Segurança - IPTables

Firewall IP Tables

Podemos afirmar que Firewall é um programa que detém autonomia concedida pelo próprio sistema para pré-determinar e disciplinar todo o tráfego existente entre o mesmo e outros hosts/redes.

Salvo situações onde o Firewall é um componente de soluções denominado "*Firewall-in-a-box*", onde neste caso, trata-se não somente de um software e sim de um agrupamento de componentes incluindo software e hardware, ambos projetados sob medida para compor soluções de controle perante o tráfego de um host/rede.

Desenvolvido pela Bell Labs em meados de 80 e sob encomenda da gigante das telecomunicações AT&T, o primeiro Firewall do mundo foi desenvolvido com o intuito de “filtrar” todos os pacotes que saíssem e entrassem na rede corporativa, de modo a manipulá-los de acordo com as especificações das regras previamente definidas pelos cientistas da Bell.

Atualmente um Firewall tem ainda a mesma funcionalidade porém foi acrescentadas algumas funcionalidades.

Segurança - IPTables

Firewall Filtro de Pacotes

O Linux tem implementado a filtragem de pacotes desde a primeira geração dos Kernels (1.X). Em meados de 1994 Alan Cox portou-o nativamente para o mesmo.

Esta classe de Firewall é responsável por filtrar todo o tráfego direcionado ao próprio host Firewall ou a rede que o mesmo isola, tal como todos os pacotes emitidos por ele ou por sua rede e isso ocorre mediante a análise de regras previamente inseridas pelo administrador do mesmo.

O Firewall filtro de pacotes possui a capacidade de analisar cabeçalhos (*Headers*) de pacotes enquanto os mesmos trafegam entre as redes. Mediante essa análise, que é fruto de uma extensa comparação de regras previamente adicionadas, pode-se decidir o destino de um pacote como um todo.

A filtragem pode então, deixar tal pacote trafegar livremente pela rede ou simplesmente parar a sua trajetória, ignorando-o por completo.

O mesmo é, sem dúvida, a classe mais utilizada de Firewall.

Firewall NAT

Um Firewall aplicado à classe NAT, a princípio, possui o objetivo de manipular a rota padrão de pacotes que atravessam o kernel do host Firewall aplicando-lhes o que conhecemos por “tradução de endereço”.

Isso lhe agrega diversas funcionalidades, como por exemplo, a de manipular o endereço de origem (SNAT) e destino (DNAT) dos pacotes, tal como realizar Masqueranding sobre conexões PPP, entre outras potencialidades.

Não há duvidas de que o Firewall NAT nos abre um amplo leque de possibilidades, porém seus conceitos vão um pouco além de mera filtragem de pacotes.

Um Firewall NAT pode, por exemplo, realizar o trabalho de um proxy de forma simples e eficiente, independente de IP, fixo ou dinâmico

Firewall Híbrido

Um Firewall Híbrido agrega a si tanto função de filtragem de pacotes quanto de NAT.

No Linux, as funções de Firewall são agregadas à própria arquitetura do Kernel, isso o torna, sem dúvida, muito superior em relação a seus concorrentes. Enquanto a maioria dos produtos Firewall pode ser definida como subsistema, o Linux possui a capacidade de transformar o Firewall no próprio.

O Linux utiliza-se de um recurso independente em termos de kernel para controlar e monitorar todo o tipo de fluxo de dados dentro de sua estrutura operacional. Mas não o faz sozinho, pois sua função deve ser a de trabalhar ao lado de processos e tarefas tão somente.

Para que o Kernel pudesse então controlar seu próprio fluxo interno lhe fora agregada uma ferramenta batizada Netfilter, sendo este um conjunto de situações de fluxo de dados agregadas inicialmente ao Kernel do Linux e dividido em tabelas.

Segurança - IPTables

Sob uma ótica mais “pratica”, podemos ver o Netfilter como um grande bando de dados que contém em sua estrutura 3 tabelas padrões:

- Filter
- Nat;
- Mangle.

Cada uma destas tabelas contém regras direcionadas os seus objetivos básicos.

A **tabela Filter**, por exemplo, guarda todas as regras aplicadas a um Firewall filtro de pacotes;

A **tabela Nat** as regras direcionadas a um Firewall Nat;

E a **Mangle** a funções mais complexas de tratamento de pacotes como o TOS.

Mas todas as tabelas possuem situações de fluxo (entrada, saída, redirecionamento, etc) que lhes proporcionam a realização de seus objetivos.

Segurança - IPTables

Tabela Filter

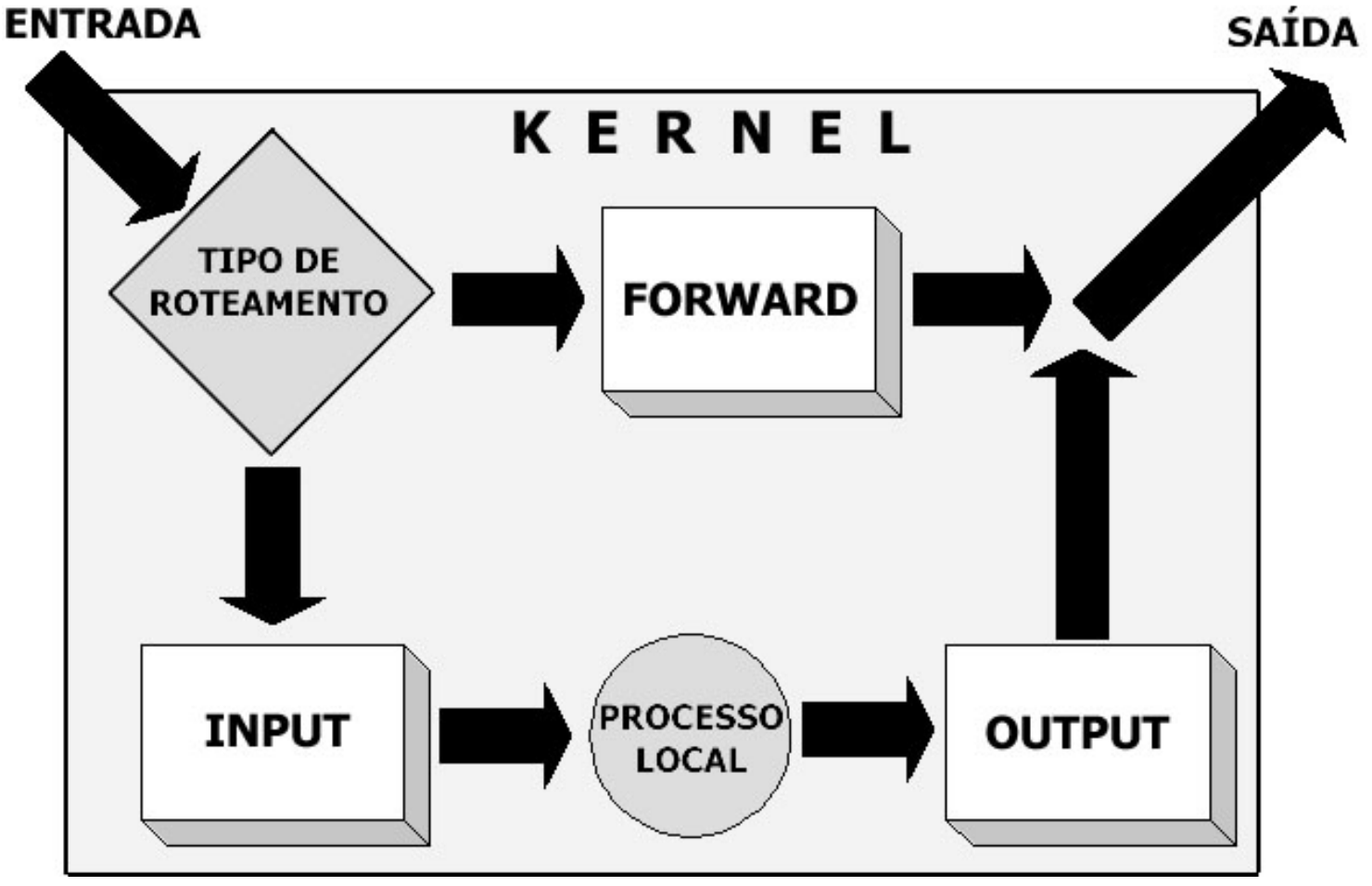
É a tabela padrão do Netfilter e trata das situações implementadas por um Firewall filtro de pacotes.

