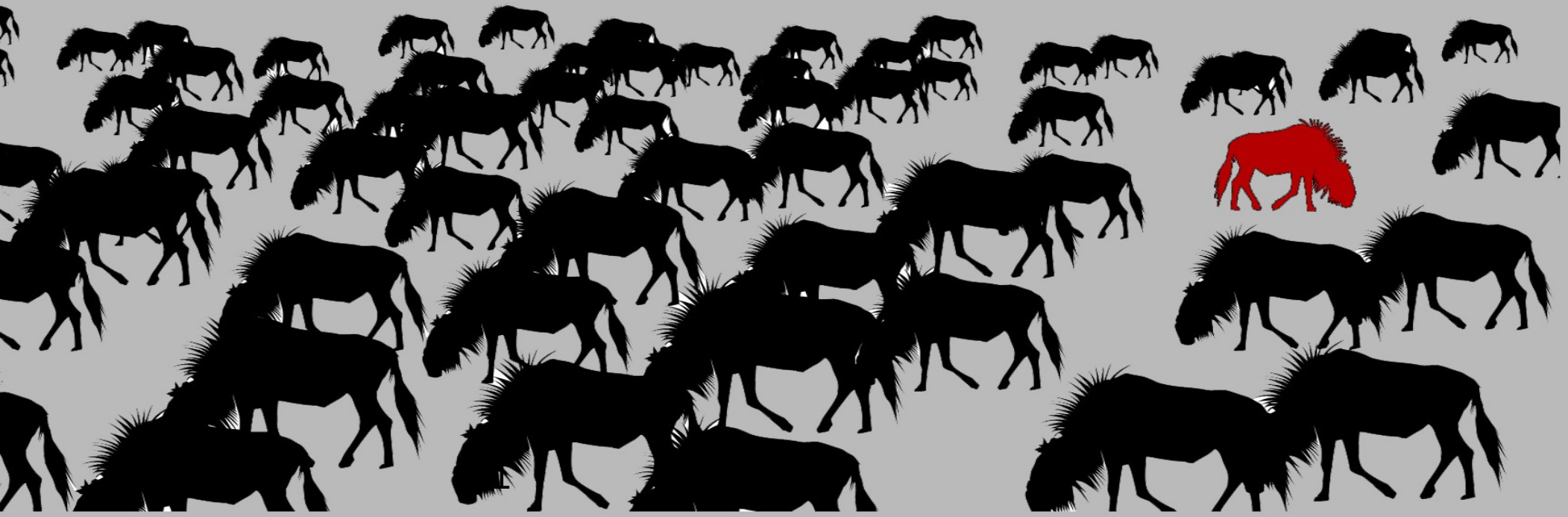


Linux Networking: tips&tricks

MSACK::Hacklab

msack.c4occupata.org

In questo Workshop:

- Settare il networking di base sulla nostra macchina GNU/LINUX

- Imparare a NATTARE la connetività

- Configurare la nostra piccola workstation come un router/firewall

- Firewalling e Hardening di base

...e, ovviamente

“we learn how to build things in the way we can smash ad use them”

Daremo uno sguardo generale agli strumenti concepiti per la gestione, l'amministrazione e il monitoring del networking, e di come questi ultimi possono tornarci utili nel momento in cui passiamo dal ruolo di Bob (il sysadmin) a Darth...r0tfl

Linux Networking Tips&Tricks: Parte 0: intro

- **Il Sistema Operativo:** ...non scherziamo.

- **La distro:** debian (ma le cose che vedremo sono equivalenti su tutti i sistemi *nix eccetto iptables (solo sistemi GNU/Linux))

- **L'Hardware:** in generale qualsiasi pc che riesca in prestazioni a superare il piccolo router di casa, va bene, a patto che abbia un sufficente numero di schede di rete (2+ dipende dall'uso che ne faremo).

- **Un caso particolare:** OpenWRT.

Parte 0 : il contesto

- **Capire il tipo di connessione che abbiamo:** questo è fondamentale per le strategie che adotteremo durante l'assestment.

