Topic 4.2 - Iptables

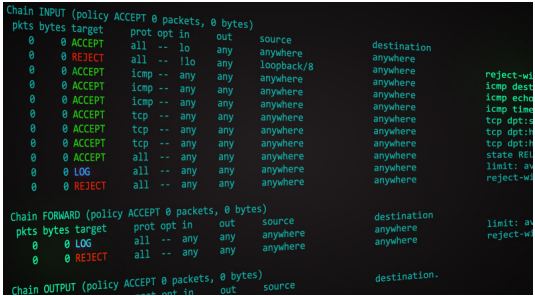
Eric Casanovas

Universitat d'Andorra

27nd April, 2022

Introduction

- Administration tool for IPv4 packet filtering and NAT
- Controls the kernel-level network module called **netfilter**
- Is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel



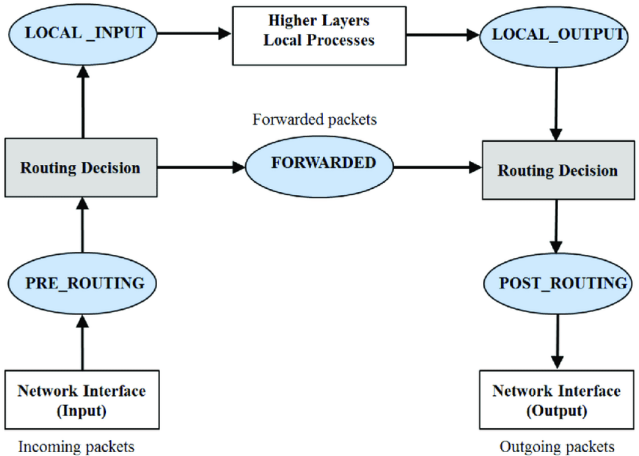
What is Netfilter

- Linux Kernel tool for filtering
- Netfilter uses **rules** for filtering
- Netfilter framework provides a series of "hooks" inside the linux kernel that are traversed by network packets
- Software as iptables and nftables uses these hooks to filter the packets

Hooks I

- There are 5 hooks until Linux 4.2:
  - NF_IP_PRE_ROUTING: Triggered by any incoming traffic before any routing decisions have been made regarding where to send the packet
  - NF_IP_LOCAL_IN: Triggered after an incoming packet has been routed if the packet is destined for the local system
  - NF_IP_FORWARD: Triggered after an incoming packet has been routed if the packet is to be forwarded to another host
  - NF_IP_LOCAL_OUT: Triggered by any locally created outbound traffic as soon it hits the network stack
  - NF_IP_POST_ROUTING: Triggered by any outgoing or forwarded traffic after routing has taken place and just before being put out on the wire

Hooks II


LOCAL _INPUT
Higher Layers
Local Processes
LOCAL_OUTPUT
Forwarded packets
Routing Decision
FORWARDED
Routing Decision
PRE_ROUTING
POST_ROUTING
Network Interface
(Input)
Network Interface
(Output)
Incoming packets
Outgoing packets

Iptables

- Iptables uses netfilter to filter the traffic
- Uses rules to filter the traffic
- Uses tables to organize the rules
- Further, rules are also organized by chains
- Chains represent the netfilter hooks which trigger them

Chains I

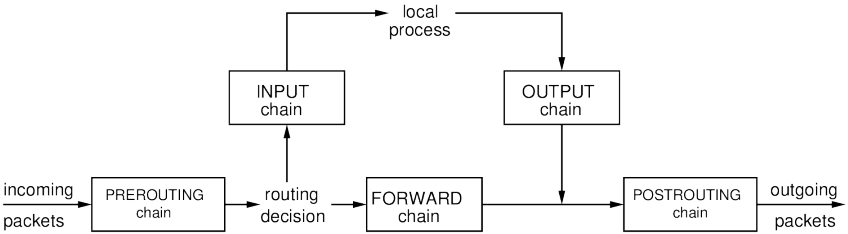
- Chains allow to control where in a packet's delivery path a rule will be evaluated
- Each table has multiple chains depending on the decisions
- Chains from multiple tables are registered at each of the hooks
- Rules in the same chain but different table will be evaluated sequential depending on the priority

