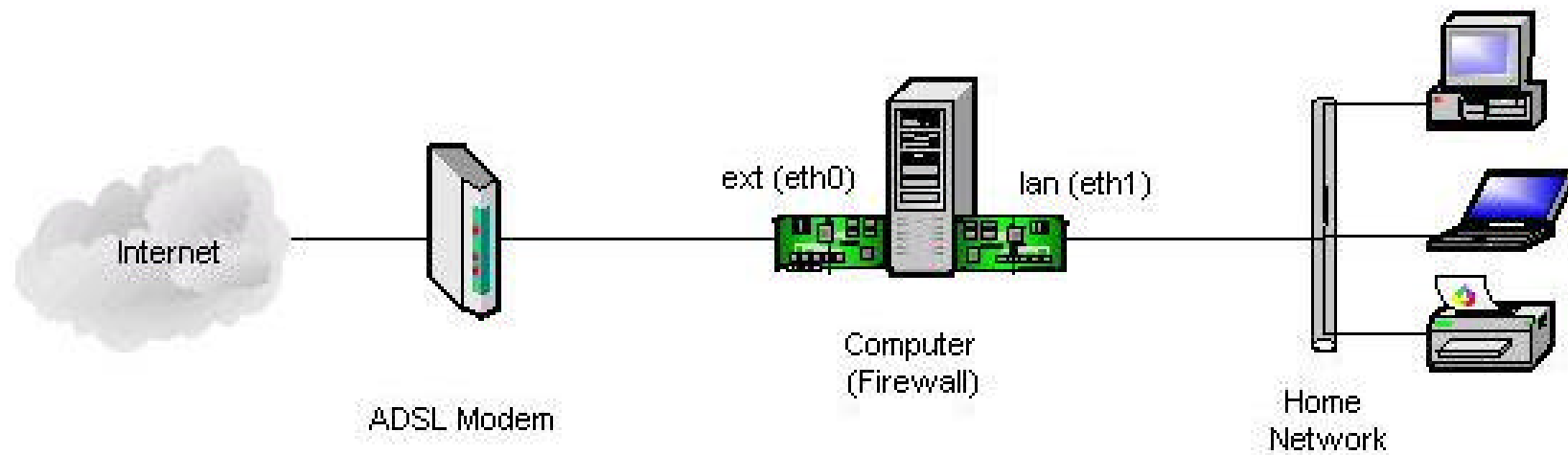


# Firewalls

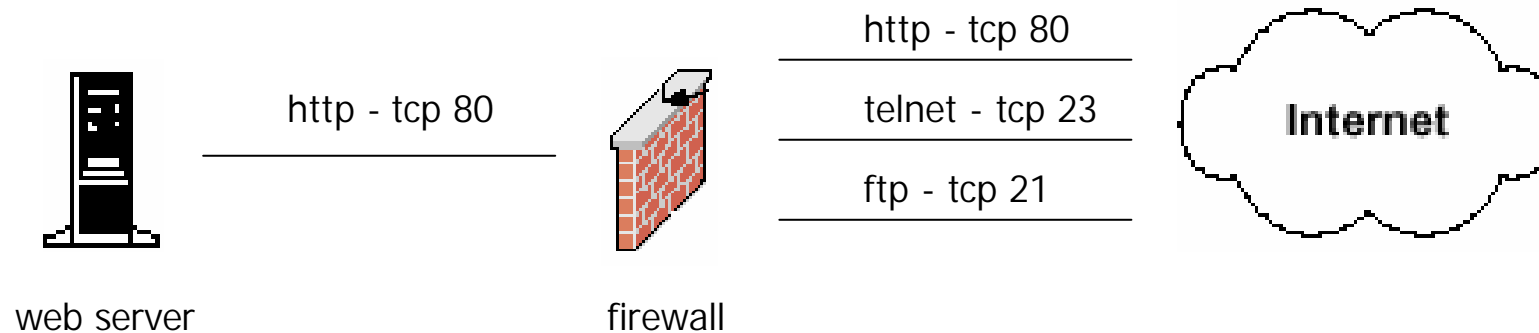
- Many people might think that a firewall is a single device on your network configured to protect your internal network from the external world
- A firewall is a system (or a group of systems) that enforces an access control policy between two networks
- Disallow unauthorized and/or malicious traffic from traveling on your network – in both directions
- Firewalls can't protect you from attacks that don't go through it
- If there's another entry point to your network not protected by a firewall, then your network isn't secured
- Firewalls do not verify the content of the traffic through it

