Introduction of Iptables

J.Y.Lee
a5000ml@yahoo.com.tw
0932-953-725

Outline

- Introduction of iptables
- Flow Monitoring
- NAT
- Firewall

Iptables & Programming

- Programmable
- Procedure Oriented
- The Data are Packets
- The Main Use is Filtering the Packets
- Can set User-Defined Subroutines
- Support Run-Time Context Change

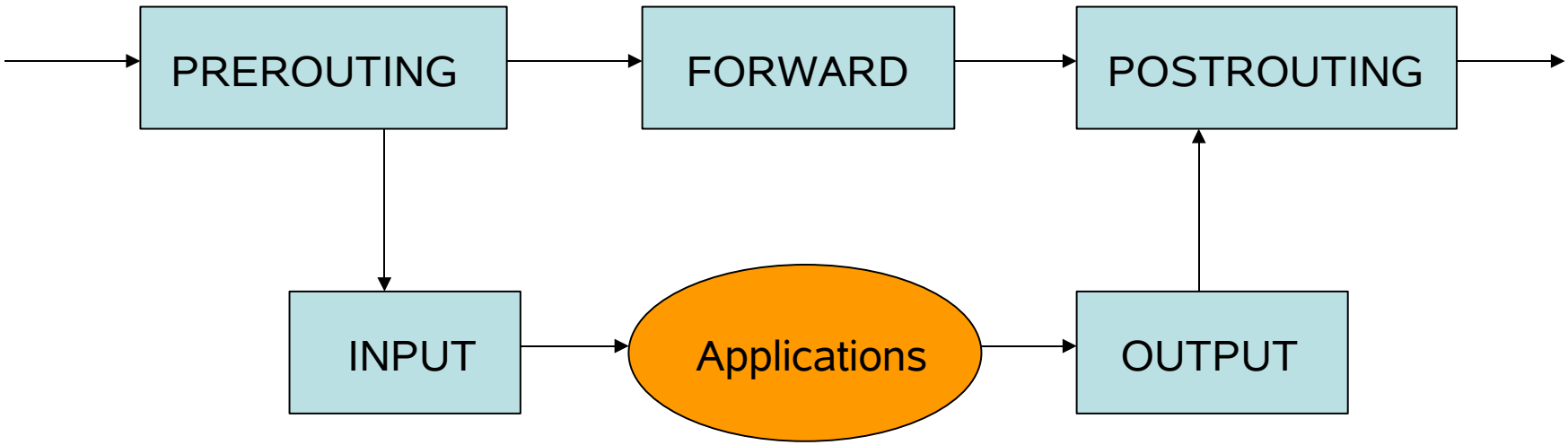
Chains (The Program Unit)

- Subroutines
- Contains many Compare and Jump Statements called “Rules”
- Built-in Chains :

