Cortafuegos (Firewalls) en Linux con iptables

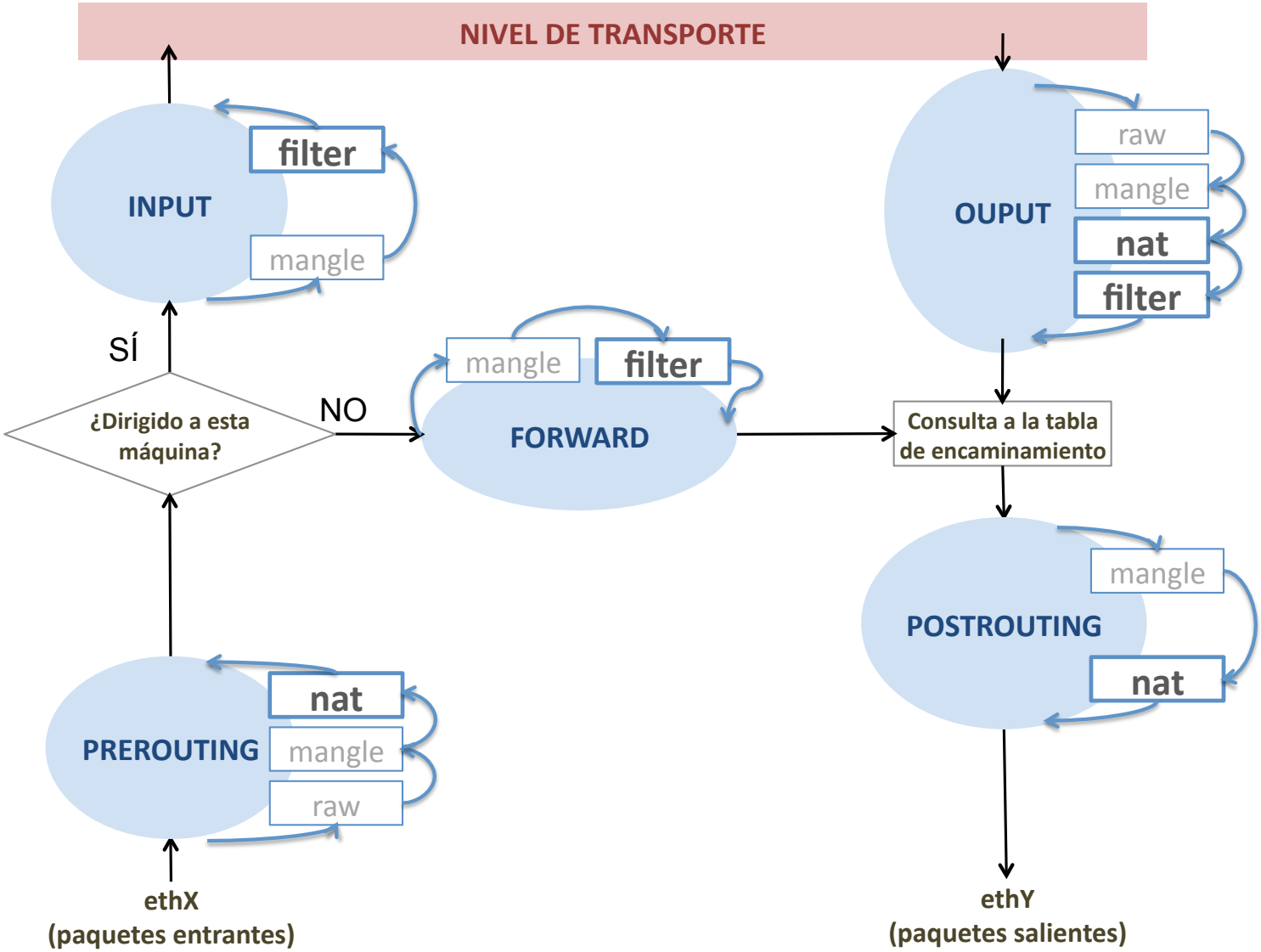
Seguridad en Redes de Ordenadores

GSyC
Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación

Abril de 2018

Contenidos

Movimiento de los paquetes por tablas y cadenas



iptables: comandos

`iptables [-t <tabla>] <comando> [<condición>] [<acción>]`

Si no se especifica una tabla se utilizará por defecto la tabla `filter`.

