Masquerading made simple HOWTO

John Tapsell

thomasNO@SPAMresonancePLEASE.org

Thomas Spellman

thomas@resonance.org

Matthias Grimm

DeadBull@gmx.net

Tutti gli autori sono contattabili sul canale #debian su irc.opensource.net

John Tapsell (JohnFlux) è il curatore ufficiale.

Io (John Tapsell) sono disponibile al contatto via email per qualsiasi domanda, feedback, polemica ed anche per appuntamenti ecc.

Si è rubato senza vergogna dal lavoro di David Ranch - `<dranch@trinnet.net>`.

Questo documento non intende rimpiazzare IP-Masquerading HOWTO ma gli è complementare, i due documenti andrebbero letti fianco a fianco. Non includerò cose di cui si occupa l'altro HOWTO, né darò spiegazioni di carattere generale. Si consulti http://ipmasq.cjb.net e il solito Masq-HOWTO per informazioni di questo tipo.

Questo documento descrive come abilitare la funzionalità di Masquerading IP su un host Linux. IP Masq è una forma di NAT (Network Address Translation) che permette alle macchine di una rete locale che non hanno uno o più indirizzi Internet registrati di comunicare con Internet attraverso delle macchine (Linux, nel nostro caso) dotate di un singolo indirizzo IP pubblico.

Traduzione a cura di Riccardo Fabris (frick at linux.it).

Sommario

1. Introduzione

La faremo breve e andremo dritti al punto.

Si ha una rete locale, che si vuol far comunicare con l'esterno:

2. Sommario: (mi piace partire con i sommari)

Si assuma che la scheda di rete esterna, verso Internet, sia eth0, che l'IP esterno sia 123.12.23.43 e che la scheda di rete interna sia eth1, quindi:

```
$> modprobe ipt_MASQUERADE # Se fallisce, si provi ad andare avanti comunque
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
$> iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 123.12.23.43
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

O, per una connessione telefonica:

```
$> modprobe ipt_MASQUERADE # Se fallisce, si provi ad andare avanti comunque
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
$> iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

Poi, per rendere il tutto più sicuro:

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$> iptables -A INPUT -m state --state NEW -i ! eth0 -j ACCEPT
$> iptables -P INPUT DROP   #solo dopo che i due precedenti hanno avuto successo
$> iptables -A FORWARD -i eth0 -o eth0 -j REJECT
```

Ora, per una connessione telefonica (con eth0 scheda di rete interna):

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$> iptables -A INPUT -m state --state NEW -i ! ppp0 -j ACCEPT
$> iptables -P INPUT DROP   #solo dopo che i due precedenti hanno avuto successo
$> iptables -A FORWARD -i ppp0 -o ppp0 -j REJECT
```

Tutto qui! Per vedere le regole di masquerading si dia "**iptables -t nat -L**"

3. Versione un po' più approfondita

Compilare il kernel (serve un kernel 2.4.x o successivo).

Serve abilitare il supporto alle seguenti funzionalità del kernel:

- Sotto "Networking Options"

  - Network packet filtering (CONFIG_NETFILTER)

- Sotto "Networking Options->Netfilter Configuration"
  - Connection tracking (CONFIG_IP_NF_CONNTRACK)
  - FTP Protocol support (CONFIG_IP_NF_FTP)
  - IP tables support (CONFIG_IP_NF_IPTABLES)
  - Connection state match support (CONFIG_IP_NF_MATCH_STATE)
  - Packet filtering (CONFIG_IP_NF_FILTER)
    - REJECT target support (CONFIG_IP_NF_TARGET_REJECT)

  - Full NAT (CONFIG_IP_NF_NAT)
    - MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE)
    - REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT)

  - Packet mangling (CONFIG_IP_NF_MANGLE)
  - LOG target support (CONFIG_IP_NF_TARGET_LOG)

Per prima cosa, se i moduli iptable e masq non sono incorporati nel kernel né installati, ma esistono come moduli caricabili, bisogna installarli. Con "**insmod ipt_MASQUERADE**" si caricheranno ip_tables, ip_conntrack e iptable_nat.

```
$> modprobe ipt_MASQUERADE
```

L'intranet da collegare a Internet potrebbe essere composta da molte macchine o solo da un paio, in realtà non fa molta differenza.

Bene, sto dando per scontato che non servano altre regole. A questo punto si dia:

```
$> iptables -F; iptables -t nat -F; iptables -t mangle -F
```

Se si ottiene un errore e dice che iptables non si trova, lo si trovi ed installi. Se dice che non c'è alcuna tabella "nat", si ricompili il kernel col supporto a nat. Se dice che non c'è alcuna tabella "mangle", non ci si preoccupi, non è necessaria al masquerading. Se dice che iptables è incompatibile con il proprio kernel, ci si procuri un kernel 2.4 e lo si compili col supporto a iptables.

