

UNIVERSIDADE REGIONAL INTEGRADA DO
ALTO URUGUAI E DAS MISSÕES
URI – CAMPUS DE ERECHIM

DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM INFORMÁTICA

**IMPLEMENTAÇÃO E CONFIGURAÇÃO DE UM FIREWALL
USANDO LINUX**

Gustavo Pilotto Diehl
Jarbas Celante

Prof. Dr. Jacques Duílio Brancher
ORIENTADOR

ERECHIM/RS, JULHO DE 2001.

IMPLEMENTAÇÃO E CONFIGURAÇÃO DE UM FIREWALL USANDO LINUX

Trabalho apresentado como requisito à obtenção do grau de Bacharel em Informática pela Universidade Regional Integrada do Alto Uruguai e das Missões – URI, Campus de Erechim sob orientação do Professor Dr. **Jacques Duílio Brancher**.

ERECHIM/RS, JULHO DE 2001.

SUMÁRIO

| | |
|-----------------------------------------------------|-------------|
| SUMÁRIO | III |
| RESUMO | VI |
| ABSTRACT | VII |
| LISTA DE ABREVIATURAS | VIII |
| ÍNDICE DE FIGURAS | X |
| ÍNDICE DE FIGURAS | X |
| ÍNDICE DE TABELAS | XI |
| INTRODUÇÃO | 12 |
| 1. CONJUNTO TCP/IP | 14 |
| 1.1. COMO OS DADOS FLUEM PELO CONJUNTO TCP/IP | 15 |
| 1.2. ETHERNET | 16 |
| 1.3. ARP | 16 |
| 1.3.1. Tabela ARP para tradução de endereços | 16 |
| 1.3.2. Exemplo de tradução de endereço | 17 |
| 1.4. PROTOCOLO DE INTERNET (IP) | 20 |
| 1.4.1. Roteamento Direto | 20 |
| 1.4.2. Roteamento Indireto | 21 |
| 1.4.3. Regras de roteamento de módulo IP | 23 |
| 1.4.4. Endereço IP | 23 |
| 1.4.5. Nomes | 23 |
| 1.4.6. Tabela de roteamento IP | 25 |
| 1.4.7. Gerenciando as Rotas | 25 |
| 1.5. USER DATAGRAM PROTOCOLO (UDP) | 26 |
| 1.5.1. Portas | 26 |
| 1.5.2. Checksum (Soma de Verificação) | 27 |
| 1.6. TRANSMISSION CONTROL PROTOCOL (TCP) | 27 |

| | |
|----------------------------------------------------------------------------------------|-----------|
| 1.7. ICMP E DIAGNÓSTICOS DE REDE | 28 |
| 1.7.1. <i>Ping</i> | 28 |
| 1.7.2 <i>Características de filtragem de pacotes do ICMP</i> | 28 |
| 2. DISPOSITIVOS E FERRAMENTAS DE ATAQUE..... | 30 |
| 2.1. DISPOSITIVOS DESTRUTIVOS..... | 30 |
| 2.2. PROGRAMAS DE VARREDURA..... | 30 |
| 2.2.1. <i>Varreduras de Ping de Rede</i> | 31 |
| 2.2.2. <i>Consultas ICMP</i> | 31 |
| 2.2.3. <i>Varredura de Portas</i> | 31 |
| 2.2.3.1. Tipos de Varredura de Portas..... | 32 |
| 2.2.3.2. Detecção do Sistema Operacional..... | 33 |
| 2.2.4. <i>Sniffers</i> | 35 |
| 3. FIREWALL | 36 |
| 3.1. TECNOLOGIAS DE FIREWALL | 36 |
| 3.1.1. <i>NAT (Network Address Translation – Conversão de Endereços de Rede)</i> | 37 |
| 3.1.2. <i>VPN (Virtual Private Network – Rede Privada Virtual)</i> | 37 |
| 3.1.3. <i>Filtragem de Pacotes</i> | 37 |
| 3.1.3.1. Política da segurança da rede através do Firewall..... | 40 |
| 3.1.3.2. Vantagens da filtragem de pacotes..... | 41 |
| 3.1.3.3. Desvantagens da filtragem de pacotes..... | 42 |
| 3.1.4. <i>Serviços de Proxy</i> | 43 |
| 3.1.4.1. Vantagens do uso de proxies..... | 44 |
| 3.1.4.2. Desvantagens do uso de proxies | 45 |
| 4. IMPLEMENTAÇÃO..... | 47 |
| 4.1. LINUX | 47 |
| 4.2. IPCHAINS..... | 47 |
| 4.3. SQUID CACHE..... | 48 |
| 4.4. INSTALAÇÕES E ATUALIZAÇÕES NECESSÁRIAS | 48 |
| 4.5. CONFIGURAÇÕES | 49 |
| 4.5.1. <i>Kernel</i> | 49 |
| 4.5.2. <i>Interfaces de Rede</i> | 49 |
| 4.5.3. <i>Squid Cache</i> | 50 |

| | |
|-----------------------------------------|-----------|
| 4.5.4. <i>Ipchains</i> | 51 |
| 4.5.4.1. Início do Script: | 51 |
| 5. ANÁLISE DE RESULTADOS | 65 |
| CONCLUSÃO | 68 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 69 |

Resumo

O presente trabalho tem por objetivo definir uma política de segurança de acesso à Internet utilizando Firewall. Após um estudo sobre os Sistemas Operacionais utilizados para tal fim, optou-se pelo Sistema Operacional Linux, devido à sua confiabilidade, estabilidade e credibilidade no quesito segurança.

Empreendeu-se também um estudo sobre o conjunto TCP/IP, base para o sucesso da Internet, e os principais dispositivos e ferramentas de ataque.

Algumas tecnologias de Firewall foram estudadas no decorrer deste estudo. Dentre elas estão a VPN (Virtual Private Network – Rede Privada Virtual), o NAT (Network Address Translation – Conversão de Endereços de Rede), o Proxy de Aplicativo e a Filtragem de Pacotes.

Após este estudo optou-se por utilizar a tecnologia de Filtragem de Pacotes juntamente com o Proxy de Aplicativo.

Foram instalados e configurados o Sistema Operacional Linux com duas interfaces de rede, o software de filtragem de pacotes e o software proxy de aplicativo. Com o Firewall devidamente implementado, realizaram-se testes necessários para a verificação do funcionamento do mesmo.

Terminados os estudos teóricos e práticos, e realizados os testes necessários, constatou-se que o Firewall tem uma grande utilidade para as redes que utilizam conexão à Internet, mostrando-se eficiente e seguro, atingindo os fins desejados.

Palavras-Chave: Firewall, Segurança, Internet, TCP/IP, Ipchains,

Abstract

The present work has for objective to define a politics of access safety to the Internet using Firewall. After a study on the operating systems used for such end, the operating system Linux was chosen, due to its reliability, stability and credibility in the inquiry safety.

It was also undertaken a study on the group TCP/IP, base for the success of the Internet, and the main devices and tools of attacks.

Some technologies of Firewall were studied in elapsing of this study. Among they are VPN (Virtual Private Network), NAT (Network Address Translation), Proxy of Application and Filtering of Packages.

After this study it was opted for using the technology of Filtering of Packages together with Proxy of Application.

They were installed and configured the operating system Linux with two network interfaces, the software of filtering of packages and the software application proxy. With Firewall properly implemented, they took place necessary tests for the verification of the operation of the same.

Finished the theoretical and practical, and accomplished studies the necessary tests, it was verified that Firewall has a great usefulness for the nets that use connection to Internet, being shown efficient and safe, reaching the wanted ends.

Keywords: Firewall, Security, Internet, TCP/IP, Ipchains

Lista de Abreviaturas

ARP – Address Resolution Protocol

CPU – Central Processor Unit

CSMA/CD – Carrier Sense Multiple Access / Collision Detect

DDoS – Distributed Denial of Service

DNS – Domain Name System

DoS – Denial of Service

FTP – File Transfer Protocol

HTTP – Hyper Text Transfer Protocol

HTTPS – Hyper Text Transfer Protocol Secure

ICMP – Internet Control Message Protocol

IP – Internet Protocol

IRC – Internet Relay Chat

ISN – Initial Sequence Number

NAT – Network Address Translation

NFS – Network File System

NIC – Network Information Center

SMTP – Simple Mail Transfer Protocol

SNMP – Simple Network Message Protocol

SSH – Secure Shell

SSL – Secure Sockets Layer

TCP – Transmission Control Protocol

TCP/IP – Transmission Control Protocol / Internet Protocol

TFTP – Trivial File Transport Protocol

ToS – Type of Service

UDP – User Datagram Protocol

URL – Uniform Resource Locator

VPN – Virtual Private Network

Índice de Figuras

| | |
|----------------------------------------------------------------------------------------------------------|-----------|
| FIGURA 1.1 – NÓ DE UMA REDE TCP/IP BÁSICA | 14 |
| FIGURA 1.2 – MULTIPLEXADOR N-PARA-1 E DEMULTIPLEXADOR 1-PARA-N .. | 15 |
| FIGURA 1.3 – MULTIPLEXAÇÃO E DEMULTIPLEXAÇÃO DE UM PACOTE ATRAVÉS DAS CAMADAS OU NÍVEIS | 16 |
| FIGURA 1.4 – REDE IP ÚNICA | 20 |
| FIGURA 1.5 – REDE DE 3 IP'S – UMA INTRANET..... | 21 |
| FIGURA 3.1 – DIAGRAMA DE UMA REDE CONECTADA À INTERNET ATRAVÉS DE UM FIREWALL | 36 |

Índice de Tabelas

| | |
|--------------------------------------------------------------------------------------------|----|
| TABELA 1.1 – EXEMPLO DE TABELA ARP | 17 |
| TABELA 1.2 – EXEMPLO DE REQUISIÇÃO ARP | 18 |
| TABELA 1.3 – EXEMPLO DE RESPOSTA ARP..... | 18 |
| TABELA 1.4 – TABELA ARP APÓS A RESPOSTA..... | 19 |
| TABELA 1.5 – ENDEREÇO NO CABEÇALHO ETHERNET DE UM PACOTE IP DE A PARA B..... | 21 |
| TABELA 1.6 – ENDEREÇO NO CABEÇALHO ETHERNET DE UM PACOTE IP DE A PARA E (ANTES DE D) | 22 |
| TABELA 1.7 – ENDEREÇO NO CABEÇALHO ETHERNET DE UM PACOTE IP DE A PARA E (DEPOIS DE D)..... | 22 |
| TABELA 1.8 – EXEMPLO DE ARQUIVO COM NOMES DOS COMPUTADORES DA REDE | 24 |
| TABELA 1.9 – EXEMPLO DE ARQUIVO COM NOMES DAS REDES IP'S..... | 24 |
| TABELA 1.10 – EXEMPLO DE ARQUIVO COM OS NOMES DO ROTEADOR | 24 |
| TABELA 1.11 – TIPOS MAIS COMUNS DE MENSAGENS ICMP..... | 29 |
| TABELA 2.1 – OPÇÕES EXISTENTES NO CABEÇALHO DO PROTOCOLO TCP..... | 32 |

Introdução

A Internet nasceu no início da década de 60, para uso, inicialmente, a fins militares, e logo após estendeu-se as grandes universidades, que tinham como preocupação principal, a criação de sistemas operacionais e uniformização de protocolos para utilização da mesma. Não havia propósitos para a preocupação com a segurança, pois as pessoas estavam focadas no intuito de fazer a Internet funcionar, e não existia na mente destas ações de ataque e invasão de sistemas alheios.

A partir da década de 90, quando a Internet tornou-se comercial, os sistemas operacionais receberam grandes facilidades quanto à instalação, e a explosão do número de usuários, resultou num ponto crucial: a política de segurança ficou em segundo plano, ou simplesmente esquecida.

À medida que o comércio eletrônico começou a crescer, as empresas iniciaram um processo lento de controle e de segurança das informações, por conta da necessidade de se manter informações sigilosas tais como os dados pessoais dos clientes, números de cartões de crédito, entre outros, longe de pessoas com intenções não muito nobres.

Os principais “autores” destas preocupações são os chamados “Hackers” e “Crackers”, que embora confundidos, tem uma pequena diferença:

- Os Hackers são geralmente programadores, interessados em descobrir falhas em sistemas e suas razões, no intuito de resolver ou avisar sobre tais fatos.
- Os Crackers são uma espécie de criminosos virtuais, interessados em descobrir falhas e brechas para invadir sistemas, tendo como principal meta a destruição e o caos dos sistemas alheios.

E são os crackers que preocupam os peritos em segurança. Segundo uma pesquisa do site especializado em segurança Attrition.org, em 1999 foram 124 sites desfigurados no Brasil, e em 2000 foram 563 sites, uma alta de 454 %!

Apesar desta alarmante situação, a indústria da computação oculta este perigo aos usuários de pequeno e médio conhecimento, vendendo produtos de segurança, e garantindo que estes são capazes de proteger totalmente os seus sistemas. Porém quem possui um conhecimento maior nesta área é ciente de que a internet é de longe moderadamente segura, pois a cada dia um ou vários mecanismos de proteção são quebrados.

Uma das alternativas para tentar garantir a segurança de servidores na Internet, sendo uma das mais eficazes contanto que bem configurado, é o Firewall, podendo ser constituído por Hardware

e Software específicos, ou somente por Software em um servidor comum, como uma alternativa mais viável.

O Firewall é um sistema ou grupo de sistemas que reforça a política de controle de acesso entre computadores de uma rede. Basicamente é composto de dois mecanismos: um existe para bloquear o tráfego e outro para permiti-lo. Dependendo da ênfase do firewall ele se preocupará mais em bloquear o tráfego, enquanto outro priorizará a permissão do mesmo.

De nada adianta ter um firewall bem configurado, senhas fortes, utilização de canais seguros SSL (Secure Sockets Layer – Camada Segura de Sockets) se algum funcionário da empresa, com acesso ao sistema, pode invadir um servidor interno devido à excessiva preocupação em proteger a rede interna de usuários externos, e negligenciar a segurança interna. Infelizmente, com o advento da explosão da Internet, este é um dos maiores problemas encontrados na administração de redes.

Assim, o que se busca com o presente trabalho, é justamente apresentar alguns dos tipos de ataques mais comuns, e a implementação de um Firewall por software. Para tal, o presente documento foi dividido da seguinte maneira:

O primeiro capítulo, Conjunto de Protocolos TCP/IP (Transmission Control Protocol / Internet Protocol – Protocolo de Controle de Transmissão / Protocolo de Internet), abrange uma explicação de como funciona e para que serve o conjunto de protocolos TCP/IP. A Motivação para tal é o fato de que o funcionamento da Internet está totalmente baseado em cima de TCP/IP.

O segundo capítulo, Principais Dispositivos e Ferramentas de Ataque, demonstra os dispositivos e ferramentas de ataques mais utilizadas. A Importância deste é demonstrar os ataques contra a rede que o Firewall vai procurar proteger.

O terceiro capítulo, Firewall, descreve o que é um Firewall e para que serve, uma descrição dos tipos principais de Firewall e uma explicação mais abrangente nos dois tipos que serão utilizados neste trabalho. Importante para demonstrar os tipos e definir quando e onde utilizá-los.

O quarto capítulo, Implementação, descreve todos os procedimentos realizados para a implementação e configuração de um Firewall em um servidor. Importante para demonstrar como é feito a implementação e configuração de um Firewall.

