

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

PAULO ROBERTO XAVIER MACHADO

FIREWALL EM REDES IPV6

**NATAL/RN
2012**

PAULO ROBERTO XAVIER MACHADO

FIREWALL EM REDES IPV6

Monografia apresentada ao curso de Especialização em Redes de Computadores do Centro Universitário do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Redes de Computadores.

Orientador:

Aluizio Ferreira da Rocha Neto, M.Sc.

PAULO ROBERTO XAVIER MACHADO

FIREWALL EM REDES IPV6

Monografia apresentada e aprovada em 25 de 06 de 2012, pela banca examinadora composta pelos seguintes membros:

Aluizio Ferreira da Rocha Neto, M.Sc.
Orientador

Natal/RN
2012

Dedico esse trabalho para dois seres especiais na minha vida, pois sem eles não teria chegado até aqui, que é meu Deus, meu porto seguro e rocha firme para minha fé; e minha querida mãe, a qual me concebeu como um presente do Senhor e cuida de mim, desde do meu nascimento até hoje, com zelo e amor.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, que iluminou meu caminho. Agradeço também aos meus pais, irmãos, minha esposa Amanda, minha sogra e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Ao professor Aluizio Rocha pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia e ao professor Ricardo Kleber, pela dica para definir meu tema de trabalho.

A todos os professores do curso, que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia. Aos amigos e colegas, pelo incentivo e pelo apoio constantes.

RESUMO

Projetadas inicialmente para uso acadêmico, as redes de computadores, se utilizando do protocolo IPv4, não receberam políticas de uso ideais dos endereços IP. Essa distribuição inadequada dos blocos de IP aliada ao início da utilização comercial da Internet resultou na rápida escassez de endereços. Medidas paliativas foram adotadas para adiar o inevitável esgotamento do IP enquanto que, simultaneamente, se desenvolvia o protocolo IP na versão 6, o IPv6. O IPv6, além de ser mais simples que o IPv4, promete resolver definitivamente o problema da distribuição de endereços na Internet. Diante deste cenário de mudança, os administradores de redes devem estar preparados. Uma das principais preocupações é com o Firewall. O Firewall não é a única ferramenta de segurança utilizada por administradores de redes, mas seguramente é a principal delas. A garantia da integridade e confiabilidade dos dados de uma empresa passa principalmente por essa ferramenta que, se mal configurada, pode comprometer toda a política de segurança da empresa. Neste trabalho serão abordadas as políticas que devem ser adotadas para garantir a segurança em redes de computadores que utilizam o protocolo IPv6. Tais políticas serão apresentadas através de breves comparativos com o, até então, atual protocolo padrão (IPv4) e a melhor maneira de se configurar um Firewall a fim de evitar falhas críticas de segurança na rede. Para isso será apresentado um cenário fictício para exemplificar a aplicação de um Firewall seguro na rede IPv6.

Palavras-Chave: IPv4. IPv6. Firewall.

ABSTRACT

Designed primarily for academic use, computer networks, using the IPv4 protocol, did not receive political ideals of use of IP addresses. This maldistribution of IP blocks together with the beginning of the commercial Internet has resulted in rapid shortage of addresses. Mitigation measures were adopted to delay the inevitable exhaustion of IP while a new version of IP protocol was developed, the version 6, or IPv6. IPv6 in addition to being simpler than the IPv4 promises to permanently solve the problem of distribution of Internet addresses. Against this backdrop of change, network administrators must be prepared. A major concern is with the Firewall. The Firewall is not the only security tool used by network administrators, but surely is the main one. Ensuring data integrity and reliability of a company is mainly for this tool which, if misconfigured, can compromise the entire security policy of a company. This paper will discuss the policies that should be taken to ensure security in computer networks using the IPv6 protocol. Such policies will be presented through brief comparison with the hitherto current standard protocol (IPv4) and the best way to configure a firewall to prevent failures in safety-critical network. To this, will be presented a fictitious scenario to illustrate the application of a secure firewall in IPv6 network.

Keywords: IPv4. IPv6. Firewall.

LISTA DE FIGURAS

Figura 1: Datagrama IPv4.....	14
Figura 2: Classes de endereços IPv4.	17
Figura 3: Processo de Tradução de Endereços (NAT).	19
Figura 4: Protocolo DHCP.	20
Figura 5: Requisição DHCP.	21
Figura 6: Datagrama IPv6.....	23
Figura 7: Cabeçalho IPv4 x IPv6.....	25
Figura 8: Cabeçalho IPv6.	25
Figura 9: Firewall Básico.	28
Figura 10: Castelo Medieval.....	29
Figura 11: Tabela <i>Filter</i>	31
Figura 12: Tabela <i>Nat</i>	31
Figura 13: Rede do Estudo de Caso.....	36

LISTA DE TABELAS

Tabela 1: Opções do cabeçalho IPv4.....	16
Tabela 2: Máscara das classes de IP.....	17
Tabela 3: Endereços IPv4 usados em Redes Privadas.....	19
Tabela 4: Endereços IPv6 Reservados.....	26
Tabela 5: Script do Firewall Ip6tables.....	37

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS.....	12
1.1.1	Objetivos Específicos.....	12
1.2	ORGANIZAÇÃO DO TRABALHO	12
2	REFERENCIAL TEÓRICO	13
2.1	SEGURANÇA DA INFORMAÇÃO	13
2.2	PROTOCOLO IPV4.....	13
2.3	SOLUÇÕES PALIATIVAS PARA O FIM DO IPV4	18
2.3.1	<i>Classless InterDomain Routing</i> (CIDR) [RFC 1519]	18
2.3.2	Endereços privados	18
2.3.3	<i>Network Address Translation</i> (NAT).....	19
2.3.4	<i>Dynamic Host Configuration Protocol</i> (DHCP).....	20
2.4	PROTOCOLO IPV6.....	22
2.4.1	Características	22
2.4.2	Datagrama	23
2.4.3	Cabeçalho base.....	23
2.4.4	Faixas de endereços reservados	26
2.4.5	Protocolo de Controle ICMPv6.....	26
2.5	FIREWALL	27
2.5.1	Requisito	27
2.5.2	Funcionamento	28
2.5.3	História	28
2.6	IPTABLES	30
2.6.1	Tabelas e Comandos	30
2.6.2	A ação ser tomada (<i>Target</i>).....	32
2.6.3	NAT e Filtragem de pacotes	32
2.7	IP6TABLES	33
2.7.1	Tabelas e Comandos	33
3	ESTUDO DE CASO	35
3.1	CARACTERÍSTICAS DA REDE ADOTADA	35
3.2	PARÂMETROS	36
4	CONCLUSÃO	43
	REFERÊNCIAS.....	44

1 INTRODUÇÃO

O crescimento exponencial que a Internet alcançou desde a sua concepção, aliada a má alocação de endereços promovida no seu surgimento, ocasionou o problema de escassez de endereços IPv4, problema esse que não pode ser contornado de maneira simples. A alternativa mais viável encontrada foi aumentar o tamanho dos endereços IP dos atuais 32 bits que são utilizados na notação decimal (Aproximadamente 4 bilhões de endereços) para 128 bits com notação hexadecimal (cerca de 79 octilhões de endereços disponíveis).

A partir do início da gradativa migração do IPv4 para o IPv6, as políticas de segurança devem ser revisadas pelos administradores de redes. A simples migração do IPv4 para o IPv6 resulta na mudança do padrão de rede adotado na maioria das empresas atualmente, ao invés de utilizar um roteador que mascara os endereços locais para o seu endereço válido através de NAT. Todas as estações de uma rede interna podem receber seu próprio IP válido onde tais mudanças alteram sensivelmente a segurança das redes locais, uma vez que aumenta a quantidade de endereços que podem estar sujeitos a alguma falha de segurança além de mais endereços estarem disponíveis para sofrer ataques do tipo DDoS, Buffer Overflow, DNS Poisoning.

Segundo o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br, 2006) o *Distributed Denial of Service (DDoS)*, “constitui um ataque de negação de serviço distribuído, ou seja, um conjunto de computadores é utilizado para tirar de operação um ou mais serviços ou computadores conectados à Internet”.

Já a Wikipédia (2012) diz que “em segurança computacional e programação um transbordamento de dados ou estouro de buffer (do inglês *buffer overflow* ou *buffer overrun*) é uma anomalia onde um programa, ao escrever dados em um buffer, ultrapassa os limites do buffer e sobrescreve a memória adjacente”.

Na Wikipédia (2012a) diz que envenenamento de cache DNS (*DNS cache poisoning*) é o comprometimento na segurança ou na integridade dos dados em um Sistema de Nomes de Domínios (*Domain Name System - DNS*). Esse problema acontece quando os dados que são introduzidos na cache de um servidor de nomes DNS não se originam do servidor de nomes DNS com autoridade real. Tal problema pode ser uma tentativa de ataque malicioso em um servidor de nomes, mas também pode ser o resultado de um erro não intencional de configuração na cache do servidor DNS.

Visando garantir maior segurança nos firewalls utilizados nas redes IPv6, este trabalho

fará diversas considerações sobre como deve ser preparado um Firewall e quais os novos desafios a serem encarados pelos administradores de redes a partir da adoção do novo protocolo IPv6.

1.1 OBJETIVOS

Este projeto tem como objetivo fazer uma análise e apontar sugestões para a utilização de um Firewall em redes IPv6, uma vez comparados ao protocolo IPv4. Para exemplificar, será utilizado o Firewall Iptables/Ip6tables que foi desenvolvido pela Netfilter.

1.1.1 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, as seguintes etapas serão seguidas:

- Descrição do protocolo IPv4;
- Análise de Firewall em redes IPv4;
- Descrição do protocolo IPv6;
- Análise de Firewall em redes IPv6;
- Implementação virtual de Firewall em uma rede IPv6;

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em quatro capítulos. O Capítulo 1 apresenta a contextualização do trabalho, o problema e os objetivos da monografia. O Capítulo 2, contempla o referencial teórico de todas as tecnologias envolvidas neste trabalho, como o funcionamento de redes IPv4, suas vantagens e desvantagens, os conceitos básicos de Firewall aplicados a algumas topologias de rede e a motivação da mudança para a nova versão. O Capítulo 3 trata das vantagens do novo protocolo e quais os riscos da utilização do mesmo, assim como a utilização de Firewall direcionado ao novo protocolo. O Capítulo 4 apresenta as conclusões deste trabalho.

