Linux Firewall: metti una marcia in più nella tua rete

Emiliano Bruni, `info@ebruni.it`

Ultima modifica: 24/02/2002 20.33

Introduzione

In questo ultimo periodo l'interesse riguardante la sicurezza delle reti ha valicato i confini degli specialisti del settore per invadere la vita di chi, volente o nolente, ha a che fare con INTERNET. L' improvvisa attenzione riguardo a questo importante argomento è dovuta essenzialmente all'uscita, sul mercato italiano, di connettività INTERNET a larga banda e a costi molto contenuti.

Questo grazie a una nuova tecnologia chiamata xDSL che permette di portare connettività a velocità anche di 2 Mbit/sec. utilizzando, come mezzo trasmissivo, un normalissimo cavo telefonico. I contratti offerti dai fornitori di connettività nazionale consentono inoltre di navigare in modalità leased-line, ossia con collegamento presente ventiquattro ore su ventiquattro, senza pagare scatti telefonici aggiuntivi.

Questa nuova offerta ha subito prosperato nel panorama delle altre soluzioni di connettività e permette, per esempio, anche a chi ha un budget limitato, di avere in casa propria un server web aziendale. Ha però anche portato alla luce la mancanza di protezione delle reti aziendali che risultano ora soggette a possibili accessi da parte di utenti non autorizzati provenienti da INTERNET.

Nei contratti associati all'offerta di connettività xDSL vi è però spesso una limitazione sul numero di indirizzi IP assegnati dal fornitore al cliente. Molti si trovano cosi nella posizione di non poter collegare in rete tutte le macchine che possiedono.

Esiste poi la categoria di chi possedeva già una connettività di tipo leased-line prima dell'avvento della tecnologia xDSL e che si trova ora nella situazione di poter scegliere un'alternativa più economica e spesso più veloce.

Il passaggio alla soluzione xDSL è però, di solito, difficilmente attuabile per delle problematiche inerenti alla conversione quali, per esempio: renumbering delle macchine interne, disservizi durante il periodo di passaggio, insufficiente numero di indirizzi IP assegnati dal fornitore xDSL, etc….

Inoltre, per chi fa un uso intenso di INTERNET, è spesso difficile rimanere sotto al tetto di consumo mensile che è presente come vincolo in molti contratti xDSL proposti dai provider[1]

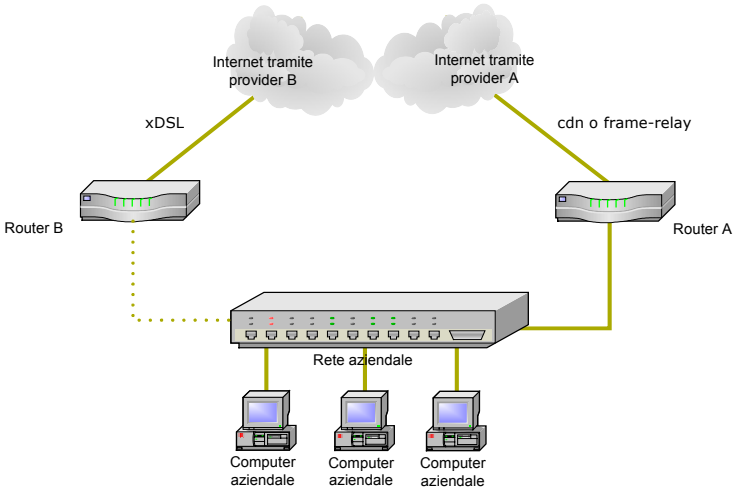
Una soluzione attuabile per questa categoria di consumatori può essere quella di affiancare, alla già presente connettività, il collegamento xDSL e cercare di incanalare il traffico su ambedue le linee bilanciando il carico di modo da non superare il tetto mensile di traffico.

Vediamo di esaminare le esigenze e le problematiche che si vengono a creare in un soluzione di tale tipo.

Consideriamo una tipica azienda che possiede già un punto di uscita verso la rete INTERNET di tipo CDN o frame relay con velocità, di solito,

[1] Sinonimo di fornitore di connettività INTERNET.

minore di 512 Kbit/sec attestata su di un'apparecchiatura chiamata router che instrada il traffico dalla rete interna verso INTERNET e viceversa. Questa azienda acquista, eventualmente da un altro fornitore di connettività, una linea xDSL e desidera distribuire il traffico su ambedue le linee.



In particolare, l'amministratore della rete, desidera che:

- la navigazione web viaggi attraverso la più veloce linea xDSL;
- alcuni computer interni, utilizzati da utenti preferenziali, passino sempre tramite la linea xDSL;
- indipendentemente dal computer interno e dal servizio utilizzato (web, ftp, icq, etc…), alcuni siti vengano visti sempre tramite la linea xDSL;
- il resto del traffico passi attraverso la vecchia linea per evitare di superare il tetto di traffico mensile imposti dal provider sulla linea xDSL.

Chiamiamo, per semplicità:

A → provider già presente	B → nuovo provider
AR → router verso A	BR → router verso B
AI → IP dati da A all'azienda	BI → IP dati da B all'azienda

Proviamo, come primo tentativo, ad aggiungere BR direttamente sulla stessa rete su chi è collegato AR e l'intera rete aziendale. Cosa succede in questo caso. Assolutamente nulla.

I computer aziendali hanno tutti indirizzi AI e hanno come gateway di default[2] AR e quindi l'attivazione e l'ingresso del router BR sulla rete non ha alcun effetto.

[2] Ogni computer ricava gli indirizzi dei computer eventualmente presenti sulla rete locale in base al proprio indirizzo e in base alla cosi detta "maschera di rete" (netmask) che individua l'intervallo di indirizzi IP che i computer della rete locale possono assumere. Il gateway di default è l'indirizzo di quell'apparato che conosce come instradare i pacchetti che hanno, come destinazione, indirizzi non appartenenti alla rete locale. In base al ragionamento di cui sopra se l'indirizzo di destinazione non è un indirizzo della rete locale, il computer invia i pacchetti al gateway di default. Il gateway di default è di solito un router.

Cambiando il gateway di default da AR a BR si ottiene solo di non riuscire più a navigare in INTERNET e questo perché gli indirizzi delle macchine interne sono di tipo AI e, ovviamente, il provider B non permette l'uscita di indirizzi diversi da BI.

Cambiando anche gli indirizzi IP si cade nel problema del renumbering che si voleva evitare e, in ogni caso, si otterrebbe di navigare tutti tramite B invece che tramite A alla faccia del bilanciamento del carico.

Proviamo allora, piuttosto che ad andare in giro per l'azienda a modificare la configurazione di tutti i computer, a lavorare sul router AR, su cui arrivano tutti i pacchetti diretti verso l'esterno, e vediamo se si può effettuare qualche operazione per redirigere parte del traffico sul router BR.

I router instradano il traffico in base al solo indirizzo di destinazione del pacchetto. L'unica cosa che si potrebbe fare è decidere che, per alcuni indirizzi remoti, il traffico venga diretto sul router BR.

Fermo restando che si devono aggiungere regole per ogni blocco di indirizzi IP che si desidera venga instradato su B rimane comunque il problema che tali pacchetti, una volta arrivati su B hanno, come indirizzo di sorgente, un indirizzo AI e quindi non vengono comunque instradati.

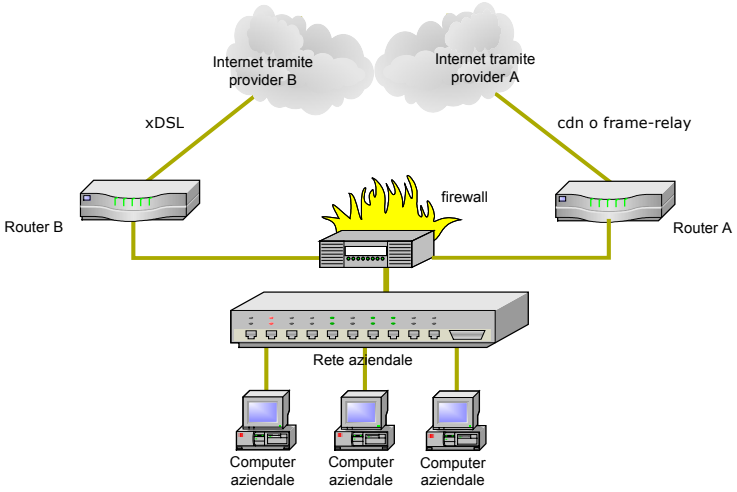
Riassumendo, per assolvere alle richieste imposte dall'amministratore di rete, deve avvenire che:

1. alcuni pacchetti siano inviati su AR mentre altri siano instradati su BR e la logica della scelta della strada da seguire deve dipendere dalle informazioni presenti su tutto l'header (intestazione) del pacchetto IP e non solo dall'indirizzo di destinazione come farebbe un router;
2. ai pacchetti inviati su BR va cambiato l'indirizzo IP del sorgente da AI a BI (network address traslation NAT) affinché il provider B instradi effettivamente tali pacchetti;
3. la soluzione deve essere trasparente rispetto alla rete già in essere per evitare di dover modificare netmask, gateway di default e indirizzi ip delle macchine della rete[3] (trasparent bridging).

L'unica soluzione è inserire un'apparecchiatura attiva tra la rete interna e i due router che esaudisca tali vincoli. Il dispositivo in questione è un firewall.