# A typical firewall setup



# Packet filtering firewalls

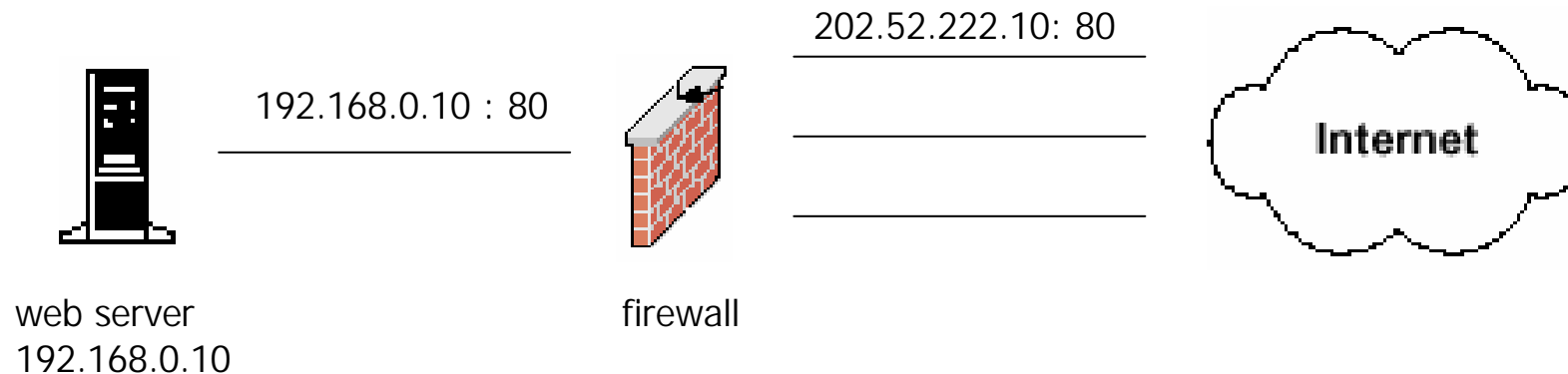
- examines the source and destination address of the data packet and either allows or denies the packet from traveling the network
- blocks access through the firewall to any packets, which try to access ports which have been declared "off-limits"



- Allow only http - tcp 80
- Drop ip any

# Application layer firewalls

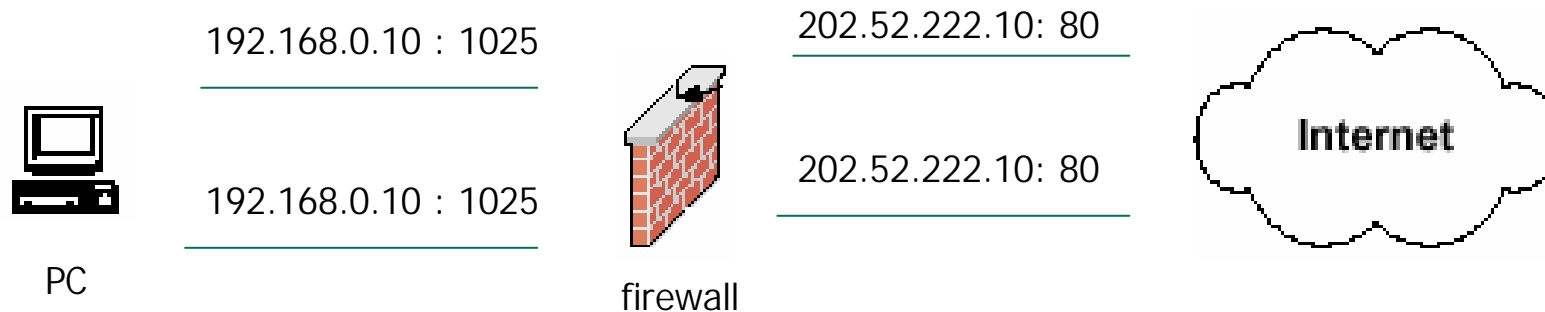
- Also known proxy firewalls, application gateway
- attempts to hide the configuration of the network behind the firewall by acting on behalf of that network/servers
- All requests for access are translated at the firewall so that all packets are sent to and from the firewall, rather than from the hosts behind the firewall