2 REFERENCIAL TEÓRICO

2.1 SEGURANÇA DA INFORMAÇÃO

A informação é essencial para os negócios de uma empresa e, por isso, deve ser protegida. Em um mundo cada vez mais conectado, isso acaba sendo mais importante até porque a quantidade de pragas e vulnerabilidades no mundo digital aumenta mais e mais.

Garantir a segurança da informação é garantir a proteção dela em relação às várias ameaças existentes atualmente e minimizar o risco que elas oferecem ao nosso negócio. Tais ameaças, considerando o meio digital, podem ser definidas como “revelação não autorizada da informação, modificações ou produções não autorizadas da informação e negação de serviço” (RIBEIRO, 2004).

Tal segurança é atingida quando da utilização de um conjunto de controles adequados para essa finalidade que vão desde a definição de políticas de segurança a funções de hardware e software. Muitos sistemas de informação não aplicam políticas específicas de controle, apesar de esses sistemas requererem procedimentos apropriados. Uma maneira de resguardar a segurança dos dados é seguindo 5 princípios básicos da informação, conhecidos como 5 pilares da segurança da informação, são eles:

- Confidencialidade;
- Integridade;
- Disponibilidade;
- Autenticidade; e
- Não repúdio

2.2 PROTOCOLO IPV4

O protocolo IP atualmente é o mais utilizado em redes locais. Isto se deve a popularização da Internet e ao fato do protocolo ter sido desenvolvido para ser utilizado na rede mundial de computadores.

Segundo Torres (2001), “Uma das grandes vantagens do TCP/IP em relação a outros protocolos existentes é que ele é roteável, isto é, foi criado pensando em redes grandes e de longa distância, onde pode haver vários caminhos para o dado atingir o computador receptor.” Para garantir o controle do envio e recebimento de dados através das redes, o protocolo IP é dividido em cabeçalho e corpo da mensagem. O cabeçalho é dividido em uma parte de tamanho

fixo de 20 bytes e outra parte de tamanho variável que é opcional. A figura 1 retrata todas as partes do cabeçalho do datagrama IP.

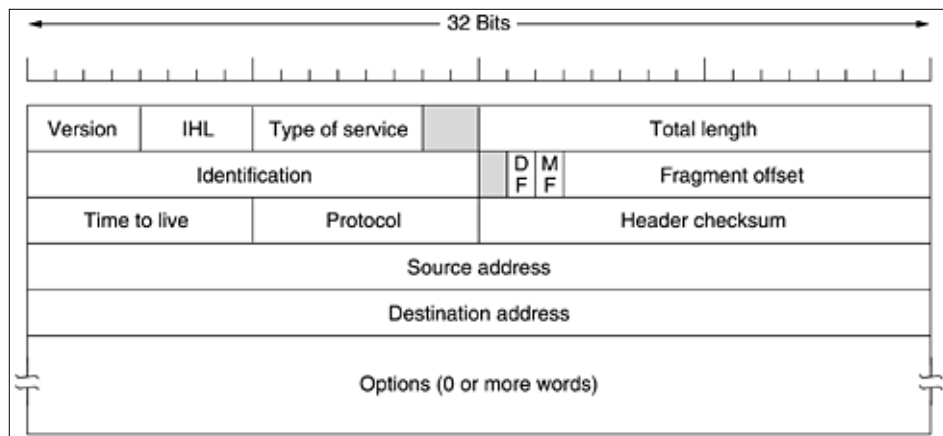


Figura 1: Datagrama IPv4.

Fonte: Tanenbaum (2003)

Cada campo tem a tarefa de controlar cada especificidade do IP. Eis uma descrição de cada um:

- Version: O campo version serve para fazer o controle da versão do protocolo a qual o datagrama pertence. Necessário para observar a mudança de versão, o que podemos observar hoje com a migração do IPv4 para o IPv6.
- IHL (Internet Header Length): O IHL serve para controlar o tamanho do datagrama IP. Nesse campo, que possui 4 bits de tamanho, é informado o seu tamanho em palavras de 32 bits além de informar onde realmente os dados começam.
- Type of services: Tipo de serviço, como o próprio nome já diz. Esse campo serve para definir a que serviço pertence o datagrama. Dessa forma os roteadores têm a possibilidade de implementar a melhor forma de administrar o fluxo de dados. Transmissões de voz, por exemplo, priorizam mais a velocidade do que a garantia de entrega, uma vez que, quando se trata de tráfego de arquivos a integridade deles é priorizada em relação à velocidade de transmissão.
- Total length: Determina o tamanho total do datagrama, podendo atingir tamanho máximo de 65.535 bytes.
- Identification: Serve para o destino identificar todos os fragmentos correspondentes a um determinado datagrama. Quando um datagrama é fragmentado, o campo identification é preenchido com um valor único em todos os

seus fragmentos. Com isso o destino identifica a qual datagrama pertence cada fragmentos que está sendo recebido

- DF: Significa não fragmentar. Essa flag é usada para os roteadores não fragmentarem o datagrama. É recomendado em casos em que o destino é incapaz de juntar os fragmentos.
- MF: Mais fragmentos. Ao contrário do DF, o MF serve para informar aos roteadores que o datagrama foi fragmentado e que ainda existem mais fragmentos. Essa flag não é utilizada no último fragmento.
- Fragment offset: Informa a ordem do fragmento. Indica se é o primeiro, segundo, terceiro ou último fragmento de um determinado datagrama. Todos os fragmentos devem ser múltiplos de 8, com exceção do último. A quantidade máxima de fragmentos por datagrama é 8.192.
- Time to live: Tempo de vida ou TTL. Serve para impedir que datagramas circulem eternamente na rede. A cada hop (salto) o TTL é decrementado, se seu valor chegar a zero e ele não tiver sido entregue ao destino então ele será descartado e uma mensagem de advertência é enviada a origem.
- Protocol: Informa o protocolo de transporte. TCP, UDP, RDP e etc.
- Header checksum: Usado para realizar leitura do cabeçalho a fim de identificar possíveis erros de bits no datagrama. Kurose e Ross definem o algoritmo de contagem de header checksum. Assim: “A soma é calculada tratando cada 2 bytes do cabeçalho como se fosse um número e somando esses números usando complementos aritméticos de 1”.
- Source address / Destination Address: Campos de IP de origem e de destino do datagrama.
- Options: Criado para incluir informações complementares que não são previstas no cabeçalho. Existe uma lista de todas as opções em www.iana.org/assignments/ip-parameters.

Tabela 1: Opções do cabeçalho IPv4.

OPÇÃO	DESCRIÇÃO
Security	Especifica o nível de segurança do datagrama
Strict source routing	Mostra o caminho completo a ser seguido
Loose source routing	Apresenta uma lista de roteadores que não devem ser esquecidos
Record route	Faz com que cada roteador anexe seu endereço IP
Timestamp	Faz com que cada roteador anexe seu endereço e seu timbre de hora

O IP é um endereço único atribuído a cada host na rede. Atualmente o IP está na versão 4 [RFC 791] e esta versão foi inicialmente definida da seguinte forma:

- Endereços de 32 bits, compostos de 4 octetos;
- Os endereços eram divididos em 5 classes mas apenas 3 delas são usadas propriamente para distribuição de endereços;
 - Classe A: definia o bit mais significativo como 0, utilizava os 7 bits restantes do primeiro octeto para identificar a rede, e os 24 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 1.0.0.0 até 126.0.0.0;
 - Classe B: definia os 2 bits mais significativo como 10, utilizava os 14 bits seguintes para identificar a rede, e os 16 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 128.1.0.0 até 191.254.0.0;
 - Classe C: definia os 3 bits mais significativo como 110, utilizava os 21 bits seguintes para identificar a rede, e os 8 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 192.0.1.0 até 223.255.254.0; (MOREIRAS; REIS; ROCHA, 2010)
 - As classes D e E são classes reservadas onde a classe D é destinada endereços de multicast e a classe E reservada para uso futuro. (TANENBAUM, 2005)

Um ponto importante ao definir as configurações de IP é referente à Máscara de Rede. Ela é formada por 32 bits, assim como o endereço IP. Cada bit 1 da máscara informa a parte do endereço que é usada para definir a rede. Cada bit 0 informa a parte do endereço que é usada para definir o endereço.

Tabela 2: Máscara das classes de IP.

CLASSE	MÁSCARAS DE REDES
Classe A	255.0.0.0
Classe B	255.255.0.0
Classe C	255.255.255.0

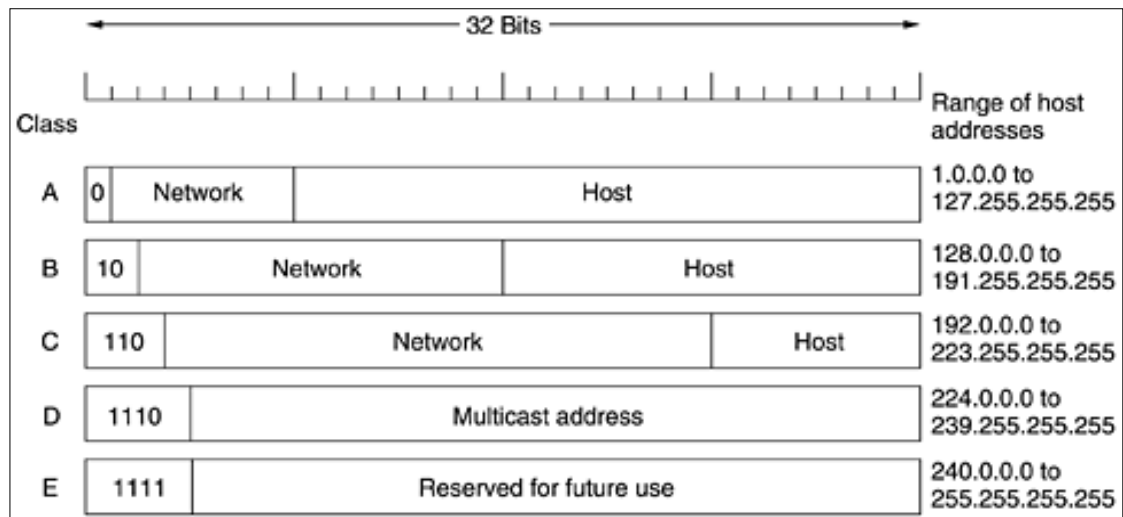


Figura 2: Classes de endereços IPv4.

