

## O USO DE FERRAMENTAS OPEN SOURCE PARA APLICAÇÕES DE SEGURANÇA EM REDES CORPORATIVAS: UM ESTUDO BASEADO EM FIREWALLS

Tharlis da Silva Barbosa<sup>1</sup> e Flavio Alexandre dos Reis<sup>2</sup>

### RESUMO

Com a alta demanda na busca por informações, cresce cada vez mais a procura por um meio fácil, rápido e prático de se consegui-las. Eis que surge a necessidade de conexão permanente com a internet, onde cada computador conectado é susceptível a ataques. Há ainda o problema do monitoramento e gerenciamento dos usuários e de todo o conteúdo que trafega pela rede, sem mencionar o desempenho nos acessos a sites freqüentemente utilizados pela empresa. Diante desta situação, este artigo tem como objetivo discorrer sobre ferramentas de controle e proteção para redes de computadores. As ferramentas aqui abordadas são os *Firewalls*, a nível de pacotes e a nível de aplicação. Tratam-se do *Iptables* e do *Squid*, que são ferramentas *Open Source*.

**Palavras-chave:** *Firewall. Iptables. Proxy. Squid. Redes.*

### ABSTRACT

With the high demand in the search for information, grows increasingly looking for an easy, fast and convenient to get them. Here arises the need for permanent connection to the Internet, where each connected computer is susceptible to attacks. There is also the problem of monitoring and management of users and all content that travels over the network, not to mention the performance in access to sites frequently used by the company. In this situation, this article aims to discuss tools for control and protection for computer networks. The tools discussed here are the

---

1- Tecnólogo em Análise e Desenvolvimento de Sistemas, Centro de Ensino Superior de Valença – CESVA/FAA; Sargento Administrador da Seção de Informática do 1º Esquadrão de Cavalaria Leve – Esquadrão Tenente Amaro – Exército Brasileiro. infinity7@ig.com.br

2- Professor da Fundação Educacional Don André Arcoverde – Valença RJ.  
reis.falexandre@gmail.com

firewalls, packet-level and application level. These are the iptables and squid, which are Open Source tools.

**Keywords:** Firewall. Iptables. Proxy. Squid. Network.

## INTRODUÇÃO

Nos dias atuais não existe outra fonte de informação tão extensa, dinâmica e de fácil acesso quanto à Internet. Através dela, é possível se ter acesso a todo e qualquer tipo de conteúdo, que muitas das vezes se faz necessário no cotidiano da empresa. Porém, nem todo o conteúdo é benéfico aos interesses da empresa, podendo em muitas das vezes, prejudicar o bom funcionamento e danificar equipamentos utilizados pela empresa.

Existe ainda, a possibilidade de acessos indevidos à rede interna da empresa, oriundos de equipamentos situados fora das dependências da mesma, podendo até ser considerados como ataques de *Crackers*<sup>3</sup> ou algo do tipo.

Mediante a este cenário e à crescente busca pela otimização e segurança das redes, surgem a partir daí, diversos desafios, a fim de solucionar problemas de segurança e ao mesmo tempo, otimizar os devidos acessos aos conteúdos que realmente se fazem necessários na empresa.

O objetivo deste artigo é demonstrar alguns dos controles que podem ser aplicados, utilizando os recursos das ferramentas propostas afim de realizar a segurança em uma rede.

O *Iptables*<sup>4</sup> e o *Squid*<sup>5</sup> são ferramentas muito úteis para segurança e monitoramento de redes. Além do fato de serem *Free / Open Source*, baseadas na Licença Pública Geral – GPL (do inglês *General Public Licence*), e utilizadas em distribuições GNU/Linux, o que proporciona alta performance e um custo quase zero (se em comparação com ferramentas e sistemas proprietários).

### Justificativa

---

3- Termo usado para designar quem pratica a quebra de um sistema de segurança, de forma ilegal ou sem ética.

4- Nome da ferramenta que permite a criação de regras de firewall.

5- Software especializado em fazer a operação de proxy.  
Saber Digital, v. 5, n. 1, p. 72-90, 2012

A relevância do tema recai sobre a importância da utilização de mecanismos de segurança em redes de computadores. Muitas redes são projetadas e mantidas sem o uso de técnicas de segurança adequadas para eximir o risco de uma possível invasão ou acesso não autorizado, e ainda a perda de performance dos acessos da rede. É necessário o uso de mecanismos que visam garantir a segurança e o controle do que se trafega na rede, com o objetivo de minimizar ao máximo a possibilidade de ataques e ao mesmo tempo otimizar o uso da rede.

## **Ameaças**

Com os avanços tecnológicos atuais e a capacidade de qualquer computador, em qualquer lugar, se conectar a qualquer outro computador, também em qualquer lugar, as facilidades de comunicação se tornam imensas. Porém para alguns indivíduos em particular (gerentes de segurança, administradores de rede, entre outros), isto acaba tornando-se um pesadelo. É fato que, atualmente, muitas empresas têm quantidades significativas de informações confidenciais on-line (segredos comerciais, planos de desenvolvimento de produtos, estratégias de marketing, análises financeiras, entre outros).