- Comandos más utilizados para la **configuración de cadenas**:

`iptables [-t <tabla>] -L [<cadena>] [-v] [-n]` lista las reglas definidas en una `cadena` de una `tabla`. Si se omite la cadena el comando actúa sobre todas. Con `-v` se mostrará también el número de paquetes y bytes que han cumplido la condición de cada regla.
`iptables [-t <tabla>] -F [<cadena>]` borra la lista de reglas que hay en una `cadena` de una `tabla`. Si se omite la cadena el comando actúa sobre todas.
`iptables [-t <tabla>] -Z [<cadena>]` reinicia los contadores de una `cadena` de una `tabla`: número de paquetes y bytes que cumplen las condiciones de sus reglas. Si se omite la cadena el comando actúa sobre todas.
`iptables [-t <tabla>] -N [<cadena-usuario>]` crea en una tabla una nueva cadena definida por el usuario.
`iptables [-t <tabla>] -X [<cadena-usuario>]` borra en una tabla la cadena definida por el usuario.
`iptables [-t <tabla>] -P <cadena> <política>` establece la política por defecto para una cadena predefinida de una tabla, donde la política puede ser `DROP` o `ACCEPT`.

iptables: comandos

- Comandos más utilizados para la configuración de **reglas en una cadena**:

```
iptables [-t <tabla>] -A <cadena> <condición> <acción>
```

añade una regla al final de las reglas que tiene definidas una cadena de una tabla. La regla queda definida por la ejecución de una `acción` si un paquete cumple una `condición`. La acción puede ser comenzar la ejecución de una cadena definida por el usuario.

```
iptables [-t <tabla>] -D <cadena> <condición> <acción>
iptables [-t <tabla>] -D <cadena> <numregla>
```

borra una regla de una cadena de una tabla dada su especificación o dado su número de regla.

```
iptables [-t <tabla>] -R <cadena> <numregla> <condición> <acción>
```

reemplaza la regla número `numregla` de una cadena por una nueva regla.

```
iptables [-t <tabla>] -I <cadena> <numregla> <condición> <acción>
```

inserta una regla en la posición `numregla` en una cadena de una tabla.

iptables: condiciones

- Condiciones:

Interfaz	`-i <interfaz>`: interfaz de entrada `-o <interfaz>`: interfaz de salida
Dirección IP	`-s <dirIP[/máscara]>`: dirección (o direcciones) origen `-d <dirIP[/máscara]>`: dirección (o direcciones) destino
Protocolo	`-p <protocolo>` Se pueden especificar adicionalmente números de puerto: `-p <protocolo> --sport <puerto|puertoInicio:puertoFin>`: puerto origen `-p <protocolo> --dport <puerto|puertoInicio:puertoFin>`: puerto destino
Estado de la conexión[3]	`-m state --state <estado>` situación de un paquete con respecto a la conexión a la que pertenece. Estado: `INVALID`: no pertenece a una conexión existente `ESTABLISHED`: es parte de una conexión existente con paquetes en ambos sentidos `NEW`: es parte de una nueva conexión que aún no está establecida `RELATED`: está relacionado con otra conexión ya existente Ejemplo: un mensaje ICMP de error
Flags TCP	`-p tcp --syn`: segmento SYN
	`-p tcp --tcp-flag <flagsAComprobar> <flagsQueDebenEstarActivados>` flags: `SYN`, `FIN`, `ACK`, `RST`, `PSH`, `URG`, `ALL`, `NONE` Ejemplo: `-p tcp --tcp-flags ALL SYN,ACK` (deben estar activados `SYN`, `ACK` y desactivados `FIN`, `RST`, `PSH`, `URG`)

- La negación de una condición se expresa anteponiendo el caracter `!` al valor de la condición. Ejemplos:

`-p tcp --sport ! 80`	protocolo TCP y puerto origen distinto del 80
`-p ! icmp`	protocolo distinto de icmp

[3] "conexión" en sentido amplio

iptables: acciones (I)

La acción se especifica empezando con `-j`

Tabla `filter`	`-j ACCEPT` se acepta el paquete
	`-j DROP` se descarta el paquete
	`-j REJECT [--reject-with <tipo>]` se rechaza el paquete, informando al origen con un ICMP, se puede especificar el tipo de ICMP, por defecto `icmp-port-unreachable`
Tabla `nat`	`-j SNAT --to-source [<dirIP>][:<puerto>]` Realiza *Source NAT* sobre los paquetes salientes (es decir, se cambia dirección IP y/o puerto origen). Sólo se puede realizar en la cadena `POSTROUTING`. **NOTA: esta regla hace que también se cambie automáticamente la dirección de destino del tráfico entrante de respuesta al saliente de la misma "conexión".**
	`-j DNAT --to-destination [<dirIP>][:<puerto>]` Realiza *Destination NAT* sobre los paquetes entrantes (es decir, se cambia dirección IP y/o puerto destino). Sólo se puede realizar en la cadena `PREROUTING`. Esta regla sólo es necesaria para "abrir puertos", es decir, permitir tráfico entrante nuevo. **NOTA: esta regla hace que también se cambie automáticamente la dirección de origen del tráfico saliente de respuesta al entrante de la misma "conexión".**

iptables: acciones (II)

Todas las tablas	`-j LOG [--log-prefix <texto>]` se guarda información de ese paquete en el fichero de `/var/log/kern.log` anteponiendo la cadena de caracteres `<texto>` y **se continúa con la siguiente regla de la cadena**
	`-j <cadena-de-usuario>` Salta a aplicar al paquete las reglas de una cadena definida por el usuario. **Si termina esa cadena sin cumplirse la condición de ninguna de sus reglas, continuará en la cadena desde la que se saltó, por la regla siguiente a la que hizo la llamada.**

Si en una regla **no se especifica ninguna acción** (no hay cláusula `-j`), si se cumple la condición se actualizan los contadores de paquetes y bytes para la regla, pero **se continúa aplicando la siguiente regla de la cadena para ese paquete**.

