Netfilter Tutorial

Lu-chuan (Luke) Kung
kung@uiuc.edu

This presentation is based on the following material:

1. Rusty Russell's presentation at Linux World 2000 Tutorial, http://www.netfilter.org/documentation/tutorials/lw-2000/
2. Oskar Andreasson's presentation at CERT Conference 2002 Proceedings, http://www.certconf.org/presentations/2002/Tracks2002Expert_files/TE-1&2.pdf

Iptables - Basic functionalities - IP Filter

- IP Filter
  - Used to filter packets
  - The command to enter a rule is called iptables
  - The framework inside kernel is called Netfilter
  - Full matching on IP, TCP, UDP and ICMP packet headers
  - Lesser matching on other packet headers possible
  - Exception in TCP is the Options field
- IP Filter rule
  - Insertion point
  - Match
  - Target

Iptables - Basic functionalities - Stateful firewalling

- Full state matching
  - TCP
  - UDP
  - ICMP
- Other protocols
- Uses a generic connection tracking module
  - The generic conntrack module is less specific
  - It is possible to write your own conntrack modules
  - Certain protocols are "complex"
    - Requires extra modules called "conntrack helpers"
    - Examples are FTP, IRC (DCC), AH/ESP and ntalk

Iptables - Basic functionalities - Stateful firewalling (cont.)

- Userland states
  - NEW
    - All new connections
    - Includes Non SYN TCP packets
  - ESTABLISHED
    - All connections that has seen traffic in both directions
  - RELATED
    - All connections/packets related to other connections
    - Examples: ICMP errors, FTP-Data, DCC
  - INVALID
    - Certain invalid packets depending on states
    - E.g. FIN/ACK when no FIN was sent

Iptables - Basic functionalities - NAT

- NAT - Network Address Translation
  - The science of switching Source or Destination Addresses
- Two types of NAT in Linux 2.4
  - Netfilter NAT
  - Fast NAT
- Usages
  - Making a LAN look as if it came from a single source (the firewall)
  - Creating separate servers with a single IP
- Netfilter NAT
  - DNAT - Destination Network Address Translation
  - SNAT - Source Network Address Translation
  - Requires Connection tracking to keep states and expectations