É conveniente pensar a respeito do que poderia acontecer, caso essas informações fossem disponibilizadas a um concorrente. Certamente as consequências não seriam das melhores. Além disso, há ainda o perigo das informações virem a público ou mesmo vazarem dentro da empresa. Há também de se considerar os *malwares* (vírus de computador<sup>6</sup>, cavalos de tróia<sup>7</sup>, *worms*<sup>8</sup>, entre outros), que podem facilmente burlar a segurança e destruir dados valiosos, ocasionando confusão e consumindo tempo e intelecto dos administradores.

Diante deste cenário, faz-se necessária a adoção de mecanismos de segurança para poder barrar o tráfego dos pacotes indevidos e liberar o dos bits realmente necessários. O *firewall* é o mecanismo o responsável por estes controles.

---

6- Programa malicioso que infecta o sistema em que se encontra, fazendo cópias de si mesmo e tentando se espalhar para outros computadores, utilizando-se de diversos meios de entrada e saída, da mesma forma que um vírus biológico.

7- Trojan Horse - Programa malicioso que tem por finalidade, agir como na história do cavalo de Tróia. Uma vez dentro do computador, ele libera uma porta para uma possível invasão.

8- Programas auto-replicantes. Se assemelham aos vírus de computador, mas não necessitam de um programa hospedeiro para se propagar.

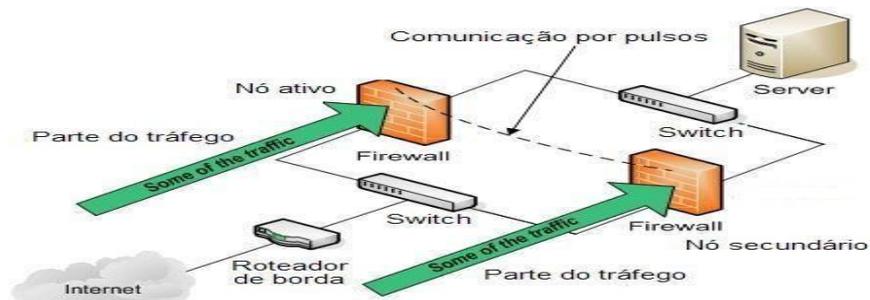
## Firewall

O *Firewall* (ou parede de fogo) atua como uma barreira entre redes distintas. Ele é quem possui o controle dos acessos às redes, permitindo maior segurança e confiabilidade do conteúdo que se tem na rede.

*Firewall*, nada mais é que um dispositivo de segurança, que visa realizar a filtragem daquilo que é ou não permitido ser acessado em uma rede, seja o acesso proveniente da rede interna ou da rede externa. É importante salientar que o *firewall* não bloqueia os *e-mails* acessados através do navegador de internet, apenas aqueles em que o seu acesso é feito por meio de programas específicos (Lotus organizer, Outlook, Thunderbird, Kmail, entre outros), denominados Clientes de *e-mail*. Tal feito se dá pelo fato da possibilidade de bloqueio de portas específicas de envio e recebimento de *e-mails*, já que tais programas trabalham baseados nestas portas. No caso dos navegadores de internet, não poderia ser utilizada esta técnica de bloqueio por portas, pois senão, todo o restante dos acessos feitos através do navegador também seria bloqueado [1].

Para contornar este problema, deve-se adotar políticas de segurança com relação às restrições relativas ao uso do correio eletrônico, ou ainda, fazer o bloqueio de sites específicos de *e-mail*, seja a nível de pacotes (no *firewall*) ou a nível de aplicação (no *proxy*) [2], [3].

O *firewall* estabelece limites e realiza o controle de todo o tráfego de dados, entre diversos computadores em uma rede, e ainda, o controle do tráfego entre redes distintas. O próprio Sistema Operacional concede autonomia para que este dispositivo discipline todo o tráfego existente [4], [5]. A figura 1, apresenta um dos possíveis esquemas de *firewall*.



**Figura 1** – Esquema de firewall duplo (tolerante a falhas)

Alguns *firewalls* são baseados tanto em *hardware*<sup>9</sup> quanto em *software*<sup>10</sup>, e outros são apenas baseados em *software* [3]. Este artigo abordará o *firewall* como *software*, basicamente do tipo *Stateful Firewall* e *Proxy*, onde um computador é configurado para funcionar como tal.

A eficiência de um *firewall* consiste, na passagem de todo o tráfego (seja de origem ou destino à internet ou à outra rede) através dele, para que assim, ele possa ter total controle do que se trafega na rede. Sua principal função é a restrição de fluxos indesejados, seja da rede interna para a internet ou o contrário [3].

## Tipos de Firewall

É importante que se esteja familiarizado com os tipos de *firewalls*, bem como suas arquiteturas, para que desta forma, ele possa trabalhar de maneira correta e eficiente. Esta ferramenta, com o passar dos anos, sofreu diversos aperfeiçoamentos, deixando de ser meramente um “filtro de pacotes” e adquirindo novas classes e conseqüentemente originando diversos tipos de *firewalls*, dos quais podemos destacar dois em especial: *Stateful Firewall* e *Proxy* [3], [5].