Estas situações são:

- **INPUT:** Tudo o que **entra** no *host*;
- **FORWARD:** Tudo o que chega ao host mas deve ser **redirecionado** a um host secundário ou outra interface de rede;
- **OUTPUT:** Tudo o que **sai** do host;

Segurança - IPTables

Tabela Filter



Segurança - IPTables

Tabela Nat

Esta tabela implementa funções de NAT (*Network Address Translation*) ao host Firewall. O Nat por sua vez, possui diversas utilidades.

Suas situações são:

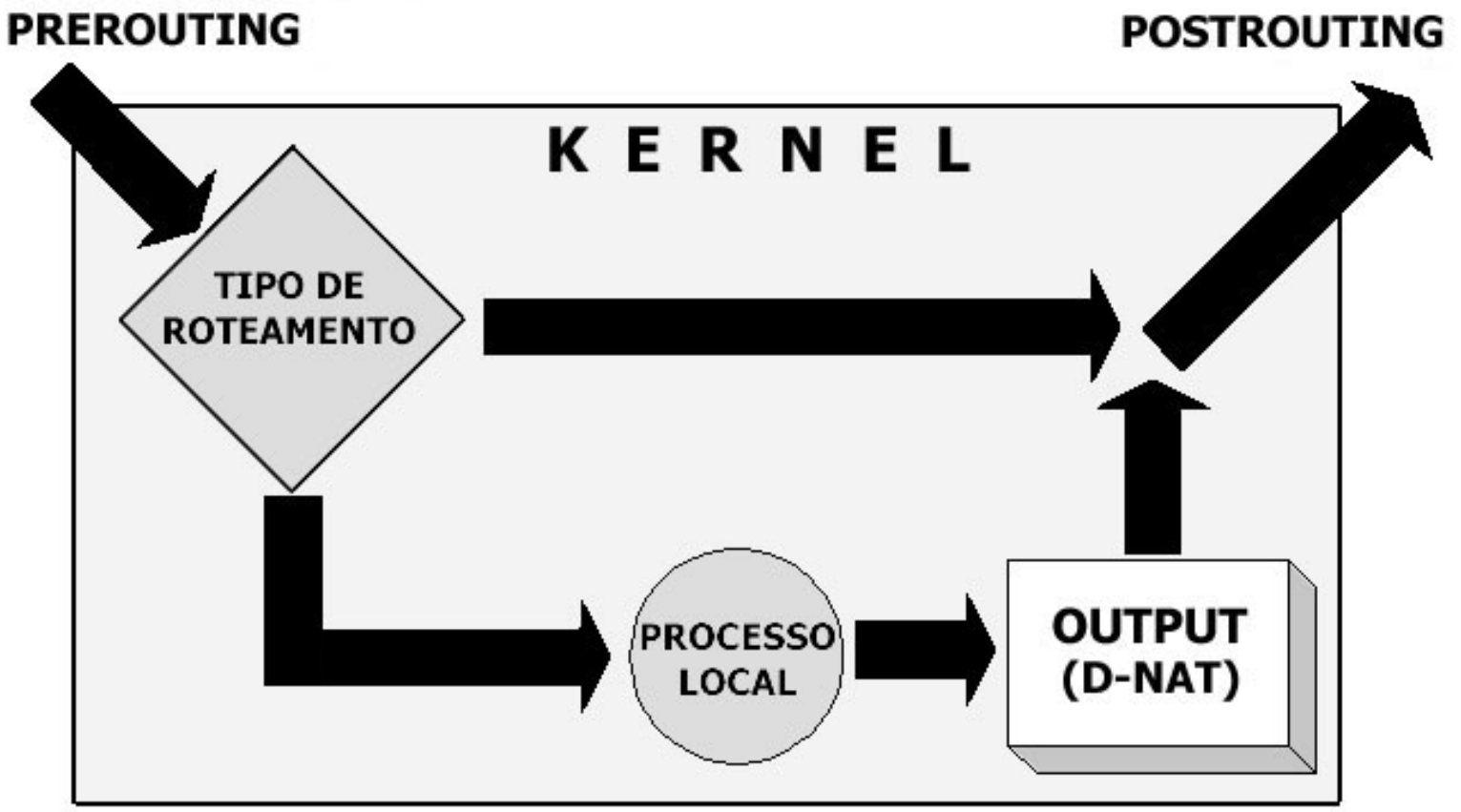
- **PREROUTING:** Utilizada quando há necessidade de se fazer alterações em pacotes antes que os mesmos sejam roteados;

- **OUTPUT:** Trata os pacotes emitidos pelo host Firewall;

- **POSTROUTING:** Utilizado quando há necessidade de se fazer alterações em pacotes após o tratamento de roteamento.

Segurança - IPTables

Tabela Nat



Segurança - IPTables

Tabela Mangle

Implementa alterações especiais em pacotes em um nível mais complexo.

A tabela Mangle é capaz, por exemplo, de alterar a prioridade de entrada e saída de um pacote baseado no tipo de serviço (TOS) o qual o pacote se destinava.

Suas situações são:

PREROUTING: Modifica pacotes dando-lhes um tratamento especial antes que os mesmos sejam roteados;

OUTPUT: Altera pacotes de forma “especial”gerados localmente antes que os mesmos sejam roteados.

Segurança - IPTables

Note que, ao analisarmos o fluxo de dados do Kernel do Linux as coisas começam a ficar mais claras. Dito que um host para fazer parte de uma rede precisa de situações (chains) de entrada (INPUT) e saída (OUTPUT).

Se agregarmos ainda uma nova situação, a de redirecionamento, ou simplesmente encaminhamento (FORWARD), poderíamos também manipular o que redirecionamos/encaminhamos a outros hosts, e assim sucessivamente.

O Iptables, trata-se, na verdade, de uma ferramenta de Front-End para lhe permitir manipular as tabelas do Netfilter, embora o mesmo seja constantemente confundido com um Firewall por si só.

O Iptables é uma versão mais robusta, completa e tão estável quanto seus antecessores IPFWARDM e IPCHAINS, implementados nos Kernels 2.0 e 2.2 respectivamente.

Segurança - IPTables

Como principais características, o Iptables além de realizar suas tarefas de forma veloz, segura, eficaz e econômica, tanto no aspecto financeiro quanto no de requerimento de hardware.

