Opções avançadas do *iptables*

Miguel Frade

Departamento de Engenharia Informática
Instituto Politécnico de Leiria

Estados das ligações

Podemos especificar estados com `-m state --state <estado>`
Os estados possíveis são:

- `NEW` para referir uma nova ligação
- `ESTABLISHED` para referir ligações já aceites
- `RELATED` para se referir a pacotes relacionados com ligações aceites, por exemplo FTP
- `INVALID` pacotes que não são início de ligações, nem estão relacionados com ligações activas
- `UNTRACKED` pacotes que não estão a ser monitorizados pela máquina de estados devido ao uso do alvo `NOTRACK` na tabela `raw` (fora do âmbito desta apresentação)

Modo *Statefull*

Para especificar regras em modo *statefull*

- Adicionar regras statefull genéricas

```
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

- Não faz sentido colocar **tudo** no modo *statefull*
- Estas regras genéricas devem ser escritas antes das regras *statefull* e depois das *stateless*

```
# inicio da script
(...)
# regras stateless
(...)
regras statefull genericas
# regras statefull adicionais
```

Modo *Statefull*

Exemplo `ssh` cliente

Modo *stateless*

```
$IPT -A OUTPUT -p tcp --sport 1024:65535 --dport ssh -j
    ACCEPT
$IPT -A INPUT -p tcp --sport ssh --dport 1024:65535 -j
    ACCEPT
```

Modo *Statefull*

```
$IPT -A OUTPUT -p tcp --sport 1024:65535 --dport ssh -m
    state --state NEW -j ACCEPT
# o retorno do pacote sera assegurado pela regra generica:
# $IPT -A INPUT -m state --state ESTABLISHED,RELATED -j
    ACCEPT
```

Modo *Statefull*

No FTP é necessário carregar primeiro um módulo para poder funcionar no modo *statefull*

Exemplo

```
# carregar modulo
/sbin/modprobe ip_conntrack_ftp
# FTP cliente
$IPT -A OUTPUT -p tcp --sport 1024:65535 --dport ftp -m
    state --state NEW -j ACCEPT
```

Modo *Statefull*

Negar pacotes inválidos

- é opcional (serão negados pela política pré-definida), mas
- ajuda a aumentar o desempenho da firewall ao serem negados no início da *script*

Negar pacotes inválidos

```
$IPT -A INPUT -m state --state INVALID -j DROP
```

- normalmente é usado para fazer `LOG` antes do `DROP`

Adicionar Listas

Para além das listas pré-definidas

- `INPUT`, `OUTPUT`, `FORWARD`, `...`

Podemos adicionar novas listas

Exemplo

```
# cria lista
iptables -N NovaLista
# adicionar regras na lista
iptables -A NovaLista -p tcp --dport http -j ACCEPT
(...)
# direcciona pacotes para a lista
iptables -A OUTPUT -p ip -j NovaLista
```

Evita a repetição de regras

Remover Listas

Para eliminar **todas** as listas personalizadas:

```
# apaga TODAS as listas personalizadas
iptables -X
```

Esta linha deve ser adicionada no **início** da *script*

```
# (...)
iptables -F # apaga as regras
iptables -X # apaga as listas
# (...)
```

Alvos

Além do `ACCEPT` e `DROP` existem mais opções:

- `<Nome lista personalizada>`
- `RETURN`
- `LOG`
- `REJECT`
- existem mais, procurar em `man iptables`

RETURN

O `RETURN` permite recolocar um pacote na verificação de regras da lista anterior.

Exemplo

```
iptables -N ListaNova
# se for TCP volta ao INPUT para SEGUINTE
iptables -A ListaNova -p tcp -j RETURN
# se for UDP descarta-se
iptables -A ListaNova -p udp -j DROP

iptables -A INPUT -p ip -j ListaNova
# SEGUINTE depois do RETURN continua aqui
```

LOG

O `LOG` para guardar registo das ligações

- podemos especificar o que queremos registar

Exemplo

