IPTABLES

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Laboratório de Software Livre

4 de fevereiro de 2010

Sumário

1 Introdução

Netfilter é um modulo do kernel linux responsável pelo controle da rede entre computadores, é ele quem controla as funções NAT, Firewall e Log do sistema. Iptables é o nome da ferramenta que controla o modulo netfilter do kernel linux, este nome é usualmente utilizado para referenciar as funcionalidades do netfilter.

2 Pré-requisitos

2.1 Dependencias

A instalação da ferramenta depende apenas do pacote libc6.

2.2 Instalação

O pacote Iptables já vem instalado por default em varias distribuições GNU/Linux, a ultima versão do ubuntu (9.04) tambem segue esta ideia. Porem se for preciso instalar-la, basta executar:

```
# aptitude install iptables
```

3 Funcionamento

O iptables se baseia em pares de regras e ações. As regras definem em quais pacotes atuar (Ex: pacotes originados de uma rede em específico) e a ação define qual atitude deve ser tomada quando um pacote bater com a regra em questão (Ex: recusar pacotes de origens duvidosas). O netfilter irá processar todas as regras sequencialmente, e quando encontrar uma que especifique um dado pacote, ele atuará com a ação par daquela regra. Caso Não encontre uma regra, a regra Default da chain será utilizada. As ações podem ser terminativas ou não. Por exemplo, uma ação que diz ao netfilter para ignorar um pacote é executada e nenhuma outra é executada. Esta é uma ação terminativa. Por outro lado, uma ação especificando para apenas notificar a existência de um pacote, faz seu papel e diz ao netfilter para continuar processando as demais regras (Ex: fazer log quando certo pacote passa pela máquina).

3.1 Tabelas / Tables

O nome iptables vem do fato de que esta ferramenta trabalha baseada em tabelas, onde cada tabela é especializada em um certo tipo de tratamento para os pacotes. Normalmente as tabelas existentes são raw, nat, mangle e filter, sendo a tabela default a filter.

*T*abela	*D*escrição
Raw	Faz pequenas mudanças nos pacotes, usualemte utilizada para colocar marcas.
Nat	Muda os cabeçarios dos pacotes, usualmente utilizada para NAT [1].
Mangle	Realiza alterações especificas nos pacotes, usualmente utilizada para aplicar TOS [2]
Filter	Utilizada para Filtragem dos pacotes.

3.2 Cadeias / Chains

No Iptables existe o conceito de cadeias ou "chains". Essas cadeias nada mais são que a divisão das regras do iptables em conjuntos mais lógicos, para melhorar o entendimento de como o Netfilter processa os pacotes. Este conceito foi implementado já no Ipchains, mas foi melhorado no iptables.

*C*adeia	*D*escrição
INPUT	Tráfego que tem como destino a própria máquina
FORWARD	Tráfego que passante/repassado pela máquina
OUTPUT	Tráfego gerado localmente, tanto como destino local quanto remoto
PREROUTING	Utilizada no tráfego ingressante na máquina [3]
POSTROUTING	Utilizada em todo tráfego de saída da máquina [4]

Observação: A cadeia FORWARD tem um tratamento especial no kernel do Linux, e vem com uma trava fora do firewall, que por padrão bloqueia

[1] Network Address Translation

[2] Type Of Service: http://en.wikipedia.org/wiki/TOS

[3] O tráfego gerado localmente com destino a própria maquina também é incluso

[4] Inclusive o tráfego gerado localmente com destino local

tráfego por ela. Para permitir seu funcionamento, é necessário configurar o parâmetro *net.ipv4.ip_forward* do kernel do Linux. Para fazer isto, basta utilizarmos da ferramenta sysctl :

```
# sysctl -w net.ipv4.ip_forward=1
```

Para verificarmos se o parâmetro foi especificado corretamente basta verificarmos o arquivo ip_forward :

```
cat /proc/sys/net/ipv4/ip_forward
```

Obs: Este bit de controle é resetado no boot para 0. Para resolver este problema basta editar o arquivo */etc/sysctl.conf* e descomentar a linha:

```
# net.ipv4.ipi_forward = 0
```

Mudando para :

```
net.ipv4.ipi_forward = 1
```

3.3 Relação Tables e Chains

Como foi dito antes, a ferramenta Iptables trabalha com tabelas e o conceito de cadeias. Uma tabela pode ter uma ou mais abstrações de cadeias, podemos discriminar essa relação deste modo:

*T*abela	*C*adeias
Raw	PREROUTING, OUTPUT
Nat	PREROUTING, OUTPUT, POSTROUTING
Mangle	INPUT, FORWARD, POSTROUTING, PREROUTING, OUTPUT
Filter	INPUT, FORWARD, OUTPUT