Benché quasi tutti i firewall soddisfano la condizione 2 e costosissimi firewall soddisfano anche la condizione 3, non conosco alcun firewall che soddisfi anche la condizione 1 se non il firewall "`iptables`" presente sulle macchine Linux.

[3] Tali modifiche andrebbero usualmente fatte in quanto tutti i firewall separano la rete in almeno due tronconi, uno esterno ed uno interno e questo significa che la rete locale va divisa e va quindi modificata la netmask dei computer interni per ridurre il range degli indirizzi ip della nuova rete locale che ora è diventata una sottorete di quella iniziale a causa dell'introduzione del firewall. Il firewall stesso diventa il gateway di default per le macchine più interne in quanto è la porta per andare verso l'esterno.



E' proprio su tale firewall che ci baseremo per la nostra trattazione futura.

Cos'è un firewall.

Un firewall è un componente attivo che seziona e collega due o più tronconi di rete. Usualmente la rete viene divisa in due sottoreti: una, detta esterna, comprende l'intera INTERNET mentre l'altra, detta interna, comprende una sezione più o meno grande di un insieme di computer locali.

Grazie alla sua posizione strategica, il firewall risulta il posto migliore ove imporre delle logiche di traffico per i pacchetti in transito e/o eseguire un monitoraggio di tali pacchetti.

La funzione principale del firewall è quella di proteggere i sistemi informatici presenti nella sezione interna dal "caos" presente nel lato esterno. Il firewall agisce sui pacchetti in transito da e per la zona interna potendo eseguire su di essi operazioni di:

- controllo,
- modifica,
- monitoraggio.

Questo grazie alla sua capacità di "aprire" il pacchetto IP per leggere le informazioni presenti sul suo header.

Struttura di un pacchetto TCP/IP

Header IP (20 bytes)	1		2		3	4	5	6	7	8	9	10
	Ver	TOS	DSF		Lungh. totale pacchetto		Identification		Flags	Offset	TTL	Proto (TCP)
	11		12		13	14	15	16	17	18	19	20
	Checksum				**Indirizzo IP sorgente (xxx.xxx.xxx.xxx)**				**Indirizzo IP destinazione (yyy.yyy.yyy.yyy)**			
Header TCP (20 bytes)	1		2		3	4	5	6	7	8	9	10
	Porta src (aaaa)				**Porta dest (bbbb)**		Numero di sequenza del pacchetto				Ack Number	
	11		12		13	14	15	16	17	18	19	20
	Ack Number				Offset	Flags	Windows size		Checksum		Urgent pointer	
Dati	Dati trasportati dal pacchetto											

I firewall di questo tipo sono detti "packet-type". Esiste un'altra tipologia, quella dei firewall di tipo "application-type", che si differenzia dalla prima in quanto agisce sull'informazione contenuta nei dati che il pacchetto trasporta e non sull'header.

I firewall "packet-type" non possono, per esempio, individuare un virus perché non agiscono sul contenuto dei dati trasportati dal pacchetto IP.

Non prenderemo però in considerazione gli "application-type" in quanto sono:

- poco performanti soprattutto su reti ad alto traffico;
- usualmente sono implementati tramite software proprietario chiuso;
- sono implementati in hardware;
- necessitano di un software lato client;
- sono costosi.

Materialmente il firewall è un componente hardware che possiede due o più schede di rete su cui viene fatto girare un ambiente operativo che analizza e gestisce il traffico dei pacchetti in base ad una configurazione data dall'amministratore della rete.

L'ambiente operativo può essere, per cosi dire, chiuso e le sue funzionalità sono decise dal costruttore e si può agire solo modificando la configurazione interna ossia modificando le regole di selezione dei pacchetti o può essere aperto e permettere, oltre che di modificare la configurazione, anche di modificare e ampliare le sue capacità operative.

Questa filosofia, adottata dal sistema operativo Linux, permette al suo firewall di operare al pari di un qualsiasi altro dispositivo commerciale dal costo di decine di migliaia di euro includendo inoltre funzionalità che non si trovano su alcun altro prodotto e con la possibilità di eseguire, in relazione per esempio al passaggio di pacchetti di dati prestabiliti, script personalizzati che possono inviare mail, accendere sirene acustiche etc….

La difesa contro il caos di INTERNET

Anche il firewall Linux, come i più blasonati firewall commerciali, possiede ovviamente la capacità di filtrare i pacchetti in transito e limitare quindi l'accesso dall'esterno ai soli servizi pubblici interni eliminando la possibilità di accesso alle risorse private presenti sulla rete aziendale.

Tale capacità viene messa in pratica tramite l'applicazione di regole applicate all'intero blocco di informazioni presenti nell'header del paccetto. Il firewall può decidere se accettare o rigettare il pacchetto in base, per esempio, all'indirizzo e/o alla porta del sorgente o del destinatario, in base al tipo di pacchetto (TCP,UDP,ICMP…) e così via.

Queste regole possono essere applicate in diversi momenti del processo di trasferimento del pacchetto dalla rete esterna alla rete interna.

`Iptables` prevede l'analisi e l'applicazione di regole sui pacchetti in processi che vengono chiamati di `PREROUTING`, `INPUT`, `FORWARD`, `OUTPUT` e `POSTROUTING`.

Per capire quando e come agiscano questi processi seguiamo il percorso di un pacchetto IP dalla scheda di rete esterna a quella interna senza la presenza del software di firewall `iptables`. Dopo essere entrato dalla scheda di rete esterna, il pacchetto viene aperto e ne viene analizzato l'header alla ricerca dell'indirizzo di destinazione. Questo indirizzo viene confrontato con la tabella di routing della macchina e quindi instradato verso una porta locale o verso la scheda di rete appropriata se l'indirizzo di destinazione è differente da quello associato al firewall.

Prima di proseguire, visto che abbiamo tirato in ballo la tabella di routing, vediamo che funzione svolge nel processo di trasporto del pacchetto.

La tabella di routing, viene utilizzata per decidere dove instradare il pacchetto IP in base all'indirizzo di destinazione presente nell'header.

Essa contiene un'associazione tra blocchi di indirizzi INTERNET e risorse con cui tali indirizzi possono essere raggiunti. Le risorse possono essere interfacce di rete locali o indirizzi IP di computer detti "gateway".

Di seguito vi è un esempio di una banale tabella di routing di una machina con una sola interfaccia ethernet. Linux nomina le interfacce di rete ethernet con il suffisso **eth**, numerandole poi in base alla posizione all'interno degli slot ISA/PCI.

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
192.168.0.0	*	255.255.255.0	U	0	0	0	eth0
0.0.0.0	192.168.0.1	0.0.0.0	UG	0	0	0	eth0

In evidenza:

- l'indirizzo `127.0.0.1`, il cosi detto indirizzo di loopback, mappato sull'interfaccia (virtuale) `lo`;
- gli indirizzi della rete locale da `192.168.0.0` a `192.168.0.255` mappati sulla scheda di rete **eth0**;
- la rotta di default, l'indirizzo `0.0.0.0/0.0.0.0`, l'ultima risorsa nel caso che le altre rotte non vengano applicate al pacchetto e che, di solito, corrisponde all'indirizzo del router/gateway verso INTERNET.

Ritorniamo all'analisi del tragitto percorso dal nostro pacchetto IP e vediamo cosa accade in presenza del firewall **iptables**.

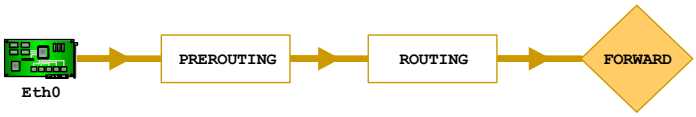
Il pacchetto entra dall'interfaccia esterna e viene sottoposto, prima del processo di routing, all'applicazione delle direttive presenti nella lista **PREROUTING**.

Usualmente, in tale processo vengono inserite regole che tendono a evidenziare il pacchetto per distinguerlo dagli altri pacchetti ed eseguire su di esso adeguate operazioni nelle successive fasi del processo di trasporto. Vedremo un esempio di ciò quando cercheremo di instradare un pacchetto in base all'indirizzo del sorgente invece che all'usuale indirizzo di destinazione.

In tale fase vengono anche applicate le regole per la gestione del destination NAT (DNAT) che vedremo nei prossimi capitoli.

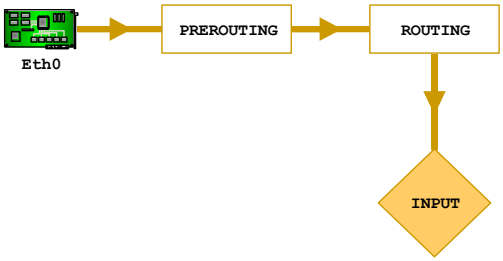
Il pacchetto subisce il processo usuale di routing in base alla tabella presente nella macchina locale.

Se il pacchetto, in base alla tabella di routing, è destinato alla interfaccia di rete interna vengono applicate le regole descritte nella lista di **FORWARD**.

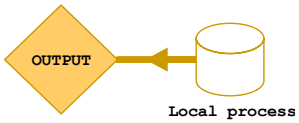


Usualmente sono questi i filtri più importanti in quanto definiscono cosa può passare dall'esterno verso l'interno e cosa no.

