CT3531

Packet Filtering and Firewalls

What is a packet filter

- A piece of software which looks at the header of packets as they pass through and decides its fate
  - DROP
  - ACCEPT
  - Or something more complicated.
- Under Linux, packet filtering is built into the kernel.

Functions of Packet Filter

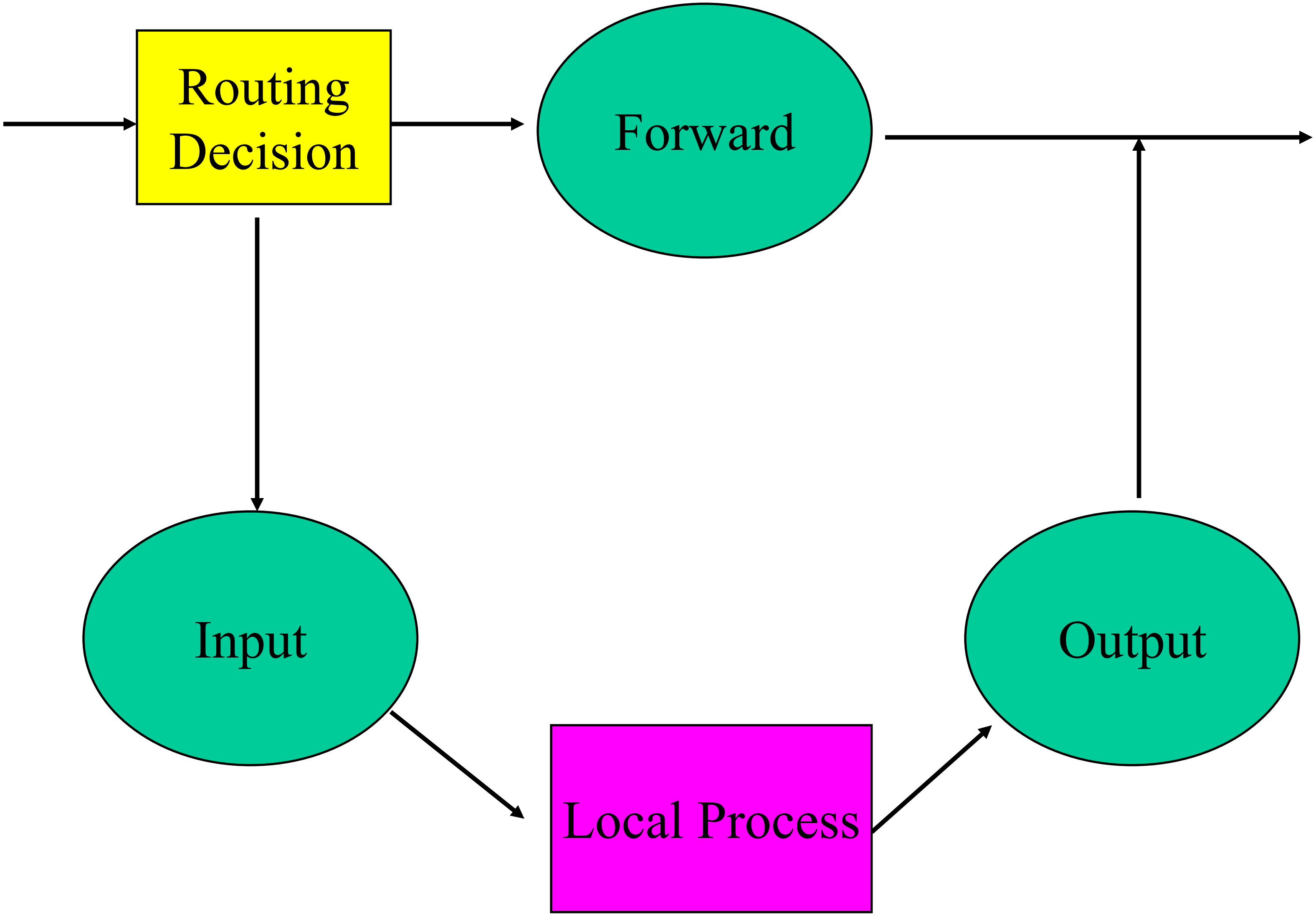
- Control
  - Allow only those packets that you are interested to pass through.
- Security
  - Reject packets from malicious outsiders
    - Ping of death
    - telnet from outside
- Watchfulness
  - Log packets to/from outside world

