Desenvolvimento de um 'Packet/Session Filter'

Marco Aurélio Spohn

Universidade de Cruz Alta
Rua Andrade Neves, 308 - 98025-810 - Cruz Alta - RS
maspohn@main.unicruz.tche.br

Raul Fernando Weber

Curso de Pós-Graduação em Ciência da Computação
Instituto de Informática, Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500, Caixa Postal 15064 - 91501-970 Porto Alegre - RS
weber@inf.ufrgs.br

Resumo

O filtro de pacotes (packet filter) é um dos componentes básicos de um "firewall". Todo pacote que flui entre a rede interna e a Internet é examinado e só é permitido continuar seu caminho caso exista uma regra que o permita. Portanto, cada pacote que passa pelo filtro deve ser verificado. Uma alternativa para reduzir o "overhead" da filtragem é apresentada pela filtragem baseada na sessão. Na Internet, o protocolo TCP é orientado a conexão; portanto, a utilização de qualquer serviço baseado no TCP implica no estabelecimento de uma sessão. Ao contrário de um filtro convencional, um "session filter" realiza a filtragem dos pacotes baseado nas sessões as quais eles pertencem. O tempo médio de filtragem é reduzido porque os pacotes (de serviços baseados no TCP) são verificados junto a uma "cache" de sessões ativas, sendo que apenas o primeiro pacote (estabelecimento de conexão) precisa ser verificado nas regras de filtragem.

Este trabalho apresenta o desenvolvimento e análise de desempenho de um 'packet/session filter'. O filtro foi implementado junto ao "kernel" do sistema operacional FreeBSD. Além de apresentar um desempenho significativamente melhor que um filtro convencional, o filtro apresenta o recurso de monitoramento de sessões.

Abstract

Packet filter is a basic component of a firewall. The packets that flow between the internal network and the Internet are examined and only allowed to continue their way if there is a rule that allow them. Thus, each packet that pass through the filter must be verified. An alternative to reduce the filtering overhead is presented by filtering based on session. In the Internet, the TCP protocol is connection oriented; thus, any service based on TCP needs first to establish a connection. On the contrary of a conventional filter a session filter accomplishes the packet filtering based on the corresponding sessions they belong to. The filtering mean time is reduced because the packets (of TCP based services) are verified next to a cache of active sessions and only the first packet (connection establishment) needs to be verified within the filtering rules.

This work presents the development and performance analysis of a packet/session filter. The filter was implemented within the kernel of the FreeBSD operating system. Beyond a better performance than a conventional filter the session filter presents the capability of session monitoring.

1 Introdução

Como um primeiro passo ao se implementar uma barreira de segurança em uma rede de computadores, é fundamental que se conheça os detalhes dos protocolos de comunicação utilizados. Na Internet, a atenção deve ser voltada aos protocolos IP, TCP, ICMP e UDP. Estes são os principais protocolos a nível de rede e transporte que são considerados e examinados ao se estabelecer regras de filtragem para um filtro de pacotes na Internet.

A filtragem de pacotes permite controlar o tipo de tráfego em qualquer segmento de rede; conseqüentemente, é possível controlar os serviços que trafegam pela rede. Serviços que comprometem a segurança da rede podem ser restringidos.

Deve-se ressaltar que o processo de filtragem de pacotes causa um "overhead" ao sistema; portanto, para uma situação de tráfego intenso é necessário que se coloque o filtro em uma máquina com uma velocidade de processamento compatível com as necessidades.

1.1 Procedimento básico de filtragem

A filtragem convencional de pacotes na Internet atua até o nível de transporte (UDP e TCP). Portanto, qualquer problema de segurança instrínseco aos protocolos superiores (aplicação, por exemplo) não pode ser resolvido utilizando um filtro de pacotes convencional.

A filtragem dos pacotes é realizada utilizando as seguintes informações presentes nos cabeçalhos dos pacotes:

- Endereço IP fonte;
- Endereço IP destino;
- Protocolo (TCP, UDP ou ICMP);
- "Port" TCP ou UDP fontes;
- "Port" TCP ou UDP destino;
- Tipo de mensagem ICMP (se for o caso).

No protocolo TCP existe um "flag" denominado ACK que é utilizado para confirmação de pacotes e também pode ser utilizado para detectar se o pacote é o primeiro de uma solicitação de conexão. Quando o "flag" não estiver marcado ("bit" com valor 0) significa que o pacote se refere a uma solicitação de conexão e, caso contrário, o pacote corresponde a alguma conexão já existente. Desta forma, o "packet filter" pode bloquear um serviço *inbound* (de fora para dentro; ou seja, o servidor está na rede interna) apenas não permitindo o fluxo de pacotes com o ACK não marcado destinado a um servidor interno associado a "port" (por exemplo, a port 23 do telnet) do serviço bloqueado. Em protocolos não orientados a conexão, como é o caso do protocolo UDP, não é possível tomar nenhuma decisão deste tipo; ou seja, nestes protocolos, nunca se sabe se o pacote que está chegando é o primeiro que o servidor está recebendo.

Não se pode confundir serviços "inbound" (a rede interna provendo algum serviço) e serviços "outbound" (o cliente está na rede interna e o servidor na Internet) com pacotes "inbound" (pacotes que chegam na rede interna) e pacotes "outbound" (pacotes que saem da rede interna).

É importante que o "packet filter" tenha facilidades de filtragem por interfaces de rede. Ou seja, todas as interfaces disponíveis são submetidas às regras de filtragem, possibilitando que as regras sejam aplicadas considerando a interface na qual o pacote chega e a interface pela qual o pacote prossegue o seu caminho.

as regras sejam aplicadas considerando a interface na qual o pacote chega e a interface pela qual o pacote prossegue o seu caminho.

1.2 Regras de filtragem

No filtro de pacotes existe uma lista de regras de filtragem aplicadas para cada pacote que trafega pelo filtro. Esta lista de regras é elaborada segundo a política de segurança adotada, onde é estabelecido o que pode e o que não pode passar pelo filtro. A estrutura básica de uma regra de filtragem é apresentada na FIGURA 1.

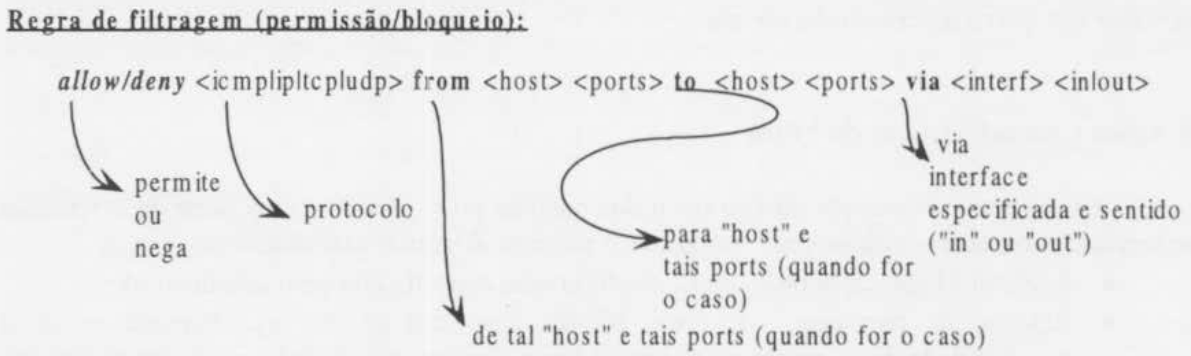


FIGURA 1 - Estrutura de uma regra de filtragem

Não existe um padrão de sintaxe de regras de filtragem. É aconselhável montar uma tabela com as possibilidades de pacotes para cada serviço a ser provido, e a partir dessa tabela escrever as regras de filtragem de acordo com a sintaxe do filtro utilizado.

O comportamento básico de um filtro de pacotes convencional é o seguinte[SIY 95] [SPO 96] [CHA 94]:

1. As regras de filtragem são armazenadas em uma lista na ordem em que foram especificadas.
2. Quando um pacote chega em uma determinada interface, os cabeçalhos do pacote são analizados.
3. Cada regra é aplicada ao pacote na ordem em que as regras estão armazenadas.
4. Se uma regra bloqueia a transmissão ou recepção do pacote, o pacote é bloqueado.
5. Se uma regra permite a transmissão ou recepção do pacote, o pacote é aceito para prosseguir.
6. Se um pacote não satisfaz qualquer regra ele é bloqueado (postura prudente).

Pelas regras 4 e 5 fica evidente que a ordem das regras de filtragem é de fundamental importância. Uma ordenação incorreta das regras pode acarretar em bloqueio de serviços válidos e em permissão de serviços que deveriam ser negados [CHA 95].