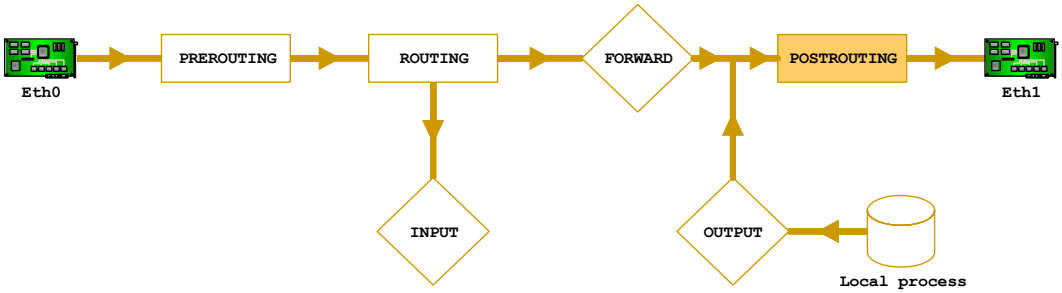
Se il pacchetto è destinato, in base alla tabella di routing, alla macchina locale vengono applicate le regole descritte nella lista di **INPUT**. Questi filtri proteggono il firewall stesso da accessi indesiderati.



Se il pacchetto ha come sorgente la macchina locale, ossia è stato generato da un processo della macchina locale vengono applicate, al pacchetto, le regole di **OUTPUT**.



Sia nel caso di forward che di output, prima di uscire dalla scheda di rete interna, il pacchetto subisce l'applicazione delle direttive di **POSTROUTING**. In tale fase vengono di solito applicate le regole per il source NAT (SNAT) che vedremo nei prossimi capitoli.



In ognuno di questi step ogni direttiva si chiede sostanzialmente: "se l'header del pacchetto verifica certe condizioni, che cosa devo fare del pacchetto" ? La risposta a questa domanda può essere o di accettare il pacchetto che continua nel suo percorso all'interno delle altre direttive e degli altri step o rigettare il pacchetto che viene definitivamente buttato via.

In ogni step, se il pacchetto non verifica nessuna delle condizioni impostate, può essere definita una regola di default da applicare al pacchetto che verrà quindi accettato o rigettato.

Usualmente si ritengono sicuri ed accettabili i pacchetti provenienti dall'interno e destinati verso l'esterno e quindi, in particolare, i pacchetti che hanno come sorgente il firewall saranno permessi e quindi l'impostazione ovvia di default del processo di **OUTPUT** sarà quella di **ACCEPT**.

Quello che si vuole invece evitare, a meno di eccezioni, e che dall'esterno si possa impunemente accedere verso l'interno. Ecco perché è usuale

impostare a **DROP** la configurazione di default dei processi di **FORWARD** e di **INPUT.**

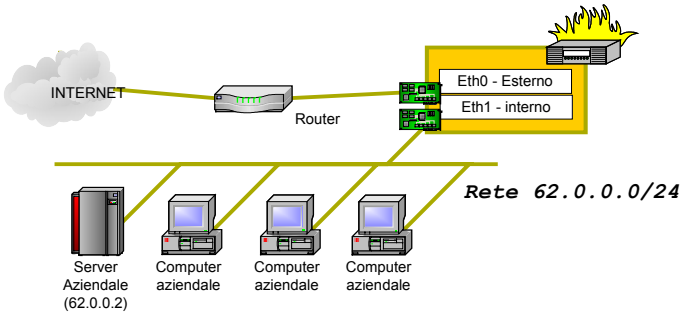
Sarebbe un errore lasciare ad **ACCEPT** queste regole e tentare di chiudere tutti i servizi interni. Il firewall serve, in particolare, per chiudere tutto, anche quello che neanche si sa di avere aperto. Negando tutto e accettando solo ciò che si considera come corretto si è sicuri che se dall'esterno di accede ad una data risorsa interna è perché siamo stati noi a richiederlo.

Il primo banale script di configurazione del nostro firewall sarà quindi (il flag `-P` imposta per l'appunto la politica di default per il processo):

```
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT ACCEPT
```

Strategie di difesa.

Impariamo a difenderci considerando, come semplice esempio, una piccola rete connessa ad INTERNET composta da un piccolo gruppo di computer client con installato il sistema operativo Windows® 2000 Professional e da un server aziendale con indirizzo `62.0.0.2` su cui vi è installato il sistema operativo Windows® 2000 Server con il server web Microsoft Internet Information Server.



Visto l'esiguo numero di macchine presenti sulla rete è stato possibile assegnare ad ogni macchina un indirizzo IP assegnato dal provider.

Il nostro scopo, come amministratori della rete, è quello di proteggerla da eventuali accessi non autorizzati provenienti dall'esterno permettendo l'accesso al solo server web.

A tale scopo separiamo la nostra rete da INTERNET inserendo un firewall Linux subito a valle del router e tentiamo di creare una configurazione tale che ci permetta di raggiungere lo scopo che ci siamo sopra preposti.

La cosa più evidente da fare è quella di bloccare l'accesso al nostro server permettendo l'accesso dall'esterno alla sua porta web.

Potremmo essere tentati di attivare una configurazione del tipo

```
-A FORWARD -p tcp -d 60.0.0.2 --dport ! web -j DROP [4]
```

[4] La regola esprime che nel processo di forward del pacchetto (**`-A FORWARD`**) i pacchetti di tipo tcp (**`-p tcp`**) [byte 10 dell'header IP] destinati al nostro server (**`-d 60.0.0.2`**) [byte 17-20 dell'header IP] e che tentano di collegarsi sul nostro server ad una porta diversa dalla porta 80 (**`--dport ! web`**) [byte 3-4 dell'header TCP] vengano rigettati (**`-j DROP`**).

ed effettivamente il nostro server sarebbe sicuro in quanto tutti i pacchetti ad esso destinati non diretti alla porta web verrebbero rigettati dalla regola sopra.

Quello che l'amministratore della rete ha dimenticato è che vi sono servizi aperti anche su tutti gli altri computer e che cosi facendo abbiamo lasciato ad un hacker di passaggio la possibilità di utilizzare tali porte per un eventuale attacco alla nostra rete.

Questo è dovuto al fatto di non aver specificato le politiche di default per i processi di **INPUT**, **OUTPUT** e **FORWARD** che, se non specificati esplicitamente, sono impostati ad **ACCEPT**. I pacchetti destinati a tutti gli altri pc della nostra rete vengono quindi implicitamente accettati e dunque tutti i servizi attivati sui vari pc interni sono accessibili dall'esterno ed eventualmente attaccabili.

Non si creda infatti che i computer Windows® 2000 Professional siano sicuri grazie al fatti di non essere dei server.

A parte le porte del protocollo NETBIOS che il sistema operativo attiva per la gestione della rete Microsoft vi possono essere altre porte aperte all'insaputa dell'amministratore di rete e, a volte, anche dello stesso utente. Per esempio, la persona che fa sviluppo ASP per il server aziendale potrebbe aver installato, sul proprio computer, un server IIS per velocizzare le fasi di sviluppo. Visto che il server IIS, a meno di non richiederlo esplicitamente, installa anche un server FTP e un server SMTP ecco che, all'insaputa dell'amministratore, c'è in rete un altro server web, un server ftp e un server di posta aperto al relay che può essere usato, per esempio, per azioni di spamming.

Si comprende quindi che il chiudere il solo server aziendale non evita alla rete di essere sottoposta ad attacchi esterni.

La filosofia corretta è, come detto in precedenza, di impostare a **DROP** i processi di **FORWARD** e di **INPUT** e di impostare la sola regola di accesso al server web come

```
-A FORWARD –p tcp –d 60.0.0.2 --dport web –j ACCEPT
```

In questo modo tutte le nostre macchine sono protette.

Il problema è che ora sono troppo protette in quanto gli utenti della nostra rete non possono più navigare e anche il server web è impossibilitato a rispondere alle, ora legittime, richieste dall'esterno.

I pacchetti provenienti dalla rete interna e destinati all'esterno vengono infatti rigettati dalla politica di default del processo di **FORWARD**.

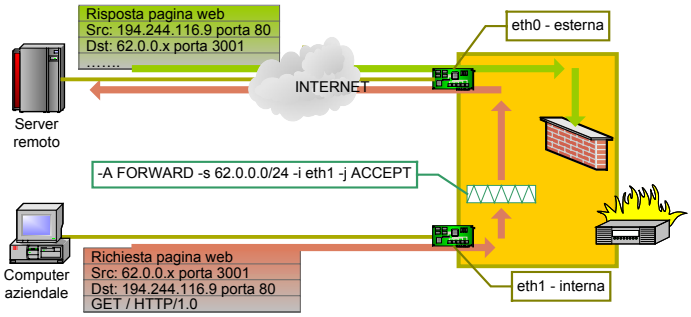
Per risolvere questo, considerando sicure tutte le connessioni che hanno come sorgente la nostra rete, dobbiamo intervenire sulla configurazione del firewall ed accettare tali pacchetti. Una corretta impostazione potrebbe essere

```
-A FORWARD –s 62.0.0.0/24 –i eth1 –j ACCEPT
```

dove l'indicazione dell'interfaccia di provenienza del pacchetto è stata esplicitata per evitare tecniche di sproofing[5] dall'esterno.