Isso nos dá um amplo leque de possibilidades tais como a implementação, tal como, filtro de pacotes utilizando a tabela Filter a NAT via tabela NAT e mais controles avançados como o desenvolvimento de QoS sobre o tráfego, suporte a SNAT e DNAT, redirecionamento de endereçamento e portas, mascaramento de conexões, detecção de fragmentos, monitoramento de tráfego, ToS, bloqueio a ataques Spoofing, etc.

E se, já não fosse o suficiente, ainda temos a opção de utilizar módulos externos na composição de regras, o que amplia ainda mais as funcionalidades do mesmo.

Quando se fala de requerimento de hardware, o Iptables é bastante generoso, necessitando apenas de um computador sobre a arquitetura 386 com aproximadamente 4 MB, e é claro, de um kernel 2.4 ou superior.

Sim, este é o básico do básico. É lógico que um pouco mais de memória e processamento só tendem a melhorar o desempenho do software tal como próprio sistema.

Segurança - IPTables

A principio, o Iptables é composto pelos seguintes aplicativos:

•**Iptables:** Aplicativo principal do pacote iptables para protocolos ipv4;

•**ip6tables:** Aplicativo do pacote iptables para protocolos ipv6;

•**iptables-save:** Aplicativo que salva todas as regras inseridas na sessão ativa e ainda em memória em um determinado arquivo informado pelo administrador de Firewall.

•**Iptables-restore:** Aplicativo que restaura todas as regras salvas pelo software iptables-save.

Os softwares Iptables e Netfilter possuem sua política de direitos e distribuição sob as regras GNU conforme publicado pela Free Software Foundation (FSF)

O Iptables é um módulo do Kernel, logo, o mesmo deve estar sendo executado pelo sistema para que possa vir a funcionar.

Segurança - IPTables

Pode ocorrer (embora que raramente) do módulo Ipchains (/lib/modules/2.6.5-63077cl/kernel/net/ipv4/netfilter/ipchains.o) estar ativo no sistema, o que impede que o Iptables possa vir a funcionar corretamente. Para desativá-lo é necessário proceder da seguinte maneira:

```
#rmmod ipchains
#lsmod
Module                  Size  Used by
ide_cd                 40836  0
cdrom                  40220  1 ide_cd
slamr                 372772  2
thermal                13584  0
processor              19248  1 thermal
fan                     4236  0
button                  6424  0
battery                 9612  0
```

Agora liste os módulos ativos em seu sistema utilizando o comando “lsmod”.

O Iptables deve ser carregado automaticamente, deste a instalação do sistema, entretanto se isso não estiver ocorrendo automaticamente por algum motivo, devemos então “levantar” o módulo do Iptables, para isso basta digitar o comando a seguir:

#insmod ip_tables

Segurança - IPTables

Por estar incorporado diretamente ao Kernel, à configuração do Iptables não se dá por via de arquivos de configuração, ao contrário, sua manipulação é realizada por síntese digitada em shell.

Quando inserimos uma regra na shell, ela estará valendo tão somente para aquela sessão “pendurada” em memória, sendo que uma vez resetado ou desligado o computador Firewall, tais regras serão perdidas e não mais poderão ser resgatadas.

Existe porém, agregado ao Iptables, uma função/programa denominada “iptables-save”, salvando suas regras no disco e resgatando-as diante de uma reinicializarão do sistema.

Para salvar as regras atuais de um sistema com o Iptables-save utilize a síntese a seguir:

#iptables-save > rc.firewall

O comando salvará as regras atuais do sistema para o arquivo rc.firewall. Tal arquivo pode ser adicionado a uma chamada no final do arquivo /etc/rc.local (que é como o autoexec.bat do linux).

Segurança - IPTables

Ora visto a teoria, passemos para a síntese do Iptables, síntese esta extremante intuitiva e lógica. A seguir a síntese do Iptables:

#iptables [tabela] [comando] [ação] [alvo]

Tabelas

São as mesmas que compõem o Netfilter (Filter, NAT, Mangle). Sendo utilizado está opção para associar uma regra a uma tabela específica:

- insere uma regra a tabela filter:

#iptables -t filter

- insere uma regra a tabela Mangle:

#iptables -t magle

- insere uma regra a tabela NAT:

#iptables -t nat

A tabela filter é a padrão do Iptables, logo, se adicionarmos uma regra sem utilizarmos a flag -t [tabela], o mesmo aplicará situações contidas na tabela filter. Já caso no caso das tabelas NAT e Mangle, é necessário especificar sempre a tabela.

Segurança - IPTables

Comando

-A Adiciona (anexa) uma nova entrada ao fim da lista de regras;
- Adiciona uma nova regra ao final da lista referente à INPUT (entrada):
#iptables -A INPUT

-D Apaga uma regra especifica da lista;

- Apaga a regra inserida anteriormente apenas trocando o comando -A pelo -D:
#iptables -D INPUT
O comando -D também lhe permite apagar uma certa regra por seu número da lista de ocorrências do Iptables, no caso abaixo é retirada a segunda regra da lista FORWARD:
#iptables -D FORWARD 2

-L Lista as regras existentes na lista:

#iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Comando

-P Altera a política padrão das chains (especificações). Inicialmente, todas as chains de uma tabela estão setadas como ACCEPT, ou seja, aceitam todo e qualquer tipo de tráfego. Para modificar esta política utilizamos a flag -P;

#iptables -P FORWARD DROP

A síntese anterior modifica a política padrão da chain FORWARD e ao invés de direcioná-lo para o alvo ACCEPT o leva ao DROP. Um pacote que é conduzido ao alvo DROP é descartado pelo sistema.

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy DROP)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Sempre que possível, antes de iniciar a configuração de um Firewall sobre qualquer tabela, modifique a política padrão de suas chains para REJECT ou DROP, isso lhe dará a tranqüilidade de que enquanto você conclui a configuração do mesmo, nenhum pacote inesperado trafegará por seu host/rede.

Segurança - IPTables

Comando

-F Este comando é capaz de remover todas as entradas adicionadas à lista de regras do Iptables sem alterar a política padrão;

#iptables -F

- Remove todas as regras

#iptables -F OUTPUT

- Remove todas as regras referentes à OUTPUT chain

-I Insere uma nova regra ao inicio da lista de regras (ao contrário do comando -A que insere ao final da lista);

#iptables -I OUTPUT

-R Substitui uma regra já adiciona por outra;

#iptables -R FORWARD 2 -s 10.0.0.3 -d 10.0.0.0/8 -j DROP

- Substitui a segunda regra referente à FORWARD chain pela seguinte: "*-s 10.0.0.3 -d 10.0.0.0/8 -j DROP*"

Segurança - IPTables

Comando

-N Este comando nos permite inserir/criar uma nova chain a tabela especificada.

#iptables -t filter -N internet

- Cria uma nova chains chamada “internet” sobre a tabela filter, a criação de uma nova chain surge apenas para tornar a organização de seu Firewall mais simples.