1.3 Vantagens e Desvantagens

Algumas vantagens dos "packet filters" são [CHA 95]:

- O filtro pode ajudar a proteger toda uma rede, principalmente caso ele seja o único ponto de conexão da rede interna à Internet;
- A filtragem de pacotes é transparente e não requer conhecimento nem cooperação dos usuários;

- Há uma série de produtos comerciais e "freeware" disponíveis;
- A maioria dos roteadores (CISCO, por exemplo) apresentam recursos de filtragem de pacotes.

Algumas desvantagens são:

- Alguns protocolos não são bem adaptados para a filtragem;
- Algumas políticas não podem ser aplicadas somente com a filtragem de pacotes.

Quando se aplica alguma restrição em algum protocolo de mais alto nível, através de números de "ports", espera-se que nada além do próprio serviço esteja associado àquela "port"; entretanto, um usuário interno mal intencionados pode subverter este tipo de controle substituindo o servidor por outro desenvolvido por ele.

1.4 Ações e características do Filtro

A máquina encarregada da filtragem dos pacotes pode executar uma série de atividades que servem, entre outras coisas, para monitorar o sistema. Algumas atividades são:

- Realizar "logs" de acordo com a configuração especificada pelo administrador.
- Retorno de mensagens de erros ICMP: caso um pacote seja barrado existe a possibilidade de se enviar ao endereço fonte alguma mensagem com o código de erro ICMP do tipo "host unreachable" ou "host administratively unreachable". Entretanto, tais mensagens, além de causar um "overhead", podem fornecer algumas informações sobre o filtro ao atacante, pois dessa forma ele poderia descobrir quais os protocolos que são barrados e quais estão disponíveis; portanto, recomenda-se que não se retorne nenhum código ICMP de erro para hosts na rede externa.

Eis algumas características altamente desejáveis a fim de que se possa realizar uma filtragem de pacotes bem apurada [CHA 95]:

- Ter um bom desempenho na filtragem dos pacotes;
- Pode ser um roteador dedicado ou um computador de propósito geral;
- Permitir uma especificação de regras de forma simples;
- Permitir regras baseadas em qualquer cabeçalho ou critério "meta packet" (por exemplo, em qual interface o pacote chegou ou está saindo);
- Aplicar as regras na ordem especificada;
- Aplicar as regras separadamente para pacotes que chegam e partem em e de cada interface de rede;
- Registrar informações sobre pacotes aceitos e rejeitados;
- Ter capacidade de teste e validação.

2 Configuração do Filtro

A configuração do filtro é uma atividade que deve ser realizada cuidadosamente. Caso contrário, pacotes que deveriam ser barrados não o são e pacotes que poderiam passar pelo filtro não o fazem. Cada novo serviço disponível na rede requer que novas regras de filtragem sejam adicionadas.

Para se elaborar as regras de filtragem é necessário que se saiba quais as informações relevantes que estão presentes em cada um dos pacotes de uma sessão. Definida a política de segurança, basta escrever as regras de filtragem de acordo com a sintaxe do filtro e ajustar os endereços dos "hosts" envolvidos.

Por exemplo, o SMTP (Simple Mail Transfer Protocol) é um serviço baseado no TCP, o servidor atende na "port" 25 e os clientes utilizam valores de "port" acima de 1023. As características dos pacotes do serviço SMTP são as seguintes:

Direção	End. Fonte	End. Destino	Protocolo	Port fonte	Port dest.	flag ACK	Observação
Entrando	Externo	Interno	TCP	> 1023	25	*	Mail chegando (sender to recipient)
Saindo	Interno	Externo	TCP	25	> 1023	Sim	Mail chegando (recipient to sender)
Saindo	Interno	Externo	TCP	> 1023	25	*	Mail saindo (sender to recipient)
Entrando	Externo	Interno	TCP	25	> 1023	Sim	Mail saindo (recipient to sender)

* ACK não está marcado no primeiro pacote deste tipo (estabelecimento de conexão) mas estará marcado nos demais.

As possíveis regras de filtragem seriam as seguintes (considerando que o endereço do servidor de correio eletrônico é 143.54.83.2):

1. Allow tcp from any >1023 to 143.54.83.2 25
2. Allow tcp from 143.54.83.2 25 to any > 1023 ACK
3. Allow tcp from 143.54.83.2 > 1023 to any 25
4. Allow tcp from any 25 to 143.54.83.2 > 1023 ACK

3 Desenvolvimento do Filtro

Esse tópico aborda aspectos do "session filter" e apresenta os pontos mais relevantes no projeto e implementação do filtro de sessões proposto. Essa implementação é o produto de uma coleta de informações sobre filtragem de pacotes, resumidamente apresentadas nos tópicos anteriores, e a escolha de uma plataforma viável ao desenvolvimento do filtro.

3.1 "Session Filter"

Na filtragem convencional cada pacote que transita pelo filtro deve ser verificado junto às regras de filtragem. A filtragem por sessão permite que se valide um pacote baseado na sessão a qual ele pertence. Dessa forma, apenas o pacote de inicialização da conexão precisa ser verificado com as regras de filtragem, os pacotes seguintes são verificados junto a uma "cache" de sessões ativas [AMO 96]: caso exista uma sessão correspondente o pacote é liberado; caso contrário, o pacote é descartado (Fig. 2).

Para que se mantenha controle sobre as sessões ativas é necessário que o protocolo seja orientado a conexão; ou seja, deve ser possível determinar exatamente quando uma nova sessão começa e termina.

No ambiente TCP/IP o protocolo TCP é orientado a conexão; portanto, existe um procedimento de estabelecimento de conexão. No TCP esse procedimento é conhecido por "three-way handshake" [COM 91](FIGURA 3a). O pacote de solicitação de conexão apresenta apenas o "bit" SYN marcado. O servidor, após receber a solicitação, responde com um pacote que apresenta os "bits" SYN e ACK marcados. O cliente, ao receber esta confirmação do servidor, envia um pacote de reconhecimento que apresenta apenas o ACK marcado. Está então estabelecida a conexão. Os demais pacotes, até o término da sessão, apresentam o ACK marcado.

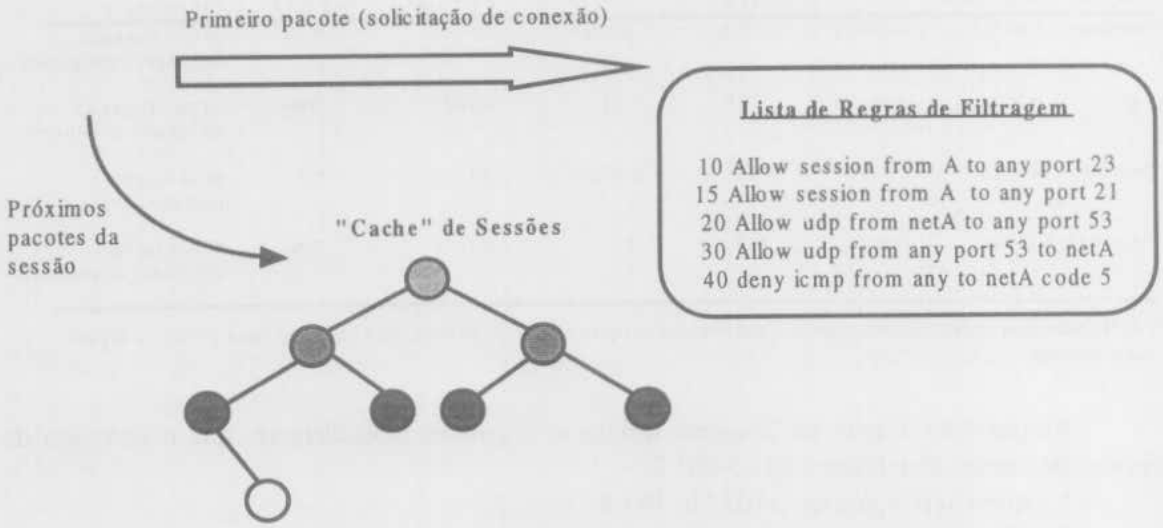


FIGURA 2 - Estrutura básica de um "Session Filter"

O procedimento de encerramento de uma conexão TCP é o seguinte (Fig. 3b): o "site" que deseja encerrar a conexão envia um pacote com o "bit" FIN marcado, o outro "site" após receber este pacote envia a confirmação (ACK marcado) de solicitação de encerramento de conexão e, após um certo atraso, envia também um pacote com o FIN marcado e fica aguardando a confirmação do outro "site", quando este confirma e o outro recebe é encerrada a conexão

