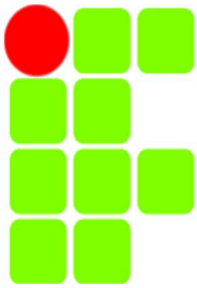

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL DE MINAS GERAIS
Câmpus Inconfidentes

**EVELYN MARIA VITOR**

# IMPLEMENTAÇÃO E COMPARAÇÃO DE FIREWALLS EM UM AMBIENTE CORPORATIVO: ACL'S X IPTABLES

**INCONFIDENTES-MG**

**2013**

**EVELYN MARIA VITOR**

# IMPLEMENTAÇÃO E COMPARAÇÃO DE FIREWALLS EM UM AMBIENTE CORPORATIVO: ACL'S X IPTABLES

Trabalho de Conclusão de Curso apresentado como pré-requisito de conclusão do curso de Graduação Tecnológica em Redes de Computadores no Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Câmpus Inconfidentes, para obtenção do título de Tecnólogo em Redes de Computadores.

Orientador: Prof. Esp. Vinícius Ferreira de Souza

**INCONFIDENTES-MG**

**2013**

**EVELYN MARIA VITOR**

# IMPLEMENTAÇÃO E COMPARAÇÃO DE FIREWALLS EM UM AMBIENTE CORPORATIVO: ACL'S X IPTABLES

Data de aprovação: ___ de _____2013

_____
Orientador: Prof. Esp. Vinícius Ferreira de Souza (IFSULDEMINAS – Câmpus Inconfidentes)

_____
Prof. Esp. Bruno Amarante Couto Rezende (IFSULDEMINAS – Câmpus Inconfidentes)

_____
Prof. Esp. Luiz Carlos Branquinho Caixeta Ferreira (IFSULDEMINAS – Câmpus Inconfidentes)

**Dedicatória**

Dedico este trabalho à minha mãe, grande mulher que batalhou sozinha na criação de seus filhos, que sempre me incentivou a estudar e me apoiou durante todo o curso principalmente nos momentos de cansaço e estresse, não deixando que eu desistisse. Espero que este trabalho e a conclusão deste curso sejam motivos de orgulho.

**Agradecimentos**

Primeiramente agradeço a Deus pelo dom da vida, por ter me dado a chance de concluir este curso superior que abrirá grandes portas para o meu futuro e que também me fez amadurecer em conhecimento, profissionalismo e em pessoa. Obrigada por ter permitido que eu fizesse grandes amizades durante estes 3 anos e meio de convivência e principalmente pela força nos momentos de desânimo.

Ao meu querido orientador e amigo Vinícius, ótima pessoa e um grande profissional, o meu muito obrigada, pelo apoio, perseverança e paciência não apenas na confecção deste trabalho, mas em todo o curso. Que Deus lhe abençoe e ilumine sempre a sua caminhada.

Fernando, Claiton, Marcio, José Luiz, Alessandro, Bruno, Carlos, Douglas, André, Leonardo, Josaphat e Mauro, agradeço pelo companheirismo de vocês nos estudos, nas cervejas e pela paciência durante estes anos. Saibam que considero todos vocês grandes amigos meus, infelizmente agora será difícil reencontrarmos já que cada um seguirá o seu caminho, mas sempre os considerarei e estarei disponível quando precisarem. Meninos, obrigada por tudo e desculpe se alguma vez os magoei, desejo tudo de bom a vocês queridos.

Muito obrigada ao meu grande amigo Henrique, por sempre ter me incentivado mesmo estando longe e por ter me ajudado na edição do Abstract.

Enfim, agradeço a todos colegas, amigos e professores pelo apoio e amizade durante todo o curso.

**Resumo**

A internet se torna cada vez mais um ambiente inseguro devido à facilidade de acesso e o aumento do número de usuários conectados a ela. Este fato e a constituição de ambientes cooperativos nas organizações levam a uma crescente preocupação quanto à segurança. Para coibir os usuários de utilizarem recursos tecnológicos que podem ser a porta de entrada de algum *cracker*, vírus, ou qualquer outra forma de ameaça à segurança da informação são necessários alguns mecanismos, como a política de segurança e o *firewall*, onde a política implementa as diretrizes, normas e procedimentos da organização e o *firewall* implementa implicitamente e de forma transparente para o usuário o que a política de segurança da informação definiu. O *firewall* é um componente ou um conjunto de componentes, entre duas ou mais redes, por onde passa todo o tráfego, permitindo o controle, autenticidade e registros de todo o tráfego, esse mecanismo é utilizado para proteger geralmente uma rede interna confiável de uma rede pública não confiável. É também definido como um sistema ou grupo de sistemas, que visa o fortalecimento das políticas de controle de acesso entre duas redes distintas, o que reflete na implementação de uma política de segurança. Este trabalho realiza um estudo sobre as características, conceitos, tipos, e importância do *firewall* na infraestrutura de segurança de uma organização. Apresenta alguns modelos de *firewalls* que podem ser implementados em organizações, mostrando uma abordagem teórica e técnica sobre a implementação e implantação corretas desses *firewalls*, além de comparar o funcionamento e configuração das duas tecnologias apresentadas. Neste trabalho também é apresentado o recurso Política de Segurança, mostrando sua importância e a maneira correta para criar este documento.

## Abstract

The Internet becomes an increasingly insecure environment due to the easy access and the increasing number of users connected to it. This fact and the establishment of cooperative environments in organizations lead to a growing safety concern. To inhibit users from using technological resources that can be the gateway to some cracker, virus, or any other form of threat to information security, some mechanisms as security policy and firewall are needed. Security policy implements the guidelines, rules and procedures of the organization; while the firewall implements implicitly and transparently to the user what the information security policy defined. The firewall is a component or set of components between two or more networks, through which passes all traffic, allowing control, authenticity and all traffic records. This mechanism is usually used to protect an internal trusted network from a public network not reliable. It is also defined as a system or group of systems, aimed at strengthening political control access between two separate networks, which reflects the implementation of a security policy. This paper conducts a study on the features, concepts, types, and importance of firewall in the security infrastructure of an organization. It presents some models of firewalls that can be implemented in organizations, showing theoretical and technical approaches about the correct implementation and deployment of these firewalls, besides comparing the operation and configuration of the two technologies presented. This paper also presents the feature Security Policy, showing its importance and the correct way to create this document.

## ÍNDICE DE FIGURAS

## ÍNDICE DE TABELAS

## ÍNDICE DE QUADROS

**LISTA DE ABREVIATURAS**

ACK - *Acknowledge*

ACL – *Access Control List*

DDOS - *Distributed Denied of Service*

DMZ - *Demilitarized Zone*

DNS - *Domain Name System*

DOS – *Denied of Service*

FTP - *File Transfer Protocol*

HTTP - *Hypertext Transfer Protocol*

HTTPS – *Secure HyperText Transfer Protocol*

ICMP - *Internet Control Message Protocol*

IDS – *Intrusion Detection System*

IOS - *Internetwork Operational System*

IP - *Internet Protocol*

LAN - *Local Area Network*

NAT - *Network Address Translation*

*POP3 - Post Office Protocol*

RPC - Remote Procedure Call

SMTP - *Simple Mail Transfer Protocol*

SSH - *Secure Shell*

SYN - *Synchronize*

TCP - *Transfer Control Protocol*

TFTP - Trivial File Transfer Protocol

TI – Tecnologia da Informação

UDP - *User Datagram Protocol*

*URL - Uniform Resource Locator*

VLAN - Virtual Local Area Network

VPN – *Virtual Private Network*

WAN - W*ide Area Network*

**SUMÁRIO**

# 1 INTRODUÇÃO

A globalização não afetou somente as barreiras continentais ou governamentais, a constante evolução da *Internet* fez com que as informações ficassem vulneráveis a grupos mal intencionados que desejam conseguir informações classificadas como privilegiadas, seja como fim a espionagem industrial, informações estratégicas de empresas e pessoas, fraudes, escândalos e assim por diante. O fato das informações serem enviadas, disponibilizadas, compartilhadas, consultadas e geradas a todo instante e na maioria das vezes em tempo real, permite que as mesmas sejam rastreadas, se os devidos cuidados não forem tomados, o que pode trazer grandes prejuízos às organizações.

No meio tecnológico é comum os especialistas em segurança das empresas se preocuparem apenas com as vulnerabilidades ligadas a tecnologia como problemas com *hardware* e *software*, porém, em segurança da informação o especialista deve se preocupar com todos os tipos de vulnerabilidades que possam vir a comprometer os negócios da empresa.

À medida que novas vulnerabilidades de segurança são exploradas surgem novos mecanismos para tentar diminuir as possibilidades de ataque. Os antivírus, por exemplo, são constantemente atualizados porque outros vírus são desenvolvidos a todo o momento. Estes mecanismos abrangem desde proteção a uma única máquina, como toda uma rede. Um destes mecanismos de segurança é o *firewall*, que interage com os usuários de forma transparente, permitindo ou não o tráfego da rede interna para a *Internet*, como também da*Internet* para o acesso a qualquer serviço que se encontre na rede interna da corporação.

Dessa forma, todo o tráfego, deve passar por este “controlador” que aplica as regras que foram baseadas nas políticas de segurança adotadas pela empresa.

Embora não seja o único dispositivo de segurança de rede, o *firewall* é um dos mais importantes, sendo essencial na infraestrutura de segurança de qualquer organização.

Neste trabalho, é feita uma comparação entre dois tipos de tecnologias de *firewall*, visando apresentar as características importantes, vantagens e desvantagens, ajudando portanto a descobrir qual produto é mais conveniente, dependendo dos requisitos de segurança desejados.

## 1.1 Objetivos

- **Objetivos Principais**
  - Editar e aplicar as regras do Iptables em um sistema Linux;
  - Configurar o roteador cisco com as ACL'S;
  - Exibir a quantidade de regras que foram utilizadas em cada tecnologia para alcançar o mesmo objetivo;
  - Verificar se é possível aplicar as regras estabelecidas na política de segurança nos dois tipos de arquitetura;
  - Testar o funcionamento dos diferentes tipos de implementação;
  - Mostrar as vantagens e desvantagens de cada opção e apresentar a forma mais adequada de se implementar um *firewall* em um ambiente corporativo.

- **Objetivos Específicos**

Apresentar os conceitos e componentes relacionados à segurança da informação, apresentar o significado e a relevância da política de segurança e por fim, mostrar a importância, os tipos de *firewall* e as suas principais características.

## 1.2 Organização do Trabalho

Este trabalho está dividido em sete capítulos:

O primeiro capítulo de introdução contextualiza o problema, evidencia os objetivos gerais e específicos, demonstra justificativas, e destaca a importância do tema deste trabalho.

O capítulo 2 mostra um pequeno resumo dos artigos e trabalhos consultados que ajudaram na confecção deste projeto.

O terceiro capítulo aborda os conceitos da Segurança da Informação, demonstrando sua necessidade, que se torna cada vez maior com a evolução da *Internet*, principalmente em ambientes corporativos. Apresenta os quesitos que devem ser observados para traçar os objetivos de segurança e as principais técnicas de ataques que são utilizadas pelos invasores. Por fim, traz o conceito de Política de Segurança, os benefícios que ela traz e como deve ser planejada.

No quarto capítulo o tema do trabalho é apresentado detalhadamente, mostrando os tipos de *firewall* existentes, as funcionalidades além do filtro de pacotes que ele possui, e as opções de arquitetura que o *firewall* pode assumir dentro da infraestrutura da rede.

O quinto capítulo descreve sobre ACL, que é uma das tecnologias de *firewall* que será comparada neste projeto. Serão mostradas as características principais, como o funcionamento; os tipos de ACL que existem; a forma que as regras devem ser editadas, fala ainda sobre a necessidade de fazer um *backup* da configuração do roteador e sobre como adotar alguns procedimentos para minimizar o impacto que as listas de controle de acesso causam no desempenho do *hardware*.

No sexto capítulo é apresentado o *iptables, firewall* do Sistema Operacional Linux, que também é uma tecnologia que será comparada neste trabalho. Serão discutidos aspectos de sua estrutura, instalação, funcionamento, criação de regras, erros cometidos pelos administradores, além de mostrar alguns exemplos de regras.

Os procedimentos da implementação do projeto são mostrados no sétimo capítulo. Serão expostos os componentes dos *hardwares* em que foram implementados os *firewalls*; a relação de computadores e ips; a topologia da rede; a política de segurança, as regras que foram implementadas em cada arquitetura, os *logs* de acessos e o desempenho das máquinas após as implementações.

Finalmente, no capítulo 8 são apresentadas as considerações finais, vantagens e desvantagens das tecnologias comparadas e conclui-se que o iptables é o *firewall* a nível de pacotes mais eficiente e estável para o cenário apresentado nesse trabalho.

# 2 ESTADO DA ARTE

Com o avanço da tecnologia tornou-se indispensável que as empresas possuam uma estrutura tanto para prover a seus funcionários acesso à *Internet*, como para fornecer acesso externo as suas aplicações para seus parceiros e clientes. Com os inúmeros benefícios que este acesso proporciona, vieram também inúmeros problemas, principalmente no que diz respeito à segurança dos dados.

Embora não seja o único dispositivo de segurança de rede, o *firewall* é um dos mais importantes, sendo essencial na infraestrutura de segurança de qualquer organização.

Neste capítulo é apresentada uma descrição sucinta dos principais artigos e trabalhos consultados, que motivaram o desenvolvimento e ajudaram na produção deste projeto.

## 2.1 Trabalhos Relevantes

ROCHA JUNIOR [2010], apresenta alguns tipos de *firewalls* corporativos, mostrando vantagens e desvantagens, que proporcionam ao administrador de rede um número maior de opções quanto à escolha de qual tecnologia utilizar na infraestrutura da empresa. Expõe uma forma correta de implementação e implantação do *firewall iptables* de acordo com a infraestrutura da empresa fictícia possuía e com as necessidades que ela apresentava.

ESQUIVEL [2006 ] mostra em seu trabalho que a criação e configuração de regras para o *Firewall Iptables* é ainda uma tarefa que exige dos administradores de rede um esforço e conhecimento da ferramenta avançados. Além disso, apresenta os conceitos relacionados a esta tecnologia, e propõe uma ferramenta gráfica de interface amigável e de fácil utilização que pretende facilitar a criação e manutenção das regras.

SANTOS [2007] destaca que para especificar o melhor tipo de *Firewall*, requer um breve conhecimento do cenário onde este irá atuar, definir quais são os riscos e falhas neste ambiente, qual o perfil de tráfego de dados e qual nível de segurança deve ser implementado. Faz análises comparativas dos tipos de dados, demonstradas em gráficos e conclui que não existe no mercado o melhor *Firewall*, o que se propõe são *Firewalls* de *software* e *hardwares*, que podem ser testados por ferramentas que definem o grau de segurança para sua utilização.

PEREIRA [2012] explica o que é uma ACL, faz uma descrição geral de quais são os seus tipos e funcionalidades apresenta alguns exemplos simples de uso de cada uma das acl's.

SPANCESKI [2004] aborda um estudo sobre política de segurança da informação que é uma das principais medidas de segurança adotadas pelas organizações com o objetivo de garantir a segurança da informação. Destaca algumas metodologias e melhores práticas em segurança da informação, as principais dificuldades para criação e implementação, os princípios de segurança da informação e a necessidade de envolvimento de toda a organização.

# 3 SEGURANÇA DA INFORMAÇÃO E SEUS PRINCIPAIS COMPONENTES

Neste capítulo serão destacados a importância da segurança da informação e os principais conceitos relacionados a ela.

## 3.1 A Segurança da Informação

A necessidade de segurança é um fato que vem transcendendo o limite da produtividade e da funcionalidade. Enquanto a velocidade e a eficiência em todos os processos de negócios significam uma vantagem competitiva, a falta de segurança nos meios que habilitam a velocidade e a eficiência pode resultar em grandes prejuízos e falta de oportunidades de negócios. (NAKAMURA ; GEUS, 2003, p.9).

Segundo Dias (2000), na sociedade da informação, ao mesmo tempo que as informações são consideradas o principal patrimônio de uma organização, elas estão também sob constante risco como nunca estiveram antes. Com isso, a segurança de informações tornou-se um ponto crucial para a sobrevivência das instituições.

O mundo da segurança é marcado pela evolução contínua, no qual novos ataques têm como resposta novas formas de proteção que levam ao desenvolvimento de novas técnicas de ataques e assim sucessivamente. Esse mesmo comportamento pode ser observado no mundo da informação, onde também se deve ter em mente que a segurança deve ser contínua e evolutiva. (NAKAMURA ; GEUS, 2003, p.9).

Os seguintes fatores justificam a preocupação com a segurança contínua: a natureza dos ataques, as novas vulnerabilidades das novas tecnologias, a criação de novas formas de ataques, o aumento da conectividade, a complexidade da defesa, o

aumento dos crimes digitais e os grandes prejuízos ocasionados pela falta de segurança. (NAKAMURA ; GEUS, 2003, p.10).

O ambiente corporativo é um ambiente que integra diversos sistemas de diferentes organizações. A rede é a tecnologia utilizada para realizar essa integração, permitindo conexões entre todos os seus elementos. A confiabilidade, integridade e disponibilidade da rede são essenciais para o próprio negócio da organização, justificando a preocupação com a segurança das informações.

## 3.2 Objetivos de segurança

Segundo Dias (2000), para identificar os objetivos prioritários para uma determinada organização é essencial fazer uma análise da natureza das aplicações, dos riscos e impactos prováveis em caso de falha de segurança. Para traçar esses objetivos os seguintes quesitos devem ser observados:

a) confidencialidade ou privacidade: proteger as informações contra o acesso de qualquer pessoa não explicitamente autorizada pelo dono da informação;

b) integridade de dados: evitar que dados sejam excluídos ou de alguma forma alterados, sem a permissão do proprietário da informação;

c) disponibilidade: proteger os serviços de informática de tal forma que não sejam degradados ou fiquem indisponíveis sem a devida autorização. As medidas relacionadas a esse objetivo, podem ser a duplicação de equipamentos ou *backup*;

d) confiabilidade: garantir que, mesmo em condições adversas, o sistema atuará conforme o esperado.

e) consistência: certificar-se de que o sistema atua de acordo com as expectativas dos usuários autorizados;

## 3.3 Fundamentos da Segurança da Informação

Esta seção tem por objetivo apresentar alguns conceitos importantes na segurança da informação, como: ativo, ameaça, vulnerabilidade, risco, impacto e incidente.

### 3.3.1 *Ativo*

Segundo SÊMOLA (2003 p.45) um ativo é: “todo elemento que compõe os processos que manipulam e processam a informação”. Podemos considerar ativo, algo de valor para a organização como: *software* e *hardware*, pessoas, instalações.

Para um melhor gerenciamento da segurança da informação os ativos devem ser divididos e classificados de acordo com o seu grau de importância e criticidade para a organização. (SÊMOLA , 2003)

### 3.3.2 *Ameaça*

Podemos definir ameaça como sendo agentes ou condições que exploram as vulnerabilidades, causando assim incidentes aos ativos e comprometendo o negócio da organização. (SÊMOLA , 2003)

No meio tecnológico é comum os especialistas em segurança das empresas se preocuparem apenas com ameaças e pragas virtuais, porém, em segurança da informação o especialista deve se preocupar com todos os tipos de ameaças que possam vir a comprometer o negócio da empresa. (SÊMOLA , 2003)

### 3.3.3 *Vulnerabilidade*

Podemos definir vulnerabilidade como sendo uma falha existente em um ou mais ativos que pode ou não ser explorada por uma ameaça. As vulnerabilidades por si só não causam incidentes, elas apenas são brechas para que uma possível ameaça os cause.
De acordo com SÊMOLA (2003) esses são alguns exemplos de vulnerabilidades:

- Naturais: incêndios, enchentes, terremotos, aumento de umidade e temperatura;
- Físicas: Instalações fora do padrão, falta de equipamentos anti-incêndios;
- Hardware: Desgaste natural das placas, erros durante a instalação;
- Software: Erros na instalação e configuração podem facilitar acessos indevidos;
- Mídias: Discos, fitas podem ser perdidos ou danificados;
- Comunicação: Acesso não autorizado ou perda da comunicação;
- Humanas: Falta de treinamento, compartilhamento de informações confidenciais, sabotagens.

### 3.3.4 *Risco*

Segundo SÊMOLA (2003 p.50) risco é: “Probabilidade de ameaças explorarem vulnerabilidades, provocando perdas de confidencialidade, integridade e disponibilidade, causando impactos nos negócios.”

O risco é calculado com base na relação entre a probabilidade de um ataque acontecer e o impacto que ele causará para a organização, com base nesse cálculo é que as empresas irão identificar quais os pontos que necessitam de um maior investimento na segurança da informação. (SÊMOLA , 2003)

### 3.3.5 *Impacto*

Segundo SÊMOLA (2003 p.50) impacto é: “Abrangência dos danos causados por um incidente de segurança sobre um ou mais processos do negócio”.

O impacto causado a um ativo de baixa importância não é o mesmo causado a um de alta importância, por isso vem a necessidade de classificar os ativos de acordo com o grau de importância para a organização para montar uma melhor estratégia com o objetivo de diminuir o impacto causado pelos incidentes caso eles venham a acontecer. (SÊMOLA , 2003)

### 3.3.6 *Incidente*

Um incidente caracteriza-se por uma violação ao ativo da organização, causado por uma ou mais ameaças que explorou uma ou mais vulnerabilidade, levando a perda dos princípios básicos da segurança da informação: confiabilidade, integridade e disponibilidade, trazendo assim impactos negativos ao negócio da empresa. (ROCHA JUNIOR, 2010)

O objetivo principal da segurança da informação é fazer com que esses incidentes não ocorram, e caso venham a acontecer, que já exista na organização uma estratégia para garantir a continuidade do negócio e minimizar o dano causado pelo incidente. (ROCHA JUNIOR, 2010)

## 3.4 Técnicas de Ataques

Este tópico irá apresentar algumas das principais técnicas de ataque que os invasores utilizam, como: Engenharia social, *packetsniffing*, *scanning de vulnerabilidades, DoS, IP spoofing, buffer overflow, smurf* e pragas virtuais.