O quinto capítulo, Análise de Resultados, demonstra algumas tentativas de ataque protegidas pelo firewall e alguns tráfegos necessários permitidos.

1. Conjunto TCP/IP

Para a implementação de um firewall é necessário ter conhecimentos sobre o que está por trás do funcionamento da Internet: o Conjunto TCP/IP.

O termo genérico TCP/IP usualmente está relacionado aos protocolos específicos TCP (Transmission Control Protocol – Protocolo de Controle de Transmissão) e IP (Internet Protocol – Protocolo de Internet). Também estão inclusos neste conjunto outros protocolos e aplicações. Exemplos destes protocolos são: UDP (User Datagram Protocol – Protocolo de Datagrama de Usuário), ARP (Address Resolution Protocol – Protocolo de Resolução de Endereço) e ICMP (Internet Message Control Protocol – Protocolo de Controle de Mensagens da Internet). Exemplos destas aplicações são: Telnet e FTP (File Transfer Protocol – Protocolo de Transferência de Arquivo). Um termo mais correto para este conjunto seria Tecnologia de Internet. Uma rede que usa esta Tecnologia de Internet é chamada intranet.

Para entender esta tecnologia, primeiro precisa-se aprender a seguinte estrutura lógica:

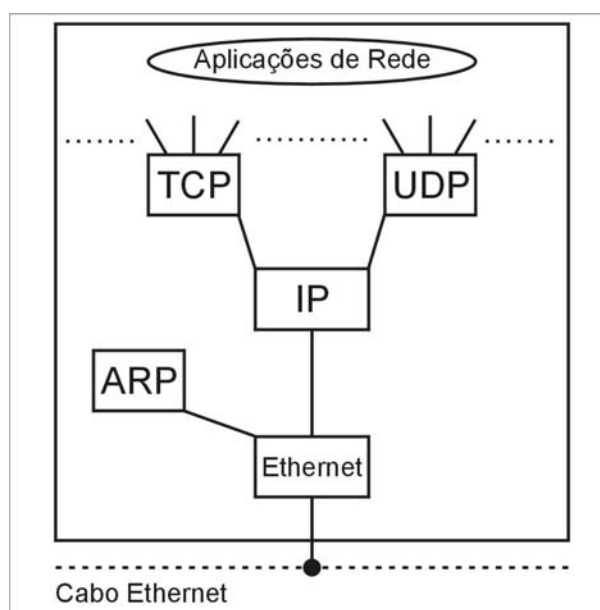


Figura 1.1 – Nó de uma rede TCP/IP básica.

Esta é a estrutura lógica dos protocolos dentro de um computador conectado à Internet. Cada computador pode se comunicar usando esta tecnologia de internet, e é esta estrutura lógica que determina o funcionamento do computador na Internet.

A figura 1.1 representa este funcionamento. Para aplicações que usam TCP, os dados passam da aplicação para o módulo TCP, e para aplicações que usam UDP os dados passam para o módulo UDP.

Tanto os módulos TCP e UDP quanto o driver Ethernet são multiplexadores n-para-1. Ou seja, como multiplexadores, eles transformam várias entradas em uma saída. Eles também são demultiplexadores 1-para-n, transformando uma entrada em várias saídas, de acordo com o campo “tipo” no cabeçalho do protocolo utilizado.

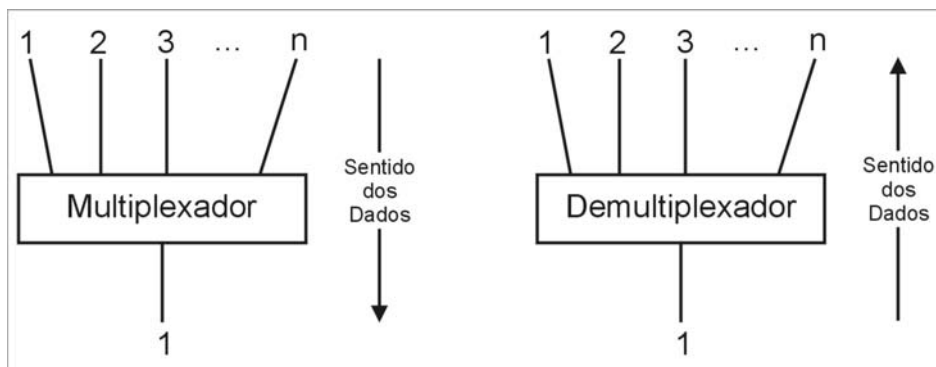


Figura 1.2 – Multiplexador n-para-1 e Demultiplexador 1-para-n.

1.1. Como os dados fluem pelo Conjunto TCP/IP

Multiplexação de um pacote:

A multiplexação é um caminho simples a ser percorrido, pois a cada ponto de início só há um caminho para o nível mais baixo, e cada protocolo adiciona ao pacote o seu próprio cabeçalho, para que o pacote possa ser demultiplexado posteriormente pelo computador destino.

Os dados passam da aplicação para os módulos TCP ou UDP, convergindo no módulo IP, onde descem ao último nível, o driver da interface de rede.

Demultiplexação de um pacote:

Se um pacote que trafega na rede passa pelo driver Ethernet (é o endereço destino), ele será repassado ao módulo IP ou ao módulo ARP, dependendo do valor contido no campo “tipo”, no cabeçalho deste pacote.

Se o pacote é um pacote IP, este é passado para o módulo IP, que repassa os dados aos módulos TCP ou UDP, dependendo do valor contido no campo “protocolo” encontrado no cabeçalho IP deste pacote.

Se o datagrama UDP é repassado ao módulo UDP, a mensagem de aplicação é repassada à aplicação de rede, baseado no valor do campo “porta” contido no cabeçalho UDP. Se o datagrama TCP é repassado ao módulo TCP, a mensagem de aplicação é repassada à aplicação de rede, baseado no valor do campo “porta” contido no cabeçalho TCP.

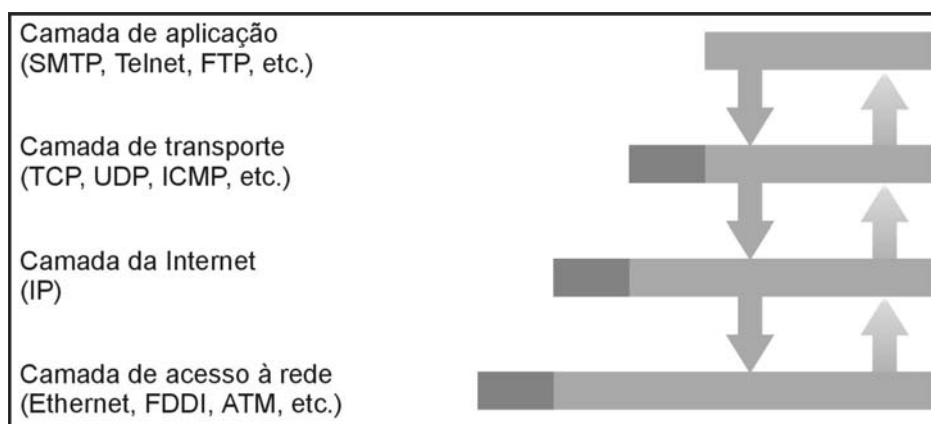


Figura 1.3 – Multiplexação e Demultiplexação de um pacote através das camadas ou níveis

1.2. Ethernet

Esta seção é uma pequena revisão da tecnologia Ethernet.

Um pacote Ethernet contém o endereço destino, endereço origem, o campo tipo e os dados.

Um endereço Ethernet tem seis bytes. Cada dispositivo tem o seu próprio endereço e só recebem os pacotes que contenham como endereço destino o próprio endereço. Porém todos os dispositivos recebem pacotes com endereço destino “FF-FF-FF-FF-FF-FF” (em hexadecimal), chamado de endereço “broadcast”.

A tecnologia Ethernet usa CSMA/CD (Carrier Sense and Multiple Access with Collision Detection – Múltiplos Acessos de Detecção de Portadora com Detecção de Colisão). O CSMA/CD funciona da seguinte forma: apenas um dispositivo pode transmitir dados por vez, porém todos podem receber estes dados simultaneamente. Se dois dispositivos tentarem transmitir ao mesmo tempo, é detectada a colisão de transmissão, então os dois dispositivos esperam um tempo randômico (mas curto) antes de tentarem transmitir novamente.

1.3. ARP

Quando um pacote IP está sendo enviado, um endereço Ethernet tem de ser determinado.

Neste momento entra em ação o ARP, que é usado para traduzir o endereço IP em endereço Ethernet. A tradução é feita somente para pacotes IP que estão sendo enviados, pois este é o momento de criação dos cabeçalho IP e Ethernet.

1.3.1. Tabela ARP para tradução de endereços

A tradução de endereços é realizada com a consulta de uma tabela. Esta tabela, chamada de Tabela ARP, é armazenada na memória e contém uma linha para cada computador. Há uma

coluna para o endereço IP e uma coluna para o endereço Ethernet. Para realizar a tradução de um endereço IP para um endereço Ethernet, a tabela é consultada para encontrar o endereço IP. A tabela a seguir é uma tabela ARP simplificada.

| Endereço IP | Endereço Ethernet |
|-------------|-------------------|
| 223.1.2.1 | 08-00-39-00-2F-C3 |
| 223.1.2.3 | 08-00-5A-21-A7-22 |
| 223.1.2.4 | 08-00-10-99-AC-54 |

Tabela 1.1 - Exemplo de tabela ARP.

Por convenção, o endereço IP, de quatro bytes, é escrito separando cada byte em decimal com um ponto. Já o endereço Ethernet, de seis bytes, é escrito separando cada byte em hexadecimal por um sinal de menos (-) ou por uma vírgula.

A tabela ARP é necessária porque tanto o endereço IP quanto o endereço Ethernet são selecionados independentemente, ou seja, não há um algoritmo para realizar esta tradução. O endereço IP é selecionado pelo administrador de rede, baseado no local em que o computador se encontra na internet. Quando um computador é movido para uma parte diferente da internet, seu endereço IP tem de ser mudado. Já o endereço Ethernet é definido pelo fabricante do dispositivo, sendo necessária a troca do dispositivo para que se mude o endereço Ethernet de um computador.

1.3.2. Exemplo de tradução de endereço

Durante uma operação normal, uma aplicação de rede, como o Telnet, envia uma mensagem para o TCP, que manda a mensagem TCP correspondente ao módulo IP. O endereço IP de destino é conhecido pela aplicação, pelo módulo TCP e pelo módulo IP. Neste ponto, o pacote IP já está construído e pronto para ser entregue ao driver Ethernet, mas primeiro o endereço Ethernet deve ser determinado.

A tabela ARP é consultada para determinar o endereço Ethernet de destino.

Mas como a tabela contém os endereços necessários? A resposta é que a tabela trabalha num sistema automático de armazenamento de endereços baseados num sistema tipo “*Procuro quando Preciso*”.

Ou seja, se a tabela ARP contém o endereço Ethernet destino, o pacote Ethernet é construído com o cabeçalho Ethernet, que contém o endereço IP de origem, endereço IP de destino, endereço Ethernet de origem e endereço Ethernet de destino, e é enviado pela rede.

Mas se na tabela ARP não consta o IP de destino para tradução, realizam-se dois procedimentos:

- a) Um Pacote de Requisição ARP com um endereço Ethernet broadcast é enviado a todos os computadores da rede;
- b) O pacote IP de saída é enfileirado.

Todos os computadores na rede recebem a requisição, e cada driver Ethernet examina o campo “tipo” na estrutura Ethernet e passa o pacote ARP para o módulo ARP. A requisição ARP diz “Se o seu endereço IP se iguala com este endereço IP de destino, então me informe seu endereço de Ethernet”.

Cada módulo ARP examina o endereço IP de destino e se este se igualar ao seu endereço IP, ele responde diretamente ao endereço Ethernet de origem. O pacote de resposta ARP diz “Sim, este endereço IP de destino é meu, deixe-me dizer meu endereço Ethernet”.

Um pacote de resposta ARP é enviado então como o endereço Ethernet requisitado.

A resposta é recebida pelo computador emissor original. O driver Ethernet examina o campo “tipo” na estrutura Ethernet e passa o pacote ARP ao módulo ARP. O módulo ARP examina o pacote ARP e adiciona o endereço IP e endereço de Ethernet do computador que respondeu à tabela ARP.

| | |
|------------------------------|-------------------|
| Endereço IP de Origem | 223.1.2.1 |
| Endereço Ethernet de Origem | 08-00-39-00-2F-C3 |
| Endereço IP de Destino | 223.1.2.2 |
| Endereço Ethernet de Destino | <em branco> |

Tabela 1.2 – Exemplo de Requisição ARP.

| | |
|------------------------------|-------------------|
| Endereço IP de Origem | 223.1.2.2 |
| Endereço Ethernet de Origem | 08-00-28-00-38-A9 |
| Endereço IP de Destino | 223.1.2.1 |
| Endereço Ethernet de Destino | 08-00-39-00-2F-C3 |

TABELA 1.3 – Exemplo de Resposta ARP.

A Tabela ARP, após a atualização fica assim:

| Endereço IP | Endereço Ethernet |
|-------------|-------------------|
| 223.1.2.1 | 08-00-39-00-2F-C3 |
| 223.1.2.2 | 08-00-28-00-38-A9 |
| 223.1.2.3 | 08-00-5A-21-A7-22 |
| 223.1.2.4 | 08-00-10-99-AC-54 |

Tabela 1.4 – Tabela ARP após a resposta.

A nova tradução então é instalada automaticamente na tabela, somente poucos milésimos de segundos depois de requisitada. Então após a tabela estar atualizada há mais três procedimentos a serem tomados para a completa emissão do pacote Ethernet. Abaixo estão listados os 5 procedimentos, que totalizam esta emissão do pacote Ethernet:

- a) Um Pacote de Requisição ARP com um endereço Ethernet broadcast é enviado a todos os computadores da rede;
- b) O pacote IP de saída é enfileirado;
- c) A resposta ARP é recebida com o endereço Ethernet necessário, e a tabela é atualizada;
- d) Para o pacote IP enfileirado, a tabela ARP é usada para traduzir o endereço IP para o endereço Ethernet.
- e) O pacote Ethernet é transmitido pelo driver Ethernet.

Cada computador tem uma tabela ARP separada para cada interface de rede Ethernet. Se o destino não existe, não haverá resposta ARP e nem entradas na tabela ARP. O módulo IP irá descartar os pacotes IP enviados para este endereço. Os níveis superiores de protocolos não sabem dizer a diferença entre uma Ethernet danificada e a ausência de um computador com este endereço IP de destino.

Algumas implementações de IP e ARP não enfileiram os pacotes IP enquanto aguardam a resposta ARP. Então como este pacote é descartado, a recuperação deste pacote fica a cargo do módulo TCP ou do aplicativo de rede UDP. Esta recuperação é feita através de Time-out e retransmissão. A mensagem retransmitida será enviada com sucesso, porque a primeira cópia já causou a atualização da tabela ARP.

1.4. Protocolo de Internet (IP)

O módulo IP é o centro para o sucesso da tecnologia da internet. Cada módulo ou driver adiciona o seu cabeçalho ao pacote enquanto o pacote vai descendo pelos níveis subseqüentes, assim como cada módulo ou driver retira o seu cabeçalho do pacote à medida em que ele sobe pelos níveis. O cabeçalho IP contém o endereço IP, que cria uma rede lógica única das múltiplas redes físicas. Esta interconexão das redes físicas é a origem do nome Internet. Um conjunto de redes físicas que limitam o alcance de um pacote IP é chamado Internet.