- **Il modem:** dov'è?

- **Settaggi:** Routed/Bridged

- **Connessioni over ethernet** (ad es. Aula C4)

Parte 1: Connessione PPPoE/PPPoA

Router domestico settato in bridged mode (solo modem)

• C'è la necessità che il pc che fa da router acquisisca la connessione **WAN** direttamente dal **dslam ATM**, tramite una interfaccia di tipo point to point.
• Il router domestico diventa quindi trasparente e interviene solo come supplicante nell'handshaking adsl (**adslctl**) .
• Il nostro pc dovrà distribuire quindi connettività **ipv4** alla nostra sottorete (**routing e nat**).

Parte 1: pppoe-discovery

Il comando pppoe-discovery, genera un pacchetto di tipo pppoe-discovery incapsulato in un frame ethernet. Questo serve a verificare la prensenza di un aggregatore point to point oltre il nostro doppino casalingo.

bob@open-GW~# ppoe-discovery
pppoe-discovery
Access-Concentrator: r-na162
Got a cookie: 0c 54 9a fb 5b c3 a1 6c dc 77 7f 17 fc ac 3e 7f
---
AC-Ethernet-Address: 00:90:1a:a2:aa:3b

Il DSLAM r-na162 (l'id del concentratore) ci risponde con il suo mac-address e un cookie.

Parte 1: ppooe-discovery/2

Per scrupolo possiamo verificare se il mac ricevuto corrisponde effettivamente ad un apparato di rete del genere, utilizzando i numerosi siti che offrono un servizio di reverse mac-lookup.

Parte 1: UP !

bob@openGW~# pppoeconf
Dovremo rispondere all domande che ci verranno poste.

bob@openGW~# pon “nome-provider” (ad es. aliceadsl)
Questo setterà la connessione su di un interfaccia del tipo ppp0

bob@openGW~# plog
Visualizza i log della sessione pppoe

bob@openGW~# poff
Chiudi la connessione.

Parte 1: UP !/2

```
bob@openGW~# ifconfig
ppp0  Link encap:Point-to-Point Protocol
      inet addr:79.9.245.117  P-t-P:192.168.100.1  Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
      RX packets:23896 errors:0 dropped:0 overruns:0 frame:0
      TX packets:32333 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:3
      RX bytes:2561476 (2.4 MiB)  TX bytes:29999930 (28.6 MiB)
```

L'interfaccia è up e pronta per ricevere e distribuire pacchetti !

Parte 1: Rete Domestica

Nel caso in cui la nostra macchina sia già dietro NAT, quindi la sua connessione è fornita da un router:

bob@openGW~# dhclient eth0
Nel caso in cui gli indirizzi sono assegnati dinamicamente da server dhcp.

bob@openGW~# ifconfig eth0 xxx.xxx.xxx.xxx
Setto su eth0 un ip della sottorete in cui ci troviamo.
bob@openGW~# route add default gw xxx.xxx.xxx.xxx
Imposto come rotta di default il router che provvede alla nostra connessione, ovvero il Gateway.
bob@openGW~# echo “nameserver xxx.xxx.xxx.xxx” > /etc/resolv.conf
Imposto i server DNS.

Parte 2: NAT

Il NAT (network address translating), è il processo di modifica degli indirizzi IP provenienti da una subnet per farli uscire all'esterno da un unico indirizzo.

In Particolare possiamo distinguere:

- S(ource)NAT
- D(estination)NAT
- IP Masquerating
- Port Forwarding

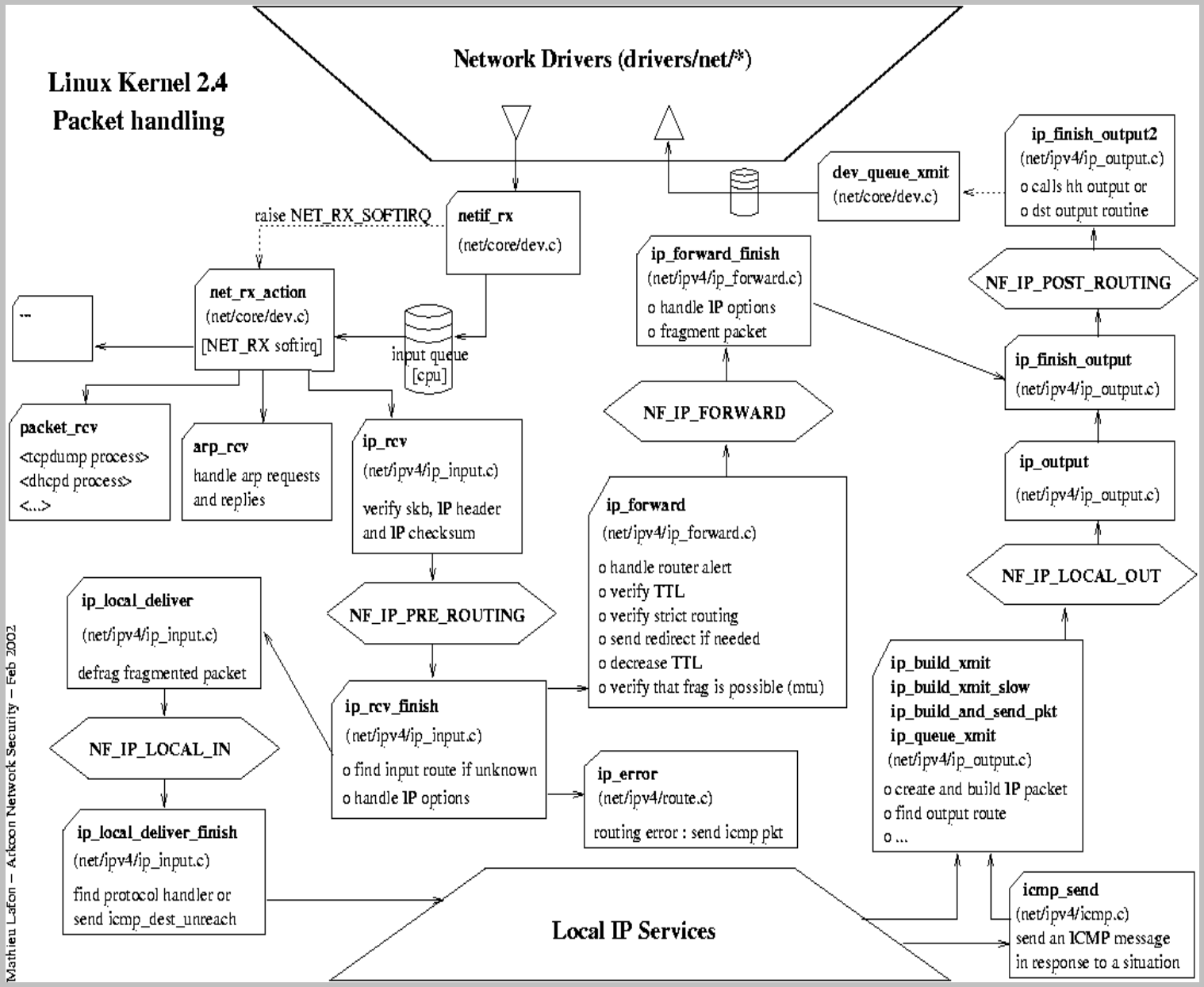
Parte 2: Strumenti per realizzare un NAT

A differenza di altri sistemi operativi il kernel linux è predisposto per effettuare complesse operazioni sui pacchetti che transitano sia nello user-space, che nel kernel space.
In particolare **Netfilter** è la componente (una serie di moduli) che implementa nel kernel gli strumenti di manipolazione filtraggio e gestione dei pacchetti.
Netfilter è il vero e proprio firewall dei sistemi GNU/LINUX.