### 3.4.1 *Engenharia social*

Muitos especialistas em segurança preocupam-se apenas em atualizar os *patches* de segurança das aplicações, utilizar as ferramentas mais modernas para proteção de rede, utilizar regras rígidas no *firewall*. Mas se esquecem de outro item capaz de comprometer a segurança da informação de uma organização, o ser humano. (NAKAMURA E GEUS, 2012)

O ataque de engenharia social consiste em explorar a fraqueza humana para a obtenção de informação. Um bom exemplo deste tipo de ataque é o invasor se fazer passar por um alto funcionário da empresa e procurar obter informações como: senhas, usuários de acesso, informações sigilosas. (NAKAMURA E GEUS, 2012)

A principal forma de se proteger contra esse tipo de ataque é realizando campanhas educativas com todos os funcionários da empresa, o uso de uma política de segurança centralizada e bem divulgada também ajuda na prevenção deste ataque. (NAKAMURA E GEUS, 2012)

### 3.4.2 *Packet Sniffing*

O ataque *packet sniffing* consiste em uma técnica que o atacante utiliza para obtenção de informações através da captura do tráfego de rede. Para o invasor fazer uso desse tipo de ataque ele tem que ter acesso a rede, com uso de *softwares* como *wireshark* e *tcpdump2* o invasor coleta o tráfego de rede, obtendo assim informações valiosas como: senhas, *e-mails*, e até mesmo conversas realizadas por *softwares* de troca de mensagem. (NAKAMURA e GEUS, 2012)

Segundo NAKAMURA E GEUS (2012) para o administrador de redes proteger a rede contra esse tipo de ataque ele pode fazer o uso dos seguintes recursos:

- Fazer uso de *switchs* ao invés de *hubs*: Com o uso de *switch* o tráfego é direcionado para apenas a porta de destino e não em *broadcast* como no uso de *hubs;*

- Utilizar VLANs : A utilização de VLANs aumenta a segurança da rede, pois ela ficará mais segmentada;
- Utilização de criptografia: Com o uso de criptografia mesmo que o atacante colete o tráfego da rede, ele não vai conseguir entendê-los.

### 3.4.3 *Port Scanning*

O *port scanning* é uma técnica que consiste em o atacante obter informações sobre os serviços que estão acessíveis através do *scanning* das portas TCP e UDP conhecidas.

O nmap é um dos port *scanning* mais conhecidos e utilizados, com ele o invasor é capaz de descobrir os serviços que estão sendo utilizados por uma determinada máquina, além do sistema operacional que ela roda. (NAKAMURA E GEUS, 2012)

Ainda segundo NAKAMURA e GEUS (2012) para detectar que a rede está recebendo um *port scanning* é necessário o uso de um *Intrusion Detection System* (IDS), porém, algumas técnicas como *slow scan*, que consiste em um *scanning* de portas de forma lenta e o *random port scan*, que consiste em um *scanning* de portas de forma aleatória, podem enganar o IDS.

### 3.4.4 *Scanning de vulnerabilidades*

O *scanning* de vulnerabilidades é uma técnica que o atacante utiliza para obter informações sobre as vulnerabilidades nos serviços que estão sendo executados na máquina. (NAKAMURA E GEUS, 2012)

Abaixo serão listados alguns riscos que são detectados com os scanners de vulnerabilidades de acordo com NAKAMURA e GEUS (2012):

- Compartilhamentos de arquivos que não são protegidos por senhas;
- *Software* desatualizado;
- Configurações erradas em serviços;
- Falhas no nível de protocolos.

Os *scanners* de vulnerabilidades não são utilizados apenas por invasores, muitos administradores de rede e segurança fazem uso deles para realizar uma auditoria na rede, detectando os pontos vulneráveis e assim corrigindo-os. (NAKAMURA E GEUS, 2012)

### 3.4.5 *IP spoofing*

O *IP spoofing* é uma técnica com a qual o atacante mascara o endereço IP real que ele utiliza. Segundo NAKAMURA e GEUS (2012), essa técnica é muito utilizada em tentativas de acesso a sistemas nos quais a autenticação tem como base o endereço IP.

As organizações podem se proteger do ataque *IP spoofing* com o uso de filtros de acordo com as interfaces de rede. (NAKAMURA E GEUS, 2012)

Apesar de o referido ataque ter sido apresentado nos tipos de ataques referente a obtenção de informações, o *IP spoofing* pode ser enquadrado também nos tipos de ataque de negação de serviço, que serão apresentados logo a seguir. (NAKAMURA E GEUS, 2012)

### 3.4.6 *Ataques de Negação de Serviço*

Segundo ROCHA JUNIOR (2010), os ataques baseados na negação de serviço, também conhecidos como DoS, consistem em deixar o serviço oferecido por um servidor indisponível. São utilizadas técnicas com o objetivo de sobrecarregar a rede ou o servidor a tal ponto do serviço ficar indisponível para os usuários que desejam acessá-lo. Os principais ataques de negação de serviço são: *SYN flooding* e o *Smurf*.

Uma variação do ataque de negação de serviços é o DDoS – *Distributed Denial of Service* (ataque distribuído de negação de serviço). O principio é o mesmo utilizado pelo ataque de negação de serviço, a diferença é que este tipo de ataque é realizado por um grupo de máquinas. (ROCHA JUNIOR, 2010)

A seguir os ataques *SYN flooding* e o *Smurf* serão apresentados de forma mais detalhada.

#### 3.4.6.1 SYN Flooding

O *SYN flooding* é uma técnica com o qual o atacante explora o estabelecimento de conexões TCP. Este ataque consiste no envio de um grande número de solicitações de conexão (Pacotes *SYN*), de forma que o servidor não consegue responder a todas elas, a pilha de memória sofre então um *overflow* e as requisições dos usuários legítimos são, então, rejeitadas. (NAKAMURA e GEUS, 2012)

Alguns autores como (URUBATAN NETO, 2004) sugerem que uma forma de se defender de ataques de *SYN flood* é com a utilização do módulo *limit* do *firewall iptables*,

ele atua limitando as conexões de acordo com o tempo estabelecido na criação da regra do *firewall*. Na opinião do autor deste trabalho o uso do módulo *limit* para se defender de ataques de *SYN flood* é equivocada, este módulo não é capaz de distinguir se a conexão *SYN* está sendo realizada por um atacante ou por uma pessoa que está simplesmente querendo acessar o sistema. O uso desse módulo para limitar as conexões *SYN* pode, portanto, prejudicar os usuários comuns que estão querendo acessar o sistema.

#### 3.4.6.2 Smurf

O ataque *smurf* consiste em enviar uma grande quantidade de pacotes *Internet Control Message Protocol* (ICMP) *echo request* para toda rede, ou seja, em *broadcast*, tendo o endereço IP da vítima como endereço de origem dessa requisição. A rede toda irá responder essa requisição com o ICMP *echo replay* deixando assim a máquina da vítima indisponível. (NAKAMURA e GEUS, 2012)

Uma forma de se defender desse tipo de ataque é configurar o roteador da rede para rejeitar pacotes do tipo *broadcast*, tal configuração já vem realizada por *default* em muitos roteadores. (NAKAMURA e GEUS, 2012)

### 3.4.7 *Ataques no Nível da Aplicação*

Os ataques no nível de aplicação exploram as vulnerabilidades presentes nas aplicações. Nesta seção serão apresentados alguns dos mais relevantes ataques no nível de aplicação, a saber: *buffer overflow*, pragas virtuais (vírus, *worms* e cavalos de tróia).

#### 3.4.7.1 Buffer Overflow

Segundo NAKAMURA e GEUS (2012), no ataque do tipo *buffer overflow* o invasor explora vulnerabilidades no código das aplicações, no qual o controle da memória temporária para armazenamento de dados (*buffer*) não é realizado de maneira adequada.

O principal objetivo desse tipo de ataque é conseguir invadir e controlar o programa que possua essa vulnerabilidade e posteriormente assumir o controle do sistema operacional. (NAKAMURA e GEUS, 2012)

Esta forma de ataque não possui uma defesa específica, sendo muito difícil inclusive a sua detecção. A defesa contra este tipo de ataque é reativa, assim que a tentativa de ataque *buffer overflow* for detectada pelo administrador de rede, ele deve

comunicar aos desenvolvedores da aplicação que deverão criar *patches* de atualização para corrigir a falha. (NAKAMURA e GEUS, 2012)

#### 3.4.7.2 Vírus, Worms e Cavalos de Tróia

Os vírus, *worms* e cavalos de tróia constituem uma ameaça cada vez mais constante e danosa nas organizações. Muitos administradores de rede e analistas de segurança tratam, erroneamente, essas pragas, e outras que não foram citadas, apenas como vírus. Cada tipo de praga se comporta de uma maneira diferente, causando danos diferentes, por isso que elas devem ser tratadas de forma separadas. A seguir serão descritos as principais pragas virtuais, a saber: vírus, *worm*, cavalo de tróia. (NAKAMURA e GEUS, 2012)

- Vírus: Podemos definir vírus como um programa de computador que se anexa a outros programas danificando-os a medida que vão se espalhando, um detalhe importe e que diferencia os vírus de outras pragas, como *worms*, é que para um vírus infectar um sistema ele necessita ser executado;
- Worms: Podemos definir um *worm* como sendo um programa capaz de se propagar automaticamente na rede, causando malefícios a mesma, enviando copias de si mesmo de computador para computador. O que diferencia o *worm* de um vírus é que ele não adiciona uma cópia de si mesmo a outros programas e ele não necessita ser executado para infectar um sistema;
- Cavalo de Tróia: Podemos definir um cavalo de tróia como sendo um programa que além de realizar as funções normais para que ele foi criado, realiza também funções maliciosas como deixar portas abertas. O que diferencia o cavalo de tróia de vírus e *worms* é que o cavalo de tróia não se propaga na rede.

Para se defender deste tipo de ataque é necessário o uso de *software* antivírus atualizado. Para as empresas é recomendado que elas utilizem as soluções corporativas de antivírus, pois elas proporcionam um maior controle, por exemplo, *scanners* programados, relatórios de máquinas que estão com as vacinas desatualizadas, se foi detectado alguma praga. (NAKAMURA e GEUS, 2012)

## 3.5 Política de Segurança

Podemos definir uma política de segurança como sendo um documento formal composto de normas e procedimentos que devem ser seguidos por todos (clientes, parceiros, funcionários e fornecedores) que fazem uso da informação na organização. Ela trata de aspectos humanos, culturais, tecnológicos, de uma organização. A política de segurança tem como principal objetivo definir padrões de comportamento que sejam largamente informados e conhecidos por todos na organização e que sirva de base para a alta administração em decisões relacionadas à segurança da informação, proporcionando coerência e menos complexidade, refletindo também em decisões mais justas e mais facilmente aceitas, já que se baseiam em uma política largamente difundida, e não apenas no critério pessoal de quem toma a decisão. (DIAS, 2000)

Segundo (DIAS, 2000) a política normalmente contém princípios legais e éticos a serem atendidos no que diz respeito à informática: direitos de propriedade de produção intelectual; direitos sobre *softwares* e normas legais correlatas aos sistemas desenvolvidos; princípios de implementação da segurança de informações; políticas de controle de acesso a recursos e sistemas computacionais; e princípios de supervisão constante das tentativas de violação da segurança da informação. Além disso, a política pode conter ainda os princípios de continuidade de negócios, procedimentos a serem adotados após a violação de normas de segurança estabelecidas na política, como investigações, julgamento e punições aos infratores da política e plano de treinamento em segurança de informações. É importante que a política estabeleça responsabilidades das funções relacionadas com a segurança e discrimine as principais ameaças, riscos e impactos envolvidos.

Outro benefício é que todos podem conhecer as regras impostas pela empresa, sabendo como se comportar nos mais diferentes aspectos dentro da organização, sabendo inclusive o que podem ou não fazer. Conforme CAMPOS (2006), a política de segurança contribui não somente para a redução de incidentes de segurança da informação, mas também para o aumento da produtividade, já que a busca de orientações sobre o comportamento será menor e cada um poderá se concentrar mais em suas atividades em vez de procurar as possibilidades de uso ou acesso às informações, fazendo com que as pessoas se sintam mais confortáveis conhecendo os limites.

### 3.5.1 *Planejamento*

Para a implantação de uma política de segurança em uma organização é necessário primeiramente a identificação dos recursos que são críticos a ela, definindo e analisando os objetivos que desejam ser alcançados juntamente com suas necessidades; discussões com diferentes áreas da empresa são necessárias para que a política de segurança atue na empresa como um todo e por fim, um documento formal deverá ser apresentado ao alto escalão da empresa. Após a sua aprovação, uma campanha de conscientização deve ser realizada com os clientes, parceiros, funcionários e fornecedores. (SPANCESKI, 2004)

O planejamento da política deve ser feito tendo como diretriz o caráter geral e abrangente de todos os pontos, incluindo as regras que devem ser obedecidas por todos. (SPANCESKI, 2004)

A política de segurança pode ser dividida em vários níveis, podendo ser de um nível mais genérico, com o objetivo que os executivos possam entender o que está sendo definido, nível dos usuários de maneira que eles tenham consciência de seus papéis para a manutenção da segurança na organização, e podendo ser de nível técnico que se refere aos procedimentos específicos como, por exemplo, a implementação das regras de filtragem do *firewall*. (SPANCESKI, 2004)

Segundo a NBR ISO 17799 a política de segurança deve seguir as seguintes orientações:

- Definição de segurança da informação, resumo das metas e escopo;
- Destacar a importância da segurança como um mecanismo que habilita o compartilhamento da informação;
- Declaração do comprometimento da alta direção, apoiando as metas e princípios da segurança da informação;
- Breve explanação das políticas, princípios, padrões e requisitos de conformidade de importância específica para a organização;
- Definição das responsabilidades gerais e especificas na gestão de segurança da informação, incluindo o registro dos incidentes de segurança;
- Referência à documentos que possam apoiar a política, como: políticas, normas e ou regras de segurança que os usuários devem seguir.

# 4 FIREWALL

Neste capítulo será apresentada uma definição sobre o que é esta ferramenta indispensável na infraestrutura de segurança das empresas. Serão abordados conceitos importantes para uma melhor compreensão do seu funcionamento, discutidos os tipos de *firewall* existentes e a sua arquitetura em relação à topologia da rede.

## 4.1 Definição

NAKAMURA E GEUS (2012), o *firewall* é um ponto entre duas ou mais redes, que pode ser um componente ou conjunto de componentes, por onde passa todo o tráfego, permitindo que o controle, a autenticação e os registros de todo o tráfego sejam realizados.

O principal recurso adotado nessas barreiras é a filtragem de pacotes. Ou seja, tudo o que entra e sai da rede interna deve passar previamente através do filtro, o qual analisa os cabeçalhos dos pacotes e verifica se o pacote pode seguir adiante, como mostra a Figura 1.

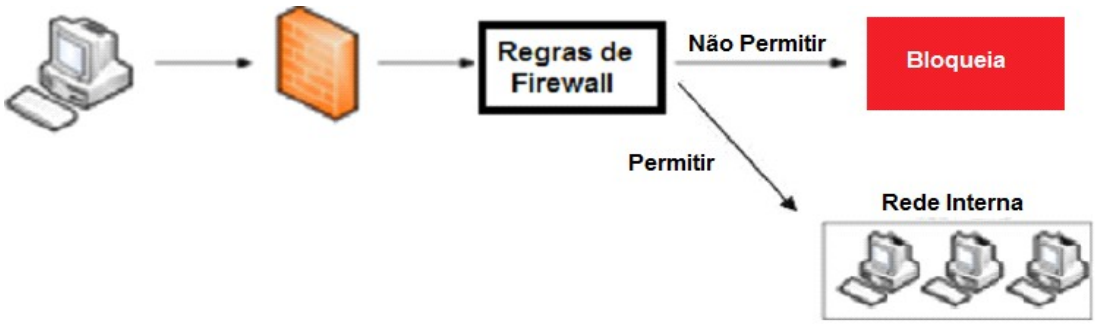


Figura 1: Filtragem de Pacotes
Fonte: Adaptado de NAKAMURA e GEUS (2012)

## 4.2 Conceitos

Com o passar do tempo o *firewall* foi evoluindo, deixando de ser um dispositivo apenas para controle de tráfego, sendo agregadas a ele novas funcionalidades, como *network address translation* (NAT) e *gateway* VPN. Nesta seção serão apresentados conceitos importantes para um melhor entendimento sobre *firewall*.

### 4.2.1 Nat

O *Network Address Translation* (NAT) é uma técnica que foi criada com o objetivo de diminuir a escassez de endereços IPs válidos na *Internet*, ele é responsável por converter endereços de rede privada para endereços válidos e roteáveis. O endereçamento IP pode ser dividido em: endereços IPs roteáveis, que são utilizados na *internet*, também chamados de endereços públicos e os endereços IPs que não são roteáveis, também chamados de endereços privados, utilizados em redes privadas, por exemplo, rede interna de uma empresa, rede residencial, entre outras. (ROCHA JUNIOR, 2010)

Na tabela 1 são apresentadas as classes de endereços IPs privados de acordo com a RFC 1918.

Tabela 1: Classes de Endereços IPs privados.

| Classe | Faixa de Endereço IP | Máscara |
|---|---|---|
| Classe A | 10.0.0.0 – 10.255.255.255 | 255.0.0.0 |
| Classe B | 172.16.0.1 – 172.31.255.255 | 255.255.0.0 |
| Classe C | 192.168.0.0 -192.168.255.255 | 255.255.255.0 |

Consideremos uma rede que possui cinquenta máquinas e que elas necessitem de acesso à *Internet*. Caso o NAT não existisse e a empresa não quisesse fazer uso de um *firewall* a nível de aplicação, cada uma dessas máquinas necessitaria ter um endereço IP público para prover tal acesso. Com o uso dessa técnica basta que um *host* (uma máquina ou um roteador) que possua o NAT habilitado tenha um IP ou um pequeno número de IPs públicos. Outra vantagem que ele possui é que a rede que faz o uso do NAT para acesso à *Internet* possui as máquinas da rede interna com endereço IP privado, e o mesmo não é roteável na *Internet*, as máquinas da rede externa não conseguirão iniciar uma conexão com destino a rede interna, proporcionando assim uma maior segurança. (ROCHA JUNIOR, 2010)

De acordo com TANENBAUM (2003) o NAT funciona da seguinte forma: suponha que uma máquina com IP 192.168.10.202 deseja acessar um determinado *site* na Internet, a requisição passará para o *gateway* da rede que é um roteador com o NAT habilitado e possui um endereço IP público 200.199.103.32. Quando a requisição chegar ao roteador, ele irá analisar o cabeçalho do pacote e criar uma associação com os IPs de origem/destino e as portas de origem/destino, guardando esta informação em uma tabela, ele irá então realizar uma substituição do endereço ip privado (192.168.10.202) para o endereço IP público (200.199.103.32). O pacote então será enviado ao servidor *web* de destino com IP 200.199.103.32, quando o servidor responder a requisição e a mesma chegar ao roteador de rede, ele irá verificar em sua tabela NAT uma associação com o IP de origem/destino e a porta de origem/destino e então realizará outra substituição, porém, dessa vez ele irá substituir o endereço IP público (200.199.103.32) pelo privado (192.168.10.202) e em seguida enviará o pacote para a máquina de origem, como podemos ver na Figura 2.

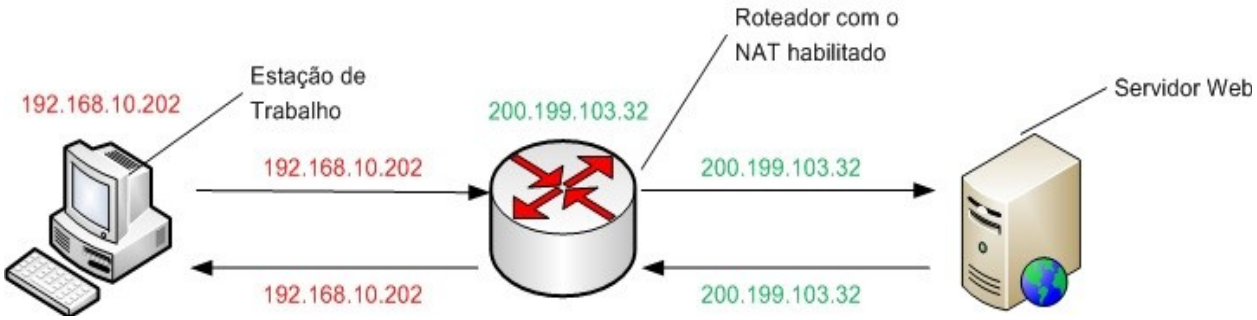


Figura 2: Funcionamento do NAT
Fonte: ROCHA JUNIOR (2010)

### 4.2.2 *VPN*

A alta competitividade do mercado gerou necessidade de comunicação entre empresas, clientes, distribuidores e parceiros. Uma das formas de prover esta conectividade é contratar junto à operadora de telefonia da região um *link* direto com a empresa, porém, esta é uma medida extremamente cara. Outra forma de realizar a conectividade entre uma matriz e uma filial, por exemplo, é com a utilização da rede pública que com o crescimento da banda larga consegue atingir altas velocidades de transmissão. Apesar de ser um meio barato, ele é extremamente inseguro e trafegar informações confidenciais pela Internet, sem a devida segurança, pode resultar em prejuízos imensuráveis. (SILVA, 2006)

Partindo desse princípio, surgiu o conceito de VPN que podemos definir como sendo uma rede privada funcionando por meio de uma rede pública. O seu funcionamento é relativamente simples: após o estabelecimento do túnel, que podemos definir como uma sessão onde duas extremidades (cliente e o servidor VPN) negociam parâmetros tais como: endereçamento, criptografia, compressão para o estabelecimento dos túneis. Os dados são criptografados e autenticados antes de serem enviados, garantindo assim o sigilo e a autenticidade das informações. (ROCHA JUNIOR, 2010)

Na Figura 3 é apresentada uma imagem que representa a comunicação entre uma Matriz e uma Filial através de uma VPN.