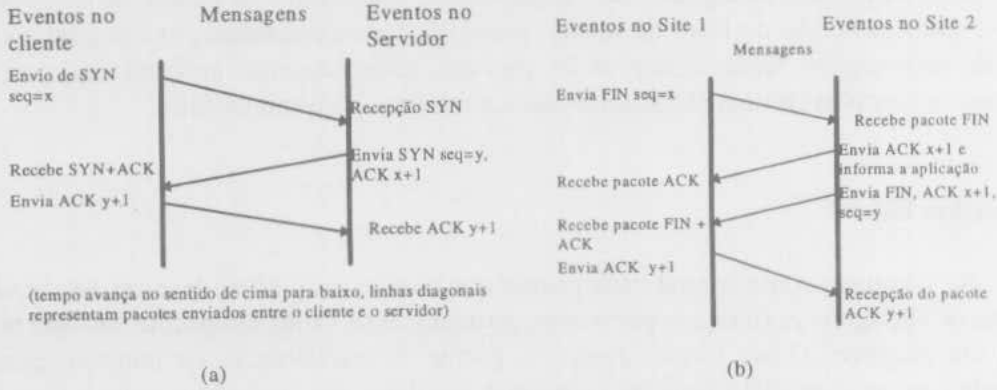


FIGURA 3 - Sequência de mensagens no procedimento de estabelecimento (a) e encerramento(b) de conexão no TCP

3.2 Identificação

Uma conexão TCP é identificada pelos pares "endereço IP fonte e destino" e "port fonte e destino" (Fig. 4). Esta identificação é única em toda a rede; ou seja, não há outra conexão entre os dois hosts envolvidos com o mesmo par de "port fonte e destino".

Além dessas informações, a "cache" deve também manter o estado da conexão dos dois "hosts" envolvidos a fim de que a sessão seja monitorada corretamente. A máquina de estados finitos do protocolo TCP [COM 91] é empregada para controlar o estado da conexão (Fig. 5). Para eliminar conexões terminadas de forma anormal, deve-se adotar um "timeout" para que estas conexões sejam eliminadas da "cache".

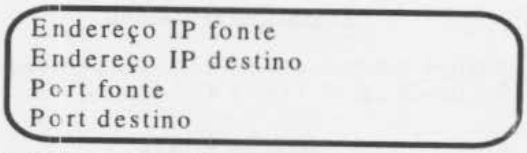
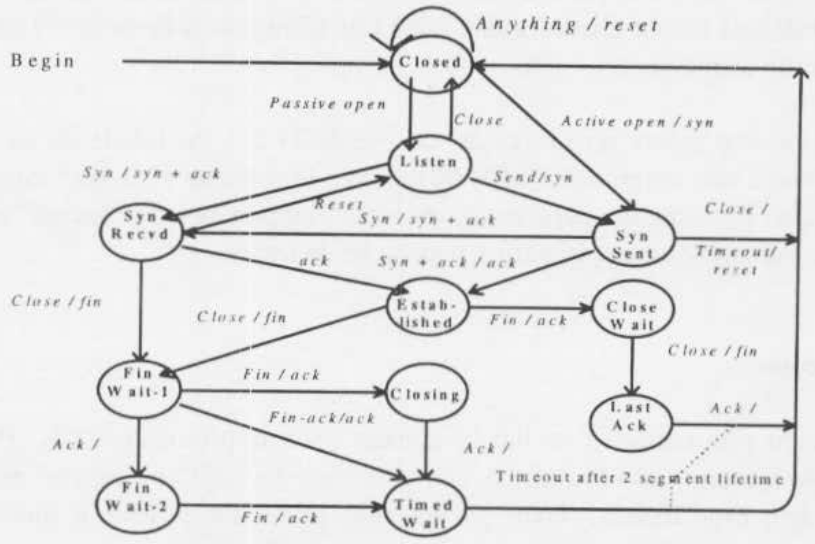


FIGURA 4 - Identificação de uma Sessão TCP

3.3 Desempenho

Uma das vantagens do filtro de sessões é o ganho em desempenho em relação a um filtro convencional. Pode-se afirmar isso em se considerando o fato de cada pacote ser analisado frente as regras de filtragem em um filtro convencional. Portanto, em média espera-se que o "overhead" acarretado pelo filtro de sessão seja consideravelmente menor. O ganho em desempenho depende do número de regras de filtragem e da quantidade de sessões presentes na "cache".

Num filtro convencional é necessário pelo menos duas regras de filtragem para cada serviço baseado no TCP (uma regra para o fluxo de pacotes em cada sentido), enquanto que é necessária apenas uma regra de filtragem para o filtro de sessão (Fig. 6). Portanto, também se diminui o tempo de pesquisa pelas regras de filtragem já que o número destas é reduzido.



(Cada ponto final começa no estado *closed*. Rótulos nas transições mostram a entrada (*input*) que causou a transição seguido pela saída (*output*) se alguma.)

FIGURA 5 - A máquina de estados finitos do protocolo TCP

3.4 Monitoramento

Outra vantagem do filtro de sessões é a possibilidade de monitoramento das sessões ativas. Ou seja, todas as sessões ativas entre a rede interna e a rede externa estão registradas na "cache" de sessões. Pode-se, dessa forma, obter estatísticas mais apuradas sobre as sessões que trafegam pelo "gateway": quantidade de pacotes, bytes, tempo médio de filtragem de cada pacote, "hosts" envolvidos, "ports" envolvidas, "timestamp" do último pacote.

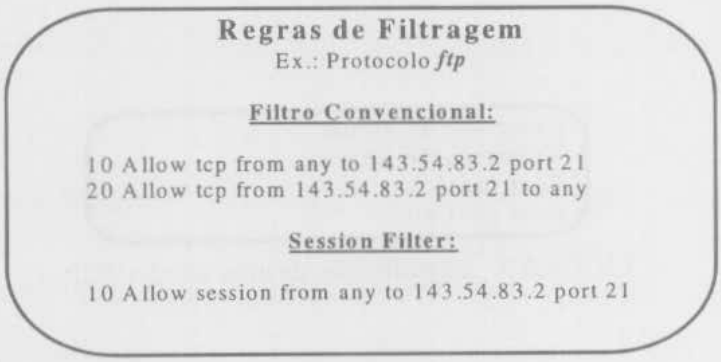


FIGURA 6 - Regras de filtragem (convencional versus session filter)

4 Projeto e implementação do Filtro

O filtro foi desenvolvido na plataforma FreeBSD (Unix para a arquitetura Intel 80x86, compatível com o Unix BSD) [LEH 96]. Essa plataforma foi escolhida porque se trata de um sistema operacional "freeware" e todo o código fonte do sistema operacional está disponível.

O ponto inicial de partida foi o filtro de pacotes disponível no FreeBSD, o "ip_fw". Ele permitiu que se estudasse como o filtro se encaixa no "kernel" do sistema operacional. Para inserir, remover, listar, e realizar outras tarefas, é utilizado o utilitário "ipfw" também disponível como "freeware". Este utilitário foi adaptado ao filtro de sessão desenvolvido, desta forma se mantém compatibilidade com a estrutura das regras de filtragem já disponíveis sendo necessária apenas uma extensão à sintaxe das regras.

Outro fator importante na utilização do FreeBSD é a facilidade de se personalizar o "Kernel", bem como a sua compilação [LEH 96] . Isto é importante visto que, como o filtro é um acréscimo ao núcleo do sistema operacional, diversas compilações do "kernel" são necessárias durante o processo de implementação para a realização de testes.

4.1 Fluxo de Pacotes

A filtragem por sessão é realizada apenas para o protocolo TCP. Para os demais protocolos (icmp, ip, udp) a filtragem é convencional; ou seja, as regras de filtragem são aplicadas na ordem especificada. Todo pacote cujo protocolo é TCP é direcionado para o "Session Filter", os demais fluem pelo filtro convencional (Fig. 7).