Chains II

There are 5 chains (one for each netfilter hook):

- PREROUTING: Triggered by the NF_IP_PRE_ROUTING hook
- INPUT: Triggered by the NF_IP_LOCAL_IN hook
- FORWARD: Triggered by the NF_IP_FORWARD hook
- OUTPUT: Triggered by the NF_IP_LOCAL_OUT hook
- POSTROUTING: Triggered by the NF_IP_POST_ROUTING hook

Chains III


local
process
INPUT
chain
OUTPUT
chain
incoming
packets
PREROUTING
chain
routing
decision
FORWARD
chain
POSTROUTING
chain
outgoing
packets

Tables I

- Tables classify rules according to the type of decisions they are used to make
- Each table has associated some chains
- Tables + Chains -> type of decision + where packet evaluated

Tables II

There are 5 tables:

- Filter: INPUT, OUTPUT, FORWARD
- NAT: OUTPUT, PREROUTING, POSTROUTING
- Mangle: INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING
- Raw: OUTPUT, PREROUTING

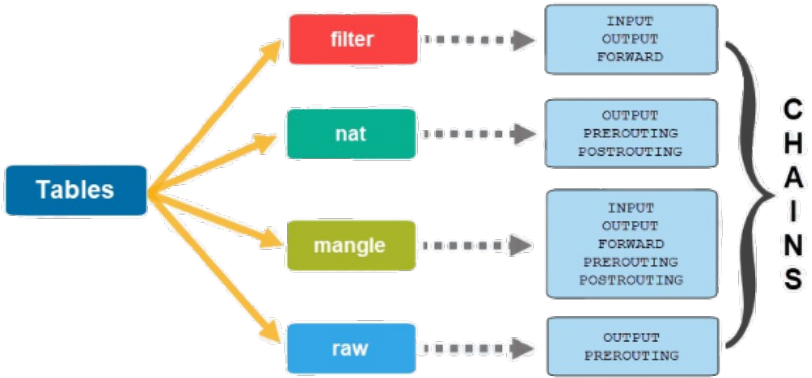
Tables III

- Filter:
  - Is one of the most widely used
  - Used to make decisions about whether to let a packet continue to its intended destination or to deny its request
  - This table provides the bulk of functionality that people think of when discussing firewalls.
- NAT:
  - Used to implement network address translation rules
  - This table will determine how to modify the packet's source or destination addresses in order to impact the way that the packet and any response traffic are routed
  - Used to route packets to networks when direct access is not possible

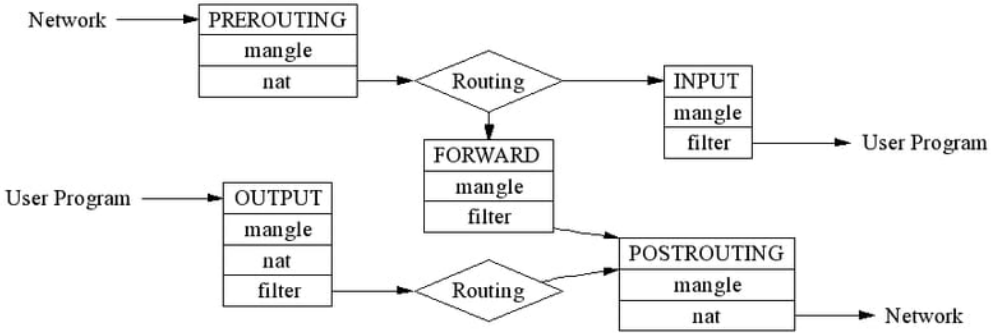
Tables IV

- Mangle:
  - Used to alter the IP headers of the packet in various ways
  - For example, modifying the TTL (Time-To-Live) value of a packet
- Raw:
  - Its purpose is to provide a mechanism for marking packets in order to opt-out of connection tracking