O IP esconde o hardware de rede das aplicações de rede. Se você criar uma nova rede física, pode pô-la em funcionamento implementando um novo driver que se conecta com o IP nativo da internet. Assim as aplicações de rede permanecem intactas e não são vulneráveis às mudanças na tecnologia de hardware.

A essência do IP é a sua tabela de roteamento. O IP usa esta tabela para tomar todas as decisões sobre rotear um pacote IP. O conteúdo da tabela de roteamento é determinado pelo administrador de rede. Erros bloqueiam a comunicação.

Para entender melhor a tabela de roteamento é melhor dar uma revisada em roteamento, e depois aprender sobre endereços de rede IP olhando detalhes.

1.4.1. Roteamento Direto

A figura a seguir mostra uma pequena intranet com 3 computadores: A, B e C. Cada computador tem o mesmo protocolo TCP/IP mostrado na Figura 1.1. Cada interface Ethernet de cada computador tem seu próprio endereço. Cada computador tem o seu endereço IP indicado ao módulo IP pelo administrador de rede, que também designou um número IP de rede para a Ethernet.

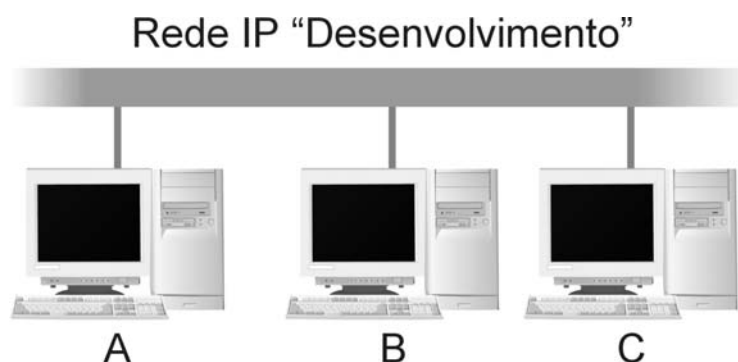


Figura 1.4 – Rede IP única.

Quando A envia um pacote IP para B, o cabeçalho IP contém como endereço IP de origem o endereço IP de A, e o cabeçalho Ethernet contém como endereço Ethernet de origem o endereço

Ethernet de A. Também, o cabeçalho IP contém como endereço IP de destino o endereço IP de B, e o cabeçalho Ethernet contém como endereço Ethernet de destino o endereço Ethernet de B.

| Endereço | Origem | Destino |
|--------------------|--------|---------|
| Cabeçalho IP | A | B |
| Cabeçalho Ethernet | A | B |

Tabela 1.5 – Endereço no cabeçalho Ethernet de um pacote IP de A para B.

Quando o modulo IP de B recebe o pacote IP enviado por A, ele verifica o endereço IP de destino, e se o endereço é o seu, ele passa os datagramas aos protocolos de níveis superiores.

Esta comunicação entre A e B usa roteamento direto.

1.4.2. Roteamento Indireto

A figura abaixo é uma visão mais realística de uma intranet. Ela é composta de 3 Ethernets e 3 redes IP's, conectadas por um roteador IP chamado computador D. Cada rede IP tem 4 computadores. Cada computador tem o seu endereço IP e o seu endereço Ethernet.

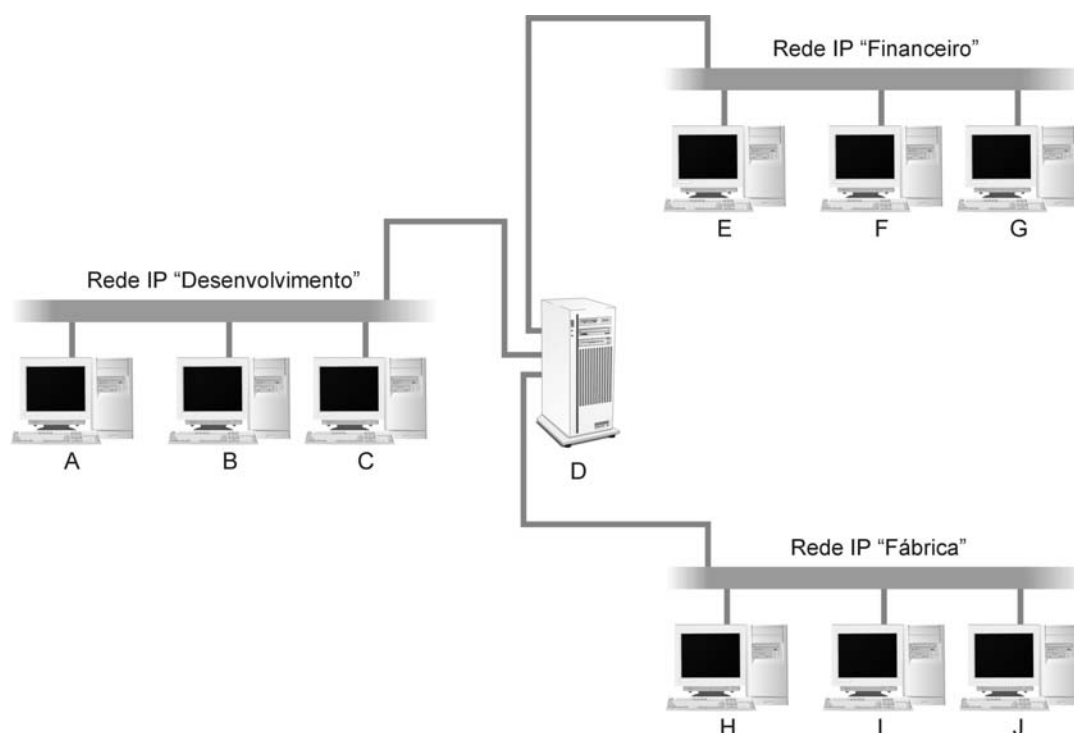


Figura 1.5 - Rede de 3 IP's – Uma Intranet.

Exceto pelo computador D, cada computador tem um protocolo TCP/IP como o demonstrado na Figura 1.1. O computador D é o roteador IP. Ele está conectado a todas as três redes, e possui três endereços IP e três endereços Ethernet. O computador D só tem um módulo IP.

O administrador de rede designou um número único, chamado de número IP de rede, para cada uma das Ethernets. O número IP de rede não é demonstrado nesta figura, somente os nomes de redes.

Quando o computador A envia um pacote IP para o computador B, o processo é idêntico ao roteamento direto. Qualquer comunicação entre computadores localizados em um único IP de rede funciona como o roteamento direto.

Quando o computador D e A se comunicam, é comunicação direta, assim como quando o computador D e E se comunicam ou quando o computador D e H se comunicam. Isto porque todos estes pares de computadores estão no mesmo IP de rede.

Contudo, quando o computador A se comunica com um computador do outro lado do roteador IP, a comunicação não é mais direta. O computador A precisa usar o computador D para enviar o pacote IP para a próxima rede IP. Esta comunicação é chamada indireta.

Estes roteamentos dos pacotes IP são feitos pelos módulos IP e acontece totalmente transparente para os protocolos TCP, UDP e as aplicações de rede.

Se A envia um pacote IP para E, o endereço IP de origem e o endereço Ethernet de origem são A. O endereço IP de destino é E, mas como o módulo IP de A enviou o pacote IP para D repassá-lo, o endereço Ethernet de destino é D.

| Endereço | Origem | Destino |
|--------------------|--------|---------|
| Cabeçalho IP | A | E |
| Cabeçalho Ethernet | A | D |

Tabela 1.6 – Endereço no cabeçalho Ethernet de um pacote IP de A para E (Antes de D).

O módulo IP de D recebe o pacote IP e, após examinar o endereço IP de destino, verifica que o pacote não é seu e repassa este pacote IP para E.

| Endereço | Origem | Destino |
|--------------------|--------|---------|
| Cabeçalho IP | A | E |
| Cabeçalho Ethernet | D | E |

Tabela 1.7 – Endereço no cabeçalho Ethernet de um pacote IP de A para E (Depois de D).

Resumindo, para a comunicação direta, os endereços IP e Ethernet de origem são os mesmos do emitente, e os endereços IP e Ethernet de destino são os mesmos do destinatário. Para a comunicação indireta, os endereços IP e Ethernet não são pares, deste modo.

1.4.3. Regras de roteamento de módulo IP

Vamos verificar as regras usadas pelo módulo IP para realizar o roteamento:

- a) Para um pacote IP de saída, recebendo um IP de um nível mais alto, o IP tem de decidir em enviar o pacote IP direta ou indiretamente, e o IP tem de escolher a interface de rede do nível mais baixo. Estas escolhas são feitas através da consulta da tabela de roteamento.
- b) Para um pacote IP de entrada, recebendo um IP de um nível mais baixo, o IP precisa decidir se repassa o pacote IP ou manda para um nível mais alto. Se o pacote IP será repassado, ele é tratado como um pacote IP de saída.
- c) Quando um pacote IP de entrada chega, ele nunca é repassado de volta pela mesma interface de rede.

Estas decisões são realizadas antes do pacote IP ser manejado pela interface mais baixa e antes da tabela ARP ser consultada.

1.4.4. Endereço IP

O administrador de redes determina os endereços IP's dos computadores de acordo com o IP da rede à qual estes computadores estão conectados. Uma parte dos quatro bytes do endereço IP é o número IP da rede, e a outra parte é o número IP do computador. Para um computador com endereço IP 223.1.2.1, o número IP de rede é 223.1.2 e o número IP do computador é 1.

O espaço de endereços IP's é administrado pelo NIC (Network Information Center – Centro de Informações de Rede). Todas as intranets que estão conectadas à Internet mundial somente podem usar números determinados pelo NIC. Se você está configurando sua própria internet, e pretende conectá-la à Internet, deve obter seu número de rede junto ao NIC. Ignorando isto corre-se o risco de criar confusões e um eventual caos, se por acaso utilizar um endereço IP já existente.

1.4.5. Nomes

As pessoas referem-se aos computadores por nomes, não números. Um computador chamado alpha, por exemplo, pode ter o endereço IP 223.1.2.1. Para pequenas redes, estas traduções de nomes para endereços são armazenadas em cada computador. Já em redes de maiores proporções, este arquivo é armazenado em um servidor e acessado pela rede quando necessário. Algumas linhas deste arquivo podem parecer assim:

| | |
|-----------|---------|
| 223.1.2.1 | alpha |
| 223.1.2.2 | beta |
| 223.1.2.3 | gamma |
| 223.1.2.4 | delta |
| 223.1.3.2 | epsilon |
| 223.1.4.2 | iota |

Tabela 1.8 – Exemplo de arquivo com nomes dos computadores da rede.

O endereço de IP é a primeira coluna e o nome do computador é a segunda coluna.

Redes IP também têm nomes. Se você tem três redes IP, o seu arquivo de redes que documenta estes nomes pode parecer assim:

| | |
|---------|-----------------|
| 223.1.2 | Desenvolvimento |
| 223.1.3 | Financeiro |
| 223.1.4 | Fabrica |

Tabela 1.9 – Exemplo de arquivo com nomes das redes IP's.

O número IP de rede é a primeira coluna e o nome da rede IP é a segunda coluna.

Para este exemplo, pode-se notar que alpha é o primeiro computador da rede desenvolvimento, assim como beta é o segundo computador da rede desenvolvimento, o epsilon é o segundo computador da rede financeiro, e assim por diante.

O arquivo de redes acima é adequado para os usuários, mas o administrador de redes provavelmente substituirá a linha do delta pelas linhas abaixo:

| | | |
|-----------|--------------|-------|
| 223.1.2.4 | desnetrouter | Delta |
| 223.1.3.1 | finnetrouter | |
| 223.1.4.1 | Fabnetrouter | |

Tabela 1.10 – Exemplo de arquivo com os nomes do roteador.

Estas três linhas do arquivo de rede concedem a delta um nome para cada endereço IP. Em fato, o primeiro endereço IP tem dois nomes listados: delta e desnetrouter são sinônimos. Na prática, delta é o nome geral do computador para propósitos gerais, e os outros três nomes são usados somente quando se está administrando a Tabela de Roteamento IP.

Estes arquivos são usados por comandos de administração de redes e aplicações de redes para providenciar os nomes dos computadores em uma rede. Eles não são requeridos para operações em uma intranet, mas tornam as coisas mais fáceis.

1.4.6. Tabela de roteamento IP

Como o IP sabe qual interface de rede do nível mais baixo usar para remeter um pacote IP? O IP consulta a tabela de roteamento usando uma chave de procura do número IP de rede, extraído do endereço IP de destino.

A tabela de roteamento contém uma linha para cada rota. As primeiras colunas da tabela de roteamento são: número IP de rede, marcador direto/indireto, endereço IP do roteador e o número da interface. Esta tabela é revisada pelo IP para cada pacote IP de saída.

Em muitos computadores, a tabela de roteamento pode ser modificada pelo comando “route”. O conteúdo da tabela de roteamento é definido pelo administrador da rede, porque é ele quem determina o endereço IP dos computadores.

Quando um pacote IP trafega por uma grande intranet, pode passar por muitos roteadores IP antes de chegar ao seu destino. O caminho que ele percorre não é determinado por uma origem central, mas do resultado da consulta em cada tabela de roteamento usada em sua jornada. Cada computador define somente o próximo ponto nesta jornada. Assim o pacote IP segue até chegar ao seu destino.

1.4.7. Gerenciando as Rotas

Manter tabelas de roteamento corretas em todos os computadores em uma grande intranet é uma tarefa difícil; configurações de redes são modificadas constantemente pelos administradores de redes para alcançar os propósitos necessários. Erros nas tabelas de roteamento podem bloquear a comunicação de tal maneira, que seu diagnóstico pode ser uma tarefa árdua e tediosa.

Manter a configuração de uma rede simples é muito diferente de manter uma intranet confiável. O melhor meio de designar redes IP para Ethernet é designando um único número IP de rede para cada Ethernet.

Certos protocolos e aplicações de rede dispõem de ajuda. O ICMP pode informar alguns problemas de roteamento. Para pequenas redes a tabela de roteamento é preenchida manualmente em cada computador pelo administrador de redes. Para redes maiores o administrador de redes automatiza esta operação manual com um protocolo de roteamento para as rotas pela rede.

Quando um computador é movido de uma rede IP para outra, o seu endereço IP tem que mudar. Quando um computador é removido de uma rede IP, seu antigo endereço torna-se inválido. Estas mudanças requerem atualizações freqüentes no arquivo de redes. Este arquivo pode tornar-se de difícil manutenção a partir de redes de médio porte. O DNS (Domain Name System – Sistema de Domínio de Nomes) ajuda a resolver estes problemas.

1.5. User Datagram Protocolo (UDP)

O UDP é um dos dois principais protocolos que ficam no nível acima do IP. Ele oferece serviços às aplicações de redes do usuário. Exemplos de aplicações de rede que usam UDP: NFS (Network File System – Sistema de Arquivos de Rede) e SNMP (Simple Network Management Protocol – Protocolo de Gerenciamento de Rede Simples). Este serviço é um pouco mais do que uma interface para o IP.

UDP é um serviço de entrega de datagramas que não garante a entrega. O UDP não mantém uma conexão fim-para-fim com o módulo UDP remoto; ele simplesmente envia os datagramas para a rede e aceita datagramas provenientes da rede.

