz; 2ª.ed.; Porto Alegre: Editora Bookman, 2005.

COMER, D. E. **Internetworking with TCP/IP**. 4. ed. New Jersey: Prentice Hall, 2000. v. 1: principles, protocols, and architectures. 750 p. 31, 34, 38, 40.

IEEE. **802.1 Q/D10, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks,** Copyright by the Institute of Electrical and Electronics Engineers, 1997.

INGHAM, K.; FORREST, S.; **A History and Survey of Network Firewalls**; Universidade do Novo México. Disponível em: http://agl.cs.unm.edu/~treport/tr/02-12/firewall.pdf. Acessado em: 10 fev. 2013.

FITZGERALD, B.; The transformation of open source software. **Mis Quarterly**, p. 587-598, 2006.

FULP, E. W. **Optimization of network firewall policies using ordered sets and directed acyclical graphs**; IEEE Internet Management Conference, 2005; Disponível em: http://www.cs.wfu.edu/~fulp/Papers/ewflist.pdf; Acesso em: 16 mai. 2013.

FOROUZAN, B., A. **Comunicação De Dados E Redes De Computadores**; 3ª ed., São Paulo: Editora Bookman, 2006.

GHEORGHE, L. **Designing And Implementing Linux Firewall And Qos Using Netfilter, Iproute2, Nat And L7-Filter**; Editora Packt; Outubro de 2006.

GUIJARRO, M.; GASPAR, R. Experience And Lessons Learnt From Running High Availability Databases On Network Attached Storage; **Journal of Physics**: Conference Series, v. 19, parte 4, 2008.

HALFOND, W. J. G.; ORSO, A. AMNESIA: Analysis and Monitoring for Neutralizing SQL-Injection Attacks; **Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering**; College of Computing Georgia Institute of Technology; 2005

HELDENBRAND, D.; CAREY, C. The Linux Router: An Inexpensive Alternative To

Commercial Routers In The Lab; **Journal of Computing Sciences in Colleges** - Papers of the Fourteenth Annual CCSC Midwestern Conference and Papers of the Sixteenth Annual CCSC Rocky Mountain Conference – v. 23 Issue 1; 2007

HICKMAN, B., et al.; **RFC 3511 - Benchmarking Methodology for Firewall Performance**; *The Internet Society. Abril;* 2003

HSUEH, C.; LIN, H.; HUANG, G. **Boosting Ethernet Using Regular Switching Hubs**, Journal of Information Science and Engineering, 2006.

HUNT C.; **TCP/IP Networking Administration**; 3ªed.; O'Reilly Media; 2002

KADLECSIK, J.; PÁSZTOR, G. **Netfilter Performance Testing** ; Disponível em: http://people.netfilter.org/kadlec/nftest.pdf. 2005. Acessado em: 23 jun. 2013.

KENYON, T. **High-Performance Data Network Designs: Design Technics And Tools** ; Editora Digital Press; 2002.

LAKHANI, K.; HIPPEL, E. How Open Source *Software* Works: “FREE” User-To-User Assistence, **Research Policy**, v. 32, jun. 2003.

MAJ, S. P.; MAKASIRANONDH, W.; VEAL, D. **An Evaluation Of Firewall Configuration Methods**; Jornal Internacional de Ciência da Computação e Segurança; v. 10; 2010.

MARMORSTEIN, R.; KEARNS, P. **Firewall Analysis With Policy-Based Host Classification**, 20th Large Installation System Administration Conference - LISA, Washington, DC: USENIX Association, p. 41 à 51; 2006. Disponível em: <http://static.usenix.org>/event/lisa06/tech/full_papers/marmorstein/marmorstein_html/>. Acesso em: 15 set. 2012.

MARTINS, A. B.; SANTOS, C. A. S. **Uma Metodologia Para Implantação De Um Sistema De Gestão De Segurança Da Informação**, Revista de Gestão da Tecnologia e Sistemas de Informação, v. 2, Número 2, 2005.

PEREIRA, H.; RIBEIRO, A. G.; CARVALHO, P. **Improving Traffic Classification And Policing At Application Layer,** IEEE Symposium on Computers and Communications, 2010. Disponível em: <http://hdl.handle.net/1822/17443>. Acesso em: 12 set. 2012;

PEREIRA, H.; RIBEIRO, A.; CARVALHO, P. **L7 Classification And Policing In The Pfsense Plataform**, Atas da CRC'2009 - 9ª Conferência sobre Redes de Computadores, IST - Taguspark, Oeiras, 2009.

POSTEL, J. **Internet control message protocol: DARPA Internet program, protocol specication. Request for Comments RFC 792**, 2002. Disponível em: <http://www.rfc-editor.org/rfc/rfc792.txt>. Acesso em: 30 out. 2012.

SHETH, C.; THAKKER, R.; **Performance Evaluation and Comparative Analysis of Network Firewalls**; 2011; Disponível em: http://jmiller.uaa.alaska.edu/cse465-fall2011/papers/sheth2011.pdf. Acessado em: 23 jun. 2013.

WELTE, H. **How To Replicate The Fire: Ha For Netfilter Based Firewalls**; Netfilter Core Team + Astaro AG; Proceeding of the Ottawa Linux Symposium; Ottawa, Ontario Canadá; 2002.

WOLL, A. **A Quantitative Study Of Firewall Configuration Erros**, Computer, v. 37, 2004.

WOLL, A. **The Use And Usability Of Direction-Based Filtering In Firewalls**; Computers & Security v. 23, Issue 6, set. de 2004, p. 459 à 468; Disponível em: <http://www.sciencedirect.com/science/journal/01674048>. Acesso em: 13 set. 2012.