3.4 Fluxo de tratamento

Fluxo de como o tráfego de pacotes é tratado pelo Iptables apartir do kernel 2.6.17:

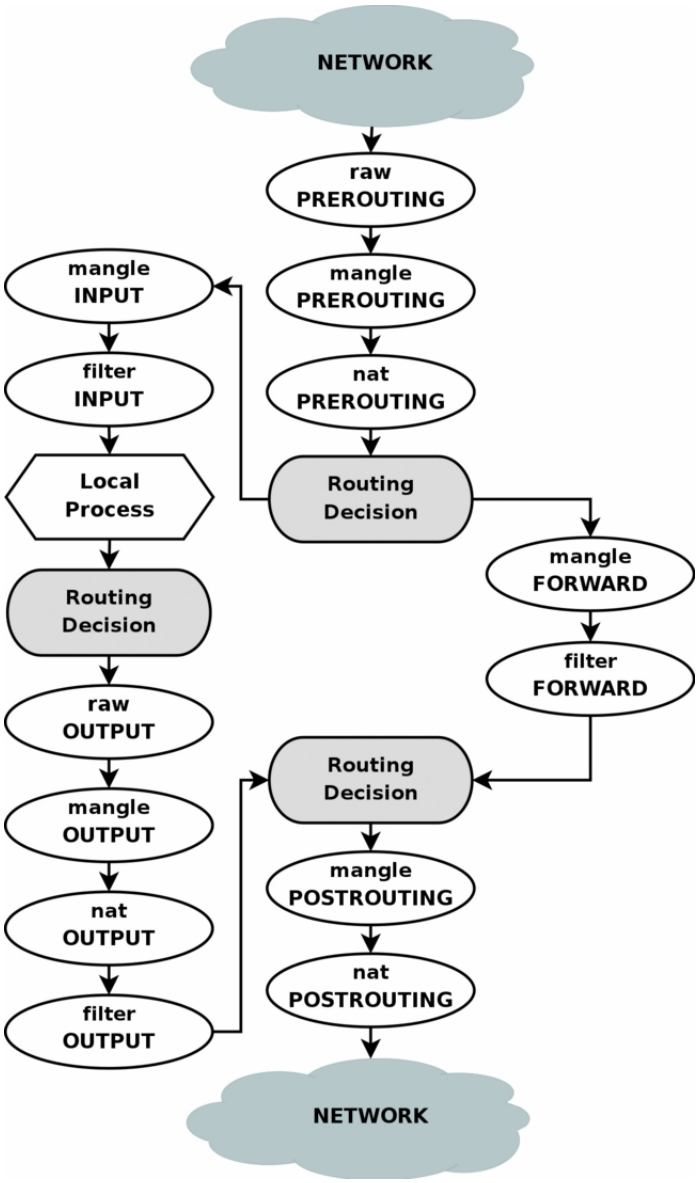

Figura 1:
Fluxo de tratamento de pacotes

3.5 Política Default

Política default do iptables consiste na regra que será utilizada caso algum pacote não se encaixe em nenhuma das regras estabelecidas. Recomenda-se que a política default seja DROP, ou seja, tudo o que não for expressamente permitido será descartado (proibido).

3.6 Targets

Toda regra tem um target, ele nada mais é do que a ação tomada pela regra em relação ao pacote. As possibilidades são:

*T*arget	*D*escrição
ACCEPT	O paacote será aceito.
DROP	O pacote será destruido.
REJECT	O pacote será rejeitado e uma mensagem icmp será enviada à origem.
USER_CHAIN	O pacote será enviado para outra chain.

4 Utilização

4.1 Comandos para manipular chains

*C*omando	*D*escrição
-N [chain]	Cria uma chain
-X [chain]	Apaga uma chain
-P [chain] [target]	Muda a política default de uma chain
-L [chain]	Lista as regras de uma chain
-F [chain]	Apaga todas as regras de uma chain
-Z [chain]	Limpa todos os contadores de bytes e pacote de uma chain

4.2 Comandos para manipular regras de chains

*C*omando	*D*escrição
-A [chain]	Acrescenta uma regra a uma chain
-I [chain] [rulenum]	Insere uma regra numa posição da chain
-R [chain] [rulenum]	Troca posição de regra na chain
-D [chain]	Apaga regra de uma chain

4.2.1 Parâmetros de especificação de regra

Os seguintes parâmetros são os mais utilizados para especificações em regras, regularmente utilizados nos comandos: add, delete, insert, replace e append.