O módulo UDP adiciona dois valores ao pacote a ser mandado para o IP. Um é a multiplexação da informação entre as aplicações baseadas em uma porta. O Outro é o “checksum” (soma de verificação), para verificar a integridade dos dados.

1.5.1. Portas

Como um programa cliente em um computador atinge um programa servidor em outro?

O caminho para a comunicação entre uma aplicação e o protocolo UDP é através das portas UDP. Estas portas são numeradas, começando do zero. Uma aplicação que está oferecendo serviços (o servidor) aguarda por mensagens que chegam em uma porta específica dedicada aquele serviço. O servidor aguarda pacientemente por qualquer cliente que requisitar o serviço.

Por exemplo, o servidor SNMP, chamado de agente SNMP, sempre aguarda na porta UDP 161. Somente pode haver um agente SNMP por computador, porque há somente uma porta UDP de número 161.

Se um cliente SNMP precisa de serviços, ele manda uma requisição à porta UDP de número 161 ao computador destino.

Quando uma aplicação manda dados através do UDP, estes chegam ao destino como uma unidade única. Por exemplo, se uma aplicação faz 5 escritas para a porta UDP, a aplicação do

destino precisará realizar 5 leituras na porta UDP. O tamanho de cada escrita tem de ser o mesmo do tamanho de cada leitura.

O UDP nunca trabalha com duas mensagens de aplicações juntas, ou divide uma mensagem de aplicação única em partes.

1.5.2. Checksum (Soma de Verificação)

Um pacote IP de entrada com um cabeçalho IP indicando UDP no campo “tipo” é repassado ao módulo UDP, após análise do módulo IP. Quando o módulo UDP recebe o datagrama UDP do IP ele examina o UDP checksum. Se o checksum é zero, pode parecer que o checksum não foi calculado pelo remetente e pode ser ignorado. O módulo UDP do computador remetente pode ou não gerar o checksum. Se a Ethernet é a única rede entre os dois módulos UDP em comunicação, então pode-se não precisar do checksumming. Entretanto, é recomendado sempre gerar o checksum, porque em algum ponto no futuro uma mudança na tabela de rotas pode enviar os dados por uma mídia menos confiável.

Se o checksum é válido (ou zero), a porta de destino é examinada e se uma aplicação está usando esta porta, uma mensagem de aplicação é enfileirada para que a aplicação a leia. Senão o datagrama UDP é descartado. Se os datagramas UDP de entrada chegam mais rápidos do que a aplicação possa ler e se a fila é preenchida ao máximo, os próximos datagramas UDP são descartados pelo módulo UDP. O UDP continuará a descartar os datagramas UDP até que haja espaço na fila novamente.

1.6. Transmission Control Protocol (TCP)

O TCP oferece um serviço diferente do UDP. O TCP oferece uma garantia de entrega dos pacotes, enquanto o UDP não.

O TCP é usado por aplicações de rede que requerem garantia de entrega e não podem perder tempo com Time-outs e retransmissões. Os dois aplicativos de rede mais típicos que usam TCP são o FTP e o Telnet. Outros aplicativos de rede populares que usam TCP são: Sistemas X-Window, SSH (Secure Shell – Shell Seguro), etc. Esta qualidade do TCP tem um custo: ele requer mais CPU (Central Unit Processor – Unidade Central de Processamento) e banda de rede. A parte interna do módulo TCP é muito mais complexa que a do módulo UDP.

Similar ao UDP, as aplicações de redes se conectam à portas TCP. As portas TCP definidas são dedicadas a aplicações específicas. Por exemplo, o Telnet usa a porta TCP de número 23. O

cliente Telnet pode achar o servidor simplesmente se conectando a porta TCP 23 no computador especificado.

Quando a aplicação inicia usando o TCP, o módulo TCP no computador cliente e o módulo TCP no computador servidor começam a se comunicarem. Estes dois pontos de módulos TCP contêm informações que definem um circuito virtual. Este circuito virtual consome recursos em ambos pontos TCP. Este circuito virtual é full-duplex, ou seja, os dados podem trafegar em ambas as direções simultaneamente. A aplicação escreve os dados para a porta TCP, os dados atravessam a rede e são recebidos pela porta TCP, sendo lidos pela aplicação no outro ponto.

Se uma aplicação realiza 5 escritas para a porta TCP, a aplicação no outro ponto pode realizar 10 leituras para pegar todos os dados. Ou pode pegar todos os dados com uma leitura só. Não há correlação entre o número e o tamanho das escritas em um ponto para com o número e o tamanho das leituras do outro.

1.7. ICMP e Diagnósticos de Rede

O ICMP e os protocolos e programas de diagnóstico são usados por administradores de rede para depurar problemas de rede.

O fato de serem protocolos de baixo nível faz com que eles sejam explorados com frequência em tentativas de ataques. Vários ataques de negação de serviço baseiam-se em pacotes ICMP mal formados.

Muitos sistemas de filtragem de pacotes permitem filtrar pacotes ICMP quase da mesma forma que pacotes TCP ou UDP. Utiliza-se para tanto, o código de tipo de mensagem ICMP em lugar do número da porta TCP ou UDP de origem ou destino.

1.7.1. Ping

O programa *ping* é usado para verificar a conectividade da rede. Ele gera um pacote ICMP “solicitação de eco”, e o sistema destino responde com um pacote ICMP “resposta de eco”. Geralmente, o ICMP é implementado no núcleo, e assim, é o núcleo que gera o pacote “resposta de eco”.

1.7.2 Características de filtragem de pacotes do ICMP

Os pacotes ICMP não têm números de portas de origem ou destino, porém, em vez disso, têm um único campo de tipo de mensagem ICMP. Muitos sistemas de filtragem de pacotes permitem

filtrar pacotes ICMP de acordo com esse campo, do mesmo modo que permitem filtrar pacotes TCP ou UDP baseados nos campos de números de portas de origem e destino.

Os tipos mais comuns de mensagens ICMP:

| Tipo de mensagem | Descrição |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | Resposta de eco (resposta para <i>ping</i>) |
| 3 | Destino inacessível. Pode indicar host inacessível, rede inacessível, porta inacessível ou outro |
| 4 | Extinção da origem (alguém dizendo ao destino “diminua a velocidade; você está falando muito rápido”) |
| 5 | Redirecionamento (alguém informando ao destino para alterar uma rota); deve ser ignorada pelos sistemas, a menos que venha de um roteador diretamente conectado. Os roteadores que fazem parte do firewall também devem ignorá-la |
| 8 | Solicitação de eco (gerada por <i>ping</i>) |
| 9 | Anúncio de roteador (usada por descoberta de roteador) |
| 10 | Seleção de roteador (usada por descoberta do roteador) |
| 11 | Tempo de duração excedido (o pacote parece estar em loop) |
| 12 | Problema de parâmetro (problema com um cabeçalho de pacote) |

Tabela 1.11 – Tipos mais comuns de mensagens ICMP.

2. Dispositivos e Ferramentas de Ataque

2.1. Dispositivos destrutivos

São programas que têm como objetivos principais a destruição de dados alheios e prejudicar outros usuários. A maioria destes dispositivos não apresenta, diretamente, riscos de segurança a um sistema, porém, podem afetar o funcionamento do mesmo. [AUT 00]

Uma grande preocupação com a segurança nestes casos, é que boa parte dos ataques é realizada por funcionários descontentes, e tendo eles um conhecimento geral do sistema da empresa, podem tornar o ataque o mais danoso possível. Mas a grande maioria dos ataques vem dos Crackers, que tem por objetivo molestar os usuários e administradores destes sistemas.

Um dos mais conhecidos dispositivos destrutivos é o DoS (Denial of Service - Ataques de Recusa de Serviço).

Os Ataques de Recusa de Serviço causam aborrecimentos que podem incapacitar temporariamente uma rede inteira. Consiste em derrubar o(s) host(s) incapacitando-o de ser acessado.

Ataques deste tipo são maliciosos e injustificáveis, pois não existem motivos para incapacitar o acesso de uma determinada rede à Internet, salvo os casos de teste de segurança.

Os ataques de DoS são direcionados à explorar falhas na construção de implementação de IP, sendo que conceituação deste é igual em quase todas as plataformas que o utilizam.

A mutação dos programas de DoS para afetar outros sistemas operacionais que não o de destino original, exige poucas alterações em sua codificação, tornando este tipo de ataque ainda mais freqüente e perigoso.

2.2. Programas de Varredura

São programas criados para detectar falhas de segurança em hosts remotos ou locais. Através destas varreduras, um usuário pode encontrar fraquezas na segurança de um computador a milhares de quilômetros de sua própria estação. [AUT 00]

Estas varreduras baseiam-se em consultas no conjunto de protocolos TCP/IP com o intuito de determinar quais serviços estão sendo executados, que usuários estão executando estes, se logins anônimos são suportados e quais serviços de rede requerem autenticação.

Estes programas não servem somente para fins maléficos, mas também são muito úteis para administradores de redes que necessitam de um padrão de segurança de alto nível, pois através deles viabiliza-se com maior facilidade a detecção de falhas da segurança de uma rede.

A grande importância destes programas de varreduras na Internet consiste na economia de tempo dos administradores de redes, sendo que para detectar tais falhas de segurança por eles achadas, necessitariam-se vários dias de serviços manuais.

Outra grande vantagem é a emissão de relatórios uniformemente formatados que facilitam referência e análise dos problemas detectados.

Abaixo, relacionaram-se os principais tipos de varreduras:

2.2.1. Varreduras de Ping de Rede

É um dos passos mais básicos no mapeamento de uma rede. Uma varredura ping automatizada em um intervalo de endereços IP e blocos de rede podem determinar se sistemas individuais estão ativos. O ping tradicionalmente envia pacotes ICMP para um determinado sistema, e aguarda sua resposta.

É muito utilizado na Internet para verificar endereços IP ativos, os quais respondem ao ping. Após conhecer os sistemas ativos, determinam-se os alvos de ataque.

2.2.2. Consultas ICMP

Com softwares de consultas de ICMP pode-se obter informações muito valiosas sobre um sistema.

Enviando pacotes ICMP pode-se obter desde a hora do sistema alvo para determinar em que fuso ele está até a máscara de uma placa de rede para determinar todas as sub-redes em uso.

Tendo este conhecimento, pode-se orientar ataques somente para as sub-redes específicas.

2.2.3. Varredura de Portas

A varredura de portas (*port scanning*) consiste na conexão à portas TCP e UDP do sistema alvo para determinar que serviços estão em execução, ou em estado de escuta.

A identificação de portas que estão escutando é fundamental para determinar o sistema operacional e aplicativos em uso. Serviços ativos ouvindo podem permitir a conexão de um usuário não autorizado a sistemas mal configurados ou que estejam executando software com falhas de segurança conhecidas.

Os objetivos principais da varredura de portas em um sistema alvo são identificar os serviços TCP e UDP em execução, identificar o tipo de sistema operacional ou identificar aplicativos ou versões específicas de um serviço em particular.

2.2.3.1. Tipos de Varredura de Portas

Existem várias técnicas de varreduras de portas disponíveis, sendo muitas delas obras diretas de Fyodor, um dos pioneiros na implementação destas. [AUT 00]

Para entender as opções do protocolo TCP, veja a tabela abaixo: [MCC 00]

| | |
|-----|-----------------|
| URG | Urgente |
| ACK | Confirmação |
| PSH | Empurrão |
| RST | Reinicialização |
| SYN | Sincronizar |
| FIN | Terminar |

Tabela 2.1 – Opções existentes no cabeçalho do protocolo TCP.

São elas:

- a) Varredura de Conexão TCP: conecta-se à porta do alvo e completa um handshake de 3 etapas completo (SYN, SYN/ACK, ACK), sendo facilmente detectável pelo sistema alvo.
- b) Varredura TCP SYN: é uma varredura semi-aberta porque não estabelece uma conexão TCP completa. Ou seja, um pacote SYN é enviado a porta alvo e se for obtido como resposta um pacote SYN/ACK pode-se dizer que a porta está escutando, porém se um pacote RST/ACK for recebido, significa que a porta não está escutando. Ainda, se o programa receber o pacote SYN/ACK ele envia um pacote RST/ACK à porta alvo em escuta para que uma conexão não se estabeleça, não havendo registros, ou rastros desta varredura.
- c) Varredura TCP FIN: esta técnica envia um pacote FIN para a porta alvo e o sistema alvo deve devolver um RST para cada porta fechada.
- d) Varredura TCP de árvore de Natal: esta técnica envia um pacote FIN, URG e PSH para a porta alvo e o sistema alvo deve devolver um RST para cada porta fechada.
- e) Varredura TCP nula: esta técnica desliga todos os flags. O sistema alvo deve devolver um RST para cada porta fechada.

f) Varredura UDP: esta técnica envia um pacote UDP para a porta alvo. Se a porta alvo responder com uma mensagem “ICMP port unreachable” a porta está fechada, do contrário deduzimos que a porta está aberta. Porém, como o UDP é um protocolo sem conexão, a precisão desta técnica depende de diversos fatores relacionados à utilização de recursos de sistema e rede, além de sua varredura ser muito lenta para dispositivos que utilizam filtragem de pacotes pesada.

2.2.3.2. Detecção do Sistema Operacional

Como visto anteriormente nosso primeiro objetivo na varredura de portas é identificar portas TCP e UDP que estão ouvindo no sistema alvo. Nosso segundo objetivo é determinar o tipo de sistema operacional do sistema que está sendo varrido.

A importância na identificação do sistema operacional alvo está no fato de poder-se, através do conhecimento de suas maiores vulnerabilidades, saber atingir o mesmo com a precisão necessária, para a obtenção do êxito esperado. Uma maneira eficiente e simples de obter informações sobre um sistema operacional é a utilização de uma técnica chamada banner através da qual verifica-se a versão dos serviços de execução.

Outra forma também eficiente é a utilização de ferramentas de obtenção de impressões digitais de pilha. Impressões Digitais de Pilha são conjuntos de sinais ou resultados sondados sobre um sistema operacional alvo, baseados na implementação da pilha de TCP/IP.

Por que obter Impressões Digitais de Pilha?

Com a obtenção destas pode-se determinar com alto grau de confiabilidade o sistema operacional alvo. Isto dá-se porque há muitas diferenças entre as implementações das pilhas TCP/IP de cada desenvolvedor.

Um desenvolvedor pode interpretar as orientações das RFC's de maneira diferente dos demais quando implementa sua pilha TCP/IP, daí pode-se averiguar qual o sistema operacional examinando as diferenças entre estas implementações.

Tipos de Sondas para a Diferenciação de um Sistema Operacional para outro.

- a) Sonda FIN: um pacote FIN é enviado a uma porta aberta. O comportamento correto seria não responder a esta requisição, porém, o Windows NT retorna um pacote FIN/ACK, por exemplo.
- b) Sonda de flag falso: no cabeçalho TCP de um pacote SYN um flag TCP indefinido é configurado em 1. Sistemas operacionais como o Linux, respondem com o flag configurado em 1 no seu pacote de resposta.