Contenidos

Contenidos

Inicialización, políticas por defecto en las cadenas

- Borrar las reglas y cadenas de usuario y reiniciar los contadores:

```
iptables -t filter -F
iptables -t filter -X
iptables -t filter -Z
```

- Definir las políticas por defecto: Descartar cualquier cosa salvo paquetes de salida:

```
iptables -t filter -P INPUT DROP
iptables -t filter -P FORWARD DROP
iptables -t filter -P OUTPUT ACCEPT
```

Contenidos

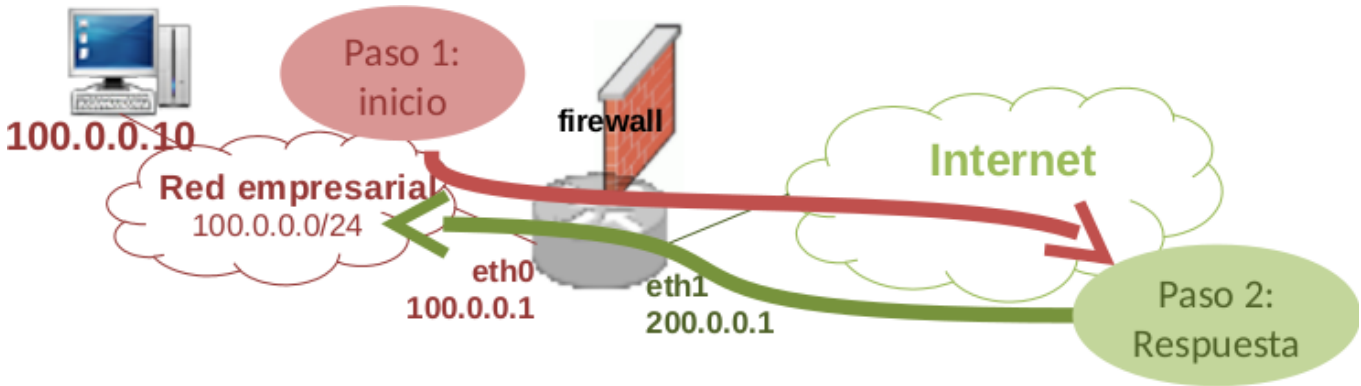
Tráfico saliente de la red empresarial y las respuestas (I)

- Permitir el reenvío de todos los paquetes que se reciben en un router a través de una interfaz (eth0) para que se envíen a través de otra interfaz (eth1) (por ejemplo, permitir tráfico saliente de una organización):

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

- Permitir el reenvío paquetes entrantes que pertenezcan a "conexiones" ya existentes:

```
iptables -t filter -A FORWARD -i eth1 -o eth0 -m state \
         --state RELATED,ESTABLISHED -j ACCEPT
```



Tráfico saliente de la red empresarial y las respuestas (II)

- **Guardar en un fichero de log el contenido de los paquetes que se reenvían**
  Para entender el funcionamiento de cómo se aplican las reglas se puede almacenar en el fichero de log la información de los paquetes a los que se les aplican las reglas que permiten el reenvío del tráfico saliente y las posibles respuestas a dicho tráfico.

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -j LOG \
      --log-prefix "Permitir salida a través de eth1 "
iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT

iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j LOG \
          --log-prefix "Permitir entrada de paquetes de conexiones existentes"
iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