Packet Filter under Linux

- 1st generation
  - ipfw (from BSD)
- 2nd generation
  - ipfwadm (Linux 2.0)
- 3rd generation
  - ipchains (Linux 2.2)
- 4th generation
  - iptables (Linux 2.4)
  - In this lecture, we will concentrate on iptables

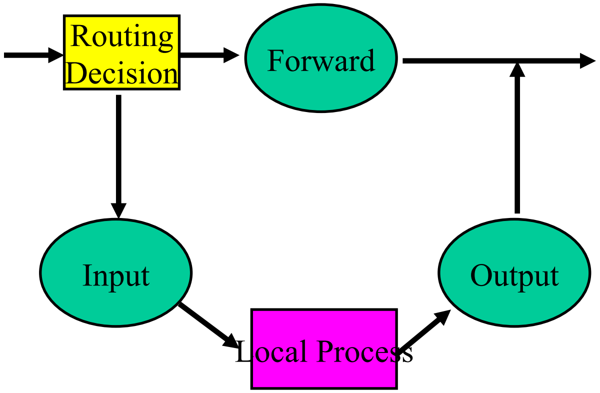
iptables

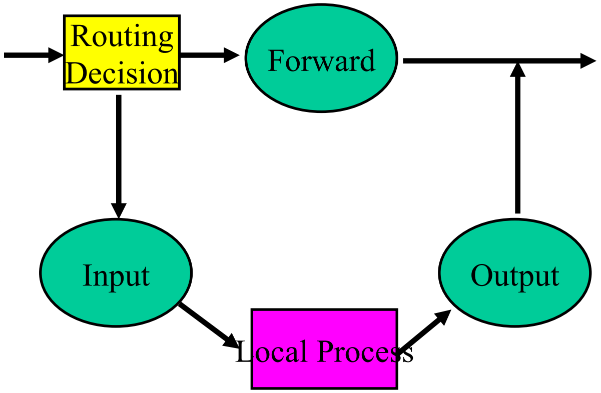
- Kernel starts with three lists of rules called (firewall) chains.
  - INPUT
  - OUTPUT
  - FORWARD
- Each rule say "if the packet header looks like this, then here's what to do".
- If the rule doesn't match the packet, then the next packet will be consulted.


Routing Decision
Forward
Input
Output
Local Process

1. When a packet comes in, the kernel first looks at the destination of the packet: this is called routing.
2. If it’s destined for this box
   - Passes downwards in the diagram
   - To INPUT chain
     - If it passes, any processes waiting for that packet will receive it.

   Otherwise go to step 3





3. If forwarding is not enabled
   - The packet will be dropped

   If forwarding is enable and the packet is destined for another network interface.
   - The packet goes rightwards on our diagram to the FORWARD chain.
     - If it is accepted, it will be sent out.
4. Packets generated from local process pass to the OUPUT chain immediately.
   - If its says accept, the packet will be sent out.

Usage

SYNOPSIS

```
iptables -[ADC] chain rule-specification [options]
iptables -[RI] chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LFZ] [chain] [options]
iptables -[NX] chain
iptables -P chain target [options]
iptables -E old-chain-name new-chain-name
```

-N	Create a new chain
-X	Delete an empty chain
-P	Change the policy for a built-in chain
-L	List the rules in a chain
-F	Flush the rules out of a chain
-Z	Zero the packet and byte counters on all rules in a chain

Operations to manage whole chains

-A	Append a new rule to a chain
-I	Insert a new rule at some position in a chain
-R	Replace a rule at some position in a chain
-D	Delete a rule at some position in a chain
-D	Delete the first rule that matches in a chain

Manipulate rules inside a chain

- A simple experiment
  - – Drop all ICMP packets coming from the IP address 127.0.0.1

```
# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.2 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
# iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
#
```

Filtering Specifications

- Specifying Source and Destination IP address
  - – Source
    - -s, --source or –src
  - – Destination
    - -d, --destination or –dst
  - – IP address can be specified in four ways.
    - Full name (e.g. www.cse.cuhk.edu.hk)
    - IP address (e.g. 127.0.0.1)

- Group specification (e.g. 199.95.207.0/24)
- Group specification
  (e.g. 199.95.207.0/255.255.255.0)

- Specifying Inversion
  - Many flags, including the ‘–s’ and ‘–d’ flags can have their arguments preceded by ‘!’ (not).
  - Match address NOT equal to the ones given.
  - E.g. ‘-s ! localhost’ matches any packet not coming from localhost.