Fonte: Elaboração Própria

A divisão em classes e sua posterior distribuição de redes causaram um grande problema para o IPv4, uma vez que blocos inteiros da classe A foram distribuídos sem análise da real necessidade. O site ipv6.br alerta para o fato de “[...] o espaço reservado para a classe A atenderia a apenas 128 entidades ligadas à Internet, no entanto, ocupava metade do espaço disponível. Não obstante, empresas e entidades como HP, GE, DEC, MIT, DISA, Apple, AT&T, IBM, USPS, dentre outras, receberam alocações classe A” (MOREIRAS, 2009).

2.3 SOLUÇÕES PALIATIVAS PARA O FIM DO IPv4

Com a distribuição de praticamente metade dos IP's disponíveis para poucas organizações e uma grande parte do que restou sendo reservado para a IANA, soluções paliativas foram tomadas para garantir uma continuidade no IPv4 até que uma solução definitiva fosse desenvolvida para substituí-lo. Dentre essas, 4 se destacam por serem as mais eficientes, vistas nas seções seguintes.

2.3.1 *Classless Inter-Domain Routing (CIDR)* [RFC 1519]

O CIDR foi definido para conter o acelerado processo de esgotamento dos endereços IP. “A ideia básica por trás do CIDR, descrito na RFC 1519 é alocar os endereços IP restantes em blocos de tamanho variável, sem levar em consideração as classes” (TANENBAUM, 2003a).

Basicamente as classes de IP deixaram de ser utilizadas e as máscaras de rede passaram a ser definidas de acordo com a necessidade de cada um. Desperdícios como, por exemplo, a atribuição de uma rede de classe B (65.536 IP's) para uma organização que possui 2.000 endereços não acontece mais.

Neste caso seria atribuído um bloco de 2.048 endereços, uma economia de 63.488 endereços. Em consequência da abertura das máscaras de rede, o encaminhamento passou a ser mais complexo.

O roteamento que anteriormente era feito basicamente a partir de 3 tamanhos fixos de redes deram lugar a algoritmos complexos que realizam comparações com todos os valores que existem na sua tabela de roteamento.

2.3.2 Endereços privados

A RFC 1918 especificou endereços privados, não válidos na Internet, que poderiam ser utilizados, por exemplo, nas redes corporativas (COMITÊ GESTOR DA INTERNET NO BRASIL, 2008). Tal modificação permitiu que instituições utilizassem uma faixa de endereços sem que houvesse conflito com endereços já distribuídos na Internet. A Tabela 3 expõe as faixas de endereços privados que foram reservados a partir da RFC 1918.

Tabela 3: Endereços IPv4 usados em Redes Privadas.

Classe	Faixa de endereços de IP	Notação CIDR	Número de Redes	Número de IPs	IPs por rede
Classe A	10.0.0.0 – 10.255.255.255	10.0.0.0/8	128	16.777.216	16.777.214
Classe B	172.16.0.0 – 172.31.255.255	172.16.0.0/16	16.384	1.048.576	65 534
Classe C	192.168.0.0 – 192.168.255.255	192.168.0.0/24	2.097.150	65.535	254

2.3.3 Network Address Translation (NAT)

O *Network Address Translation* (NAT) permitiu que redes locais que utilizam endereços privados se conectassem a Internet. Utilizando o NAT pode-se, a partir de um único endereço válido, conectar mesmo que de forma limitada, toda uma instituição a Internet.

A figura 3 mostra os passos do procedimento de NAT feito pelo roteador de borda de uma rede privada.

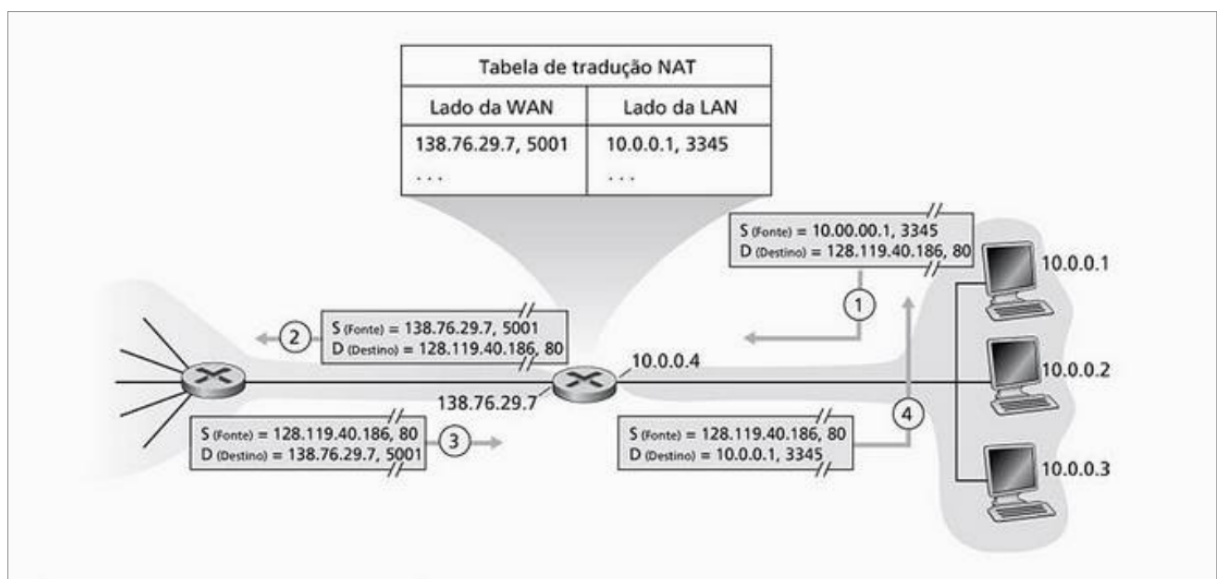


Figura 3: Processo de Tradução de Endereços (NAT).

Fonte: Kurose; Ross (2005).

De acordo com Tanenbaum (2003a):

“Sempre que um pacote de saída entra na caixa NAT, o endereço de origem 10.x.y.z é substituído pelo endereço IP verdadeiro da empresa. Além disso, o campo Source port do TCP é substituído por um índice para a tabela de conversão de 65.536 entradas da caixa NAT. Essa entrada de tabela contém a porta de origem e o endereço IP original.”

Embora seja muito eficiente quanto à diminuição da utilização de novos IP's válidos, o uso do NAT gera uma série de problemas: ele acaba com o modelo de funcionamento peer to peer e, com isso, aplicações como o FTP que inserem os endereços no corpo da mensagem passam a necessitar de ajustes especiais; da mesma forma com aplicações de voz sobre IP baseadas em SIP; ele não escala bem, pois exige processamento pesado; ele não funciona com IPsec; o NAT viola a regra mais fundamental da distribuição de protocolos em camadas onde uma camada não pode fazer qualquer suposição ou modificação sobre uma camada superior; entre outros.

2.3.4 *Dynamic Host Configuration Protocol (DHCP)*

Descrito pela RFC 2131 e 2132, esse protocolo permite a alocação dinâmica de endereços IP, o que trouxe a possibilidade aos provedores de reutilizarem endereços válidos fornecidos a seus clientes para conexões não permanentes, como as realizadas através de linhas discadas ou ADSL, de acordo com a figura 4 (COMITÊ GESTOR DA INTERNET NO BRASIL, 2008).

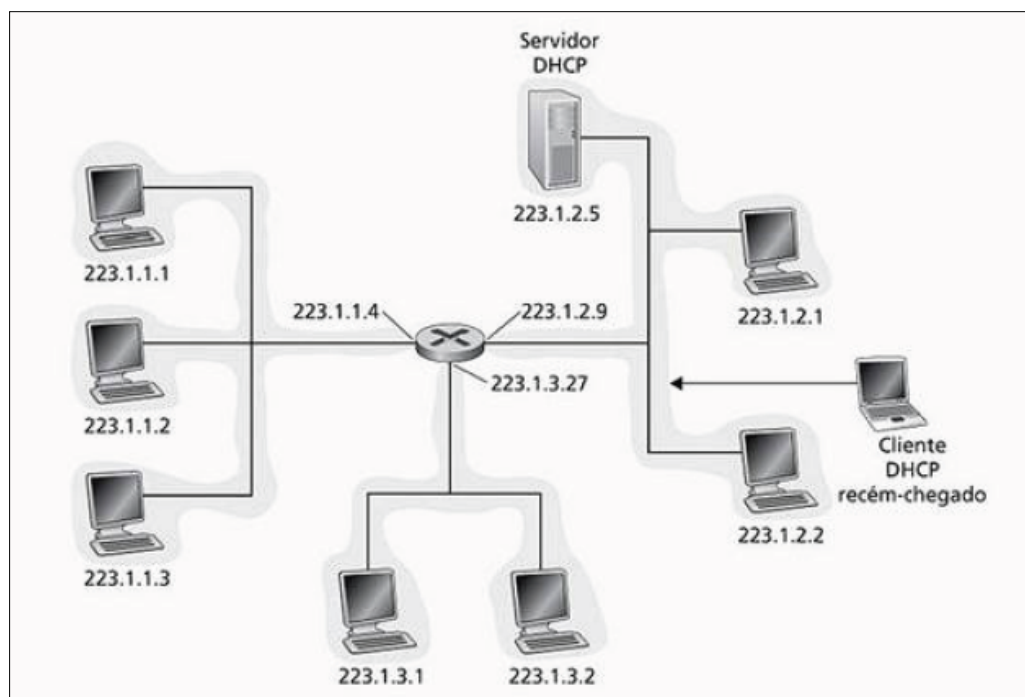


Figura 4: Protocolo DHCP.