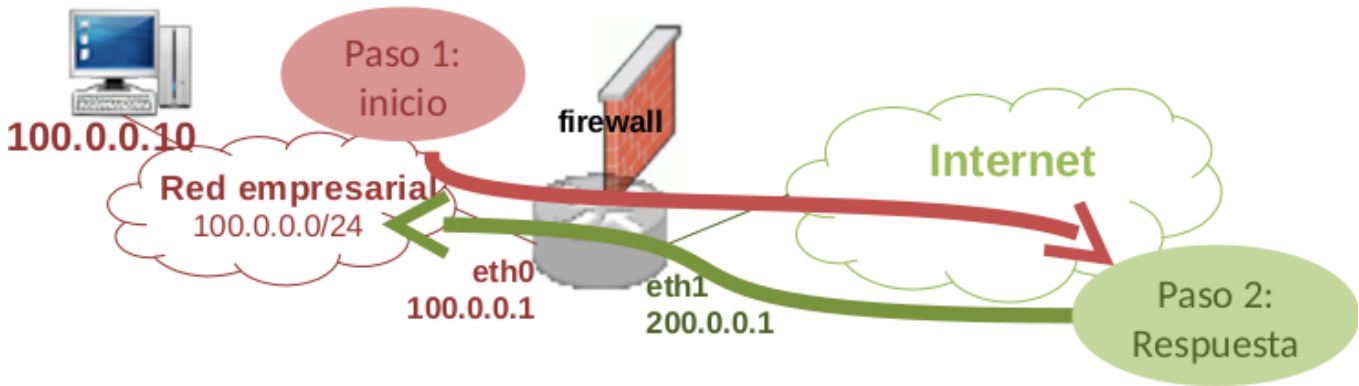
- **Mostrar la información de una configuración**
  Se ha enviado un paquete ICMP *echo request* desde la red empresarial a Internet y se ha recibido respuesta. La información muestra la cantidad de paquetes a los que se les ha aplicado cada regla.

```
firewall:~# iptables -t filter -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target  prot opt  in  out   source  destination

Chain FORWARD (policy ACCEPT 4 packets, 336 bytes)
 pkts bytes target  prot opt  in  out   source  destination
 1    84    LOG     all  --  eth0 eth1 anywhere  anywhere    LOG level warning prefix
                                                             'Permitir salida a través de eth1'
 1    84    ACCEPT  all  --  eth0 eth1 anywhere  anywhere
 1    84    LOG     all  --  eth1 eth0 anywhere  anywhere    state RELATED,ESTABLISHED
                                                             LOG level warning prefix
                                                             'Permitir entrada de paquetes de conexiones existentes'
 1    84    ACCEPT  all  --  eth1 eth0 anywhere  anywhere    state RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target  prot opt  in  out   source  destination
```

Tráfico saliente de la red empresarial y las respuestas (IV)

- **Mostrar el fichero de log**
  Los mensajes generados por la configuración previa de LOG en iptables pueden consultarse al final del fichero `/var/log/messages`. En este caso el LOG debe contener el paquete ICMP *echo request* y el ICMP *echo reply*:

```
firewall:~# less /var/log/messages
Nov  4 19:07:00 r1 kernel: Permitir salida a través de eth1 IN=eth0 OUT=eth1
 SRC=100.0.0.10 DST=12.0.0.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=0 DF PROTO=ICMP TYPE=8
 CODE=0 ID=32522 SEQ=1

Nov  4 19:07:00 r1 kernel: Permitir entrada de paquetes de conexiones existentes IN=eth1
 OUT=eth0 SRC=12.0.0.10 DST=100.0.0.10 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=20573
 PROTO=ICMP TYPE=0 CODE=0 ID=32522 SEQ=1
```



Tráfico saliente de la red empresarial y las respuestas (V)

- **Consultar el sistema de seguimiento ip_conntrack**
  El sistema de seguimiento no muestra ninguna información porque hay una anotación del paquete ICMP *echo request* pero en cuanto se recibe el paquete ICMP *echo reply* se borra la información de dicha "conexión". Por tanto, el siguiente comando no muestra ninguna información.

```
firewall:~# watch -n 1 cat /proc/net/ip_conntrack
```

Nótese que si la máquina a la que se dirige el paquete ICMP *echo request* no responde, el sistema de seguimiento tendría anotado durante un tiempo el paquete ICMP `echo request`:

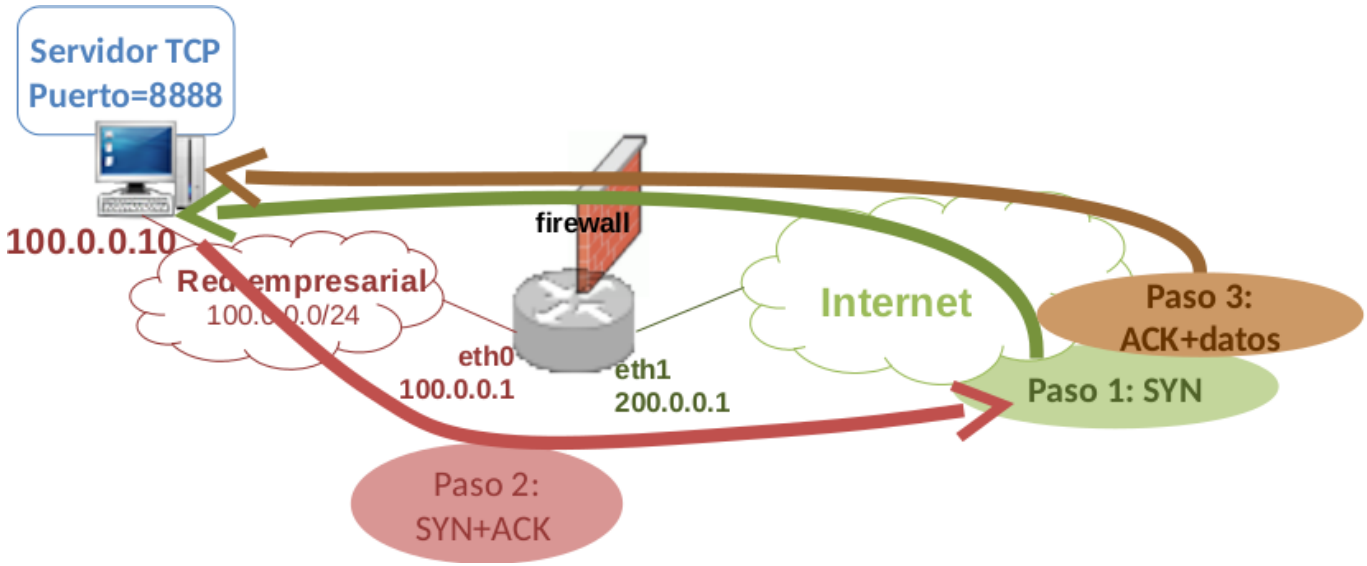
```
firewall:~# watch -n 1 cat /proc/net/ip_conntrack

icmp 1 28 src=100.0.0.10 dst=12.0.0.10 type=8 code=0 id=11023 packets=1 bytes=84 [UNREPLIED]
          src=12.0.0.10 dst=100.0.0.10 type=0 code=0 id=11023 packets=0 bytes=0 mark=0 use=2
```