4.2 Filtro convencional

As regras de filtragem são armazenadas numa lista duplamente encadeada respeitando a ordem que foi estabelecida. Cada regra tem um número. A estrutura de dados adotada para conter uma regra de filtragem apresenta os seguintes dados:

- Número da regra de filtragem;
- Número de bytes e de pacotes que cumpriram a regra;
- Endereço IP fonte e destino;
- Máscara para o endereço fonte e destino;
- Especificação da interface;

- Opções do protocolo IP;
- "Flags" da regra;
- "Ports" fonte e destino;
- Tipos ICMP;
- Timestamp;

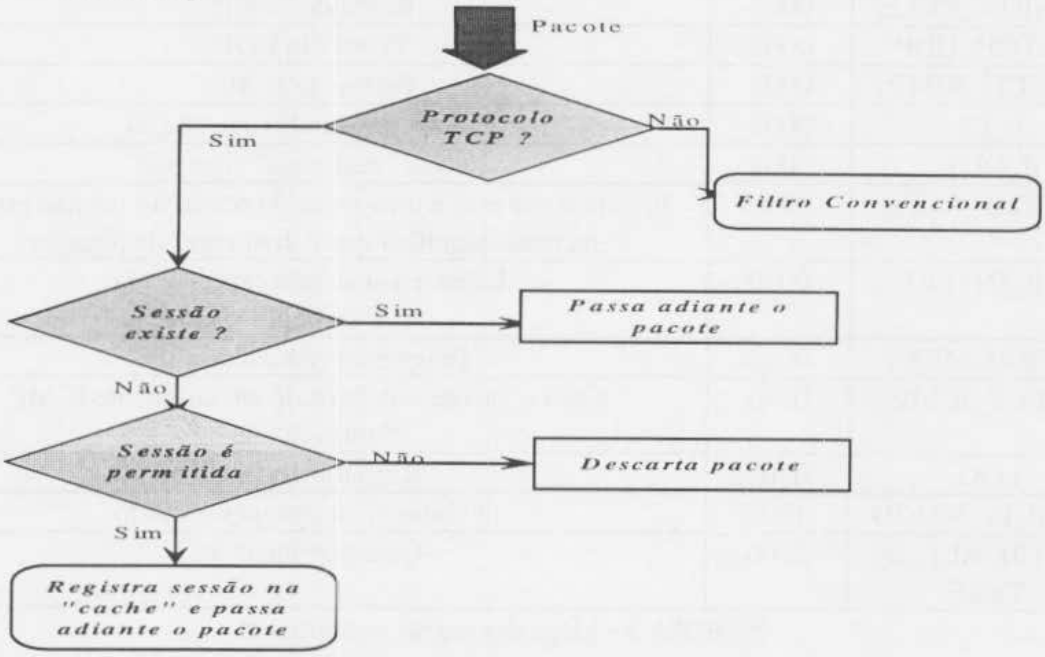


FIGURA 7 - Fluxo de pacotes TCP

A máscara dos endereços IP fonte e destino permite que se aplique uma regra a uma rede (por exemplo: endereço IP 143.54.83.0 e máscara 255.255.255.0 significa que a regra deve ser aplicada a qualquer "host" da sub-rede especificada; ou seja, 143.54.83).

A interface para a qual a regra se aplica também pode ser especificada utilizando o nome simbólico (ex.: ed0) ou o endereço IP correspondente a interface.

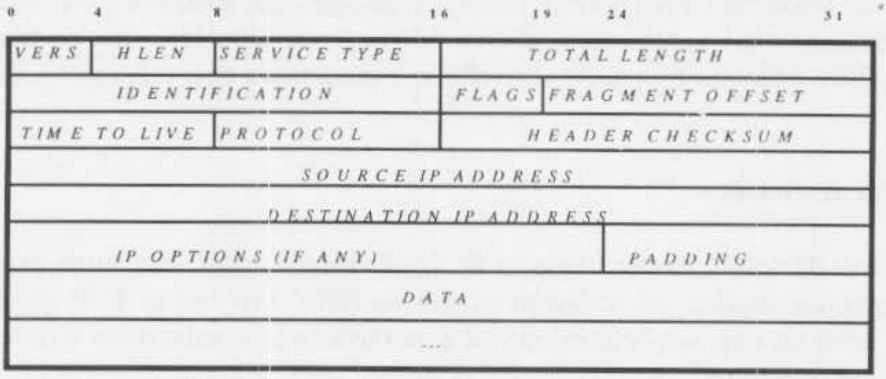


FIGURA 8 - Formato de um datagrama IP

O campo de opções do protocolo IP (Fig. 8) permite que se especifique quais as opções a serem consideradas. Uma opção que deveria ser negada é o "IP source routing" que representa uma grave ameaça a segurança, pois permite ao atacante descrever no próprio pacote qual a rota que o pacote deve seguir, permitindo desta forma que o atacante utilize um endereço fonte qualquer (geralmente o de uma máquina considerada confiável pela vítima) e garantindo que a resposta será enviada para a rota especificada no pacote. As principais opções são: "Loose source routing" (utilizado para rotear um datagrama ao longo de um caminho especificado, "Strict source routing" (idem a opção anterior), "Record packet route" (utilizada para traçar uma rota) e "Internet timestamp" (utilizada para registrar os timestamps ao longo de uma rota).

Os "flags" da regra contém informações acerca de quais ações devem ser tomadas pela regra. Os "flags" adotados são os seguintes:

Flag	valor (hex)	descrição
MATCH_ALL	0000	Aplicar a qualquer protocolo
SESSION_RULE	0001	Regra de Sessão
MATCH_UDP	0002	Protocolo UDP
MATCH_ICMP	0003	Protocolo ICMP
P_IN	0004	Pacote "inbound" (que chega)
P_OUT	0008	Pacote "outbound" (que sai)
ACCEPT_RULE	0010	Significa que esta é uma regra de aceitação (se não estiver marcado significa que é uma regra de rejeição)
USE_INTERF	0020	Utilizar a interface especificada ("through")
IP_FRAGMENT	0040	Fragmentos de pacotes IP
REPLY_ICMP	0080	Caso o pacote seja barrado enviar pacote ICMP "unreachable"
LOG	0200	Registrar no "log"
ICMP_IS_VALID	1000	O "bitmap" *icmptypes* é válido
MATCH_ALL_INTERF	2000	Qualquer interface

FIGURA 9 - Flags das regras de filtragem

Pode-se especificar até quatro "ports" tanto para a fonte como para o destino ou então até dois intervalos (ex: 1023-65535, significando valores de 1023 a 65535).

Foi adotada a postura prudente; ou seja, "tudo aquilo que não é expressamente permitido é proibido". Em outras palavras: "se não há uma regra de filtragem que permita a passagem do pacote este é descartado ('dropped')". O filtro "freeware" *ip_fw* garante isso colocando como última regra de filtragem (regra número 65535) uma regra que bloqueia todo e qualquer pacote. Na implementação aqui apresentada não se adotou esta estratégia; ao contrário, a postura "default" do filtro é bloquear o pacote caso não seja encontrada uma regra que o permita.

4.3 Filtragem de Sessões

O filtro de sessão requer um grau de detalhamento maior. Em suma existe um registro para cada uma das sessões que trafegam através do filtro. Um pacote TCP que não pertença a uma sessão ativa ou cujo estabelecimento de conexão não seja autorizado é descartado e não é enviado adiante pelo filtro.

Para justificar o melhor desempenho desse esquema de filtragem, o acesso à "cache" de sessões deve ser mais rápido que, em média, o acesso ao filtro convencional para cada pacote da sessão. Portanto, a "cache" deve ter uma estrutura que permita a inserção, consulta e remoção de nodos com um número reduzido de operações. Para tanto, é utilizado uma estrutura em "Árvore AVL" (Fig. 10) que apresenta uma complexidade no tempo de ordem $O(\log n)$ na execução das três operações citadas. Essa ordem de complexidade se deve sobretudo ao fato desse tipo de árvore ser quase que totalmente equilibrada; ou seja, por exemplo, com 1024 nodos ela tem uma profundidade de no máximo 10 nodos. Esse equilíbrio é garantido porque não pode existir

nenhum nodo que tenha uma diferença de "altura" maior do que 1 (um) entre os seu ramos esquerdo e direito. Toda a vez que é feita uma inserção ou remoção de um nodo isso é verificado. Existem 6 casos de excessões na inserção e 10 casos na remoção para os quais é necessário fazer um rearranjo (rotações) da árvore para restabelecer o equilíbrio [TER 91, STA 95].