- Translates 202.52.222.10 : 80 to 192.168.0.10 : 80

# Stateful inspection firewalls

- Examines the state and the context of the packets
- Remembers what outgoing requests have been sent and only allow responses to those requests back through the firewall
- Attempts to access the internal network that have not been requested by the internal network will be denied



- Only allows reply packets for requests made out
- Blocks other unregistered traffic

# Firewall Best Practices

- Explicitly deny all traffic except for what you want
- The default policy should be that if the firewall doesn't know what to do with the packet, deny/drop it
- Don't rely only on your firewall for the protection of your network - remember that it's only a device, and devices do fail
- Make sure you implement what's called "defense in depth." - multiple layers of network protection
- Make sure all of the network traffic passes through the firewall
- If the firewall becomes disabled, then disable all communication

# Firewall Best Practices

- If there's another way in to the network (like a modem pool or a maintenance network connection), then this connection could be used to enter the network completely bypassing the firewall protection
- Disable or uninstall any unnecessary services and software on the firewall - limit the number of applications
- Use firewalls internally to segment networks between different departments and permit access control based upon business needs

# Firewall products

- Iptables [www.iptables.org](http://www.iptables.org)
- Cisco PIX [www.cisco.com](http://www.cisco.com)
- Checkpoint [www.checkpoint.com](http://www.checkpoint.com)
- Border Manager [www.novell.com](http://www.novell.com)
- Netscreen [www.netscreen.com](http://www.netscreen.com)
- Winroute [www.winroute.com](http://www.winroute.com)



# IPTables

- Features:
  - Linux kernel contains advanced tools for packet filtering
  - the framework inside the Linux 2.4.x kernel
  - re-designed and heavily improved successor of the previous 2.2.x ipchains and 2.0.x ipfwadm systems
  - Provides functionality of packet filtering (stateless or stateful)
  - All kinds of network address translation (NAT)
  - Packet mangling (manipulation)
  - flexible and extensible infrastructure
  - Large number of additional features as modules / patches
  - generic table structure for the definition of rulesets
  - consists of classifiers (matches) and one connected action (target)

# What all can I do with iptables ?

- Build internet firewalls based on stateless and stateful packet filtering
- Use NAT and masquerading for sharing internet access where you don't have enough addresses
- Use NAT for implementing transparent proxies
- Aid the tc+iproute2 system used to build sophisticated QoS routers
- Do further packet manipulation (mangling) like altering the TOS field of the IP header

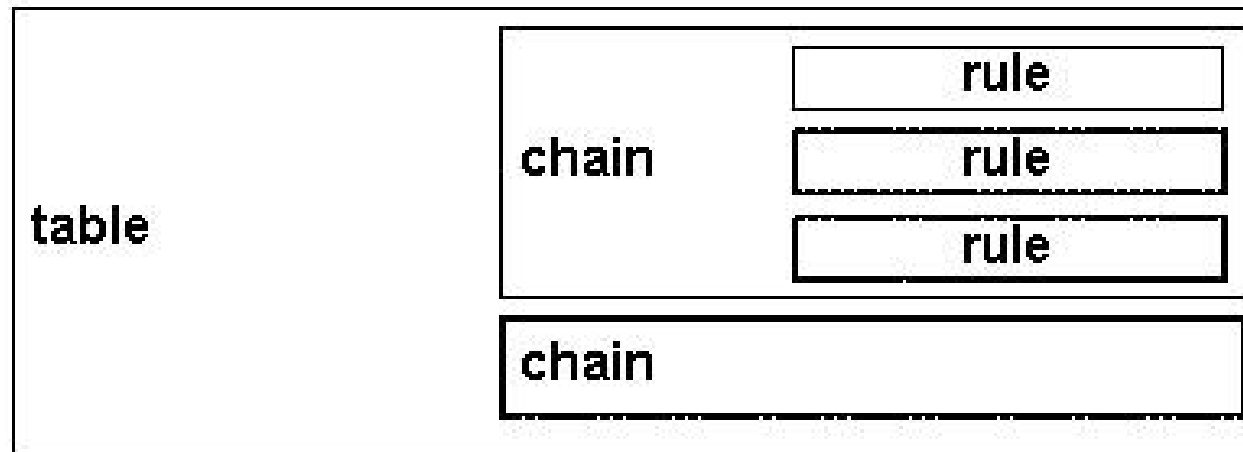
# So What's A Packet Filter?