Contenidos

Tráfico TCP entrante en red empresarial y sus respuestas (I)

- Permitir el paso de segmentos TCP de establecimiento de conexión de entrada dirigidos a una dirección IP de la red empresarial (100.0.0.10) y a un puerto (8888).

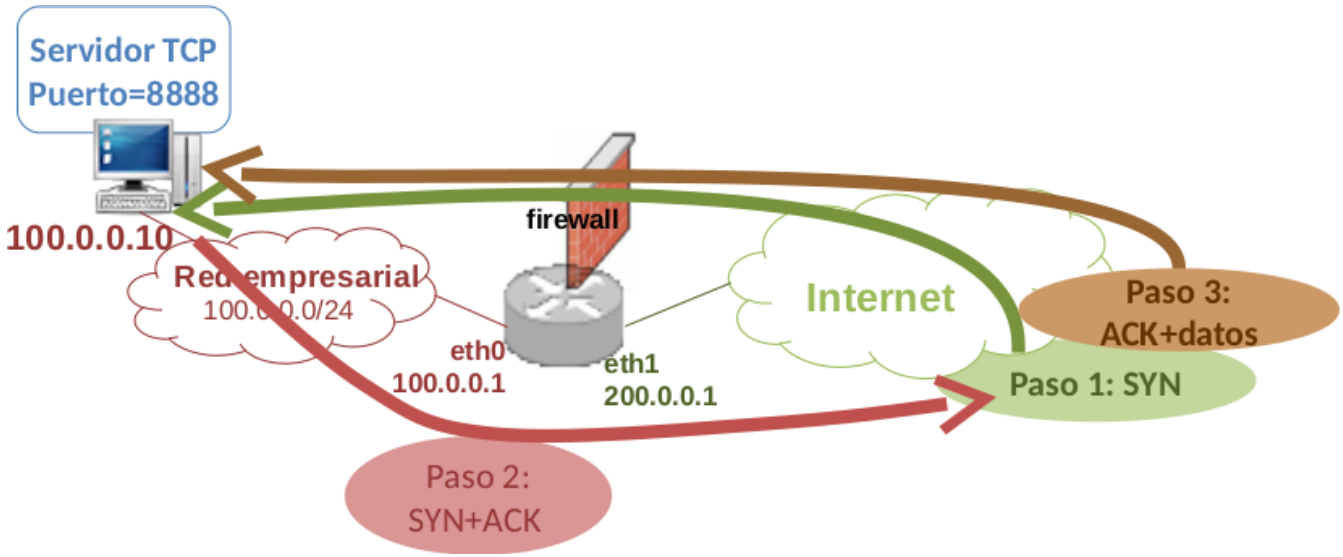
```
iptables -t filter -A FORWARD -p tcp -d 100.0.0.10 --dport 8888 --syn -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT

iptables -t filter -A FORWARD -i eth1 -o eth0 -m state \
              --state RELATED,ESTABLISHED -j ACCEPT
```



Tráfico TCP entrante en red empresarial y sus respuestas (II)

- Configuramos iptables para almacenar los paquetes que cumplan la regla en el fichero de log:

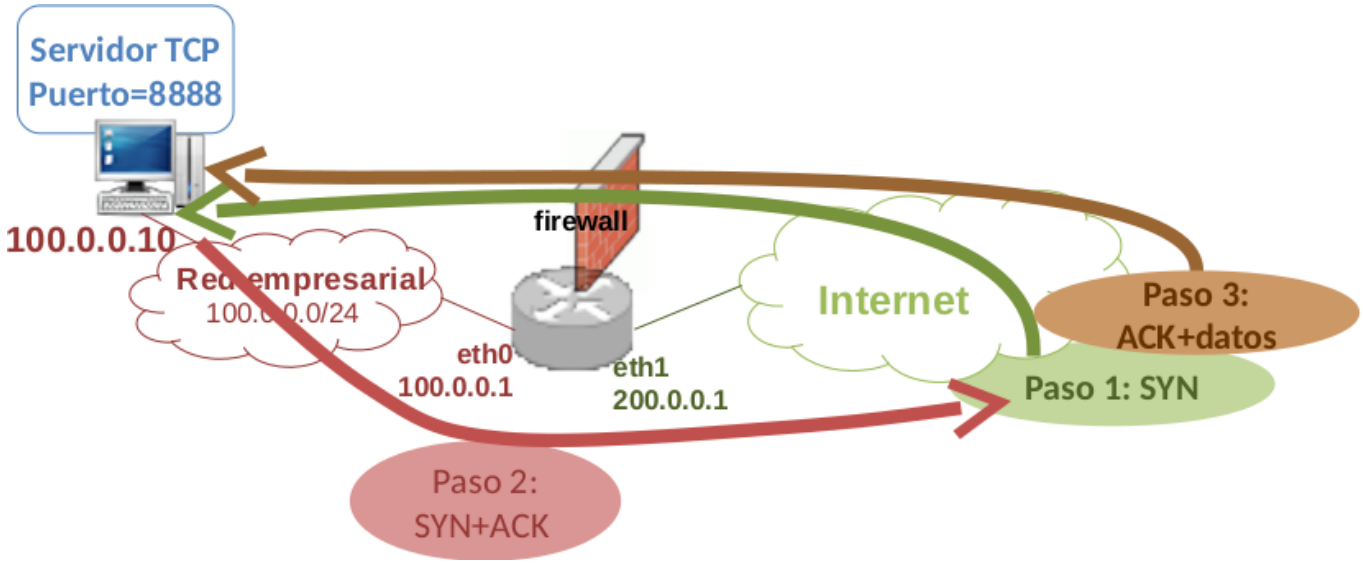
```
iptables -t filter -A FORWARD -p tcp -d 100.0.0.10 --dport 8888 --syn -j LOG \
         --log-prefix "Aceptar conexión TCP al servidor 100.0.0.10:8888"
iptables -t filter -A FORWARD -p tcp -d 100.0.0.10 --dport 8888 --syn -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -j LOG \
      --log-prefix "Permitir salida a través de eth1 "
iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT

iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j LOG \
         --log-prefix "Permitir entrada de paquetes de conexiones existentes"
iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```



Tráfico TCP entrante en red empresarial y respuestas (III)

- Se abre una conexión TCP desde una máquina externa (SYN, SYN+ACK, ACK). Al mostrar el fichero de log:

```
firewall:~# less /var/log/messages
Nov  5 00:31:04 r1 kernel: Aceptar conexión TCP al servidor 100.0.0.10:8888 IN=eth1 OUT=eth0
SRC=12.0.0.10 DST=100.0.0.10 LEN=60 TOS=0x00 PREC=0x00 TTL=63 ID=11521 DF PROTO=TCP SPT=58228
DPT=8888 WINDOW=5840 RES=0x00 SYN URGP=0
Nov  5 00:31:04 r1 kernel: Permitir salida a través de eth1  IN=eth0 OUT=eth1 SRC=100.0.0.10
DST=12.0.0.10 LEN=60 TOS=0x00 PREC=0x00 TTL=63 ID=0 DF PROTO=TCP SPT=8888 DPT=58228 WINDOW=5792
RES=0x00 ACK SYN URGP=0
Nov  5 00:31:04 r1 kernel: Permitir entrada de paquetes de conexiones existentes IN=eth1 OUT=eth0
SRC=12.0.0.10 DST=100.0.0.10 LEN=52 TOS=0x00 PREC=0x00 TTL=63 ID=11522 DF PROTO=TCP SPT=58228
DPT=8888 WINDOW=2920 RES=0x00 ACK URGP=0
```



Tráfico TCP entrante en red empresarial y respuestas (IV)

- Al mostrar el sistema de seguimiento ip_conntrack después del establecimiento de la conexión:

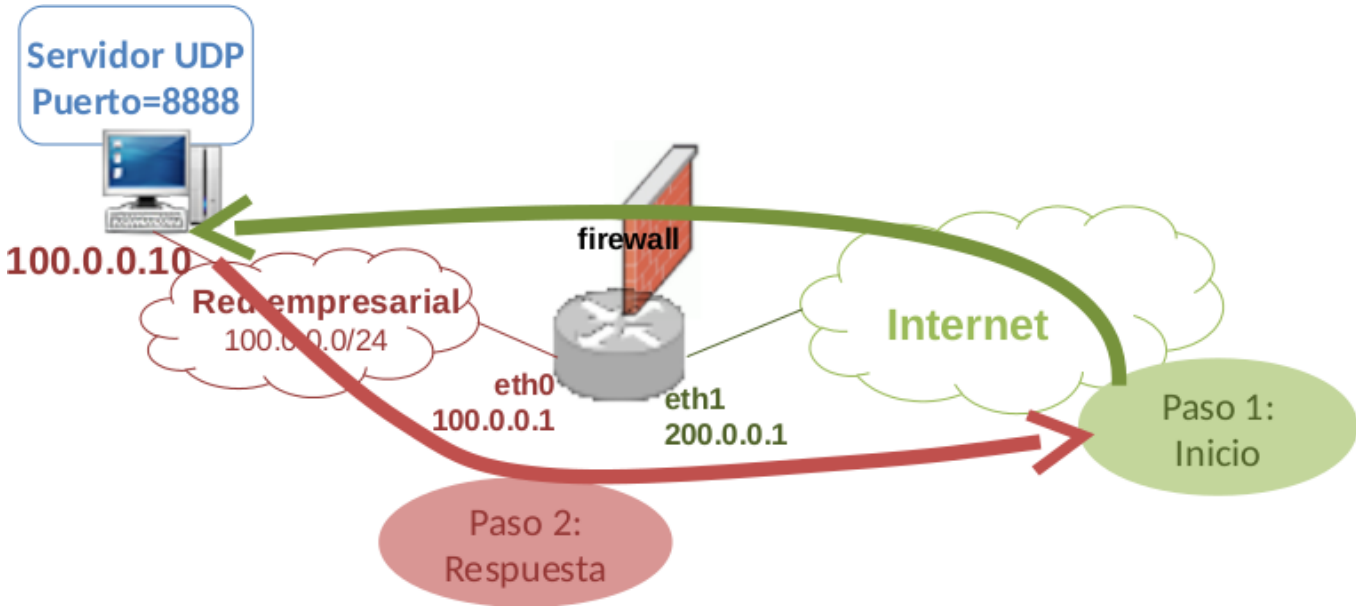
```
firewall:~# watch -n 1 cat /proc/net/ip_conntrack