  PREROUTING, POSTROUTING,

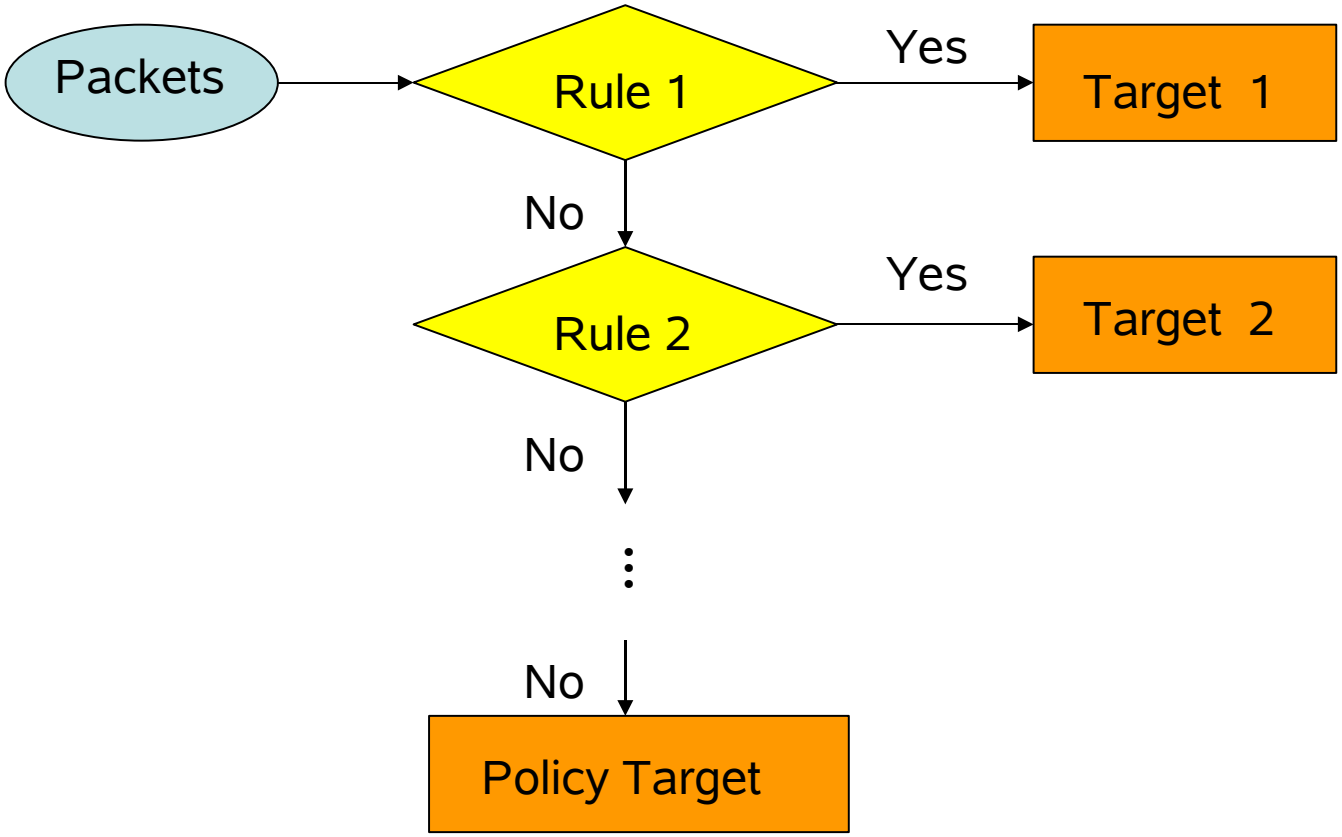
  FORWARD, INPUT, OUTPUT

Built-in Chains & Packets Flow

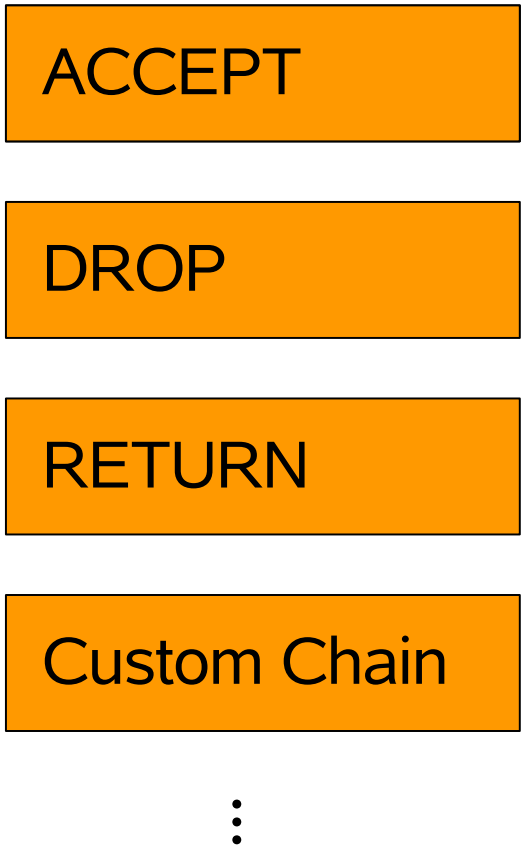


Rules

- Compare the Packets
- Jump to Which Target Unit



Targets



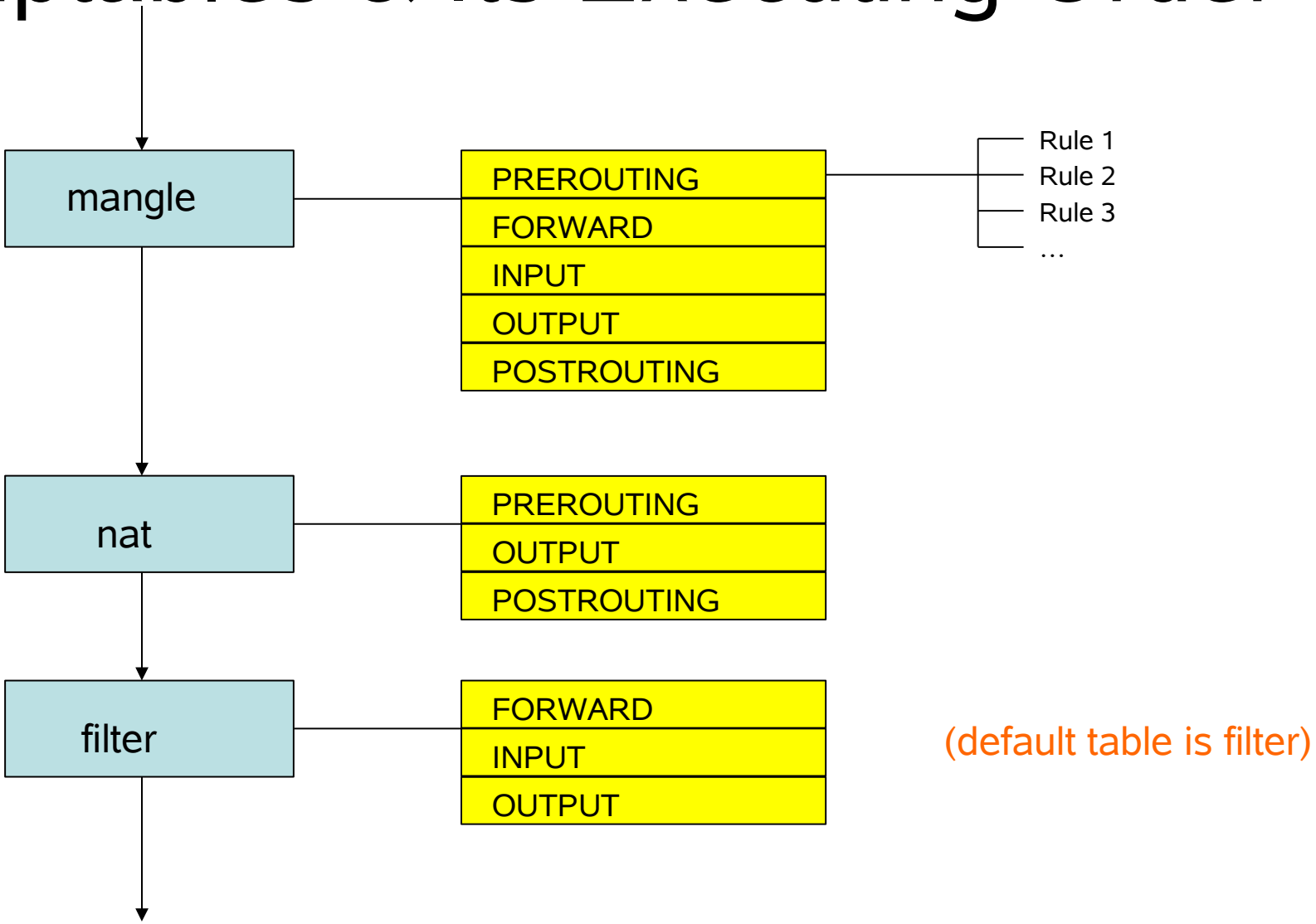
Accept the Packet to Next Chain

Drop the Packet

Return to the Parent Chain

User-Defined Chain (Subroutine)

iptables & Its Executing Order



Add & Delete Rules & Chains

- -I <chain> insert rule to the chain
- -A <chain> append rule to the chain
- -D delete rule
- -R replace rule
- -F [chain] flush all rules in the chain
- -N <chain> create a new chain
- -X <chain> remove user-defined chain

Compare the Packets

- -t <table> specify which table
- -i <input interface> eg. eth0,eth1…
- -o <output interface>
- -p <protocol type> eg. tcp, udp, icmp …
- -s <source ip address>
- -d <destination ip address>
- --sport <source port>
- --dport <destination port>

Jump & Policy

- -j <target> jump to which target
- -P <target> policy target (default target)