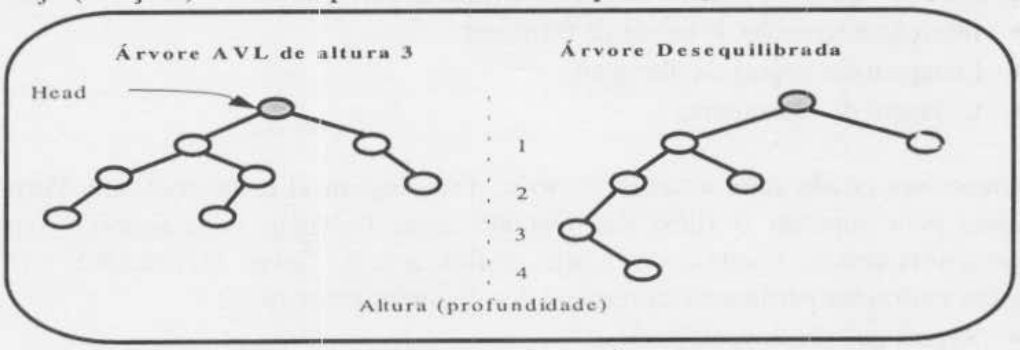


FIGURA 10 - Exemplo de "Árvore AVL"

Para a manipulação dos nodos da árvore é utilizada como chave primária os pares de endereço IP e valores de "port". Para formar uma chave única os dois pares são agrupados da seguinte forma: o menor endereço IP é a parte menos significativa do identificador relativo ao endereço e o outro endereço forma a parte mais significativa; da mesma forma para o par de "ports" (Fig. 11). Seguindo essa regra de formação, obtém-se um valor único.

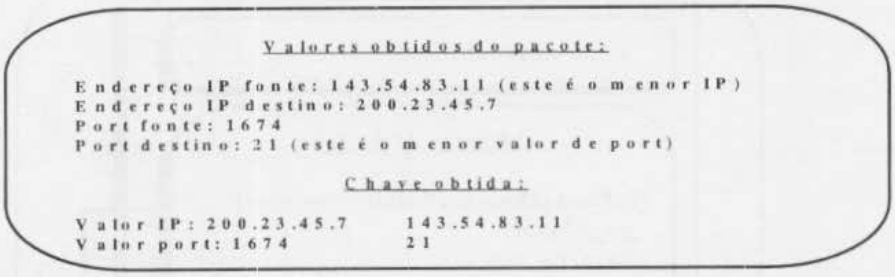


FIGURA 11 - Obtenção da chave única para identificação da Sessão

Mesmo após a detecção dos últimos pacotes de uma sessão, deve-se manter o registro da sessão na "cache" durante um certo tempo (Veja novamente o diagrama de estados na Fig. 5). Caso contrário, pode ocorrer o extravio de algum pacote e a sua consequente retransmissão, e se não houver mais registro da sessão na "cache" o pacote válido seria descartado.

4.4 O Filtro e o Kernel do FreeBSD

O filtro é incorporado ao módulo de rede do "kernel" do sistema operacional FreeBSD (Fig. 12). O filtro *ip_fw* distribuído junto com o sistema operacional foi a referência básica para o desenvolvimento do "session filter".

Quando o sistema de rede é inicializado (no "boot") é chamada uma rotina de inicialização do filtro. Cada pacote que flui pelo módulo de rede, após o processamento normal executado, é enviado ao filtro para verificar sua permissão. Caso o filtro responda negativamente (valor zero) o pacote é descartado; de outra forma, o pacote é encaminhado para o seu destino.

4.5 O utilitário "sipfw"

O utilitário "sipfw" é a versão alterada do "ipfw" desenvolvido para interagir com o filtro de pacotes. O "ipfw" possibilita as seguintes facilidades:

- Inserção e remoção de regras de filtragem;
- Listagem das regras de filtragem;
- Listagem de "accounting";

Como fora citado anteriormente, o "ipfw" está disponível como freeware. Portanto, ele foi adaptado para suportar o filtro desenvolvido nesse trabalho. Para suportar regras que especifiquem uma sessão, a sintaxe das regras de filtragem do "ipfw" foi estendida no "sipfw". As alterações realizadas permitem o acréscimo dos seguintes recursos:

- Regras que descrevem sessões;
- Listagem das sessões presentes na "cache";
- Estatísticas (para cada sessão ativa): número de pacotes, número de bytes, tempo médio de processamento de cada pacote da sessão (em microsegundos) e "timestamp" do último pacote;
- Alteração no valor do "timeout" de exclusão de um elemento da "cache".

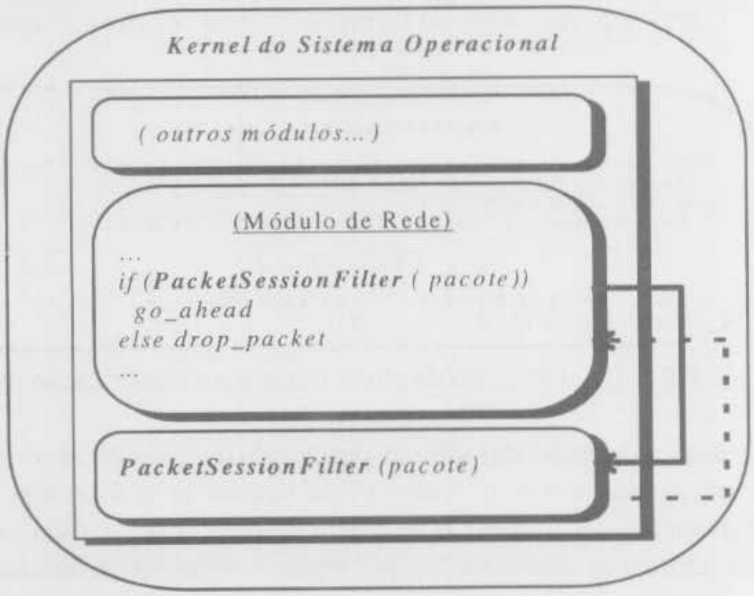


FIGURA 12 - Incorporação do Filtro ao Kernel do Sistema Operacional

O "sipfw" se comunica com o filtro de pacotes via "raw sockets". Este tipo de sockets só pode ser utilizado pelo superusuário ("root"), garantindo que as alterações são realizadas por pessoal autorizado.

A mensagem é enviada e armazenada em um "buffer". O código da mensagem é utilizado para saber qual o tipo de operação (inserção, remoção, listagem das regras, etc) deve ser executada. A rotina de tratamento do filtro de pacotes é ativada e a operação é executada .

O monitoramente permite ao administrador acompanhar as sessões existentes entre a rede interna e a Internet. Um exemplo dos resultados fornecidos pelo "sipfw" é apresentado na Figura 13. Cada entrada contém informações referentes as máquinas envolvidas, "ports", número de pacotes, número de bytes, tempo médio de processamento (em microssegundos) de filtragem dos pacotes da referida sessão e "timestamp" do último pacote registrado.

```
Information about sessions currently in the Session Cache:

ijui.inf.ufrgs.br corvete.inf.ufrgs.br          1111 http       120 (packets) 11234 (bytes) time = 8
-- Wed Aug 13 13:31:24 1997
porsche-gw.inf.ufrgs.br caracol-gw.inf.ufrgs.br      1027 ftp        20 (packets) 1286 (bytes)
time = 13 -- Wed Aug 13 13:29:14 1997
porsche-gw.inf.ufrgs.br cepheus.inf.ufrgs.br         1028 telnet     65 (packets) 2999 (bytes)
time = 13 -- Wed Aug 13 13:29:57 1997
porsche-gw.inf.ufrgs.br caracol-gw.inf.ufrgs.br      40001 ftp-data  8 (packets) 763 (bytes) time
= 12 -- Wed Aug 13 13:31:59 1997
porsche-gw.inf.ufrgs.br rigel.inf.ufrgs.br    1715 finger     10 (packets) 563 (bytes) time = 14 -
- Wed Aug 13 13:32:39 1997
```

FIGURA 13 - Informações fornecidas pelo "sipfw"

5 Avaliação de desempenho

O objetivo dos testes realizados é comparar o desempenho do filtro de sessão com relação ao filtro convencional *ip_fw*. Isto é feito através da medida do tempo médio de filtragem dos pacotes em cada um dos filtros. O tempo é obtido utilizando a função "microtime", disponível na plataforma FreeBSD, que apresenta uma definição na ordem de microssegundos.

5.1 Plataforma de Testes

É importante descrever a plataforma de testes porque tem influência direta no desempenho do filtro. Os testes com os dois filtros foram realizados na mesma máquina, com a mesma configuração. As principais características do "gateway" onde o filtro estava ativo são as seguintes:

- Processador Pentium 133 MHz;
- 32 Mbytes de memória RAM;
- Interfaces de Rede Ethernet.

De fato, como o filtro faz parte do "Kernel" e está sempre na memória, os recursos mais importantes para um melhor desempenho são o processador e a memória principal.

5.2 Cenários

A filtragem se dá a nível do Kernel do sistema operacional e é realizada atomicamente; portanto, a filtragem dos pacotes referentes a diferentes serviços pode ser mensurada sem que ocorra interferência.

