Netfilter Basics
(iptables)

What is Netfilter?

- Kernel package
- Packet filtering & manipulation (mangling)
- Consists of many individual kernel modules

  ... and growing

  Only a dozen needed for “basic” functions

Uses

- Firewall
- Routing
- Transparent proxy / cache
- Bridging
- Use policies

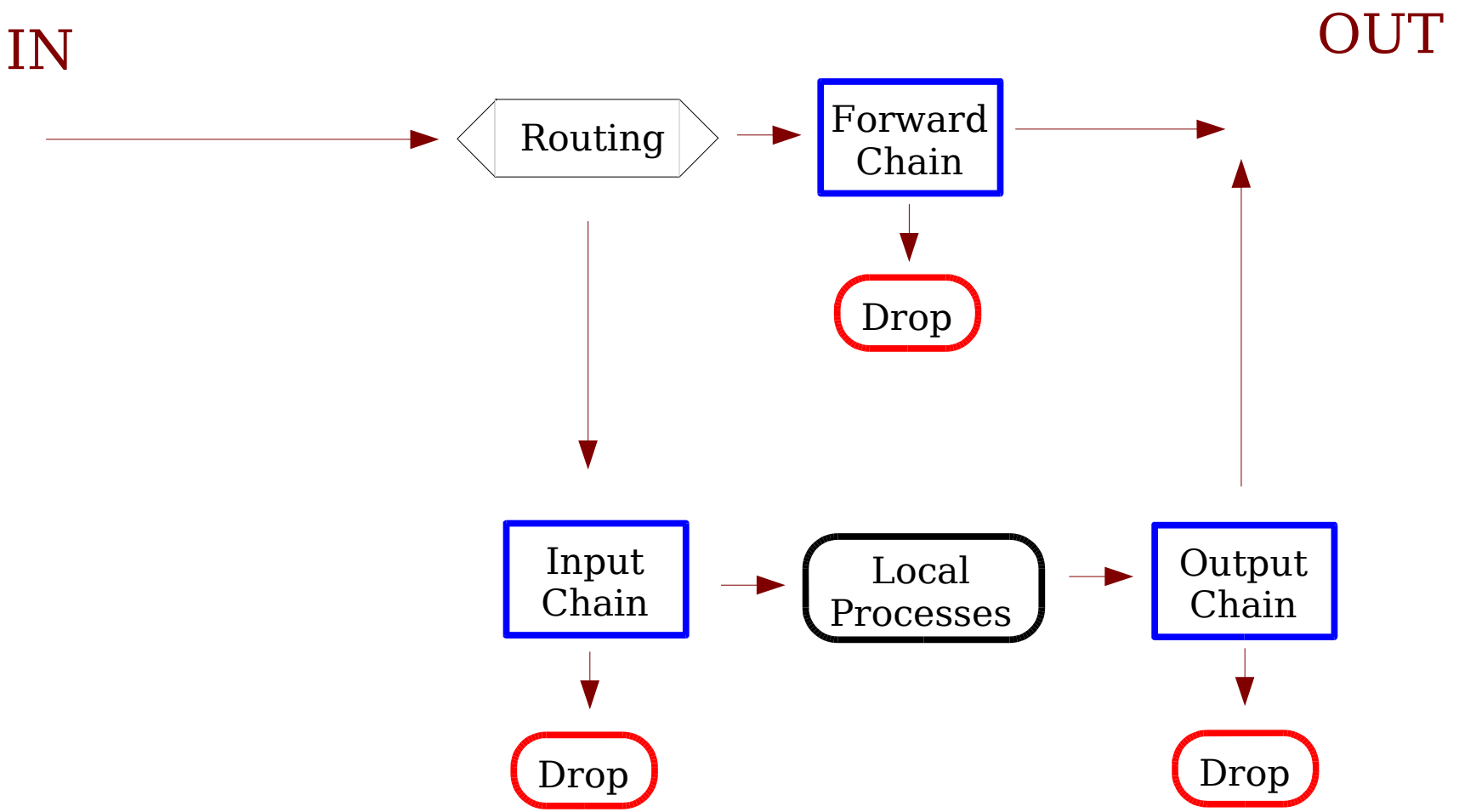
History

- Ported from BSD in mid-1990s
- IPFW
  ipfwadm - 2.0 kernel
  ipchains - 2.2 kernel
- NETFILTER
  iptables - 2.4 & 2.6 kernels