## Stateful Firewall

Trata-se da evolução do *firewall* do tipo *Packet Filter*. Pode identificar o protocolo dos pacotes transeuntes e verificar as respostas legítimas.

9- É a parte física do computador, ou de qualquer outro equipamento eletrônico, composta por circuitos, cabos e placas.

10- É uma aplicação, um programa de computador ou de outro equipamento eletrônico, que permite executar uma determinada tarefa.

É uma tecnologia implementada em filtros de pacotes, na qual o *firewall* guarda o estado das transações efetuadas (comunicação TCP e UDP) e inspeciona o tráfego, evitando pacotes falsos. Somente os pacotes conhecidos podem trafegar pelo filtro, sendo que em alguns casos, este ainda é capaz analisar o conteúdo de um pacote buscando perfis de ataque.

O *Stateful Firewall* controla seletivamente o fluxo de dados e o tratamento do cabeçalho TCP, além de lidar com protocolos mais específicos (como o FTP, por exemplo), manter informações de estado de conexão e ainda manipular os campos de um *datagrama* [8]. Como exemplo de um *Stateful Firewall*, podemos citar o *Iptables*.

### ***Iptables***

O *Iptables* é, na verdade, uma ferramenta de *front-end*<sup>11</sup> que permite a manipulação das tabelas do *NetFilter*<sup>12</sup>, apesar de ser constantemente confundido com um firewall, devido à maneira como ele trata as regras junto ao *Kernel*<sup>13</sup> do Sistema.

De acordo com o site [www.iptables.cjb.net](http://www.iptables.cjb.net), o *Iptables* é comumente conhecido como *firewall* do tipo filtro de pacotes, devido ao fato de examinar cada pacote de cada conexão da rede. Mas não é só, o *Iptables* pode fazer mais que um simples filtro de pacotes, pode também memorizar o estado das conexões, de forma a saber o que o pacote representa para a conexão, ou seja, o seu contexto na conexão. Por isso é considerado um *Stateful Firewall*.

O *Iptables* realiza a implementação de filtros de pacotes através da Tabela *Filter*, a tradução de endereços de rede com a tabela *NAT* e diversos outros controles ainda mais avançados, como por exemplo, o desenvolvimento e a aplicação de qualidade de serviço (QOS – *Quality Of Service*), redirecionamento de portas, mascaramento de conexões, detecção de fragmentos, monitoramento de

---

11- Interface (tela de inserção ou consulta) entre o usuário e o interior do programa ou sistema.

12- É um módulo do Kernel GNU/Linux que é manipulado pelo *Iptables*. Uma ferramenta que atua na filtragem dos pacotes, tradução de endereços e geração de logs do sistema, entre outras funções. É como um banco de dados, que em sua estrutura estão contidas três tabelas padrões: Tabela *Filter*, Tabela *Nat* e Tabela *Mangle* [5].

13- Núcleo do sistema operacional.

tráfego, TOS (*Type Of Service*), bloqueio a ataques diversos, entre outros. Em termos de hardware, ele não é muito exigente, pois com apenas 4 MB (megabytes) de memória em uma arquitetura 386, com a versão 2.4 (ou superior) do *Kernel* do sistema, é possível implementá-lo sem quaisquer problemas [5].

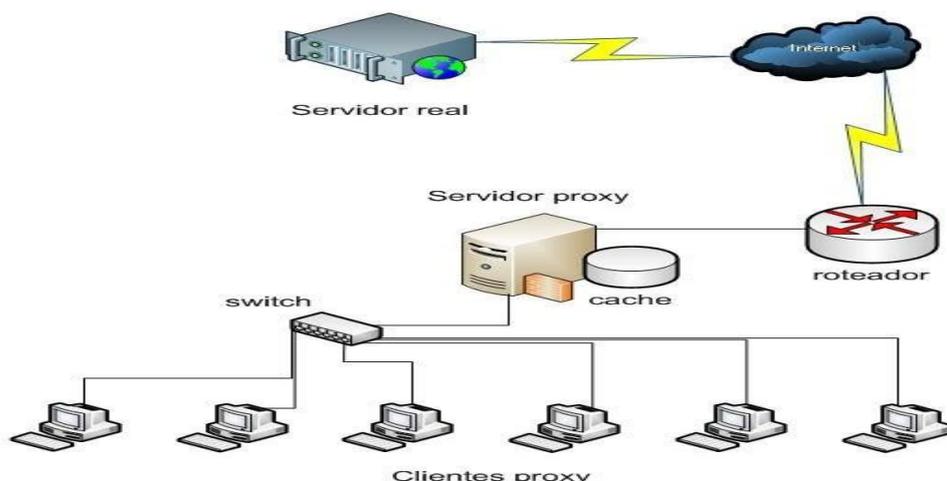
## **Proxy**

*Proxy* é um *firewall* a nível de aplicação. Trata-se de um servidor onde são armazenados em seu *cache*<sup>14</sup>, objetos (páginas web, imagens, arquivos de áudio, programas, entre outros) da Internet para posterior distribuição. Funciona como um servidor intermediário entre os clientes e os servidores reais, aos quais são feitas as requisições. O cliente faz a requisição ao *proxy*, então este é quem na verdade, irá repassar a requisição do cliente ao servidor de destino. Após o atendimento da solicitação por parte do servidor real, o *proxy* repassa ao cliente a requisição que anteriormente ele havia feito. É então, enviado ao cliente o que ele havia requerido [4], [6].

