# Firewall Tutorial

KAIST
Dept. of EECS
NC Lab.

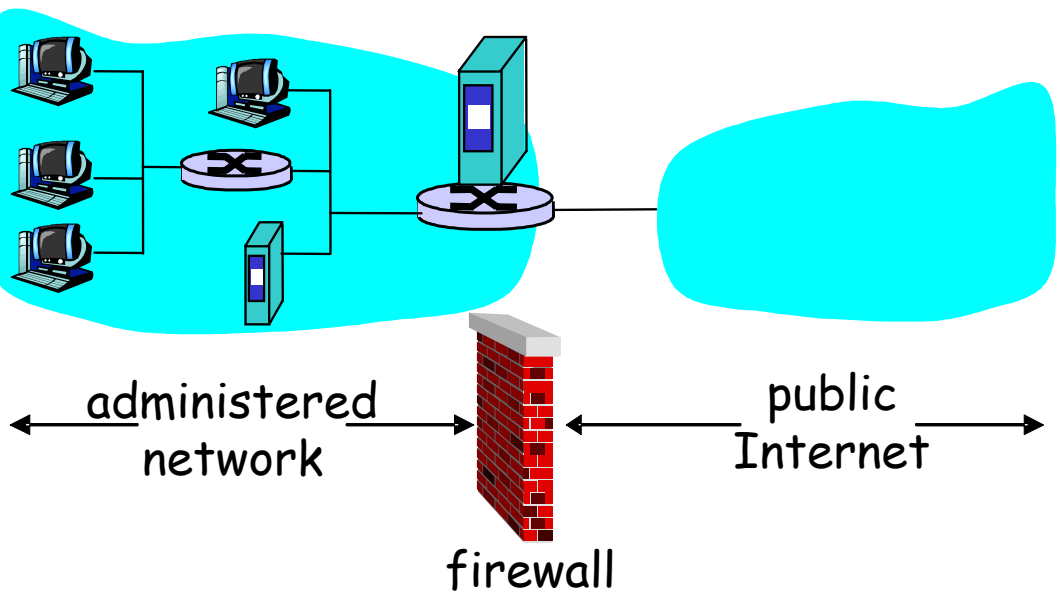
# Contents

- What is Firewalls?
- Why Firewalls?
- Types of Firewalls
- Limitations of firewalls and gateways
- Firewalls in Linux

# What is Firewalls?

**firewall**

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others.