[5] Lo sproofing è un tipo di attacco per cui una macchina, non appartenente alla nostra LAN, fa credere ad una nostra macchina di essere una macchina locale ottenendo eventualmente dei privilegi che in realtà non le sono concessi. Un attacco di tale tipo può,

Ora le cose cominciano ad andare un po' meglio. Siamo protetti ed il server web risponde alle richieste dall'esterno. I nostri utenti interni però non riescono ancora a navigare. I loro pacchetti di richiesta di servizi esterni filtrano correttamente verso l'esterno grazie alla regola appena inserita ma le risposte a tali richieste non riescono a rientrare in quanto vanno a cadere nella regola di `DROP` del processo di `FORWARD`.



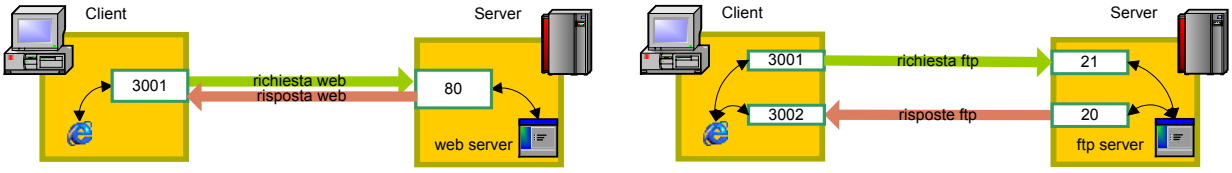
Quello che dobbiamo garantire è che pacchetti provenienti da macchine localizzate in "territorio nemico" possano giungere a noi solo in risposta ad una nostra richiesta e non in generale.

Il problema risiede però nel fatto che l'analisi del solo pacchetto in ingresso non permette di evincere se esso sia relativo o meno ad una nostra richiesta.

A causa di ciò, quasi tutti i firewall commerciali e tutti i firewall Linux ad eccezione di `iptables` non riescono a risolvere questo dilemma e si devono accontentare di "intuire" questa cosa.

Il metodo usuale con cui si tenta di individuare i pacchetti di risposta è basato sul modo con gli applicativi comunicano tra di loro.

Vi sono due modi con cui un client che si collega ad un server remoto attiva una comunicazione bidirezionale. Nel primo caso il client, che parla da una porta locale si collega sulla porta standard remota del servizi e su questo canale, aperto dal client viaggiano i pacchetti in ambedue le direzioni. Nel secondo caso il client comunica al server di aspettare una risposta dal server su una determinata porta locale. Il server si collega a questa nuova porta locale e i dati viaggiano in un senso sulla prima connessione e nell'altro sull'altra connessione.



In ambedue i casi comunque le porte locali sono comprese tra la porta 1024 e la 65535. Tale intervallo di porte prende il nome di porte non privilegiate dal fatto che esse possono essere aperte da software che si mettono in ascolto su di esse anche se il processo su cui girano non è controllato dall'utente amministrativo (root).

---

per l'appunto, essere bloccato specificando le interfacce da cui ci aspettiamo arrivare e/o uscire certi pacchetti.

Al contrario le porte dalla 1 alla 1023 sono dette porte privilegiate e solo processi che girano sotto l'utente di root possono mettersi in ascolto su tali porte.

Ciò che usualmente fanno i firewall per permettere ai server remoti di rispondere alle richieste dei client locali è di permettere il passaggio di tutti i pacchetti destinati a macchine interne sulle porte non privilegiate.

Tale soluzione risolve il problema ma non è esente da problemi. Un programma "maligno" potrebbe aprire una di queste porte per permettere l'accesso ad un hacker (la cosi detta backdoor) che, grazie a questa nuova regola di firewall, potrebbe dall'esterno entrare nella macchina. È si vero che il processo backdoor non gira sotto l'utente di root ma intanto un hacker è entrato e uno bravo potrebbe poi risalire i livelli del sistema e giungere al livello di root diventando quindi il padrone della macchima.

Inoltre molti dei sistemi operativi della famiglia Windows® non hanno il concetto di utenti e quindi delle backdoor installare su porte non privileggiate permetterebbero immediatamente all'hacker di prendere il completo controllo del sistema.

Ricordiamo inoltre che alcuni server si mettono in ascolto su porte non privilegiate. Un esempio tipico è il server proxy SQUID che si installa di solito sulla porta 3128 o 8080. Tale server sarebbe raggiungibile dall'esterno anche se il servizio proxy dovesse essere un servizio privato ad esclusivo uso interno.

Se è quindi vero che l'apertura delle porte non privilegiate è una soluzione al problema è anche vero che essa risulta una apertura indiscriminata e quindi potrebbe permettere accessi non desiderati dall'esterno.

Come risolve invece egregiamente il problema delle risposte a richieste locali il firewall **iptables**?

Dato che abbiamo già detto che è impossibile risalire alla correlazione tra richiesta e risposta analizzando il solo pacchetto in ingresso, **iptables** risolve la cosa tenendo traccia di tutte le richieste interne e verificando che i pacchetti di ritorno siano effettivamente associati ad una di queste richieste. Se il pacchetto verifica tale condizione viene inviato al client altrimenti il pacchetto viene rigettato.

Questa funzione è possibile grazie al modulo **ip_conntrack**. Tramite questo modulo viene aggiunta un'ulteriore analisi al pacchetto in transito che prescinde dall'analisi dell'header TCP/IP e che confronta il pacchetto con tutti gli altri pacchetti transitati in precedenza nel firewall. Quest'analisi ritorna un cosi detto "stato" del pacchetto relativo agli altri pacchetti. Lo stato di un pacchetto inteso all'interno dell'analisi del modulo **ip_conntrack** può assumere i seguenti valori:

- **NEW** → il pacchetto non è correlato a nessun altro pacchetto transitato in precedenza ed è quindi teso alla creazione di una nuova connessione
- **ESTABLISHED** → il pacchetto appartiene ad una connessione già esistente ossia è un pacchetto di risposta relativo ad richiesta di dati sulla connessione esistente (caso richiesta pagina web)

- **RELATED** → un pacchetto che è correlato ma non appartiene ad una connessione esistente (caso risposta ftp)
- **INVALID** → non è stato possibile ricavare lo stato del pacchetto.

Avendo aggiunto questa nuova informazione all'analisi del pacchetto è ora possibile scrivere una regola che permetta l'ingresso nella nostra rete di pacchetti provenienti dalla "zona nemica" solo se correlati a richieste di host interni. La regola, da aggiungere alla configurazione del firewall risulta essere:

```
-A FORWARD –d 62.0.0.0/24 –m state --state ESTABLISHED,RELATED –j ACCEPT
```

In questo modo siamo riusciti a proteggere egregiamente la nostra rete.

Rileggiamo e commentiamo a parole cosa fa la nostra configurazione attuale:

```
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT ACCEPT
-A FORWARD –s 62.0.0.0/24 –i eth1 –j ACCEPT
-A FORWARD –d 62.0.0.0/24 –m state --state ESTABLISHED,RELATED –j ACCEPT
-A FORWARD –p tcp –d 60.0.0.2 --dport web –j ACCEPT
```

ossia tutti i pacchetti in transito sul nostro firewall vengono rigettati ad eccezione dei pacchetti in partenza dalla nostra rete interna e dei pacchetti provenienti dall'esterno che o sono correlati a richieste provenienti dall'interno o sono pacchetti destinati al nostro server web.

Abbiamo quindi raggiunto lo scopo prepostoci.

Source Nat e Masquerade: navigare senza indirizzi INTERNET.

Uno degli aspetti sottovalutati nei giorni in cui INTERNET si chiamava ancora ARPANET è stato quello relativo alle dimensioni dello spazio degli indirizzi IP e della loro assegnazione.

Nella visione di chi ha creato la Rete si riteneva che, per come erano stati definiti gli indirizzi IP, il loro numero fosse talmente grande da essere praticamente inesauribile.

L'analisi era talmente ottimistica che vennero assegnati, alle prime istituzioni che ne fecero richiesta, blocchi di indirizzi ampiamente sovradimensionati rispetto alle loro esigenze del tempo e anche future.

Nessuno avrebbe potuto neanche lontanamente immaginare l'enorme espansione delle Rete e l'enorme numero di computer che avrebbero richiesto di farne parte.

Oggi gli indirizzi IP liberi sono una risorsa molto importante e l'iter necessario per la loro assegnazione è complesso e prevede la richiesta di una descrizione analitica e particolareggiata del loro utilizzo da parte del richiedente.

È per questo motivo che i provider assegnano ai clienti finali un numero minimo di indirizzi IP e fanno molto pesare, in termini di costi, l'assegnazione di indirizzi IP aggiuntivi.

Non è quindi rara la situazione di un cliente che si vede assegnare dal provider un numero di indirizzi IP minore del numero di macchine presenti

sulla sua rete e che quindi si trova nella necessità di limitare la navigazione a pochi computer.

Un problema analogo lo ha chi si collega ad INTERNET tramite un modem e un abbonamento ad un Internet Service Provider (ISP). Il provider, in questo caso, assegna, per la durata del collegamento, un indirizzo IP al computer che diviene a tutti gli effetti una macchina della INTERNET.

Ma se questo computer è collegato anche ad una rete locale a cui accedono altri computer l'utente potrebbe voler condividere la connessione ottenuta con gli altri in modo che tutti possano accedere ad INTERNET usando quell'unico accesso. Ovviamente solo il computer con il modem può navigare in quanto è l'unico ad avere un indirizzo IP reale.