Targets:

ACCEPT, DROP, RETURN, MASQUERADE,

SNAT, DNAT, User-Defined Chain, …

Query

- -L list rules in iptables
- -h brief help

--- output extensions ---

- -v verbose output
- -n numerical output
- -x expand number output
- --line-numbers

Brief Help

- Examples:

  iptables -h

  iptables -m mac -h

  iptables -j DNAT -h

Examples (Add Rules)

- iptables -A OUTPUT -p icmp -d www.google.com -j DROP

  in the OUTPUT chain of filter table, append a rule, that all icmp packets to www.google.com will be DROP

- iptables -t mangle -I POSTROUTING -p tcp -d www.google.com --dport 80 -j DROP

  in the POSTROUTING chain of mangle table, insert a rule, that all tcp packets to www.google.com:80 will be DROP

No Ordered Rule Parameters

- When assign a Rule, the parameters order is NOT important.
  For example, the following assignments are equivalent.

- iptables -A OUTPUT -p icmp -d www.google.com -j DROP
- iptables -p icmp -A OUTPUT -d www.google.com -j DROP
- iptables -j DROP -p icmp -d www.google.com -A OUTPUT
- iptables -d www.google.com -j DROP -A OUTPUT -p icmp

- All means: in the OUTPUT chain of filter table, append a rule, that all icmp packets to www.google.com will be DROP

Examples (Delete Rules 1)

- iptables -A OUTPUT -p icmp -d www.google.com -j DROP
- iptables -A OUTPUT -p tcp -d www.google.com -j DROP
- iptables -A OUTPUT -p tcp -d tw.yahoo.com -j DROP

  (add 3 rules)
- iptables -D OUTPUT -p tcp -d www.google.com -j DROP

  (delete the rule of dropping tcp of google)

Result: We can connect to google but still can't ping it, and we can't connect to yahoo.

- iptables -F OUTPUT

  (flush all rules in OUTPUT chain)

Result: We can connect to google and yahoo, and can ping google.

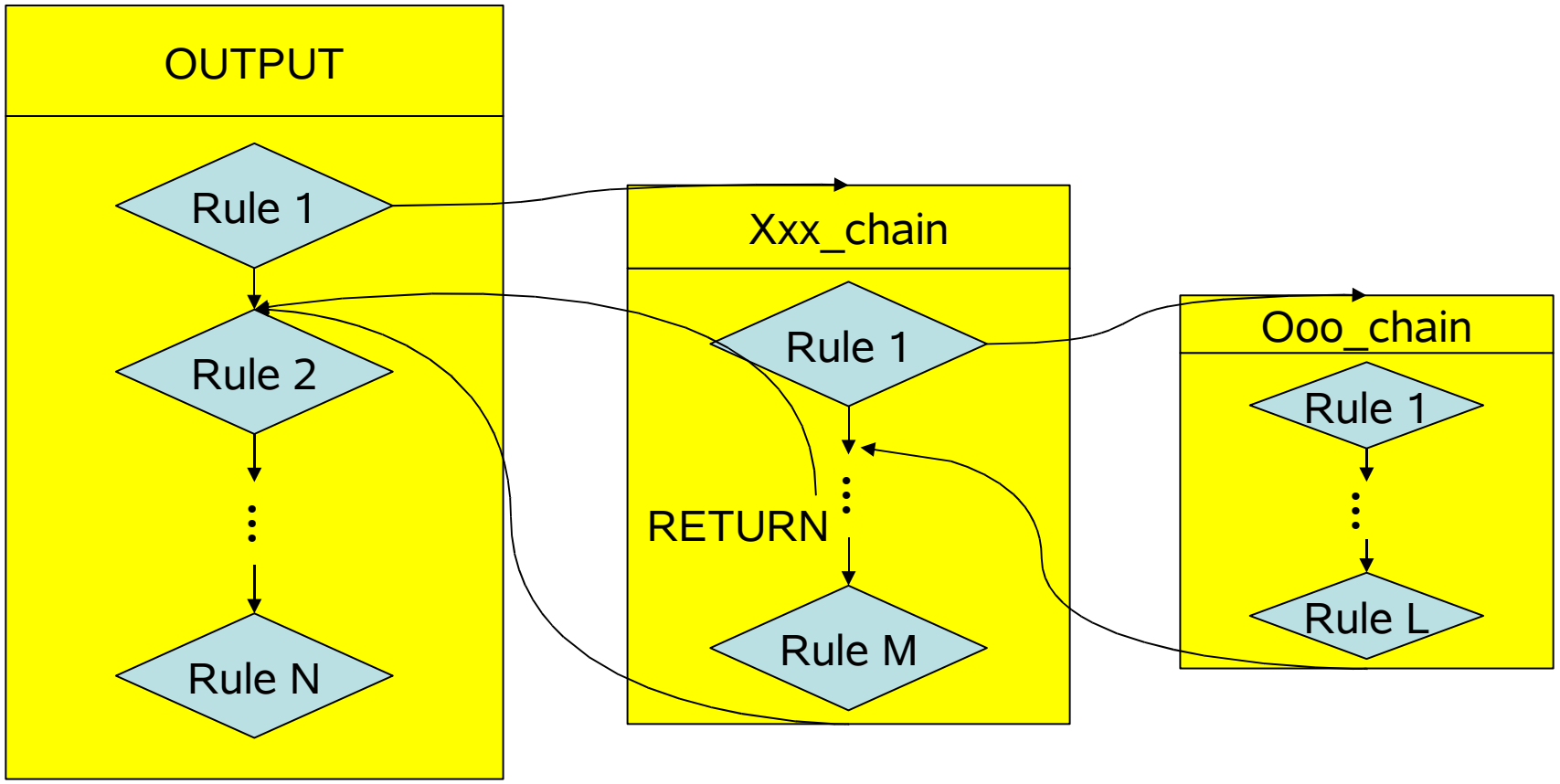
Examples (Delete Rules 2)