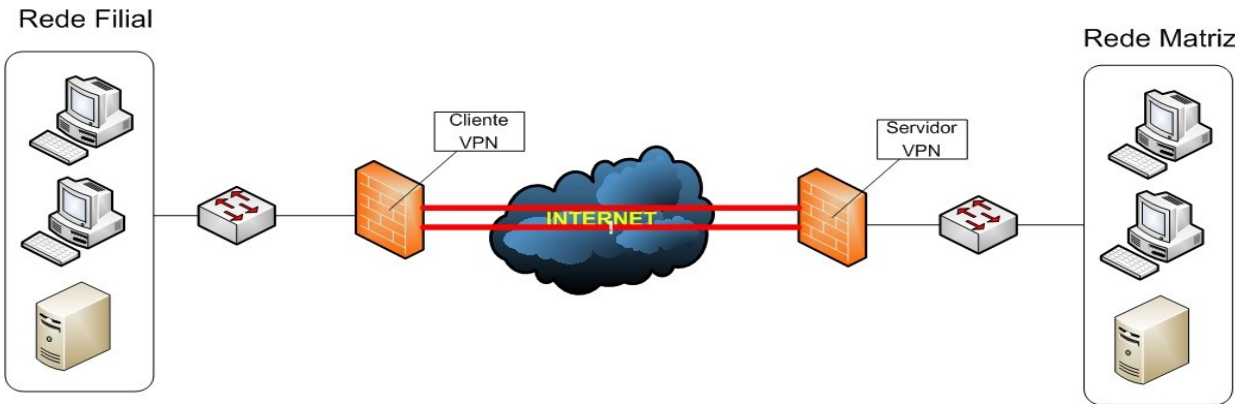


Figura 3: Funcionamento VPN.
Fonte: ROCHA JUNIOR (2010)

De acordo com PINHEIRO (2007), fazem parte da criação de uma VPN os seguintes elementos:

- Servidor VPN: responsável por aceitar, autenticar e prover as conexões dos

clientes VPN;

- Cliente VPN: é aquele que solicita ao servidor VPN uma conexão;
- Túnel: é o caminho por onde os dados passam pela rede pública. Corresponde a uma sessão, onde as duas extremidades negociam a configuração dos parâmetros (endereçamento, criptografia e compressão) para o estabelecimento do túnel;
- Protocolos de tunelamento: são os responsáveis pelo gerenciamento e encapsulamento dos túneis criados na rede pública;
- Rede Pública: efetua as conexões da VPN. Normalmente trata-se de um provedor de *Internet*.

O principal benefício da VPN, a segurança, é provido através de uma série de protocolos que encapsulam e criptografam os dados trafegados entre o cliente e o servidor.

De acordo com SILVA (2006) esses são os principais protocolos utilizados em uma VPN:

- IPSec: *O Internet Protocol Security* (IPSec) é uma extensão do protocolo IP com a capacidade de prover uma maior segurança. Trata-se de um protocolo padrão da camada 3 do modelo OSI, que oferece transparência segura de informações fim-a-fim através de rede IP pública ou privada;
- L2TP: *O Level 2 Tunneling Protocol* (L2TP) é o protocolo que atua na camada 2 e faz o tunelamento de PPP utilizando vários protocolos de rede como IP, ATM, etc, sendo utilizado para prover acesso discado a múltiplos protocolos;
- PPP: *O Point to Point Protocol* (PPP) é responsável por verificar as condições da linha telefônica (no caso das conexões *dial up*), pela validação dos usuários na rede, além de estabelecer as configurações dos pacotes;
- PPTP: *O Point to Point Tunneling Protocol* (PPTP) é uma variação do protocolo PPP, que encapsula os pacotes em um túnel fim-a-fim, porém não oferece os serviços de criptografia;
- SSL: *Secure Sockets Layer* (SSL) provê a segurança e integridade na transferência de dados pela *Internet.* É o protocolo de segurança utilizado pelo *Openvpn.*

### 4.2.3 *Bastion Hosts*

De acordo com NAKAMURA e GEUS (2012) os *bastion hosts* são os servidores que possuem instalados serviços a serem oferecidos para a *Internet.* Por estarem

expostos a rede externa, os *bastion hosts* necessitam de cuidados especiais de segurança, eles devem executar só os serviços necessários e estarem com os *patches* de segurança sempre atualizados. Os *bastion hosts* devem ser instalados em uma zona desmilitarizada (DMZ) devido ao nível de segurança que ela propicia.

### 4.2.4 *Zona Desmilitarizada (DMZ)*

De acordo com NAKAMURA e GEUS (2012), a zona desmilitarizada (DMZ) é uma rede que fica entre a rede interna, que deve ser protegida, e a rede externa. Este modelo é bastante utilizado por organizações que possuem servidores que necessitam ser acessados tanto pela rede interna, quanto pela rede externa, os chamados *Bastion Hosts.*

A grande vantagem que as DMZs proporcionam é que caso algum servidor que possua acesso externo seja invadido, a rede interna continuará segura. (ROCHA JUNIOR, 2010)

A forma mais simples de criar uma DMZ é possuir um *firewall* com três ou mais placas de rede na qual uma placa de rede será conectada a um roteador de acesso à *Internet* (rede externa), uma placa será conectada a um *switch* para acesso da rede interna e a outra placa de rede será conectada a um *switch* que pertencerá a DMZ. (ROCHA JUNIOR, 2010)

Na Figura 4 é apresentado um exemplo de uma DMZ separando a rede interna, dos servidores que precisam ser acessados pela rede externa. Como podemos perceber as quatro máquinas que estão localizadas no lado direito da imagem são chamadas de *bastion hosts.*

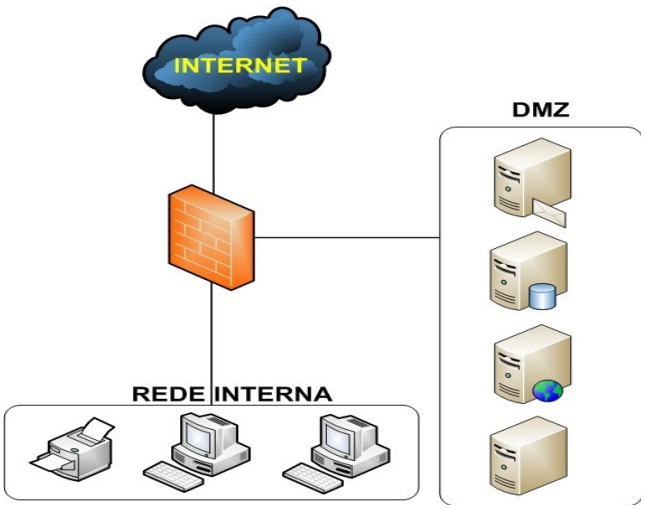


Figura 4: Modelo de DMZ.
Fonte: ROCHA JUNIOR (2010)

## 4.3 Tipos de Firewalls

Existem três tipos principais de *firewall: firewall* a nível de pacote, que funciona nas camadas de rede e de transporte, *firewall* a nível de pacotes baseado em estados e o *firewall* a nível de aplicação, que funciona nas camadas de aplicação, sessão e transporte. (ROCHA JUNIOR, 2010)

### 4.3.1 *Firewall a Nível de Pacotes*

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

Como o *firewall* a nível de pacotes trabalha apenas nas camadas de rede e transporte do modelo OSI, isso faz com que ele seja mais simples e flexível de ser implementado. A maioria dos roteadores, por serem *gateways* das redes já possuem esta funcionalidade. (ROCHA JUNIOR, 2010)

As decisões sobre aceitar ou descartar o pacote são baseadas em:

- Endereço IP de origem e de destino;
- Mensagens ICMP;
- Tipo de porta (TCP ou UDP) de origem e de destino;
- *Flags* SYN, SYN-ACK e ACK do *handshake* TCP.

Em contra partida ao ganho em desempenho e na simplicidade de ser implementado, o *firewall* a nível de pacote apresenta um fator negativo no quesito segurança. Como já foi dito, a decisão de aceitar ou descartar os pacotes é tomada com base em apenas alguns parâmetros do cabeçalho dos pacotes, ou seja, ele não examina o conteúdo do pacote. Os pacotes podem ser facilmente falsificados, este tipo de ataque é chamado de *IP spoofing*. (ROCHA JUNIOR, 2010)

Na Figura 5 é apresentado o cabeçalho TCP cujos parâmetros que são analisados pelo firewall a nível de pacotes são: Porta de origem, porta de destino, e as *flags.*

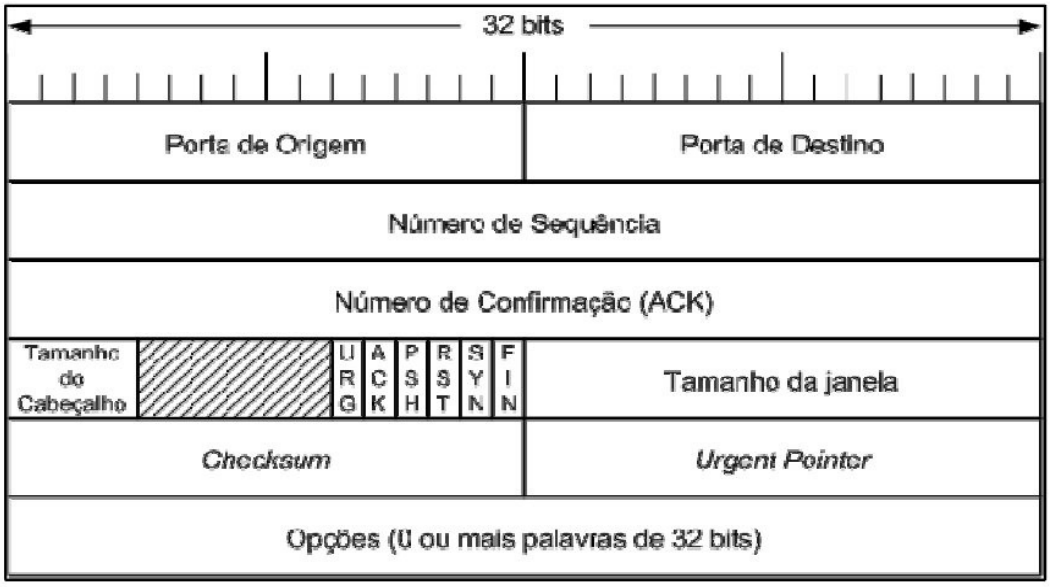


Figura 5: Cabeçalho TCP.
Fonte:NAKAMURA E GEUS (2012)

A Figura 6 apresenta o cabeçalho UDP cujos parâmetros que são analisados pelo *firewall* a nível de pacotes são: Porta de origem e porta de destino.

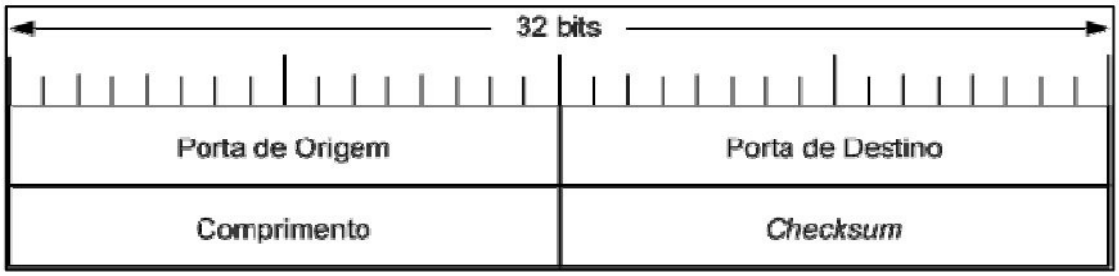


Figura 6: Cabeçalho UDP.
Fonte: NAKAMURA E GEUS (2012)

### 4.3.2 *Filtro de pacotes com estado*

Visando aumentar a segurança, um *firewall* deve guardar informações e controlar todo fluxo de comunicação passando através dele. Para fazer decisões de controle para serviços baseados em TCP/IP (aceitar ou rejeitar pedidos de conexões, autenticar, cifrar, etc), um *firewall* deve obter, armazenar, recuperar e manipular informação derivada de todas as camadas de comunicação incluindo a de aplicação. (ROCHA JUNIOR, 2010)

O *firewall* a nível de pacotes baseado em estado, também chamado de *stateful packet filter,* é semelhante ao *firewall* a nível de pacotes sendo que ele possui uma

funcionalidade a mais, a tomada de decisão se o pacote será aceito ou descartado baseada em dois elementos: Informações do cabeçalho do pacote e uma tabela de estados que guarda o estado das conexões. (ROCHA JUNIOR, 2010)

De acordo com NAKAMURA e GEUS (2012) o uso da tabela de estados melhora o desempenho do sistema, pois apenas os pacotes que iniciam a conexão serão comparados com a tabela de regras, os pacotes restantes são comparados com a tabela de estados. A quantidade de regras na tabela de estados é menor e a verificação da regra não é feita de forma seqüencial, e sim por meio de tabelas *hash*.

A figura 7 apresenta o funcionamento do *firewall* de pacotes baseado em estados. Quando o pacote chega, o *firewall* verifica se já existe uma regra na tabela de estados para ele, caso exista, o *firewall* aceita o pacote sem consultar a tabela de regras, caso não exista significa que é um pacote novo e então é consultada a tabela de regras, caso exista uma regra que aceite o pacote, ele é aceito e é adicionada uma regra na tabela de estados, os demais pacotes descendentes dessa conexão serão verificados apenas pela tabela de estados. (ROCHA JUNIOR, 2010)

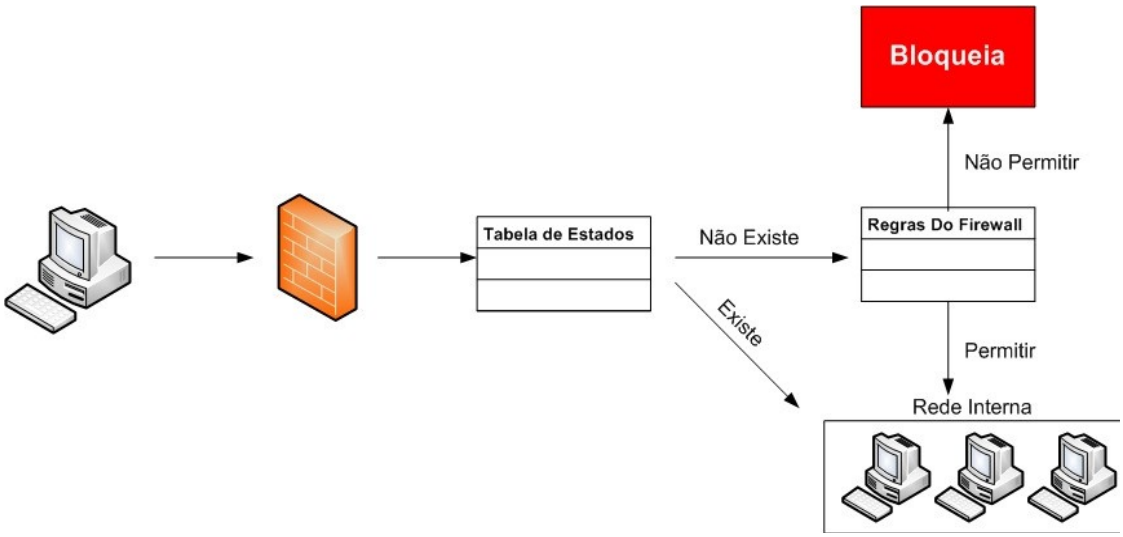


Figura 7: Firewall de pacotes baseado em estados.
Fonte: Adaptado de NAKAMURA E GEUS (2012)

Por manter estados do andamento da comunicação, um filtro de pacotes com estados permite o suporte a protocolos sem conexão, tais como UDP, e a serviços baseados em RPC, que usam portas alocadas dinamicamente. Com estes filtros é possível extrair informações de contexto (o campo de dados) dos pacotes UDP, mantendo dessa forma uma conexão virtual de forma a fazer uma filtragem mais inteligente. (NAKAMURA E GEUS, 2012)

### 4.3.3 *Firewall a Nível de Aplicação*

Um *firewall* a nível de aplicação, também conhecido como servidor *proxy*, proporciona um nível mais refinado de segurança, ele faz mais do que analisar o cabeçalhos TCP, UDP e IP, ele toma decisões com base em dados da aplicação. (ROCHA JUNIOR, 2010)

Um servidor *proxy* funciona da seguinte maneira: Um cliente, que neste caso pode ser um navegador *web*, se conecta ao servidor *proxy* e realiza uma requisição de um *site* qualquer, o servidor então recebe esta requisição e a encaminha para o servidor *web* de destino. O servidor *web* irá responder a requisição ao servidor *proxy* que irá repassar os dados para o cliente que realizou a requisição. (ROCHA JUNIOR, 2010)

Um servidor *proxy* pode melhorar a segurança por examinar o fluxo da conexão na camada de aplicação, mantendo informações de contexto para o processo de decisão. Com isso, ele permite fazer uma filtragem na camada de aplicação evitando ataques relacionados às vulnerabilidades dos protocolos de aplicação, possíveis em filtros de pacotes. Além disso, pode ser também utilizado para implementar *cache* de dados para um determinado serviço e autenticação de usuários. (NAKAMURA E GEUS, 2012)

## 4.4 Arquitetura

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

Nesta seção serão apresentadas as três seguintes possíveis arquiteturas para a implantação de um *firewall: Dual-homed Host, Screened Host e Screened Subnet*.

### 4.4.1 *Dual-Homed Host*

Essa arquitetura normalmente é montada sobre um computador que possui no mínimo duas interfaces de rede, agindo também como um roteador entre as duas redes conectadas às placas de rede, o que permite a retransmissão de pacotes diretamente entre as redes e que sistemas dentro do *firewall* se comuniquem diretamente com a *Internet* e com a rede protegida, desde que estes acessos não estejam bloqueados. (ROCHA JUNIOR, 2010)

Conforme ROCHA JUNIOR (2010), apesar de esta arquitetura prover alto controle, sua performance fica degradada, pois realiza o maior processamento em cima de conexões. Como toda a rede interna depende deste *host* para se conectar à *Internet,* esse deve ser o mais seguro possível, para evitar que a conexão de toda a rede interna perca o acesso à *Internet.* Esse tipo de arquitetura é indicado para redes onde o tráfego pela *Internet* não é vital para a empresa, não prove serviços, como por exemplo, um servidor de páginas ou contenha dados de alto valor. Esse *host* fica posicionado entre a *Internet* e a rede interna, conforme mostrado na figura 8.

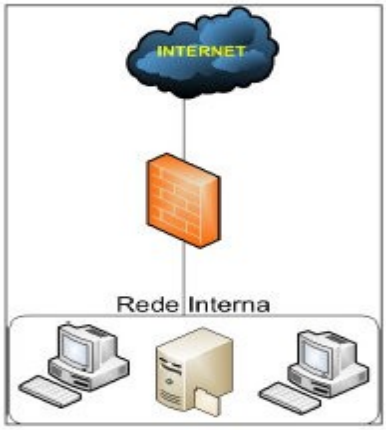


Figura 8: Dual-Homed Host.
Fonte: Adaptado de
NAKAMURA E GEUS
(2012)

### 4.4.2 *Screened Host*

Nessa arquitetura existe um roteador separado, normalmente chamado de roteador de borda e um *host* chamado de *bastion host,* que nada mais é que um dispositivo que implementa um alto grau de segurança e um roteador separado para a rede interna. Neste caso a segurança primária é implementada pelo filtro de pacotes que está implementado no roteador de borda e a segurança secundária é implementada pelo *bastion host,* o qual provê serviços somente para a rede interna. (ROCHA JUNIOR, 2010)

O filtro de pacotes que está implementado no roteador de borda, deve permitir acesso direto à *Internet* somente para o *bastion host*, e os computadores da rede interna devem estabelecer as conexões à *Internet* somente passando por ele. Se o *bastion host* for atacado, ou se o roteador de borda estiver comprometido, toda a rede interna estará vulnerável. Essa arquitetura é indicada para redes onde chegam poucas conexões pela

*Internet,* onde a rede Interna for relativamente segura e o *screened host* não seja um servidor *web*. (ROCHA JUNIOR, 2010)

A figura 9 mostra um diagrama do que é a arquitetura *screened host*.