La risposta a queste richieste è il cosi detto masquerade che è un sottoinsieme di un metodo più generale di modifica del pacchetto IP chiamato source NAT o SNAT.

Che cos'è l'operazione di SNAT e come può aiutare a risolvere il problema del ridotto numero di indirizzi IP assegnati dal provider?

Tecnicamente l'SNAT modifica, nell'header dei pacchetti, l'indirizzo IP del sorgente facendo credere al destinatario del pacchetto che esso provenga da un altro indirizzo IP.

Questo permette anche a chi non è fisicamente in INTERNET di navigare per la Rete. Spieghiamo con un esempio questo, a prima vista, controsenso.

Consideriamo due computer, uno con Windows® 2000 Professional e uno con Linux, collegati alla stessa rete locale.

Ai due computer sono assegnati due indirizzi IP intranet[6] del blocco `192.168.0.0/24` e, in particolare, la macchina Linux ha un indirizzo IP assegnato alla sua scheda di rete `eth0 192.168.0.1` mentre la macchina Windows® ha l'indirizzo `192.168.0.2`.

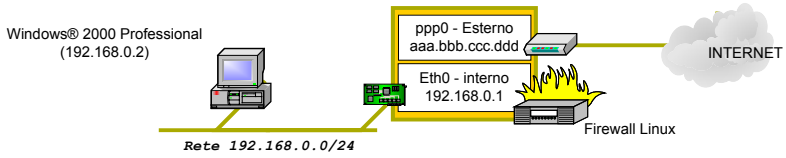
Il protocollo TCP/IP è indipendente dal sistema operativo e quindi, tramite esso, le due macchine "si vedono" sulla rete locale.

Sulla macchina Linux vi è anche un modem e vi è configurato un abbonamento tramite un ISP. Durante il periodo di collegamento con l'ISP, alla macchina viene assegnato un indirizzo INTERNET aaa.bbb.ccc.ddd. La macchina Linux ha quindi, per la durata del collegamento, due indirizzi IP assegnati, uno interno, con cui vede la macchina Windows®, e uno esterno, sulla connessione modem, tramite il quale risulta visibile e vede la INTERNET.

Ci chiediamo come si può ottenere di far navigare anche la macchina Windows® tramite il collegamento attivato dalla macchina Linux.

---

[6] Gli indirizzi IP devono essere compresi tra `0.0.0.0` e `255.255.255.255` (in esadecimale `0.0.0.0` e `ff.ff.ff.ff`). Alcuni indirizzi IP sono però stati riservati alla costituzione di reti private e non sono utilizzabili su INTERNET. Gli indirizzi da `10.0.0.0` a `10.255.255.255`, quelli tra `172.16.0.0` e `172.31.255.255.255` e quelli tra `192.168.0.0` e `192.168.255.255` sono chiamati indirizzi intranet e non può esistere alcuna macchina di INTERNET ad avere un indirizzo appartenente ad uno di questi intervalli.



Il primo step consiste nell'informare il modulo TCP/IP della macchina Windows® che esiste, sulla rete locale, una macchina che sa come inviare pacchetti ad INTERNET ossia consiste nel configurare il gateway di default della macchina Windows® impostandolo con l'indirizzo IP della macchina Linux.

Ma attenzione, la macchina Linux ha, ora che è connessa, due indirizzi IP. Quale dei due va impostato sulla macchina Windows®?

Ovviamente l'indirizzo locale `192.168.0.1` in quanto l'altro è già un indirizzo INTERNET e la macchina Windows® 2000 non sa come raggiungere la Rete.

Con questa configurazione una richiesta della macchina Windows® destinata a INTERNET giunge sulla macchina Linux che instrada il pacchetto sulla sua rotta di default che, da quando è attiva la connessione modem, risulta settata proprio su tale linea.

Sembra che già tutto funzioni ma in realtà non funziona niente. L'unica macchina che il provider ha autorizzato a navigare è la macchina Linux. Quando le apparecchiature dell'ISP vedono arrivare dei pacchetti dalla macchina Windows® con indirizzo di sorgente `192.168.0.1` rigettano tali pacchetti in quanto, dal loro punto di vista, sono pacchetti anomali perché lungo tale connessione telefonica gli unici pacchetti che dovrebbero arrivare devono avere indirizzo di sorgente **`aaa.bbb.ccc.ddd`**.

Cosa fa allora il SNAT se attivato sulla macchina Linux?

Quando arriva il pacchetto dalla macchina Windows® **`l'iptables`** si segna l'header che dovrebbe assumere il pacchetto di risposta ad esso associato e sostituisce l'indirizzo di sorgente `192.168.0.2` del pacchetto in transito con il suo indirizzo INTERNET aaa.bbb.ccc.ddd instradandolo lungo la linea modem.

Ora le apparecchiature del provider non hanno più nessun motivo per rigettare il pacchetto in quanto, dal loro punto di vista, esso proviene dal soggetto legittimamente autorizzato a navigare lungo quella connessione.

La cosa non finisce però qui in quanto il server remoto invia la risposta alla macchina Linux che si vede arrivare una risposta per una richiesta che non ha fatto. Qui interviene di nuovo il SNAT che si accorge, controllando l'header del pacchetto con quello salvato sopra, che questo flusso di dati remoti va dirottato alla macchina Windows®.

Sostituisce allora all'indirizzo del destinatario, attualmente settato sull'header del pacchetto di risposta, l'indirizzo IP della macchina Windows® e glielo invia. La macchina Windows® si vede arrivare il pacchetto di risposta che si aspettava ed è felice ☺.

La macchina Windows® naviga quindi per "interposta persona" in quanto è la macchina Linux che la fa navigare mascherando di volta in volta l'indirizzo sorgente della macchina locale.

È intuibile che questa operazione risolva i problemi anche degli utenti con collegamento xDSL e con pochi indirizzi IP. Ne vederemo un esempio nei prossimi capitoli.

Vediamo ora tecnicamente come si configurano le cose sulla macchina Linux con **iptables**.

La configurazione dell'SNAT è molto semplice e consiste nell'attivazione della tabella di NAT nel processo di **POSTROUTING** tramite l'impostazione:

```
-t nat –A POSTROUTING –o ppp0 –j MASQUERADE
```

deve **ppp0** è l'interfaccia modem di uscita verso INTERNET.

Come si vede non appare alcun indirizzo IP nella configurazione in quanto il sistema prevede automaticamente a ricavare l'indirizzo dinamico assegnato di volta in volta dal provider e di utilizzarlo per lo SNAT.

Se la situazione è quella di un indirizzo statico assegnato come nel caso di un collegamento xDSL con pochi indirizzi IP è meglio utilizzare questa configurazione

```
-t nat –A POSTROUTING –o eth0 –j SNAT --to xxx.xxx.xxx.xxx
```

dove **eth0** è l'interfaccia di rete lato INTERNET e **xxx.xxx.xxx.xxx** è l'indirizzo INTERNET di questa interfaccia.

Un'ultima domanda: perché questa regola va applicata nel processo di **POSTROUTING**?

La spiegazione è semplice; per far si che, almeno fino all'ultimo momento, prima di uscire dall'interfaccia verso INTERNET il pacchetto sia ancora identificato come proveniente dalla macchina interna e su cui poter applicare regole di filtro o di routing. Se questo mascheramento avvenisse prima, il pacchetto sembrerebbe del tutto indistinguibile da un pacchetto generato localmente dal box Linux e non potrebbero quindi essere applicarti eventuali filtri personalizzati.

Destination Nat: il server che c'è ma non c'è.

Analogamente al source NAT anche il destination network address traslation agisce modificando l'header dei pacchetti in transito ed in particolare l'indirizzo e la porta di destinazione.

Al contrario dell'SNAT, ma per lo stesso motivo, il DNAT viene applicato nel processo di **PREROUTING** cosi che tutti i restanti processi di **iptables** e di routing agiscono sul pacchetto già modificato.

L'operazione di DNAT può essere utilizzata per operazioni di "port forwarding" o di "trasparent proxy". Tralasciando questo secondo caso che tratteremo nel prossimo capitolo, occupiamoci ora dell'operazione di "port forwarding", di capire cos'è e di come viene eseguita utilizzando il destination NAT.

Il "port forwarding" è quell'operazione per cui una porta presente su di un server in realtà è un "puntatore" ad un servizio presente su di un'altra porta molto spesso attiva su di un altro server. Tutte le richieste di connessione sulla prima porta vengono inoltrate al servizio reale attivo sulla seconda porta.

Un utilizzo del "port forwarding" si ha nella situazione in cui uno o più servizi presenti in una rete con indirizzi intranet debbono essere esportati e visibili anche sulla rete INTERNET.

Portiamo quindi ad esempio una azienda che abbia acquistato una linea xDSL da un fornitore di connettività INTERNET che le ha assegnato quattro indirizzi IP nell'intervallo da `194.244.12.0` a `194.244.12.3`.

Dei quattro indirizzi, lo zero è l' indirizzo della rete, l'uno è l'indirizzo del router che collega l'azienda al provider e il tre è l'indirizzo di broadcast. L'unico indirizzo libero risulta quindi l'`194.244.12.2` e, visto che l'azienda in realtà possiede molti computer che desiderano navigare sulla Rete, ha utilizzato l'SNAT attivandolo su di una macchina Linux. L'indirizzo libero è stato assegnato all'interfaccia di rete esterna mentre a quella interna è stata data un indirizzo intranet della classe `192.168.0.0/24`.

