

```
helo openbeer.it
mail from: -=mayhem=- <mayhem at openbeer dot it>
rcpt to: everyone at openbeer dot it
data
```

**Subject: OpenBSD, PF e due connessioni ad Internet.**

Lo scopo di questo documento non e' guidare alla corretta configurazione del vostro firewall quanto introdurre l'uso del routing basato sul tipo di traffico.

Il problema nasce, ad esempio, nel momento in cui la banda offertaci da una linea a basso costo (es. ADSL) non e' piu' sufficiente, ma non vogliamo spendere un capitale per una linea piu' performante.

Una seconda ADSL ha sicuramente un costo affrontabile, ma il nostro firewall supporta, ovviamente, un solo default gateway per volta.

Per questa ragione dobbiamo trovare un sistema per scegliere quale traffico instradare attraverso una connessione, e quale attraverso un'altra.

Utilizzeremo route-to, una istruzione di PF, il firewall, ma non solo, di OpenBSD, pensato principalmente per il load-balancing, che puo' essere realizzato come descritto qui o come descritto nelle FAQ relative.

Scenario:

```
LAN -- OBSD_BOX --< /-- RouterA
                        \-- RouterB
```

per l'esempio utilizzerò solo indirizzi IP privati:

```
RouterA (ethernet):      10.0.0.1/24
RouterB (ethernet):      10.0.1.1/24
OBSD_BOX xl0 (outsideA): 10.0.0.2/24
OBSD_BOX xl1 (outsideB): 10.0.1.2/24
OBSD_BOX xl2 (inside):   192.168.0.1/24
LAN:                     192.168.0.0/24
```

Chiarito lo scenario cerchiamo di capire meglio il nostro obiettivo: avendo a disposizione 2 linee verso Internet (immaginiamo 2 ADSL) vogliamo suddividere il traffico sulle due linee, magari utilizzando la linea piu' veloce per navigazione/ssh e la piu' lenta per la posta e le news. PF, attraverso l'istruzione route-to, ci permette di fare questo.

Partiamo da un pf.conf assolutamente basilare, fingendo che il routerB e la sua rete non esistano; permettiamo alla LAN di accedere ai servizi sopracitati.

```
# pf.conf - listato 1 - versione base
```

```
outsideA="x10"  
inside="x12"  
routerA="10.0.0.1"  
frwA="10.0.0.2/32"  
lan="192.168.0.0/24"  
tcp_ports="{ 25, 110, 80, 443, 22 }"
```

```
# Nat del traffico dalla LAN sull'IP di x10  
nat on $outsideA from $lan to any -> 10.0.0.2
```

```
# Blocco tutto di default  
block in all
```

```
# Permetto alla LAN di accedere ai servizi su Internet:  
pass in quick on $inside proto tcp from $lan to any port $tcp_ports keep state  
pass in quick on $inside proto udp from $lan to any port 53 keep state
```

```
# Autorizzo il traffico a cui ho applicato il NAT  
pass out quick on $outside from $frwA to any keep state
```

```
# Genero il log per il restante traffico (rifiutato)  
block in log quick from any to any
```

```
# eof pf.conf - listato 1 - versione base
```

Dal precedente file di configurazione, fin troppo semplice, noi ci permettiamo di aggiungere un po' di parametri in ordine sparso, al fine di ottenere il risultato che ci eravamo prefissi.

La parola [chiave] magica e' "route-to" che va inserito in una normale riga del tipo "pass in" e prende come argomento l'interfaccia ed il router a cui instradare tutto il traffico descritto dalla riga in questione.

Facciamo un esempio:

```
pass in quick on $inside proto tcp from $lan to any port 22 keep state
```

Questa riga identifica tutto il traffico ssh in uscita dalla mia rete; ecco come la modifica un route-to:

```
pass in quick on $inside route-to ( x11 routerB ) \  
    proto tcp from $lan to any port 22 keep state
```

che potrebbe essere tradotta con un "permetti tutto il traffico ssh in uscita dalla mia LAN, ma, indipendentemente dalle informazioni contenute nella tabella di routing, instradalo attraverso l'interfaccia x11 e fallo uscire su Internet attraverso il routerB".

Per questa ragione dovremo configurare diversamente il NAT ed avere una interfaccia fisica per ogni router che vogliamo gestire (come sarebbe in realta' corretto).