Fonte: Kurose; Ross (2005).

A distribuição de endereços através do DHCP é realizada em 4 etapas, como mostra a figura 5:

- A descoberta do servidor DHCP;

- A oferta do servidor DHCP;
- A solicitação DHCP; e
- A resposta do servidor a esta solicitação (DHCP ACK).

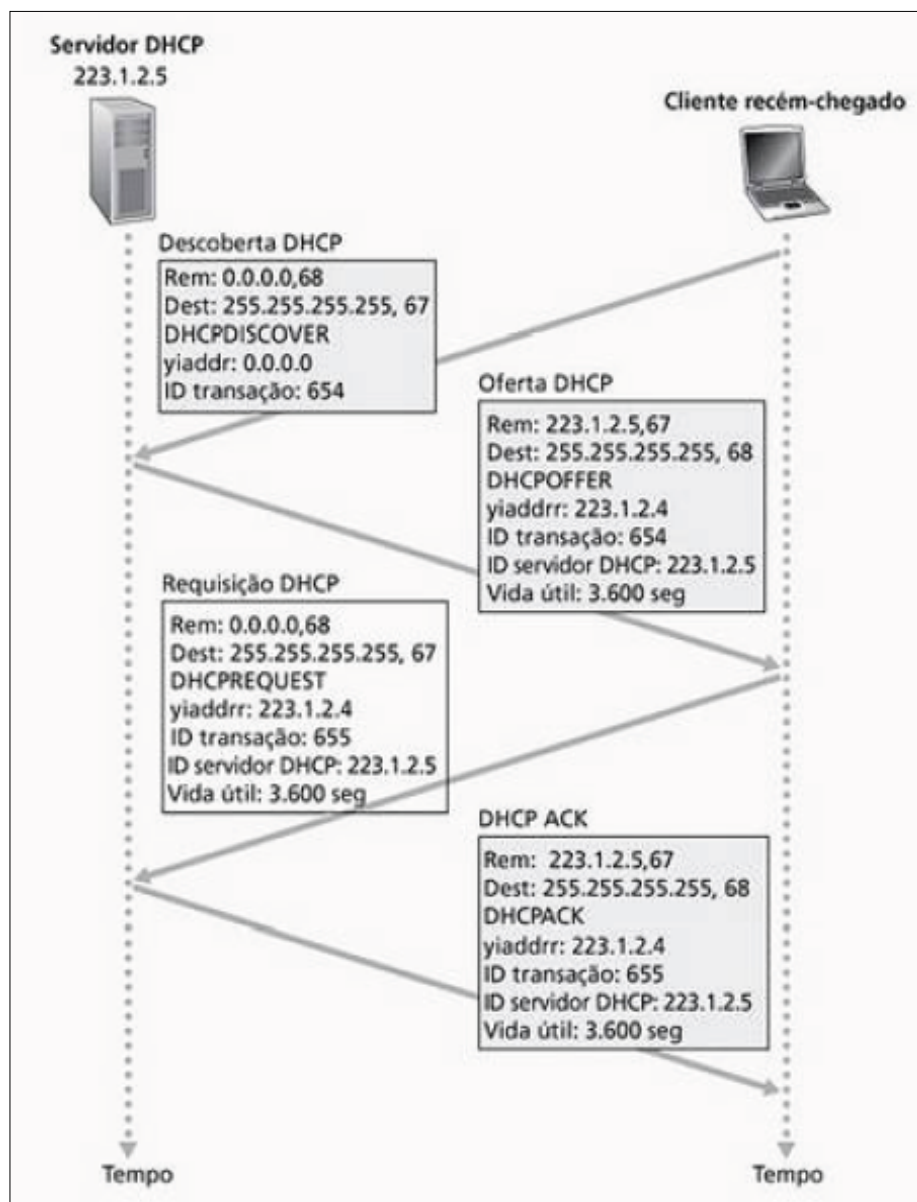


Figura 5: Requisição DHCP.

Fonte: Kurose; Ross (2005).

Durante o processo de comunicação um host novo na rede realiza a descoberta de servidores através de um pedido enviado pela porta 67 UDP em broadcast na rede e aguarda uma resposta. O(s) servidor (es) DHCP ativo(s) na rede responde a solicitação com uma oferta DHCP também em broadcast. Na mensagem de oferta, o servidor DHCP envia o IP oferecido ao cliente, máscara de rede e o tempo de concessão do endereço IP. Uma vez de posse das ofertas enviadas pelos servidores DHCP o cliente escolhe uma e responde ao servidor com

uma mensagem de solicitação DHCP repetindo os parâmetros informados anteriormente pelo mesmo. Finalmente o servidor envia uma mensagem DHCP ACK confirmando a concessão de IP ao cliente.

Depois de atribuído o IP ao cliente, ele já pode utilizar o endereço para se comunicar com os outros hosts da rede durante o período de concessão definido pelo servidor DHCP. Quando está próximo da expiração da concessão, o cliente envia nova requisição ao servidor solicitando renovação da concessão sobre o endereço IP.

2.4 PROTOCOLO IPV6

IPv6 é a nova geração do Protocolo Internet e sua concepção se deve ao fato do esgotamento iminente dos endereços IP da versão 4. Apesar de já estar sendo utilizado há algum tempo, a nova versão do protocolo IP ainda não está sendo largamente utilizada, mas a sua utilização é imprescindível para a continuidade do crescimento da Internet.

2.4.1 Características

O IPv6 manteve a maioria das características facilitadoras do seu antecessor mas, apesar disso algumas mudanças precisavam ser feitas para garantir melhorias na comunicação host a host. Eis algumas características do novo protocolo:

- Tamanho de endereço: Seu endereço sofreu uma mudança considerável de tamanho, passando dos atuais 32 bits para 128 bits;
- Formato do cabeçalho: Quase todos os campos do cabeçalho IPv6 são diferentes do IPv4; alguns foram apenas mudados e outros foram substituídos;
- Suporte a áudio e vídeo: No IPv6 foi implementado um mecanismo que permite determinar um caminho de alta qualidade por onde vão trafegar os datagramas de áudio e vídeo;
- Protocolo extensível: O novo protocolo possui um esquema que possibilita adicionar novas informações no datagrama, permitindo que novas características possam ser agregadas ao protocolo a qualquer momento, diferente do IPv4 que impossibilita esse tipo de intervenção;
- Cabeçalho de extensão: Na nova versão do protocolo IP, as informações são codificadas em cabeçalhos separados, onde o datagrama passa a consistir de um

cabeçalho base IPv6, seguido por zero, para indicar o fim do cabeçalho, ou mais cabeçalhos de extensão.

2.4.2 Datagrama

Formado por cabeçalho base, cabeçalhos de extensão (opcionais) e área de dados. Observe o exemplo na figura 5.

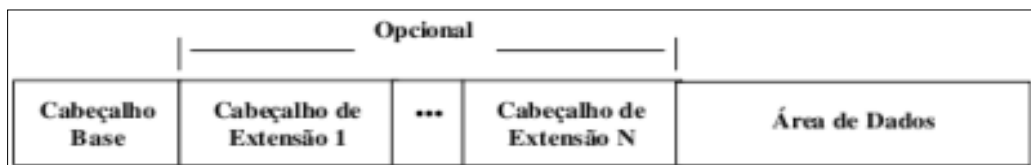


Figura 6: Datagrama IPv6.

Fonte: Comer, 2007.

2.4.3 Cabeçalho base

Embora seja maior em tamanho que o cabeçalho IPv4, vide figura abaixo, a nova versão do cabeçalho IP possui menos informações:

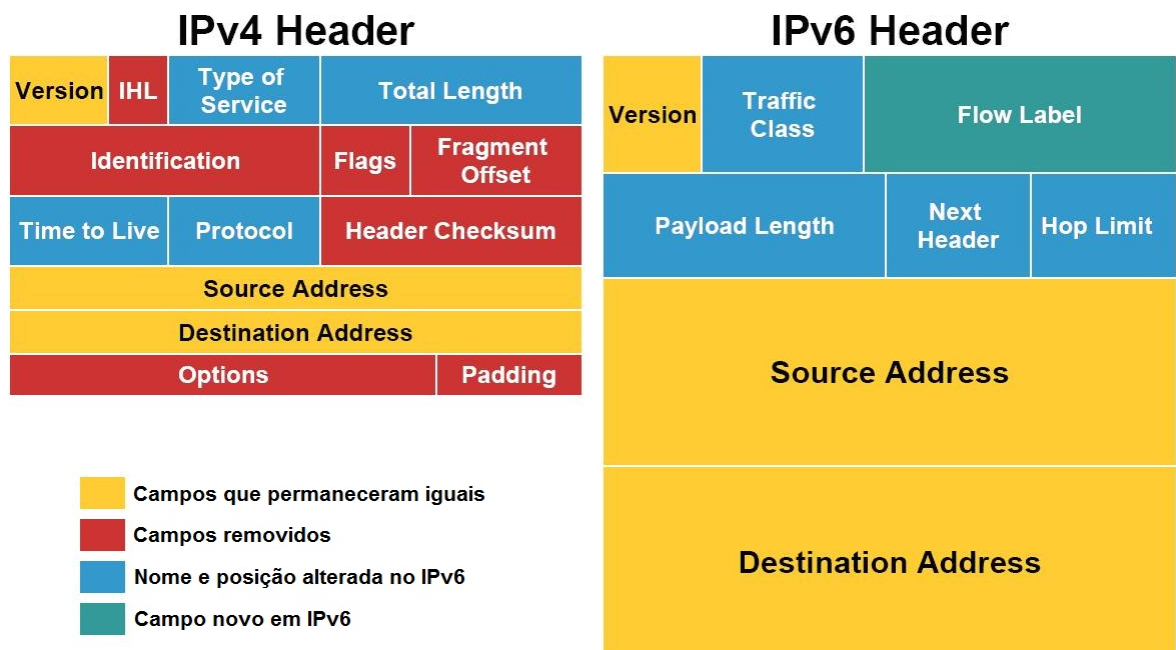


Figura 7: Cabeçalho IPv4 x IPv6.