L'azienda decide quindi di trasferire il proprio sito, attualmente in housing presso un provider, sulla rete aziendale per averne una migliore gestione e manutenzione.

Attivare il server web sul firewall potrebbe essere impossibile per incompatibilità di sistema operativo ma, anche nel caso in cui non esistesse questa incompatibilità, l'installazione di ulteriori servizi sul firewall è totalmente inaccettabile. Il firewall, come unico baluardo tra noi è "il nemico" deve essere totalmente sicuro e ogni servizio aggiuntivo tenderebbe a ridurre questo fattore di sicurezza.

Il server viene quindi agganciato alla rete interna e gli viene assegnato l'indirizzo intranet `192.168.0.1`.

Come si rende visibile il server alla INTERNET visto che ora ha un indirizzo intranet irraggiungibile dalla Rete?

Se è vero che l'installazione del servizio web sul firewall era inaccettabile è però vero che tentare di mappare la porta web del firewall con la porta web del server `192.168.0.1` non preclude assolutamente la sicurezza della rete.

L'idea è proprio questa: esternamente il server web verrà visto attivo sulla porta `80` (web) dell'indirizzo `194.222.12.2` ma tutti i pacchetti diretti su questa posta verranno poi dirottati dal firewall sulla porta 80 del server `192.168.0.1`.

Come segugi, seguiamo il pacchetto TCP/IP di richiesta di una pagina del nostro sito e il relativo pacchetto di risposta.

Il pacchetto di richiesta diretto verso la porta `80` del server `194.222.12.2` entra dalla scheda esterna `eth0` del firewall.

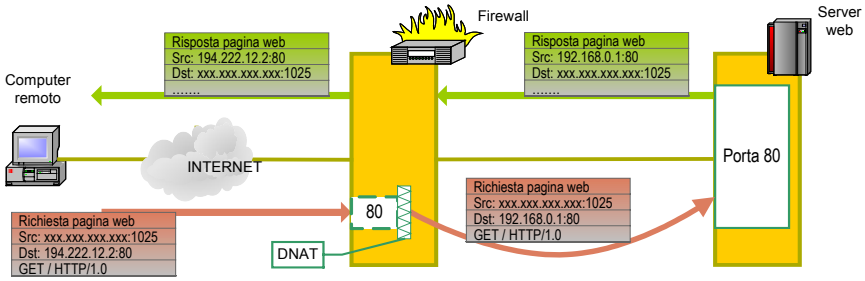
Nel processo di `PREROUTING` il firewall verifica che questo pacchetto corrisponde ad una determinata regola e ne modifica il destinatario (DNAT) sostituendo l'indirizzo di destinazione con il `192.168.0.2`.

La tabella di routing vede ora un pacchetto destinato a questo host interno e lo instrada sulla scheda `eth1` permettendo al pacchetto di giungere sul server web realmente presente sulla rete.

Il server web interno processa la richiesta e prepara un pacchetto di risposta.

Questo pacchetto torna quindi sul firewall che riconosce questo come pacchetto di risposta relativo al pacchetto che poco prima lui aveva modificato e modifica questo pacchetto sostuendo all'indirizzo di sorgente intranet il proprio indirizzo e lo inoltra all'host remoto.

Risultato, dal punto di vista dell'host remoto:



e quindi tutto quadra perfettamente.

La configurazione dell'iptables per ottenere questo risultato è molto semplice

```
-t nat -A PREROUTING -p tcp -d 194.222.12.2 --dport 80 -j DNAT ⬇
 --to 192.168.0.1
```

Come si vede, non è stato necessario specificare la porta di destinazione in quanto la mappatura delle porte è la stessa. Se internamente il server web era attivato su una porta non standard e quindi differente dalla porta 80 si sarebbe dovuto aggiungere questa informazione alla riga di configurazione del DNAT.

Ovviamente quello che funziona per un servizio funziona anche con molti. Si possono cosi avere più server interni e mapparli esternamente utilizzando il solo indirizzo IP del firewall. Se però si cerca di esportare due server dello stesso tipo, per esempio due server web attivati su due macchine differenti, solo uno dei due potrà essere visto esternamente presente su firewall sulla porta web standard. L'altro dovrà essere mappato su una porta xx non standard e sarà visibile esternamente solo tramite l'url

```
http://www.miosito.it:xx/
```

Trasparent proxy: come dirottare i pacchetti TCP/IP.

Una forma specializzata di DNAT è la così detta redirezione dei pacchetti che permette, in modo trasparente, di dirottare alcune tipologie di dati su di una singola macchina per motivi di monitoraggio o di trasparent proxy.

Analizziamo il perché si debba e si voglia eseguire tale operazione considerando il trasparent proxy di tipo web applicato ad un ISP che offre collegamenti in dial-up ad una utenza privata e/o commerciale. L'ISP ha, presso il suo data center, un proxy server che, se utilizzato dagli utenti, intercetta le richieste web facendosi carico della ricezione delle pagine e dell'invio all'utente con il vantaggio di salvarsi in locale le pagine ricevute. Se una successiva richiesta fa riferimento a una pagina già presente sull'hard disk locale viene inviata questa all'utente con ovvi risparmi in termini di banda uscente del provider e in termini di risposta all'utente visto che la pagina arriva da una locazione "più vicina".

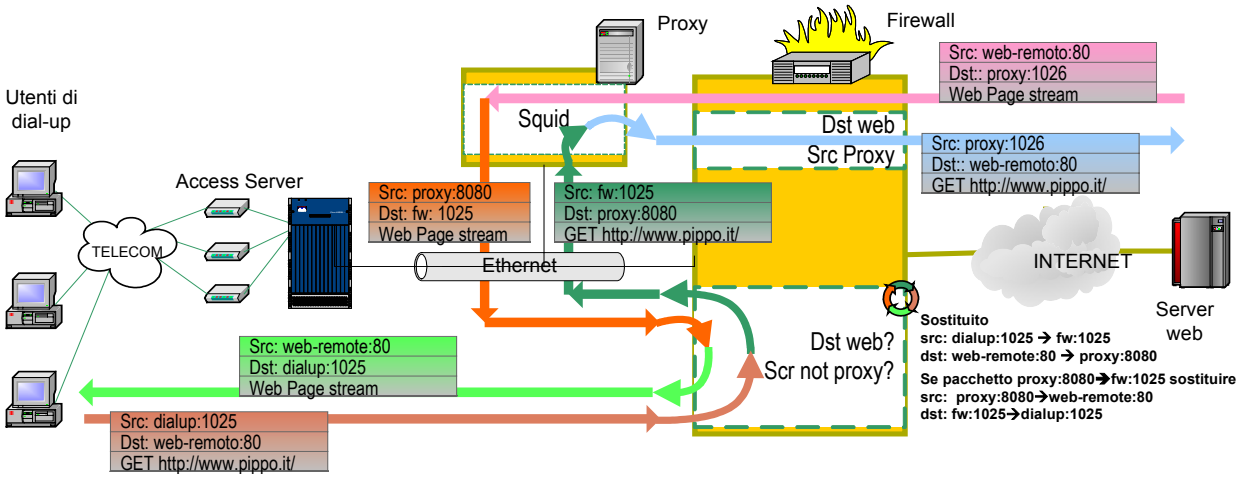
Il tallone di Achille di questo sistema sta nella frase "se utilizzato dall'utente". Infatti l'utilizzo o meno del proxy è a discrezione dell'utente

che, sul browser del proprio computer, può attivare o meno l'utilizzo del proxy dell'ISP.

Ovviamente il vantaggio massimo con il proxy si ha se tutti lo utilizzano in modo massimizzando cosi la probabilità che una pagina di richiesta venga trovata in locale e non debba essere recuperata dal sito remoto.

Il trasparent proxy è quell'operazione di inviare in ogni caso tutte le richieste web al server proxy indipendentemente dal fatto che l'utente abbia o meno attivato l'utilizzo del proxy sul suo computer.

Come funziona a livello di flusso dati la cosa?



Il pacchetto di richiesta di una pagina web remota da parte di un utente locale attraversa il firewall che, applicando una determinata regola di DNAT, sostituisce all'indirizzo di destinazione quello del proxy. Il pacchetto viene quindi instradato dalla tabella di routing sulla stessa interfaccia da cui era entrato in quanto il server proxy si trova, rispetto al firewall, dalla stesso lato dell'utente, il lato protetto.

Il proxy prende in consegna la richiesta web e recupera la pagina in locale o da remoto in base alla disponibilità della stessa sull'hard disk. Una volta ottenuta la pagina di richiesta invia la risposta all'utente prendendo questa informazione dall'header del pacchetto.

Aggiungiamo un' osservazione, prima di mostrare la riga di configurazione dell'iptables che esegue l'operazione di DNAT per il trasparent proxy: il software proxy va istruito del fatto che sta per essere usato come trasparent proxy e che il pacchetto di richiesta che gli arriva non è un pacchetto di richiesta proxy ma è direttamente un pacchetto http e quindi deve modificare il protocollo di interpretazione del pacchetto ricevente per capire qual è l'informazione remota da cercare. Il server proxy SQUID ha per esempio questa funzionalità di utilizzare anche il protocollo http impostandosi come server in trasparent proxy.