- c) Amostragem de ISN (Initial Sequence Number - Número de Seqüência Inicial): implica em encontrar um padrão no número seqüência inicial escolhida pela implementação TCP quando responde a uma solicitação de conexão.
- d) Monitoramento do “bit não fragmentar”: alguns Sistemas Operacionais, para melhorar o desempenho, configuram o “bit não fragmentar”. Através do monitoramento deste bit, pode-se determinar os sistemas operacionais com tal comportamento.
- e) Tamanho da janela inicial do TCP: consiste no rastreamento do tamanho inicial da janela de pacotes retornados. Em algumas implementações de pilha este tamanho é tão específico, que pode melhorar a precisão do mecanismo de impressões digitais.
- f) Valor ACK: o valor de seqüência para o campo ACK dos pacotes difere de uma pilha TCP/IP para outra. Há implementações que devolvem o número de seqüência enviado, enquanto outras devolvem o número de seqüência +1.
- g) Supressão de mensagens de erro ICMP: um sistema operacional pode limitar a taxa de envio de mensagens de erro. Enviando pacotes UDP para alguma porta de número alto aleatório, é possível contar o número de mensagens “não alcançáveis” recebidas dentro de um intervalo de tempo.
- h) Citação de mensagens ICMP: os sistemas operacionais diferem na quantidade de informações citadas quando erros ICMP são encontrados. Analisando estas mensagens de erro, pode-se realizar suposições sobre o sistema operacional do alvo.
- i) Mensagem de erro ICMP – Integridade de eco: ao examinar mensagens de erro ICMP pode-se fazer suposições sobre o sistema operacional do alvo, pois algumas implementações de pilha podem alterar os cabeçalhos IP quando devolvem estas mensagens de erro.
- j) TOS (Type of Service - Tipo de Serviço): em mensagens “porta ICMP não alcançada” o TOS é examinado. Grande parte das implementações de pilha usa 0, podendo haver variações.
- k) Manipulação de Fragmentação: cada pilha manipula fragmentos sobrepostos de forma diferente. Algumas pilhas sobrescrevem os dados antigos com os novos, e vice-versa, quando os fragmentos são remontados. Suposições poderão ser feitas ao verificar de que maneira os pacotes de sondagem são remontados.
- l) Opções do TCP: enviando um pacote com múltiplas opções configuradas, tais como nenhuma operação, tamanho máximo de segmento, fator de escala de janela e estampas de tempo pode-se determinar algumas características específicas do sistema operacional do alvo.

2.2.4. Sniffers

São dispositivos que capturam pacotes de rede. Seu principal propósito é analisar o tráfego de rede e identificar áreas potenciais de preocupação, ou seja, servem também para determinar possíveis problemas na rede.

No entanto, a principal utilização dos sniffers está longe deste propósito: é utilizado para representar um dos mais altos níveis de risco. Eis o por quê:

Os sniffers podem capturar senhas;

Os sniffers podem capturar informações confidenciais de proprietário;

Os sniffers podem ser utilizados para abrir brechas na segurança de redes vizinhas ou ganhar acessos de alto nível;

Complementando, se o seu sistema possui um sniffer não autorizado, possivelmente já está comprometido.

Eles capturarão todos os pacotes da rede, mas o atacante tem de ser seletivo. Os crackers geralmente examinam os primeiros 200-300 bytes de cada pacote. O nome de usuário e senha estão contidos nesta parte, o que, geralmente, é o objetivo deles.

3. Firewall

Firewall é um dispositivo criado para proteger uma rede, controlando o tráfego de pacotes de entrada e saída da mesma. O Firewall baseia-se em regras, definidas pelo administrador, e de acordo com a implementação realizada, ou nega todas requisições exceto as definidas, ou aceita todas as requisições e nega as definidas.

As regras são baseadas praticamente no endereço IP de origem das requisições, tanto de entrada quanto de saída, ou seja, quando uma máquina solicita o acesso à determinado IP/Porta, o Firewall avalia a origem da requisição e verifica se o mesmo é autorizado ou não.

Alguns Firewalls implementam funcionalidades adicionais como a capacidade de verificar conteúdo, a fim de localizar e bloquear ou liberar Java, JavaScript, VBScript e Scripts ActiveX e cookies, protegendo o usuário diretamente.

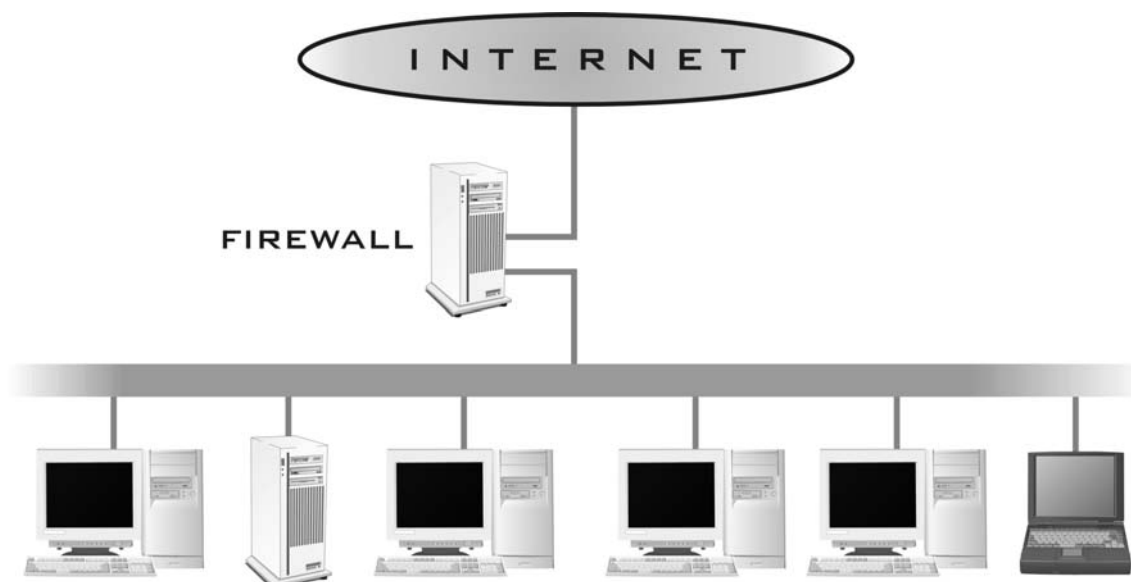


Figura 3.1 – Diagrama de uma rede conectada à Internet através de um Firewall.

3.1. Tecnologias de Firewall

Existem quatro tecnologias principais de implementação de Firewall: Proxy, Filtragem de pacotes, NAT (Network Address Translation – Tradução de Endereços de Rede) e VPN (Virtual Private Network – Rede Privada Virtual). Como a implementação do Firewall proposto por este trabalho é baseado em Filtragem de Pacotes e Proxy, estes serão focados com maior ênfase a seguir. [ZWI 00]

3.1.1. NAT (Network Address Translation – Conversão de Endereços de Rede)

Um procedimento pelo qual o roteador altera os dados em pacotes para modificar os endereços de rede. Isso permite a um roteador ocultar os endereços de hosts de rede em um dos seus lados. Essa técnica pode permitir que um grande número de hosts se conecte a Internet usando um pequeno número de endereços alocados ou pode permitir a uma rede configurada com endereços não válidos ou impossíveis de rotear conectar-se à Internet usando endereços válidos. Na realidade, essa não é uma técnica de segurança, embora possa proporcionar uma pequena dose de segurança adicional. Contudo, em geral ela é executada nos mesmos roteadores que fazem parte do firewall.

3.1.2. VPN (Virtual Private Network – Rede Privada Virtual)

Uma rede na qual os pacotes que são internos a uma rede privada passam através de uma rede pública sem que isso seja óbvio para hosts na rede privada. Em geral, as VPNs utilizam criptografia para proteger os pacotes a medida que eles passam através da rede pública. As soluções de VPN são populares, porque frequentemente é mais econômico conectar duas redes por meio de redes públicas (por exemplo, conectando ambas à Internet) que uni-las através de redes privadas (como conexões tradicionais de linhas dedicadas entre os sites).

3.1.3. Filtragem de Pacotes

Os sistemas de Filtragem de Pacotes fazem o roteamento de pacotes entre hosts internos e externos, mas o fazem seletivamente. Eles permitem ou bloqueiam certos tipos de pacotes de um modo que reflete a própria política de segurança de um site. O tipo de roteador usado em um firewall de filtragem de pacotes é conhecido como um roteador de triagem. [ZWI 00]

Todo pacote tem um conjunto de cabeçalhos contendo certas informações. As principais informações são:

- a) Endereço IP de Origem;
- b) Endereço IP de Destino;
- c) Protocolo (TCP, UDP ou ICMP);
- d) Porta TCP ou UDP de origem;
- e) Porta TCP ou UDP de destino;
- f) Tipo de mensagem ICMP;
- g) Tamanho do pacote.

O roteador pode inspecionar, além dos cabeçalhos de pacotes, dados presentes mais adiante no pacote; isso permite, por exemplo, filtrar pacotes com base em informações mais detalhadas (como o nome da página da web que alguém está solicitando) e verificar quais pacotes parecem estar formatados da maneira esperada para sua porta de destino. O roteador também pode ter certeza de que o pacote é válido (por exemplo, tem realmente o tamanho que afirma ter e esse é um tamanho válido), o que ajuda a captar vários ataques de negação de serviço baseados em pacotes defeituosos.

Além disso, o roteador conhece detalhes sobre o pacote que não estão refletidos no próprio pacote, como:

- a) A interface a qual chega o pacote;
- b) A interface para onde o pacote vai.

Finalmente, um roteador que mantém o controle de pacotes que enviou/repassou conhece alguns fatos históricos úteis, tais como:

- a) Se esse pacote parece ser uma resposta a outro pacote (isto é, sua origem era o destino de um pacote recente e seu destino é a origem deste outro pacote);
- b) Quantos outros pacotes foram vistos recentemente de/para o mesmo host;
- c) Se esse pacote é idêntico a um pacote visto recentemente;
- d) Se esse pacote é parte de um pacote maior que foi dividido em partes (fragmentado).

Para entender como a filtragem de pacotes funciona, vamos observar a diferença entre um roteador comum e um roteador de triagem.

Um roteador comum simplesmente examina o endereço de destino de cada pacote e escolhe o melhor caminho que conhece para enviar este pacote em direção ao destino. A decisão sobre como tratar o pacote se baseia unicamente em seu destino: há duas possibilidades: o roteador sabe como enviar o pacote ao seu destino e assim o faz; ou o roteador não sabe como enviar o pacote em direção a seu destino, e esquece o pacote e retorna uma mensagem ICMP “Destination Unreachable” à origem do pacote.

Por outro lado, um roteador de triagem examina pacotes mais de perto. Além de determinar se pode ou não rotear um pacote até seu destino, um roteador de triagem também descobre se deve ou não fazê-lo. O fato de “dever” ou “não dever” é estabelecido pela política de segurança do site que o roteador de triagem foi configurado para impor.

A filtragem de pacotes também pode ser executada por dispositivos que prestam atenção apenas à “dever” e “não dever” e que não tem nenhuma capacidade para rotear. Dispositivos que fazem isso são pontes de filtragem de pacotes. Eles são mais raros que os roteadores de filtragem de pacotes, porque são dispositivos de segurança dedicados que não fornecem todas as outras funções que os roteadores oferecem. A maioria dos sites prefere adicionar recursos e roteadores dos quais eles precisam de qualquer maneira, em vez de adicionar um dispositivo dedicado. Porém, ser um dispositivo dedicado fornece vantagens para as pontes de filtragem de pacotes; em particular, elas são mais difíceis de detectar e atacar que os roteadores de filtragem de pacotes.

Depois de examinar todas as informações, um roteador de filtragem de pacotes direto pode executar qualquer das ações a seguir:

- a) Encaminhar o pacote ao destino para o qual ele foi criado;
- b) Descartar o pacote – simplesmente esquece-lo, sem notificar o remetente;
- c) Rejeitar o pacote – recusar-se a encaminhá-lo e retornar um erro ao remetente;
- d) Registrar informações sobre o pacote.
- e) Ativar um alarme para notificar alguém sobre o pacote imediatamente;

Roteadores mais sofisticados também podem ter a capacidade de executar uma ou mais destas ações:

- a) Modificar o pacote (por exemplo, fazer a conversão de endereços de rede);
- b) Encaminhar o pacote a um destino diferente daquele para o qual ele foi criado (por exemplo, forçar transações através de um servidor proxy ou executar o balanceamento da carga);
- c) Modificar as regras de filtragem (por exemplo, para aceitar respostas a um pacote UDP ou negar todo o tráfego de um site que tenha enviado pacotes hostis).

O fato de servidores para determinados serviços da Internet residirem em certos números de portas torna possível ao roteador bloquear ou permitir determinados tipos de conexões, simplesmente especificando o número de porta apropriada (por exemplo, a porta TCP 23 para conexões Telnet) no conjunto de regras especificadas para filtragem de pacotes.

Três exemplos de como programar um roteador de triagem para encaminhar seletivamente os pacotes de ou para um site:

- a) Bloquear todas as conexões de entrada provenientes de sistemas fora da rede interna, exceto conexões de entrada SMTP (Simple Mail Transfer Protocol – Protocolo Simples de Transferência de Email) (de forma que seja permitido receber correio eletrônico);
- b) Bloquear todas as conexões de ou para certos sistemas nos quais não se confia;
- c) Permitir serviços de correio eletrônico e FTP, mas bloquear serviços perigosos como TFTP (Trivial File Transport Protocol – Protocolo Simplificado de Transferência de Arquivos), o X Window System e os serviços “r” (rlogin, rsh, rcp, etc.).

Os dispositivos de filtragem de pacotes que controlam os pacotes que vêm são chamados com frequência filtros de pacotes de estado (porque mantêm informações sobre o estado de transações). Eles também podem ser chamados filtros de pacotes dinâmicos, porque mudam dinamicamente seu tratamento de pacotes dependendo do tráfego que vêm. Dispositivos que examinam o conteúdo de pacotes, em vez de observarem apenas seus cabeçalhos, são chamados com frequência filtros de pacotes inteligentes. Na prática, quase todos os filtros de pacotes de estado também são capazes de examinar o conteúdo de pacotes, e muitos também são capazes de modificar o conteúdo de pacotes; assim, pode-se ver todos esses recursos reunidos sob o título geral “filtragem de pacotes de estado”. Porém, algo pode ser chamado de forma legítima um “filtro de pacotes de estado”, sem ter a capacidade de realizar filtragem ou modificação de conteúdo avançado.

Um sistema de filtragem de pacotes também é um lugar lógico para fornecer serviços de conversão de VPN ou de NAT. Tendo em vista que o filtro de pacotes já está examinando todos os pacotes, ele pode identificar facilmente pacotes que devem ir para um destino que faz parte da rede privada virtual, criptografar esses pacotes e encapsulá-los em outro pacote rumo ao destino apropriado.

3.1.3.1. Política da segurança da rede através do Firewall

A política de segurança da rede através do firewall determina quais serviços, protocolos e portas serão permitidos ou barrados. Definem-se também as exceções dessas regras (política).

Dois métodos podem ser utilizados como definição de política de segurança:

- a) Barra-se tudo e especifica-se o que se deseja liberar;
- b) Libera-se tudo e barra-se o que for especificado.

O primeiro método bloqueia todo o tráfego entre redes, com exceção dos serviços, portas e protocolos liberados. Este método é o mais seguro, porém, exige um alto grau de conhecimento do administrador de redes, pois os serviços deverão ser configurados (liberados) um a um.