-E Renomeia uma nova chain (criada por você)

#iptables -e internet Intranet

- Renomeia a chain internet para Intranet

-X Apaga uma chain criada pelo administrador do Firewall

#iptables -X Intranet

Ação

-p Especifica o protocolo aplicado à regra. Pode ser qualquer valor numérico especificado no arquivo /etc/protocols ou o próprio nome do protocolo (tcp, icmp, udp, etc...);

ex. -p icmp

-i Especifica a interface de entrada a ser utilizada. Como um firewall possui mais de uma interface de rede, esta regra acaba sendo muito importante para distinguir a que interface de rede o filtro deve ser aplicado. Atenção à regra -i não deve ser aplicada na OUTPUT chain pois refere-sapenas a interface de entrada (INPUT e FORWARD);

ex. -i eth0

Podemos especificar também todas as interfaces “eth” de nosso host da seguinte forma:

ex. -i eth+

-o Especifica a interface de saída a ser utilizada e se aplica da mesma forma que a regra -i, porém, somente as chains OUTPUT e FORWARD se aplicam a regra;

ex. -o eth0

Ação

-s Especifica a origem (source) do pacote ao qual a regra deve ser aplicada. A origem pode ser um host ou uma rede. Nesta opção geralmente utilizamos o IP seguido de sua sub-rede;

ex. -s 10.0.0.0/255.0.0.0

A máscara de rede pode ser omitida deste comando, neste caso o iptables optará (de forma autônoma) pela máscara 255.255.255.0 (o que acaba se aplicando a todas as sub-redes). O -s também possibilita a utilização de nomes ao invés do IP;

ex. -s www.google.com.br

Obviamente para resolver este endereço o Iptables utilizará a resolução de nomes que consta configurado em seu host Firewall.

-d Especifica o destino do pacote (destination) ao qual a regra deve ser aplicada. Sua utilização se dá mesma maneira que a ação -s.

Ação

! Significa exclusão e é utilizado quando se deseja aplicar uma exceção a uma regra. É utilizado juntamente com as ações -s, -d, -p, -i, -o, etc;

ex. -s ! 10.0.0.3

- Refere-se a todos os endereços possíveis com exceção do 10.0.0.3

ex. -p ! icmp

- Refere-se a todos os protocolos possíveis com exceção do icmp

-j Define o alvo (target) do pacote caso o mesmo se encaixe a uma regra. As principais ações são ACCEPT, DROP, REJECT e LOG que serão citadas mais a frente.

--sport Porta de origem (source port), com esta regra é possível aplicar filtros com base na porta de origem do pacote. Só pode ser aplicada a porta referentes aos protocolos UDP e TCP;

ex. --p tcp -sport 80

--dport Porta de destino (destination port), especifica a porta de destino do pacote e funciona de forma similar a regra –sport.

Alvo

Quando um pacote se adequar a uma regra previamente criada ele deve ser direcionado a um alvo e quem o especifica é a própria regra. Os alvos (targets) aplicáveis são:

ACCEPT Corresponde a aceitar, ou seja, permite a entrada/passagem do pacote em questão.

DROP Corresponde a descartar imediatamente um pacote que é conduzido a este alvo. O Target DROP não informa ao dispositivo emissor do pacote o que houve. No caso de um ping (solicitação icmp), o mesmo não retornará mensagens ao host de origem, isso dará a entender que o host que sofreu a solicitação não existe, pois não enviará nenhum retorno ao mesmo.

REJECT Corresponde a rejeitar; Um pacote conduzido para este alvo é automaticamente descartado, a diferença do REJECT para o DROP é que o mesmo retorna uma mensagem de erro ao host emissor do pacote informando o que houve.

Segurança - IPTables

Alvo

LOG Cria uma entrada de log no arquivo /var/log/messages sobre a utilização dos demais alvos, justamente por isso deve ser utilizado antes dos demais alvos.

RETURN Retorna o processamento do chain anterior sem processar o resto do chain atual.

QUEUE Encarrega um programa em nível de usuário de administrar o processamento do fluxo atribuído ao mesmo.

SNAT Altera o endereço de origem das máquinas clientes. Pode, por exemplo, enviar um pacote do host “A” ao host “B” e informar ao host “B” que tal pacote fora enviado pelo host “C”.

Segurança - IPTables

Alvo

DNAT Altera o endereço de destino das máquinas clientes. Pode, por exemplo, receber um certo pacote destinado à porta 80 do host ”A” e encaminha por conta própria à porta 3128 do host “B”. Isso é que chamamos de Proxy transparent, um encaminhamento dos pacotes dos clientes dos pacotes dos clientes sem que os mesmos possuam a opção de escolher ou não tal roteamento.

REDIRECT Realiza redirecionamento de portas em conjunto com a opção –to-port.

TOS Prioriza a entrada e saída de pacotes baseados em seu “tipo de serviço”, informação esta especificada no header do Ipv4.

Segurança - IPTables

- Exemplos

Lembrando as regras de firewall geralmente, são compostas assim:

#iptables [-t tabela] [opção] [chain] [dados] -j [ação]

Exemplo:

#iptables -A FORWARD -d 192.168.1.1 -j DROP

A linha acima determina que todos os pacotes destinados à máquina 192.168.1.1 devem ser descartados. No caso:

tabela: filter (é a default)

opção: -A

chain: FORWARD

dados: -d 192.168.1.1

ação: DROP

Segurança - IPTables

- Exemplos

-L --> List (listar). Lista as regras existentes.

Exemplos:

#iptables -L
#iptables -L FORWARD

Segurança - IPTables

- Exemplos

-P --> Policy (política).

Altera a política da chain. A política inicial de cada chain é ACCEPT. Isso faz com que o firewall, inicialmente, aceite qualquer INPUT, OUTPUT ou FORWARD. A política pode ser alterada para DROP, que irá negar o serviço da chain, até que uma opção -A entre em vigor. O -P não aceita REJECT ou LOG.

Exemplos:

#iptables -P FORWARD DROP

#iptables -P INPUT ACCEPT

Segurança - IPTables

- Exemplos

-A --> Append (anexar).

Acresce uma nova regra à chain. Tem prioridade sobre o -P. Geralmente, como buscamos segurança máxima, colocamos todas as chains em política DROP, com o -P e, depois, abrimos o que é necessário com o -A.

Exemplos:

#iptables -A OUTPUT -d 172.20.5.10 -j ACCEPT

#iptables -A FORWARD -s 10.0.0.1 -j DROP

#iptables -A FORWARD -d www.chat.com.br -j DROP

Segurança - IPTables

- Exemplos

-D --> Delete (apagar).

Apaga uma regra. A regra deve ser escrita novamente, trocando-se a opção para -D.

Exemplos:

Para apagar as regras anteriores, usa-se:

#iptables -D OUTPUT -d 172.20.5.10 -j ACCEPT

#iptables -D FORWARD -s 10.0.0.1 -j DROP

#iptables -D FORWARD -d www.chat.com.br -j DROP

Também é possível apagar a regra pelo seu número de ordem. Pode-se utilizar o -L para verificar o número de ordem. Verificado esse número, basta citar a chain e o número de ordem. Exemplo:

#iptables -D FORWARD 4

Isso deleta a regra número 4 de forward.

Segurança - IPTables

- Exemplos

-F --> Flush (esvaziar).

Remove todas as regras existentes. No entanto, não altera a política (-P).

Exemplos:

#iptables -F
#iptables -F FORWARD

Segurança - IPTables

- Exemplos

Lista todas as regras existentes:
#iptables -L

Apaga todas as regras sem alterar a política:
#iptables -F

Estabelece uma política de proibição inicial de passagem de pacotes entre sub-redes.
#iptables -P FORWARD DROP

Todos os pacotes oriundos de qualquer sub-rede e destinados a qualquer sub-rede deverão ser descartados.
#iptables -A FORWARD -j DROP

Todos os pacotes oriundos de qualquer sub-rede e destinados a qualquer sub-rede deverão ser aceitos.
#iptables -A FORWARD -j ACCEPT

Segurança - IPTables

- Exemplos

Os pacotes oriundos da sub-rede 10.0.0.0 (máscara 255.0.0.0) e destinados ao host www.chat.com.br deverão ser descartados.