Il NAT andra' fatto sull'interfaccia, non sull'indirizzo IP, ed il gruppo di porte autorizzate andra' diviso in porte autorizzate su una linea e sull'altra. Faccio notare che se l'interfaccia ha degli alias, ogni connessione prendera' un diverso indirizzo assegnato all'interfaccia, in round-robin.

```
# pf.conf - listato 2 - versione con route-to
```

```
outsideA="x10"  
outsideB="x11"  
inside="x12"  
routerA="10.0.0.1"  
routerB="10.0.1.2"  
frwA="10.0.0.2/32"  
frwB="10.0.1.2/32"  
lan="192.168.0.0/24"  
tcp_portsA="{ 25, 110 }"  
tcp_portsB="{ 80, 443, 22 }"
```

```
# Nat del traffico dalla LAN sull'IP di x10  
nat on $outsideA from $lan to any -> ($outsideA)  
nat on $outsideB from $lan to any -> ($outsideB)
```

```
# Blocco tutto di default  
block in all
```

```
# Permetto alla LAN di utilizzare la posta attraverso la connessione A  
pass in quick on $inside route-to ( $outsideA $routerA ) \  
    proto tcp from $lan to any port $tcp_portsA keep state
```

```
# Permetto alla LAN di navigare attraverso la connessione B
pass in quick on $inside route-to ( $outsideB $routerB) \
    proto tcp from $lan to any port $tcp_portsB keep state

# Le query DNS vengono gestite attraverso il normale routing
pass in quick on $inside proto udp from $lan to any port 53 keep state

# Autorizzo il traffico a cui ho applicato il NAT
pass out quick on $outside from $frwA to any keep state
pass out quick on $outside from $frwB to any keep state

# Genero il log per il restante traffico (rifiutato)
block in log quick from any to any

# eof pf.conf - listato 2 - versione con route-to
```

Il precedente listato ci permette di suddividere il carico di lavoro su due diverse connessioni. Ma immaginiamo di avere due connessioni con le stesse caratteristiche e di voler smistare il traffico dinamicamente, sapendo anche di non avere problemi rispetto all'indirizzo con cui mi presentero' (navigare con un ip o un altro potrebbe essere indifferente, magari non lo e' per una connessione ssh ....).

Per fare questo utilizziamo un'altra feature di route-to, cioe' poter specificare un elenco di gateway da utilizzare. Utilizzando lo scenario proposto, ripartendo dal listato 1, possiamo trasformare la riga:

```
pass in quick on $inside proto tcp from $lan to any port $tcp_ports keep state
```

specificando di usare per ogni connessione un diverso gateway, in round-robin, cioe' nel caso di due router, prima A, poi B, poi A, poi B e cosi' via.

```
pass in quick on $inside proto tcp route-to { ( $outsideA $routerA), \
    ( $outsideB $routerB) } round-robin \
    from $lan to any port $tcp_ports keep state
```

Ecco come si trasformerà il nostro pf.conf in questa variante:

```
# pf.conf - listato 3 - versione con route-to in round-robin
```

```
outsideA="x10"
outsideB="x11"
inside="x12"
routerA="10.0.0.1"
routerB="10.0.1.2"
frwA="10.0.0.2/32"
frwB="10.0.1.2/32"
lan="192.168.0.0/24"
tcp_ports="{ 25, 110, 80, 443, 22 }"
```

```
# Nat del traffico dalla LAN sull'IP di x10
nat on $outsideA from $lan to any -> ($outsideA)
nat on $outsideB from $lan to any -> ($outsideB)

# Blocco tutto di default
block in all

# Permetto alla LAN di accedere ai servizi su Internet
pass in quick on $inside route-to { ( $outsideA $routerA), \
                                     ( $outsideB $routerB) } round-robin \
    proto tcp from $lan to any port $tcp_ports keep state

# Le query DNS vengono gestite attraverso il normale routing
pass in quick on $inside proto udp from $lan to any port 53 keep state

# Autorizzo il traffico a cui ho applicato il NAT
pass out quick on $outside from $frwA to any keep state
pass out quick on $outside from $frwB to any keep state

# Genero il log per il restante traffico (rifiutato)
block in log quick from any to any

# eof pf.conf - listato 3 - versione con route-to in round-robin
```

Nel caso in cui dovessimo avere dei servizi pubblicati su Internet ricordiamo, per semplicita', di pubblicarli tutti, se possibile, sull'indirizzo IP dell'interfaccia a cui appartiene il default gateway.

Se cio' non fosse possibile, basta ricordare di marcare tutto il traffico di ritorno dei servizi pubblicati con un route-to che lo invii alla corretta destinazione.

--

Everyone is gay, with someone's else ass

Reference: OpenBSD PF FAQ - <ftp://ftp.openbsd.org/pub/OpenBSD/doc/pf-faq.txt>

Reference: mayhem - tutte le prove che ho fatto a casa mia ;P

.