In caso si abbia un IP statico (p.e. non si usa DHCP), si dia:

```
$> iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 123.12.23.43
```

per un IP dinamico (p.e. un modem, ma ci si deve prima collegare!):

```
$> iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Infine, per dire al kernel che si desidera iniziare a fare il forward dei pacchetti (cosa necessaria solo una volta per reboot, ma a farlo più volte non succede nulla di male), si dia:

```
$> echo 1 > /proc/sys/net/ipv4/ip_forward
```

Una volta che si è controllato che tutto funziona (si vedano le istruzioni di post- installazione più avanti) si deve permettere il masquerading solo dalla rete interna; non si vorrà mica che qualcuno su Internet ne abusi! :)

Per prima cosa, si consentano le connessioni preesistenti o qualunque cosa ad esse correlate (p.e. un server ftp che si collega di rimando):

```
$> iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Se dà un errore, molto probabilmente nel kernel non è abilitato il "connection tracking", perciò lo si ricompili. Poi si permettano nuove connessioni solo dalla propria intranet (rete interna o locale). Si rimpiazzi ppp0 con eth0 o qualunque altra cosa sia il proprio device di rete verso l'*esterno*. (Il "!" è una negazione, significa "qualsiasi cosa tranne").

```
$> iptables -A INPUT -m state --state NEW -i ! ppp0 -j ACCEPT
```

Ora si neghi tutto il resto:

```
$> iptables -P INPUT DROP   #solo se i due precedenti hanno avuto successo
```

Se una delle due regole precedenti è fallita, l'ultima regola impedirà completamente il funzionamento del masquerading. Per annullare l'ultima regola si usi "**iptables -P INPUT ACCEPT**".

4. Istruzioni di post-installazione

Ora dovrebbe funzionare tutto. Non ci si dimentichi di:

- Configurare tutti i client della rete interna affinché usino l'IP interno della macchina Linux come gateway.
- Configurare tutti i client affinché usino un eventuale proxy HTTP del provider, un proxy trasparente (ATTENZIONE: ho sentito dire che i proxy trasparenti possono essere molto lenti in caso di reti molto ampie) o Squid fatto girare sul gateway Linux (un'opzione da prendere in considerazione per grosse reti).
- Ci si assicuri di specificare un server DNS nella configurazione dei client, altrimenti si otterranno errori circa la mancata risoluzione degli indirizzi. Se il DNS funzionava prima di configurare il masquerading e dopo non funziona più, accade perché il server DHCP del provider non può più fornire l'indirizzo del server DNS.

  [OT] Si potrebbe provare semplicemente a inviare un messaggio dhcp broadcast che inoltri ai client l'indirizzo del server DNS (e già che ci siamo anche l'indirizzo del proxy http) senza dover configurare un server DHCP (o anche configurandolo). Mi piacerebbe discuterne con qualcuno via email :)

  Grazie a Richard Atcheson per avermi fatto notare la cosa.

- Ora si dovrebbe iniziare a pensare alla sicurezza! Prima di tutto si disabiliti il forward generico: "**iptables -P FORWARD DROP**", quindi ci si informi su come usare iptables, `/etc/hosts.allow` e `/etc/hosts.deny` per rendere più sicuro il sistema. ATTENZIONE: non usare la regola di iptables summenzionata se non dopo aver accertato che il masquerading effettivamente funzioni. Si deve permettere esplicitamente il passaggio dei pacchetti dove lo si desidera in caso si voglia negare il forward per default. (Il comando si annulla con "**iptables -P FORWARD ACCEPT**").
- Si permetta l'accesso a qualsiasi servizio che si vuole rendere visibile su Internet.

  Per esempio, per permettere l'accesso al web server:

  ```
  $> iptables -A INPUT --protocol tcp --dport 80 -j ACCEPT
  $> iptables -A INPUT --protocol tcp --dport 443 -j ACCEPT
  ```

  Per consentire l'ident (per connessioni ad irc ecc.):

```
$> iptables -A INPUT --protocol tcp --dport 113 -j ACCEPT
```

Per testare la configurazione:

- Provare a connettersi da un client all'indirizzo numerico di un web server. Per esempio uno degli IP di Google è 216.239.33.100. Si dovrebbe ottenere una replica da esso. Per esempio: "**ping 216.239.33.100**" "**lynx 216.239.33.100**".
- Provare una connessione completa con il nome. Per esempio "**ping google.com**" "**lynx google.com**" o usando Internet Explorer o Netscape.

Dove eth0 è la scheda di rete ethernet verso Internet e 123.12.23.43 è l'IP pubblico della macchina.

5. FAQ - le lament... le domande poste più di frequente

- Come posso ottenere un elenco delle regole introdotte?

  - Prova a dare:

  ```
  $> iptables -L
  $> iptables -t nat -L
  ```

- Non risolve gli indirizzi! Ho provato con "www.microsoft.com" ma mi dice che non lo trova!

  - Assicurati di fornire a tutti i client l'IP del server DNS.

- Non funziona! Non accetta iptables / NAT / SNAT / MASQ.

  - Recupera il kernel più recente e compilalo con il supporto completo a iptables e NAT.

- Non funziona! Il masquerading non funziona proprio, mannaggia!