Concludiamo questa discussione mostrando come appare la riga di configurazione del trasparent proxy su iptables:

```
-t nat -A PREROUTING -I eth1 -p tcp -dport 80 -s ! xxx.xxx.xxx.xxx ↴
 -j REDIRECT -to xxx.xxx.xxx.xxx:8080
```

dove `xxx.xxx.xxx.xxx` è l'indirizzo del server su cui è attivo il server proxy collegato alla porta `8080`.

Trasparent bridging: trasforma il tuo Linux in un hub.

Benché poco inerente all'argomento firewall, il problema del trasparent bridging nasce ogni qual volta si debba implementare un firewall all'interno di una rete preesistente soprattutto se questa rete è globalmente connessa a INTERNET.

Infatti, usualmente, l'introduzione di un firewall seziona la rete in due sottosezioni. Quindi ciò comporta inevitabilmente la modifica della netmask negli host delle due sezioni e, per i soli host della sezione interna, il firewall deve diventare il nuovo gateway di default e quindi anche questo parametro va aggiornato. Questa situazione è aggravata dal fatto che, di solito, il firewall viene posizionato subito sotto il router aziendale cosi che le modifiche sopra accennate vanno eseguite su praticamente tutti gli host.

Inoltre anche la configurazione del router va modificata per adeguarsi alla nuova netmask e per informarlo che la sottorete mancante può essere raggiunta tramite il gateway firewall.

Può però capitare che questa modifica risulti impossibile da eseguire in quanto, in alcuni casi, il router non è di proprietà dell'azienda ma è di proprietà del carrier che lo ha ceduto in locazione all'azienda assieme alla connettività INTERNET.

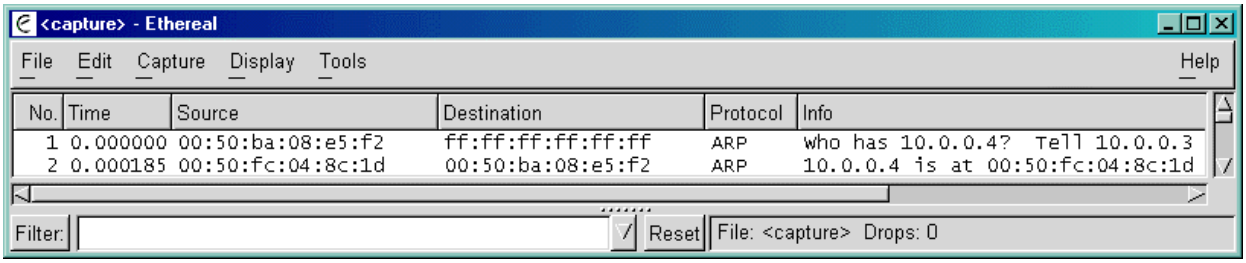
Per tutti questi motivi l'introduzione di un firewall può essere problematica se non irrisolvibile.

Vediamo come si risolve la cosa grazie alle potenzialità del sistema operativo Linux che, a quanto il sottoscritto conosca, è l'unico che permetta di affiancare un potente firewall con una soluzione di trasparent bridging.

L'idea è quella di rendere "trasparente" la nuova macchina introdotta evitando sezionamenti e modifiche alle configurazioni di netmask e di gateway tramite l'utilizzo del proxy ARP.

L'ARP è un protocollo con cui le schede di rete richiedono informazioni alle altre schede presenti sulla LAN.

Quando una scheda di rete vuole sapere se un qualche host ha un particolare indirizzo IP richiede all'eventuale scheda che sulla LAN ha quell'indirizzo IP di identificarsi. Questo messaggio viene ascoltato da tutte le schede di rete, si parla cosi di messaggio di broadcast. Se una di queste riconosce l'indirizzo IP richiesto come il proprio, risponde alla richiesta con il proprio indirizzo hardware che è un identificativo univoco di ogni scheda di rete prodotta nel mondo.



Quando si assegna ad una determinata scheda di rete un particolare indirizzo IP automaticamente si esegue un'operazione all'interno della cosi detta tabella ARP che associa all'indirizzo IP l'indirizzo hardware della

scheda di rete informando la scheda di farsi riconoscere sulla rete qualora qualcuno richieda quell'indirizzo IP.

Nell'esempio sopra la macchina con indirizzo IP `10.0.0.3` e con indirizzo hardware `00:50:ba:08:e5:f2` ha chiesto a tutte le schede (`ff:ff:ff:ff:ff:ff`) chi aveva l'indirizzo `10.0.0.4`. La scheda di rete `00:50:fc:04:8c:1d`, presente su un'altra macchina si è identificata come scheda che ha associata l'indirizzo richiesto. La tabella ARP per la macchina `10.0.0.3` risulta essere:

```
Address   HWtype   HWaddress           Flags Mask  Iface
10.0.0.3  ether    00:50:BA:08:E5:F2   C           eth0
```

mentre per la `10.0.0.4` si ha che

```
Interface: 10.0.0.4 on Interface 0x1000003
  Internet Address      Physical Address      Type
  10.0.0.4              00-50-fc-04-8c-1d     dynamic
```

dove la differente visualizzazione della tabella di ARP mostrata è dovuta soltanto al fatto che la prima è stata ricavata da una macchina Linux mentre la seconda è come viene stampata la tabella di ARP su di una macchina Windows® 2000 professional.

L'idea alla base del proxy ARP sta nell'aggiungere manualmente alla tabella di ARP della scheda di rete esterna tutto il range di indirizzi che ora è stato spostato internamente.

Seguiamo quindi, come nostro solito, il percorso di un pacchetto IP proveniente dall'esterno e diretto ad una macchina locale.

[DISEGNO FLUSSO DEL PACCHETTO]

Il pacchetto destinato all'indirizzo locale `194.12.12.12` entra nel router che sa, in base alla sua tabella di routing, che la classe `194.12.12.0/24` è direttamente connessa sulla sua interfaccia ethernet. Lungo tale interfaccia invia quindi una richiesta ARP chiedendo, in broadcast, chi è che ha l'indirizzo `194.12.12.12`.

Sulla rete direttamente connessa alla ethernet del router c'e solo la `eth0` del firewall che è stata istruita, grazie al proxy ARP, a rispondere positivamente ad ogni richiesta ARP per indirizzi della classe `194.12.12.0/24`.

Il router invia allora il pacchetto alla `eth0` del firewall credendo che sia la scheda di rete con l'indirizzo `194.12.12.12`. Il pacchetto entra nel firewall e, in base alla tabella di routing, il firewall scopre che la classe `194.12.12.0/24` è quasi totalmente presente sulla `eth1`. Invia quindi, dopo aver applicato le eventuali regole di firewall, il pacchetto sulla `eth1` al corretto destinatario che si è fatto riconoscere avendo risposto alla richiesta ARP eseguita dal firewall sulla `eth1`.

Quindi, tramite il proxy ARP, il pacchetto in ingresso è giunto sul firewall senza che sia stato necessario modificare la configurazione del router.

Una soluzione analoga si ha dal lato interno della rete istruendo la `eth1` a rispondere positivamente ai pacchetti ARP di richiesta per l'indirizzo IP del router.

Quindi, quando l'host `194.12.12.12` tenta di rispondere all'host remoto cercando il suo gateway di default, il router, la `eth1` gli fa credere di essere il

router e prende in consegna il pacchetto che poi si premunisce di inviare sulla **eth0** verso il router non prima di aver applicato le eventuali regole di firewall.

Dopo aver visto come funziona la cosa vediamo come si attiva sul firewall.

La prima cosa è la configurazione manuale degli indirizzi IP sulle schede di rete[7]:

```
ifconfig eth0 add 194.12.12.12 netmask 255.255.255.255 up
ifconfig eth1 add 194.12.12.12 netmask 255.255.255.255 up
ip address add 194.12.12.12/32 dev eth0
ip address add 194.12.12.12/32 dev eth1
```

Si da, per semplicità e risparmio, a tutte due le schede di rete un unico indirizzo IP e un netmask con solo quell'indirizzo IP in modo che nella tabella di routing non vengano aggiunte , per ora, nessuna rotta in quanto le rotte vengono aggiunte a mano dai comandi:

```
route add –net 194.12.12.0 netmask 255.255.255.0 eth1
route add –host 194.12.12.1 netmask 255.255.255.255 eth0
ip route add 194.12.12.0/24 dev eth1
ip route add 194.12.12.1 dev eth0
```

che informano il firewall dove sia il router e dove sia il resto della rete. Le due rotte, come si vede, si sovrappongono per l'indirizzo IP del router **194.12.12.1** ma verrà applicata, per il router, la seconda in quanto ha una netmask più specifica.

E ora, la configurazione del proxy ARP lato **eth0** [DA VERIFICARE]

```
arp –i eth0 -Ds 194.12.12.0 eth0 netmask 255.255.255.0 pub
ip neigh add proxy 194.12.12.0/24 nud permanent dev eth0
```

e lato **eth1**

```
arp –i eth1 –Ds 194.12.12.1 eth1 netmask 255.255.255.255 pub
ip neigh add proxy 194.12.12.1 nud permanent dev eth1
```

che aggiunge la classe **192.12.12.0** alla tabella ARP della **eth0** e l'indirizzo del router alla tabella di ARP della **eth1**.

Advancing routing: il routing alla n-esima potenza.

L'accoppiata personal computer e sistema operativo Linux permette di realizzare uno dei router più complessi attualmente disponibili.