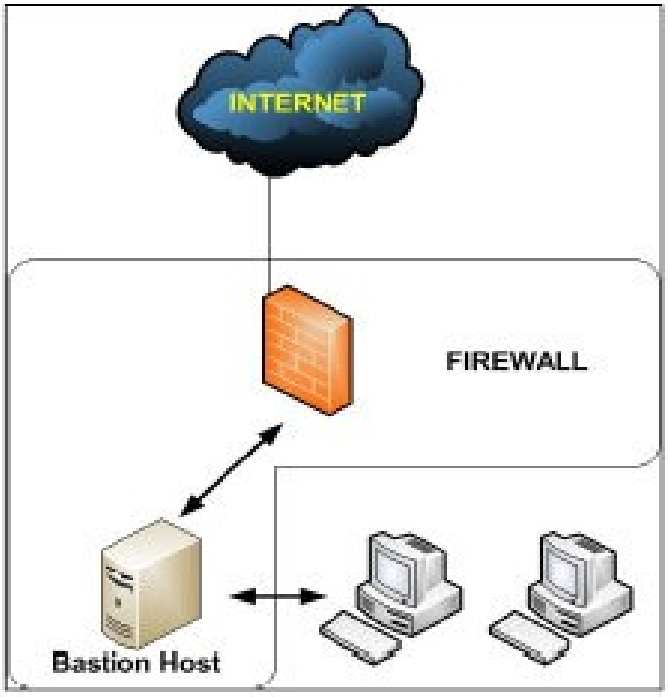


Figura 9: Screened Host.
Fonte: Adaptado de NAKAMURA E GEUS (2012)

### 4.4.3 *Screened Subnet*

A arquitetura *screened subnet* é um melhoramento da arquitetura *screened host* pois proporciona um aumento considerável da segurança ao fazer uso de uma DMZ. (ROCHA JUNIOR, 2010)

No modelo anterior se o *bastion host* fosse comprometido o invasor teria total acesso a rede interna, isso não ocorre na arquitetura *screened subnet.* O *bastion host* fica em uma DMZ, que é uma zona que fica entre a rede interna e a externa, caso ele seja comprometido o filtro interno ainda protegerá a rede interna. (ROCHA JUNIOR, 2010)

De acordo com NAKAMURA e GEUS (2012) o filtro externo deve permitir o acesso externo aos serviços que estão na DMZ, assim como as requisições dos usuários internos. Permitir o tráfego do *bastion host* para a rede interna pode comprometer a segurança, caso ele seja atacado, sendo assim, o tráfego originado do *bastion host* em direção a rede interna deve ser bloqueado. A figura 10 mostra o funcionamento dessa arquitetura.

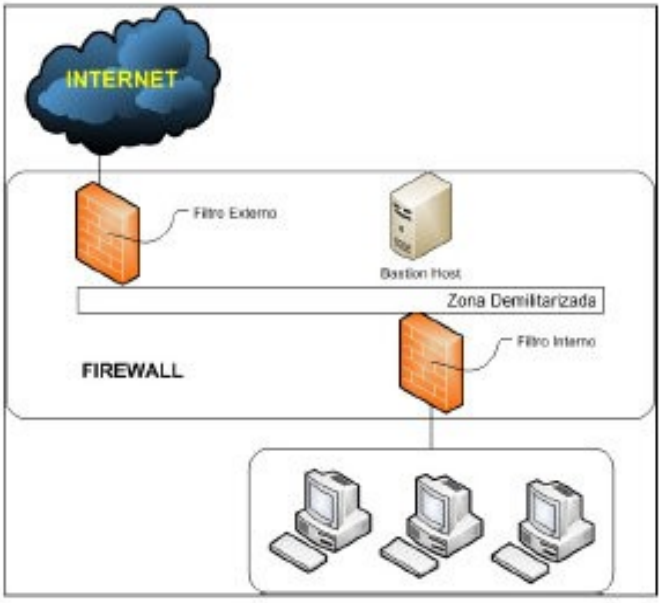


Figura 10: Screened Subnet.
Fonte: Adaptado de NAKAMURA E GEUS (2012)

# 5 ACL'S

Uma lista de controle de acesso (ACL) é uma lista de regras ordenadas que permitem ou bloqueiam o tráfego de pacotes baseadas em certas informações presentes nos cabeçalhos dos pacotes da camada 3 ou da camada 4. (OLIVEIRA, 2001)

Por ser uma lista ordenada de regras, isso significa que a ordem de criação das regras na lista de acesso é muito importante, um dos erros mais comuns durante a criação de listas de acesso é a configuração das regras em ordem incorreta. É importante saber também que, no final da lista de acesso, existe uma regra implícita que bloqueia tudo; um pacote que não é explicitamente permitido será bloqueado por estra regra. (OLIVEIRA, 2001)

## 5.1 Funcionamento da ACL

Conforme OLIVEIRA (2001) o processo tem início quando uma interface recebe um pacote. O roteador verifica a tabela de roteamento à procura de uma rota para o pacote. Caso não tenha uma rota, este pacote será descartado e será enviada para a origem uma mensagem de ICMP (*unreachable destination*). Caso contrário, verifica-se se existe uma lista de controle de acesso aplicada à interface. Não existindo, o pacote é enviado para o *buffer* da porta de saída. Existindo, o pacote é analisado pela lista de controle de acesso da interface em questão. Uma vez que o fluxo de dados através de uma determinada interface é bidirecional, uma ACL pode ser aplicada em uma direção específica da interface:

- Entrada (*inbound*) - verifica se o processamento do pacote deve continuar após o seu recebimento em uma determinada interface;
- Saída (*outbound*) - verifica se o pacote deve ser enviado para uma interface de saída ou bloqueado.

## 5.2 Fluxo do pacote através da ACL

A lista de acesso é conferida em ordem sequencial, ou seja, o pacote é testado a partir da primeira regra. Assim, se o pacote enquadrar-se em alguma regra, é verificada a condição do mesmo se é permitido ou não. Caso o pacote não se enquadre em nenhuma das regras, o mesmo será bloqueado pela última regra, a qual é implícita e bloqueia tudo que não está explicitamente permitido. Isto pode ser visto através da Figura 11. (OLIVEIRA, 2001)

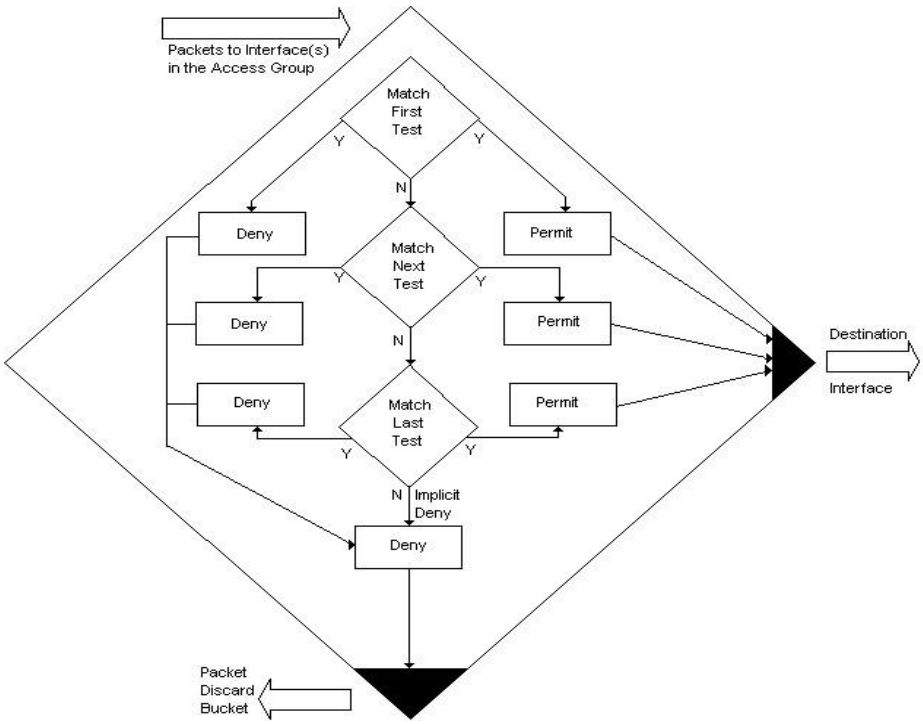


Figura 11: Funcionamento de uma ACL.
Fonte: OLIVEIRA (2001)

## 5.3 Tipos de Listas de Acesso

De acordo com OLIVEIRA (2001), os dois principais tipos de listas de acesso são:

- **Padrão (*standard*)** - A lista de acesso padrão verifica o IP de origem de um pacote que pode ser roteado. Baseado na rede/sub-rede/endereço dos *hosts*, é permitido

ou bloqueado o envio do pacote.

- **Estendida (*extended*)** - A lista de acesso estendida possui mais recursos de verificação. Com ela, pode-se analisar o IP de origem, o IP de destino, a porta de origem, a porta de destino, os protocolos e alguns outros parâmetros, de forma a permitir ao administrador de segurança uma maior flexibilidade na elaboração das regras.

## 5.4 Identificando as Listas de Acesso

Ainda de acordo com OLIVEIRA (2001), ao se configurar listas de acesso em um roteador, deve-se identificar cada lista de forma única. Em alguns protocolos, as listas de acesso devem ser identificadas por nome; já em outros, devem ser identificadas por número; e alguns protocolos permitem a identificação por nome ou número. Quando utilizamos números para identificar as listas de acesso, estes devem pertencer a um conjunto de números que "identificam" o protocolo. A partir do IOS 11.2, é permitida a identificação da lista de acesso utilizando-se nomes definidos pelo administrador. Isso para listas de acesso padrão ou estendidas.

Baseado no identificador, o roteador decide qual *software* de controle de acesso deve ser utilizado. Observe, nos quadros 1; 2 e 3, o agrupamento de nomenclatura para listas de acesso:

Quadro 1: Agrupamento por tipo de lista de acesso.

| Tipo de Lista de Acesso | Número/Identificador |
|---|---|
| IP Standard<br>Extended | 1 - 99<br>100 - 199<br>por nome ( IOS >= 11.2 ) |
| IPX Standard<br>Extended<br>Filtro SAP | 800 - 899<br>900 - 999<br>por nome ( IOS >= 11.2F ) |
| AppleTalk | 600 - 699 |

Quadro 2: Protocolos identificados por nome.

| Protocolo |
| --- |
| Apollo Domain |
| IP |
| IPX |
| ISO CLNS |
| NetBIOS IPX |
| Source-router bridging Netbios |

Quadro 3: Protocolos identificados por números.

| Protocolo | Faixa |
| --- | --- |
| IP | 1 - 99 |
| Extended IP | 100 - 199 |
| Ethernet type code | 200 - 299 |
| Ethernet address | 700 - 799 |
| Transparent bridging ( protocol type ) | 200 - 299 |
| Transparent bridging ( vendor code ) | 700 - 799 |
| Extended transparent bridging | 1100 - 1199 |
| DECnet and extended DECnet | 300 - 399 |
| XNS | 400 - 499 |
| Extended XNS | 500 - 599 |
| AppleTalk | 600 - 699 |
| Sourc-route bridging ( protocol type ) | 200 - 299 |
| Source-router bridging ( vendor code ) | 700 - 799 |
| IPX | 800 - 899 |
| Extended IPX | 900 - 999 |
| IPX SAP | 1000 - 1099 |
| Standard VINES | 1 - 100 |
| Extended Vines | 101 - 200 |
| Simple VINES | 201 - 300 |

## 5.5 O Funcionamento da Wildcard em Roteadores Cisco

A filtragem de endereço ocorre com a utilização de máscaras *wildcard* para identificar o que é permitido ou bloqueado nos *bits* do IP. As máscaras *wildcard* para os *bits* de endereço IP utilizam o número 1 e o número 0 para a identificação do que deve ser filtrado, conforme mostra a Figura 12. (OLIVEIRA, 2001)

- O valor 0 significa que o *bit* deve ser checado;
- O valor 1 significa que o *bit* deve ser ignorado;

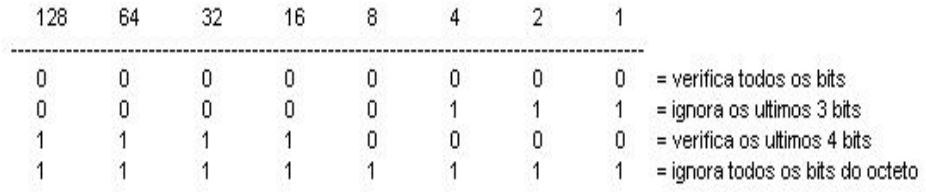


Figura 12: Funcionamento da Wildcard.
Fonte: OLIVEIRA (2001)

## 5.6 Mantendo Backup dos Arquivos de Configuração

OLIVEIRA (2001), fala ainda sobre a grande necessidade de fazer um *backup* de configuração. A manutenção de cópias de segurança dos arquivos de configuração e do IOS é de fundamental importância, pois, eventualmente, devido a falhas de corrente elétrica, estes arquivos pode ser danificados ou apagados da memória *flash.* Além disso, a manutenção de *backup* dos arquivos facilita a administração de redes com vários roteadores.

O administrador de rede possui várias possibilidades de realização de cópias de segurança, uma delas é o TFTP. O ideal é existir uma rede segregada dedicada para a função de gerência de rede, onde se inclui a gerência da segurança. Esta deve estar isolada da rede de dados e prover canais de comunicação *out-of-band*. No caso de necessidade de utilização dos canais de dados para funções de gerência, é recomendável a utilização de canais seguros. (OLIVEIRA, 2001)

De qualquer forma, como o serviço TFTP não requer autenticação, é extremamente recomendável a implementação de algum mecanismo que controle a origem das conexões ao servidor TFTP. (OLIVEIRA, 2001)

Os comandos TFTP para realizar cópia de segurança e atualização de arquivos são:

- **copy tftp running-config** - Configura o roteador de forma direta, copiando os arquivos do servidor TFTP para a DRAM do roteador;
- **copy startup-config tftp** - Realiza o *backup* dos arquivos de inicialização da NVRAM para o servidor TFTP;
- **copy tftp startup-config** - Atualiza o arquivo de inicialização, copiando do servidor TFTP e gravando na NVRAM.

## 5.7 Desempenho do Roteador

Segundo OLIVEIRA (2001), geralmente o desempenho do roteador é uma das principais preocupações dos administradores de rede quando se fala em implementação de listas de controle de acesso no mesmo. Sabemos que as regras da lista de controle de acesso são analisadas sequencialmente até que seja encontrada uma regra, que coincida com o pacote analisado, ou que se chegue à última regra que bloqueia tudo que não está permitido. Dessa forma, devemos observar alguns procedimentos que devem ser adotados no sentido de minimizar o impacto que as listas de controle de acesso possam causar:

- Mensurar os recursos do roteador (memória, processador, outros);
- Avaliar os serviços habilitados no roteador (criptografia, outros);
- Entender o tráfego da rede;
- Mensurar o volume de pacotes;
- Classificar o volume de tráfego por servidor, protocolo e sentido do tráfego.

A análise dessas informações é de vital importância para a implementação das regras de forma a minimizar os possíveis impactos que possam vir a ocorrer no roteador.

# 6 IPTABLES

O *firewall* tornou-se um dispositivo básico de segurança em qualquer corporação e está sendo cada vez mais utilizado pelas empresas.

Neste capítulo será apresentado o iptables, *firewall* do Sistema Operacional Linux, que é bastante utilizado pelos administradores de rede. Serão discutidos aspectos de sua estrutura, funcionamento, criação de regras, erros cometidos pelos administradores tanto na criação das regras como em sua implantação e uma análise crítica sobre as suas vantagens e desvantagens.

## 6.1 Definição

De acordo com NETO (2004), o iptables é o *firewall* do Sistema Operacional Linux, que foi concebido pelo australiano Paul Rusty Russel em colaboração com Michel Neuling sendo incorporada a versão 2.4 do *kernel* do Linux em julho de 1999. Ele é o sucessor do IPFWADM e IPCHAINS, que são os *firewalls* respectivamente implementados nos *kernels* 2.0.e 2.2.

Ele é um *front-end* que permite aos usuários manipular as tabelas do *netfilter*. De acordo com NAKAMURA e GEUS (2012), o *netfilter* é um *framework* para a manipulação de pacotes. Ele é agregado ao *kernel* do Sistema Operacional Linux e é responsável por tratar a entrada e saída dos pacotes. Portanto, o iptables trabalha juntamente com o *netfilter* na análise dos pacotes sendo, a solução, muitas vezes chamada pelos administradores de *netfilter/iptables*.

O iptables é um *firewall* a nível de pacotes, ou seja, ele toma as suas decisões de aceitar ou descartar o pacote com base nas informações de endereço IP de origem/destino, porta, estado da conexão, entre outros parâmetros encontrados nos cabeçalhos dos pacotes. (ROCHA JUNIOR, 2010)

Uma das principais novidades do iptables é que foi incorporado a ele a tecnologia *statefull*, que foi originalmente concebida pela empresa Check Point para o seu *firewall* comercial. Esta tecnologia permite que o *firewall* guarde o estado da conexão de um pacote em sua tabela de estados, caso um novo pacote referente a uma conexão já estabelecida chegue para ser analisado, o mesmo não será analisado pelas regras e sim pela tabela de estados da conexão. (ROCHA JUNIOR, 2010)

A seguir serão apresentadas algumas características do iptables, de acordo com SILVA (2006):

- Especificação de portas/endereço de origem/destino;
- Suporte a protocolos TCP/UDP/ICMP;
- Suporte a interfaces de origem/destino de pacotes;
- Tratamento de tráfego dividido em *chains*, proporcionando uma melhor organização;
- Possui mecanismos internos para rejeitar automaticamente pacotes duvidosos ou mal formados;
- Suporte ao roteamento de pacotes;
- Suporte a especificação de tipo de serviço para priorizar o tráfego de determinados tipos de pacotes;
- Permite enviar alertas personalizados ao *syslog* sobre o tráfego aceito/bloqueado;
- Redirecionamento de endereços IPs e portas;
- Suporte a modificação de endereços IPs.

## 6.2 Pré-Requisitos e Instalação

O iptables é uma aplicação extremamente leve e, portanto, não necessita de um *hardware* robusto, podendo ser executado em uma máquina com processador 386 com 4 MB de memória RAM. O seu requisito mais importante é que o Sistema Operacional Linux que ele esteja instalado tenha o *kernel* 2.4.x instalado, e que tenha sido compilado com suporte ao iptables. (NETO,2004)

A maioria das distribuições Linux já vem com o iptables instalado por padrão, porém, caso o mesmo não venha pré-instalado, é possível baixar o pacote principal através do site do desenvolvedor http://www.netfilter.org/ ou caso a distribuição possua um gerenciador de pacotes é possível realizar a instalação por ele. (ROCHA JUNIOR, 2010)

Caso a instalação seja através do gerenciador de pacotes, basta digitar o comando "`aptget install iptables`" para distribuições baseadas em Debian ou "`yum install iptables`" para distribuições baseadas em Red Hat. (ROCHA JUNIOR, 2010)

Com a instalação do pacote principal alguns comandos são criados na base do Sistema Operacional, eles serão listados logo a seguir:

- iptables – Comando principal para trabalhar com protocolos ipv4;
- ip6tables - Comando principal para trabalhar com protocolos ipv6;
- iptables-save - Salva as regras atuais inseridas na sessão ativa em um arquivo especificado pelo administrador;
- iptables-restore - Restaura regras salvas pelo utilitário iptables-save.

Apesar da existência de comandos para manipulação das regras, muitos administradores preferem criar as suas regras na forma de um *script*, fazendo uso apenas dos comandos iptables e ip6tables. É feito então uma chamada ao *script* contendo as regras do *firewall* no arquivo /etc/rc.local, que é responsável pela execução de arquivos após a inicialização do sistema. (ROCHA JUNIOR, 2010)

## 6.3 Estrutura

A estrutura do iptables é formada basicamente por regras, tabelas e *chains*. Tal organização permite um melhor entendimento sobre o seu funcionamento, as tabelas possuem *chains* que por sua vez armazenam as regras.

As regras são basicamente os comandos criados pelos administradores de rede para aceitar ou descartar o pacote com base nas informações de endereço IP de origem/destino, porta de destino/origem, estado da conexão, entre outras opções. As regras, quando executadas, rodam no *kernel* do sistema operacional. Portanto, caso a máquina venha a ser reiniciada, o sistema operacional não salvará as regras, sendo necessário executá-las novamente pelo o administrador quando a máquina for inicializada novamente. (ROCHA JUNIOR, 2010)

As *chains* são os locais onde as regras ficam armazenadas. Existem dois tipos

delas: as que são padrão do sistema como a *INPUT, OUTPUT, FORWARD, PREROUTING e POSTROUTING*, e as criadas pelo próprio administrador de rede, não tendo um número limite e exigindo apenas que o nome possua no máximo trinta e um caracteres sem espaços em branco. (ROCHA JUNIOR, 2010)

As tabelas armazenam as chains e as regras criadas pelos usuários. O iptables possui três tabelas, cada uma delas é responsável por um determinado tipo de tráfego. A seguir serão descritas as tabelas *filter, nat e mangle* e suas *chains* de acordo com SILVA (2006).

A tabela *filter* possui uma grande importância no quesito de segurança, pois ela trata as ações de entrada, saída e passagem de pacotes, é ela que irá definir se o pacote será aceito ou descartado. Possui por padrão três *chains:*

- INPUT – Responsável por inspecionar os dados com destino ao próprio *firewall*. Ela é utilizada quando o administrador deseja permitir ou bloquear pacotes com destino ao próprio *firewall*;
- OUTPUT – Responsável por inspecionar dados que são originados no próprio *firewall*. Ela é utilizada quando o administrador deseja permitir ou bloquear pacotes com saída do próprio *firewall*;
- FORWARD – Responsável por inspecionar os dados que passam pelo *firewall,* no qual o destino é uma máquina qualquer que não seja ele próprio. Ela é utilizada quando o administrador deseja bloquear ou permitir o tráfego com destino ou origem a outra máquina na rede e que este dado passa obrigatoriamente pelo *firewall;*

A tabela NAT incorpora as funções do *Network Address Translation* (NAT) atuando na parte de redirecionamento de pacotes e substituição de endereços ips. Possui por padrão três *chains:*

- PREROUTING – É utilizada para realizar alterações nos pacotes antes do roteamento. Esta *chain* é usada quando o administrador deseja fazer o redirecionamento de ips e portas;
- POSTROUTING - É utilizada para realizar alterações nos pacotes depois do roteamento. Esta *chain* é usada quando o administrador deseja fazer alterações no endereço ip, por exemplo, a troca de um endereço privado por um ip público;
- OUTPUT – Trata pacotes emitidos pelo próprio *localhost*.