O fator mais importante na escolha do serviço para testes é que este seja baseado no protocolo TCP a fim de permitir comparações de desempenho entre o filtro convencional e o filtro de sessão. Na medida do "overhead" do filtro não influi o serviço em questão e sim a localização da regra na lista de regras. O serviço escolhido para a realização dos testes é o "finger".

A lista de regras de filtragem (Fig. 14) foi obtida considerando os serviços utilizados na sub-rede em teste. As quatro primeiras regras são relativas ao protocolo ICMP: a primeira permite que pacotes ICMP fluam da rede interna (endereço 143.54.83.0) para qualquer outra máquina, as duas regras seguintes impedem que ingressem na sub-rede pacotes ICMP do tipo "redirect" (código 5) e "echo request" (código 8) e a quarta permite que demais tipos de pacotes ICMP ingressem na rede. As demais regras se referem a serviços baseados nos protocolos UDP e TCP, os quais são os seguintes: Autenticação (port 113), telnet (23), rlogin (513), nntp (119),

http (80), smtp (25), ftp (20 e 21), gopher (70), whois (43), talk (517,518, 1023-65535), finger (79), dns (53), rip (520), nfs (2049), lpr (515). Vale a pena salientar que o número de regras é ampliado quando o serviço é "inbound" (servidor está na sub-rede interna) e "outbound" (servidor é externo, na Internet), como por exemplo o serviço de telnet (2 regras de filtragem para cada uma das situações).

No filtro de sessão as regras de filtragem são diferentes para os serviços baseados no protocolo TCP (Fig. 15). Apenas uma regra é necessária para habilitar um serviço, a qual é consultada uma única vez a cada nova conexão. O número de regras reduziu de 56 (filtro convencional) para 37.

1 allow icmp from 143.54.83.0/24 to any
2 deny icmp from any to 143.54.83.0/24 icmptypes 8
3 deny icmp from any to 143.54.83.0/24 icmptypes 5
4 allow icmp from any to 143.54.83.0/24
5 allow tcp from 143.54.83.0/24 1023-65535 to any 113
6 allow tcp from any 113 to 143.54.83.0/24 1023-65535
7 allow tcp from any 1023-65535 to 143.54.83.0/24 113
8 allow tcp from 143.54.83.0/24 113 to any 1023-65535
9 allow tcp from 143.54.83.0/24 1023-65535 to any 23
10 allow tcp from any 23 to 143.54.83.0/24 1023-65535
11 allow tcp from any 1023-65535 to 143.54.83.0/24 23
12 allow tcp from 143.54.83.0/24 23 to any 1023-65535
13 allow tcp from 143.54.83.0/24 1023-65535 to any 513
14 allow tcp from any 513 to 143.54.83.0/24 1023-65535
15 allow tcp from 143.54.83.0/24 1023-65535 to any 119
16 allow tcp from any 119 to 143.54.83.0/24 1023-65535
17 allow tcp from 143.54.83.0/24 1023-65535 to any 80
18 allow tcp from any 80 to 143.54.83.0/24 1023-65535
19 allow tcp from 143.54.83.0/24 1023-65535 to any 25
20 allow tcp from any 25 to 143.54.83.0/24 1023-65535
21 allow tcp from any 1023-65535 to 143.54.83.0/24 25
22 allow tcp from 143.54.83.0/24 25 to any 1023-65535
23 allow tcp from 143.54.83.0/24 1023-65535 to any 70
24 allow tcp from any 70 to 143.54.83.0/24 1023-65535
25 allow tcp from any 1023-65535 to 143.54.83.0/24 43
26 allow tcp from 143.54.83.0/24 43 to any 1023-65535
27 allow udp from 143.54.83.0/24 1023-65535 to any 517,518
28 allow udp from any 517,518 to 143.54.83.0/24 1023-65535
29 allow udp from any 1023-65535 to 143.54.83.0/24 517,518
30 allow udp from 143.54.83.0/24 517,518 to any 1023-65535
31 allow tcp from 143.54.83.0/24 1023-65535 to any 1023-65535
32 allow tcp from any 1023-65535 to 143.54.83.0/24 1023-65535
33 allow udp from 143.54.83.0/24 1023-65535 to any 53
34 allow udp from any 53 to 143.54.83.0/24 1023-65535
35 allow tcp from 143.54.83.0/24 1023-65535 to any 53
36 allow tcp from any 53 to 143.54.83.0/24 1023-65535
37 allow udp from 143.54.83.0/24 1023-65535 to any 520
38 allow udp from any 520 to 143.54.83.0/24 1023-65535
39 allow udp from any 1023-65535 to 143.54.83.0/24 520
40 allow udp from any 143.54.83.0/24 to any 1023-65535
41 allow udp from 143.54.83.0/24 520 to any 520
42 allow udp from any 520 to 143.54.83.0/24 520
43 allow tcp from 143.54.83.0/24 1023-65535 to 143.54.0.0/16 2049
44 allow tcp from 143.54.0.0/16 2049 to 143.54.83.0/24 1023-65535
45 allow tcp from 143.54.83.0/24 to any 515
46 allow tcp from any 515 to 143.54.83.0/24
47 allow tcp from 143.54.83.0/24 1023-65535 to any 21
48 allow tcp from any 21 to 143.54.83.0/24 1023-65535
49 allow tcp from any 20 to 143.54.83.0/24 1023-65535
50 allow tcp from 143.54.83.0/24 1023-65535 to any 20
51 allow tcp from any 1023-65535 to 143.54.83.0/24 21
52 allow tcp from 143.54.83.0/24 21 to any 1023-65535
53 allow tcp from any 1023-65535 to 143.54.83.0/24 20
54 allow tcp from 143.54.83.0/24 20 to any 1023-65535
55 allow tcp from 143.54.83.0/24 1023-65535 to any 79
56 allow tcp from any 79 to 143.54.83.0/24 1023-65535

FIGURA 14 - Regras de filtragem para o filtro convencional

O tempo médio de filtragem de cada pacote depende da posição da regra que permite o fluxo do pacote. Isto é relevante para o filtro convencional, já no filtro de sessão o mais relevante é o número de sessões registradas na "cache". O tempo médio de acesso a "cache" é proporcional a $\log n$ onde n é o número de nodos ("sessões") na cache.

Em cada cenário de testes a variação na quantidade de sessões na "cache" é garantida através de múltiplos testes tendo cada um deles um número superior de sessões simultâneas. A regra de filtragem que permite o serviço testado também é deslocada para averiguar qual a alteração de comportamento. No pior caso, a regra é a última; no melhor caso, ela é a primeira e, no caso médio, ela é a regra intermediária. Ou seja, com relação as regras apresentadas na Figura 14, os três cenários apresentam as seguintes posições: melhor caso, regras 1 e 2; caso médio, regras 27 e 28 e, pior caso, regras 55 e 56.

1. allow icmp from 1435483O/24 to any
2. deny icmp from any to 1435483O/24 icmptypes 8
3. deny icmp from any to 1435483O/24 icmptypes 5
4. allow icmp from any to 1435483O/24
5. allow session from 1435483O/24 900-65535 to any 113
6. allow session from any 900-65535 to 1435483O/24 113
7. allow session from 1435483O/24 900-65535 to any 23
8. allow session from any 900-65535 to 1435483O/24 23
9. allow session from 1435483O/24 900-65535 to any 513
10. allow session from 1435483O/24 900-65535 to any 119
11. allow session from 1435483O/24 900-65535 to any 80
12. allow session from 1435483O/24 900-65535 to any 25
13. allow session from any 900-65535 to 1435483O/24 25
14. allow session from 1435483O/24 900-65535 to any 70
15. allow session from 1435483O/24 900-65535 to any 43
16. allow udp from 1435483O/24 900-65535 to any 517,518
17. allow udp from any 517,518 to 1435483O/24 900-65535
18. allow udp from any 900-65535 to 1435483O/24 517,518
19. allow udp from 1435483O/24 517,518 to any 900-65535
20. allow session from 1435483O/24 900-65535 to any 900-65535
21. allow session from any 900-65535 to 1435483O/24 900-65535
22. allow udp from 1435483O/24 900-65535 to any 53
23. allow udp from any 53 to 1435483O/24 900-65535
24. allow session from 1435483O/24 900-65535 to any 53
25. allow udp from 1435483O/24 900-65535 to any 520
26. allow udp from any 520 to 1435483O/24 900-65535
27. allow udp from any 900-65535 to 1435483O/24 520
28. allow udp from any 1435483O/24 to any 900-65535
29. allow udp from 1435483O/24 520 to any 520
30. allow udp from any 520 to 1435483O/24 520
31. allow session from 1435483O/24 900-65535 to 1435400/16 2049
32. allow session from 1435483O/24 to any 515
33. allow session from 1435483O/24 900-65535 to any 21

FIGURA 15 - Regras de filtragem para o filtro de sessão

Foram realizadas um total de 59400 sessões finger. Para cada um dos filtros (convencional e sessão) se executou uma bateria de testes com 90, 360 e 540 sessões simultâneas. Os testes foram repetidos 10 vezes para cada um dos casos (melhor, médio e pior) nos dois filtros; ou seja, 10 vezes 90 sessões para cada uma das três diferentes listas de regras de filtragem em cada um dos filtros, idem para as outras quantidades de sessões.