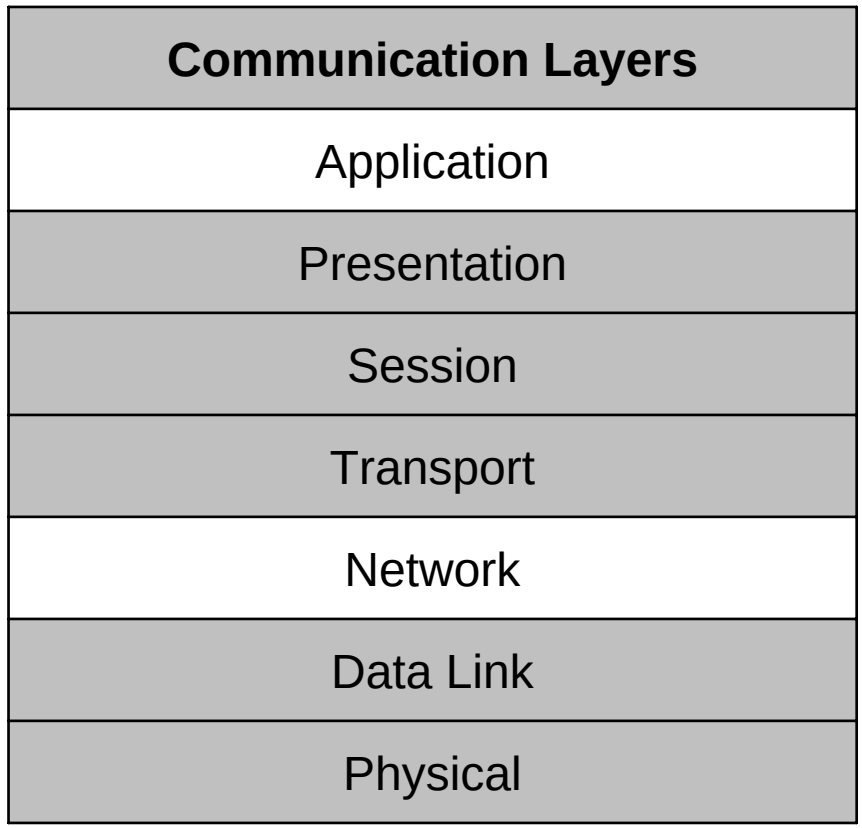


# Why Firewalls?

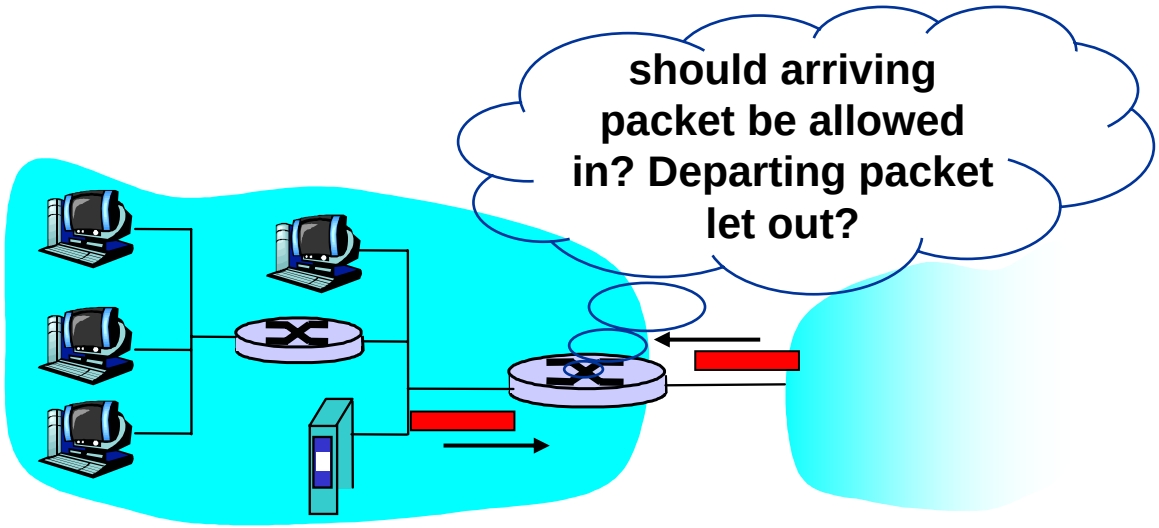
- prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections.
- prevent illegal modification/access of internal data.
  - e.g., attacker replaces CIA's homepage with something else
- allow only authorized access to inside network (set of authenticated users/hosts)

# Types of Firewalls

- packet-filtering firewall
  - At the network layer
- Application-level gateway
  - At the application layer



# Network layer: Packet Filtering




- internal network connected to Internet via router firewall
- router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

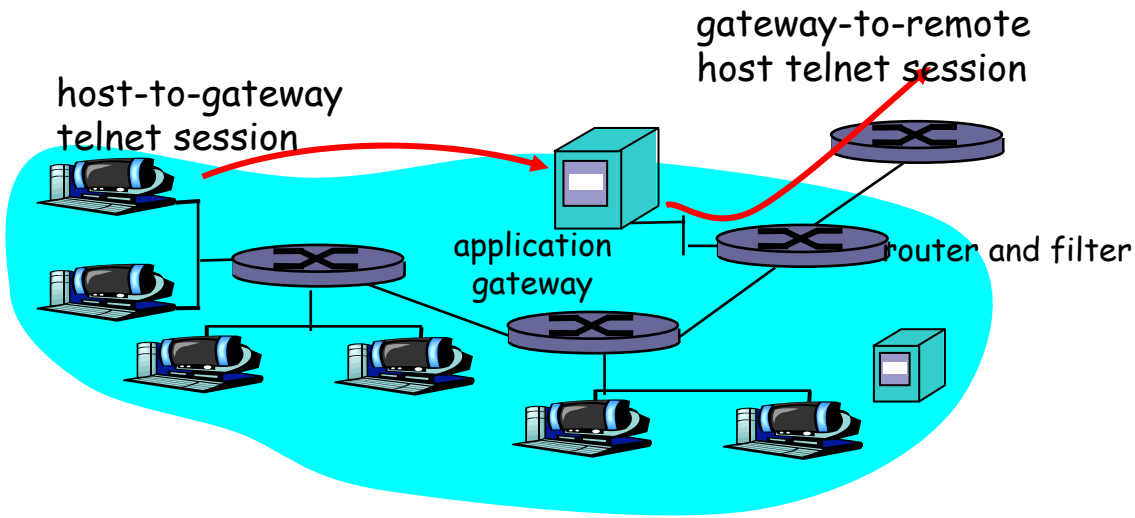
# Packet Filtering

- Example 1: block incoming and outgoing datagrams with **IP protocol field = 17** and with either **source or dest port = 23**.
  - All incoming and outgoing UDP flows and telnet connections are blocked.
- Example 2: Block inbound TCP segments with **ACK=0.**
  - Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