- the process of controlling network packets as they attempt to enter, move through, and exit your system
- looks at the *header* of packets as they pass through, and decides the fate of the entire packet
  - **DROP** the packet - discard the packet as if it had never received it
  - **ACCEPT** the packet - let the packet go through
  - **LOG** the packet – just log the information for monitoring purpose
- or something more complicated!
- Under Linux, packet filtering is built into the kernel

# Why Would I Want to Packet Filter?

- Control – allow certain types of traffic, and disallow others
- Security – prevent unauthorized access or attacks to/from your network
- Watchfulness - monitor abnormal / suspicious activity to/from your network

# IPTables components



- Rule – operation to be performed on a packet
  - Chains – collection of rules
  - Table – collection of chains
- 
- Iptables – userspace command for configuring the system
  - Modules – kernel modules for diff. features / tasks

# IPTables “tables”

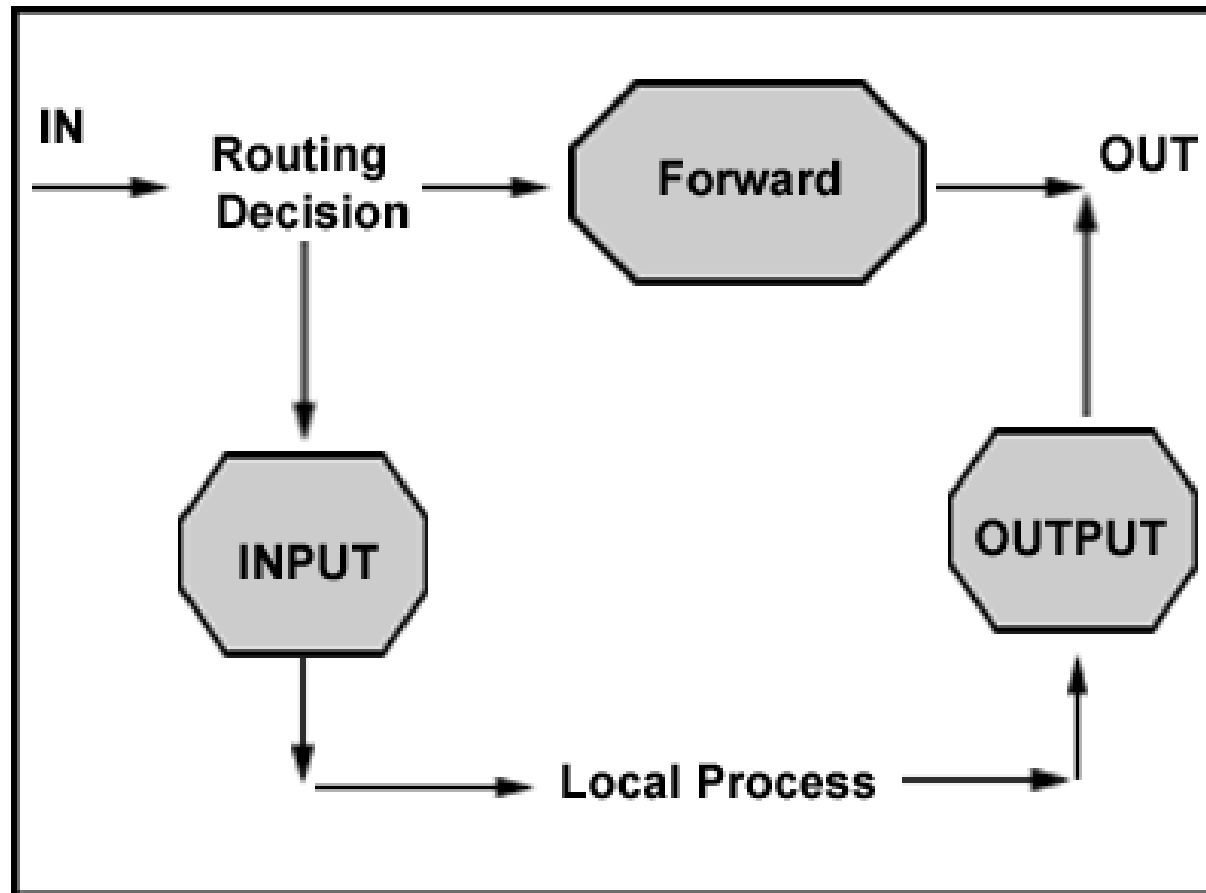
IPTables has three tables:

- Filter – performs packet filtering
- NAT – performs address translation between hosts on internal network and external addresses on the internet
- Mangle - modifies contents of specific packet header fields

## Filter table “chains”

- INPUT – packets destined for a local interface
- FORWARD – routable packets to another network
- OUTPUT – packets originating from a local interface

How packets traverse the filter table chains?





# IPTables configuration

To manage chains in a table:

- Creating a new chain -N
- Delete an empty chain -X
- Flush all rules in a chain -F
- Change the default policy of a chain -P
- List all the rules in a chain -L

To manage rules in a chain:

- Append a rule in a chain -A
- Insert a rule at some position -I
- Replace a rule at some position -R
- Delete a rule -D

# IPTables configuration

Possible targets for a rule in a filter table chain:

- |               |  |
|---------------|--|
| <b>ACCEPT</b> | – to accept & let the packet pass through            |
| <b>DROP</b>   | – to simply drop the packet w/o any error message    |
| <b>REJECT</b> | – to deny the packet w/ an error message (polite)    |
| <b>LOG</b>    | – to log info of the packet to syslog for monitoring |

# IPTables configuration

Default chain policies:

```
INPUT ACCEPT any any
```

```
FORWARD ACCEPT any any
```

```
OUTPUT ACCEPT any any
```

Change the default policy to DROP all packets:

```
iptables -P INPUT DROP
```

# IPTables configuration

```
iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
```

```
iptables -D INPUT 1
```

```
iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

```
iptables -A INPUT -s 0/0 -j DROP
```

```
iptables -A INPUT -j DROP
```

To get help:

```
iptables -h
```

```
iptables -p tcp -h
```

```
iptables -m state -h
```

```
iptables -j ACCEPT -h
```

# IPTables configuration

```
iptables -A INPUT -p tcp -dport 25 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp dport 25 -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -o eth1 -p tcp -dport  
25 -j ACCEPT
```

```
iptables -A FORWARD -s 202.52.225.0/24 -d  
202.52.255.1 -p tcp -dport 25 -j ACCEPT
```

```
iptables -A FORWARD -s 202.52.225.0/24 -d  
202.52.255.5 -p udp -dport 53 -j ACCEPT
```

```
iptables -A FORWARD -d 202.52.255.5 -p tcp -j LOG  
-log-prefix "TCP log "
```

# IPTables - Connection Tracking

- Stateful connection tracking of traversing packets

NEW - packets that create a NEW connection

ESTABLISHED - packets belonging to an existing connection

RELATED - packets related to an existing connection

INVALID - packets not corresponding to any existing connection

```
iptables -A FORWARD -d 202.52.225.0/24 -p tcp -m  
-state --state ESTABLISHED -j REJECT
```