```
iptables -A logdrop2 -j LOG --log-prefix "DROPPED "
    --log-level 4 --log-ip-options  --log-tcp-options
    --log-tcp-sequence
```

Consultar `iptables -j LOG -h` para ver mais opções

Ver os últimos registos em tempo real (`ctr+c` para terminar)

```
tail -f /var/log/kern.log
```

REJECT

Com o `REJECT`, tal como o `DROP`, o pacote é descartado, mas o emissor será notificado

- é possível especificar qual será a notificação

Exemplo

```
# TCP
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset

# UDP
iptables -A INPUT -p udp -j REJECT --reject-with
    icmp-port-unreachable
```

- Para mais opções consultar `iptables -j REJECT -h`

ICMP

Quando especificamos o protocolo `ICMP` é possível destinguir o diversos tipos de pacotes.

Exemplos

```
iptables -A INPUT -p icmp --icmp-type
    destination-unreachable -j ACCEPT
iptables -A INPUT -p icmp --icmp-type time-exceeded -j
    ACCEPT
iptables -A INPUT -p icmp --icmp-type parameter-problem -j
    ACCEPT
```

- Existem mais tipos
- Consultar `iptables -p icmp -h`
- Qual é o tipo adequado ao `ping` ?

Limites

- Pode-se limitar o número de vezes que uma regra é aplicada:
  `-m limit`
- Especificar a taxa: `--limit 3/hour`
  - Unidades de tempo reconhecidas: `second`, `minute`, `hour` e `day`
- Especificar a quantidade a partir da qual a taxa é aplicada
  `--limit-burst 5`
- Valores por omissão
  - `--limit 3/hour`
  - `--limit-burst 5`

Limites

Exemplo

```
iptables -A INPUT -m limit --limit 3/hour --limit-burst 5
    -j LOG
```

- Os primeiros 5 pacotes são registados no `LOG`, o `limit-brust` ainda não foi atingido
- Depois é aplicada a taxa de `3/h`, ou seja só são registados pacotes de 20 em 20 minutos

Para que é que serve esta regra?

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

iptables-save

Guardar as regras que estão activas

Sintaxe

```
iptables-save [-c] [-t tabela]
```

- `-c` guarda os valores dos contadores
- `-t tabela` guarda apenas as regras da tabela especificada
- o resultado é enviado para o `stdout`

Exemplo

```
iptables-save -c > /tmp/iptables.txt
```

iptables-restore

Repor as regras guardadas com o `iptables-save`

Sintaxe

```
iptables-restore [-c] [-n]
```

- `-c` restaura os valores dos contadores
- `-n` não apaga as regras que já estejam activas
- os valores são lidos do `stdin`

Exemplo

```
iptables-restore -c < /tmp/iptables.txt
```

Referências

Para saber mais consulte:

- `man iptables`
- `man iptables-save`
- `man iptables-restore`
- `iptables <opção> -h`
- Links úteis
  - Linux 2.4 Packet Filtering HOWTO
  - Iptables Tutorial 1.2.1 by Oskar Andreasson

Exercício - Modo Statefull

Altere a script da aula anterior para:

1. remover listas que já existam antes de adicionar novas regras
2. adicionar as regras genéricas *statefull*
3. os serviços TCP e UDP devem passar a ter regras *statefull*

Exercício - Listas e Alvos

Continue a script anterior para:

1. criar a lista `RegistaLog`
   - a informação registada deve ter o prefixo `"RegistaLog "`
   - todas os pacotes da `RegistaLog` devem ser aceites
   - redireccione todos os serviços `TCP` para a lista `RegistaLog`
2. a sua máquina não tem servidor de `DNS`, por isso deve rejeitar quaisquer pedidos `TCP` ou `UDP` que receba, enviando uma resposta adequada

Exercício - Ping e Limites

Faça uma script nova só para este exercício, para:

1. inicie a script com a política por omissão "negar tudo"
2. permita acesso total à interface `loopback`
3. permita **apenas** os pacotes do `ping` em modo `stateless` (especificar o `icmp-type`)
   - o seu computador como cliente deve conseguir fazer `ping` apenas de 10 em 10 segundos
   - os primeiros 4 pacotes devem ser aceites antes de aplicar o limite
     - $\ldots \rightarrow$ aceita 1 $\rightarrow$ rejeita 10 $\rightarrow$ aceita 1 $\rightarrow$ rejeita 10 $\rightarrow \ldots$
   - Depois altere a script para que o comportamento passe a ser o inverso
     - $\ldots \rightarrow$ rejeita 1 $\rightarrow$ aceita 10 $\rightarrow$ rejeita 1 $\rightarrow$ aceita 10 $\rightarrow \ldots$

Exercício - Ping e Limites

Resultado pretendido