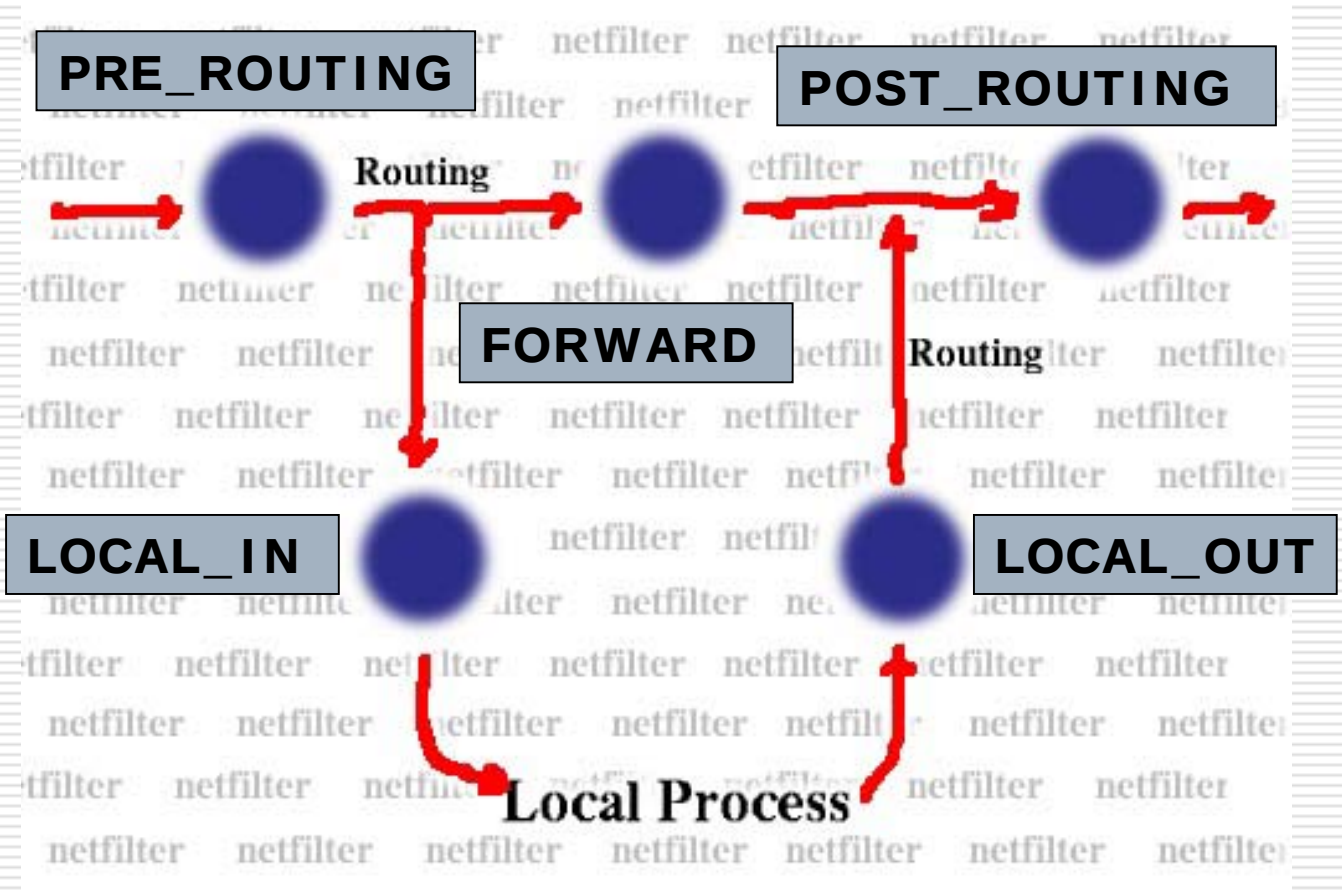
Iptables - Basic functionalities - Packet Mangling

- Mangling packets going through the firewall
- Gives you the ability to a multitude of possibilities.
- Example usages
  - Strip all IP options
  - Change TOS values
  - Change TTL values
  - Strip ECN values
  - Clamp MSS to PMTU
  - Mark packets within kernel
  - Mark connections within kernel

Netfilter Architecture

- The Hooks
  - Parts of the kernel can register with netfilter to see packets at various points in the stack
  - IPv4: PRE_ROUTING, LOCAL_IN, FORWARD, LOCAL_OUT, POST_ROUTING.
  - Each hook can alter packets, return NF_DROP, NF_ACCEPT, NF_QUEUE, NF_REPEAT or NF_STOLEN.

The Hooks (cont.)