- iptables -A OUTPUT -p icmp -d www.google.com -j DROP
- (add icmp drop rule)
- iptables -L
  (list the rules)
- iptables -D OUTPUT -p icmp -d 66.102.7.147 -j DROP
  (delete the rule in one of google ip)
- iptables -L
  (list the rules)

Result: We can see the iptables store the rule in numerical address, and although we just add one rule, the www.google.com have 3 ip addresses, so we got 3 rules.

User-Defined Chain

User-defined chains act as the role of subroutines in programming.



List in Verbose Mode

iptables -L [chain] -v

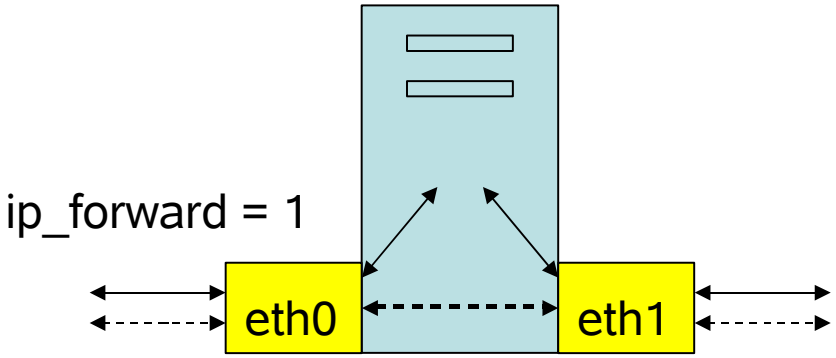
pkt	# of packets flowed	
bytes	# of bytes flowed	(use this to flow statistics)
target	Rule target	
prot	protocol	
opt	options	
in	input interface	
out	output interface	
source	source ip address	
destination	destination ip address	