#iptables -A FORWARD -s 10.0.0.0/8 -d www.chat.com.br -j DROP

Os pacotes oriundos da sub-rede 10.0.0.0 (máscara 255.0.0.0) e destinados ao host www.chat.com.br deverão ser descartados. Deverá ser enviado um ICMP avisando à origem.

#iptables -A FORWARD -s 10.0.0.0/8 -d www.chat.com.br -j REJECT

Os pacotes oriundos de qualquer lugar e destinados ao host www.chat.com.br deverão ser descartados.

#iptables -A FORWARD -d www.chat.com.br -j DROP

Os pacotes destinados à sub-rede 10.0.0.0 (máscara 255.0.0.0) e oriundos do host www.chat.com.br deverão ser descartados.

#iptables -A FORWARD -d 10.0.0.0/8 -s www.chat.com.br -j DROP

Segurança - IPTables

- Exemplos

Os pacotes oriundos do host www.chat.com.br e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -s www.chat.com.br -j DROP

Os pacotes oriundos da sub-rede 200.221.20.0 (máscara 255.255.255.0) e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -s 200.221.20.0/24 -j DROP

Os pacotes icmp oriundos do host 10.0.0.5 e destinados a qualquer lugar deverão ser descartados.

#iptables -A FORWARD -s 10.0.0.5 -p icmp -j DROP

Os pacotes que entrarem pela interface eth0 serão aceitos.

#iptables -A FORWARD -i eth0 -j ACCEPT

Segurança - IPTables

- Exemplos

Os pacotes que entrarem por qualquer interface, exceto a eth0, serão aceitos.

#iptables -A FORWARD -i ! eth0 -j ACCEPT

O tráfego de pacotes TCP oriundos da porta 80 do host 10.0.0.5 e destinados a qualquer lugar deverá ser gravado em log. No caso, /var/log/messages.

#iptables -A FORWARD -s 10.0.0.5 -p tcp --sport 80 -j LOG

Os pacotes TCP destinados à porta 25 de qualquer host deverão ser aceitos.

#iptables -A FORWARD -p tcp --dport 25 -j ACCEPT

Segurança - IPTables

Impasses e ordem de processamento

Se houver impasse entre regras, sempre valerá a primeira. Assim, entre as regras:

#iptables -A FORWARD -p icmp -j DROP
#iptables -A FORWARD -p icmp -j ACCEPT

Valerá:
#iptables -A FORWARD -p icmp -j DROP

Já entre as regras:
#iptables -A FORWARD -p icmp -j ACCEPT
#iptables -A FORWARD -p icmp -j DROP

Valerá:
#iptables -A FORWARD -p icmp -j ACCEPT

Impasses e ordem de processamento

Isso ocorre porque as regras são processadas na ordem em que aparecem. Depois do processamento da regra, pode haver continuidade de processamento ou não. Isso irá depender da ação:

ACCEPT --> Pára de processar regras para o pacote atual;

DROP --> Pára de processar regras para o pacote atual;

REJECT --> Pára de processar regras para o pacote atual;

LOG --> Continua a processar regras para o pacote atual;

O retorno

Ao se fazer determinadas regras, devemos prever o retorno. Assim, digamos que exista a seguinte situação:

```
#iptables -P FORWARD DROP
#iptables -A FORWARD -s 10.0.0.0/8 -d 172.20.0.0/16 -j ACCEPT
```

Com as regras anteriores, fechamos todo o FORWARD e depois abrimos da sub-rede 10.0.0.0 para a sub-rede 172.20.0.0. No entanto, não tornamos possível a resposta da sub-rede 172.20.0.0 para a sub-rede 10.0.0.0.

O correto, então, seria:

```
#iptables -P FORWARD DROP
#iptables -A FORWARD -s 10.0.0.0/8 -d 172.20.0.0/16 -j ACCEPT
#iptables -A FORWARD -d 10.0.0.0/8 -s 172.20.0.0/16 -j ACCEPT
```

Segurança - IPTables

1- Listar as regras anexadas ao iptables nas tabelas: filters; nat; mangle. E também observar as políticas das chains (situações).

2- Configurar o alvo padrão das chains referente a tabela filter como DROP.

3- Liberar totalmente o trafego de entrada de nossa interface de loopback (lo). Essa regra deve sempre fazer parte do script de inicialização para permitir a comunicação entre processos locais. Lembre-se está regra não é opcional se você estiver usando políticas de negar tudo (DROP) o que não for permitido.

4 – Proibir qualquer pacote oriundo de nossa LAN (10.0.30.0) endereçados ao site www.terra.com.br.

5 – especificar que qualquer pacote oriundo do host www.cacker.com não pode penetrar em nossa rede (10.0.30.0).

Segurança - IPTables

6 – Faremos agora com que os pacotes provenientes do site da universidade www.universidade.br penetrem livremente em nossa rede.

7 – Todos os pacotes oriundos de qualquer rede que penetre em nosso Firewall pela interface eth2 serão redirecionados para o computador 10.0.30.47.

8 – Fazer com que qualquer pacote que deseje sair da rede local para outra rede possua seu endereço de origem alterado para 192.168.0.33, implementando assim o conceito de máscaramento ip. Este mesmo pacote somente sofrerá modificações se sair pela interface de rede eth2.

9 – Utilizar a tabela filter para rejeitar pacotes entrantes pela interface de rede eth1.

10 – Todos os pacotes que entram por qualquer interface de rede com exceção da eth0 sejam descartados.

Segurança - IPTables

11 – Deletar a segunda regra inserida sobre a chain FORWARD.

12 – Listar agora as regras associadas a chain OUTPUT.

13 – Descartar qualquer pacote oriundo do IP 10.0.80.32 destinado ao IP 10.0.30.4.

14 – Pacotes TCP destinados à porta 80 de nosso host firewall deverão ser descartados;

15 – Fazer com que pacotes destinados a porta 25 de nosso host Firewall sejam arquivados em log e em seguida sejam descartados.

Segurança - IPTables

Resposta da tarefa

1 - #iptables –L

2 - #iptables –P INPUT DROP
#iptables –P FORWARD DROP
#iptables –P OUTPUT DROP

3 - #iptables –A INPUT –I lo –j ACCEPT

4- #iptables –A FORWARD –s 10.0.30.0/8 –d www.sexo.com.br –j DROP

5- #iptables –A FORWARD –s www.terra.com.br –d 10.0.30.0 –j DROP

Segurança - IPTables

Resposta da tarefa

6- #iptables –A FORWARD –s www.universidade.br –d 10.0.30.0 –j ACCEPT

7- #iptables –t nat –A PREROUTING –I eth2 –j DNAT –to 10.0.30.47

8- #iptables –t nat –A POSTROUTING –o eth2 –j SNAT –to 192.168.0.33

9- #iptables –A FORWARD –i eth1 –j REJECT

10- #iptables –A FORWARD –i ! eth0 –j DROP

Segurança - IPTables

Resposta da tarefa

11 - #iptables –D FORWARD 2

12 - #iptables –L OUTPUT

13 - #iptables –A FORWARD –s 10.0.80.32 –d 10.0.30.2 –j DROP

14 - #iptables –A INPUT –p tcp –dport 80 –j DROP

15 - #iptables –A INPUT –p tcp –dport 25 –j LOG
#iptables –A INPUT –p tcp –dport 25 –j DROP

SNAT

Uma das funções de um Firewall Nat é o que conhecemos por SNAT, ou simplesmente “tradução de endereçamento de origem” (source nat).