A tabela *mangle* é utilizada para alterações especiais nos pacotes, como por

exemplo, a prioridade de tráfego proporcionando assim uma Qualidade de Serviço (QoS). Esta tabela também é necessária quando o administrador deseja implantar o serviço de roteamento avançado em sua rede. Possui por padrão cinco *chains:*

- INPUT – É Consultada quando os pacotes precisam ser modificados antes de serem enviados para o chain *INPUT* da tabela *filter.*
- PREROUTING – É utilizada para realizar alterações nos pacotes antes do roteamento;
- OUTPUT – Trata pacotes emitidos pelo próprio *localhost;*
- FORWARD – Responsável por tratar os dados que passam pela máquina;
- POSTROUTING - É utilizada para realizar alterações nos pacotes depois do roteamento;

Na Figura 13 é apresentado um diagrama criado com base na estrutura do iptables, os desenhos em amarelo representam as tabelas e os verdes representam as suas respectivas *chains.*

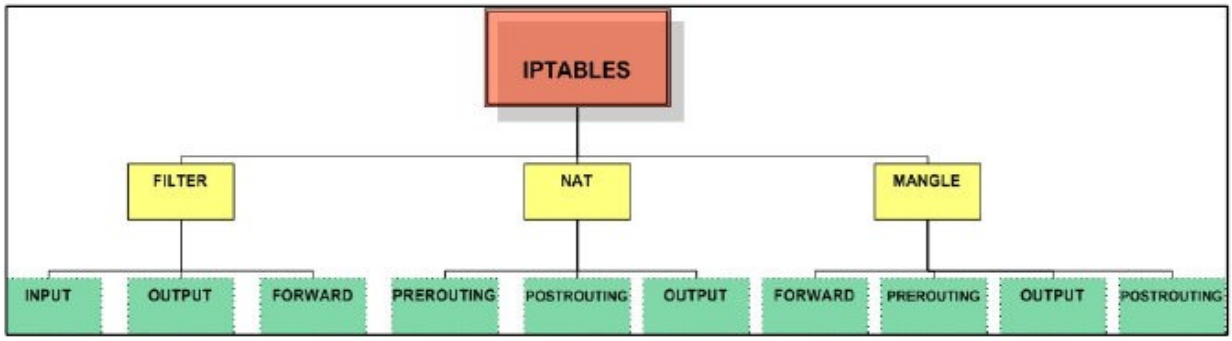


Figura 13: Estrutura do Iptables.
Fonte: ROCHA JUNIOR (2010)

## 6.4 Funcionamento do Iptables

Como foi apresentado na seção anterior, o *firewall* é formado por tabelas que possuem *chains* e que por sua vez armazenam as regras. Os pacotes ao chegarem ao *firewall* não são analisados por esses itens ao mesmo tempo, existe uma ordem específica na qual ele é submetido. (ROCHA JUNIOR, 2010)

O administrador de rede precisa estar atento ao funcionamento do iptables, pois irá influenciar na criação das regras. A chain *PREROUTING*, por exemplo, altera alguns atributos do pacote como o IP de destino e porta de destino. Se chegar um pacote ao

*firewall* e existir uma regra na *PREROUTING* para o mesmo, ele será alterado e caso o administrador de rede necessite liberá-lo com a *chain FORWARD*, ele precisa saber que na ordem de prioridade o pacote passa primeiro pela *chain PREROUTING*, ou seja, a regra na *FORWARD* deverá ser criada de acordo com os atributos alterados na *chain PREROUTING.* (ROCHA JUNIOR, 2010)

Segundo NAKAMURA e GEUS (2012), cada *chain (INPUT, OUTPUT e FORWARD)* possui seu próprio conjunto de regras de filtragem. Quando o pacote atinge uma das cadeias é examinado pelas regras dessa cadeia. Se a cadeia tiver, por exemplo, uma regra que define que o pacote deve ser descartado, ele será descartado nesse ponto. É importante salientar também que quando um pacote está de acordo com alguma das regras criadas no iptables, ele não será mais inspecionado por outras.

Na Figura 14 é apresentado um diagrama com a ordem de checagem das regras em relação às tabelas e *chains* a que elas pertencem, desde que o pacote chega ao *firewall* até a sua análise e uma tomada de decisão.

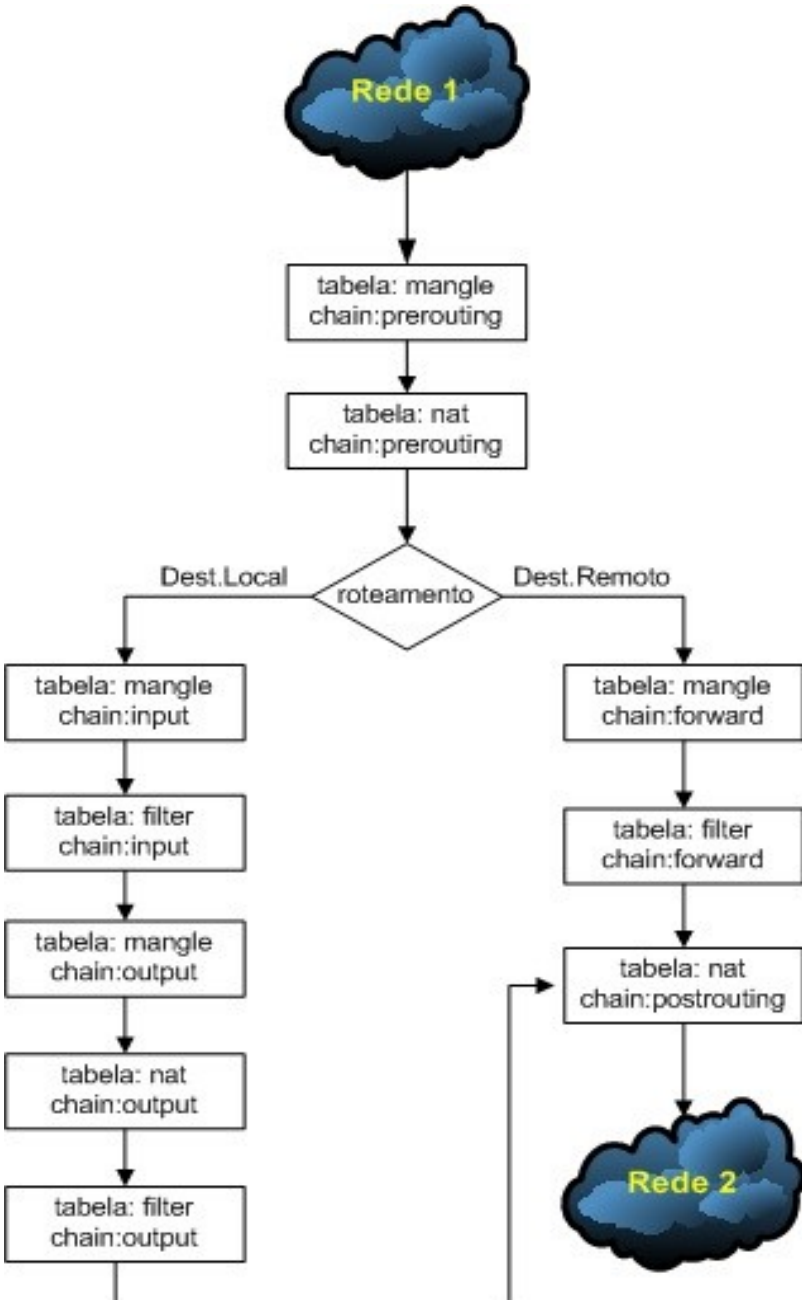


Figura 14: Funcionamento do Iptables.
Fonte: ROCHA JUNIOR (2010)

Primeiramente quando o pacote chega a placa de rede, ele será analisado pelo *kernel* do Sistema Operacional. A primeira checagem a que o pacote será submetido é a tabela *mangle*, que é responsável por realizar alterações especiais nos pacotes e priorização de tráfego, e a primeira *chain* será a *PREROUTING,* utilizada para realizar alterações nos pacotes antes do roteamento. Em seguida o pacote será analisado pela mesma *chain*, porém da tabela *nat*, que é responsável pelo redirecionamento de pacotes e substituição de endereços ips. (ROCHA JUNIOR, 2010)

Em seguida acontece a decisão de roteamento na qual o pacote será analisado de acordo com os dados do seu cabeçalho, se o destino for o próprio *firewall* ou outra máquina da rede, tal decisão é de grande relevância visto que a ordem de checagem das tabelas e *chains* serão diferentes para cada caso. (ROCHA JUNIOR, 2010)

Caso o destino seja o próprio *firewall* ou originado dele mesmo, todas as checagens a partir desse ponto serão realizadas respectivamente pelas chains *INPUT* e *OUTPUT* nas tabelas *nat, mangle* e *filter*. A primeira checagem que o pacote será submetido nesse ponto é a tabela *mangle* e a *chain* é a *INPUT*, que é consultada quando os pacotes com destino a própria máquina precisam ser modificados, logo em seguida os dados serão submetidos à tabela *filter*, que trata as ações de bloquear ou aceitar a entrada, saída e passagem do tráfego, nesse caso como a *chain* é a *INPUT,* será tratado o tráfego de entrada. Posteriormente a checagem será realizada pela *chain OUTPUT*, responsável por tratar os pacotes emitidos pelo próprio *localhost* com destino a outras redes, nas tabelas *mangle, nat* e *filter* respectivamente. (ROCHA JUNIOR, 2010)

Caso o destino seja outra máquina da rede, a primeira checagem que o pacote será submetido nesse ponto é a tabela *mangle* e a *chain* é a *FORWARD*, que é responsável por tratar os dados que passam pela máquina e em seguida, o pacote será inspecionado pela mesma *chain*, porém, da tabela *filter.* (ROCHA JUNIOR, 2010)

Em ambos os casos, seja em pacotes com destino a própria máquina ou para outra, a última tabela a que o pacote será submetido é a *nat* e a *chain* é a *POSTROUTING*, que é responsável por realizar alterações no endereço ip dos pacotes depois do roteamento. (ROCHA JUNIOR, 2010)

É importante salientar que se o *kernel* não tiver o *forwarding* habilitado ou se não souber para onde encaminhar o pacote o mesmo será descartado. Para habilitar o *forwarding* o administrador deverá executar o comando “`echo "1" >/proc/sys/net/ipv4/ip_forward`” no prompt de comando.

### 6.5 Criando as Regras

O processo de criação das regras é a etapa mais importante na construção de um *firewall,* pois, são elas que irão determinar qual a ação que um pacote irá tomar ao ser analisado. (ROCHA JUNIOR, 2010)

A sintaxe do iptables é simples e bem sugestiva. São passados parâmetros que informam a tabela e *chain* que se deseja manipular, endereços de origem/destino, portas

de origem/destino, protocolos, entre outros. (ROCHA JUNIOR, 2010)

Para criar uma regra, basta o administrador digitar no prompt de comando como usuário root a seguinte sintaxe:

```
iptables – t <tabela> <opção> <chain> <parâmetro(s)> -j <alvo>
```

A partir do momento que o administrador digita o comando acima, passando as opções e parâmetros, a regra já começa a valer. O que é mais recomendado e que muitos administradores fazem é criar um *script* contendo todas as regras do *firewall* e em seguida esse arquivo é colocado para ser executado durante a inicialização do Sistema Operacional. Caso o administrador necessite adicionar uma nova regra ou alterar uma já existente, deve-se realizar a alteração no *script* e então executá-lo novamente. (ROCHA JUNIOR, 2010)

Caso alguma regra esteja com erro de sintaxe será apresentado um erro quando a mesma for executada. O administrador de rede deve ter uma atenção especial com a ordem que as regras serão criadas. O iptables interpreta as regras na ordem em que elas são criadas. Caso exista uma regra para um determinado pacote e o mesmo seja verificado por ela, o pacote não será mais inspecionado por outras regras. (ROCHA JUNIOR, 2010)

### 6.5.1 *Opções e Parâmetros do Iptables*

Nesta subseção serão apresentadas as principais opções e parâmetros para a criação das regras do iptables. Para um melhor entendimento as opções e parâmetros foram divididos em três quadros que serão apresentados a seguir.

A tabela 2 apresenta as opções para manipulação de chains e tabelas. Para manipulação das *chains* as opções são definidas com letras maiúsculas e para manipulação da tabela é utilizado letra minúscula. Caso o administrador não especifique a tabela que deseja manipular, o iptables por padrão irá utilizar a tabela *filter.* (ROCHA JUNIOR, 2010)

Tabela 2: Manipulação de Chains e Tabelas.

| TIPO | OPÇÃO | DESCRIÇÃO |
|---|---|---|
| **Manipulando Chains e Tabelas** | **-A** | Adiciona uma nova regra |
| | **-I** | Insere uma regra, é uma linha específica |
| | **-L** | Lista as regras de uma tabela específica |
| | **-D** | Exclui uma determinada regra |
| | **-R** | Substitui uma regra |
| | **-N** | Cria uma nova chain |
| | **-E** | Renomeia uma chain |
| | **-F** | Limpa as regras de uma tabela específica |
| | **-X** | Exclui uma chain criada pelo usuário |
| | **-Z** | Zera os contadores de uma tabela específica |
| | **-P** | Altera a política padrão |
| | **-t** | Especifica uma tabela |

Na Tabela 3 são apresentados os parâmetros para especificação de protocolos, portas de origem/destino, interfaces de entrada/saída e endereços ips de origem/destino. Caso o administrador necessite especificar as portas de origem e destino, ele tem que obrigatoriamente especificar o tipo de protocolo. (ROCHA JUNIOR, 2010)

Tabela 3: Parâmetros Iptables.

| ALVO | OPÇÃO | DESCRIÇÃO |
|---|---|---|
| **Protocolos e Portas** | **-sport** | Especifica porta de origem (Ex: 80, 22, 20, etc.) |
| | **-p** | Especifica o tipo do protocolo (Ex: tcp, udp, icmp) |
| | **-dport** | Especifica porta de destino (Ex: 80, 22, 20, etc.) |
| **Interfaces** | **-i** | Especifica a interface de entrada (Ex: eth0, eth1, etc.) |
| | **-o** | Especifica a interface de saída (Ex: eth0, eth1, etc.) |
| **Endereços IP's** | **-s** | Especifica o endereço ip de origem |
| | **-d** | Especifica o endereço ip de destino |

Na tabela 4 são apresentadas as opções de alvo do pacote, ou seja, qual o destino que o dado deverá ter se estiver de acordo com a regra criada. Para especificar o alvo é utilizada a opção "–j" ao final da regra. A diferença entre as opções *DROP* e *REJECT* é que a *REJECT* retorna uma mensagem de erro ao barrar o dado enquanto a *DROP* simplesmente descarta o pacote. (ROCHA JUNIOR, 2010)

Tabela 4: Opções de Alvo do Pacote.

| ALVO | DESCRIÇÃO |
|---|---|
| **ACCEPT** | Aceita a entrada/passagem do pacote |
| **DROP** | Descarta a entrada/passagem do pacote |
| **REJECT** | Rejeita o pacote |
| **LOG** | Cria um log de acesso |
| **SNAT** | Altera o endereço de origem |
| **DNAT** | Altera o endereço de destino |
| **MASQUERADE** | Realiza o mascaramento ip |
| **REDIRECT** | Realiza o redirecionamento de portas |
| **TOS** | Prioriza a entrada/saída de pacotes |

### 6.5.2 *Sintaxe e Exemplos*

Como podemos perceber, as opções do iptables são bem sugestivas, por exemplo, a opção para indicar o endereço de origem (-s) vem da palavra *source* (que significa origem em inglês). (ROCHA JUNIOR, 2010)

Quanto mais for detalhada a regra, ou seja, quanto mais opções e parâmetros a regra tiver, maior será o seu nível de segurança, pois mais parâmetros devem estar de acordo para que os pacotes sejam aceitos ou descartados. (ROCHA JUNIOR, 2010)

Esta subseção tem objetivo de explicar detalhadamente a sintaxe de algumas regras criadas com base no iptables, apresentando alguns exemplos de regras e os mesmos serão explicados passo a passo.

**Ex.1 #iptables –t filter –A FORWARD –s 192.168.10.10 –d 10.83.1.10 –p tcp --dport 22 –j ACCEPT**

No exemplo 1 o administrador criou uma regra na tabela *filter* (-t), na *chain FORWARD* (-A), especificando o endereço de origem (-s) 192.168.10.10, de destino (-d) 10.83.1.10, com o protocolo (-p) tcp, com destino a porta (--dport) 22 e com alvo (-j) *ACCEPT*. Esta regra libera pacotes que irão passar pelo *firewall* caso os parâmetros do mesmo estejam de acordo com os da regra.

**Ex.2 #iptables –t filter -A INPUT -p tcp --dport 22 -j LOG --log-prefix "FIREWALL:SSH"**

No exemplo 2 o administrador criou uma regra na tabela *filter* (-t), *chain INPUT* (-A), especificando o protocolo tcp (-p), como destino a porta 22 (--dport 22) e com alvo (-j) *LOG.* O parâmetro "*--log-prefix*" serve para salvar os *logs* com um nome específico, no caso desse exemplo o nome "FIREWALL:SSH". Esta regra irá realizar o *log* das tentativas de acesso a porta 22 que utilizarem o protocolo tcp.

**Ex.3 #iptables -t nat -A POSTROUTING –s 192.168.10.202 –o eth0 –j SNAT --to 200.199.103.34**

No exemplo 3 o administrador criou uma regra na tabela *nat* (-t), *chain POSTROUTING* (-A), especificando o endereço de origem (-s) 192.168.10.202, interface de saída (-o) eth0 e alvo (-j) *SNAT* para o endereço 200.199.103.34. Esta regra realiza a substituição para o endereço ip 200.199.103.34 caso o endereço de origem seja o 192.168.10.202 e a interface de saída seja a eth0.

**Ex.4 Alterar a política padrão:**

- **iptables –t filter –P INPUT DROP**
- **iptables –t filter –P OUTPUT DROP**
- **iptables –t filter –P FORWARD DROP**

No exemplo 4 o administrador está alterando a política padrão (-P), da tabela *filter* (-t), das chains *INPUT, OUTPUT* e *FORWARD* para *DROP.* Por padrão o iptables vem com a sua política padrão para *ACCEPT*, e o administrador tem que ir bloqueando o tráfego assim como ele desejar, porém, quando se quer atingir um nível mais alto de segurança é aconselhável que a política padrão seja alterada para *DROP* e o administrador ir liberando o tráfego de acordo com a sua necessidade.

**Ex.5 iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -d 200.199.103.35 -j DNAT --to-destination 172.10.10.35:80**

No exemplo 5 o administrador está criando uma regra na tabela *nat* (-t), *chain PREROUTING* (-A), interface de entrada (-i) eth0, protocolo (-p) tcp, com destino a porta (--dport) 80, endereço de destino (-d) 200.199.103.35 e o alvo (-j) *DNAT* para o destino 172.10.10.35:80. Esta regra realiza o redirecionamento das requisições com destino ao ip 200.199.103.35 e porta 80 para o endereço 172.10.10.35 porta 80.

# 7 IMPLEMENTAÇÃO DOS FIREWALLS

Este capítulo apresentará os procedimentos, materiais utilizados para a implementação e testes do projeto. Serão mostrados os componentes dos *hardwares* em que foram implementados os *firewalls*; os sistemas operacionais utilizados; a relação de computadores e ips; a topologia da rede; a política de segurança; as regras que foram implementadas em cada arquitetura; os *logs* de acessos e o desempenho das máquinas após as implementações.

## 7.1 Cenário

Na implementação o cenário utilizado é baseado no cénário da empresa COMEXIM LTDA, ele é composto por 19 estações que são divididas entre os departamentos da empresa; 1 Access Point; 1 Servidor de dados, que também executa as funções de DNS e FTP; 1 Roteador Cisco e um Servidor Linux com a função de Firewall utilizando Iptables.

## 7.2 Materiais

### 7.2.1 *Computador utilizado para o Firewall Iptables*

A tabela 5 mostra os componentes da máquina em que o Iptables será instalado.

Tabela 5: Hardware Iptables.

| Firewall Iptables | |
|---|---|
| **Componentes** | **Especificação** |
| Processador | Intel(R) Pentium(R) Dual CPU E2180 @ 2.00GHz |
| HD | HD PATA 80 GB |
| Memória Ram | 2 GB |
| Placa de Rede | Realtek Semiconductor Co., Ltd. RTL8101E/RTL8102E PCI |
| Placa de Rede | VIA Technologies, Inc. VT6105/VT6106S [Rhine-III] |
| Sistema Operacional | Linux debian 2.6.32-5-686 #1 SMP UTC 2011 i686 GNU/Linux |

### 7.2.2 *Roteador Cisco*

A tabela 6 exibe os componentes do roteador no qual foram configuradas as ACL's

.