Outra maneira é a liberação total com o bloqueio apenas do tráfego especificado. Com a crescente demanda por segurança, consequência do crescimento dos ataques às redes, este método não é o mais recomendável. Sempre surgirão novos bugs de segurança que utilizarão outras portas e/ou serviços que não estarão especificados como regra de firewall e, neste caso, o firewall de nada adianta.

3.1.3.2. Vantagens da filtragem de pacotes

a) Um roteador de triagem pode ajudar a proteger uma rede inteira

Uma das vantagens da filtragem de pacotes é que um único roteador de filtragem de pacotes posicionado estrategicamente pode ajudar a proteger uma rede inteira. Se apenas um roteador conectar um site à Internet, obter-se-á um enorme impulso em segurança de rede, não importando o tamanho do site, fazendo a filtragem de pacotes nesse roteador.

b) A filtragem de pacotes simples é extremamente eficiente

Como a filtragem de pacotes simples exige atenção apenas em alguns cabeçalhos de pacotes, ela pode ser feita com overhead muito baixo. O uso de proxies é uma operação bastante demorada, e adicionar o proxy significa orientar conexões através de outro programa, em geral em uma máquina que do contrário não seria necessária para o processo de roteamento. A filtragem de pacotes ocorre em uma máquina que já estava no caminho crítico e introduz um retardo muito menor.

Contudo, não existe nada grátis; quanto maior o trabalho executado por seus filtros de pacotes, mais lentos eles serão. Se os filtros de pacotes se comportam como proxies, executando complicadas operações orientadas para dados que exigem o controle de vários pacotes, eles também tenderão a funcionar como proxies.

c) Filtragem de pacotes está amplamente disponível

Os recursos de filtragem de pacotes estão disponíveis em muitos produtos de hardware e software para roteamento, tanto comerciais quanto gratuitamente disponíveis na internet. A maioria dos sites já tem recursos de filtragem de pacotes presentes nos roteadores que eles utilizam.

A maioria dos produtos comerciais de roteadores inclui recursos de filtragem de pacotes. Os recursos de filtragem de pacotes também estão disponíveis para vários computadores de uso geral.

3.1.3.3. Desvantagens da filtragem de pacotes

a) As ferramentas de filtragem atuais não são perfeitas.

Apesar de ampla disponibilidade de filtragem de pacotes em diversos produtos de hardware e software, a filtragem de pacotes ainda não é uma ferramenta perfeita. Os recursos de filtragem de pacotes de muitos desses produtos compartilham, em maior ou menor grau, limitações comuns:

a) As regras de filtragem de pacotes tendem a ser difíceis de configurar. Embora a dificuldade varie, ela vai desde o ligeiramente complicado até o praticamente impossível;

b) Uma vez configuradas, as regras de filtragem de pacotes tendem a ser difíceis de testar;

c) Os recursos de filtragem de pacotes de muitos produtos são incompletos, tornando difícil ou impossível a implementação de certos tipos de filtros altamente desejáveis;

d) Como qualquer outra coisa, os produtos de filtragem de pacotes podem ter bugs; esses bugs têm maior probabilidade que os bugs de proxies de resultarem em problemas de segurança. Em geral, um proxy que falha simplesmente deixa de repassar dados, enquanto uma implementação de filtragem de pacotes que falha pode permitir a passagem de pacotes que deveriam ser negados.

b) A filtragem de pacotes reduz o desempenho do roteador

A filtragem de pacotes impõe uma carga extra significativa sobre um roteador. Filtros mais complexos impõem uma carga maior sobre o roteador; porém, em alguns casos, a simples ativação da filtragem de pacotes em uma dada interface também pode custar muito em desempenho no caso de alguns roteadores, porque a filtragem é incompatível com certas estratégias de cache de uso comum para melhorar o desempenho.

c) Algumas normas não podem ser impostas prontamente pelos roteadores normais de filtragem de pacotes.

As informações que um roteador de filtragem de pacotes tem disponíveis para ele não permitem que sejam especificadas algumas regras que talvez fossem desejáveis. Por exemplo, os pacotes informam de qual host eles vêm, mas em geral não informam de qual usuário. Então, não é possível impor restrições sobre usuários específicos. De modo semelhante, os pacotes que informam para que porta estão indo, mas não para qual aplicativo; ao impor restrições sobre

protocolos de nível mais alto, faz-se por número de porta, na esperança de que nada mais esteja funcionando na porta atribuída a esse protocolo. Usuários maliciosos podem subverter com facilidade esse tipo de controle.

Esse problema é acentuado pelo uso de filtros de pacotes mais inteligentes; porém, em cada caso, pode ser necessário desistir de algumas vantagens da filtragem de pacotes normal. Por exemplo, um filtro de pacotes pode insistir que os usuários se autentiquem antes de enviar pacotes, e depois pode filtrar pacotes pelo nome de usuário. Contudo, isso remove a vantagem da transparência da filtragem de pacotes normal. Um filtro de pacotes também pode realizar a verificação da validade de protocolos, mas isso não é perfeito e também aumenta o overhead da filtragem.

3.1.4. Serviços de Proxy

Em geral, um proxy (procurador) é algo ou alguém que faz algo em nome de outro. [ZWI 00]

Os serviços de proxy são programas aplicativos ou servidores especializados que tomam as solicitações de usuários de serviços da Internet (como FTP, Telnet e HTTP (Hyper Text Transfer Protocol – Protocolo de Transferência de Hiper Texto)) e os encaminham aos serviços reais. Os proxies fornecem conexões substitutas e atuam como gateways para os serviços. Por essa razão, os proxies são às vezes conhecidos como gateways do nível de aplicativos.

Existem também proxies projetados principalmente visando à eficiência de rede em vez da segurança; esses proxies são proxies de cache, que mantêm cópias das informações correspondentes a cada pedido que eles representam. A vantagem de um proxy de cache é que, se diversos hosts internos solicitam os mesmos dados, os dados podem ser diretamente fornecidos pelo proxy. Os proxies de cache podem reduzir significativamente a carga em conexões de rede. Existem servidores proxy que oferecem tanto segurança quanto cache (Squid-Cache).

Os serviços de proxy estão situados, de forma mais ou menos transparente, entre um usuário no lado de dentro (na rede interna) e um serviço no lado de fora (na Internet). Em vez de comunicarem diretamente uns com os outros, cada um deles se comunicam com um proxy. Os proxies manipulam nos bastidores toda a comunicação entre usuários e serviços de Internet.

A transparência é o principal benefício de serviços de proxy. Ela é essencialmente ilusória. Para o usuário, um servidor proxy apresenta a ilusão de que o usuário está lidando diretamente com o servidor real. Para o servidor real, o servidor proxy apresenta a ilusão de que o servidor real está lidando diretamente com um usuário no host proxy.

Um serviço de proxy exige dois componentes: um servidor proxy e um cliente proxy. Um cliente proxy é uma versão especial de um programa cliente normal (por exemplo, cliente FTP ou Telnet) que se comunica com o servidor proxy, em vez de se comunicar diretamente com o servidor real na Internet; em algumas configurações, programas clientes normais podem ser usados como clientes proxy. O servidor proxy avalia pedidos do cliente proxy e decide o que aprovar e o que negar. Se um pedido é aprovado, o servidor proxy entra em contato com o servidor real em nome do cliente (daí o termo proxy – procurador) e continua a retransmitir pedidos do cliente proxy ao servidor real e respostas do servidor real ao cliente proxy.

Existem sistemas que oferecem um híbrido entre filtragem de pacotes e o uso de proxies, onde um dispositivo de rede intercepta a conexão e atua como proxy ou redireciona o tráfego para um servidor proxy. Isso permite a utilização de proxies sem a necessidade de alterar as configurações dos hosts dos usuários.

Além de apenas encaminhar os pedidos dos usuários para os serviços reais da Internet, os servidores proxy podem controlar o que os usuários fazem, de modo que, é capaz de decidir sobre os pedidos que processa, concedendo ou não o acesso a determinado serviço. Por exemplo, o proxy HTTP pode impedir o acesso a determinados sites. Essas configurações também podem ser ou não aplicadas a todos os hosts da rede.

Dá-se o nome de proxies genéricos ou encaminhadores de portas aos proxies que simplesmente encaminham pacotes, sem fazer qualquer análise sobre eles.

3.1.4.1. Vantagens do uso de proxies

a) Os serviços de proxy podem ser bons no registro de log

Tendo em vista que servidores proxy podem reconhecer o protocolo do aplicativo, eles podem permitir que o registro de log seja executado de maneira mais eficiente. Por exemplo, um proxy FTP pode gerar um log apenas com os comandos enviados pelo usuário e com as respostas recebidas do servidor, ignorando assim os dados transferidos e obtendo um log muito menor e muito mais útil.

b) Os serviços de proxy podem fornecer cache

Como todas as requisições de acesso passam através do serviço de proxy, o servidor proxy pode oferecer o serviço de cache, ou seja, pode manter cópias locais dos dados solicitados. Isso pode reduzir consideravelmente a carga sobre link de rede além de fornecer maior desempenho nas solicitações de dados dos usuários.

c) Os serviços de proxy podem fazer uma filtragem inteligente

Os proxies, ao examinar conexões específicas, são capazes de efetuar filtragem de pacotes de modo mais inteligente que um filtro de pacotes. Por exemplo, um proxy HTTP é capaz de fazer filtragem por tipo de conteúdo, eliminando, por exemplo, JAVA ou JavaScript.

d) Os sistemas de proxy podem executar autenticação ao nível de usuário

Como um sistema de proxy está ativamente envolvido na conexão, é fácil para ele fazer a autenticação do usuário e executar ações que dependam do usuário envolvido.

e) Os sistemas de proxy fornecem proteção automática no caso de implementações IP deficientes ou com falhas

Ao situar-se entre um cliente e a Internet, um sistema proxy gera pacotes IP completamente novos para o cliente. Dessa maneira, ele pode proteger clientes contra pacotes IP mal formados. Para isso, precisa-se apenas, que o sistema proxy não seja vulnerável a pacotes defeituosos.

3.1.4.2. Desvantagens do uso de proxies

a) Os serviços de proxy ficam defasados em relação a serviços que não usam proxies

Para os serviços mais simples como FTP e Telnet, existe software de proxy amplamente disponível. O mesmo não é verdadeiro para serviços novos ou menos utilizados. Há uma defasagem perceptível entre a introdução de um serviço e a disponibilidade de servidores para ele que utilizem proxies. Até surgir software proxy adequado para um novo serviço que precisa ser oferecido por um site, este pode exigir a necessidade de funcionar fora do firewall abrindo assim brechas potenciais de segurança.

b) Os serviços de proxy podem exigir servidores diferentes para cada serviço

Talvez seja necessário um servidor proxy diferente para cada protocolo, porque o servidor proxy pode precisar reconhecer o protocolo, a fim de determinar o que será permitido ou não, e também para se disfarçar com um cliente para o servidor real e como o servidor real para o cliente proxy. Reunir, instalar e configurar todos esses servidores pode ser muito trabalhoso. Pode ser menos complicado utilizar um proxy genérico, mas os proxies genéricos só oferecem os mesmos tipos de proteção e funcionalidade que um filtro de pacotes pode oferecer.

c) Os serviços de proxy normalmente exigem modificações em clientes, aplicativos ou procedimentos

Exceto no caso de serviços projetados para utilizar proxies, será necessário empregar clientes, aplicativos, e/ou procedimentos modificados. Essas modificações podem ter desvantagens; as pessoas nem sempre usam as ferramentas prontamente disponíveis com suas instruções normais.

Por causa dessas modificações, os aplicativos preparados para o uso de proxies nem sempre funcionam tão bem quanto os aplicativos que não usam proxies. Eles tendem a mudar as especificações de protocolos, e alguns clientes e servidores são menos flexíveis que outros.

4. Implementação

Este capítulo visa demonstrar como foi realizada a implementação de um firewall, contendo os softwares utilizados, uma descrição sobre eles e as instalações, atualizações e configurações necessárias para tal.

Para implementar o firewall utilizaram-se o Sistema Operacional Linux com o software nativo Iptables (Filtragem de Pacotes) e o software Squid Cache (Proxy de Aplicativo).

4.1. Linux

O Linux é um sistema operacional freeware, baseado no Unix, e tem o seu código fonte aberto (*opensource*). É recomendado para instalações de servidores de Internet, servidores de Email e Firewalls por sua confiabilidade e estabilidade, mas também pode ser utilizado para estações de trabalho (processamento de textos, edições de imagens, planilhas eletrônicas, navegação na Internet, etc...).

“Linux” é usado de duas maneiras: especificamente para referir-se ao kernel em si e para referir-se à qualquer conjunto de aplicativos que sejam executados no kernel, normalmente referido como distribuição. A tarefa do kernel é oferecer o ambiente global em que os aplicativos possam ser executados, incluindo as interfaces básicas com o hardware e o sistema de gerenciamento de tarefas e programas que estejam em execução.

4.2. Iptables

O Iptables é um software *opensource* destinado à administração de regras de firewall. Seu objetivo é configurar, manter e inspecionar regras do kernel do Linux que são divididas nas seguintes partes:

- a) Iptables de entrada (input);
- b) Iptables de saída (output);
- c) Iptables de retransmissão (forwarding).

As regras de entrada (input) definem as permissões dos serviços que poderão ser executados e acessados pelos usuários externos ou não à rede interna. Um bom exemplo de serviço a ser liberado é o acesso ao site da empresa. Se uma máquina requisita um serviço de qualquer máquina que está protegida pelo firewall e, que está na rede interna, o firewall, poderá ter uma regra que impeça o acesso a este serviço, e o mesmo será negado. O contrário também é válido, ou seja, se um serviço solicitado está liberado, o mesmo poderá ser acessado. Nesta política

podem ser definidos escopos de máquinas que poderão acessar, bem como, simultaneamente, máquinas que não poderão acessar o determinado recurso.

Regras de saída (output) são usadas para definir os serviços e portas que os usuários da rede interna poderão acessar externamente. É imprescindível a definição destas regras. Muitos usuários podem utilizar-se do acesso à internet para utilização de ferramentas de varreduras e invasão de redes. A preocupação dos administradores de rede concentra-se também neste fato. Muitos dos seus usuários poderão tentar acessos indevidos, e se, identificados poderão acarretar sérios problemas para a sua empresa.

O encaminhamento (forwarding) consiste em estabelecer regras de retransmissão de solicitações de acesso. Se um usuário está sob um firewall ele necessitará uma regra de forwarding para receber a resposta da sua requisição. A rede protegida pelo firewall, se não tiver regras de encaminhamento, não enxergará nada além da rede interna na qual faz parte. O encaminhamento de pacotes é necessário para a navegação e utilização de serviços externos de rede.

4.3. Squid Cache

O Proxy de Aplicativo Squid Cache é um software freeware.

Além de servir como cache, mantendo cópias dos sites acessados com maior frequência, serve também para permitir ou negar acesso a determinadas URL's (Uniform Resource Locator – Localizador de Recursos Uniformes) - para determinadas redes.

O Squid Cache usa por default a porta TCP 3128 para receber as requisições de acesso dos clientes e para repassar os dados por eles solicitados.

4.4. Instalações e atualizações necessárias

O computador utilizado para a implementação do trabalho foi equipado com duas placas de rede Ethernet 10/100 Mbps, para separar fisicamente a rede externa (Internet) das redes internas (Intranets).