O alvo (target) SNAT lhe dá a possibilidade de alterar os endereços/portas de origem dos pacotes que atravessam seu host Firewall antes que os mesmos sejam roteados a seu destino final.

Alguns conceitos práticos do nat são:

a) Qualquer regra aplicada a SNAT utiliza-se somente da chain POSTROUNTING. Logo se é SNAT que você quer, é POSTROUTING que você usa!

b) Antes de iniciar a manipulação de qualquer regra que utilize NAT, é importante que habilitemos a função de redirecionamento de pacotes (forward) em nosso kernel através do seguinte comando:

#echo “1” >/proc/sys/net/ipv4/ip_forward

c) Compreendido e feito isso é possível utilizar o SNAT.

Segurança - IPTables

SNAT

Por exemplo:

Criar uma regra na tabela NAT sob a chain POSTROUTING de forma que qualquer pacote que possua como origem 10.0.3.0 e que saia pela interface eth1 deverá possuir se endereço de destino alterado para 192.111.22.33

#iptables –t nat –A POSTROUTING –s 10.0.3.1 –o eth0 –j SNAT --to 192.168.22.33

Especificar que qualquer pacote que possua como origem a rede 10.0.3.0/8 e que sair por eth0 deverá ter seu endereço alterado para 192.111.22.33.

#iptables –t nat –A POSTROUTING –s 10.0.3.0/8 –o eth0 –j SNAT --to 192.168.22.33

Segurança - IPTables

SNAT

Por exemplo:

Criar uma regra na tabela NAT sob a chain POSTROUTING de forma que qualquer pacote que possua como origem 10.0.3.0 e que saia pela interface eth1 deverá possuir se endereço de destino alterado para 192.111.22.33

#iptables –t nat –A POSTROUTING –s 10.0.3.1 –o eth0 –j SNAT --to 192.168.22.33

Especificar que qualquer pacote que possua como origem a rede 10.0.3.0/8 e que sair por eth0 deverá ter seu endereço alterado para 192.111.22.33.

#iptables –t nat –A POSTROUTING –s 10.0.3.0/8 –o eth0 –j SNAT --to 192.168.22.33

Segurança - IPTables

DNAT

Outra função agregada a um Firewall NAT é o DNAT, ou “tradução de endereçamento de destino” (detination nat).

O alvo DNAT lhe dá a possibilidade de alterar os endereços/portas de destino dos pacotes que atravessam seu host Firewall antes que os mesmos sejam roteados a seu destino final. Com isso o DNAT nos possibilita o desenvolvimento de Poxys transparentes, balanceamento de carga, entre outros.

As regras do DNAT, ao contrário do SNAT, utilizam-se tão somente da chain PREROUTING. Logo, se é DNAT que você quer, é PREROUNTING que você vai fazer!

DNAT

Da mesma forma que o SNAT o DNAT precisa da função forward habilitada no kernel:

#echo “1” >/proc/sys/net/ipv4/ip_forward

Exemplos de DNAT:

Criar uma regra na tabela NAT sob a chain PREROUTING, informando que qualquer pacote que possua como origem o host 10.0.3.1 e que entre por nossa interface eth1 deve ter seu endereço de destino alterado para 192.111.22.33.

A podemos observar que mesmo que o host 10.0.3.1 tenha enviado um pacote ao host 192.55.55.55, o mesmo será forçadamente redirecionado sem consentimento do cliente (10.0.3.1) para o host 192.11.22.33.

#iptables –t nat PREROUTING –s 10.0.3.1 –i eth1 –j DNAT --to 192.111.22.33

Segurança - IPTables

DNAT

É possível fazer balanceamento de carga:

#iptables –t nat –A PREROUTING –i eth0 –j DNAT --to 192.11.22.10-192.11.22.13

É possível redirecionar a porta

#iptables –t nat –A PREROUTING –I eth2 –j DNAT --to 192.11.22.58:22

Anunciamos ao iptables que qualquer pacote que entre por nossa interface de rede eth2, independentemente de quem enviou, deve ser automaticamente redirecionado ao host 192.11.22.10, e independentemente da porta solicitada pelo cliente (origem), deverá ser enviado à porta 22 do mesmo.

Vemos então um típico conceito onde todo o tráfego é filtrado por um Firewall e apenas os pacotes entrantes por uma certa interface deverão ser aceitos e redirecionados a um serviço especifico, no caso do exemplo acima, o SSH.

Tansparent Proxy

O Proxy transparente é a forma que a tabela NAT possui para realizar um redirecionamento de portas em um mesmo host de destino.

Este método é comumente utilizado, por exemplo, pelo Proxy-cache Squid, o mesmo costuma por padrão disponibilizar consultas www através da porta 3128, enquanto a maioria dos clientes costuma realizar tais solicitações à porta 80 do mesmo.

Logo, podemos concluir que o Squid faz um redirecionamento das portas solicitadas por seus clientes, uma vez que os mesmos solicitam conexões via porta 80 e são redirecionados à porta 3128.

Utilizamos então para esta finalidade tão somente as chains PREROUTING e OUTPUT da tabela nat, tal como o alvo REDIRECT.

Segurança - IPTables

Tansparent Proxy

Tome apenas cuidado para não confundir Proxy transparente com DNAT. Lembre se que a única forma de se fazer redirecionamento de portas de destino em um mesmo host é via REDIRECT, esta uma vez utilizada caracteriza então um modelo de Proxy transparente.

Exemplo:

#iptables –t nat –A PREROUTING –i eth0 –p tcp --dport 80 –j REDIRECT --to-port 3128

Especificando fragmentos

Às vezes um pacote é muito grande para ser enviado todo de uma vez. Quando isso ocorre, o pacote é dividido em fragmentos, e é enviado como múltiplos pacotes. Quem o recebe, remonta os fragmentos reconstruindo o pacote.

O problema com os fragmentos é que o fragmento incial tem os campos de cabeçalho completos (IP + TCP, UDP e ICMP) para examinar, mas os pacotes subsequentes só têm um pequeno grupo de cabeçalhos (somente IP, sem os campos dos outros protocolos). Logo examinar o interior dos fragmentos subsequentes em busca de cabeçalhos de outros protocolos (como extensões de TCP, UDP e ICMP) é impossível.

Se você está fazendo acompanhamento de conexões ou NAT, todos os fragmentos serão remontados antes de atingirem o código do filtro de pacotes, então você não precisa se preocupar sobre os fragmentos.

Caso contrário, é importante entender como os fragmentos são tratados pelas regras de filtragem. Qualquer regra de filtragem que requisitar informações que não existem não será válida. Isso significa que o primeiro fragmento é tratado como um pacote qualquer. O segundo e os seguintes não serão. Então uma regra -p TCP --sport www (especificando 'www' como porta de origem) nunca se associar-se-á com um fragmento (exceto o primeiro). O mesmo se aplica à regra oposta -p TCP --sport ! www.