  - Non è che hai tralasciato di dare **echo 1 > /proc/sys/net/ipv4/ip_forward**?!

- Non funziona! Non mi funziona manco la rete, ti odio!

  - Prova con

  ```
  $> iptables -F
  $> iptables -t nat -F
  $> iptables -t mangle -F
  ```

  (così si eliminano tutte le regole), poi rilancia tutte le altre regole di iptables.

  - Prova con **iptables -P FORWARD ACCEPT**

- Non funziona ancora!

- Hmhmhm, "**dmesg | tail**" dà qualche errore? E "**cat /var/log/messages | tail**" ? Non posso mica pensare a tutto io...

- Non ce la faccio, non mi funziona proprio!

  - Boh! ...comunque *prima* di iniziare a sperimentare con il masquerading dovresti assicurarti di:

  ```
  1) poter fare il ping di un indirizzo Internet dal gateway;
  2) poter fare il ping delle macchine della rete interna dal gateway;
  3) poter fare il ping del gateway dalle macchine della rete interna.
  ```

- Dove devo ficcare tutta questa roba?

  - Nel file `/etc/network/interfaces` o firewall.rc. Se lo metti nel file interfaces, mettilo nell'opzione pre-up per l'interfaccia esterna e poni nel post-down "**iptables -t nat -F**".

- Che devo fare per far sì che la connessione ppp venga attivata solo quando serve ("ppp on demand")?

  - Se per esempio l'IP del gateway del provider è 23.43.12.43, aggiungi una riga:

  :23.43.12.43

  in fondo a `/etc/ppp/peers/provider`. (Questo se ti viene attribuito un IP dinamico, in caso di IP statico potresti usare **IP_statico:23.43.12.43**). [La parte sul "ppp on demand" mi pare un po' confusa, p.e. qui, aggiungendo le opzioni ipcp-accept-local e ipcp-accept-remote, si possono usare anche due IP fittizi NdT].

  Quindi aggiungi su una nuova riga in fondo allo stesso file:

  demand

  Pppd rimarrà in background per riconnettersi a richiesta se la connessione cade, fino a quando non darai il comando "**ifdown ppp0**" o "**poff**". Se invece aggiungi l'opzione "**nopersist**", pppd uscirà una volta attivata la connessione. Puoi anche aggiungere su una nuova riga "**idle 600**" per disconnetterti dopo 10 minuti di inattività.

- La connessione continua a cadermi!

  - Innanzitutto usi una connessione a richiesta? Funziona come dovrebbe? Controlla `/etc/ppp/peers/provider` e assicurati che la connessione telefonica funzioni correttamente prima di provare col masquerading.

  - In seconda battuta, se continua a non funzionare potrebbe trattarsi di quel che è capitato a me: per motivi misteriosi sono dovuto ritornare al kernel 2.4.3 affinché tutto funzionasse, ma non so il perché.

- Non ho alcuna voglia di fare tutto a mano! Voglio uno script prefabbricato, una GUI e tutto il resto.

  - Prova questo: http://shorewall.sourceforge.net/ (http://shorewall.sourceforge.net/)

  Struggiti dal dolore!

- I cable modem li devo considerare tra gli IP statici o dinamici?

  - Buona domanda... potrebbero pure essere dinamici.

- Devo considerare le schede di rete DHCP come IP statici o dinamici?

  - Sono dinamici.

- Come posso gestire i servizi con connessioni in entrata?

  - Prova con il forward o la redirezione delle porte - ancora una volta assicurati di configurare il firewall a tal fine se necessario.

- Riesco a fare il ping dell'indirizzo esterno del gateway Linux dai client della rete, ma non riesco ad accedere a Internet.

  - Ok, prova con "**rmmod iptable_filter**"; fornirò maggiori informazioni quando le avrò disponibili.

  - Assicurati che non stai facendo girare *routed* o *gated*, per saperlo lancia "**ps aux | grep -e routed -e gated**".

  - Consulta http://ipmasq.cjb.net.

- Come posso vedere le connessioni stabilite? Una cosa tipo netstat...

  - prova con "**cat /proc/net/ip_conntrack**"

- Mi servono maggiori informazioni su squid, l'instradamento e roba simile!

  - Dai un'occhiata all'"Advanced Routing HOWTO" (http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html).

- Questo howto è una vaccata! Dove posso beccare i tizi che l'hanno scritto per rinfacciarglielo?

  - Entra nel canale irc #debian sul server irc.opensource.net e cerca JohnFlux. - Scrivimi (sono JohnFlux) a tapselj0@cs.man.ac.uk

- Questo howto fa schifo! Dove posso trovare documentazione migliore?

  - Da' un'occhiata a http://ipmasq.cjb.net.

  - Consulta il Masq-HOWTO di LDP.

- Su cos'altro state lavorando?

  Attualmente sto scrivendo una guida "Linux e i sistemi missile-anti-missile per principianti". Non ci sono documenti decenti per newbie sulla protezione del proprio sistema da attacchi nucleari. La gente sembra pensare che si tratti di missilistica o roba simile...