Parte 2: Gestione di Netfilter

Netfilter viene interfacciato all'utente con varie applicazioni, iptables è la più completa e potente per interfacciarsi ad esso.
Netfilter comunica con lo user-space tramite variabili poste nell'intero albero del fs unix **/proc/net**

Parte 2: Netfilter nel Kernel



Parte 2: Iptables

- Introdotto nel marzo del 2000 durante la fase di sviluppo del kernel 2.4.x (ora siamo a 2.6.x).

- È definito un builder per firewall di tipo SPI (stateful packet inspection).

- SPI, ispezione dei pacchetti “intelligente”, tiene traccia delle connessioni per raggruparle e processarle in cluster (a gruppi).

Parte 2: Organizzazione di iptables

IP tables è organizzato in tabelle, in particolare esistono quattro tabelle prestabilite:

Filter → filtraggio dei pacchetti.
Nat → Network address translating.
Mangle → modifica dei pacchetti (ad esempio agire sul QoS).
Raw → introdotta dalla versione 2.6.x per applicare regole a quei pacchetti che non vogliamo filtrare in modo stateful, bloccando quindi il connection tracking.

È poi possibile aggiungerne altre personalizzate per creare nuovi gruppi di regole.

Parte 2: Organizzazione di iptables/2

Le tabelle a loro volta sono organizzate in catene:

Filter: INPUT-OUTPUT-FORWARD

NAT: PREROUTING-POSTROUTING-OUTPUT

Mangle: INPUT-OUTPUT-FORWARD-PREROUTING-POSTROUTING-OUTPUT

Raw: OUTPUT-PREROUTING

Parte 2: Organizzazione di iptables/3

Le catene sono una forma di lista di controllo degli accessi (**ACL**), ogni regola è costituita da due parti: la specifica delle caratteristiche che un pacchetto deve avere affinché la regola stessa venga applicata (**match**) e un obiettivo o target, che indica cosa fare quando il pacchetto rispetta le caratteristiche indicate. A ciascuna catena è anche associata una **politica di default**, che definisce come vengono trattati i pacchetti che non corrispondono ad alcuna regola.

Parte 2: Organizzazione di iptables/4

Analizziamo ora le regole che possiamo applicare ad ogni pacchetto:

ACCEPT → accetta il pacchetto.
DROP → scarta (ignora) il pacchetto.
REJECT → rifiuta il pacchetto.
RETURN → ritorna il pacchetto (la decisione è presa dalla default policy)
QUEUE → metti in coda (se il pacchetto deve essere processato da una seconda applicazione come ad. es. un IDS)
LOG → logga il pacchetto (viene poi droppato se non viene specificata altra regola)
DNAT → destination nat
SNAT → source nat
MASQUERADE → applica maschera (forma speciale di source nat)

Parte 2: Organizzazione di iptables/5

È anche possibile tenere conto dello “state” di una connessione:

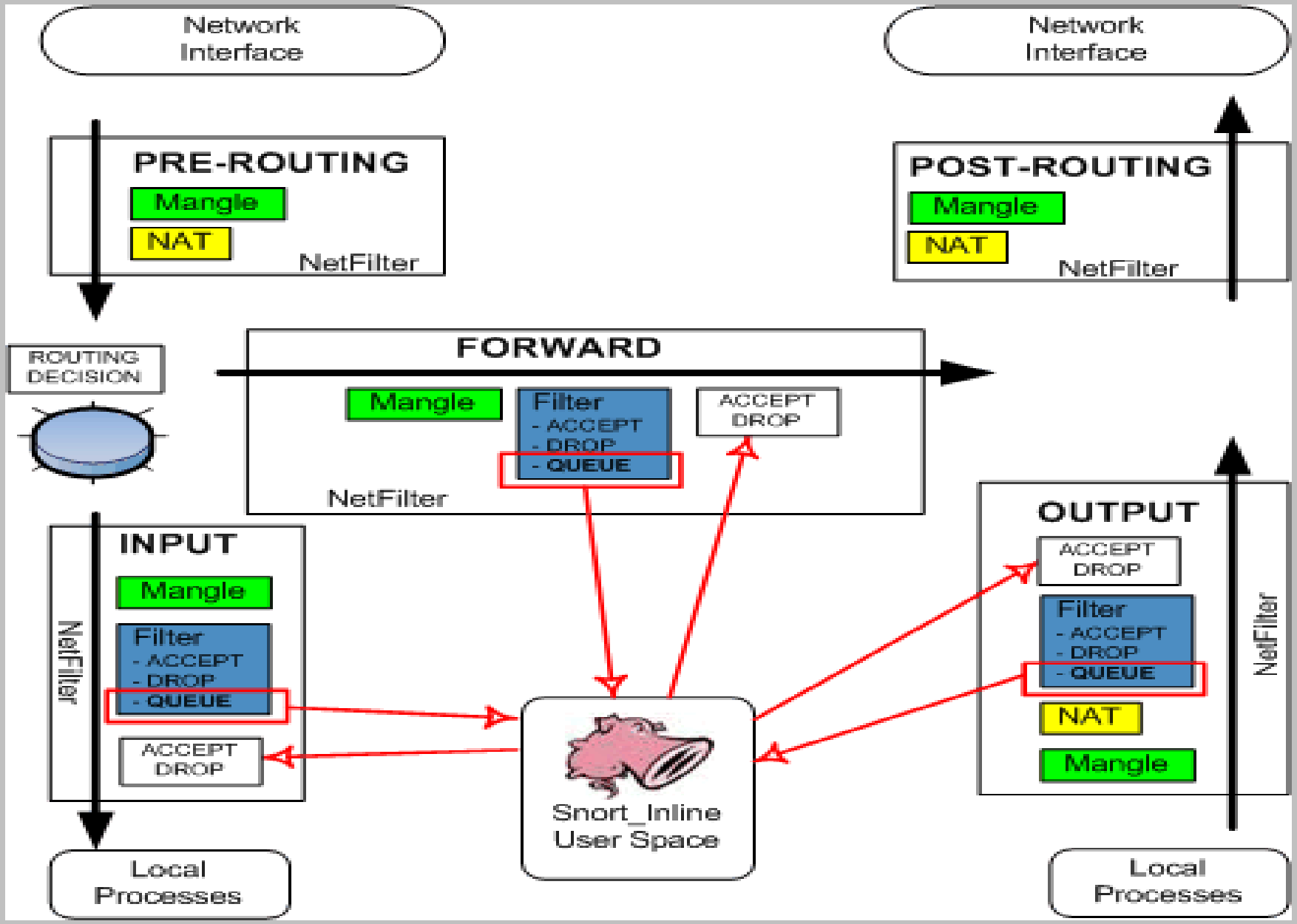
- **NEW (NUOVO)**, il pacchetto inizia una nuova connessione;

- **ESTABLISHED (STABILITO)**, il pacchetto fa parte di una connessione già stabilita;

- **RELATED (IN RELAZIONE)**, il pacchetto ha qualche relazione con un'altra connessione già stabilita;

- **INVALID (INVALIDO)**, il pacchetto non fa parte di alcuna connessione e non è possibile crearne.

Parte 2: Organizzazione di iptables/6



Parte 2: Scrivere il firewall

Nella scrittura di ui firewall è bene tener conto che esso verrà eseguito con uno script bash, quindi per evitare l'accavallamento e la ridondanza di regole, o il loro fallimento, è necessario pulire le tabelle per liberarsi di tutte le regole precedentemente scritte.

Esempio di flushing:

```
iptables -F
iptables -t mangle -F
iptables -t nat -F
iptables -X
iptables -t mangle -X
iptables -t nat -X
```

Parte 2: Scrivere il firewall/2

Nella costruzione di un firewall è buona norma procere alla scrittura di politiche di default alle quali attenersi nel caso in cui il pacchetto non venga processato da regole specifiche

Esempio di politiche di default:

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

Parte 2: Scrivere il firewall/3

Le politiche di default che abbiamo impostato settano un livello di sicurezza già abbastanza soddisfacente per una piccola rete di tipo domestico, ma è importante ricordarsi che la sensazione di sicurezza è un delle più grandi vulnerabilità... quindi vediamo come applicare regole sullo stato delle connessioni che transitano sul nostro router.