A distribuição Linux utilizada foi o Slackware 7.1, devido ao fato de ser a distribuição mais utilizada e recomendada pelos profissionais da área. Originalmente com o Kernel versão 2.2.16, optou-se pela atualização para a versão 2.2.19, sendo a última atualização do Kernel 2.2 até a presente data, para fins de correções de falhas de segurança e aperfeiçoamentos do sistema.

Outras atualizações imprescindíveis para a correção de bugs de segurança foram necessárias:

- a) Bind (DNS): O Bind é o serviço utilizado para a resolução de nomes de hosts na rede pelo Linux. A versão original (8.2.2.P5) continha bugs de segurança e precisou ser atualizada para a versão 8.2.4.
- b) Wuftp: O Wuftp é o serviço de FTP do Linux. A versão original (2.6.0(1)) continha bugs de segurança e precisou ser atualizada para a versão 2.6.1(0).
- c) OpenSSL: O OpenSSL oferece bibliotecas para a utilização de SSL necessárias para a utilização de muitos outros serviços da máquina, dentre eles o OpenSSH. A versão original (0.9.3) continha bugs de segurança e precisou ser atualizada para a versão 0.9.6a.
- d) OpenSSH: O OpenSSH é um serviço de conexão remota segura. Funciona como o Telnet, porém utiliza criptografia no tráfego de dados. A versão original (2.3.0) continha bugs de segurança e precisou ser atualizada para a versão 2.9p2.

O software Squid Cache foi instalado na versão 2.4.STABLE2.

O software Ipchains, como visto anteriormente, é nativo do sistema e não precisou atualizações.

4.5. Configurações

Para o correto funcionamento do firewall, foram realizadas configurações no Kernel, nas interfaces de rede, no Squid Cache e no Ipchains. Abaixo estão descritas estas configurações necessárias e seus propósitos.

4.5.1. Kernel

Habilitou-se as opções “Network Firewalls” e “IP: firewalling”, no módulo “Networking options” do menu principal de configuração do Kernel, para dar suporte à utilização do Ipchains.
[DAN 00]

4.5.2. Interfaces de Rede

Foram utilizadas duas interfaces físicas: a eth0 (Interface Externa – Acesso com a Internet) e a eth1 (Interface Interna – Acesso à intranet local).

A interface eth0 recebeu um IP válido na Internet, de número 200.228.193.171, máscara de sub-rede 255.255.255.192.

A interface eth1 recebeu um IP de rede interna (não válido para a Internet e reservado para estes fins), de número 192.168.0.1, máscara de sub-rede 255.255.255.0.

Também foram definidas regras para o roteamento dos pacotes que trafegaram na rede. O destino das solicitações de acesso externo passaram obrigatoriamente pela eth0 que tiveram o seu destino definido para o roteador 200.228.193.133. Este equipamento é a última instância entre o firewall e a Internet propriamente dita. O destino dos pacotes da intranet foram encaminhados para eth1.

4.5.3. Squid Cache

O Squid Cache foi instalado com o propósito de oferecer cache e bloquear URL's indesejadas. O arquivo de configuração do Squid Cache é o "squid.conf", que contém as configurações padrões. Porém algumas mudanças foram necessárias para adequar este serviço à esta implementação:

```
acl palavras url_regex "/usr/local/squid/etc/palavras"
```

A linha acima declara uma "acl - access control list" (Lista de controle de acessos) com o nome "palavras" do tipo "url_regex" (URL's a serem rejeitadas/bloqueadas) que aponta para o arquivo "/usr/local/squid/etc/palavras" que tem a lista das URL's a serem bloqueadas.

```
acl INTERNA src 192.168.0.0/255.255.255.0
```

```
acl SERVERS src 200.228.193.128/255.255.255.192
```

Aqui estão sendo declaradas duas acl's que definem as origens (src) das solicitações de acesso. A "INTERNA" com o endereço da rede interna e a "SERVERS" com o endereço da rede válida na Internet. As duas foram definidas para atender as máquinas com IP's de rede interna e também outras máquinas que por algum motivo precisaram ter IP real e utilizaram o serviço de Proxy.

```
http_access allow INTERNA SERVERS localhost
```

```
http_access deny palavras
```

```
http_access allow all
```

A tag "http_access" permite ou nega os acessos definidos pelas acl's. As linhas acima habilitam (allow) o tráfego para a rede interna (INTERNA), para as máquinas servidoras ou com IP's reais (SERVERS) e para o host local (localhost que é o próprio firewall). Em seguida, nega-se (deny) a acl palavras (que foi definida acima para bloquear URL's indesejadas) e, por fim, permite-se o acesso (allow) a todos os outros sites que não se encaixaram na acl "palavras".

```
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 80 -j REDIRECT 3128 -l
```

Esta última linha não faz parte do arquivo de configuração do Squid Cache. É a linha de configuração do ipchains que redireciona todo o tráfego requisitado pela rede interna ao Squid Cache, integrando assim um serviço de filtragem de pacotes (ipchains) com um proxy de aplicativo (Squid Cache).

4.5.4. Ipchains

O ipchains é nativo no Sistema Operacional Linux. Ele foi utilizado para definir todas as regras de filtragem de pacotes. Como as regras definidas não ficam salvas automaticamente no kernel do sistema houve a necessidade de criar um script que redefina as regras todas as vezes que a máquina firewall for reiniciada. O Script será detalhado logo após a sintaxe *ipchains*:

-A (append) – adicionar uma regra para a ipchain informada (input, output ou forward)

input – solicitações de serviço e/ou recurso na máquina

output – acessos da máquina para fora

forward – compartilhamento/encaminhamento de pacotes

-s (source) – determina a origem para uma regra. Pode ser um IP, uma classe de IPs ou tudo (any/0). Também pode ser precedida de uma porta. Exemplo: -s any/0 80. Qualquer ip de origem, porém da porta 80

-d (destination) – define o destino para um determinado pacote. Pode ser any/0 e também pode ser precedido de uma porta.

-p – informa o tipo de protocolo. Pode ser tcp, udp ou icmp. Se omitido, significa todos (all ou 0).

-j (jump) – determina o que deve ser feito com a regra: DENY (negar), REJECT (rejeitar), ACCEPT (aceitar), REDIRECT (redirecionar) ou MASQ (compartilhar/encaminhar). A diferença entre DENY e REJECT é que REJECT nega e envia uma resposta informando a negação do acesso e DENY apenas nega sem informar nada ao solicitante.

-l – logar as ações que se encaixem com a regra.

4.5.4.1. Início do Script:

A linha inicial do script rc.firewall (assim denominado para seguir o padrão dos nomes de arquivos de scripts de inicialização do Linux – Por exemplo, rc.inet1, rc.netdevice, etc.) habilita o encaminhamento (forwarding) de pacotes.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Em seguida, o comando *ipchains -F* remove todas as regras existentes definidas. A instrução *echo* apenas imprime na tela uma mensagem para informar que o serviço de firewalling está sendo iniciado. E nas três linhas seguintes têm-se as definições da política padrão de trabalho do firewall: negar tudo e permitir apenas o que for definido.

```

ipchains -F
echo "Iniciando o servico de Firewalling..."
# define o padrao para deny
ipchains -P input DENY
ipchains -P output REJECT
ipchains -P forward REJECT

```

A declaração de constantes facilita o trabalho de manutenção do firewall, uma vez que, o valor de uma constante pode ser utilizado muitas vezes dentro do mesmo script, haveria muito trabalho para modificar todas as linhas com esse determinado valor. Assim sendo, quando houverem modificações a serem feitas, altera-se apenas o valor na declaração das constantes.

Em muitas linhas do *ipchains* foi utilizada a constante `PORTAS_ALTAS`. O motivo deve-se ao fato dos clientes utilizarem portas altas (de 1024 a 65535) nas suas requisições.

```

# declaracoes de constantes
REDE_INTERNA="192.168.0.0/24"
REDE_EXTERNA="200.228.193.128/26"
INTERFACE_EXTERNA="eth0"
INTERFACE_INTERNA="eth1"
INTERFACE_LOOPBACK="lo"
IP_INTERFACE_INTERNA="192.168.0.1"
IP_INTERFACE_EXTERNA="10.0.11.6"
IPADDRESS="200.228.193.171"
QUALQUER_O_D="any/0"
CLASSE_A="10.0.0.0/8"
CLASSE_B="172.16.0.0/12"
CLASSE_C="192.168.0.0/24"
CLASSE_D_MULTICAST="224.0.0.0/4"
CLASSE_E_RESERVED_NET="240.0.0.0/5"
ORIGEM_BROADCAST="0.0.0.0"
DESTINO_BROADCAST="255.255.255.255"
PORTAS_BAIXAS="0:1023"
PORTAS_ALTAS="1024:65535"
LOOPBACK="127.0.0.0/8"

```

```
SMTP_SERVER="firewall.uri.com.br"
DNS1="200.228.193.171"
DNS2="200.228.193.140"
SYSLOG_SERVER="200.228.193.171"
SYSLOG_CLIENT="10.0.0.0/8"
PORTAS_SSH="1022:1023"
PORTAS_ORI_TRACEROUTE="32769:65535"
PORTAS_DEST_TRACEROUTE="33434:33523"
```

Nega pacotes forjados como se fossem do IP externo da máquina.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -s ${IPADDRESS} -j DENY -l
```

Ativa o “masquerading”, ou seja, o compartilhamento/encaminhamento de pacotes com a rede interna.

```
ipchains -A forward -s ${REDE_INTERNA} -j MASQ
```

Permite o tráfego ilimitado na interface de loopback (definida como “lo”). Essa interface não existe fisicamente. É apenas uma definição interna usada pelo sistema operacional para executar todos os programas e processos de rede. Se este tráfego for negado, nenhum aplicativo de rede (por exemplo, o inetd) conseguirá ser executado.

```
ipchains -A input -i ${INTERFACE_LOOPBACK} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_LOOPBACK} -j ACCEPT
```

As linhas do script acima negam o tráfego que chega à interface externa (eth0) “dizendo” ser de uma classe A, B ou C. A interface externa pode aceitar apenas pacotes de endereços IP’s válidos. Pacotes de redes classe A, B ou C na interface eth0 teriam sido modificados para tentativa de um provável ataque.

```
# nega pacotes de uma Classe A
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -s ${CLASSE_A} -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -d ${CLASSE_A} -j DENY -l
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -s ${CLASSE_A} -j REJECT -l
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -d ${CLASSE_A} -j REJECT -l
```

```
# nega pacotes de uma Classe B
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -s ${CLASSE_B} -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -d ${CLASSE_B} -j DENY -l
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -s ${CLASSE_B} -j REJECT -l
```

```

ipchains -A output -i ${INTERFACE_EXTERNA} -d ${CLASSE_B} -j REJECT -l
# nega pacotes de uma Classe C
ipchains -A input -i ${INTERFACE_EXTERNA} -s ${CLASSE_C} -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -d ${CLASSE_C} -j DENY -l
ipchains -A output -i ${INTERFACE_EXTERNA} -s ${CLASSE_C} -j REJECT -l
# linha abaixo para aceitar retorno do proxy squid cache.
ipchains -A output -i ${INTERFACE_EXTERNA} -d ${CLASSE_C} -j ACCEPT -l

```

Negar pacotes de uma falsa interface de loopback.

```

ipchains -A input -i ${INTERFACE_EXTERNA} -s ${LOOPBACK} -j DENY -l
ipchains -A output -i ${INTERFACE_EXTERNA} -s ${LOOPBACK} -j REJECT -l

```

Negar pacotes de endereços reservados (classe E) ou não permitidos (classe D)

```

ipchains -A input -i ${INTERFACE_EXTERNA} -s ${CLASSE_D_MULTICAST} -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s ${CLASSE_E_RESERVED_NET} -j DENY -l

```

Negar pacotes de redes não permitidas e/ou reservadas.

```

ipchains -A input -i ${INTERFACE_EXTERNA} -s 1.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 2.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 5.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 7.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 23.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 27.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 31.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 37.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 39.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 41.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 42.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 58.0.0.0/7 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 60.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 65.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 66.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -s 67.0.0.0/8 -j DENY -l

```

```
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 68.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 69.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 70.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 71.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 72.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 73.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 74.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 75.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 76.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 77.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 78.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 79.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 80.0.0.0/4 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 96.0.0.0/4 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 112.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 113.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 114.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 115.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 116.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 117.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 118.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 119.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 120.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 121.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 122.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 123.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 124.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 125.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 126.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 217.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 218.0.0.0/8 -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNALA} -s 219.0.0.0/8 -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -s 220.0.0.0/6 -j DENY -l
```

Evita ataques de DoS ou DDoS.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 0 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 3 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 4 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 11 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 12 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${REDE_EXTERNA} 8 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

O ICMP é um protocolo que pode ser utilizado para ataques de negação de serviço DoS ou ataques de DDoS (Distributed Denial of Service - Negação de Serviço Distribuído). As linhas acima permitem apenas o tráfego aceitável de ICMP até certo alcance. Barrá-lo por completo prejudicaria a negociação do tamanho de fragmentação de um pacote. O alcance definido aqui é sempre, no máximo a rede externa.

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 0 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
    -s ${QUALQUER_O_D} 3 \
    -d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
-s ${QUALQUER_O_D} 4 \
-d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
-s ${QUALQUER_O_D} 11 \
-d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
-s ${QUALQUER_O_D} 12 \
-d ${REDE_EXTERNA} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p icmp \
-s ${REDE_EXTERNA} 8 \
-d ${REDE_EXTERNA} -j ACCEPT
```

Permitir apenas *traceroute* (*tracert* nos sistemas Microsoft) apenas para as máquinas pertencentes à “rede externa”. Rede externa neste script significa todas as máquinas da rede que utilizam um IP real, ou seja, um IP válido na Internet e não as máquinas que estão fora da rede interna.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
-s ${REDE_EXTERNA} ${PORTAS_ORI_TRACERROUTE} \
-d ${IPADDRESS} ${PORTAS_DEST_TRACERROUTE} -j ACCEPT -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
-s ${QUALQUER_O_D} ${PORTAS_ORI_TRACERROUTE} \
-d ${IPADDRESS} ${PORTAS_DEST_TRACERROUTE} -j DENY -l
```

Aceitar as solicitações de consultas DNS no servidor DNS1 e DNS2 feitas pela rede interna e também permitir o sincronismo de “zonas de DNS” entre o servidor DNS1 e o servidor DNS2. O serviço DNS usa a porta UDP e TCP 53.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
-s ${DNS1} 53 \
-d ${IPADDRESS} 53 -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p udp \
-s ${IPADDRESS} 53 \
```

```

        -d ${DNS1} 53 -j ACCEPT
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
        -s ${DNS2} 53 \
        -d ${IPADDRESS} 53 -j ACCEPT

ipchains -A output -i ${INTERFACE_EXTERNA} -p udp \
        -s ${IPADDRESS} 53 \
        -d ${DNS2} 53 -j ACCEPT

# clientes DNS
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
        -s ${DNS1} 53 \
        -d ${IPADDRESS} ${PORTAS_ALTAS} -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p udp \
        -s ${IPADDRESS} ${PORTAS_ALTAS} \
        -d ${DNS1} 53 -j ACCEPT
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp ! -y \
        -s ${DNS1} 53 \
        -d ${IPADDRESS} ${PORTAS_ALTAS} -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp \
        -s ${IPADDRESS} ${PORTAS_ALTAS} \
        -d ${DNS1} 53 -j ACCEPT
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
        -s ${DNS2} 53 \
        -d ${IPADDRESS} ${PORTAS_ALTAS} -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p udp \
        -s ${IPADDRESS} ${PORTAS_ALTAS} \
        -d ${DNS2} 53 -j ACCEPT
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp ! -y \
        -s ${DNS2} 53 \
        -d ${IPADDRESS} ${PORTAS_ALTAS} -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp \
        -s ${IPADDRESS} ${PORTAS_ALTAS} \