Ao receber uma requisição de um cliente, o *proxy* irá verificar se constam armazenados em seu *cache*, os objetos constantes da requisição do cliente. Caso estes objetos já estejam armazenados em seu *cache*, ele simplesmente irá devolver ao cliente o que este havia requerido, sem a necessidade de realizar alguma solicitação a algum servidor real, o que acelera o tempo de resposta. Observe a figura 2, representando um esquema de *proxy*.

---

14- Dispositivo de acesso rápido, interno a um sistema, intermediário entre um operador de um processo e o dispositivo de armazenamento ao qual esse operador acede.  
Saber Digital, v. 5, n. 1, p. 72-90, 2012



**Figura 2** – Esquema de Proxy

Além de ser utilizado como cache, o *proxy* ainda cria uma barreira de segurança entre a rede interna e a *Internet*, permitindo o compartilhamento da conexão por vários usuários. Um dos proxies mais utilizados atualmente é o *Squid*.

### **Squid**

O *Squid* é uma ferramenta utilizada como *firewall* a nível de aplicação, por se tratar de um servidor *proxy*. Intermediando a rede privada e a rede externa (*internet*), ele realiza a análise do conteúdo dos pacotes procurando por restrições previamente especificadas nas regras de seu arquivo de configuração. Trata-se de uma ferramenta *Free / Open Source* sendo licenciada nos termos da (GPL), como resultado da colaboração de diversas pessoas a nível mundial, contribuindo com o seu desenvolvimento.

De acordo com o site [www.squid-cache.org](http://www.squid-cache.org), muitas pessoas se utilizam dos serviços do *Squid*, mas nem se dão conta deste fato. Diversas empresas o estão adotando em suas redes, havendo também a sua utilização em grande escala em servidores *Web*, para acelerar o acesso ao conteúdo disponibilizado.

Também oferece a possibilidade de restrição de acessos à páginas proibidas, para usuários ou grupos e podendo também limitar horários para os acessos. É possível também, liberar o acesso somente de determinadas categorias de domínios (. gov, .edu, .mil, e outros), e ainda, proibir *downloads* de certos arquivos, seja por

nome ou por extensões (.exe, .com, .bat, .mp3, e outros). As regras aplicadas ao *Squid* são bem flexíveis, podendo-se estabelecer vários tipos de bloqueios [4], [6].

No *Squid*, é possível a criação de listas de controle de acessos (*Access Control List* – ACL), que desempenham papel muito significativo no que tange a configuração dos acessos através do *proxy*, pois agregam grande flexibilidade e eficiência no trabalho desta ferramenta. Através delas é possível criar regras de controle de acesso à internet, das mais diversas formas possíveis. É correto dizer que quase todo o processo de controle do *Squid*, é feito com o uso das ACLs [2], [6].

As ACLs definem quais usuários ou equipamentos tem permissão a determinados serviços no *Squid*. Basicamente são políticas de acessos, definindo os tipos de acessos para cada usuário, equipamento ou grupo. As ACLs devem ser tratadas como uma camada de complemento à segurança da rede [2].

## **Outros Tipos de *Firewall***

É importante a familiarização com os diversos tipos de *firewalls* e com suas arquiteturas. Aqui serão abordados alguns outros tipos de *firewalls*, que são: *Packet Filter*, *Circuit-Level Gateways* e *Bastion Host*.

### ***Packet Filter***

O filtro de pacotes, como o nome sugere, é uma classe de *firewall* (a nível de pacotes) que se responsabiliza pela filtragem de todo o tráfego destinado a ele ou à rede protegida por ele, assim como todos os pacotes emitidos pela rede protegida. Ao implementar este tipo de *firewall*, é necessário o conhecimento sobre campos dos protocolos dos quais será feito o bloqueio ou permitido o acesso, como o TCP, UDP, IP e o ICMP. Ele analisa o cabeçalho dos pacotes durante sua passagem, decidindo o destino dos mesmos como um todo, descartando (DROP - descartando-o como se não o tivesse recebido), aceitando (ACCEPT - deixar o pacote seguir seu caminho) ou rejeitando (REJECT - descarta o pacote informando o motivo do descarte) os pacotes [4] [3].

## ***Circuit-Level Gateways***

Este tipo de *Firewall* atua nas camadas de Transporte, tanto do Modelo OSI quanto do Modelo TCP/IP. É responsável por criar um circuito cliente/servidor, sem interpretar o protocolo de aplicação. Monitora o TCP *handshaking* entre pacotes, com a finalidade de determinar se é legítima ou não, a sessão requisitada. As informações passadas através de um *Circuit-Level Gateway* parecem ter sido originadas do roteador, o que facilita a ocultação de informações sobre as redes protegidas. Operam através de protocolos mais simples (como o *socks*), com informações sobre a conexão a estabelecer [4].