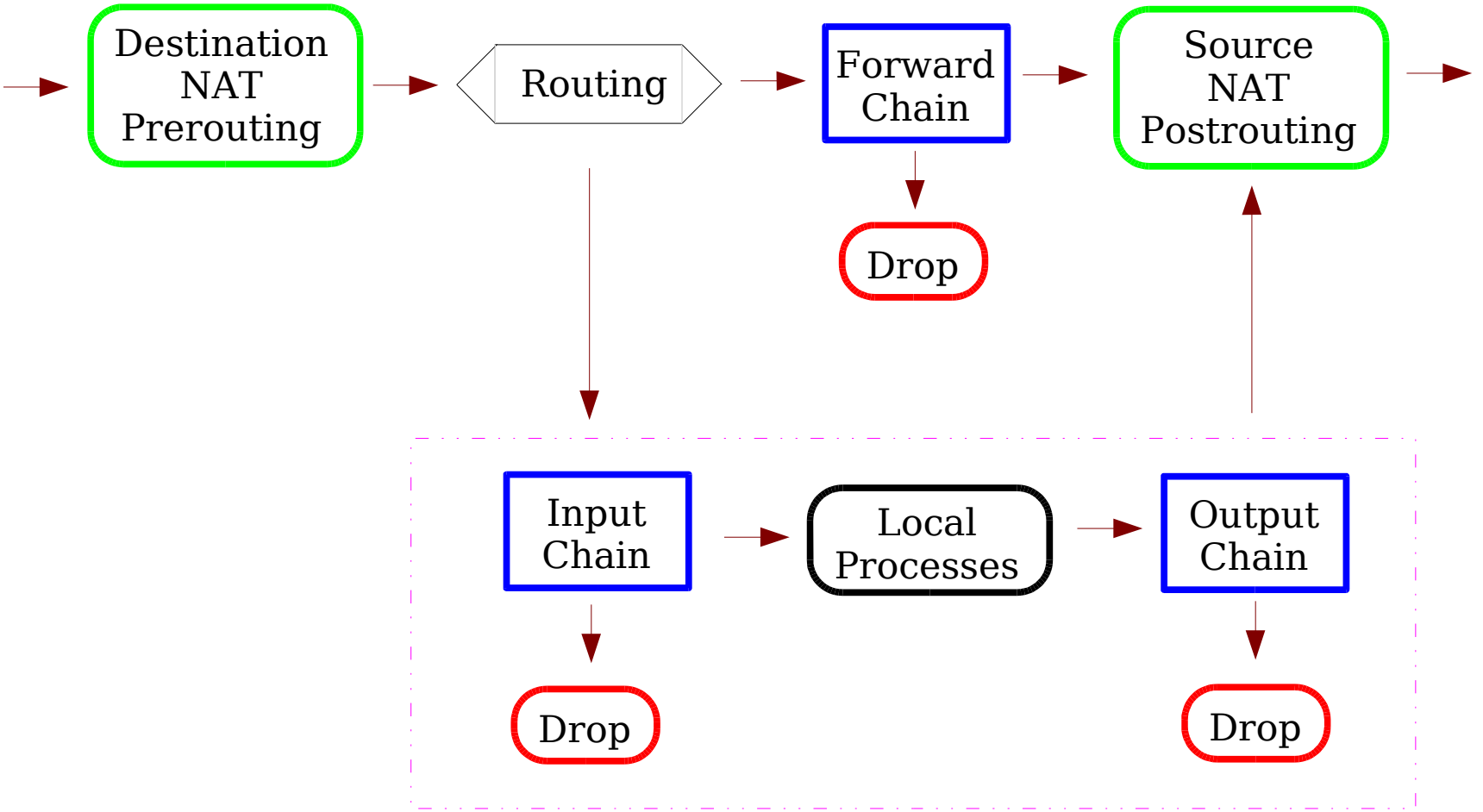
Kernel Configuration

- Implemented in 2.4 & 2.6 kernels
- Consists of many (58) individual features (kernel 2.6.11.12)
- Only a dozen modules for basic configurations
- 2.6.11.12 kernel configuration
  - Device Drivers
    - Networking Support
      - Networking Options
        - Network Packet Filtering
          - IP: Netfilter Configuration

Life of a Packet



Life of a Packet



Rule Tables

- Different types of packet processing

- ***filter*** table is the default
  - Input chain
  - Output chain
  - Forward chain
- ***nat*** table
  - Prerouting chain
  - Postrouting chain
- ***mangle*** table
  - Prerouting chain
  - Output chain

Filter Table

RULES & TARGETS

If a rule matches

The target is executed

No more rules in chain checked

If a rule doesn't match

The next rule in chain is checked

Filter Table Targets

- Accept
- Drop
- Reject
  ICMP type, tcp reset, echo reply
- Log
  Then continues with next rule
- User chain

Table Rules (Match)

For both SOURCE or DESTINATION
- IP address
- Port (tcp & udp)

For both INPUT and OUTPUT
- Physical network device (interface)

- Protocol (tcp, udp, icmp, all)
- TCP flags
- ICMP type
- MAC source address (“in” interface)

Table Rules (Match)

- State
  New, Established, Related, Invalid

- Limit
  Initial burst, maximum in time frame
  (seconds, minutes, hours, days)