Pode-se supor que no filtro de sessão a posição da regra de filtragem que habilita a passagem do pacote não tenha muita relevância visto que, apesar do primeiro pacote passar pelas regras de filtragem, o tempo dispendido na filtragem inicial disperse nos próximos tempos de acesso à "cache".

5.3 Resultados

5.3.1 Cenário 1: 90 sessões simultâneas

O primeiro cenário corresponde a 90 sessões simultâneas representando um total de 2700 sessões (900 para cada um dos casos) em cada um dos filtros. O comportamento do filtro convencional se apresentou como esperado (Fig. 16), o pior caso teve um tempo médio de 38,4 μs, o caso médio 20,4 μs e o melhor caso 5,2 μs. Ou seja, o tempo médio de filtragem dos pacotes é proporcional a posição ocupada pela regra na lista de regras. O tempo no caso médio (regras números 27 e 28) corresponde aproximadamente a metade do tempo no pior caso (regras 55 e 56).

FIGURA 16 - Filtro Convencional (90 sessões)

O filtro de sessão também apresentou um comportamento esperado (Fig. 17). Além das 90 sessões, 60 sessões de autenticação estiveram presentes na "cache". As sessões de autenticação foram inicializadas a partir das máquinas servidoras do "finger". A posição da regra não afetou significativamente o resultado. O tempo médio de filtragem apresentou um valor de 9,3 µs no pior caso, 8,9 µs no caso médio e 8,3 µs no melhor caso. Pode-se dizer que o tempo obtido representa praticamente o tempo médio de acesso à "cache" de sessões ativas.

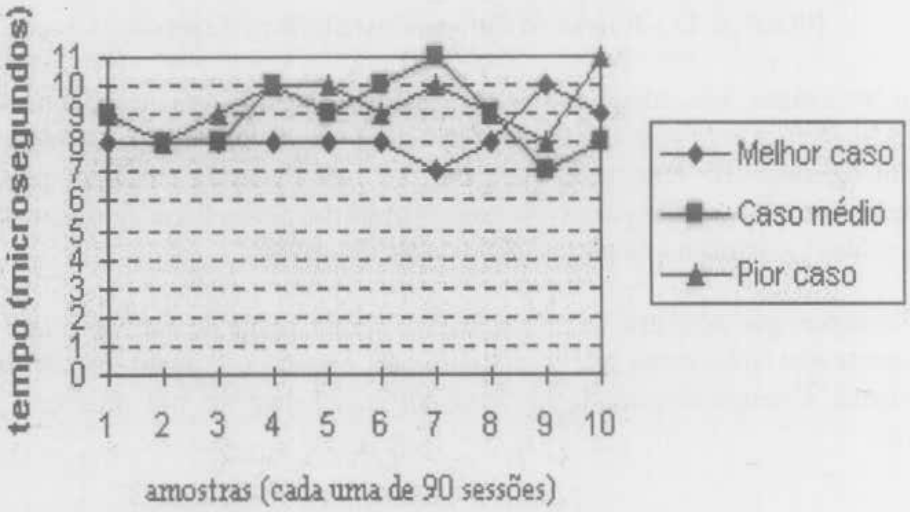


FIGURA 17 - Filtro de Sessão (90 sessões)

5.3.2 Cenário 2: 360 sessões simultâneas

Neste cenário 360 sessões simultâneas foram executadas 10 vezes em cada um dos casos, totalizando 10800 sessões em cada um dos filtros. O comportamento do filtro convencional (Fig. 18) foi praticamente idêntico ao registrado no primeiro cenário. Isto era esperado considerando que o número de sessões não tem influência porque o "overhead" de filtragem de cada pacote deve ser praticamente o mesmo para uma determinada posição da regra de filtragem. O pior caso apresentou um tempo médio de 39,2 µs, o caso médio 20,2 µs e o melhor caso 5,0 µs.

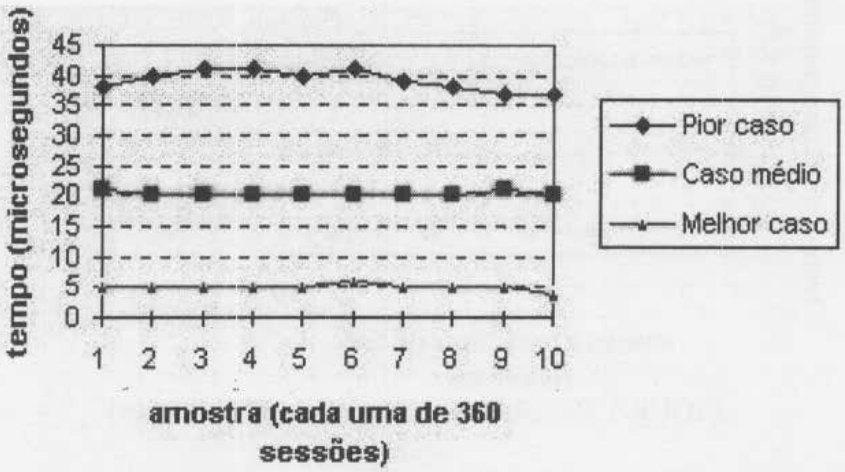


FIGURA 18 - Filtro Convencional (360 sessões)

Além das 360 sessões finger estiveram presentes na "cache" 120 sessões de autenticação. O filtro de sessão também teve um comportamento semelhante ao apresentado no primeiro cenário (Fig. 19). O tempo médio de filtragem apresentou um valor de 8,9 µs no pior caso, 9,1 µs no caso médio e 10 µs no melhor caso. Novamente fica explícito que a ordem da regra não tem influência significativa, visto que o "melhor" caso apresentou um valor maior que o "pior" caso. Atribui-se essa diferença entre os três casos a precisão na obtenção do tempo.

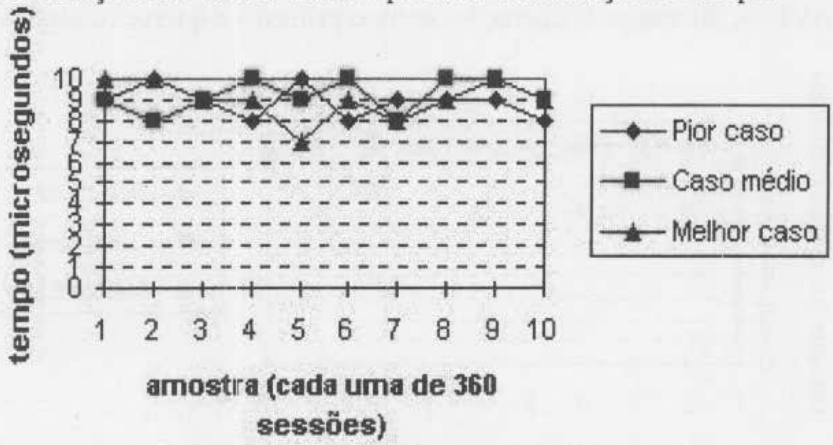


FIGURA 19 - Filtro de Sessão (360 sessões)

Considerando que havia 480 (360 finger + 120 auth) sessões registradas na "cache" o tempo médio de acesso foi proporcional a 8,9 (*"log 480"*) operações sobre a árvore AVL, enquanto que no primeiro cenário havia 150 (90 finger + 60 auth) sessões e o tempo médio de acesso foi proporcional a 7,2 operações. Ou seja, este pode ser o motivo da proximidade dos resultados nos dois cenários, sobretudo porque há uma diferença média de apenas 1,7 operações sobre a árvore AVL.

5.3.3 Cenário 3: 540 sessões simultâneas

Neste cenário foram executadas 540 sessões finger simultâneas, 10 vezes para cada um dos casos, totalizando 16200 sessões em cada um dos filtros. O tempo médio de filtragem apresentou um valor de 38,7 µs no pior caso, 20,5 µs no caso médio e 4,8 µs no melhor caso. O comportamento, esperado, foi semelhante ao ocorrido nos dois cenários anteriores (Fig. 20).