Especificando fragmentos

De qualquer forma, você pode especificar uma regra especial para o segundo e os fragmentos que o sucedem, utilizando a flag `-f' (ou `--fragment'). Também é interessante especificar uma regra que não se aplica ao segundo e os outros fragmentos, precedendo a opção `-f' com `!'.

Geralmente é reconhecido como seguro deixar o segundo fragmento e os seguintes passarem, uma vez que o primeiro fragmento será filtrado, logo isso vai evitar que o pacote todo seja remontado na máquina destinatária. De qualquer forma, há bugs que levam à derrubada de uma máquina através do envio de fragmentos. A decisão é sua.

Nota para os administradores de rede: pacotes mal formados (pacotes TCP, UDP e ICMP muito pequenos para que o código do firewall possa ler as portas ou o código e tipo dos pacotes ICMP) são descartados quando ocorrem tais análises.

Como um exemplo, a regra a seguir vai descartar quaisquer fragmentos destinados ao endereço 192.168.1.1:

```
# iptables -A OUTPUT -f -d 192.168.1.1 -j DROP
```

Extensões ao iptables:

O iptables é extensível, assim, tanto o kernel quanto o iptables podem ser estendidos para fornecer novas funcionalidades.

As extensões do kernel geralmente estão no subdiretório de módulos do kernel como, por exemplo, /lib/modules/2.3.15/net. Elas são carregadas por demanda se seu kernel foi compilado com a opção CONFIG_KMOD marcada, não havendo a necessidade de inserí-las manualmente. As extensões do iptables são bibliotecas compartilhadas as quais geralmente estão em /usr/local/lib/iptables/, mas uma distribuição os colocaria em /lib/iptables ou /usr/lib/iptables.

Alguns protocolos oferecem automaticamente o uso de extensões: atualmente são eles TCP, UDP e ICMP como mostrado abaixo.

Para esses protocolos você poderá especificar novos testes na linha de comando depois da opção `-p', que carregará a extensão. Para novos testes explícitos, utilize a opção `-m' para carregar a extensão, depois de -m, todas as opções da extensão estarão disponíveis.

Para conseguir ajuda com uma extensão, utilize a opção que a carrega (`-p', `-j' ou `-m') sucedida por `-h' ou `--help', conforme o

exemplo:

iptables -p tcp --help

Extensões TCP

As extensões TCP são automaticamente carregadas se é especificada a opção `-p tcp'. Elas provêm as seguintes opções.

--tcp-flags

Seguida por uma opcional `!', e por duas strings indicando flags, permite que sejam filtradas flags TCP específicas. A primeira string de flags é a máscara: uma lista de flags que serão examinadas. A segunda string diz quais flags devem estar ativas para que a regra se aplique.

Por exemplo:

iptables -A INPUT --protocol tcp --tcp-flags ALL SYN,ACK -j DROP

Essa regra indica que todas as flags devem ser examinadas (`ALL' é sinônimo de `SYN,ACK,FIN,RST,URG,PSH'), mas apenas SYN e ACK devem estar ativadas. Também há um argumento `NONE' que significa nenhuma flag.

Extensões TCP

--syn

Opcionalmente precedido por `!', é um atalho para `--tcp-flags SYN,RST,ACK SYN'.

Uma explicação sobre as flags TCP

Às vezes é util permitir conexões TCP em uma única direção, mas não nas duas.

Por exemplo, você permitirá conexões em um servidor WWW externo, mas não conexões vindas deste servidor.

Uma tentativa ingênua seria bloquear pacotes TCP vindos do servidor. Infelizmente, conexões TCP necessitam de pacotes bidirecionais para um funcionamento perfeito.

Uma explicação sobre as flags TCP

A solução é bloquear apenas os pacotes utilizados para requisitar uma conexão. Tais pacotes são chamados pacotes SYN (tecnicamente eles são pacotes com a flag SYN marcada, e as flags RST e ACK desmarcadas). Ao não permitir tais pacotes, nós impedimos conexões vindas do servidor.

A opção `--syn' é utilizada para isso: só é válida para regras que especificam TCP como protocolo. Por exemplo, para especificar conexões TCP vindas do endereço 192.168.1.1:

```
-p TCP -s 192.168.1.1 --syn
```

Essa opção pode ser invertida se precedida por `!', o que significa qualquer pacote exceto os que iniciam conexões.

Extensões UDP

Essas extensões são automaticamente carregadas se a opção `-p udp' é especificada.

Ela provê as opções `--source-port', `--sport', '--destination-port' and `--dport'.

Extensões ICMP

Essas extensões são automaticamente carregadas se a opção `-p icmp' é especificada. Ela só possui uma opção diferente das demais:

--icmp-type

Seguida por `!' opcional, e um nome de tipo icmp (por exemplo, `host-unreachable'), ou um tipo numérico (exemplo `3'), ou um tipo numérico e código separados por `/' (exemplo `3/3').

Uma lista de tipos icmp é passada utilizando-se `-p icmp --help'.

Outras extensões

Outras extensões no pacote netfilter são extensões de demonstração, que (caso instaladas) podem ser chamadas com a opção `-m' ou `--match`.

mac

Este módulo deve ser explicitamente especificado com `-m mac' ou `--match mac'. Ele é usado para associar-se com o endereço Ethernet (MAC) de origem do pacote, e logo só é útil nas chains PREROUTING e INPUT.

Só provê uma única opção:

--mac-source

Seguida por `!' opcional e um endereço ethernet passado em notação hexa, exemplo `--mac-source 00:60:08:91:CC:B7'.

Outras extensões

Limit

Este módulo deve ser explicitamente especificado com `-m limit' ou `--match limit'. É usado para restringir a taxa de pacotes, e para suprimir mensagens de log. Vai fazer com que a regra seja válida apenas um número de vezes por segundo (por padrão 3 vezes por hora, com um limite máximo def 5).

Possui dois argumentos opcionais:

--limit

Seguido de um número; especifica a média máxima de pacotes (ou LOGs, etc) permitida por segundo. Pode-se especificar a unidade de tempo, usando `/second', `/minute', `/hour' ou `/day', ou abreviações dos mesmos (assim, `5/second' é o mesmo que `5/s').

Outras extensões

--limit-burst

Seguido de um número, indicando o máximo de entradas antes do limite tornar-se válido.

Essa extensão pode ser usada com o alvo (target) LOG para criar registros (logs) limitados por taxa de incidência. Para entendero funcionamento disso, olhe a regra abaixo, que loga pacotes com os parâmetros padrão de limite:

iptables -A FORWARD -m limit -j LOG

Outras extensões

Na primeira vez que essa regra é analizada, o pacote será logado; na realidade, uma vez que o padrão máximo é 5, os cinco primeiros pacotes serão logados.

Depois disso, passar-se-ão vinte minutos antes de que essa regra faça um novo log, independente do número de pacotes que entrarem.

Nota: não é possível criar uma regra com tempo de recarga superior a 59 horas, então se você configura uma taxa média de um por dia, seu limite máximo deve ser menor que 3.

Esse módulo também pode ser usado para evitar uma série de ataques do tipo negativa de serviço (denial of service 'DoS') com uma taxa mais rápida, a fim de aumentar a sensibilidade do sistema.