```
$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_req=1 ttl=127 time=0.229 ms
64 bytes from 192.168.0.1: icmp_req=2 ttl=127 time=0.227 ms
64 bytes from 192.168.0.1: icmp_req=3 ttl=127 time=0.165 ms
64 bytes from 192.168.0.1: icmp_req=4 ttl=127 time=0.224 ms
ping: sendmsg: Operation not permitted
(...)
ping: sendmsg: Operation not permitted
64 bytes from 192.168.0.1: icmp_req=11 ttl=127 time=0.221 ms
ping: sendmsg: Operation not permitted
(...)
ping: sendmsg: Operation not permitted
64 bytes from 192.168.0.1: icmp_req=21 ttl=127 time=0.223 ms
ping: sendmsg: Operation not permitted
(...)
```

Exercício Final 1

Implemente uma script com `iptables` com a política negar tudo, coloque em modo `statefull` as regras que se adequarem

1. permitir acesso total ao `loopback`
2. permitir a resolução de nomes
3. permitir obter as configurações IP por `DHCP`
4. permitir acesso ao `http`, `https` e `ssh` como cliente
5. permitir acesso ao `ssh` como servidor, excepto para a rede `192.168.226.0/24` para os quais deve ser enviado um pacote `tcp-reset`
6. permitir a entrada e saída de:
   - `ping`, `destination-unreachable`, `time-exceeded` e `parameter-problem`

(continua)

Exercício Final 1

(continuação)

7. prevenir ataques por `syn flood` limitando o acesso de entrada ao máximo de 10 pedidos por segundo
8. usar listas para:
   - nos serviços `TCP` aceites devem ser guardado em `LOG` informação sobre os pacotes a cada 10 minutos
   - nos serviços `UDP` aceites devem ser guardado em `LOG` informação sobre os pacotes a cada 5 minutos
9. gravar as regras submetidas no ficheiro `\tmp\firewall.txt`

Nota

a ordem pela qual os pontos devem ser executados na script pode não ser a mesma que aparece no enunciado

Exercício Final 2 - Bloquear facebook

Faça uma script nova para:

- permitir acesso total à `loopback`
- permitir a resolução de nomes via `udp`
- permitir acesso a todos os sítios via `http` e `https`
- bloquear o acesso `http` e `https` ao Facebook

IPs do Facebook

Obter os endereços IP usados pelo Facebook

```
$ whois -h whois.radb.net -- '-i origin AS32934' | grep
    ^route
route:      204.15.20.0/22
route:      69.63.176.0/20
route:      66.220.144.0/20
route:      66.220.144.0/21
route:      69.63.184.0/21
route:      69.63.176.0/21
route:      74.119.76.0/22
(...)
```

- O protocolo `whois` está bloqueado pela firewall da ESTG
- Esta lista muda com frequência, por isso devem repetir este comando regularmente
- ▸ Para saber mais clique aqui

Dicas

Podem usar ciclos `for` para regras quase iguais

Exemplo

```
LISTA="ssh http https"
for servico in $LISTA; do
  echo "  " $servico
  iptables -A OUTPUT -p tcp --dports $servico -j ACCEPT
done
```

Pergunta

A script que fez consegue negar o acesso ao Facebook se for usado um proxy? Justifique.