Fonte: Velozo, 2012.

- Version: Identifica a versão do protocolo;
- Traffic class: Informa a classe de tráfego usada para escolher a rota. São características gerais necessárias ao datagrama. Podemos, com ele, especificar uma classe geral com baixo atraso para, por exemplo, transmissão em tempo real de áudio e vídeo;
- Payload length: Especifica o tamanho dos dados que estão sendo transportados. É importante frisar que o tamanho do cabeçalho não é considerado;
- Hop limit: Análogo ao Time-to-live do IPv4, quando esse campo chega a zero e o pacote não foi entregue ao destino, então ele é descartado;
- Flow label: É usado para especificar um caminho na rede. Quando um caminho é determinado através do *Traffic class*, o campo terá um identificador do caminho determinado que será colocado nesse campo para que os datagramas sejam roteados através desse caminho;
- Next header: Se o datagrama possuir um cabeçalho de extensão, esse campo irá informar que tipo de cabeçalho é esse. Caso contrário, o campo informa o tipo de dado que está sendo carregado no datagrama;
- Source address: Endereço de origem;
- Destination address: Endereço de destino.

A figura 8 mostra o formato do cabeçalho de um datagrama IPv6:

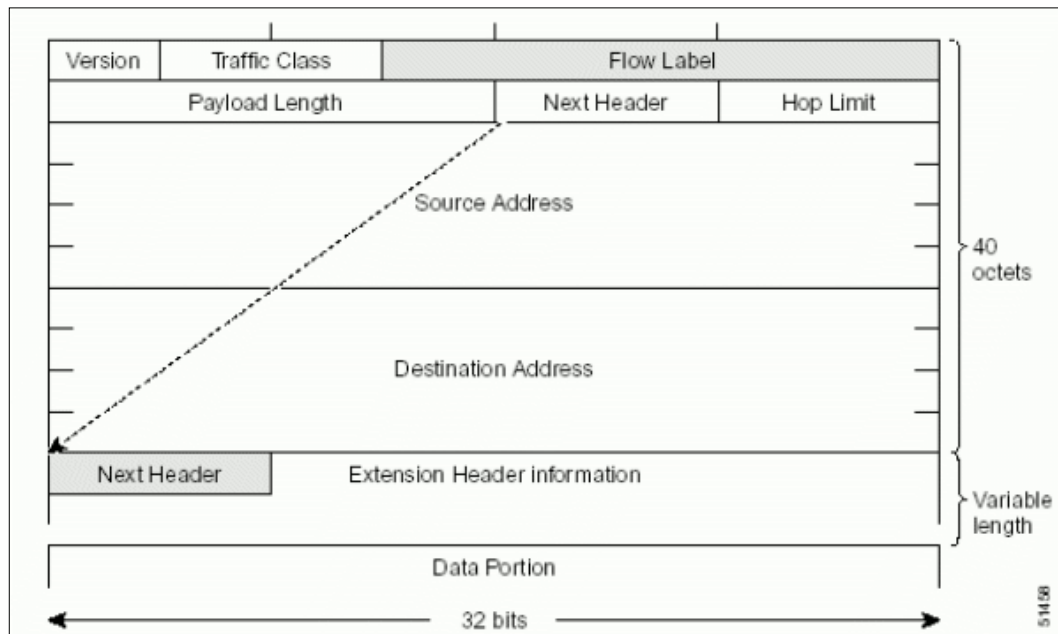


Figura 8: Cabeçalho IPv6.

Fonte: Tanenbaum, 2003.

2.4.4 Faixas de endereços reservados

Da mesma forma que o IPv4 possui faixas de endereços reservados, o novo protocolo também o faz. Na tabela 4 estão as faixas de endereços reservadas.

Tabela 4: Endereços IPv6 Reservados.

2000::/3	Global Unicast – Similar ao endereço público IPv4
FE80::/64	Link Local – Deve ser usado apenas localmente
FC00::/7	Unique Local – Globalmente único
::1/128 (0:0:0:0:0:0:0:1)	Localhost
::/128 (0:0:0:0:0:0:0:0)	Não especificado
64:FF9B::w.x.y.z	IPv4-Mapeado
2002::/16	Faixa especial 6to4
2001:db8::/32	Documentação
2001:0000::/32	Teredo

2.4.5 Protocolo de Controle ICMPv6

O *Internet Control Message Protocol* versão 6 (ICMPv6) foi definido na RFC 4443 e é caracterizado por possuir as mesmas funções do ICMPv4, mas apesar disso eles não são compatíveis entre si. Além disso, o ICMPv6 assume também as funcionalidades de outros protocolos presentes no IPv4, que são o ARP/RARP e o IGMP. Esse protocolo é de fundamental importância no IPv6 por realizar as tarefas de gerenciamento de grupo multicast, descoberta de vizinhança (*Neighbor Discovery*) entre outros.

O ICMPv6 pode ser dividido em duas classes de mensagens: Mensagens de erro e mensagens de informação, onde cada uma possui uma subdivisão. Quando o bit na parte superior do campo tipo de mensagem é 0, identifica uma mensagem de erro, quando este bit é 1 identifica uma mensagem de informação.

As mensagens de erro podem ser as seguintes:

Destination Unreachable: É uma mensagem que é gerada pelo host ou seu gateway de entrada para informar ao cliente que o destino está inacessível por algum motivo;

Packet Too Big: Mensagem enviada por um nó da rede ao endereço de origem para informar que o pacote deve ser enviado por outro nó que possua MTU menor;

Time Exceeded: Mensagem enviada ao endereço de origem informando que o campo

“hop limit” foi excedido e a mensagem foi descartada;

Parameter Problem: Indica erro em algum campo do cabeçalho IPv6.

As mensagens de informação podem ser as seguintes:

Echo Request e Echo Reply: Usados para envio de mensagem de comunicação entre hosts;

Multicast Listener Query: Utilizado para descobrir ouvintes multicast na rede;

Multicast Listener Report: Usada por ouvintes multicast para reportarem interesse em receber o tráfego multicast para endereços específicos ou para responderem a uma query;

Multicast Listener Done: Usada por um ouvinte que queira deixar de receber tráfego de certo endereço multicast;

Router Solicitation e Router Advertisement: Mensagens usadas para detectar roteadores presentes na rede;

Neighbor Solicitation e Neighbor Advertisement: Mensagens usadas para determinar o endereço MAC e a acessibilidade de um vizinho, além de detectar a existência de endereços duplicados;

Redirect: Utilizada por roteadores IPv6 para informar ao host de origem um roteador mais indicado para se alcançar um destino.

2.5 FIREWALL

Segundo a RFC 2979, Firewall é o agente que filtra o tráfego de rede de alguma forma, bloqueando o tráfego que acredita ser impróprio, perigoso ou ambos. Para Kurose/Ross (2005), um Firewall é o resultado da combinação de hardware e software que isola uma rede interna da Internet, realizando o filtro dos pacotes que são trocados entre essas redes.

Geralmente as regras estabelecidas no Firewall se baseiam em controlar o tráfego a partir do IP de origem ou destino e o número da porta. É o que acontece, por exemplo, com uma conexão SMTP oriunda da Internet com destino a rede interna.

2.5.1 Requisito

A RFC 2979 recomenda que a introdução de um Firewall não deve causar falhas não intencionais, ou seja, um Firewall deve ser usado de tal forma a não causar falhas de comunicação que não estejam previstas nas políticas de segurança.

2.5.2 Funcionamento

As regras de segurança a serem determinadas no Firewall possuem 3 ações definidas: aceitar, descartar ou rejeitar. Ainda, podem utilizar as ações para pacotes provenientes da máquina local, da rede externa ou serem encaminhamentos da rede interna com destino a rede externa.

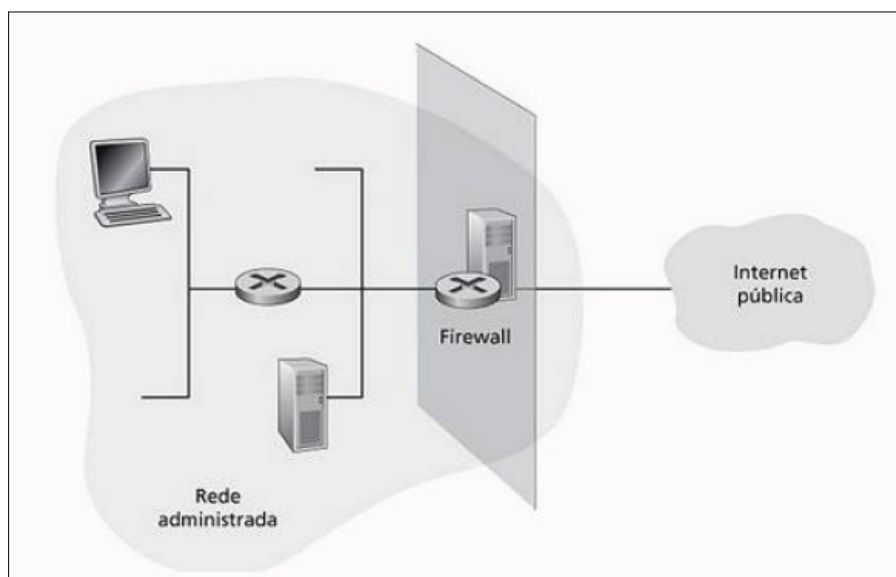


Figura 9: Firewall Básico.