tcp 6 430970 ESTABLISHED src=12.0.0.10 dst=100.0.0.10 sport=58228 dport=8888 packets=2 bytes=112
   src=100.0.0.10 dst=12.0.0.10 sport=8888 dport=58228 packets=1 bytes=60 [ASSURED] mark=0 use=1
```

Contenidos

Tráfico UDP entrante en red empresarial y sus respuestas (I)

- Permitir el paso de datagramas UDP de entrada dirigidos a una dirección IP de la red interna (100.0.0.10) y a un puerto (8888).

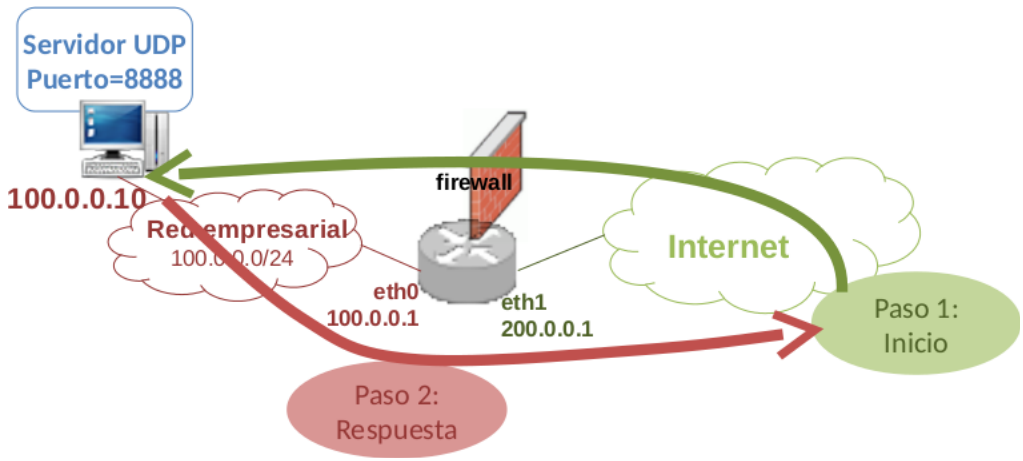
```
iptables -t filter -A FORWARD -d 100.0.0.10 \
    -p udp --dport 8888 -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT
```



Tráfico UDP entrante en red empresarial y respuestas (II)

- Configuramos iptables para almacenar los paquetes que cumplan la regla en el fichero de log:

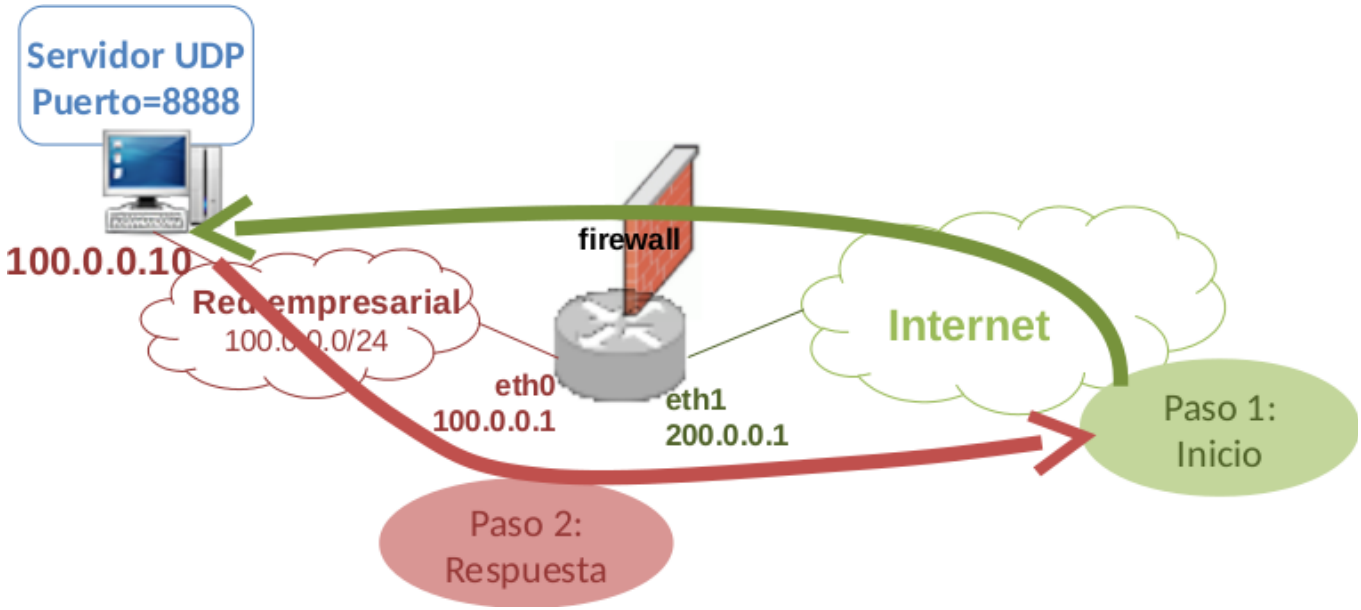
```
iptables -t filter -A FORWARD -p udp -d 100.0.0.10 --dport 8888 \
 -j LOG --log-prefix "Aceptar paquetes UDP al servidor 100.0.0.10:8888"
iptables -t filter -A FORWARD -p udp -d 100.0.0.10 --dport 8888 \
 -j ACCEPT

iptables -t filter -A FORWARD -i eth0 -o eth1 -j LOG \
      --log-prefix "Permitir salida a través de eth1 "
iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT
```



Tráfico UDP entrante en red empresarial y respuestas (III)

- Se envía un paquete de datos UDP desde una máquina externa. Al mostrar el fichero de log:

```
r1:~# less /var/log/messages
Nov  5 00:32:04 r1 kernel: Aceptar paquetes UDP al servidor 100.0.0.10:8888 IN=eth1 OUT=eth0
SRC=12.0.0.10 DST=100.0.0.10 LEN=34 TOS=0x00 PREC=0x00 TTL=63 ID=2916 DF PROTO=UDP SPT=33666
DPT=8888 LEN=14
```



- Al mostrar el sistema de seguimiento ip_conntrack:

```
r1:~# watch -n 1 cat /proc/net/ip_conntrack

udp 17 21  src=12.0.0.10 dst=100.0.0.10 sport=58300 dport=8888 packets=1 bytes=34 [UNREPLIED]
   src=100.0.0.10 dst=12.0.0.10 sport=8888 dport=58300 packets=0 bytes=0 mark=0 use=1
```

Contenidos

Cadena de usuario

- Para crear una nueva cadena, definida por el usuario:
  `iptables -t filter -N miCadena`
- Configurar reglas en una cadena de usuario:

```
iptables -t filter -A miCadena -p tcp \
    --dport 80 -j LOG \
    --log-prefix "Paquete al puerto 80: "
```

- Para ejecutar las reglas de la cadena definida por el usuario:
  `iptables -t filter -A FORWARD  -p tcp -j miCadena`
  Cunado un paquete cumple la condición se ejecutan las reglas de la cadena de usuario `miCadena`. Si la ejecución de todas las reglas de la cadena de usuario termina, se vuelve al mismo punto desde donde se ha llamado a la cadena de usuario.

Referencias

- Iptables Tutorial:
  `http://www.iptables.info`
- Linux Advanced Routing & Traffic Control:
  `http://www.lartc.org`