Esempio di state rules:

```
iptables -A INPUT -f -j DROP
iptables -A INPUT -m state --state INVALID -j DROP
iptables -A OUTPUT -f -j DROP
iptables -A OUTPUT -m state --state INVALID -j DROP
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

Parte 2: Scrivere il firewall/4

É giunto ora il momento di dare connettività alla nostra sottorete. Finalmente abbiamo le basi per scrivere delle regolte sulla tabella nat.

Esempio di Masquerating

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.1.0/24 -j MASQUERADE
```

Con questo comando diciamo a netfilter che tutte le connessioni provenienti dalla sottorete indicata con **-s (--source)** andranno mascherate se indirizzate verso l'interfaccia eth1

Parte 2: Scrivere il firewall/5

Impostate le regole di NAT, i computer connessi (configurati come precedentemente abbiamo visto) alla nostra sottorete saranno abilitati al traffico verso l'esterno.
Ora è possibile fare le opportune considerazioni su quali servizi aprire verso l'esterno e come aqgire su iptables in modo che solo i pacchetti effettivamente desiderati, transitino sulla nostra macchina.
È giunta l'ora del Port Forwarding

Parte 2: Scrivere il firewall/6

Capito il funzionamento delle regole e dopo essersi tagliati fuori innumerevoli volte dal proprio stesso server, si riuscirà a scrivere una regola come la seguente per l'apertura di una determinata porta sulla macchina che fa da router.

iptables -A INPUT -i eth1 -p tcp --dport 22 -j ACCEPT

Abbiamo appena abilitato le connessioni entranti dall'interfaccia eth1 a connettersi alla porta 22 (ssh server)

Parte 2: Scrivere il firewall/7

Di solito il router/firewall non ospita servizi particolari aperti verso l'esterno, è un compito demandato ad altre macchine della sottorete, vediamo quindi come forwardare una connessione entrante ad una porta di un calcolatore interno al nat.

Soluzione?

Parte 2: Scrivere il firewall/7

Destination Nat(ting) !
Occorre effettuare il redirect con dnat , prima pero' occorre abilitare il traffico in forwarding verso l'interno:

```
iptables -A FORWARD -s xxx.xxx.xxx.xxx -j ACCEPT

iptables -A INPUT -i ppp0 -p tcp --dport 80 -j ACCEPT

iptables -t nat -A PREROUTING -i ppp0 -p tcp -m tcp --dport 5000 -j DNAT --to
192.168.0.202:80
```

Parte 2: Considerazioni generali

Il firewall è lo strumento che interviene per primo nel caso di attacchi provenienti dall'esterno.
Ma non è possibile capire tutto cio' che accade, si limita solo a scartare o accettare i pacchetti.
Per avere una sicurezza massima, ed individuare attacchi di ogni tipo, sia locali che remoti, dobbiamo ricorrere agli IDS.

Parte 3: Intrusion Detection System

- Come i firewall puo' essere sia software che hardware.

- Di solito si colloca dietro interfacce di spanning non dotate di indirizzo IP.

- Filtra con potenti parser tutto il traffico di rete.

- Rileva anche abusi e attacchi in locale.

Parte 3: Snort

Esistono numerosi software IDS, ma in ambiente UNIX (e non solo), i più utilizzati sono Snort e Cisco PIIX ids.
Noi ovviamente ci concentreremo su snort.

Parte 3: Snort/2

Installazione:

Snort è presente in tutti i repository, è quindi facilmente installabile su tutte le distribuzioni.

AcidBase, è un potente strumento per la consultazione dei log di snort.

Parte 3: Snort/3

Per ottimizzare i rilevamenti di snort bisogna scegliere attentamente l'interfaccia su cui metterlo in ascolto.
Conviene creare un interfaccia di spanning attraverso la quale venga replicato tutto il traffico del nostro router.

Come fare?

Parte 3: Snort/4

Grazie a **brctl** possiamo creare un bridge di tutte le interfaccie dalle quali vogliamo attingere informazioni.