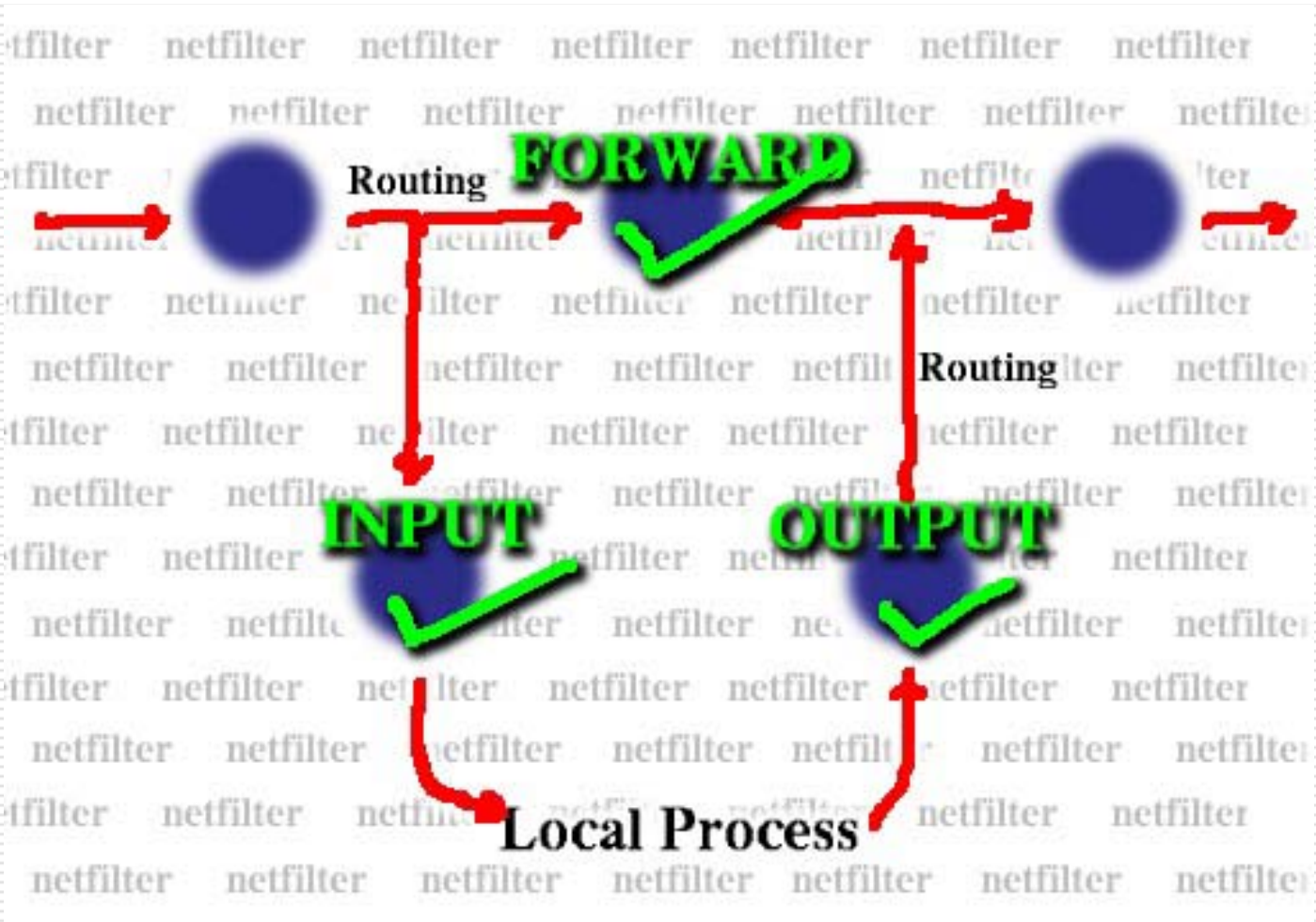


What We Use It For

Currently there are three tables: **filter**, **nat**, **mangle**.

- **filter table** used by packet filtering system
  - hooks in at **LOCAL_IN** (INPUT), **FORWARD**, **LOCAL_OUT** (OUTPUT)
  - iptable_filter hooks in at those points and passes all packets to the table
  - default table operated on by iptables program