Proteção contra Syn-flood:

```
# iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

Segurança - IPTables

Outras extensões

Esse módulo também pode ser usado para evitar uma série de ataques do tipo negativa de serviço (denial of service 'DoS') com uma taxa mais rápida, a fim de aumentar a sensibilidade do sistema.

Proteção contra Syn-flood:

```
# iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

Port scanner suspeito:

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT
```

Ping da morte:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
```

Segurança - IPTables

Módulo de extensão state

O módulo state atribui regras mediante a análise do estado da conexão de um pacote.

Estes estados de forma mais detalhadas são:

NEW – Indica que o pacote está criando uma nova conexão;

ESTABLISHED – Informa que o pacote pertence a uma conexão já existente, logo, trata-se de um pacote de resposta;

RELATED – Refere-se a pacotes que se relacionam indiretamente com outro pacote, a exemplos das mensagens de erros de conexões;

INVALID – Referente a pacotes não identificados por algum motivo desconhecidos, geralmente aconselha-se que sejam descartados.

É possível usar mais de um estado por regra, separando-os por vírgulas.

Segurança - IPTables

Módulo de extensão state

A regra abaixo permite qualquer nova conexão que parta da interface eth0:

#iptables –A INPUT –m state --state NEW –i eth0 –j ACCEPT

A regra abaixo bloqueia qualquer pacote cujo estado de conexão seja considerado inválido:

#iptables –A INPUT –m state --state INVALID –i eth0 –j DROP

Abaixo aceitamos qualquer pacote sob os estados ESTABLICHED e RELATED, ou seja, a pacotes que já estabelecem uma conexão e pacotes não identificados mas que possuem alguma ligação indireta com outros pacotes identificados.

#iptables –A INPUT –m state --state ESTABLICHED, RELATED –j ACCEPT

Segurança - IPTables

Módulo multiport

Com o módulo multiport é possível especificar múltiplas portas a um determinado alvo sob o limite máximo de 15.

#iptables –A INPUT –p tcp –i eth0 –m multiport --dport 80,25,53,110 –j DROP

Assim indicamos em uma só regra que nosso firewall descartará qualquer pacote que entre pela interface eth0 destinados às portas 80,25,53 e 110.

Tal regra também pode ser aplicada a --sport, ou para --port para ambas as portas.

Segurança - IPTables

Modulo string

O módulo string é útil quando precisamos realizar um controle de tráfego baseado no conteúdo de um pacote.

No exemplo de bloquear conteúdos com a palavra “sexo” usamos a seguinte regra:

#iptables –A INTPU –m string --string “sexo” –j DROP

É possível também gravar em logs tais acessos:

#iptables –A INPUT –m string --string “sexo” –j LOG --log-prefix “ATENÇÃO: Site de SEXO”

É possível bloquear também serviços como o kazza:

#iptables –A INPUT –m string --string “X-kazza” –j DROP

#iptables –A INPUT –m string --string “GET /.hash=” –j DROP

Segurança - IPTables

Modulo owner

O owner é capaz de determinar precisamente algumas informações valiosas sobre o criador de um determinado pacote definido em regra, de modo que se tornará possível identificar o emissor real do pacote (no nível de usuário).

Este módulo pode ser utilizado tão somente em combinação com a chain OUTPUT conforme vemos no exemplo a seguir, que bloqueará a saída de qualquer pacote UDP que seja criado por um usuário do grupo 81 (GID 81).

#iptables –A OUTPUT –m owner --gid-owner 81 –p udp –j DROP

Segurança - IPTables

Modulo owner

É possível utilizar as seguintes s opções com o owner:

--uid-owner – Controla e executa regras se os pacotes fora criado por um usuário sob userid especificado.

--gid-owner - Controla e executa regras se os pacotes fora criado por um grupo de usuários sob o groupid especificado.

--pid-owner - Controla e executa regras se os pacotes fora criado por um número de processo concebido especificado.

--sid-owner - Controla e executa regras se o pacotes fora criado por processo concebido por session group.

--unclean - opção experimental, realiza checagem aleatórias nos pacotes a fim de identificá-los como suspeitos.

Outros exemplos de owner:

#iptables –A OUTPUT –m owner --uid-owner 42 –p tcp –j ACCEPT

#iptables –A OUTPUT –m owner --gid-owner 50 –d www.playboy.com -j ACCEPT

#iptables –A OUTPUT –d www.playboy.com –j DROP

Tabela Mangle

Conceito TOS

O IPv4 combina diversas informações em seu cabeçalho. Um destes é o “tipo de serviço”, que é um campo que possibilita aplicar um TOS, ou seja, controle do tipo de serviço a um tráfego destinado a um host/rede.

Isso nos permite então dizer a um Firewall que qualquer pacote cujo “tipo de serviço” seja “SSH” possui uma prioridade de trafego “x”, e que outros cujo “tipo de pacote” seja “ICQ”, possuem prioridade “y”.

Com isso, o TOS torna-se então uma forma simples, porém eficiente de lhe dar total controle sobre o tráfego de entrada e saída de sua rede.

Segurança - IPTables

Tabela Mangle

Aplicando o TOS

A Mangle nos serve para realizar controle e alterações “especiais” em pacotes. O TOS é, mais especificamente e como visto anteriormente, o nome dada a um alvo (tal como DROP, ACCEPT, etc) que fará então este controle.

Este mesmo alvo (TOS), combinado com o argumento –set-tos lhe permitirá aplicar os conceitos de gerenciamento de tráfego por “tipo de serviço”.

Para que haja a possibilidade de se alterar o bit de prioridade de um pacote por seu “tipo de serviço”, possuímos então uma tabela com especificações e valores que deverão ser alterados conforme veremos a seguir:

Bit TOS	
Espera Mínima (Minimize-Delay)	16 ou 0x10
Máximo Processamento (Maximize-Throughput)	8 ou 0x08
Máximo de Confiança (Maximize-Realiability)	4 ou 0x04
Custo mínimo (minimize-cost)	2 ou 0x02
Prioridade Normal (Normal-Service)	0 ou 0x00

Tabela Mangle

Controlando o tráfego de saída:

O Exemplo a seguir nos demonstra na prática como aplicar uma prioridade máxima (espera mínima) para tráfego de saída na interface eht0 em pacotes sob o protocolo ssh. Isso na prática quer dizer que qualquer pacote SSH será direcionado a seu destino primeiro que qualquer outro.

#iptables –t mangle –A OUTPUT –o eth0 –p tcp --dport 22 –j TOS --set-tos 16

Essa regra controla apenas o tráfego de saída e não o de entrada para controla tal tráfego cabe a seguinte regra:

iptables –t mangle –A PREROUNTING –i eth0 –p tcp --sport 22 –j TOS --set-tos 16

Segurança - IPTables

Tabela Mangle

Para o tratamento deste tráfego, o de entrada, utilizamos tão somente a chain PREROUTING, e é claro, ao especificarmos a interface, utilizamos a ação –i, o que indica pacotes entrantes na mesma.

O exemplo a seguir dará máximo processamento (Maximize-Throughput) a pacotes que entrem em seu host/rede sob o protocolo FTP.

#iptables –t mangle –A PREROUTING –i eth0 tcp --sport 20 –j TOS --set-tos 8

Ainda com prioridade acima do tráfego normal mas abaixo dos especificados acima, o exemplo seguinte dará máxima confiança a pacotes que entrem em seu host/rede sob o protocolo web.

#iptables –t mangle –A PREROUTING –i eth0 –p tcp --sport 80 –j TOS --set-tos 4