*P*arâmetro	*D*escrição
-s [address]	Especifica o endereço de origem
-d [address]	Especifica o endereço de destino
-p [protocolo][5]	Especifica o protocolo
-i [input_name]	Especifica a interface de entrada dos pacotes
-o [output_name]	Especifica a interface de saida dos pacotes
-f	Indica que a regra se aplica só a fragmentos a partir do segundo pacote
-j [target]	Indica qual a ação/target deve ser tomada caso a regra seja ativada
-g [chain]	Especifica que o processo deve continuar em uma outra user-chain especifica
-m [modulename]	Especifica o uso de um modulo extendido

[5]TCP, UDP, ICMP, ALL

4.2.2 Extensão parâmetro protocolo (-p)

Os protocolos possíveis são TCP,UDP,ICMP,ALL. Cada um deles possuie suas próprias extensões:

TCP:

Para este protocolo as extensões são:

*E*xtensão	*D*escrição
–tcp-flags [mask] [set][6]	Mask indica as *flags* que serão "vigiadas"e o *set* indica quais serão selecionadas.
–syn	Examina a flag TCP SYN
–sport [port]	Indica a porta TCP da origem
–dport [port]	Indica a porta TCP de destino

UDP:

Para este protocolo as extensões são:

*E*xtensão	*D*escrição
–sport [port]	Indica a porta UDP de origem
–dport [port]	Indica a porta UDP de destino

ICMP:

Para este protocolo as extensões são:

*E*xtensão	*D*escrição
–icmp-type [typename][7]	Examina os tipos de icmp

[6] As Mascaras e o set/conjunto podem ser: SYN,ACK,FIN,RST,URG,PSH,ALL,NONE

[7] Ver as variáveis *name* na tabela ICMP do apêndice

4.2.3 Extensão parâmetro match module (-m)

Os modulos mais utilizados são MAC, OWNER, STATE. Cada um deles abre uma nova gama de comandos extras:

MAC:

Para este modulo extensões são:

*E*xtensão	*D*escrição
–mac-soure [address]	Examina o Ethernet MAC address do pacote

OWNER:

Para este modulo extensões são:

*E*xtensão	*D*escrição
–uid-owner [userid]	Aceita pacotes criados pelo user id
–gid-owner [groupid]	Aceita pacotes criados pelo group id
–pid-owner [processid]	Aceita pacotes criados pelo processo pid

STATE:

Para este modulo extensões são:

*E*xtensão	*D*escrição
–state [state]	Analisa o estado dos pacotes

Obs: Os estados possíveis são:

- NEW: Indica pacote que cria uma nova conexão.
- ESTABLISHED: Indica um pacote que pertence a uma conexão já existente.
- RELATED: Indica um pacote relacionado com uma conexão já eexistente.
- INVALID: Indica um pacote não identificado.

5 Exemplos

Os comandos de iptables devem ser executados em um shell com permissão de super usúario (root). Portanto a inserção de todas as regras pode ser feita via shell-script. Alguns exemplos simples do funciomanemto do iptables são:

5.1 Listando as regras

A tabela default do iptables é a FILTER, portanto para mostrar as regras de cada chain desta tabela basta executar:

```
# iptables -L
```

Se não há ainda nenhuma regra nesta tabela a saida deverá ser algo como :

```
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
```

Para listar as regras de cada chain de uma outra tabela específica, deve-se explicitar-la com o parâmetro *-t*:

```
# iptables -L -t mangle
```

Saída:

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

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

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```

Como esperado, mostrou as cinco chains que a tabela mangle implementa.

5.2 Bloqueando uma porta

Uma das funcionabilidades mais utilizadas no iptables é o bloqueio de portas. Para este exemplo, digamos que queremos bloquear a porta ssh (22) de nossa maquina.O comando seria:

```
# iptables -A INPUT -p tcp –dport 22 -j DROP
```

ou simplesmente:

```
# iptables -A INPUT -p tcp –dport 22 -j REJECT
```

Obs: As duas regras possuem praticamente a mesma finalidade, bloquear o acesso externo a porta 22 de nossa maquina. Porem na regra número um, o iptables simplesmente discarta o pacote para essa porta, na regra número dois, ele descarta o pacote mas envia uma notificação à origem deste pacote avisando que ele não foi aceito.

Lembre-se também que pelo fato de não especificarmos uma tabela, a *filter* é utilizada.