The Hooks of filter

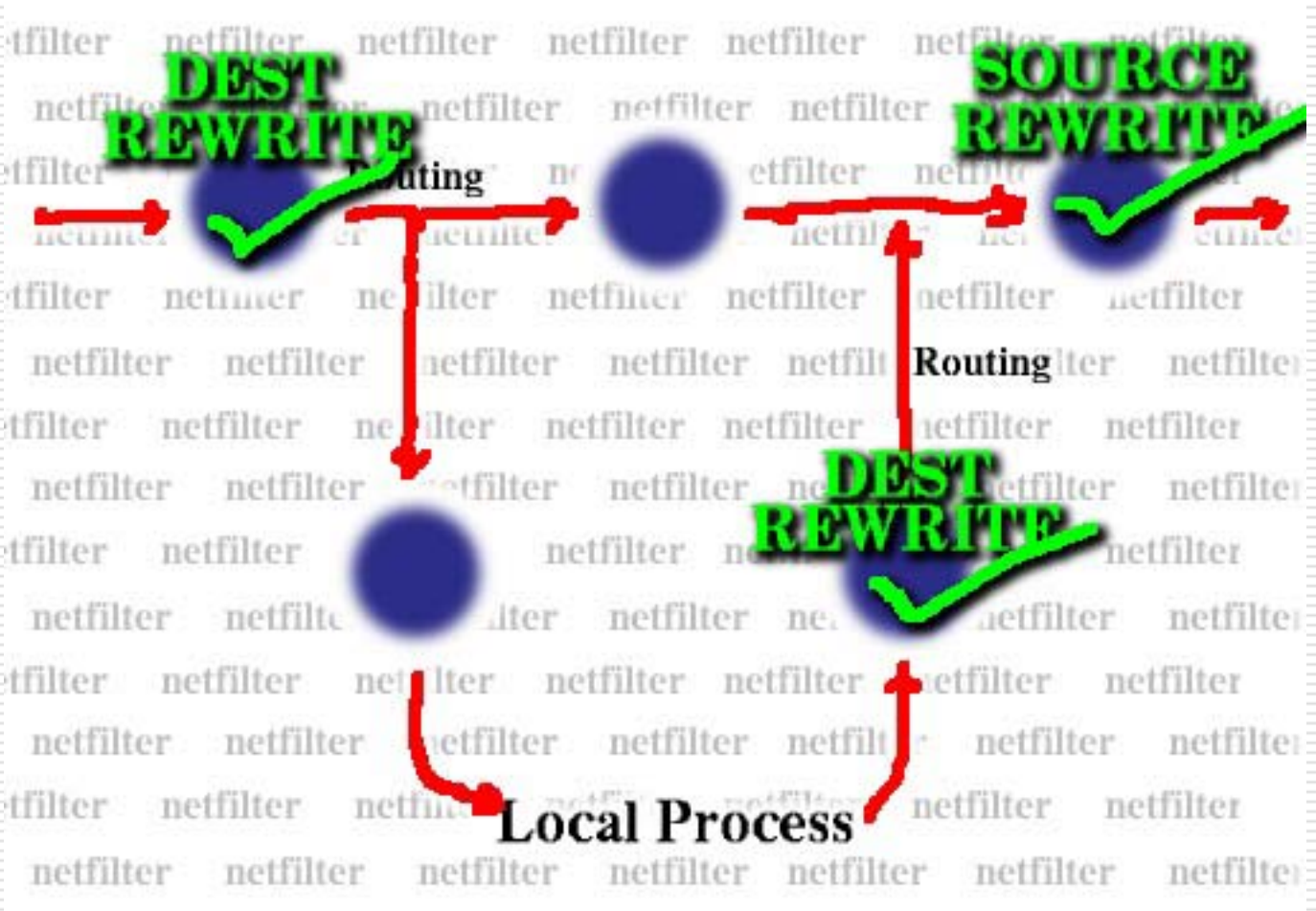


The nat Table

nat table used to control nat

- hooks in at **LOCAL_OUT** (OUTPUT), **PREROUTING**, **POSTROUTING**
- iptable_nat hooks in and passes packets whose connections have not seen NAT table to the table

The Hooks of nat



The mangle Table

- **mangle table** used for special effects
  - hooks in at **LOCAL_OUT** (OUTPUT), **PREROUTING**
  - iptable_mangle hooks in and passes all packets to the table

Iptables syntax - The basic iptables syntax

iptables [command] [options] <matches> <target>

- Commands:
  - append, insert, replace, delete, list, policy, etc.
- Options:
  - verbose, line numbers, exact, etc.
- Matches:
  - dport, dst, sport, src, states, TCP options, owner, etc.
- Targets:
  - ACCEPT, DROP, REJECT, SNAT, DNAT, TOS, LOG, etc.

Iptables syntax - A few matches

Protocol

-p, --protocol [!] [protocol]

- tcp, udp, icmp or all
- Numeric value
- /etc/protocols

Destination IP & Port

-d, --destination [!] address[/mask]

- Destination address
- Resolvable (/etc/resolve.conf)

--dport, --destination-port [!] port[:port]

- Destination port
- Numeric or resolvable (/etc/services)
- Port range

Iptables syntax - A few matches (cont.)

Source IP & Port

-s, --source [!] address[/mask]

- Source address
- Resolvable (/etc/resolve.conf)

--sport, --source-port [!] port[:port]

- Source port
- Numeric or resolvable (/etc/services)
- Port range

Iptables syntax - A few matches (cont.)