Tabela 6: Componentes do Roteador Cisco.

| Roteador Cisco 2811 |
|---|
| 120pps de Performance |
| Duas portas Fast Ethernet 10/100 Mbps ( RJ-45 ) |
| Porta Console ( RJ45 ) , porta Auxiliar ( RJ45 ) |
| 4 Slots para HWIC, WIC, VWIC ou VIC |
| 1 Slot para NM ou NME |
| 2 Slots para PVDM |
| 2 Slots para AIM |
| Fonte interna AC 110/240V com chaveamento automático |
| Memória FLASH de 64MB expansível até 256MB |
| Memória DRAM de 256MB expansível até 760MB (em 2 Slots) |

## 7.3 Política de Segurança Interna da Empresa

Este tópico apresenta a estrutura de uma Política de Segurança Interna, que foi baseada na política de segurança da empresa COMEXIM LTDA escrita por SCAGLIONE (2013).

### 7.3.1 *Introdução*

A segurança é um dos assuntos mais importantes dentre as preocupações de qualquer empresa. Nesse documento apresentaremos um conjunto de instruções e procedimentos para normatizar e melhorar nossa visão e atuação em segurança.

A intenção da área de TI com a publicação da Política de Uso Aceitável é adaptar as restrições com o objetivo de proteger a empresa e nossos colaboradores de ações ilegais ou danosas praticadas por qualquer indivíduo, de forma proposital ou inadvertidamente.

### 7.3.2 *Regras Gerais*

#### 7.3.2.1 Autenticação

Todos os colaboradores terão uma conta de acesso à rede corporativa, a autenticação nos sistemas de informática será através de usuário e senha cadastrados no domínio da empresa.

#### 7.3.2.2 Política de senhas

Como todos os usuários estão cadastrados no domínio, a autenticação através da senha é obrigatória. A escolha da senha é pessoal, mas deve seguir os seguintes critérios:

- Uma senha segura deverá conter no mínimo 6 caracteres alfanuméricos (letras e números);
- As senhas terão um tempo de vida útil de 30 dias, após esse prazo será solicitado no próprio computador do usuário a troca da senha;
- Tudo que for executado com a sua senha será de sua inteira responsabilidade, por isso tome todas as precauções possíveis para manter sua senha secreta;

- Se acaso achar que a sua senha foi descoberta por outro colaborador, entre em contato com a equipe técnica para que seja feita a mudança da senha.

#### 7.3.2.3 Política de e-mail

- Não abra anexos com as extensões .bat, .exe, .src, .lnk e .com se não tiver certeza absoluta de que solicitou esse *e-mail*;
- Não utilize o *e-mail* da empresa para assuntos pessoais;
- Evite anexos muito grandes;
- Utilize sempre sua assinatura criptográfica para a troca de *e-mails*;
- É obrigatória a utilização do *Web Mail* ou dos programas *Mozilla Thunderbird* ou *Microsoft Outlook*;
- É proibido fomentar corrente de *e-mails* sobre fotos, vídeos, pornografia, racismo, apologia ao crime, piadas, e demais conteúdos não relacionados ao desenvolvimento das tarefas diárias no local de trabalho.

#### 7.3.2.4 Política de acesso à Internet

- O uso recreativo da *internet* não deverá se dar no horário de expediente;
- Acesso a *sites* com conteúdo pornográfico, jogos, bate-papo, apostas, *youtube*, redes sociais como *facebook, twitter, orkut*, etc e sites para *downloads* como o *4shared*, está bloqueado e monitorado;
- É proibido o uso de *Messengers e Skypes* com contas pessoais;
- Apenas a Rede Sem Fio localizada na Sala de Reuniões, será liberada para qualquer tipo de acesso.

#### 7.3.2.5 Política de uso da estação de trabalho

- Cada usuário possui uma conta no *Active Directory*, onde são salvos todos os dados do seu perfil e registrados os *logs* de acesso;
- Mantenha na sua estação somente o que for supérfulo ou pessoal. Todos os dados relativos à empresa devem ser salvos na unidade Z: que é uma pasta mapeada do servidor, onde existe um sistema de *backup* diário;
- Antes de ausentar-se do seu local de trabalho, o usuário deverá efetuar o

*logout/logoff* da rede ou bloqueio do computador através de senha;

- Não é permitido criar e/ou remover arquivos fora da área alocada ao usuário e/ou que venham a comprometer o desempenho e funcionamento dos sistemas;
- A pasta PÚBLICA, não deverá ser utilizada para armazenamento de arquivos que contenham assuntos sigilosos ou de natureza sensível, devem ser armazenadas apenas informações comuns a todos;
- Cada estação possui apenas os programas necessários para o bom funcionamento do computador e para o trabalho do colaborador;
- É proibido o uso de programas de *download* como *ares, emule, utorrent,* etc;
- Alterações das configurações de rede e instalação de programas são bloqueadas para todos os usuários, sendo assim, caso haja necessidade de alguma alteração, o usuário deverá procurar a equipe técnica;
- É proibida a abertura de computadores para qualquer tipo de reparo, caso seja necessário o reparo deverá ocorrer pelo departamento técnico.

#### 7.3.2.6 Política Social

- Não fale sobre a política de segurança da empresa com terceiros ou em locais públicos;
- Não diga sua senha a ninguém. Nossa equipe técnica jamais irá pedir sua senha;
- Não digite suas senhas ou usuários em máquinas de terceiros;
- Somente aceite ajuda técnica de um membro de nossa equipe técnica previamente apresentado e identificado;
- Nunca execute procedimentos técnicos cujas instruções tenham chego por *e-mail.*

### 7.3.3 *Ciência da Política de Segurança*

Colaborador:

_____

Cargo:

_____

O colaborador acima nominado declara, para os fins de Direito, livre de qualquer impedimento, que por serem de propriedade da empresa, todos os equipamentos, sistemas, acessos à rede corporativa e *e-mails* corporativos, somente poderão ser

utilizados para o desempenho das funções profissionais correspondentes as atividades da empresa, e nos seus limites, conforme a legislação pertinente e normas internas, no interesse da empresa.

Assim sendo, o colaborador reconhece a legitimidade da empresa para monitorar suas atividades laborativas, com a finalidade de manutenção da ordem e segurança pessoal de seus colaboradores, bem como da integridade da rede corporativa; dos equipamentos e sistemas de sua propriedade.

## 7.4 Estrutura Organizacional

Neste tópico será exibida de forma física e lógica a infraestrutura da rede da organização onde será implementado este projeto.

### 7.4.1 *Infraestrutura Física*

A Figura 15 mostra como a rede é dividida fisicamente e os endereços IPs das máquinas.

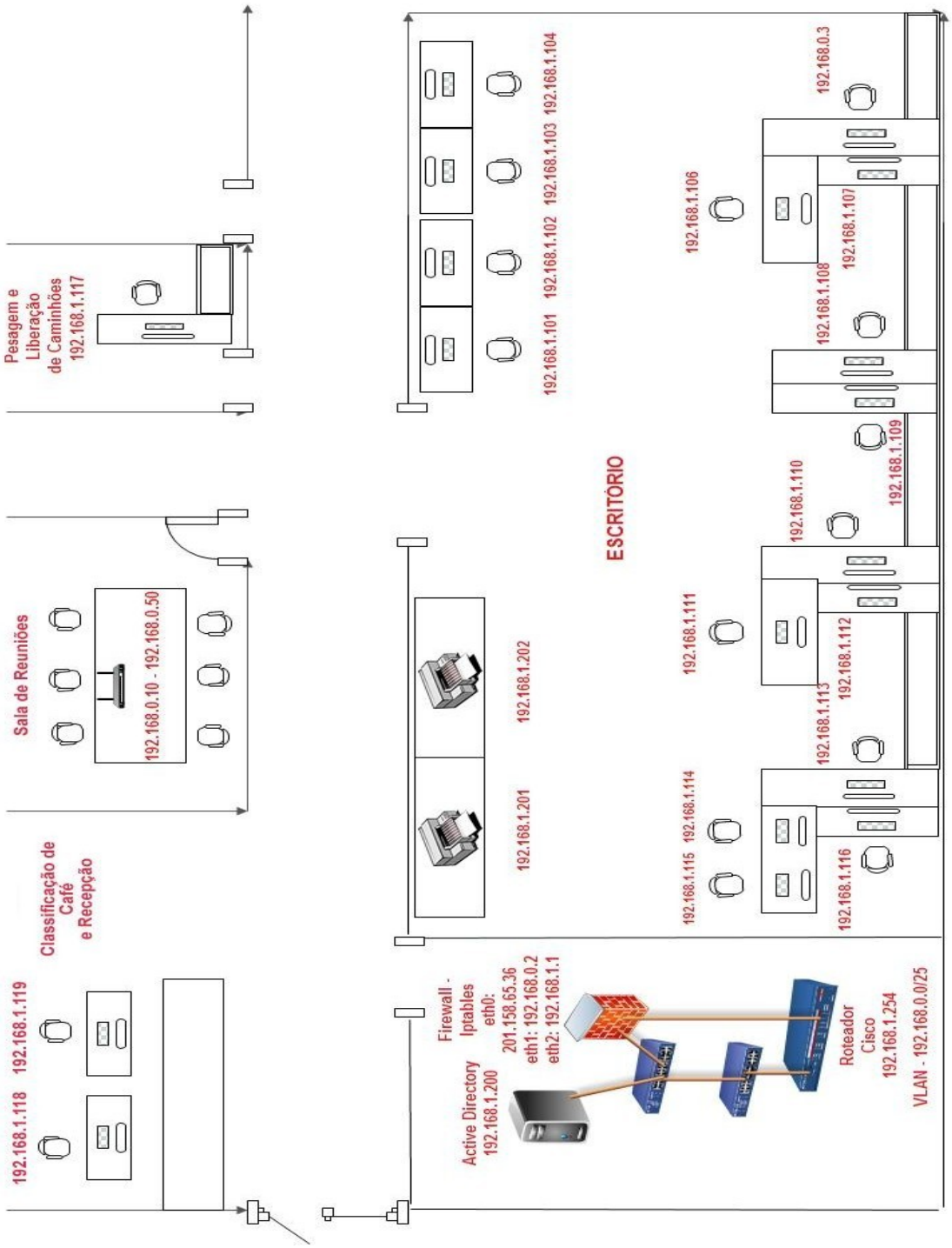


Figura 15: Infraestrutura Física da Rede.
Fonte: Própria Autora

### 7.4.2 *Infraestrutura Lógica*

A Figura 16 apresenta a Infraestrutura Lógica e a divisão de departamentos da empresa.

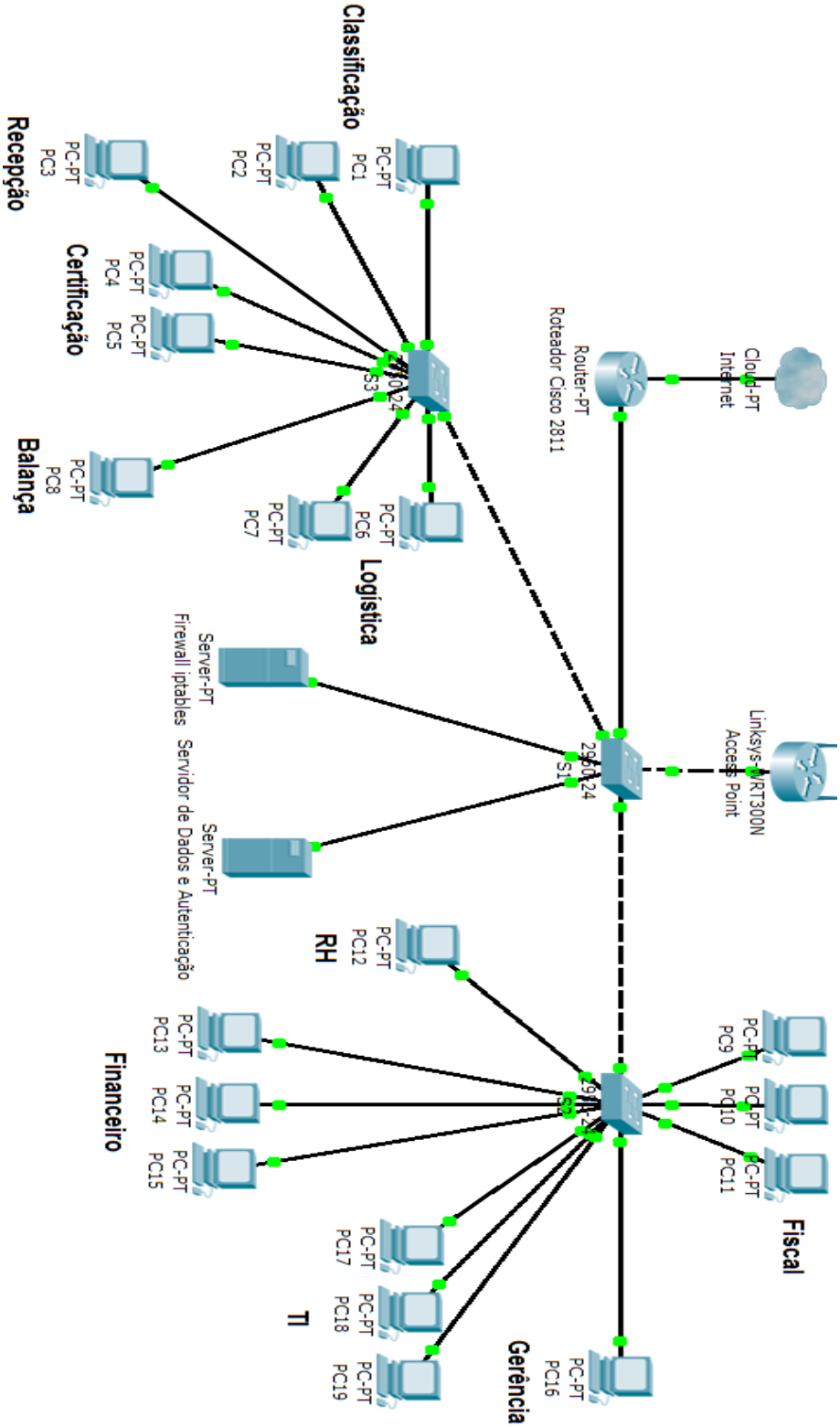


Figura 16: Infraestrutura Lógica da Rede.
Fonte: Própria Autora

## 7.5 Tabela de Endereçamento

Na implementação as duas tecnologias foram testadas separadamente, dessa forma, em cada implementação as máquinas foram configuradas com o gateway específico do Roteador Cisco ou do Firewall Iptables.

A tabela 7 apresenta a relação entre os departamentos da empresa, os componentes da rede e os respectivos Endereços IPs. Nessa tabela, os gateways especificados são os endereços IP do Roteador Cisco, porém na implementação houve também o uso dos endereços IP do Firewall Iptables como gateway das máquinas.

Tabela 7: Relação entre Máquinas, Departamentos e IPs.

| Dispositivo | Interface | Endereço IP | Máscara de sub-rede | Gateway Padrão |
|---|---|---|---|---|
| Roteador Cisco 2811 | Fa0/0 | 201.65.158.34 | 255.255.255.40 | 201.65.158.33 |
| | Fa0/1 | N/A | N/A | N/A |
| | VLAN 1 Fa0/1.1 | 192.168.1.1 | 255.255.255.0 | N/A |
| | VLAN 2 Fa0/1.2 | 192.168.0.1 | 255.255.255.0 | N/A |
| Firewall Iptables | Eth0 | 201.65.158.36 | 255.255.255.40 | 201.65.158.34 |
| | Eth1 | 192.168.1.2 | 255.255.255.0 | 192.168.1.1 |
| | Eth2 | 192.168.0.2 | 255.255.255.0 | 192.168.0.1 |
| Access Point | WAN | 192.168.0.3 | 255.255.255.0 | 192.168.0.1 |
| PC1/Classificação | Placa de Rede | 192.168.1.3 | 255.255.255.0 | 192.168.1.1 |
| PC2/Classificação | Placa de Rede | 192.168.1.4 | 255.255.255.0 | 192.168.1.1 |
| PC3/Recepção | Placa de Rede | 192.168.1.5 | 255.255.255.0 | 192.168.1.1 |
| PC4/Certificação | Placa de Rede | 192.168.1.6 | 255.255.255.0 | 192.168.1.1 |
| PC5/Certificação | Placa de Rede | 192.168.1.7 | 255.255.255.0 | 192.168.1.1 |
| PC6/Logística | Placa de Rede | 192.168.1.8 | 255.255.255.0 | 192.168.1.1 |
| PC7/Logística | Placa de Rede | 192.168.1.9 | 255.255.255.0 | 192.168.1.1 |
| PC8/Balança | Placa de Rede | 192.168.1.10 | 255.255.255.0 | 192.168.1.1 |
| PC9/Fiscal | Placa de Rede | 192.168.1.11 | 255.255.255.0 | 192.168.1.1 |
| PC10/Fiscal | Placa de Rede | 192.168.1.12 | 255.255.255.0 | 192.168.1.1 |
| PC11/Fiscal | Placa de Rede | 192.168.1.13 | 255.255.255.0 | 192.168.1.1 |
| PC12/RH | Placa de Rede | 192.168.1.14 | 255.255.255.0 | 192.168.1.1 |
| PC13/Financeiro | Placa de Rede | 192.168.1.15 | 255.255.255.0 | 192.168.1.1 |
| PC14/Financeiro | Placa de Rede | 192.168.1.16 | 255.255.255.0 | 192.168.1.1 |
| PC15/Financeiro | Placa de Rede | 192.168.1.17 | 255.255.255.0 | 192.168.1.1 |
| PC16/Gerência | Placa de Rede | 192.168.0.4 | 255.255.255.0 | 192.168.0.1 |
| PC17/TI | Placa de Rede | 192.168.1.19 | 255.255.255.0 | 192.168.1.1 |
| PC18/TI | Placa de Rede | 192.168.1.20 | 255.255.255.0 | 192.168.1.1 |
| PC19/TI | Placa de Rede | 192.168.1.21 | 255.255.255.0 | 192.168.1.1 |
| Serv.de Dados/DNS/FTP | Placa de Rede | 192.168.1.50 | 255.255.255.0 | 192.168.1.1 |

## 7.6 Política de Regras

Para a Sub Rede 192.168.1.0/24 haverão as seguintes regras:

- Sites que serão bloqueados: *facebook, twitter, 4shared, youtube, orkut, hotmail*, etc;
- Portas TCP Liberadas : FTP, SMTP, HTTP, HTTPS, DNS, POP3;
- Porta UDP Liberada: DNS;
- Todos os usuários da rede deverão ter acesso ao servidor de dados 192.168.1.50;
- Para liberar o acesso remoto ao servidor linux a porta SSH deve ser liberada para os *hosts* 192.168.1.19/24 e 192.168.1.20/24 do departamento de TI;
- Os departamentos Classificação, Certificação, Balança e Logística precisam de permissão para acessar um programa via VPN na rede 192.168.10.0/24;
- O computador da Recepção não terá acesso à *internet*, deste modo serão bloqueadas as portas HTTP e HTTPS;
- Para facilitar o gerenciamento e a solução de problemas de rede foi permitida a entrada de ICMP na rede;
- Liberação da porta 3456 da receita federal para declaração de imposto de renda;
- Liberação da porta 5160 do protocolo SIP que trabalha na camada de aplicação, cuja função é iniciar, modificar ou terminar sessões ou chamadas multimídia entre usuários, estas sessões podem ser videoconferências, aulas pela *internet*, telefonia sobre *internet*, etc;
- A porta 123 do protocolo NTP usada para sincronização de horário na rede deve ser liberada.
- Para estabelecer conexões remotas, a porta 3389 deve estar liberada.

A Sub Rede 192.168.0.0/24 seguirá as seguintes regras:

- Não haverá bloqueio de *sites*;
- Portas liberadas: SMTP, FTP, HTTP, HTTPS, POP3, DNS;
- Porta UDP Liberada: DNS;
- Para facilitar o gerenciamento e a solução de problemas de rede foi permitida a entrada de ICMP na rede;
- Os usuários dessa rede não terão acesso ao servidor de dados.

## 7.7 Metodologia de Teste

Para testar o funcionamento das duas tecnologias, os testes foram executados durante 60 minutos, as cargas de teste utilizadas na rede 192.168.1.0/24 foram: acesso aos sites bloqueados (facebook, youtube, 4shared, twitter, orkut, hotmail, ebay, outlook, msn); acesso web para teste das portas 80, 443 e 53; envio e recebimento de e-mails para testar o funcionamento das portas 110 e 587; acesso FTP na porta 21; acesso remoto SSH que utiliza a porta 22; acesso remoto ao Servidor de Dados e a rede 192.168.10.0/24 para verificar o funcionamento da VPN; acesso web no computador da recepção, para comprovar o bloqueio das portas 80 e 443 e utilização do comando ping para conferir o funcionamento do protocolo ICMP.

Para a rede 192.168.0.0/24 os testes foram os seguintes: acesso web para teste das portas 80, 443 e 53; envio e recebimento de e-mails para testar o funcionamento das portas 110 e 587; acesso FTP na porta 21, tentativa de acesso ao Servidor de Dados, para verificar o funcionamento da regra de bloqueio.

## 7.8 Implementação com Iptables

Para a configuração do Iptables foi criado um script com todas as regras e para este *script* ser inicializado com o sistema, foi criada uma cópia do arquivo no diretório /etc/init.d.