5.3 Apagando uma regra

As duas formas mais utilizadas para apagar regras são: simplesmente apagar todas as regras de uma tabela de uma unica vez ou apagar uma regra específica de uma chain de uma determinada tabela. Para apagar todas as regras de uma tabela, basta utilizarmos o iptables com a seguinte sintaxe:

```
# iptables -t [table] -F
```

Lembrando que se não for especificado uma table, a *filter* será utilizada. Agora, imagine que sua tabela filter esteja do seguinte modo :

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere            tcp dpt:smtp
DROP       tcp  --  anywhere             anywhere            tcp dpt:http
DROP       tcp  --  anywhere             anywhere            tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Digamos que você deseja apenas remover a regra de DROP da porta http (80).Para isto:

```
# iptables -t filter -D INPUT 2
```

A sintaxe geral para este tipo de remoção de regra é:

```
# iptables -t [table] -D [chain] [ruleid]
```

Onde o *ruleid* indica qual é a regra dentro da chain. Este indentificador é apenas o número da linha em que a regra se encontra na chain contado de cima para baixo.

5.4 Forward de uma porta para uma maquina local

Outra funcionabilidade comum para o iptables é o redirecionamento de porta. Digamos que a topologia da rede necessite que quando chegar uma requisição na porta 80 do host "A"seja encaminhado para o host 192.168.1.3 nesta mesma porta. Para realizarmos esta função, temos que alterar as regras de iptables do host "A", devemos lembrar que possivelmente a política default das tables sejam de DROP, portanto temos que primeiramente liberar a porta:

```
# iptables -A INPUT -p tcp –destination-port 80 -j ACCEPT
```

```
# iptables –table nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Feito isto resta apenas realizar o redirecionamento propriamente dito:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp –dport 80 -j DNAT –to-dest 192.168.1.3:80
# iptables -A FORWARD -p tcp -i eth0 –dport 80 -d 192.168.1.3 -j ACCEPT
```

Obs: Deve-se lembrar de que o ip_forward deve estar definido como "1"para q isto funcione.

5.5 Defindo regra Padrão

As regras padrões, como dito antes, definem as atitudes que devem ser tomadas quando não é encontrada uma regra para um pacote específico em uma chain. Para definir estas politicas padrões basta seguir o modelo:

```
# iptables -P [chain] [target]
```

Por exemplo para definir como política padrão o DROP para a chain INPUT:

```
# iptables -P INPUT DROP
```

5.6 Liberando Ping para um host específico

Como a política padrão normalmente é DROP, digamos que desejamos que apenas o host 192.168.1.2 possa receber respostas de ping da nossa maquina. Para isto basta executarmos:

```
# iptables -A INPUT -s 192.168.1.2 -i eth0 -p icmp -m icmp –icmp-type 8 -j ACCEPT
```

6 Apêndice

6.1 Tabela ICMP

*T*ype	*N*ame	*R*eference
0	Echo Reply	[RFC792]
1	Unassigned	[JBP]
2	Unassigned	[JBP]
3	Destination Unreachable	[RFC792]
4	Source Quench	[RFC792]
5	Redirect	[RFC792]
6	Alternate Host Address	[JBP]
7	Unassigned	[JBP]
8	Echo	[RFC792]
9	Router Advertisement	[RFC1256]
10	Router Solicitation	[RFC1256]
11	Time Exceeded	[RFC792]
12	Parameter Problem	[RFC792]
13	Timestamp	[RFC792]
14	Timestamp Reply	[RFC792]
15	Information Request	[RFC792]
16	Information Reply	[RFC792]
17	Address Mask Request	[RFC950]
18	Address Mask Reply	[RFC950]
19	Reserved (for Security)	[Solo]
20-29	Reserved (for Robustness Experiment)	[ZSu]
30	Traceroute	[RFC1393]
31	Datagram Conversion Error	[RFC1475]
32	Mobile Host Redirect	[David Johnson]
33	IPv6 Where-Are-You	[Bill Simpson]
34	IPv6 I-Am-Here	[Bill Simpson]
35	Mobile Registration Request	[Bill Simpson]
36	Mobile Registration Reply	[Bill Simpson]
37	Domain Name Request	[RFC1788]
38	Domain Name Reply	[RFC1788]
39	SKIP	[Markson]
40	Photuris	[RFC2521]
41	ICMP messages utilized by experimental mobility protocols such as Seamoby	[RFC4065]
42-255	Reserved	[JBP]

Referências

[1] http://en.wikipedia.org/wiki/Type_of_Service

[2] http://www.faqs.org/docs/iptables/

[3] http://ornellas.apanela.com/dokuwiki/pub:pt-br:linuxfwrt

[4] I Workshop do POP-MG Firewall/Iptables

7 Créditos

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Raphael Ottoni Santiago Machado de Faria