Iptables Example

Drop inbound packets from my neighbor

```
iptables  -A  INPUT  --source  64.200.123.123  -j DROP
```

and / or

```
iptables -A INPUT
        --in-interface  eth0
        --mac-source  fe:00:0e:12:34:56
        --jump  DROP
```

/usr/sbin/iptables

Filter Table Rules (Match)

if

match **AND** match **AND** match **AND** match ...

then

target

else

check next rule

Filter Table Rules (Match)

Rule

[iptables] [chain] [match] [match] [target]

Next Rule

[iptables] [chain] [match] [match] [target]

Next Rule

[iptables] [chain] [match] [match] [target]

Filter Table Rules Setup

Flush all rules in a table except user

iptables -F

Delete all user defined chains in a table

iptables -X

Flush all rules from **nat** table

iptables -F -t nat

Set chain policy

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

Append rule to a chain

iptables -A INPUT .....

Filter Table Rules Setup

```
iptables -F
iptables -X

iptables -P  INPUT  DROP
iptables -P  OUTPUT  DROP
iptables -P  FORWARD  DROP

iptables -A  INPUT .....
iptables -A  INPUT ....
iptables -A  OUTPUT ....
iptables -A  INPUT ....
iptables -A  FORWARD ...
```

Filter Table Rules Setup

- rc.firewall

- List rules

  iptables -L
  iptables -L INPUT
  iptables -L -t nat
  iptables -L -v -n -x --line-numbers

- Clear counters

  iptables -Z

Network Services

Service	Port / Protocol
ftp	21 / tcp
ssh	22 / tcp
telnet	23 / tcp
smtp	25 / tcp
domain	53 / tcp
domain	53 / udp
http	80 / tcp
https	143 / tcp
pop3	110 / tcp

Example # 1

- Flush tables & user defined chains
- Set policies to drop packets
- Permit new & established tcp sessions
- Permit established inbound packets
- Log new TCP sessions attempted from the outside

Note: This rule set is not recommended it is merely an example to show the iptables commands.

Example # 1

```
iptables -F
iptables -P  INPUT  DROP
iptables -P  OUTPUT  DROP
iptables -P  FORWARD  DROP
iptables -A  OUTPUT  -p tcp -m state
      --state NEW,ESTABLISHED -j ACCEPT
iptables -A  INPUT  -p tcp -m state
      --state ESTABLISHED -j ACCEPT
iptables -A  INPUT -p tcp --syn -j LOG
      --log-prefix "(In ZIN New Syn)”
iptables -A  INPUT -p tcp --syn -j DROP
```

Example # 2

```
# Permit DNS traffic

DNS1=”206.13.31.12”
USRPORT=”1024:65535”

iptables -A  INPUT  -s $DNS1 -p udp
      --sport 53 --dport $USRPORT -j  ACCEPT

iptables -A  INPUT -s $DNS1 -p tcp
      --sport 53 --dport $USRPORT -m state
      --state ESTABLISHED  -j  ACCEPT
```

Example # 3

```
# Limit ping rate

LIMIT_SLOW="-m limit --limit 12/minute --limit-
  burst 10"
LIMIT_FAST="-m limit --limit 120/minute --limit-
  burst 50"

iptables -A INPUT  -p icmp --icmp-type echo-reply
  $LIMIT_FAST -j ACCEPT

iptables -A  INPUT -p icmp --icmp-type echo-request
  $LIMIT_FAST -j ACCEPT
```

User Chains

```
# Limit ping rate

iptables -N LIMITPING

iptables -A LIMITPING -p icmp --icmp-type echo-
  reply $LIMIT_FAST -j ACCEPT

iptables -A LIMITPING  -p icmp --icmp-type echo-
  request $LIMIT_FAST -j ACCEPT

...

iptables -A  INPUT  -j  LIMITPING
```

User Chains (Return)

```
# Allow pings but continue checking

iptables -N LIMITPING

iptables -A LIMITPING -p icmp --icmp-type echo-
  reply  -j  RETURN

iptables -A LIMITPING  -p icmp --icmp-type echo-
  request  -j RETURN

...

iptables -A  INPUT  -j  LIMITPING
```

NAT

- Drop externally originated broadcasts

```
iptables -t nat -A PREROUTING -i  eth1
        -d 192.168.0.255 -j DROP
```

- Change outbound address to that of NIC

```
iptables -t nat -A POSTROUTING -0  eth1
        -j  SNAT  --to-source 64.73.3.28
```

Discussion

References

- Linux Firewalls – Second Edition (2002)
  Robert Ziegler
- Linux Firewalls – Third Edition (Oct 2005)
  Steve Suehring, Robert Ziegler

- Rusty Russell

- www.netfilter.org
- www.linuxguruz.com/iptables/
- www.linuxsecurity.com/resource_files/firewalls/IP Tables-Tutorial/iptables-tutorial.html

... and a lot more