```
iptables -A FORWARD -m -state --state INVALID -j  
DROP
```

# NAT with IPtables

- NAT is a standard that enables a network to use one set of IP addresses for moving data packets on the local area network and a second set of IP addresses for external traffic the Internet
- The firewall acts as the address translation device between addresses on the home side of the network and addresses on the internet side of the network
- NAT enables the user to shield address on the internal network from address on the Internet network

## NAT table “chains”

- PREROUTING – before routing decision is made and packet enters the system
- OUTPUT – packets leaving the system
- POSTROUTING – after routing decision is made and packet leaves the system



## NAT chain “targets”

- DNAT - mainly used in cases where you have a public IP and want to redirect accesses to the firewall to some other host
- SNAT - mainly used for changing the source address of packets
- MASQUERADE - used in exactly the same way as SNAT, but the MASQUERADE target automatically checks for the IP address to use, instead of doing as the SNAT target does - just using the single configured IP address.

# NAT chain rules

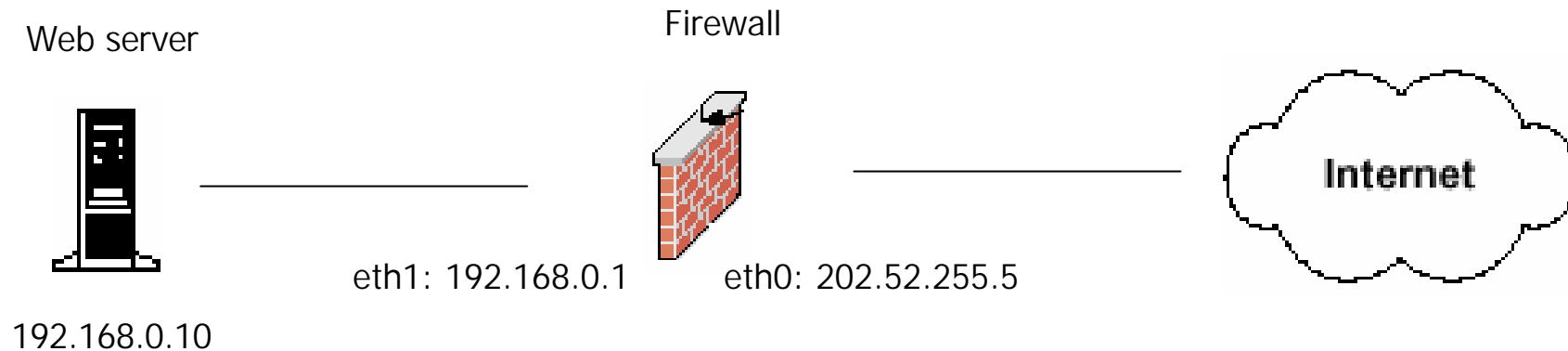
To NAT all outbound traffic with the IP of eth0:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Mapping the web port with squid:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp  
--dport 80 -j DNAT --to 202.52.202.52:3128
```

# NAT: Fixed IP mapping (inbound)

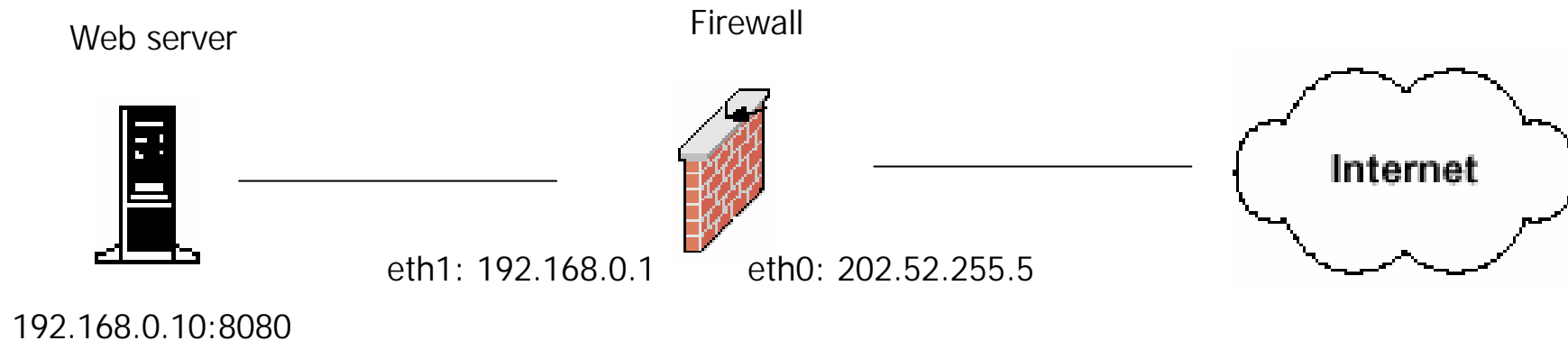


Makes it possible to hide internal server IP from the Internet

```
iptables -t -nat -A PREROUTING -i eth1 -d 202.52.255.5  
-j DNAT -to-destination 192.168.0.1
```

This rule maps the IP addresses in both the requests sent to the server and the server's reply