Incoming and Outgoing interface

- -i, --in-interface [!] interface
- -o, --out-interface [!] interface

Iptables syntax - Some targets

- ACCEPT
  - Accepts the packet
  - Ends further processing of the specific chain
  - Ends processing of all previous chains
  - Except other main chains and tables
- DROP
  - Drops the packet
  - No reply
  - Ends all further processing

Iptables syntax - Some targets (cont.)

- REJECT
  - Drops packet
  - Returns a reply
    - User specified reply
    - Calculated reply
    - TCP-RST or ICMP errors
  - Ends all further processing
- RETURN
  - Returns from a chain to the calling chain

Iptables syntax - ... and a few simple rules

- iptables -A INPUT -p tcp -m state --state NEW ! --syn -j REJECT --reject-with-tcp-reset
- iptables -A INPUT -p tcp --dport 80:1024 -j DROP
- iptables -A FORWARD -p tcp --dport 22:113 -j DROP
- iptables -A FORWARD -p tcp --dport ftp-data:ftp -j DROP
- iptables -A OUTPUT -p tcp -o eth0 -j ACCEPT
- iptables -A OUTPUT -p tcp -o lo -j ACCEPT
- iptables -P OUTPUT DROP

Iptables syntax

- Listing the rules
  - -L, --list [chain]
- -F, --flush [chain]
  - Flushes (erases) all rules in a chain
  - Or a table
- -N, --new chain
  - Creates a user-specified chain
  - There must be no target with that name previously
- -X, --delete-chain [chain]
  - Deletes a user-created chain
  - No rules may reference the chain
  - Can delete all user-created chains in a table

Iptables syntax - Creating & Deleting user-created chains

Creating...

- iptables -t filter -N badtcppackets

and Deleting a chain

- iptables -t filter -X badtcppackets

and Deleting all user-created chains

- iptables -t filter -X

A simple example ruleset – The Goals

- The firewall
  - Will act as its own firewall
  - Incoming:
    - ICMP Echo request & reply
    - Identd requests
    - HTTP requests
  - Outgoing:
    - Everything generated by the host
    - Except "nonet" group
- And a LAN
  - From Internet to LAN
    - Related traffic
    - Established traffic
  - From LAN to Internet
    - Everything

A simple example ruleset - The technical details

- Firewall
  - LAN on eth0
  - LAN IP 192.168.1.1
  - Internet on eth1
  - Internet IP 10.0.0.1/32
- LAN
  - IP range 192.168.1.0/24

A simple example ruleset - The POSTROUTING chain

- We need SNAT to let our LAN out on the Internet. Without this, the Internet don't know where to route the packets
  - iptables -t nat -A POSTROUTING -i eth0 -o eth1 -j SNAT --to-source 10.0.0.1

A simple example ruleset - The INPUT chain

- Need to allow all incoming traffic specified in goals
- Need to allow return traffic for everything we send
- Default to DROP

```
iptables -P INPUT DROP
iptables -A INPUT -p tcp --dport 113 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -m state --state
    ESTABLISHED,RELATED -j ACCEPT
```

A simple example ruleset - The OUTPUT chain

- Accept everything except the nonet group to leave
  - iptables -A OUTPUT -m owner --gid-owner nonet -j DROP

A simple example ruleset - The FORWARD chain

- Everything from LAN to Internet
- ICMP replies, related and Established traffic from Internet to LAN
  - iptables -P FORWARD DROP
  - iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
  - iptables -A FORWARD -i eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT

End of the Tutorial

On Top of Netfilter

- Currently, four major subsystems exist on top of netfilter:
  - The backwards-compatibility ipchains & ipfwadm +masq/redir modules.
  - The `iptables' packet classification system.
  - The connection-tracking system.
  - The NAT system.

iptables

- What It Is
  - Kernel: Lists of packet matching rules similar to ipchains/ipfwadm
  - Userspace: program `iptables' and library `libiptc' which access tables
  - Simple functionality (IP header matching) built in
  - Supports multiple tables