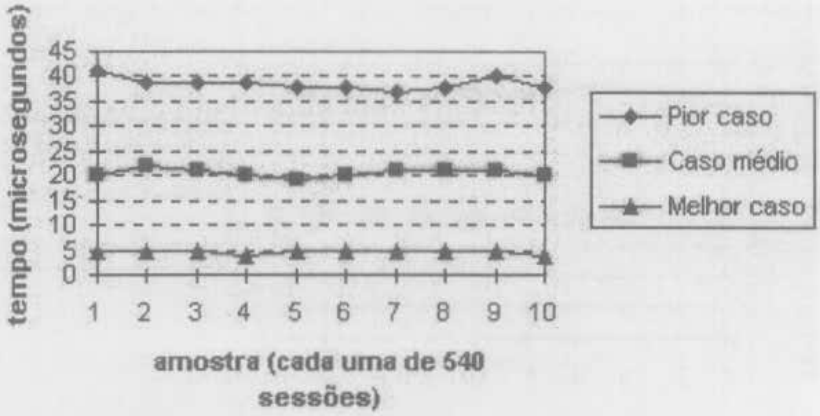


FIGURA 20 - Filtro Convencional (540 sesssões)

Além das 540 sessões finger estiveram presentes na "cache" 240 sessões de autenticação. O tempo médio de filtragem apresentou um valor de 9,5 μs no pior caso, 9,1 μs no caso médio e 9,2 μs no melhor caso (Fig. 21).

Considerando que havia 780 (540 finger + 240 auth) sessões registradas na "cache" o tempo médio de acesso foi proporcional a 9,6 (*"log 780"*) operações sobre a árvore AVL, enquanto que no segundo cenário havia 480 (360 finger + 120 auth) sessões e o tempo médio de acesso foi proporcional a 8,9 operações. Ou seja, este pode ser o motivo da proximidade dos resultados nos dois cenários, sobretudo porque há uma diferença média de apenas 0,7 operações sobre a árvore AVL. A diferença de operações entre o primeiro e o terceiro cenário é de 2,4.

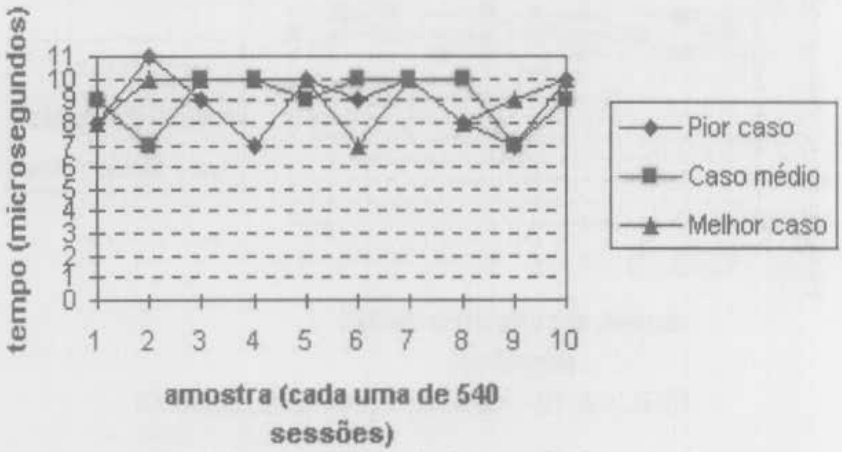


FIGURA 21 - Filtro de Sessão (540 sessões)

5.4 Análise Comparativa

Os três cenários de teste permitem afirmar o seguinte:

- A ordem da regra na lista de filtragem afeta significativamente o tempo médio de filtragem apenas no filtro convencional;
- O número de sessões ativas não afeta o tempo médio de filtragem no filtro convencional porque cada pacote é analisado sem considerar pacotes anteriores (sessão);
- O filtro de sessão representa, considerando a média entre as três posições da regra de filtragem, um ganho significativo na filtragem dos pacotes;
- A estrutura em "Árvore AVL" é uma boa solução para a "cache" de sessões ativas.

Obtendo o valor médio entre os três casos em cada um dos cenários, pode-se supor o ganho real na filtragem dos pacotes de todos os serviços TCP se considerarmos que todas as regras de filtragem das respectivas sessões presentes na lista são acessadas com a mesma frequência. Os valores médios nos três cenários são apresentados na tabela abaixo.

Tempo médio de filtragem

Filtro	Cenário 1	Cenário 2	Cenário 3
Filtro de Sessão	8,83μs	9,33μs	9,1μs
Convencional	21,33μs	21,46μs	21,33μs

6 Conclusões

A filtragem de pacotes representa um recurso importante na segurança de redes conectadas à Internet. Por exemplo, para limitar o tráfego entre dois segmentos de rede um filtro de pacotes já é suficiente. A postura prudente também pode ser aplicada facilmente com o recurso de filtragem.

A plataforma escolhida, o sistema operacional FreeBSD, é uma ótima alternativa para se explorar os recursos disponíveis no módulo de rede. Todo o código fonte do sistema operacional é distribuído livremente, garantindo dessa forma livre acesso para alterações. O filtro de pacotes disponível nessa plataforma (*ip_fw*) serviu como uma referência básica ao filtro desenvolvido. A compilação do Kernel foi realizada diversas vezes durante a fase de implementação do filtro. Novamente, a plataforma FreeBSD se mostrou fácil de compilar. Todavia, não se encontrou uma documentação específica para desenvolvimento a nível de Kernel. É necessário um esforço maior do que o desenvolvimento de uma aplicação, sobretudo porque qualquer "bug" a nível de kernel pode levar facilmente a um "crash" do sistema.

O utilitário "ipfw" foi alterado para suportar o filtro desenvolvido. A sintaxe das regras de filtragem foi mantida salvo algumas pequenas alterações, garantindo assim que quase todas as regras de filtragem utilizadas com o filtro convencional "ip_fw" possam ser reutilizadas. A possibilidade de monitoramento das sessões é um recurso valioso porque permite em qualquer instante listar quais as sessões ativas entre a rede interna e a Internet.

A avaliação de desempenho do filtro desenvolvido apresentou dados esperados. Os resultados obtidos após a execução de 59400 sessões "finger" permitem concluir que o filtro de sessões acarreta um "overhead" consideravelmente menor ao filtro convencional. A escolha de uma estrutura em "Árvore AVL" para a "cache" de sessões foi fundamental para o bom desempenho do filtro de sessões. O número de regras de filtragem também é reduzido com o acréscimo da regra de sessão ("session"). No filtro convencional são necessárias duas regras de filtragem para habilitar um serviço baseado no protocolo TCP; enquanto que no filtro de sessões apenas uma regra é necessária, considerando-se sobretudo que os pacotes seguintes à solicitação de conexão são verificados na "cache" de sessões ativas.

O melhor desempenho, a possibilidade de monitoramento e o grande número de serviços baseados no protocolo TCP justificam a utilização de um filtro de sessão. Deve-se observar que outras estruturas de dados podem ser testadas como "cache" a fim de se obter um desempenho ainda melhor.

Bibliografia

[AMO 96] AMOROSO, E.; SHARP, R. **PCWeek Intranet and Internet Firewall Strategies**. Emeryville, CA: Ziff Davis Press. 1996. 218p.

[CHA 94] CHAPMAN, D. B. **Network (in)security through IP packet filtering**. Mountain View, CA: Great Circles Associates, 1994. Disponível por http em www.greatcircle.com. (julho 1996).

[CHA 95] CHAPMAN, D. B.; ZWICKY, E. D. **Building Internet Firewalls**. Sebastopol, CA: O'Reilly & Associates, 1995.

[COM 91] COMER, D. E. **Internetworking with TCP/IP**: principles, protocols, and architecture. 2° ed. Englewood Cliffs: Prentice Hall, 1991. 547p.

[LEH 96] LEHEY, G. **Installing and Running FreeBSD**. Walnut Creek, CA: Walnut Creek CDROM, 1996. 232 p.

[SPO 96] SPOHN, M. A. **Internet Firewalls**: trabalho Individual. Porto Alegre: CPGCC da UFRGS, 1996. 69 f. (TI-554).

[SIY 95] SIYAN, K.; HARE, C. **Internet Firewalls and Network Security**. Indianapolis, Indiana: New Riders Publishing,1995.

[[STA 95] STANDISH, T. A. **Data Structures, algorithms, and software principles**. Reading: Addison Wesley. 1995.

[TER 91] TERADA, R. Desenvolvimento de Algoritmo e Estruturas de Dados. São Paulo: Editora McGraw-Hill, 1991.