## **Bastion Hosts**

*Bastion Host* é qualquer máquina configurada de modo a desempenhar um papel crítico, com relação à segurança da rede interna. Nada mais é do que uma máquina que fica entre a rede interna e a internet, ou outra rede qualquer, sendo o único meio de tráfego de dados entre elas. Deve ter uma estrutura simples, de maneira a facilitar a garantia da segurança.

As responsabilidades de um *Bastion Host* se diferem das do *Packet Filter*, pois ele trabalha a nível de aplicação. Tanto as funções de filtragem de pacotes, como também o provimento de outros serviços, podem ser cumulativas a um *host*, sendo que neste caso, este seria um *Packet Filter* em conjunto a um *Bastion Host* (*Dual-Homed Host*<sup>15</sup>) [4].

## **Exemplos de uso**

Em um ambiente de trabalho no qual se utilizam de microcomputadores para o desenvolvimento das atividades relativas ao cotidiano da empresa, provavelmente a rede interna deve estar interligada à Internet para dar prosseguimento aos diversos trabalhos realizados via web.

---

<sup>15</sup>- Arquitetura montada sobre um computador que possui duas interfaces de rede, sendo que uma está conectada à internet e a outra à rede interna, semelhante a um roteador, porém sem realizar o roteamento [4].

Visando atender as necessidades de comunicação entre os processos que utilizam recursos de TI com ênfase no acesso à Internet, centralizado em um servidor, onde devem ser utilizados mecanismos para o controle de acesso à rede mundial de computadores e filtragem de pacotes, sendo para isto, adotada a utilização das ferramentas *Iptables* e *Squid*, determinados aspectos foram analisados com o intuito de desenvolver regras de acesso e filtragem. São elas:

- Regra nº 1: Estabelecer quais usuários da instituição podem ter acesso à Internet;
- Regra nº 2: Estabelecer quais domínios / sites devem ser bloqueados;
- Regra nº 3: Estabelecer quais usuários ou máquinas devem ter acesso livre ou restrito;
- Regra nº 4: Estabelecer controle de banda, de forma que o consumo total da largura da banda não exceda 128 Kbps por usuário;
- Regra nº 5: Estabelecer quais extensões de arquivos devem ser bloqueadas;
- Regra nº 6: Realizar o compartilhamento da conexão de internet com as demais máquinas da rede;
- Regra nº 7: Impedir respostas às solicitações ICMP (bloquear ping), exceto da rede interna para o servidor;
- Regra nº 8: Permitir acesso SSH somente a determinados endereços, realizando a proteção contra tentativas originadas de outros endereços.
- Regra nº 9: Estabelecer quais máquinas não devem ter acesso à Internet.
- Regra nº 10: Estabelecer quais os processos nos quais são utilizados recursos de TI (ver tabela 60), tendo como canal de comunicação a Internet, devem ter livre acesso;

**Tabela 1** – Exemplo de endereços e portas utilizados

Sistema	Endereço	Portas de utilização
SIAFI	161.148.40.200/24	1:23000
stemas DAPROM	200.140.140.208/24	13001
Sistemas DGP	200.140.140.213	1024:65535
	200.140.140.212	
	200.140.140.199	
Servidor Asterisk / VoIP	200.140.140.216	5060:5063
	10.12.184.7	
Sistemas FAP	10.67.4.20	1:65000
Túnel VPN		5002
COMPRASNET	161.148.173.66/24	80

O sistema utilizado foi o GNU/Linux distribuição Debian 5.0, juntamente com a conjugação das ferramentas *Iptables* e *Squid*, que são *firewalls* do tipo *Stateful Firewall* e *Proxy*, respectivamente. Ambas as ferramentas foram aplicadas em um servidor do tipo *Gateway*<sup>16</sup>, compondo um *firewall* com arquitetura do tipo *Dual-Homed Host* (ver figura 1).

### Utilizando o Squid

Com os parâmetros de configuração do *Squid* definidos, a nova política de acesso à Internet na instituição foi definida e acompanhada, observando o fator segurança. Baseando-se em uma configuração do *Squid* com autenticação de usuários, a figura 3 demonstra a tentativa de acesso sem o uso do login e senha de usuário do *proxy*, baseando-se na primeira regra estabelecida.