# NAT: Port mapping (inbound)

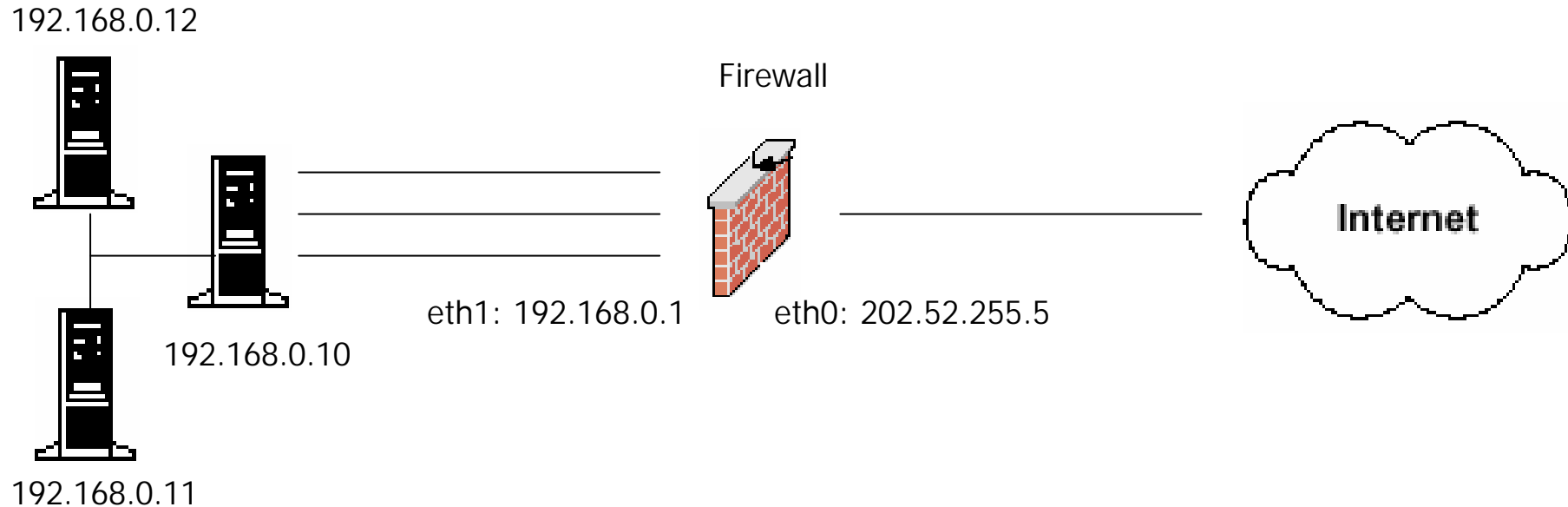


Entails the modification of the destination port and enables clients to access a service via destination port other than that on which service listens.

```
iptables -t -nat -A PREROUTING -i eth0 -d 202.52.255.5 -p tcp  
-m tcp -dport 80 -j DNAT --to-destination 192.168.0.1:8080
```

This rule maps port 80 of host with IP 202.52.255.5 to port 8080 of the internal host having IP 192.168.14.2