```

```

        -d ${DNS2} 53 -j ACCEPT

# clientes DNS internos
ipchains -A input -i ${INTERFACE_INTERNA} -p udp \
        -s ${REDE_INTERNA} \
        -d ${IP_INTERFACE_INTERNA} -j ACCEPT
ipchains -A output -i ${INTERFACE_INTERNA} -p udp \
        -s ${QUALQUER_O_D} \
        -d ${REDE_INTERNA} -j ACCEPT

```

Aceitar tráfego para os serviços de SSH.

```

ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \
        -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
        -d ${IPADDRESS} 22 -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp ! -y \
        -s ${IPADDRESS} 22 \
        -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \
        -s ${QUALQUER_O_D} ${PORTAS_SSH} \
        -d ${IPADDRESS} 22 -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp ! -y \
        -s ${IPADDRESS} 22 \
        -d ${QUALQUER_O_D} ${PORTAS_SSH} -j ACCEPT

```

Permitir o acesso ao servidor apache nas portas 80 (default para HTTP) e 443 (HTTPS (Hyper Text Transfer Protocol Secure – Protocolo Seguro de Transferência de Hiper Texto)).

```

# controle do apache server (porta 80 - http)
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \
        -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
        -d ${IPADDRESS} 80 -j ACCEPT
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp ! -y \
        -s ${IPADDRESS} 80 \
        -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT
# controle do apache server (porta 443 - https)
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \

```

```
-s ${QUALQUER_O_D} ${PORTAS_ALTAS} \  
-d ${IPADDRESS} 443 -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp ! -y \  
-s ${IPADDRESS} 443 \  
-d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT
```

Permitir o tráfego entre o Squid Cache e as estações de trabalho e vice-versa.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \  
-s ${QUALQUER_O_D} 80 \  
-d ${IPADDRESS} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp \  
-s ${QUALQUER_O_D} \  
-d ${QUALQUER_O_D} -j ACCEPT
```

O *syslogd* roda na porta UDP 514 e é necessário para logar todas as ações do sistema. Aqui, permite-se também que uma outra máquina (SYSLOG_SERVER) fique com uma cópia dos logs gerados pela máquina firewall.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \  
-s ${SYSLOG_CLIENT} \  
-d ${IPADDRESS} 514 -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p udp \  
-s ${IPADDRESS} 514 \  
-d ${SYSLOG_SERVER} 514 -j ACCEPT
```

Permite o tráfego do serviço de E-mail (SMTP).

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp ! -y \  
-s ${SMTP_SERVER} 25 \  
-d ${IPADDRESS} ${PORTAS_ALTAS} -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp \  
-s ${IPADDRESS} ${PORTAS_ALTAS} \  
-d ${SMTP_SERVER} 25 -j ACCEPT
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp ! -y \  
-s ${REDE_INTERNA} ${PORTAS_ALTAS}\  
-d ${IPADDRESS} 110 -j ACCEPT
```

```
ipchains -A output -i ${INTERFACE_EXTERNA} -p tcp \
    -s ${IPADDRESS} 110 \
    -d ${PORTAS_ALTAS} ${PORTAS_ALTAS} -j ACCEPT
```

O FTP usa as portas TCP 20 e 21. A porta 21 é usada para estabelecer a conexão e autenticar os usuários. A porta 20 (conhecida como *ftp-data*) é a porta usada para as transferências de arquivos.

```
ipchains -A input -p tcp \
    -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
    -d ${QUALQUER_O_D} 21 -j ACCEPT -l
```

```
ipchains -A output -p tcp ! -y \
    -s ${QUALQUER_O_D} 21 \
    -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT -l
```

```
ipchains -A input -p tcp ! -y \
    -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
    -d ${QUALQUER_O_D} 20 -j ACCEPT -l
```

```
ipchains -A output -p tcp \
    -s ${QUALQUER_O_D} 20 \
    -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT -l
```

```
ipchains -A input -p tcp \
    -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
    -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT -l
```

```
ipchains -A output -p tcp ! -y \
    -s ${QUALQUER_O_D} ${PORTAS_ALTAS} \
    -d ${QUALQUER_O_D} ${PORTAS_ALTAS} -j ACCEPT -l
```

Ativar o serviço de log para os pacotes que serão barrados.

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p tcp \
    -d ${IPADDRESS} -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
    -d ${IPADDRESS} ${PORTAS_BAIIXAS} -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p udp \
    -d ${IPADDRESS} ${PORTAS_ALTAS} -j DENY -l
```

```
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
```

```

-s ${QUALQUER_O_D} 5 -d ${IPADDRESS} -j DENY -l
ipchains -A input -i ${INTERFACE_EXTERNA} -p icmp \
-s ${QUALQUER_O_D} 13:255 -d ${IPADDRESS} -j DENY -l

```

Esta regra define o proxy transparente para HTTP. Todas as requisições de acesso a sites (comumente porta 80) serão redirecionadas para o Squid Cache que avaliará se a URL é permitida ou não.

```
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 80 -j REDIRECT 3128 -l
```

Regras para negar os serviços de IRC, imesh (software utilizado para procurar vídeos e MP3 na Internet que usa a porta tcp 5000) e napster (utilizado para procura de MP3).

```

# negar irc
ipchains -A input -p tcp -s ${REDE_INTERNA} -d 0/0 6667 -j DENY

# negar imesh
ipchains -A input -p tcp -s ${REDE_INTERNA} -d 0/0 5000 -j DENY

# negar napster
ipchains -A input -p tcp -s ${REDE_INTERNA} -d 0/0 8888 -j DENY

```

Regras usadas para impedir o ataque por ferramentas como trojans. Por exemplo, WinNuke, WinCrash, etc.

```

ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 59 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 170 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 171 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 666 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 667 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 668 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 1243 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2000 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2115 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2583 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 5742 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 12345 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 20034 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 21544 -j DENY

```

```
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 27374 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 30100 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 30303 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 50505 -j DENY
ipchains -A input -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 54320 -j DENY
ipchains -A input -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 31337 -j DENY
ipchains -A input -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 54321 -j DENY
ipchains -A input -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2140 -j DENY
ipchains -A input -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 31789 -j DENY
ipchains -A input -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 10167 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 59 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 170 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 171 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 666 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 667 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 668 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 1243 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2000 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2115 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2583 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 5742 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 12345 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 20034 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 21544 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 27374 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 30100 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 30303 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 50505 -j DENY
ipchains -A output -p tcp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 54320 -j DENY
ipchains -A output -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 31337 -j DENY
ipchains -A output -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 54321 -j DENY
ipchains -A output -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 2140 -j DENY
```

```
ipchains -A output -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 31789 -j DENY  
ipchains -A output -p udp -s ${REDE_INTERNA} -d ${QUALQUER_O_D} 10167 -j DENY
```

5. Análise de Resultados

Ao analisar o arquivo de log do firewall, percebeu-se que os tráfegos de dados necessários para a navegação foram permitidas, e tentativas de varreduras de portas, requisições de ping e outros ataques. Abaixo estão comentadas algumas linhas deste arquivo.

As conexões FTP utilizam as portas 20 e 21 do servidor e portas altas (acima de 1024) no cliente. As linhas abaixo mostram uma conexão estabelecida e aceita entre o servidor (IP 192.168.0.1) e o cliente (IP 192.168.0.2). Foram aceitos pacotes da chain INPUT e da chain OUTPUT.

```
Jul  6 22:15:33 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=48 S=0x00 I=25857 F=0x4000 T=128 SYN (#82)
Jul  6 22:15:33 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=48 S=0x00 I=1623 F=0x4000 T=64 (#25)
Jul  6 22:15:33 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=40 S=0x00 I=26113 F=0x4000 T=128 (#82)
Jul  6 22:15:35 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=133 S=0x10 I=1632 F=0x0000 T=64 (#25)
Jul  6 22:15:35 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=40 S=0x00 I=26369 F=0x4000 T=128 (#82)
Jul  6 22:15:39 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=53 S=0x00 I=26625 F=0x4000 T=128 (#82)
Jul  6 22:15:39 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=40 S=0x10 I=1633 F=0x0000 T=64 (#25)
Jul  6 22:15:39 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=75 S=0x10 I=1634 F=0x0000 T=64 (#25)
Jul  6 22:15:39 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=40 S=0x00 I=26881 F=0x4000 T=128 (#82)
Jul  6 22:15:40 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=55 S=0x00 I=27137 F=0x4000 T=128 (#82)
Jul  6 22:15:40 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=68 S=0x10 I=1635 F=0x0000 T=64 (#25)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=40 S=0x00 I=27393 F=0x4000 T=128 (#82)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=63 S=0x00 I=27649 F=0x4000 T=128 (#82)
```

```

Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=70 S=0x10 I=1636 F=0x0000 T=64 (#25)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=46 S=0x00 I=27905 F=0x4000 T=128 (#82)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=40 S=0x10 I=1637 F=0x0000 T=64 (#25)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:20 192.168.0.2:1056 L=60 S=0x08 I=1638 F=0x0000 T=64 SYN (#26)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1056 192.168.0.1:20 L=48 S=0x00 I=28161 F=0x4000 T=128 (#83)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:20 192.168.0.2:1056 L=40 S=0x08 I=1639 F=0x0000 T=64 (#26)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=93 S=0x10 I=1640 F=0x0000 T=64 (#25)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:20 192.168.0.2:1056 L=894 S=0x08 I=1641 F=0x0000 T=64 (#26)
Jul  6 22:15:41 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:20 192.168.0.2:1056 L=40 S=0x08 I=1642 F=0x0000 T=64 (#26)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1056 192.168.0.1:20 L=40 S=0x00 I=28417 F=0x4000 T=128 (#83)
Jul  6 22:15:41 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1055 192.168.0.1:21 L=40 S=0x00 I=28673 F=0x4000 T=128 (#82)
Jul  6 22:15:42 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1056 192.168.0.1:20 L=40 S=0x00 I=28929 F=0x4000 T=128 (#83)
Jul  6 22:15:42 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:20 192.168.0.2:1056 L=40 S=0x08 I=1643 F=0x0000 T=64 (#26)
Jul  6 22:15:42 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:21 192.168.0.2:1055 L=64 S=0x10 I=1644 F=0x0000 T=64 (#25)

```

Tráfego aceito entre a máquina cliente e a máquina servidor para conexões via Squid Cache. O Squid Cache usa a porta TCP 3128 e o cliente usa uma porta alta.

```

Jul  6 22:29:39 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1061 192.168.0.1:3128 L=48 S=0x00 I=12034 F=0x4000 T=128 SYN
(#83)
Jul  6 22:29:39 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:3128 192.168.0.2:1061 L=48 S=0x00 I=2099 F=0x4000 T=64 (#27)

```

```
Jul  6 22:29:39 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1061 192.168.0.1:3128 L=40 S=0x00 I=12546 F=0x4000 T=128 (#83)
Jul  6 22:29:39 firewall kernel: Packet log: input ACCEPT eth1 PROTO=6
192.168.0.2:1061 192.168.0.1:3128 L=311 S=0x00 I=12802 F=0x4000 T=128 (#83)
Jul  6 22:29:39 firewall kernel: Packet log: output ACCEPT eth1 PROTO=6
192.168.0.1:3128 192.168.0.2:1061 L=40 S=0x00 I=2100 F=0x0000 T=64 (#27)
```

Negação de mensagem ICMP tipo 5 (solicitação de redirecionamento de rota) solicitada pela máquina 200.198.170.28.

```
Jul  7 08:56:25 firewall kernel: Packet log: input DENY eth0 PROTO=1
200.198.170.28:5 200.228.193.171:1 L=88 S=0xC0 I=46840 F=0x0000 T=255 (#87)
Jul  7 08:56:26 firewall kernel: Packet log: input DENY eth0 PROTO=1
200.198.170.28:5 200.228.193.171:1 L=88 S=0xC0 I=46848 F=0x0000 T=255 (#87)
Jul  7 08:56:27 firewall kernel: Packet log: input DENY eth0 PROTO=1
200.198.170.28:5 200.228.193.171:1 L=88 S=0xC0 I=46856 F=0x0000 T=255 (#87)
Jul  7 08:56:28 firewall kernel: Packet log: input DENY eth0 PROTO=1
200.198.170.28:5 200.228.193.171:1 L=88 S=0xC0 I=46864 F=0x0000 T=255 (#87)
```

Conclusão

O objetivo do trabalho foi atingido, visto que se tratava da configuração e implementação de um Firewall, para a proteção de uma rede interna da Internet.

Para uma correta avaliação do funcionamento do Firewall, tornou-se necessário um estudo aprofundado do conjunto TCP/IP, base para o funcionamento da Internet e do estudo de ferramentas utilizadas pelos atacantes, a fim de verificar-se os principais pontos a serem protegidos.

Considerando uma ampla oferta de Sistemas Operacionais para servidores existentes no mercado, optou-se pelo Linux, não só pelo fato de ser *freeware*, mas também pelo fato de ser um dos mais seguros e utilizados neste quesito.

Utilizaram-se, então, os softwares Ipchains, que é um software de filtragem de pacotes, e o Squid Cache, que é um software proxy de aplicativo, para fins de proteção da rede interna à Internet, e pelo controle do conteúdo acessado pelos usuários da rede interna.

Realizando-se testes de segurança, após a completa configuração e implementação do Firewall, constatou-se que a rede interna estava protegida, podendo-se navegar na Internet, enviar e receber emails, fazer acessos FTP, ou seja, utilizar todas as vantagens da Internet tendo um Firewall realizando seu trabalho transparentemente ao usuário.

Referências Bibliográficas

- [ARN 96] ARNETT, Matthew Flint. **Desvendando o TCP/IP**. Rio de Janeiro: Campus, 1996, 543p.
- [AUT 00] Autor Anônimo. **Segurança Máxima: O guia para proteger seu site na Internet e sua rede**. Rio de Janeiro: Campus, 2000. 823p. 2nd ed.
- [COM 98] COMER, Douglas E.. **Interligação em rede com TCP/IP: Princípios, protocolos e arquitetura**. Rio de Janeiro: Campus, 1998. 672p.
- [DAN 00] DANESH, Arman. **Dominando o Linux**. São Paulo: Makron Books, 2000. 574p.
- [MCC 00] McCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. **Hackers Expostos: segredos e soluções para a segurança de redes**. São Paulo: Makron Books, 2000. 469p.
- [ZWI 00] ZWICKY, Elizabeth D.; COOPER, Simon; CHAPMAN, D. Brent. **Construindo Firewalls para a Internet**. Rio de Janeiro: Campus, 2000. 889p. 2nd ed.
- NET-3-HOWTO. Disponível por WWW em <http://www.poli.org/LDP-PT/HOWTO/NET-3-HOWTO/NET-3-HOWTO.html>
- RFC 793. Disponível por WWW em <http://rfc.net/rfc793.html>
- RFC 1180. Disponível por WWW em <http://rfc.net/rfc1180.html>
- HOWTO de Firewall e Servidor Proxy. Disponível por WWW em <http://ano2001.sti.com.br/howt-2.html>