16- Máquina intermediária, geralmente destinada a interligar redes.  
Saber Digital, v. 5, n. 1, p. 72-90, 2012

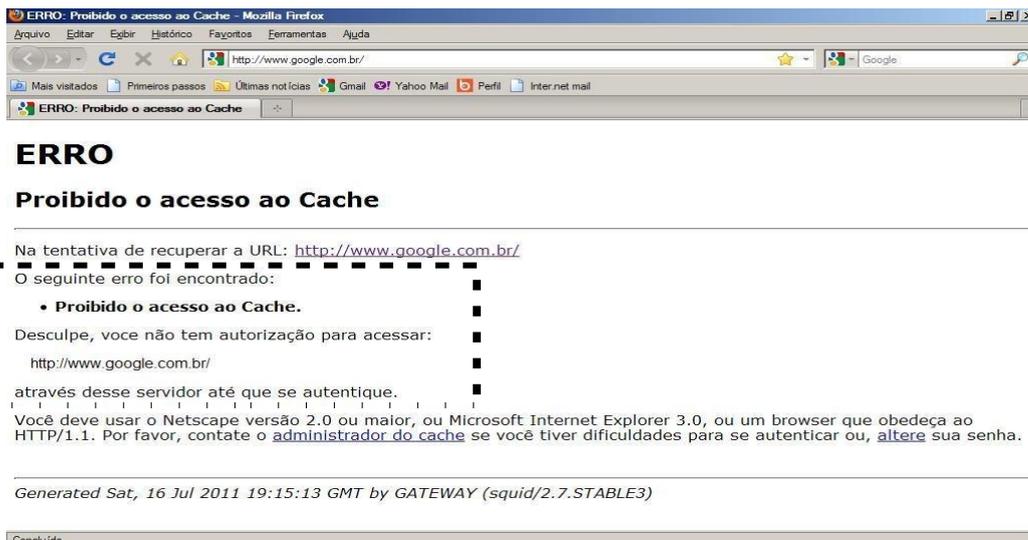


Figura 3 – Tela de tentativa de acesso – Acesso proibido

As figuras 4 e 5 demonstram respectivamente os acessos realizados Terceira regras.

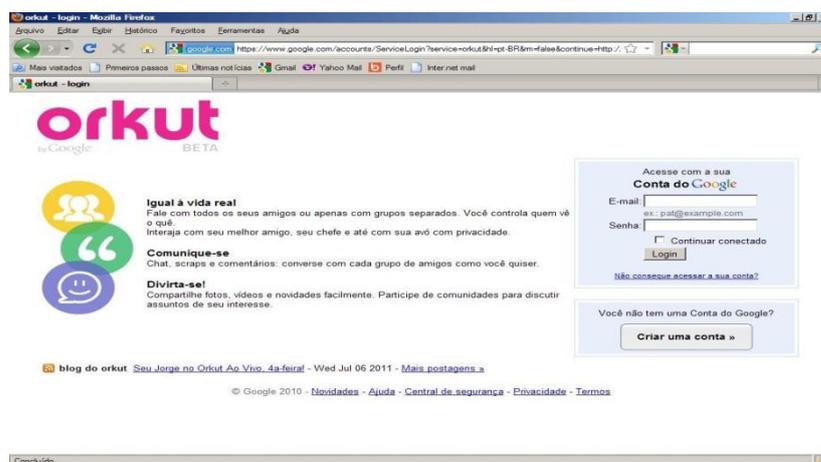


Figura 4 – Tela de conteúdo acessado – Acesso à página [www.orkut.com/Main](http://www.orkut.com/Main)

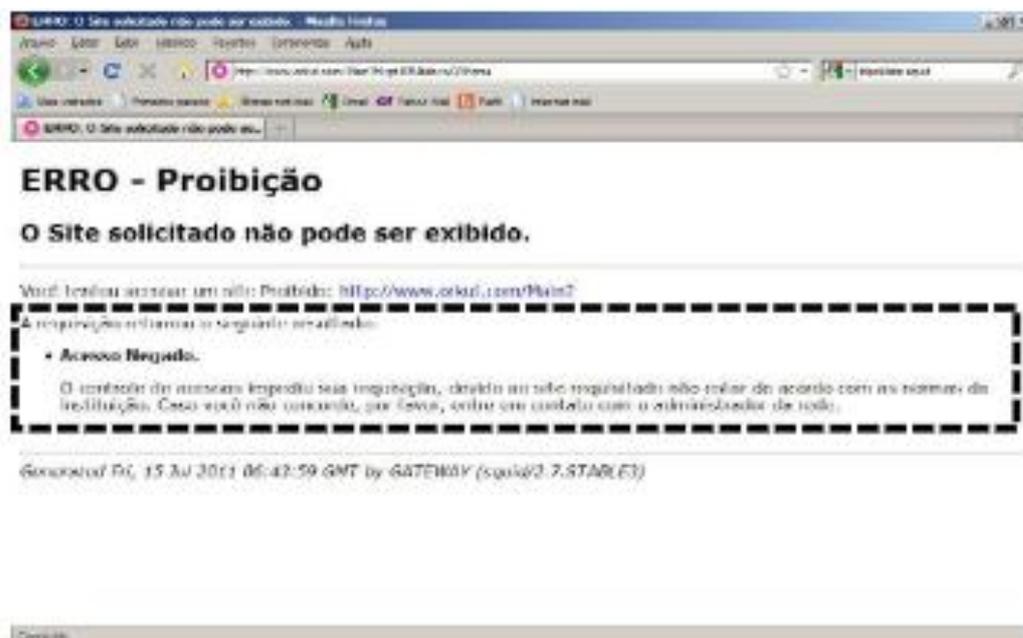


Figura 5 – Tela de conteúdo bloqueado – Acesso à página [www.orkut.com/Main](http://www.orkut.com/Main)

As figuras 6 e 7, demonstram respectivamente as medições de velocidade de acessos, com a finalidade de demonstrar o controle de banda de Internet realizado. Os acessos foram realizados antes e depois da adoção das políticas, baseando-se na quarta regra estabelecida.