- **Interfaces da máquina**

Eth0 – WAN

Eth1 – LAN

Eth2 – LAN 2

- **Módulos necessários para o funcionamento**

Para o funcionamento do Iptables é necessário carregar os seguintes módulos:

```
modprobe ip_tables
modprobe ip_conntrack
modprobe iptable_filter
modprobe iptable_nat
modprobe iptable_mangle
modprobe ipt_LOG
```

```
modprobe ipt_limit
modprobe ipt_state
modprobe ipt_MASQUERADE
modprobe ip_nat_ftp
modprobe ip_nat_irc
modprobe ip_conntrack_ftp
modprobe ip_conntrack_irc
```

- **Setando as políticas padrões**

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P POSTROUTING ACCEPT
```

- **Habilitando o mascaramento NAT para compartilhar a internet**

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- **Bloqueio de sites com Iptables**

No iptables, há 3 diferentes formas de bloquear *sites:*

- **Bloqueia o acesso à web a partir de um determinado IP**

```
iptables -A FORWARD -p tcp -s 192.168.1.67 -j REJECT
```

Esta regra deverá estar no início do *script*, antes das regras que abrem as portas de saída, caso contrário não surtirá efeito. O Iptables processa as regras sequencialmente: se uma compartilha a conexão com todos os micros da rede, não adianta tentar bloquear para determinados endereços depois. As regras com as exceções devem sempre vir antes da regra mais geral.

- **Bloqueio por domínios**

É possível ainda bloquear ou permitir com base no domínio, tanto para entrada quanto saída. Isso permite bloquear *sites* e programas diretamente a partir do *firewall*,

sem precisar instalar um servidor *Squid*. Nesse caso, usamos o parâmetro “-d” (destiny) do Iptables, seguido do domínio desejado.

Para bloquear os acessos ao *Facebook,* por exemplo, as regras seriam as seguintes:

```
iptables -A OUTPUT -d http://www.facebook.com -j REJECT
iptables -A FORWARD -d http://www.facebook.com -j REJECT
```

- **Conferindo o conteúdo do pacote**

O módulo *string* do iptables permite a inspeção de conteúdo de um pacote e define a ação se determinado tipo de tráfego for encontrado em um pacote. Esta técnica pode ser usada tanto para segurança como para economia de banda dentro da rede. O *firewall* em nível de pacotes fazendo inspeção de conteúdo, chega a ser 3 a 10 vezes mais rápido do que um *proxy*, assim seu uso deve ser analisado dependendo do tráfego que circula pelo *link* e da segurança dos dados que trafegam através dele.

Outra utilidade eficiente é a diminuição de tráfego, pois podemos barrar programas que sobrecarregam o *link* em uma rede com muitos usuários como, por exemplo, usando o *Ares* ou qualquer outro programa para cópia de arquivos via *Internet.* Veja o exemplo:

```
# Bloqueia qualquer tentativa de acesso ao programa Ares
iptables -A INPUT -m string –string “ares” -j DROP
```

Por ser a forma mais completa de análise de pacotes e possuir regras mais abrangentes, o bloqueio de *sites* neste trabalho foi feito através dessa opção.

- **Bloqueio de sites proibidos utilizado na implementação**

```
iptables  -I FORWARD -m string --algo bm --string "facebook" -j DROP
iptables  -I FORWARD -m string --algo bm --string "youtube" -j DROP
iptables  -I FORWARD -m string --algo bm --string "4shared" -j DROP
iptables  -I FORWARD -m string --algo bm --string "twitter" -j DROP
iptables  -I FORWARD -m string --algo bm --string "orkut" -j DROP
iptables  -I FORWARD -m string --algo bm --string "hotmail" -j DROP
iptables  -I FORWARD -m string --algo bm --string "live" -j DROP
iptables  -I FORWARD -m string --algo bm --string "msn" -j DROP
iptables  -I FORWARD -m string --algo bm --string "outlook" -j DROP
```

```
iptables  -I FORWARD -m string --algo bm --string "ebay" -j DROP
```

- **Liberando portas para a rede 192.168.1.0/24**

```
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 8080 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 80 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 443 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 587 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 110 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p udp --dport 53 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 21 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 3456 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 5160 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 123 -j ACCEPT
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 3389 -j ACCEPT
```

- **Liberando acesso SSH aos hosts do departamento de TI**

```
iptables -I FORWARD -s 192.168.1.19/24 -p tcp --dport 22 -j ACCEPT
iptables -I FORWARD -s 192.168.1.20/24 -p tcp --dport 22 -j ACCEPT
```

- **Os departamentos Classificação, Certificação, Balança e Logística precisam de permissão para  acessar um programa via VPN na rede 192.168.10.0/24**

```
iptables -A FORWARD -s 192.168.1.3/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.4/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.6/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.7/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.8/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.9/24 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.1.10/24 -d 192.168.10.0/24 -j ACCEPT
```

- **Bloqueio do acesso à Internet para o computador da Recepção**

```
iptables -I FORWARD -s 192.168.1.5/24 -p tcp --dport 80 -j DROP
iptables -I FORWARD -s 192.168.1.5/24 -p tcp --dport 443 -j DROP
```

- **Liberando ICMP**

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

- **Permitindo acesso ao Servidor de Dados 192.168.1.50 para a rede 192.168.1.0/24**

```
iptables -A FORWARD -s 192.168.1.0/24 -d 192.168.1.50 -j ACCEPT
```

- **Liberando portas para a rede 192.168.0.0/24**

```
iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 80 -j ACCEPT
iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 443 -j ACCEPT
iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 587 -j ACCEPT
iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 110 -j ACCEPT
iptables -I FORWARD -s 192.168.0.0/24 -p udp --dport 53 -j ACCEPT
iptables -I FORWARD -s 192.168.0.0/24 -p tcp --dport 21 -j ACCEPT
```

- **Negando acesso ao Servidor de Dados 192.168.1.50 para a rede 192.168.0.0/24**

```
iptables -A FORWARD -s 192.168.0.0/24 -d 192.168.1.50 -j DROP
```

## 7.9 Implementação com ACL Cisco

Na implementação do *firewall* no Roteador Cisco, serão utilizadas 2 ACL's, uma para cada interface.

**ACL para a interface 1, da rede 192.168.1.0/24:**

**Nome da ACL: ip access-list extended FW-INT1**

- **Bloqueio de sites com ACL**

Com ACL é possível bloquear *sites,* de duas formas: pelo endereço IP do *host* de destino e pela URL do domínio do *site*.

- Bloqueando o acesso pelo endereço IP

Existem *sites* como o *www.facebook.com* que possuem vários servidores, sendo assim, para bloquear *sites* como este pelo endereço do *host* de destino, deve-se bloquear todos os endereços ips a que ele responde, se o site tiver 10 endereços ips, todos deverão ser bloqueados. A desvantagem deste tipo de bloqueio, é que se mais servidores forem adicionados, a regra deverá ser atualizada.

**Exemplo:** `deny ip 192.168.1.0 0.0.0.255 host 74.117.178.90`

A regra acima bloqueia o acesso da rede 192.168.0.0/25 ao endereço IP 74.117.178.90.

- Bloqueio por domínio

O bloqueio através da URL do domínio, é mais seguro, pois são poucos *sites* que respondem por mais de um domínio. Porém, se o *site* possuir mais de um domínio, todos os domínios correspondentes devem ser bloqueados.

**Exemplo:** `deny tcp any host www.youtube.com`

- **Bloqueio de sites proibidos utilizado na implementação**

Na implementação deste trabalho, foram utilizadas as duas formas de bloqueio de *sites*, pois alguns sites como o *youtube* possuem vários domínios como exemplo o site *video.google.com*, assim para maior segurança, alguns *sites* foram bloqueados pelo endereço IP e pela URL do domínio.

- **Bloqueio pela URL**

```
deny tcp any host www.facebook.com eq www
deny tcp any host www.youtube.com eq www
deny tcp any host www.4shared.com eq www
deny tcp any host www.twitter.com eq www
deny tcp any host www.orkut.com eq www
deny tcp any host www.hotmail.com eq www
deny tcp any host www.msn.com eq www
deny tcp any host www.live.com eq www
deny tcp any host www.ebay.com eq www
```

```
deny tcp any host www.outlook.com eq www
deny tcp any host video.google.com eq www
```

- **Bloqueio pelo endereço IP**

**Bloqueio do youtube:**

```
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.192
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.193
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.194
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.195
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.196
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.198
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.200
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.201
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.206
```

**Bloqueio do 4shared:**

```
deny ip 192.168.1.0 0.0.0.255 host 74.117.178.89
deny ip 192.168.1.0 0.0.0.255 host 74.117.178.90
deny ip 192.168.1.0 0.0.0.255 host 74.117.178.91
deny ip 192.168.1.0 0.0.0.255 host 199.101.134.235
```

**Bloqueio do hotmail, live, outlook e msn:**

```
deny ip 192.168.1.0 0.0.0.255 host 65.55.77.28
deny ip 192.168.1.0 0.0.0.255 host 65.55.85.12
deny ip 192.168.1.0 0.0.0.255 host 65.55.72.151
deny ip 192.168.1.0 0.0.0.255 host 65.55.72.183
deny ip 192.168.1.0 0.0.0.255 host 65.55.72.199
deny ip 192.168.1.0 0.0.0.255 host 157.56.172.28
deny ip 192.168.1.0 0.0.0.255 host 157.56.242.182
deny ip 192.168.1.0 0.0.0.255 host 132.245.2.6
deny ip 192.168.1.0 0.0.0.255 host 131.253.13.21
```

```
deny ip 192.168.1.0 0.0.0.255 host 157.56.242.182

deny ip 192.168.1.0 0.0.0.255 host 157.56.238.6

deny ip 192.168.1.0 0.0.0.255 host 157.56.236.102
```

- **Liberando portas para a rede 192.168.1.0/24**

```
permit tcp 192.168.1.0 0.0.0.255 any eq 21
permit tcp 192.168.1.0 0.0.0.255 any eq 80
permit tcp 192.168.1.0 0.0.0.255 any eq 443
permit udp 192.168.1.0 0.0.0.255 any eq 53
permit tcp 192.168.1.0 0.0.0.255 any eq 587
permit tcp 192.168.1.0 0.0.0.255 any eq 110
permit tcp 192.168.1.0 0.0.0.255 any eq 123
permit tcp 192.168.1.0 0.0.0.255 any eq 5160
permit tcp 192.168.1.0 0.0.0.255 any eq 3456
permit tcp 192.168.1.0 0.0.0.255 any eq 3389
```

- **Permitindo acesso ao Servidor de Dados 192.168.1.50 para a rede 192.168.1.0/24**

```
permit ip 192.168.1.0 0.0.0.255 host 192.168.1.50
```

- **Liberando acesso SSH aos hosts do departamento de TI**

```
permit tcp any host 192.168.1.19 eq 22
permit tcp any host 192.168.1.20 eq 22
```

- **Liberando ICMP**

```
permit icmp any any echo-reply

permit icmp any any unreachable
```

- **Os departamentos Classificação, Certificação, Balança e Logística precisam de permissão para acessar um programa via VPN na rede 192.168.10.0/24**

```
permit tcp host 192.168.1.3 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.4 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.6 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.7 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.8 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.9 192.168.10.0 0.0.0.255
permit tcp host 192.168.1.10 192.168.10.0 0.0.0.255
```

- **Bloqueio do acesso à Internet para o computador da Recepção**

```
deny tcp any host 192.168.1.5  eq 80
deny tcp any host 192.168.1.5  eq 443
```

**ACL para a interface 2, da rede 192.168.0.0/24:**

**Nome da ACL: ip access-list extended FW-INT2**

- **Liberando portas para a rede 192.168.0.0/24**

```
permit tcp 192.168.1.0 0.0.0.255 any eq 21
permit tcp 192.168.1.0 0.0.0.255 any eq 80
permit tcp 192.168.1.0 0.0.0.255 any eq 443
permit udp 192.168.1.0 0.0.0.255 any eq 53
permit tcp 192.168.1.0 0.0.0.255 any eq 587
permit tcp 192.168.1.0 0.0.0.255 any eq 110
```

- **Liberando ICMP**

```
permit icmp any any echo-reply

permit icmp any any unreachable
```

- **Negando acesso ao Servidor de Dados 192.168.1.50 para a rede 192.168.0.0/24**

```
deny ip 192.168.0.0 0.0.0.255 host 192.168.1.50
```

## 7.10 Análise do Registro de Logs

Para saber se as regras foram executadas e a quantidade de vezes que ela foi chamada é necessário consultar o registro de *logs.*

Para a obtenção dos *logs* dos dois tipos de *firewall,* as implementações foram testadas durante 60 minutos.

### 7.10.1 *Registro de Logs ACL Cisco*

Para obter os registros de *logs* das regras basta inserir a palavra "*log*" ao final da ACL.

**Logs da ACL da interface 1, rede 192.168.1.0/24:**

**Nome da ACL: ip access-list extended FW-INT1**

- **Logs de Sites Bloqueados**

**Bloqueio do youtube:**

```
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.192 (98 acessos)
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.193
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.194 (103 acessos)
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.195 (64 acessos)
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.196
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.198 (31 acessos)
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.199
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.200
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.201
deny ip 192.168.1.0 0.0.0.255 host 74.125.234.206 (5 acessos)
```

**Bloqueio do Facebook:**

```
deny ip 192.168.1.0 0.0.0.255 host 31.13.71.33  (65 acessos)
deny ip 192.168.1.0 0.0.0.255 host 31.13.73.113 (29 acessos)
deny ip 192.168.1.0 0.0.0.255 host 66.220.152.19 (6 acessos)
deny ip 192.168.1.0 0.0.0.255 host 199.101.134.235
```

- **Logs de Portas Liberadas**

```
permit tcp 192.168.1.0 0.0.0.255 any eq 21 log (13 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 80 log (1838 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 443 log (1411 acessos)
permit udp 192.168.1.0 0.0.0.255 any eq 53 log (579 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 587 log (137 acessos)
```

```
permit tcp 192.168.1.0 0.0.0.255 any eq 110 log (1199 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 123 log (4 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 5160 log (7 acessos)
permit tcp 192.168.1.0 0.0.0.255 any eq 3389 log (11 acessos)
```

- **Log de acesso ao Servidor de Dados 192.168.1.50**

```
permit ip 192.168.1.0 0.0.0.255 host 192.168.1.50 log (10 acessos)
```

- **Log de acesso SSH vindos dos hosts do departamento de TI**

```
permit tcp any host 192.168.1.19 eq 22 log (6 acessos)
permit tcp any host 192.168.1.20 eq 22  log(4 acessos)
```

- **Log ICMP**

```
permit icmp any any echo-reply log (37 acessos)
permit icmp any any unreachable log (14 acessos)
```

- **Log de acesso dos departamentos Classificação, Certificação, Balança e Logística que acessam um programa via VPN na rede 192.168.10.0/24**

```
permit tcp host 192.168.1.3 192.168.10.0 0.0.0.255 log (2 acessos)
permit tcp host 192.168.1.4 192.168.10.0 0.0.0.255 log (5 acessos)
permit tcp host 192.168.1.6 192.168.10.0 0.0.0.255 log
permit tcp host 192.168.1.7 192.168.10.0 0.0.0.255 log
permit tcp host 192.168.1.8 192.168.10.0 0.0.0.255 log
permit tcp host 192.168.1.9 192.168.10.0 0.0.0.255 log  (4 acessos)
permit tcp host 192.168.1.10 192.168.10.0 0.0.0.255 log
```

- **Log de acesso à Internet do computador da Recepção**

```
deny tcp any host 192.168.1.5  eq 80 log (53 acessos)
deny tcp any host 192.168.1.5  eq 443 log (27 acessos)
```

**Logs da ACL da interface 2, rede 192.168.0.0/24:**
**Nome da ACL: ip access-list extended FW-INT2**

- **Logs de Portas Liberadas**

```
permit tcp 192.168.0.0 0.0.0.255 any eq 21 (11 acessos)
permit tcp 192.168.0.0 0.0.0.255 any eq 80 (1622 acessos)
permit tcp 192.168.0.0 0.0.0.255 any eq 443 (237 acessos)
permit udp 192.168.0.0 0.0.0.255 any eq 53 (270 acessos)
```

```
permit tcp 192.168.0.0 0.0.0.255 any eq 587 (91 acessos)
permit tcp 192.168.0.0 0.0.0.255 any eq 110 (213 acessos)
```

- **Liberando ICMP**

```
permit icmp any any echo-reply (22 acessos)
permit icmp any any unreachable (19 acessos)
```

### 7.10.2 *Registro de Logs Iptables*

Para que os *logs* das regras iptables sejam salvos a regra deverá estar no seguinte formato:

```
iptables -I FORWARD -s 192.168.1.0/24 -p tcp --dport 80 -j LOG --log-prefix "acesso a internet"
```

Para uma visualização melhor dos arquivos de logs do Iptables, foi utilizada uma ferramenta chamada *Fwlogwatch* que é uma ferramenta usada para análise de registros de *logs* e geração de relatórios personalizados.

Essa ferramenta gera relatórios a partir das entradas de *log* de arquivos especificados via linha de comando ou arquivo de configuração, deixando os *logs* do Iptables em formatos facilmente legíveis, em texto puro ou no formato HTML.

A tabela 8 Mostra detalhadamente os *logs* das regras.

Tabela 8: Logs Iptables

| Interf. | Protoc. | Origem | Porta de Origem | Destino | Porta de Destino | Status | Acessos |
|---|---|---|---|---|---|---|---|
| Eth0 | tcp | 201.65.158.36 | 41022/41023 | 192.168.0.1/24 | 443 HTTPS | Aceito | 1947 |
| Eth1 | udp | 192.168.1.2 | 37384 | 192.168.1.50 | 53 DNS | Aceito | 668 |
| Eth0 | tcp | 201.65.158.36 | 53097 | 192.168.0.1/24 | 80 HHTP | Aceito | 2273 |
| Eth1 | tcp | 192.168.1.0/24 | 110 POP3 | 200.234.210.12 Servidor de E-mails | 2198 | Aceito | 1525 |
| Eth1 | tcp | 200.234.210.12 Servidor de E-mails | 587 SMTP | 192.168.0.1/24 | 2207 | Aceito | 930 |
| Eth1 | tcp | 192.168.1.19 | 1018 | 192.168.1.2 | 22 SSH | Aceito | 13 |
| Eth1 | tcp | 192.168.1.0/24 | 1590 | 192.168.1.50 | 21 FTP | Aceito | 34 |
| Eth0 | icmp | 201.65.158.36 | 3 | 192.168.0.1/24 | 1 | Aceito | 88 |
| Eth0 | tcp | 201.65.158.36 | 36891 | 74.125.234.237 Youtube | 80 HTTP | Bloqueado | 49 |
| Eth0 | tcp | 74.125.234.197 Youtube | 80 HTTP | 192.168.0.1/24 | 2036 | Bloqueado | 57 |
| Eth0 | tcp | 74.125.234.192 Youtube | 80 HTTP | 192.168.0.1/24 | 2052 | Bloqueado | 196 |
| Eth0 | tcp | 74.125.234.199 Youtube | 80 HTTP | 192.168.0.1/24 | 2050 | Bloqueado | 109 |
| Eth0 | tcp | 31.13.71.33 Facebook | 443 HTTPS | 192.168.0.1/24 | 1832/1834 | Bloqueado | 79 |
| Eth0 | Tcp | 201.65.158.36 | 58622 | 31.13.71.33 Facebook | 80 HTTP | Bloqueado | 68 |
| Eth0 | tcp | 31.13.73.113 Facebook | 443 | 192.168.0.1/24 | 443 HTTPS | Bloqueado | 45 |
| Eth0 | Tcp | 201.65.158.36 | 58622 | 31.13.73.113 Facebook | 80 HTTP | Bloqueado | 37 |
| Eth1 | Tcp | 192.168.1.19 | 2035 | 192.168.1.50 | 3389 | Aceito | 3 |
| Eth1 | tcp | 192.168.1.5 | 2035 | 192.168.10.5 | 3389 | Aceito | 2 |
| eth1 | tcp | 192.168.1.7 | 2035 | 192.168.10.5 | 3389 | Aceito | |
| Eth0 | tcp | 201.65.158.36 | 41023/41022 | 192.168.0.0/24 | 443 HTTPS | Aceito | 434 |
| Eth2 | udp | 192.168.0.2 | 37384 | 192.168.1.50 | 53 DNS | Aceito | 219 |
| Eth0 | tcp | 201.65.158.36 | 53097 | 192.168.0.0/24 | 80 HTTP | Aceito | 1026 |
| Eth2 | tcp | 192.168.0.0/24 | 110 POP3 | 200.234.210.12 Servidor de E-mails | 2198 | Aceito | 525 |
| Eth2 | tcp | 200.234.210.12 Servidor de E-mails | 587 SMTP | 192.168.0.0/24 | 2207 | Aceito | 317 |
| Eth0 | icmp | 201.65.158.36 | 3 | 192.168.0.0/24 | 1 | Aceito | 14 |

## 7.11 Análise de Consumo de Hardware gerado pelos Firewalls

O uso de máquinas e roteadores como *firewall* causam um aumento no consumo dos recursos dos *hardwares* como cpu e memória, podendo até causar lentidão no funcionamento de outros serviços. Este tópico apresenta o consumo dos *hardwares* antes e durante a execução dos *firewalls*.

### 7.11.1 *Consumo da ACL no Roteador Cisco*

#### 7.11.1.1 Consumo de CPU

Para visualizar os processos que estão no cpu, o seguinte comando deve ser utilizado:

**`show process cpu`**

Quando este comando é executado os seguintes dados são exibidos:

- **Processos antes da execução das ACL's**