# Application layer: Application Gateways

- Example
  - allow select internal users to telnet outside.
  - Users authenticate themselves to create telnet connection

# Application Gateways




- Solution
  - Router filter blocks all telnet connections not originating from gateway.
  - For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections

# Limitations of Firewalls and Gateways

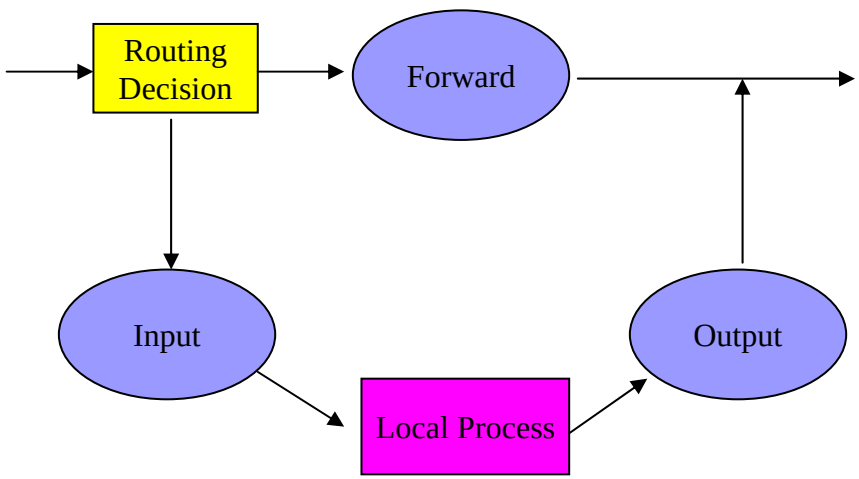
- IP spoofing
  - router can't know if data "really" comes from claimed source
- If multiple app's. need special treatment, each has own app. gateway.
- client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- Tradeoff
  - degree of communication with outside world, level of security
  - Performance problem

# Firewalls in Linux

- Before kernel 2.2 : ***ipfwadm***
- kernel 2.2.x : ***ipchains***
- After kernel 2.3.15 : ***netfilter***
  - *netfilter* module in linux can handle packet flow
  - New alternative command *iptables*
  - Backward compatible for *ipfwadm* and *ipchains*

# Rules

- There are three types of built-in chains (or lis ts of rules):
  - INPUT – destined for the local system
  - OUTPUT – originate from the local system
  - FORWARD – enter the system and is forwarde d to another destination



# Operations (1/3)

- There are mainly three types of operations:
  - ACCEPT – accept the packet
  - DROP – discard the packet silently
  - REJECT – actively reply the source that the packet is rejected.
- All the rules are consulted until the first rul e matching the packet is located.
- If no rules match the packet, the kernel loo ks at the chain policy.

# Operations (2/3)

- Operations to manage whole chains
  - N: create a new chain
  - P: change the policy of built-in chain
  - L: list the rules in a chain
  - F: flush the rules out of a chain
- Manipulate rules inside a chain
  - 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 in a chain

# Operations (3/3)

- Some filtering specifications:
  - j: specify the rule target
  - s: specify the source addresses
  - d: specify the destination addresses
  - p: specify the protocol used (e.g. tcp, udp, icmp)
  - i: specify the input interface
  - o: specify the output interface
  - !: specify the inversion (i.e. NOT)

# Extension of iptable

- TCP Extensions:
  - --tcp-flags: filter on specific flags
  - --syn: shorthand of --tcp-flags SYN, RST, ACK SYN
  - --source-port (or --sport): specify the source port
  - --destination port (or --dport): specify the destination port
- UDP Extensions:
  - --sport and --dport

# Examples

- Drop all icmp (such as ping) packets
  - iptables –A INPUT –p icmp –j DROP
- Flush all chains
  - iptables –F
- List all existing rules
  - iptables –L
- Accept the ssh service from eureka machines
  - iptables –A INPUT –p tcp –s 143.248.37.197 –d 0/0 --dport 23 –j ACCEPT

# Examples

- Reject all incoming TCP traffic destined for ports 0 to 10 23
  - iptables –A INPUT –p tcp –s 0/0 –d 0/0 –dport 0:1023 –j REJ ECT
- Reject all outgoing TCP traffic except the one destined fo r 137.189.96.142
  - iptables –A OUTPUT –p tcp –s 0/0 –d ! 137.189.96.142 –j RE JECT
- Drop all SYN packets from pc89184
  - Iptables –A INPUT –p TCP –s 137.189.89.184 --syn –j DROP

- References
  - Linux iptables HOWTO, by Rusty Russell
    - http://www.linuxguruz.org/iptables/howto/iptables-HOWTO.html