Fonte: Kurose; Ross, 2005.

2.5.3 História

Um firewall (ou parede de fogo) é uma alusão a uma parede corta fogo. Parede essa definida pela ABNT com função de conter as chamas e o calor proveniente do fogo, razão pela qual ela é aplicada nas saídas de emergência e escadas de incêndio. Para Tanenbaum (2003), o Firewall é uma adaptação moderna do sistema de segurança de castelos medievais que cavavam fossos profundos ao redor de seus limites (figura 9). Assim, todos aqueles que quisessem entrar ou sair do castelo precisariam usar a única ponte existente passando, obrigatoriamente, pela revista dos guardas do castelo.

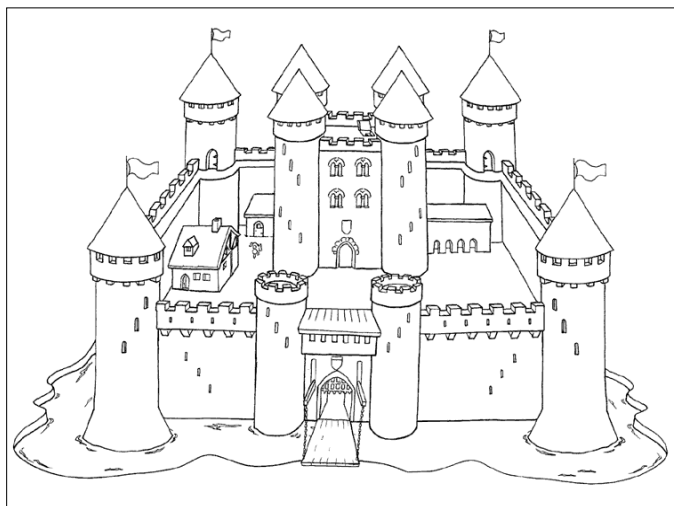


Figura 10: Castelo Medieval.

Fonte: Puglia [2012]

O início da utilização do Firewall se deu no final da década de 1980 e seguiu evoluindo até os dias atuais. O processo de amadurecimento do Firewall é dividido em 4 gerações onde cada geração é marcada por um avanço de suas funcionalidades.

- 1ª Geração: Realizava filtro de pacotes restringindo o tráfego baseado no IP de origem ou destino ou pela porta de serviço TCP ou UDP.
- 2ª Geração: Além das regras definidas na 1ª geração passou a fazer restrições em início de conexões, tráfego de pacotes com número de sequência incorreto e filtro de conexões já estabelecidas.
- 3ª Geração: Chamados de Firewall de aplicação pelo fato de combinar as características de filtro de pacote e proxy, eles possuíam, além de todas as funcionalidades dos seus antecessores, a possibilidade de restringir acesso FTP de usuários anônimos, uso de protocolos desconhecidos na porta 443 (HTTPS) e acesso a portais de entretenimento.
- 4ª Geração: Consolida-se como solução comercial. Começou a ser aperfeiçoado para ser aplicado em uso doméstico, o chamado “firewall pessoal” e também para servidores dedicados. Essa geração ficou marcada por ampliar as características já presentes nas gerações anteriores, entre elas:
 - *Stateful Inspection*, analisando pacotes baseado nas características de cada aplicação. Diferente das gerações anteriores que só analisavam as camadas de rede e aplicação, ele analisa todas as camadas do modelo OSI, além de analisar também a partir do estado das conexões e seções ativas;

- *Deep package inspection*, associa as funcionalidades do stateful inspection com técnicas de IDS;
- Previne o sistema de intrusões identificadas pelo abuso do protocolo TCP/IP, mesmo nas conexões aparentemente legítimas.

2.6 IPTABLES

Desenvolvido por Rusty Russel em colaboração com Michael Neuling e componente da 4ª geração de Firewalls o Iptables, é uma poderosa ferramenta que permite realizar desde controle de pacotes a tratamento de *Quality Of Service* (QoS).

2.6.1 Tabelas e Comandos

O Iptables organiza suas regras de acesso e bloqueio em 3 tabelas, de acordo com o tipo de tráfego que será analisado:

- Filter: Responsável pelas regras de filtro de pacote;
- NAT: Relacionada às regras de Firewall NAT. Tradução de endereços de rede;
- Mangle: Usada para manipular campos do cabeçalho IP. Por exemplo, TOS e TTL.

Cada tabela possui um conjunto de regras que podem ser aplicadas. No caso da tabela *Filter*, teremos 3 *Chains* diferentes:

- Input: Tudo que é enviado para a máquina local;
- Output: Tudo o que é enviado da máquina local; e.
- Forward: Tudo que chega a máquina local, mas que deve ser redirecionado a outra máquina ou interface.

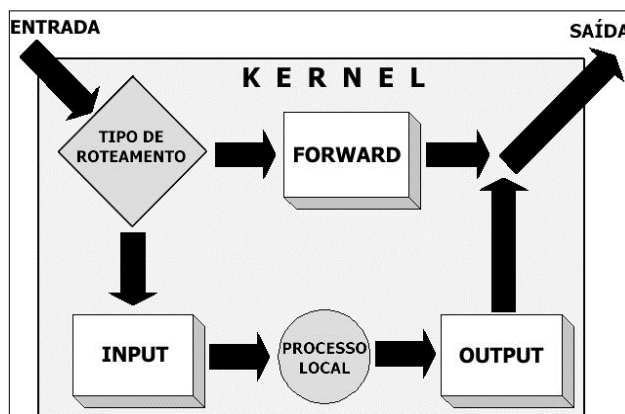


Figura 8: Tabela *Filter*.

Fonte: Mota Filho, 2005.

Já a tabela NAT possui outras Chains, são elas:

- *Prerouting*: Usada para fazer alterações em pacotes antes de eles serem enviados;
- *Output*: Trata os pacotes emitidos pela máquina local; e,
- *Postrouting*: Realiza alterações em pacotes depois de serem roteados.

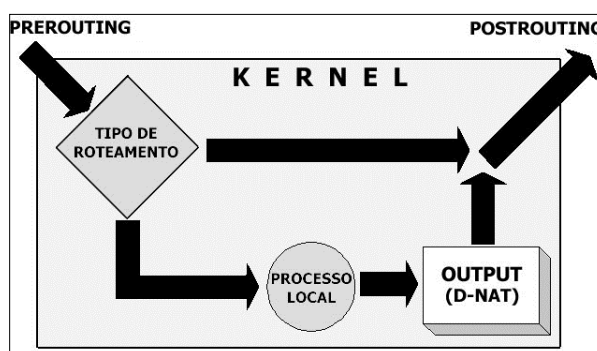


Figura 9: Tabela *Nat*.

Fonte: Mota Filho, 2005.

A tabela *Mangle* possui apenas 2 Chains: *Prerouting* e *Output*, onde *Prerouting* modifica o cabeçalho dos pacotes antes que eles sejam roteados e *Output* altera o cabeçalho dos pacotes da máquina local antes deles serem roteados.

Os comandos aceitos pelo Iptables são os seguintes:

- A : Adiciona uma nova regra ao fim da lista de regras;
- D : Apaga uma regra especificada na lista;
- L : Mostra as regras existentes na lista;

- P : Altera a política padrão de uma Chain; a política padrão definida no Iptables é Accept;
 - F : Apaga todas as regras existentes em uma lista sem alterar a política padrão;
 - I : Insere uma nova regra no início da lista de regras;
 - R : Substitui uma regra já existente;
 - N : Permite criar uma nova Chain a tabela especificada;
 - E : Renomeia uma nova Chain. Só funciona com Chains criadas pelo administrador do firewall;
 - X : Apaga uma Chain criada pelo administrador do firewall;
- As ações do Iptables são as seguintes:
- p : Especifica o protocolo aplicado a regra (tcp, udp, icmp, etc);
 - i : Especifica a interface de entrada a ser utilizada. São utilizadas nas Chains Input e Forward;
 - o : Especifica a interface de saída a ser utilizada. São utilizadas nas Chains Output e Forward;
 - s : Especifica a origem do pacote que passarão pela regra do firewall, seu valor pode ser apenas um host ou uma rede inteira;
 - d : Especifica o destino do pacote que passará pela regra do firewall; seu valor pode ser apenas um host ou uma rede inteira;
 - ! : Adicionado quando se deseja aplicar uma exceção a uma regra. É usado junto às ações -p, -i, -o, -s e -d;
 - j : Define o alvo do pacote que passará pela regra. Suas ações são ACCEPT, REJECT, DROP e LOG;
 - sport : Aplicado para definir regras a partir da porta de origem do pacote;
 - dport : Aplicado para definir regras a partir da porta de destino do pacote;

2.6.2 A ação ser tomada (*Target*)

Target é a ação tomada pela regra em relação ao pacote. Toda regra possui um target e é ele quem determina que tratamento será dado ao pacote, seja aceitando (ACCEPT), descartando (DROP) ou rejeitando (REJECT). Ainda existe um target específico para gerar log (LOG) dos pacotes submetidos à regra.

2.6.3 NAT e Filtragem de pacotes

Como dito anteriormente, o NAT é uma solução bastante difundida em redes IPv4. Essa solução é tão usada que chega a ser um dos principais fatores da resistência de alguns administradores de rede em iniciar a migração para a nova versão do protocolo IP. Além disso, o Iptables funciona bem realizando a tarefa de NAT aliado a estratégia de filtragem de pacotes que passam pelo firewall.

Primeiramente, você projeta e constrói seu filtro de pacotes ignorando totalmente qualquer NAT a ser feito. As origens e destinos que o filtro de pacotes verá serão as origens e destinos reais. Por exemplo, se você fará DNAT para mandar conexões do endereço 1.2.3.4 porta 80 para 10.1.1.1 porta 8080, o filtro de pacotes verá pacotes indo para 10.1.1.1 porta 8080 (o destino real), e não 1.2.3.4 porta 80. Similarmente, você pode ignorar o masquerading: os pacotes vão parecer que têm como origem o real endereço IP interno (por exemplo 10.1.1.1), e as respostas vão parecer que têm como destino esse mesmo endereço (RUSSELL, 2001).