Figura 6 – Tela de medição de velocidade de conexão – Acesso à página [www.rjnet.com.br](http://www.rjnet.com.br)



Figura 7 – Tela de medição de velocidade de conexão – Acesso à página www.rjnet.com.br

As figuras 8 e 9, demonstram respectivamente as tentativas de downloads de um arquivo executável (.exe). Os acessos foram realizados antes e depois da adoção das políticas, baseando-se na quinta regra estabelecida.



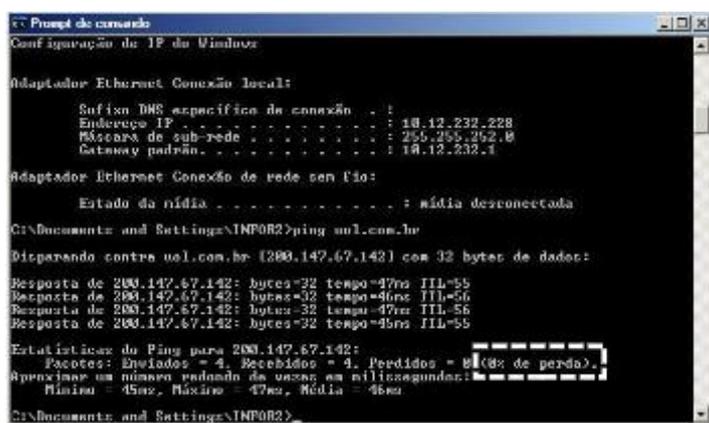
Figura 8 – Tela de download de arquivo – Tentativa de download do arquivo “putty.exe”

## Utilizando o *Iptables*

Tendo sido definidas as políticas de acesso, verificou-se o aumento na segurança da rede, tendo em vista o fato de que antes da adoção da ferramenta *iptables*, os endereços das máquinas pertencentes à rede poderiam ser vistos pela Internet, o que possibilitaria um suposto ataque. Além do fato de que qualquer máquina configurada para acessar a rede, poderia também acessar a Internet sem problemas. Com a adoção de um servidor rodando o *iptables* na instituição, foram solucionados estes problemas, mas ainda restava a segurança contra o acesso indevido ao próprio servidor, problema este que foi solucionado com a permissão de acesso ao servidor, somente a um determinado endereço IP.

A adoção de *firewalls* na instituição em questão foi de grande contribuição para a segurança das informações da mesma. A seguir será demonstrada a aplicação do *iptables* em alguns controles, baseado nas regras estabelecidas anteriormente.

As figuras 9 e 10 demonstram respectivamente as tentativas de obtenção de resposta do comando “ping” (*icmp-echo-request*), com a finalidade de demonstrar o bloqueio realizado. As tentativas foram realizadas antes e depois da adoção das políticas, baseando-se na sétima regra estabelecida.



```

C:\Documents and Settings\NINFOR2>ipconfig

Adaptador Ethernet Conexão local:
    Sufixo DNS específico da conexão . . . :
    Endereço IP . . . . . : 10.12.232.228
    Máscara de sub-rede . . . . . : 255.255.252.0
    Gateway padrão . . . . . : 10.12.232.1

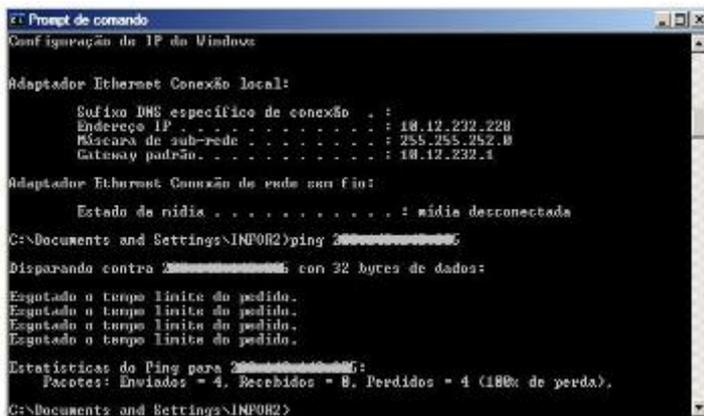
Adaptador Ethernet Conexão de rede sem fio:
    Estado da mídia . . . . . : mídia desconectada

C:\Documents and Settings\NINFOR2>ping uol.com.br

Disparando contra uol.com.br (200.147.67.142) com 32 bytes de dados:
Resposta de 200.147.67.142: bytes=32 tempo=47ms TTL=55
Resposta de 200.147.67.142: bytes=32 tempo=46ms TTL=56
Resposta de 200.147.67.142: bytes=32 tempo=47ms TTL=56
Resposta de 200.147.67.142: bytes=32 tempo=45ms TTL=55

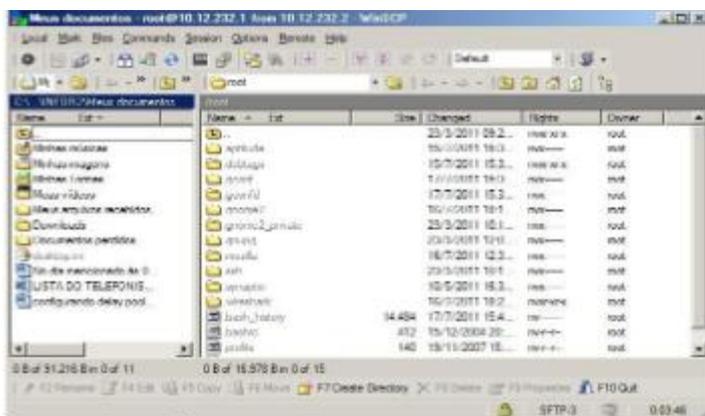
Estatísticas de Ping para 200.147.67.142:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
    Aproximar um número rodando de vezes em milissegundos:
    Mínimo = 45ms, Máximo = 47ms, Média = 46ms
C:\Documents and Settings\NINFOR2>
```