- Specifying an Interface
  - Physical device for packets to come in
    - -i, --in-interface
  - Physical device for packets to go out
    - -o, --out-interface
  - Packets traversing the INPUT chain don’t have an output interface
    - Rule using ‘-o’ in this chain will never match.
  - Packets traversing the OUPUT chain don’t have an input interface
    - Rule using ‘-i’ in this chain will never match.

- Specifying Protocol
  - The protocol can be specified with the ‘-p’ flag.
  - Protocol can be a number if you know the numeric protocol values for IP.
  - Protocol can be a name for special cases of
    - TCP
    - UDP
    - ICMP
  - Case insensitive (e.g. tcp works as well as TCP)
  - Can be prefixed by a ‘!’, e.g. ‘–p ! TCP’

- Specifying Fragments
  - Sometimes a packet is too large
    - Divided into fragments
    - Sent as multiple packets.
  - IP header contains in the first segment only.
  - Impossible to look inside the packet for protocol headers such as TCP, UDP, ICMP.
  - This means that the first fragment is treated like any other packet. Second and further fragments won’t be.

- E.g ‘-p TCP -sport www’ (specifying a source port of ‘www’), will never match a fragment other than the first fragment.
- You can specify a rule specifically for second and further fragments, using the ‘-f’

  (or –fragment) flag.
- E.g. The following rule will drop any fragments going to 192.168.1.1
- # iptables -A OUTPUT -f -d 192.168.1.1 -j DROP

- TCP extensions
  - Automatically loaded if ‘--protocol tcp’ is specified.
  - --tcp-flags
    - Allows you to filter on specific TCP flags.
    - The first string of flags is the mask
    - The second string of flags tells which one(s) should be set.
    - E.g.

```
# iptables  -A INPUT –protocol tcp –tcp-flags ALL SYN,ACK –j DROP
```

    - Indicates that all flags should be examined
    - ALL is synonymous with 'SYN,ACK,FIN,RST,URG,PSH'
    - But only SYN and ACK should be set.
    - There is also an argument 'NONE' meaning no flags.
  - --syn
    - Optionally preceded by a '!'.
    - Shorthand for --tcp-flags SYN,RST,ACK SYN'.
  - --source-port
    - Single port or range of ports
    - Can be specified by names listed in /etc/services

- --sport
  - Synonymous with ‘--source-port’.
- --destination-port or --dport
  - Specify the destination port.
- --tcp-option
  - Followed by an optional ‘!’ and a number.
  - Matches a packet with a TCP option equaling that number.
- E.g.

```
-p TCP –s 192.168.1.1 --syn
```

- Specify TCP connection attempts from 192.168.1.1

- UDP Extensions
  - Loaded if ‘--protocol udp’ is specified.
  - Provides the following options
    - --source-port
    - --sport
    - --destination-port
    - --dport
- ICMP Extensions
  - Loaded if ‘--protocol icmp’ is specified.
  - --icmp-type
    - Specify ICMP type (numeric type or name)

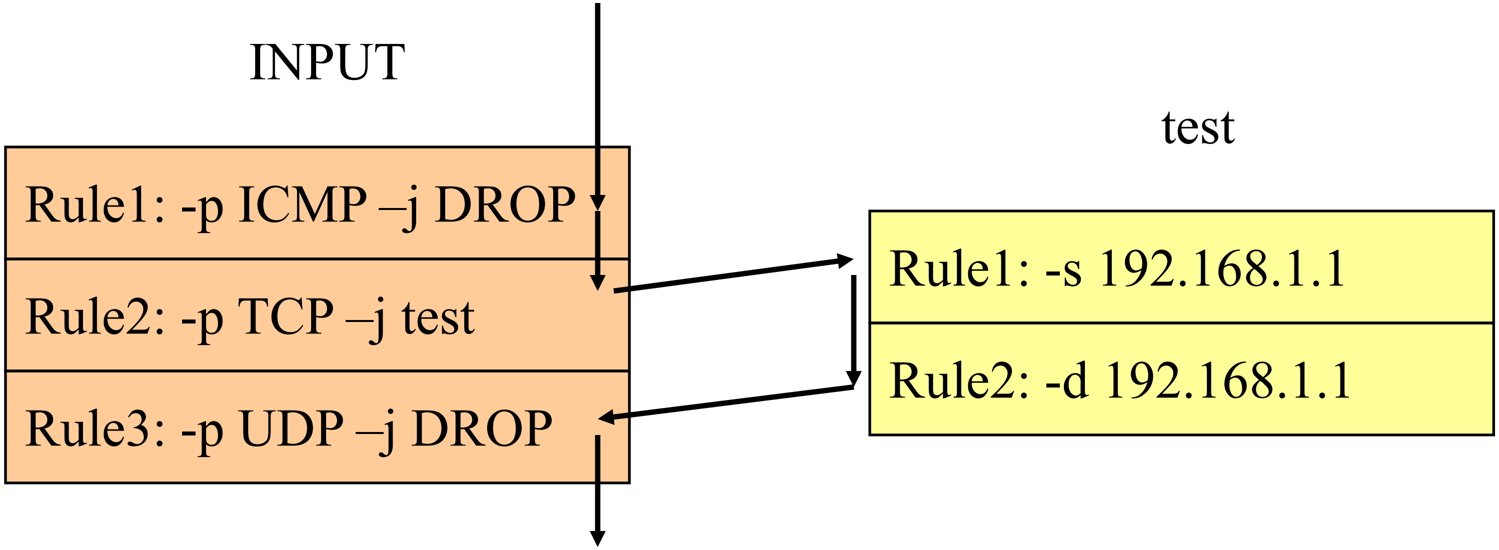
- Other Match Extension
  - Invoked with the ‘-m’ option.
  - Mac
    - Specified with ‘-m mac’ or –match mac’
    - Used for matching incoming packet's source Ethernet address. (MAC).
    - Only one option ‘--mac-source’
    - E.g. –mac-source 00:60:08:91:CC:B7
  - Limit
    - Specified with ‘-m limit’ or --match limit’.
    - Restrict the rate of matches, such as for suppressing log messages.