2.7 IP6TABLES

Ip6tables é utilizado para configurar, manter e inspecionar as tabelas de regras de pacotes IPv6 no kernel Linux. Várias tabelas diferentes podem ser definidas. Cada tabela contém um número de chains internas, mas também pode conter chains acrescentadas pelo usuário.

2.7.1 Tabelas e Comandos

Assim como o iptables, o ip6tables também possui 3 tabelas nativas. A diferença fica na saída da tabela NAT e a entrada da tabela Raw:

- *Filter*: Ele contém o chains **INPUT** (para pacotes destinados a sockets locais), **FORWARD** (para pacotes sendo roteados através da máquina) e **OUTPUT** (para pacotes gerados localmente).
- *Mangle*: Contém as chains **PREROUTING** (para alterar pacotes que chegam antes do roteamento), **OUTPUT** (para alterar pacotes gerados localmente antes do roteamento), **INPUT** (para pacotes entrando na própria máquina), **FORWARD** (para alterar pacotes sendo roteados através da máquina) e **POSTROUTING**

(para alterar pacotes quando eles estão prestes a sair).

- *RAW*: Esta tabela é usada principalmente para a configuração de isenções de rastreamento de conexões em combinação com o alvo NOTRACK. Ele registra nos ganchos netfilter com maior prioridade, e é assim chamado antes ip_conntrack, ou quaisquer outras tabelas IP. Ele fornece as seguintes chains: **PREROUTING** (para pacotes que chegam através de qualquer interface de rede) **OUTPUT** (para pacotes gerados por processos locais)

Os comandos utilizados pelo ip6tables são os mesmos que estão presentes no iptables; portanto, não há necessidade de repeti-los nesse espaço.

3. ESTUDO DE CASO

Neste capítulo será mostrado um estudo de caso da utilização dos Firewalls Ip6tables na rede Ipv6, a fim de ilustrar a configuração que deve ser adotada na mudança de protocolo, além da mudança na sintaxe de comandos.

3.1 CARACTERÍSTICAS DA REDE ADOTADA

Para efeitos de estudo de caso será considerado uma rede local composta por servidores de controle de rede onde um desses servidores será nosso firewall que também agirá como gateway da rede. Esse cenário de rede corporativa terá as seguintes características:

- 1 Servidor Firewall/Gateway;
- 1 Servidor Web, FTP, SSH;
- 1 Servidor Controlador de domínio
- 5 estações de trabalho;
- 1 Switchs gigabit ethernet; e
- Conexão Internet banda larga.

Onde o servidor Firewall/Gateway terá o sistema operacional Linux Debian Versão 6 (Squeeze) com 2 placas de rede gigabit ethernet. Uma conectada a rede local e outra diretamente a Internet, além de possuir o serviço SSH devidamente instalado e ativo. Teremos outro servidor genérico que irá trabalhar com os serviços FTP, HTTP e outro servidor Microsoft Windows como controlador de Domínio. A configuração dos serviços de rede não será abordada nesse projeto. A rede 2001:db8:d0ca:: vai ser utilizada para representar a internet nos testes de laboratório.

A figura 13 ilustra a estrutura organizacional da infraestrutura de rede desse projeto, levando em consideração a disposição dos equipamentos.

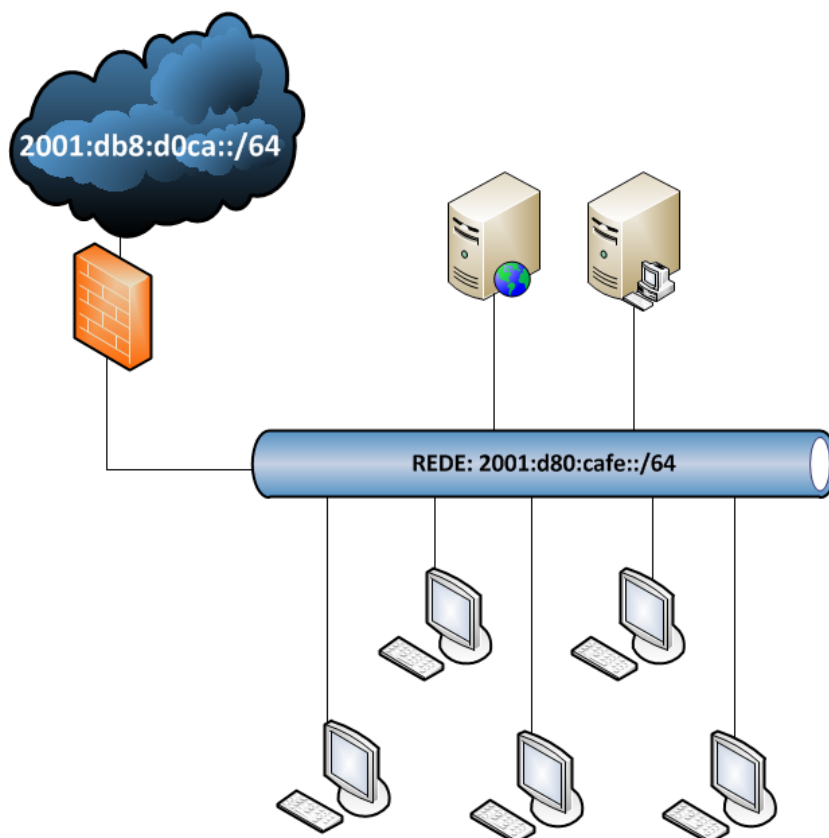


Figura 10: Rede do Estudo de Caso.

Fonte: Elaboração Própria

3.2 PARÂMETROS

A seguir, é exibida a lista das características do nosso firewall de rede que serão definidas para segurança da infraestrutura em questão:

- Prevenir os tipos de ataques mais conhecidos;
- Filtro de dados oriundos da rede externa (Internet);
- Disponibilizar o acesso aos serviços Web, FTP e SSH tanto das estações da rede interna quanto da internet;
- Negar conexões não autorizadas às estações da rede interna; e
- Negar todo tipo de acesso ao Firewall de rede que não foi previsto anteriormente.

A Tabela 5 contém o script base iptables com todas as regras que serão utilizadas no nosso firewall de rede Linux, o qual será analisado.

Tabela 5: Script do Firewall Iptables.

```
#####
##### REGRAS FW – IPV6 #####
#####

#!/bin/bash
ip6tables -F
ip6tables -P INPUT DROP
ip6tables -P OUTPUT ACCEPT
ip6tables -P FORWARD DROP
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
ip6tables -A INPUT -p udp -m udp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe:: \
--dport 33434:33523 -m state --state NEW -j REJECT --reject-with \
icmp6-port-unreachable
ip6tables -A INPUT -p icmpv6 --icmpv6-type 135 -d ff02::1:ff00:0/104 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -d 2001:db8:cafe:: -m limit \
--limit 10/s -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d 2001:db8:cafe:: -m limit \
--limit 10/s -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -d 2001:db8:cafe:: \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -d 2001:db8:cafe:: -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-transit -d 2001:db8:cafe:: \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d \
2001:db8:cafe:: -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-option -d 2001:db8:cafe:: \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-header-type -d 2001:db8:cafe:: \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type bad-header -d 2001:db8:cafe:: -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 136 -d 2001:db8:cafe:: -j ACCEPT
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 22 \
-j ACCEPT
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 20 \
-j ACCEPT
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 21 \
-j ACCEPT
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 80 \
-j ACCEPT
```

```

ip6tables -A FORWARD -m state -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::/48 \
--state RELATED,ESTABLISHED -j ACCEPT
ip6tables -A FORWARD -m state -s 2001:db8:cafe::/48 -d 2001:db8:d0ca::/48 \
--state NEW,RELATED,ESTABLISHED -j ACCEPT
ip6tables -A FORWARD -p udp -m udp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::/48 \
--dport 33434:33523 -m state --state NEW -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type echo-request -d 2001:db8:cafe::/48 \
-m limit --limit 10/s -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type echo-reply -d 2001:db8:cafe::/48 -m \
limit --limit 10/s -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type destination-unreachable \
-d 2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type packet-too-big -d 2001:db8:cafe::/48 \
-j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type ttl-zero-during-transit \
-d 2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type ttl-zero-during-reassembly \
-d 2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type unknown-option -d 2001:db8:cafe::/48 \
-j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type unknown-header-type \
-d 2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type bad-header -d 2001:db8:cafe::/48 \
-j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type 136 -d 2001:db8:cafe::/48 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -d fe80::/64 \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-transit -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d fe80::/64 \
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-option -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-header-type -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type bad-header -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 134 -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 135 -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 136 -d fe80::/64 -j ACCEPT
#####

```

Explicando as ações tomadas no script, bloco a bloco:

- Política padrão, Permitindo OUTPUT e proibindo INPUT e FORWARD

```
ip6tables -F
```

```
ip6tables -P INPUT DROP
```

```
ip6tables -P OUTPUT ACCEPT
```

```
ip6tables -P FORWARD DROP
```

- Permitindo tráfego pela interface Loopback

```
ip6tables -A INPUT -i lo -j ACCEPT
```

- Permitindo saída e retorno somente para conexões existentes

```
ip6tables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

- Permitindo traceroute da rede externa para o roteador

```
ip6tables -A INPUT -p udp -m udp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe:: --dport 33434:33523 -m state --state NEW -j REJECT --reject-with icmp6-port-unreachable
```

- Permitindo respostas do roteador para solicitações icmp

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type 135 -d ff02::1:ff00:0/104 -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -d 2001:db8:cafe:: -m limit --limit 10/s -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d 2001:db8:cafe:: -m limit --limit 10/s -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -d 2001:db8:cafe:: -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -d 2001:db8:cafe:: -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-transit -d 2001:db8:cafe:: -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d 2001:db8:cafe:: -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-option -d 2001:db8:cafe:: -j
ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-header-type -d 2001:db8:cafe:: -j
ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type bad-header -d 2001:db8:cafe:: -j ACCEPT
```

```
ip6tables -A INPUT -p icmpv6 --icmpv6-type 136 -d 2001:db8:cafe:: -j ACCEPT
```