```
CPU utilization for five seconds: 1%/0%; one minute: 1%; five minutes: 0%
 PID Runtime(uS)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1       12000          13        923  0.00%  0.00%  0.00%   0 Chunk Manager
   2        4000          16        250  0.00%  0.00%  0.00%   0 Load Meter
   3        4000           1       4000  0.00%  0.00%  0.00%   0 LICENSE AGENT
   4       48000          10       4800  0.63%  0.06%  0.01%   0 Check heaps
   5        4000           2       2000  0.00%  0.00%  0.00%   0 Pool Manager
   6           0           1          0  0.00%  0.00%  0.00%   0 DiscardQ Backgr
   7           0           2          0  0.00%  0.00%  0.00%   0 Timers
   8           0           1          0  0.00%  0.00%  0.00%   0 License Client
   9       40000         115        347  0.00%  0.02%  0.00%   0 Exec
  10           0           1          0  0.00%  0.00%  0.00%   0 IPC Process lev
  11           0           2          0  0.00%  0.00%  0.00%   0 IPC Dynamic Cac
  13           0          20          0  0.00%  0.00%  0.00%   0 IPC Service NonC
  14        4000          95         42  0.00%  0.00%  0.00%   0 IPC Periodic Tim
  15           0          96          0  0.00%  0.00%  0.00%   0 IPC Deferred Por
  18           0          20          0  0.00%  0.00%  0.00%   0 IPC Loadometer
  21           0           4          0  0.00%  0.00%  0.00%   0 Environmental mo
  22        4000          29        137  0.00%  0.00%  0.00%   0 ARP Input
  23        8000         117         68  0.00%  0.00%  0.00%   0 ARP Background
  35           0          87          0  0.00%  0.00%  0.00%   0 GraphIt
  40       12000          65        184  0.00%  0.00%  0.00%   0 Net Background
  42        4000          14        285  0.00%  0.00%  0.00%   0 Logger
  43           0          87          0  0.00%  0.00%  0.00%   0 TTY Background
  44           0         106          0  0.07%  0.01%  0.00%   0 Per-Second Jobs
  45           0           7          0  0.00%  0.00%  0.00%   0 c2800 Periodic
  51           0          33          0  0.00%  0.00%  0.00%   0 HC Counter Timer
  60           0         450          0  0.00%  0.01%  0.00%   0 Netclock Backgro
  65           0         115          0  0.00%  0.00%  0.00%   0 Ether-Switch RBC
```

```
 68           0           5           0  0.00%  0.00%  0.00%   0 Call Management
 73           0         118           0  0.00%  0.00%  0.00%   0 PI MATM Aging Pr
 74           0          13           0  0.00%  0.00%  0.00%   0 EtherChnl
 76           0         116           0  0.00%  0.00%  0.00%   0 Ethernet Timer C
 81        4000          19         210  0.00%  0.00%  0.00%   0 CDP Protocol
 91        4000        3973           1  0.15%  0.11%  0.05%   0 IPAM Manager
 95       56000         119         470  0.00%  0.01%  0.01%   0 IP Input
 98           0           3           0  0.00%  0.00%  0.00%   0 MOP Protocols
106        4000         539           7  0.00%  0.01%  0.00%   0 SSS Feature Time
110        8000           5        1600  0.00%  0.00%  0.00%   0 CEF background p
120           0           3           0  0.00%  0.00%  0.00%   0 DFS flush period
121       16000           4        4000  0.23%  0.02%  0.00%   0 Licensing Auto U
123        4000         152          26  0.00%  0.00%  0.00%   0 CEF: IPv4 proces
129        8000           9         888  0.00%  0.00%  0.00%   0 IP Background
136           0         132           0  0.00%  0.00%  0.00%   0 RUDPV1 Main Proc
138           0         147           0  0.00%  0.00%  0.00%   0 bsm_xmt_proc
159           0           8           0  0.00%  0.00%  0.00%   0 CRM_CALL_UPDATE_
173           0           2           0  0.00%  0.00%  0.00%   0 AAA SEND STOP EV
174           0          15           0  0.00%  0.00%  0.00%   0 RMON Recycle Pro
178           0           2           0  0.00%  0.00%  0.00%   0 DHCPD Timer
180           0           8           0  0.00%  0.00%  0.00%   0 Syslog
182           0          31           0  0.07%  0.00%  0.00%   0 Compute load avg
183       24000           3        8000  0.00%  0.01%  0.00%   0 Per-minute Jobs
187        4000           2        2000  0.00%  0.00%  0.00%   0 DHCPD Receive
188           0         163           0  0.00%  0.00%  0.00%   0 DHCP Client
190        4000         292          13  0.00%  0.00%  0.00%   0 IP NAT Ager
193           0           3           0  0.00%  0.00%  0.00%   0 DHCPD Database
```

- **Processos durante a execução das ACL's**

```
CPU utilization for five seconds: 0%/0%; one minute: 2%; five minutes: 1%
 PID Runtime(uS)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1       24000          17       1411   0.00%  0.00%  0.00%   0 Chunk Manager
   2        4000        1237          3   0.00%  0.00%  0.00%   0 Load Meter
   3        4000           1       4000   0.00%  0.00%  0.00%   0 LICENSE AGENT
   9      540000        2446        220   0.15%  0.07%  0.04%   0 Exec
  10           0           1          0   0.00%  0.00%  0.00%   0 IPC Process leve
  11           0         104          0   0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  13           0        1240          0   0.00%  0.00%  0.00%   0 IPC Service NonC
  14       12000        6053          1   0.00%  0.00%  0.00%   0 IPC Periodic Tim
  15           0        6053          0   0.00%  0.00%  0.00%   0 IPC Deferred Por
  18        4000        1240          3   0.00%  0.00%  0.00%   0 IPC Loadometer
  21           0         208          0   0.00%  0.00%  0.00%   0 Environmental mo
  22      364000        1130        322   0.00%  0.00%  0.00%   0 ARP Input
  23        8000        6478          1   0.00%  0.00%  0.00%   0 ARP Background
  34           0           2          0   0.00%  0.00%  0.00%   0 SMART
  35           0        6186          0   0.00%  0.00%  0.00%   0 GraphIt
  40       16000        3693          4   0.00%  0.00%  0.00%   0 Net Background
  42       12000        1794          6   0.00%  0.00%  0.00%   0 Logger
  43        4000        6183          0   0.00%  0.01%  0.00%   0 TTY Background
  44       12000        6203          1   0.00%  0.01%  0.00%   0 Per-Second Jobs
  45           0         414          0   0.00%  0.00%  0.00%   0 c2800 Periodic
  51        4000        1862          2   0.00%  0.00%  0.00%   0 HC Counter Timer
  60        4000       24759          0   0.00%  0.00%  0.00%   0 Netclock Backgro
  65        4000        6191          0   0.00%  0.00%  0.00%   0 Ether-Switch RBC
  68           0         208          0   0.00%  0.00%  0.00%   0 Call Management
  73        4000        6192          0   0.00%  0.00%  0.00%   0 PI MATM Aging Pr
  74           0         621          0   0.00%  0.00%  0.00%   0 EtherChnl
  76           0        6048          0   0.00%  0.00%  0.00%   0 Ethernet Timer C
  81        4000         728          5   0.00%  0.00%  0.00%   0 CDP Protocol
  91       52000      193685          0   0.15%  0.14%  0.15%   0 IPAM Manager
  95    20144000       27190        740   0.00%  0.59%  0.44%   0 IP Input
  98           0          12          0   0.00%  0.00%  0.00%   0 MOP Protocols
 106        4000       24269          0   0.00%  0.00%  0.00%   0 SSS Feature Time
 110        8000         106         75   0.00%  0.00%  0.00%   0 CEF background p
```

```
120           0         104           0    0.00%  0.00%  0.00%   0 DFS flush period
121      828000         105        7885    0.00%  0.01%  0.00%   0 Licensing Auto U
123        4000        8412           0    0.07%  0.01%  0.00%   0 CEF: IPv4 proces
129        8000         110          72    0.00%  0.00%  0.00%   0 IP Background
136        4000        6202           0    0.00%  0.00%  0.00%   0 RUDPV1 Main Proc
138           0        6217           0    0.00%  0.00%  0.00%   0 bsm_xmt_proc
159           0         250           0    0.00%  0.00%  0.00%   0 CRM_CALL_UPDATE_
173           0           4           0    0.00%  0.00%  0.00%   0 AAA SEND STOP EV
174           0         622           0    0.00%  0.00%  0.00%   0 RMON Recycle Pro
178           0          52           0    0.00%  0.00%  0.00%   0 DHCPD Timer
180        4000         900           4    0.00%  0.00%  0.00%   0 Syslog
182           0        1244           0    0.07%  0.00%  0.00%   0 Compute load avg
183     1236000         104       11884    0.00%  0.01%  0.00%   0 Per-minute Jobs
187       36000         144         250    0.00%  0.00%  0.00%   0 DHCPD Receive
188       12000        6222           1    0.00%  0.00%  0.00%   0 DHCP Client
190        8000       12135           0    0.00%  0.00%  0.00%   0 IP NAT Ager
193        4000         104          38    0.00%  0.00%  0.00%   0 DHCPD Database
```

A tabela 9 exibe o nome dos campos e suas funções.

Tabela 9: Descrição dos dados gerados pelo comando show process cpu.

| Campo | Descrição |
|---|---|
| CPU utilization for five seconds | Utilização do CPU nos últimos 5 segundos |
| One minute | Utilização do CPU no último minuto |
| Five minutes | Utilização do CPU nos últimos 5 minutos |
| PID | Código de Identificação do Processo |
| Runtime (uS) | Tempo em microssegundos que o processo consumiu |
| Invoked | Número de vezes que o processo foi chamado |
| uSecs | Microssegundos do cpu para cada chamado de um processo |
| 5Sec | Consumo do processo nos últimos 5 segundos |
| 1Min | Consumo do processo durante o último minuto |
| 5Min | Consumo do processo durante os últimos 5 minutos |
| TTY | Terminal que controla o processo |
| Process | Nome do Processo |

Após a análise, podemos notar que alguns processos, tiveram aumento no tempo de execução do processo, no número de vezes em que ele foi chamado e na porcentagem de consumo do cpu.

Exemplos:

- **Processos de entrada de IP**

**Antes da ACL:**

| PID | Runtime(uS) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY Process |
|---|---|---|---|---|---|---|---|
| 95 | 56000 | 119 | 470 | 0.00% | 0.01% | 0.01% | 0 IP Input |

**ACL em execução:**

| PID | Runtime(uS) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY Process |
|---|---|---|---|---|---|---|---|
| 95 | 20144000 | 27190 | 740 | 0.00% | 0.59% | 0.44% | 0 IP Input |

- **Processos de tradução NAT**

**Antes da ACL:**

| PID | Runtime(uS) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY Process |
|---|---|---|---|---|---|---|---|
| 190 | 4000 | 292 | 13 | 0.00% | 0.00% | 0.00% | 0 IP NAT Ager |

**ACL em execução:**

| PID | Runtime(uS) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY Process |
|---|---|---|---|---|---|---|---|
| 190 | 8000 | 12135 | 0 | 0.00% | 0.00% | 0.00% | 0 IP NAT Ager |

### 7.11.1.2 Consumo de Memória

Para visualizar informações sobre os processos ativos no roteador e a quantidade de memória que eles ocupam, o seguinte comando deve ser executado:

```
show processes memory
```

- **Processos antes da execução das ACL's**

```
Processor Pool Total:  457243040 Used:   36123412 Free:  421119628
      I/O Pool Total:   10485760 Used:    3647680 Free:    6838080

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process

   0   0      12140      68644      12140          0          0 *Sched*
   0   0      57648   12253164       3060     171764     171764 *Dead*
   1   0     544708          0     551880          0          0 Chunk Manager
   2   0        236        236       4172          0          0 Load Meter
   3   0      12624        580      37660          0          0 LICENSE AGENT
   9   0     102896       4392     111012          0          0 Exec
  43   0      13560        236       7460          0          0 TTY Background
 121   0    1205476    1205476       7172          0          0 Licensing Auto U
 159   0      13820      13820       7172          0          0 CRM_CALL_UPDATE_
 173   0        236        236       7172          0          0 AAA SEND STOP EV
 180   0       6272       6272      13172          0          0 Syslog
 183   0          0        528       7172          0          0 Per-minute Jobs
 190   0      11908      11908       7172          0          0 IP NAT Ager
```

- **Processos durante a execução das ACL's**

```
Processor Pool Total:  457243040 Used:   36508540 Free:  420734500
      I/O Pool Total:   10485760 Used:    3647680 Free:    6838080

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process

   0   0      12140      94748      12140          0          0 *Sched*
   0   0    1024248   12922336     119360     171764     171764 *Dead*
   1   0     821332          0     828504          0          0 Chunk Manager
   2   0        236        236       4172          0          0 Load Meter
   3   0      12624        580      37660          0          0 LICENSE AGENT
   9   0     112752      16448     107604          0          0 Exec
  43   0      40336        496       7492          0          0 TTY Background
 121   0   25295484   25295484       7172          0          0 Licensing Auto U
 159   0     303236     303236       7172          0          0 CRM_CALL_UPDATE_
 173   0        236     186516       7172          0          0 AAA SEND STOP EV
 180   0     296128     296128      13172          0          0 Syslog
 183   0          0       2296       7172          0          0 Per-minute Jobs
 190   0      31980      31980       7172          0          0 IP NAT Ager
```

A tabela 10 mostra o nome de cada campo e a sua respectiva função.

Tabela 10: Descrição dos dados gerados pelo comando show processes memory.

| Campo | Descrição |
|---|---|
| Total | Quantidade total de memória |
| Used | Quantidade total de memória utilizada |
| Free | Quantidade de memória livre |
| PID | Código de Identificação do Processo |
| TTY | Terminal que controla o processo |
| Allocated | Bits de memória alocados pelo processo |
| Freed | Bits de memória liberados pelo processo |
| Holding | Quantidade de memória que o processo está utilizando |
| Getbufs | Número de vezes que o processo solicitou um buffer de pacote |
| Retbufs | Número de vezes que o processo abandonou um buffer de pacote |
| Process | Nome do Processo |

Após a análise, podemos notar que alguns processos, tiveram aumento na quantidade de bits de memória alocados e na quantidade de memória utilizada.

Exemplos:

- **Processos de tradução NAT**

**Antes da ACL:**

| PID | TTY | Allocated | Freed | Holding | Getbufs | Retbufs | Process |
|---|---|---|---|---|---|---|---|
| 190 | 0 | 11908 | 11908 | 7172 | 0 | 0 | IP NAT Ager |

**ACL em execução:**

| PID | TTY | Allocated | Freed | Holding | Getbufs | Retbufs | Process |
|---|---|---|---|---|---|---|---|
| 190 | 0 | 31980 | 31980 | 7172 | 0 | 0 | IP NAT Ager |

- **Processos de Log**

**Antes da ACL:**

| PID | TTY | Allocated | Freed | Holding | Getbufs | Retbufs | Process |
|---|---|---|---|---|---|---|---|
| 180 | 0 | 6272 | 6272 | 13172 | 0 | 0 | Syslog |

**ACL em execução:**

| PID | TTY | Allocated | Freed | Holding | Getbufs | Retbufs | Process |
|---|---|---|---|---|---|---|---|
| 180 | 0 | 296128 | 296128 | 13172 | 0 | 0 | Syslog |

### 7.11.2 *Consumo do Iptables no Linux*

Para visualizar um processo e o consumo de cpu e memória que ele gera no linux, basta executar o seguinte comando:

```
ps -aux | grep iptables
```

Resultado:

```
USER     PID     %CPU     %MEM    VSZ     RSS     TTY     STAT     START
root    3610      0.0      0.0    3312    800      ?        R      12:59
```

A tabela 11 exibe o nome dos campos e a sua função.

Tabela 11: Descrição dos dados gerados pelo comando ps -aux | grep iptables

| **Campo** | **Descrição** |
|---|---|
| USER | Usuário dono do processo |
| PID | Identificação do processo |
| %CPU | Porcentagem de uso do processador |
| %MEM | Porcentagem de uso da memória |
| VSZ | Indica o tamanho virtual do processo |
| RSS | Indicada a quantidade em KB de memória usada |
| TTY | Indica o identificador do terminal do processo |
| STAT | Mostra o estado atual do processo |
| START | Hora em que o processo foi iniciado |

Ao analisar o resultado do comando podemos ver que o iptables não apresenta nenhum consumo de processador e memória.

# 8 CONCLUSÃO

Neste trabalho, foram apresentados conceitos importantes da segurança da informação, os principais tipos de ataques e vulnerabilidades, assim como as principais técnicas e ferramentas de segurança. O funcionamento, tipos e arquitetura do *firewall* também foram itens explicados com clareza e que deram um suporte para o entendimento da importância do *firewall* na infraestrutura de segurança de qualquer organização.

Ainda que a escolha de qual tecnologia utilizar na empresa dependa da necessidade e disponibilidade de recursos, os *firewalls* corporativos apresentados neste trabalho proporcionam ao administrador de rede um número maior de opções quanto à escolha de qual tecnologia utilizar na infraestrutura da empresa, pois todos os detalhes como instalação, configuração, elaboração das regras, funcionamento e desempenho foram expostos de forma concisa.

Para a implementação do *firewall* com ACL, o único *hardware* utilizado foi um roteador cisco. A vantagem da ACL é que sua configuração e aplicação é simples em comparação ao Iptables, basta o administrador de redes ter conhecimentos sobre comandos do IOS Cisco e conceitos de redes. Todavia, as acl's trazem desvantagens, o bloqueio de certos serviços como sites, não são tão precisos e seguros como o Iptables, que tem a opção de bloqueio pelo conteúdo do pacote, dispensando assim os bloqueios pelo endereço IP e URL do site que são feitos com regras ACL. Outra desvantagem da ACL é o alto consumo de memória e processamento, não apenas pela execução das regras, mas também pelo registro de *logs*, esse alto consumo pode até interferir em outras funções do roteador como o roteamento de pacotes.

A configuração do Iptables, exige um considerável conhecimento de comandos do linux, parâmetros e estrutura do Iptables (*regras, tabelas e chains*), além de conceitos básicos sobre *shell script*, porém, para uma configuração segura não são necessárias muitas regras e estas poucas regras garantem o perfeito funcionamento da rede, e bloqueio de serviços e aplicações indesejados. Além disso, ele não gera nenhum consumo no roteador, já que é uma máquina separada e no próprio computador onde o serviço é executado, também não há aumento no consumo dos *hardwares.*

Por meio deste estudo e após avaliar as duas tecnologias apresentadas, foi possível concluir com base nas características e vantagens apresentadas, que o *iptables* é o *firewall* a nível de pacotes mais eficiente e estável em comparação a ACL's, para o cenário proposto nesse trabalho. Porém, não se deve delegar a ele toda a segurança da rede de uma organização outras ferramentas e técnicas de segurança devem ser implantadas a fim de dificultar a ação dos invasores.como: IDS, IPS, Política de Segurança, entre outras.

## REFERÊNCIAS BIBLIOGRÁFICAS


BRITO, Samuel Henrique B. **Laboratórios de Tecnologias Cisco em Infraestrutura de Redes**. 1. ed. São Paulo: Novatec, 2012. 160 p.

CAMPOS, André L. N. **Sistema de segurança da informação:** controlando os riscos. Florianópolis: Visual Books, 2006.

Cisco Systems. **The show processes Command.** Disponível em: <http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_tech_note09186a00800a65d0.shtml> Acesso em: 03 ago. 2013.

DAVIS, David.TechRepublic. **Block access to a Web site using the Cisco IOS.** Disponível em: **<**http://www.techrepublic.com/article/block-access-to-a-web-site-using-the-cisco-ios/**>** Acesso em: 10 ago. 2013

DIAS, Cláudia. **Segurança e auditoria da tecnologia da informação**. Rio de Janeiro: Axcel Books do Brasil, 2000.

ESQUIVEL, Claudyson Jonathas. **Gerenciamento de Regras de Firewalls em Ambiente Linux.** Uberlândia: UNIMINAS, 2006. 59 p.

JUNIOR, Vamberto de Freitas R. **Estudo e Implementação de Firewalls em Ambientes Corporativos.** Paraíba, 2010. 98 p.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet**. 5. ed. São Paulo: Pearson, 2011. 614 p.

MARINHO, Hermógenes. **Algumas dicas no Iptables.** Disponível em: <http://rede100fio.wordpress.com/2011/08/13/algumas-dicas-no-iptables-netfilter/> Acesso em: 17 jul. 2013.

MORIMOTO, Carlos E. **Como criar um Firewall e compartilhar conexão usando Iptables.** Rio de Janeiro, 2007

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício de. **Segurança de Redes em Ambientes Cooperativos**. 1 ed. São Paulo: Novatec, 2012. 483 p.

NETO, Urubatan. **Dominando Linux Firewall Iptables**. Rio de Janeiro: Ciência Moderna, 2004. 108 p.

OLIVEIRA, Frank Ned S. C. **Segurança em Roteadores**. Infotec Campinas, 2001. 24 p.

PEREIRA, Pedro. **acl's entenda de uma vez por todas.** São Paulo, 2012. 3 p.
PINHEIRO, José Maurício Santos. **Redes Privadas Virtuais.** Projeto de Redes: 2007. 10 p.

ROCHA JUNIOR, Vamberto de Freitas. **Estudo e Implementação de F*irewall*s em Ambientes Corporativos.** João Pessoa: Faculdade de Tecnologia de João Pessoa, 2010. 98 p.

SANTOS, Marcos Antônio de Souza. **Firewall: Requisitos e Primícias na escolha de sua Utilização.** Recife: Faculdade de Santa Maria, 2007. 77 p.

SCAGLIONE, Luiz. **POLÍTICA INTERNA DE T.I.** COMEXIM LTDA. Santos: 15 jun 2013. 5 p.

SÊMOLA, Marcos. **Gestão da Segurança da Informação**: Uma visão executiva. 1ª Ed. Rio de Janeiro: Campus, 2003.

SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Vitória, 2006.

**SPANCESKI**, Francini Reitz. **Política de Segurança da Informação – Desenvolvimento de um modelo voltado para Instituições de Ensino.** Joinville, 2004. 102 p.

TANENBAUM, A. S. **Redes de Computadores.** 4 ed. São Paulo: Editora Campus, 2003. 968 p.