- Two options
  - --limit
    - Followed by a number
    - Specifies the maximum average number of matches to allow per second.
    - Can specify other unit such as ‘/second’, ‘/minute’, ‘/hour’, or ‘/day’.
    - E.g. --limit 5/second or --limit 5/s
  - --limit-burst
    - Followed by a number.
    - The maximum initial number of packets to match.
    - This number gets recharged by one every time the limit specified above is not reached.
    - Often used with the LOG target.
  - Default 3 matches per hour, with a burst of 5
  - E.g. iptables –A FORWARD –m limit –j LOG

- State Match
  - Specifying ‘-m state’ allows an additional ‘--state’ option.
  - NEW
    - A packet which creates a new connection.
  - ESTABLISHED
    - A packet which belongs to an existing connection
  - RELATED
    - A packet which is related to, but not part of, an existing connection such as ICMP error.
  - INVALID
    - A packet which could not be identified for some reasons.

- Target Specifications
  - Two built-in targets
    - DROP
    - ACCEPT
  - Extensions
    - LOG
      - --log-level
        - » Specify the level of log 0 to 7.
      - --log-prefix
        - » Followed by a string up to 14 chars
        - » Sent at the start of the log
    - REJECT
      - DROP + send an ICMP port unreachable error message

- User-defined chains
  - User can create new chains.
  - By convention, user-defined chains are lowercase.
  - Packet matches rule whose target is a user-defined chain, the packet begins traversing the rules in that user-defined chain.
  - If that chain doesn’t decide the fate of the packet, then once traversal of that chain has finished, traversal resumes on the next rule on the current chain.



INPUT

Rule1: -p ICMP –j DROP

Rule2: -p TCP –j test

Rule3: -p UDP –j DROP

test

Rule1: -s 192.168.1.1

Rule2: -d 192.168.1.1

User-defined chains can jump to other user-defined chains. Your packets will be dropped if they are found to be in a Loop.

Network Address Translation

- We are not going to cover NAT in detail
- Here's an example here for redirecting ICMP echo request.

```
iptables -A PREROUTING -t nat -p icmp -d 137.189.89.176 \
  -j DNAT --to 137.189.89.178
```

- Iptables can also achieve masquerading for internet sharing, port forwarding etc.

Firewall

- A firewall is a computer system dedicated to ‘isolate’ a LAN from the Internet.
- It is at the entry point of the LAN it protects.
- It inspects and makes decisions about all incoming packets before it reaches other parts of the system.
- Outgoing traffic may also be inspected and or blocked.

- A firewall can be a simple packet filter.
- It can also be an enormously complex computer system with
  - extensive logging systems,
  - intrusion detection systems.
- Nowadays, it is easy to download a free firewall that can be run on your Linux or Unix system.

- These firewalls usually provide a user interface to specify rules to
  - Block particular incoming connections from systems outside your LAN.
  - Block all connections to or from certain systems you distrust.
  - Allow email and ftp services, but block dangerous services like TFTP, RPC, rlogin etc.
  - Block certain type of packets.