Flow Statistics

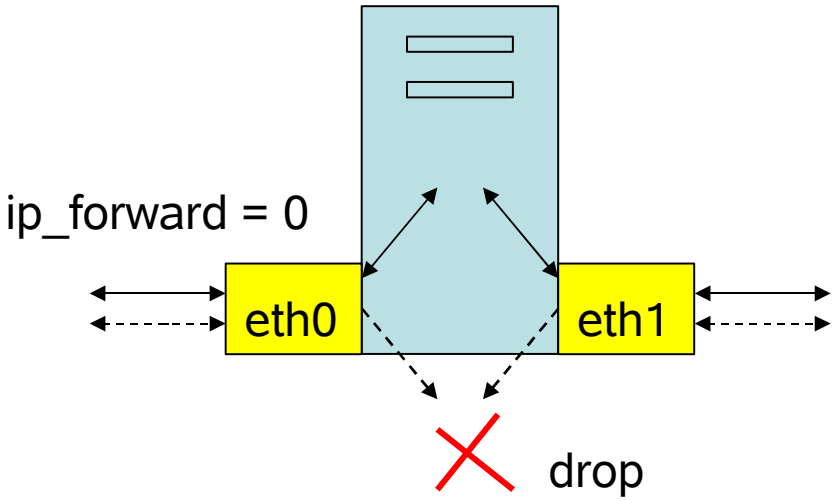
- iptables -N www
- iptables -A INPUT -p tcp --sport 80 -j www
  (make a new subchain www, and set that all http input call it)
- iptables -A www -s tw.yahoo.com -j RETURN
- iptables -A www -s www.ntu.edu.tw -j RETURN
- iptables -A www -s www.pchome.com.tw -j RETURN
  (set 3 monitors in www, monitoring the source ip address)
- iptables -I www
- iptables -A www
  (set 2 empty monitors in www, monitoring the total & exclusive flow)
- iptables -L INPUT -v
- iptables -L www -v -x
  (list the INPUT & www chains in verbose mode, to see the www flow and these 3 website flow, -x to expand number)
- iptables -Z www
  (reset the packet & byte counters in subchain www)

The ip_forward Option

/proc/sys/net/ipv4/ip_forward



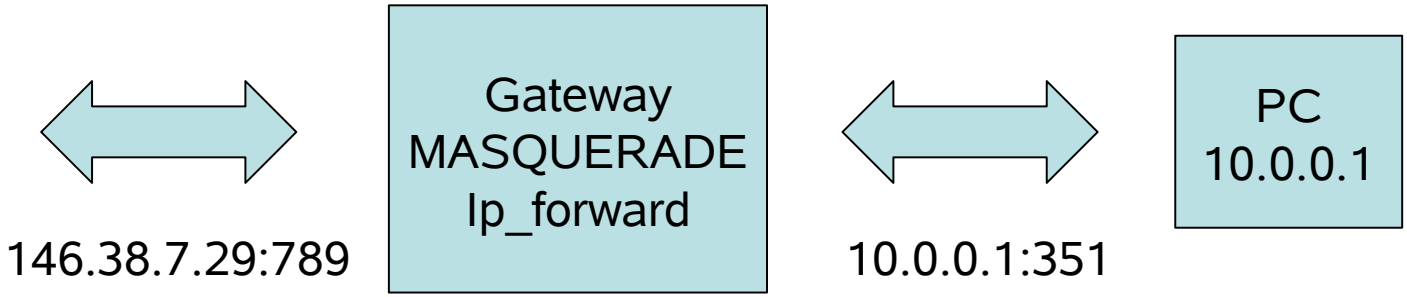
Packets Associated to This Machine



Packets Not Associated to This Machine

MASQUERADE

- This Target is only valid in nat table, POSTROUTING chain.
- The main use is for NAT (Network Address Translation).



NAT (Server)

- If we have 2 NIC, eth0 connect to Internet while eth1 connect to local net, and we want to configure this machine as a NAT gateway, the procedure to this effect is as follow.

- ifconfig eth1 10.0.0.254
  (set eth1 ip address as 10.0.0.254)
- echo 1 > /proc/sys/net/ipv4/ip_forward
  (bring up ip_forward)
- iptables -t nat -A POSTROUTING -j MASQUERADE
  -o eth0 -s 10.0.0.0/24
  (configure NAT ip MASQUERADE for all packets want to go out through eth0 and whose ip in the range of 10.0.0.0~10.0.0.254)