Tables V


Tables
filter
nat
mangle
raw
INPUT
OUTPUT
FORWARD
OUTPUT
PREROUTING
POSTROUTING
INPUT
OUTPUT
FORWARD
PREROUTING
POSTROUTING
OUTPUT
PREROUTING
CHAINS

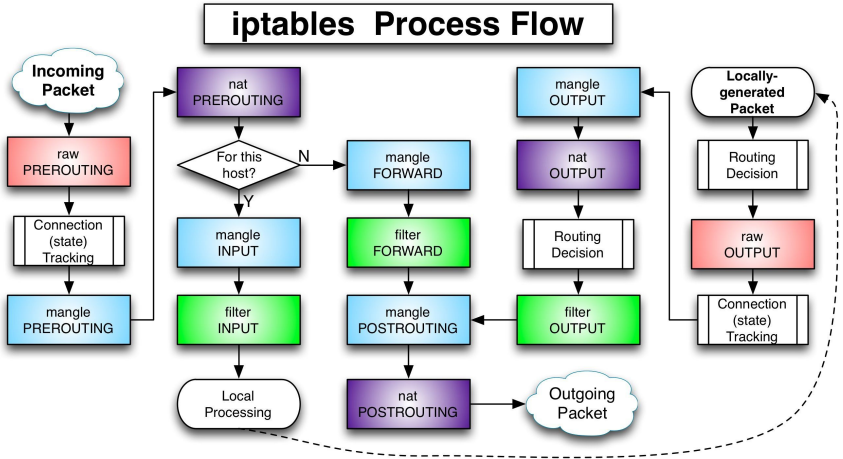
Tables VI


Network
PREROUTING
mangle
nat
Routing
INPUT
mangle
filter
User Program
FORWARD
mangle
filter
User Program
OUTPUT
mangle
nat
filter
Routing
POSTROUTING
mangle
nat
Network

Tables and Chains I

- What happen when two tables have PREROUTING chains, in which order are they evaluated?

Tables and Chains II



Tables and Chains III

```
sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
sudo iptables -t mangle -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
sudo iptables -t raw -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Rules syntax I

iptables <table> <op> <chain> <params>

- table: Selects the table to work with
  - "-t table_name" where table_name is a the name of a table (filter, nat, mangle, raw, security)
  - The dafault table is filter (selected if no table is specified)
- chain: The chain name to operate with:
  - One of the chains supported by the table, for example, for filter: INPUT, OUTPUT, FORWARD.

Rules syntax II

iptables <table> <op> <chain> <params>

- op: Desired option to perform to the rule:
  - "-A": Append a new rule to the chain
  - "-C": Check if the rule exists
  - "-D": Delete the rule
  - "-L": List the rules
  - And many others (see $man iptables)

Rules syntax III

iptables <table> <op> <chain> <params>

- params: There are many params that we can use, but the most important are:
  - "-i interface_name": input interface (e.g. eth0)
  - "-o interface_name": output interface
  - "-p protocol": the protocol (e.g. ICMP)
  - "-s ip/hostname": the source ip/hostname
  - "-d ip/hostname": the destination ip/hostname
  - "-j action": the action to perform
    - In case of filter table: DROP (drops the packet) or ACCEPT (accepts the packet)
    - In case of nat table: DNAT or SNAT (see $ man iptables-extensions)

Examples I

- Add a rule in the INPUT chain to drop packets whose source IP address is 192.168.1.1 and protocol TCP
- Reject all pings incoming and outgoing pings

Examples II

- Add a rule in the INPUT chain to drop packets whose source IP address is 192.168.1.1 and protocol TCP
  - iptables -t filter -A INPUT –s 192.168.1.1 -p tcp -j DROP
- Reject all pings incoming and outgoing pings
  - $sudo iptables -t filter -A INPUT -p icmp –icmp-type echo-request -j DROP
  - $sudo iptables -t filter -A OUTPUT -p icmp –icmp-type echo-reply -j DROP

The END!