Questo è un elenco, per nulla esaustivo, delle possibilità offerte da un router Linux:

- regolare e limitare la banda in base all'host sorgente o remoto,
- suddividere e condividere la banda disponibile su di una linea
- fondere insieme due o più linee in modo da avere virtualmente un'unica linea somma delle linee reali,
- esporre più servere come se fossero uno solo in modo da ottenere un bilanciamento del carico

[7] La "vechia" sintassi di configurazione dello stack TCP/IP del Linux viene indicata con il carattere normale mentre la "nuova" configurazione del potente pacchetto **iproute2** viene indicata in corsivo.

- eseguire operazioni di routing in base all'utente unix, all'indirizzo del sorgente o del remoto, del tipo di servizio, ora del giorno etc.

Quello su cui vogliamo focalizzare la nostra attenzione è la funzione di routing basata sull'indirizzo del sorgente. Per ulteriori informazioni si può dare un'occhiata al Linux Advancing Routing nei riferimenti alla fine di questo articolo.

Finora, parlando della tabella di routing, si è detto che, quando un pacchetto entra nella macchina, essa si basa su tale tabella per confrontare l'indirizzo di destinazione del pacchetto e trovare la rotta corretta per instradare il pacchetto al destinatario.

Alcune volte però questa logica non può essere sufficiente. Supponiamo che si posseggano due linee di connettività, una poco potente ma a costi forfetari e una molto veloce ma a consumo. Si può volere, per esempio, far si che le macchine di utenti "privilegiati" possano utilizzare il collegamento più veloce senza peraltro permettere a tutti il suo utilizzo per evitare costi di consumo eccessivi su tale linea.

La soluzione esiste se è possibile eseguire un algoritmo di routing basato sull'indirizzo sorgente del pacchetto in modo da inviare tutti i pacchetto sul primo collegamento ad esclusione di quelli provenienti dagli utenti privilegiati.

Vediamo come funziona il routing su una macchina Linux con la suite **iproute2** e kernel superiore alla release 2.4.0.

Su tali macchina non esiste un'unica tabella di routing ma si possono generare diverse tabelle di routing costringendo i pacchetti ad utilizzare una di queste tabelle in base a logiche diverse.

All'attivazione di una o più schede di rete su di una macchina è associata una tabella di routing che specifica il routing per gli indirizzi locali, gli indirizzi delle reti presenti sulle LAN collegate alle schede di rete e l'indirizzo del gateway di default. Nella suite **iproute2** già questa tabella di routing si divede in tre regole di routing chiamate **local**, **main** e **default**.

Queste regole vengono applicate a tutti pacchetti in transito e mentre la regola local ha una priorità molto elevata le altre hanno una bassa priorità così che, se l'utente decide di aggiungere nuove regole di routing, queste possono essere applicate prima di quelle di **main** e di **default** che eventualmente possono essere ignorate.

Consideriamo quindi una semplice situazione, come descritta sopra, in cui vi sia necessità di attivare un source routing e supponiamo che la situazione base del router sia di avere come gateway di default quello sulla connessione forfetaria.

Se vogliamo instradare un indirizzo IP sulla connettività veloce possiamo aggiungere una tabella di routing a priorità più elevata rispetto alla **main** e alla **default** e far applicare tale tabella solo ai pacchetti provenienti dall'indirizzo IP in questione. Tale tabella conterrà ovviamente una nuova impostazione del gateway di default puntando sul router collegato alla connessione più veloce.

Vediamo come si esegue molto semplicemente la cosa. Le regole di routing di default attivate sono come detto le:

```
[root@K7 root]# ip rule list
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

dove si evince, nell'ordine il numero di priorità della regola, il campo di applicazione della regola (*from all*) e il nome della regola.

Aggiungiamo una regola di routing da applicare solo nel caso di indirizzo sorgente **xxx.xxx.xxx.xxx**

```
[root@K7 root]# ip rule add from xxx.xxx.xxx.xxx table linkFast
```

e se andiamo a visualizzare le regole di routing

```
[root@K7 root]# ip rule list
0:      from all lookup local
32765   from xxx.xxx.xxx.xxx lookup linkFast
32766:  from all lookup main
32767:  from all lookup default
```

troviamo che la nuova regola verrà applicata prima della **main** e della **default**. Come detto in questa regola aggiungiamo il gateway di default al router collegato al link veloce **yyy.yyy.yyy.yyy** tramite il codice

```
[root@K7 root]# ip route add default via yyy.yyy.yyy.yyy dev eth1 ⤵
 table linkFast
```

e questo è tutto. I pacchetti provenienti da **xxx.xxx.xxx.xxx** troveranno come gateway di default il router **yyy.yyy.yyy.yyy**. Tutti gli altri seguiranno la via più lenta.

In realtà è possibile, accoppiando alle funzioni di **iproute2** quelle di firewall **iptables** eseguire del routing basandosi su regole qualsivoglia complesse.

Supponiamo per esempio di voler inviare lungo il link veloce anche tutti i pacchetti di tipo web, di modo da far navigare i nostri utenti in modo molto veloce senza caricare questa linea con il traffico di tipo posta elettronica per esempio che non ha bisogno di link veloci ma che ci permette di non caricare troppo il costo sulla banda a consumo.

Ovviamente questa funzionalità non è applicabile utilizzando soltanto la suite **iproute2**. Vediamo come funziona la cosa in accoppiata con la suite **iptables**.

L'algoritmo di routing della suite **iproute2** può basare la sua decisione su quali regole di routing applicare in base all'indirizzo di sorgente come abbiamo or ora visto. Ma può basare l'algoritmo di scelta anche sulla presenza e sul valore di tag nell'header del pacchetto IP. Tale tag può essere inserito da una regola di firewall nel processo di **PREROUTING**.

Quindi, il firewall marca determinati pacchetti in base a regole qualsivoglia complesse e il routing poi applica a questi pacchetti marcati una tabella di routing differente da quella di default.

Possiamo allora scrivere subito la regola di firewall e la regola di routing che sono, spero, sufficientemente esplicative:

```
iptables –A PREROUTING –i eth0 –t mangle –p tcp --dport 80 –j MARK ⤵
 --set-mark 1
```

che applica, prima del processo di routing, un tag a tutti i pacchetti in ingresso dalla scheda interna e destinati a porte web e la regola

```
ip rule add fwmark 1 lookup linkFast
```

che invia tutti i pacchetti che hanno tag impostato dal firewall uguale a uno sul link veloce.

Un esempio reale: il firewall dell'I.Z.S. di Teramo.

Nei primi mesi del 2001 fui contattato dall'Istituto Zooprofilattico Sperimentale "G. Caporale" dell'Abruzzo e del Molise in qualità di consulente per l'analisi e la risoluzione di un'esigenza operativa scaturita dall'acquisto, da parte dell'I.Z.S. di un'ulteriore linea di connettività INTERNET oltre a quella già esistente collegata alla rete universitaria italiana GARR.

Le esigenze che avevano spinto l'I.Z.S. ad fare tale scelta era dettate dalla necessità di operare un aggiornamento di banda e dall'uscita, sul mercato italiano, di soluzioni di connettività di tipo xDSL a basso costo.

L'I.Z.S. ritenendo troppo complicato il renumbering interno necessario qualora avessero deciso di cessazione della linea con il GARR ha preferito aggiungere un altro link e cercare di distribuire il carico sulle due linee tenendo conto della maggiore banda disponibile sul canale xDSL bilanciato però da un tetto di traffico mensile posto contrattualmente dal provider "Interbusiness" sforato il quale si passa ad una fatturazione a consumo.

Per evitare di superare tale tetto l'amministratore della rete interna dell'I.Z.S. ha deciso di distribuire il traffico sulle due linee inviando il traffico web degli utenti interni e tutto il traffico generato da utenti privilegiati sulla più veloce linea xDSL relegando tutto il restante traffico a "bassa priorità" sulla linea GARR più lenta.

Il sottoscritto è stato convocato per l'attuazione di queste politiche.

La soluzione proposta e attualmente in funziona presso l' I.Z.S. è stata quella di mettere un firewall Linux subito sotto il router preesistente e porre su un'altra scheda di rete il router xDSL. Su una terza scheda di rete è stato inserito il collegamento alla dorsale proveniente dalla rete interna.

Il firewall è stato configurato in trasparent bridging in modo da risultare trasparente alle macchine evitando problemi di riconfigurazione degli host interni.

Sul firewall è stata attivata una regola di PREROUTING che marchia i pacchetti provenienti dagli host privilegiati o diretti ai servizi web.

Nel successivo processo di ROUTING ai pacchetti cosi marchiati viene fatta seguire una tabella di routing differente che ha configurato come rotta di default il router xDSL.

Prima di uscire dalla macchina a tali pacchetti viene applicato, nel processo di POSTROUTING un SNAT per evitare che pacchetti con sorgente appartenente alla rete GARR vengano rigettati dalla rete Interbusiness in quanto non provenienti dalla rete assegnata all'I.Z.S.

Quindi, con una soluzione di firewall Linux, e riciclando un personal computer che altrimenti sarebbe stato dimesso si è sciolto un problema che altrimenti sarebbe stata impossibile da risolvere.

Riferimenti

- Linux Networking Howto
- Linux 2.4 Packet Filtering Howto
- Linux 2.4 Nat Howto
- Iproute2 Utility Suite Howto
- Linux 2.4 Advanced routine Howto
- MonMotha’s Firewall configuration files.