- Permitindo acesso SSH, FTP e HTTP para 2001:db8:cafe::1

```
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 22 -j
ACCEPT
```

```
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 20 -j
ACCEPT
```

```
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 21 -j
ACCEPT
```

```
ip6tables -A FORWARD -p tcp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::1 --dport 80 -j
ACCEPT
```

- Permitindo tráfego de conexões iniciadas pela rede interna

```
ip6tables -A FORWARD -m state -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::/48 --state
RELATED,ESTABLISHED -j ACCEPT
```

```
ip6tables -A FORWARD -m state -s 2001:db8:cafe::/48 -d 2001:db8:d0ca::/48 --state
NEW,RELATED,ESTABLISHED -j ACCEPT
```

- Permitindo traceroute da rede externa para a rede interna

```
ip6tables -A FORWARD -p udp -m udp -s 2001:db8:d0ca::/48 -d 2001:db8:cafe::/48 --dport
33434:33523 -m state --state NEW -j ACCEPT
```

- Permitindo respostas da rede interna para solicitações icmp da rede externa

```
ip6tables -A FORWARD -p icmpv6 --icmpv6-type echo-request -d 2001:db8:cafe::/48 -m
limit --limit 10/s -j ACCEPT
```

```
ip6tables -A FORWARD -p icmpv6 --icmpv6-type echo-reply -d 2001:db8:cafe::/48 -m limit
--limit 10/s -j ACCEPT
```

```
ip6tables -A FORWARD -p icmpv6 --icmpv6-type destination-unreachable -d
```

```

2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type packet-too-big -d 2001:db8:cafe::/48 -j
ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type ttl-zero-during-transit -d
2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d
2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type unknown-option -d 2001:db8:cafe::/48 -j
ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type unknown-header-type -d
2001:db8:cafe::/48 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type bad-header -d 2001:db8:cafe::/48 -j
ACCEPT
ip6tables -A FORWARD -p icmpv6 --icmpv6-type 136 -d 2001:db8:cafe::/48 -j ACCEPT

```

- Permitindo tráfego icmp de link local no roteador

```

ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -d fe80::/64 -j
ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-transit -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type ttl-zero-during-reassembly -d fe80::/64 -j
ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-option -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type unknown-header-type -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type bad-header -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 134 -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 135 -d fe80::/64 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 136 -d fe80::/64 -j ACCEPT

```

O script inicia definindo a política padrão para aceitar todo o tráfego de saída da rede e bloquear todo o tráfego externo com destino ao firewall e a rede interna. Dessa forma livramos tanto o firewall quando um servidor da rede interna da ameaça de ser acessado da

rede externa indevidamente.

Em seguida é definido que o tráfego de loopback será aceito além de liberar a entrada de tráfego, desde que a conexão tenha sido iniciada pelo próprio firewall e uso de traceroute da rede externa com destino ao firewall. Essas definições permitem que serviços executados no próprio firewall funcionem normalmente além de definir que, inicialmente, a única abertura para troca de tráfego entre o firewall e a rede externa será a partir de conexões iniciadas pelo próprio firewall.

A liberação do traceroute acontece para permitir acompanhamento de comunicação entre um host externo e o firewall, a liberação acontece com o mesmo intuito para echo-reply e echo-request. Em seguida as mensagens de controle e de erro são liberadas para manter a comunicação destination-unreachable, packet-too-big, ttl-zero-during-transit, ttl-zero-during-reassembly, unknown-option, unknown-header-type e bad-header. Por último libera-se o Neighbor Advertisement para a rede interna.

Na segunda parte foram determinadas as regras de acesso aos serviços habilitados na rede interna tais como FTP, HTTP e SSH, desde que a origem seja a rede externa determinada, especificada nas respectivas portas 20, 21, 22 e 80, permitindo tráfego da rede externa com destino a rede interna somente para conexões já iniciadas e tráfego da rede interna para a rede externa desde que sejam novas ou já iniciadas. Em seguida é liberada a comunicação de controle e de erro do icmp, como foi feito anteriormente, nesse caso, com o destino sendo a rede externa.

Na terceira parte são definidas as mesmas liberações para link local, permitindo o tráfego de mensagens de controle e erro e permitindo também os protocolos de descoberta de vizinhança.

O script acima foi usado em um cenário de testes utilizando máquinas virtuais e apresentou resultados satisfatórios com relação à segurança da rede. O controle mais efetivo do protocolo icmp é de fundamental importância para o bom funcionamento do firewall ipv6. As mensagens de controle e de erro aparecem para definir uma ordem na comunicação da nova rede. Sabe-se que as políticas de segurança de uma rede de computadores não se resumem apenas ao firewall, mas o mesmo é de fundamental importância para a garantia de integridade dela. Portanto, possuir um Firewall bem configurado é um passo muito importante para prevenção de incidentes de segurança na nossa rede.

4. CONCLUSÃO

Após verificar as mudanças na forma de endereçamento e observando também que várias modificações foram aplicadas ao IPv6 quando comparado ao IPv4, podemos concluir que a utilização de um firewall a partir do novo protocolo passou a contar com grandes modificações que visam garantir melhor comunicação na rede de computadores. A mais significativa foi o uso mais intenso do protocolo ICMP.

Algumas soluções adotadas visando postergar a adoção do novo protocolo solucionaram alguns problemas e geraram outros para o IPv4 que, inicialmente, não aplicava boas políticas de integridade das informações que transportava. Podemos afirmar que novas ameaças surgem com o novo protocolo, mas outras mais são deixadas para trás a partir da sua utilização.

O processo de comunicação promete ser mais simples e direto, entretanto, também mais aberto. A adoção do IPv6 é necessária e benéfica, volta a proporcionar uma comunicação direta, fim-a-fim, que foi perdida nos últimos tempos no IPv4. A redução do processamento de roteadores possibilita o uso de criptografia nativa para garantir mais segurança na comunicação, além de solucionar o problema da escassez de endereços.

O uso de um firewall iptables não se resume apenas em reproduzir no novo protocolo as regras existentes no antigo. Devemos observar que o IPv6 faz uso massivo do ICMPv6 para estabelecer comunicação, manter o controle das conexões e realizar a descoberta de computadores na vizinhança.

REFERÊNCIAS

CERT.br. **Cartilha de segurança para internet, versão 3.1**. São Paulo: Comitê Gestor da Internet no Brasil, 2006. Disponível em: <
<http://cartilha.cert.br/conceitos/sec7.html#subsec7.1>>. Acesso em: 14 maio 2012.

COMER, Douglas E. **Redes de computadores e internet**. São Paulo: Bookman Companhia, 2007.

COMITÊ GESTOR DA INTERNET NO BRASIL. **Por que o IPv4 ainda não acabou?**. 2008. Disponível em:
 <http://www.ipv6.br/IPV6/AjudaIPV6Tecnico#Por_que_o_IPv4_ainda_n_o_acabou>.
 Acesso em: 16 maio 2012.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a internet: uma abordagem top-down**. 5.ed. Rio de Janeiro: Pearson Education do Brasil, 2005.

MOREIRAS, Antônio M. **Entenda o esgotamento do IPv4**. 2009. Disponível em:
 <<http://www.ipv6.br/IPV6/ArtigoEsgotamentoIPv4>>. Acesso em: 16 maio 2012.

MOREIRAS, Antônio M; REIS, Eduardo Ascenço; ROCHA, Ailton Soares da. **Curso IPv6 básico**. São Paulo: Núcleo de Informação e Coordenação do ponto BR, 2010. Disponível em:
 <<http://www.ipv6.br/pub/IPV6/MenuIPv6CursoPresencial/IPv6-apostila.pdf>>. Acesso em:
 16 maio 2012.

MOTA FILHO, João Eriberto. **Firewall com iptables**. 2005. Disponível em:
 <<http://eriberto.pro.br/iptables/3.html>>. Acesso em: 24 maio 2012.

PUGLIA, San Giuliano di. **Midisegni.it**. [2012]. Disponível em:<<http://www.midisegni.it/storia/disegni/medCastello.gif>>. Acesso em: 24 maio 2012.

RIBEIRO, Sildenir Alves. **Firewall em Linux**. Larvas, 2004. Dissertação (Pós-Graduação Lato Sensu em Administração em Redes Linux). Universidade Federal de Lavras.

RUSSELL, Rusty. **Linux 2.4 Packet Filtering HOWTO**. 2001. Disponível em:
 <<http://www.netfilter.org/documentation/HOWTO/pt/packet-filtering-HOWTO-9.html>>.
 Acesso em: 16 maio 2012.

TANENBAUM, Andrew S. **Computer Networks**. 4 ed. New Jersey: Prentice Hall PTR, 2003a.

TANENBAUM, Andrew S. **Redes de Computadores**. Rio de Janeiro: Campus, 2003.

TORRES, Gabriel. **Redes de computadores: curso completo**. Rio de Janeiro: Axcel Books, 2001.

VELOZO, Priscilla Lusie Coelho. **IPv4 x IPv6**. [2012]. Disponível em:
 <http://www.gta.ufjf.br/ensino/eel879/trabalhos_vf_2009_2/priscilla/images/cabecalho.jpg>.
 Acesso em: 19 maio 2012.

WIKIPEDIA. **DNS cache poisoning**. 2012a. Disponível em: <
http://pt.wikipedia.org/wiki/DNS_cache_poisoning >. Acesso em: 14 maio 2012.

WIKIPEDIA. **Endereço IP**. 2011. Disponível em:
<http://pt.wikipedia.org/wiki/Endere%C3%A7o_IP>. Acesso em: 16 maio 2012.

WIKIPEDIA. **Transbordamento de dados**. 2012. Disponível em:
<http://pt.wikipedia.org/wiki/Transbordamento_de_dados>. Acesso em: 14 maio 2012.