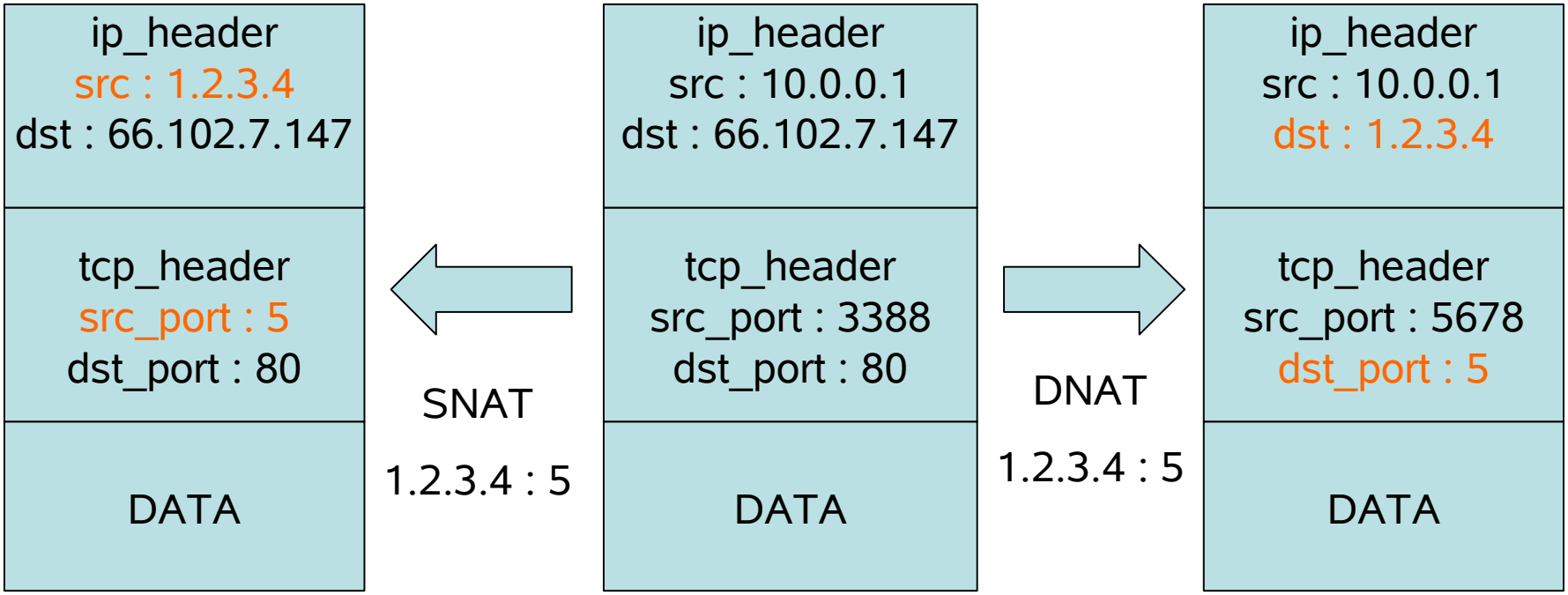
NAT (Client)

- ifconfig eth0 10.0.0.2/24
  (set client eth0 ip address as 10.0.0.2 subnet mask 24bit)

- route add default gw 10.0.0.254
  (set client default gateway as 10.0.0.254)

Target SNAT & DNAT

- SNAT : Source Network Address Translation
- DNAT : Destination Network Address Translation



Server behind NAT

- http Server
- iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80
  -j DNAT --to 10.0.0.2:80

- ssh Server
- iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22
  -j DNAT --to 10.0.0.2:22

Note:

we must set DNAT in PREROUTING chain, because the dst. ip of these packets are to the gateway, after routing, these packets may be accept by the demon on the gateway or dropped for no suitable demon.

REDIRECT

- Only Valid in nat table, PREROUTING & OUTPUT chain

- Redirect local http issue to Gateway Proxy
- iptables -t nat -A PREROUTING -i eth1 -p tcp -dport 80 -j REDIRECT --to-port 3128

MAC Address Locking

- iptables -t nat -A PREROUTING -s 10.0.0.1
  -m mac --mac-source ! 01:02:03:04:05:06
  -j DROP

in the PREROUTING chain of nat table, append a rule, that all packets from 10.0.0.1 with mac address NOT 01:02:03:04:05:06 will be DROP

Note: ! is a valid prefix in iptables means NOT, you can use it everywhere in iptables.

Firewall

- iptables -P INPUT DROP
- iptables -A INPUT -i lo -j ACCEPT
- iptables -A INPUT-p tcp --dport 80
  --syn -m state --state NEW
  -j ACCEPT
- iptables -A INPUT-p tcp --dport 1024:65535
  -m state --state ESTABLISHED,RELATED
  -j ACCEPT