Figura 9 – Tela do prompt de comando – Resposta à solicitação do comando “ping”



**Figura 10** – Tela do prompt de comando – Resposta negada à solicitação do comando “ping”

As figuras 11 e 12 demonstram tentativas de obtenção de acesso SSH ao servidor (ip 10.12.232.1), com e sem sucesso respectivamente. A finalidade é demonstrar o bloqueio realizado, sendo que, as tentativas foram realizadas antes e depois da adoção das políticas, baseando-se na sétima regra estabelecida.



**Figura 11** – Tela do WinSCP – Acesso SSH realizado com sucesso



**Figura 12** – Tela do WinSCP – Tentativa de acesso SSH sem sucesso

A figura 13, demonstra a tentativa de acesso à Internet oriunda de um computador bloqueado pelo firewall. Baseando-se na oitava regra estabelecida.



Figura 13 – Tela de tempo limite atingido – Tentativa de acesso de uma máquina bloqueada

## CONSIDERAÇÕES FINAIS

Baseando-se no que foi apresentado por ocasião deste artigo, é possível concluir que a adoção de mecanismos de otimização e políticas de controle de acesso à Internet em conjunto com a filtragem de pacotes em um ambiente corporativo é, sem dúvidas, de fundamental importância para o bom andamento das atividades e para segurança da informação. Vale lembrar, que os *firewalls* apenas fazem parte de um sistema de segurança da informação e, não representam o sistema em sua totalidade.

A junção de fatores humanos, físicos e tecnológicos está ligada diretamente à segurança da informação, daí a complexidade existente na manutenção de políticas e técnicas que abordam este assunto no ambiente corporativo. Baseando-se nas referências bibliográficas percorridas neste artigo, é possível afirmar que políticas de controle de acesso à Internet em conjunto com a filtragem de pacotes, não garantem a totalidade da segurança das informações no ambiente em questão, mas sim minimizam riscos e ameaças consequentes do acesso à Internet.

A segurança da informação é o alvo principal, porém, a adoção de um *firewall* na porta de entrada e saída de dados, não se torna o ponto principal a ser contemplado. De acordo com o que se verificou por ocasião deste artigo, a conscientização e a capacitação dos recursos humanos sobre a importância da segurança e o respeito aos riscos e ameaças, deve ser enfatizada cada vez mais no

decorrer dos afazeres das instituições. Vale ressaltar que, nenhum sistema de informação está livre de ameaças, o que causa a necessidade de que todos os envolvidos nos processos de TI, estejam comprometidos com as práticas de comportamento seguro e com a constante atualização de seus conhecimentos relativos às técnicas de segurança da informação.

## REFERÊNCIAS BIBLIOGRÁFICAS

ZWICHY, E. D.; COOPER, S; CHAPMAN, D. B. **Construindo Firewalls para a Internet**. 2ª ed. Rio de Janeiro: Campus, 2000.

MORIMOTO, C. E. **Servidores Linux – Guia prático**. Porto Alegre: Meridional, 2008.

JUCÁ, H. L. **Técnicas Avançadas de Conectividade e Firewall: em GNU/Linux**. Rio de Janeiro: Brasport, 2005.

FILHO, A. S. **Linux – Controle de Redes**. Florianópolis: Visual Books, 2009

NETO, U. **Dominando Linux Firewall Iptables**. Rio de Janeiro: Ciência Moderna, 2004.

MARCELO, A. **Squid - Configurando Proxy para Linux**: Brasport, 2005.

BASTOS, E. R. **Configurando um squid "ninja" -**  
<http://www.linuxman.pro.br/squid/>, 2003. Acesso em junho de 2011.

MEDEIROS, L. C. L. L.; SOARES, W. **Formação de Suporte Técnico – Proinfo**: RNP, 2010.

RICCI, B.; MENDONÇA, N. **Squid: Solução Definitiva**. Rio de Janeiro: Ciência Moderna, 2006.