# NAT: IP Masquerading (outbound)

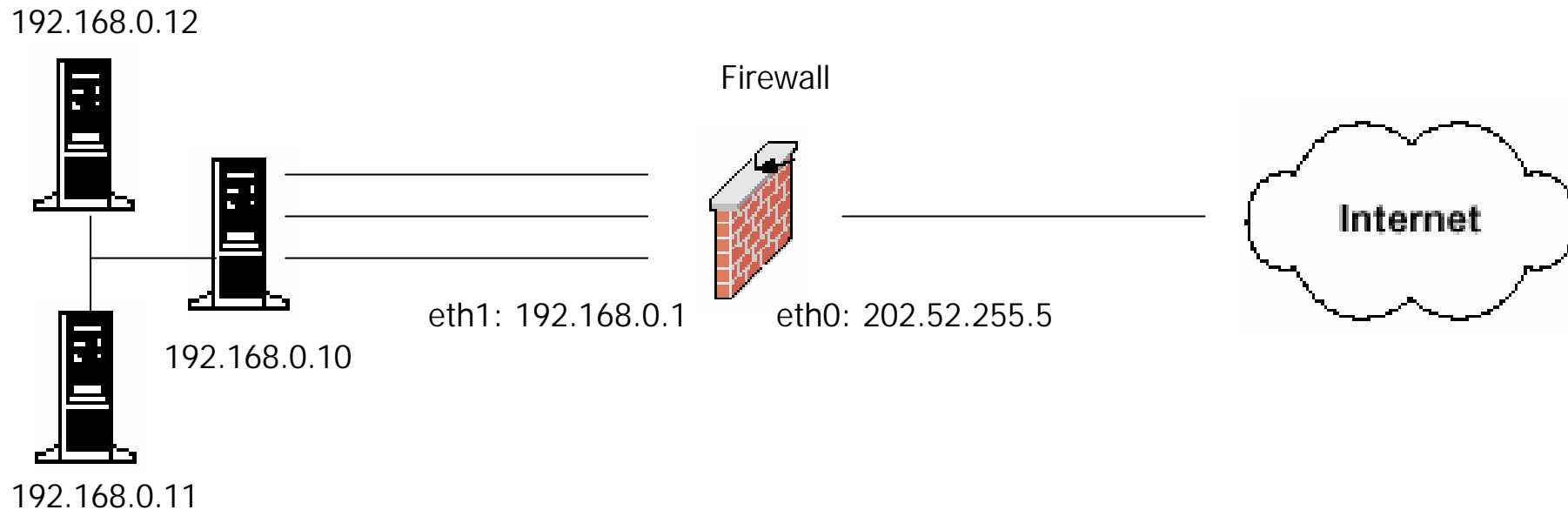


Outbound packets receive the IP of the output interface as their source address. It is useful when there is no fixed IP addresses of output interface.

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

This rule translates the source IP of all outbound packets to 202.52.255.5, the IP of eth0

# NAT: SNAT (outbound)

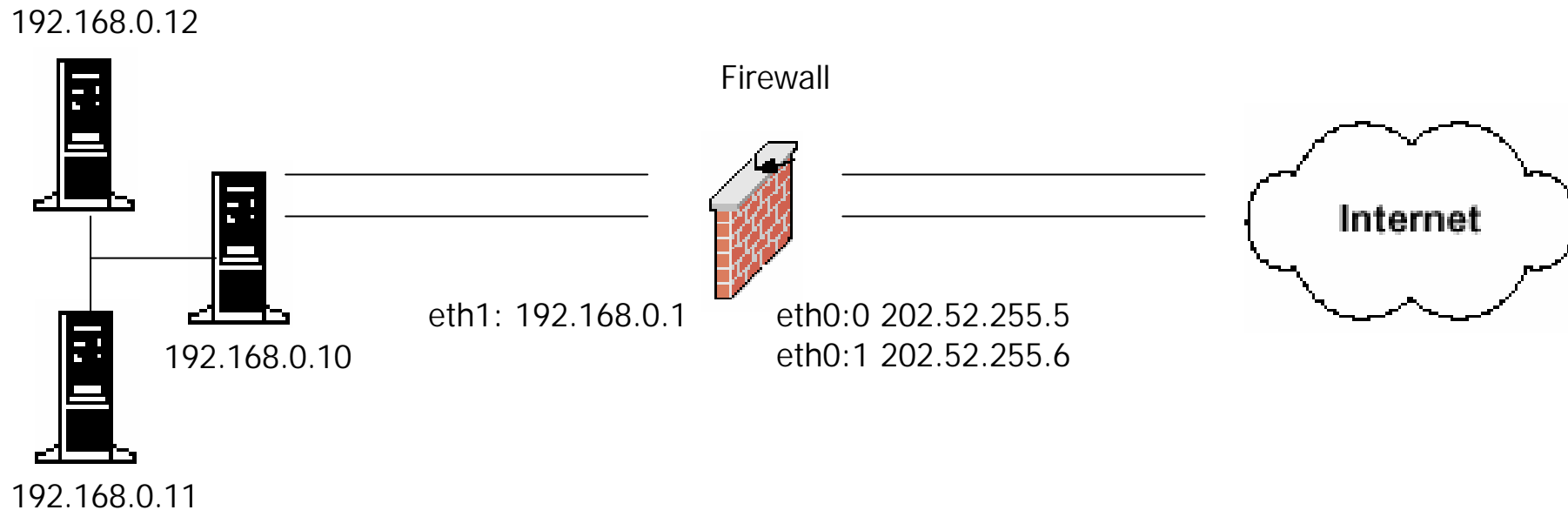


Another way to sharing a single public IP by all private hosts is to SNAT (Source NAT) it.

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24  
-j SNAT --to-source 202.52.255.5
```

The source IP of all outbound packets will be converted to 202.52.255.5

# NAT: Fixed IP mapping (outbound)



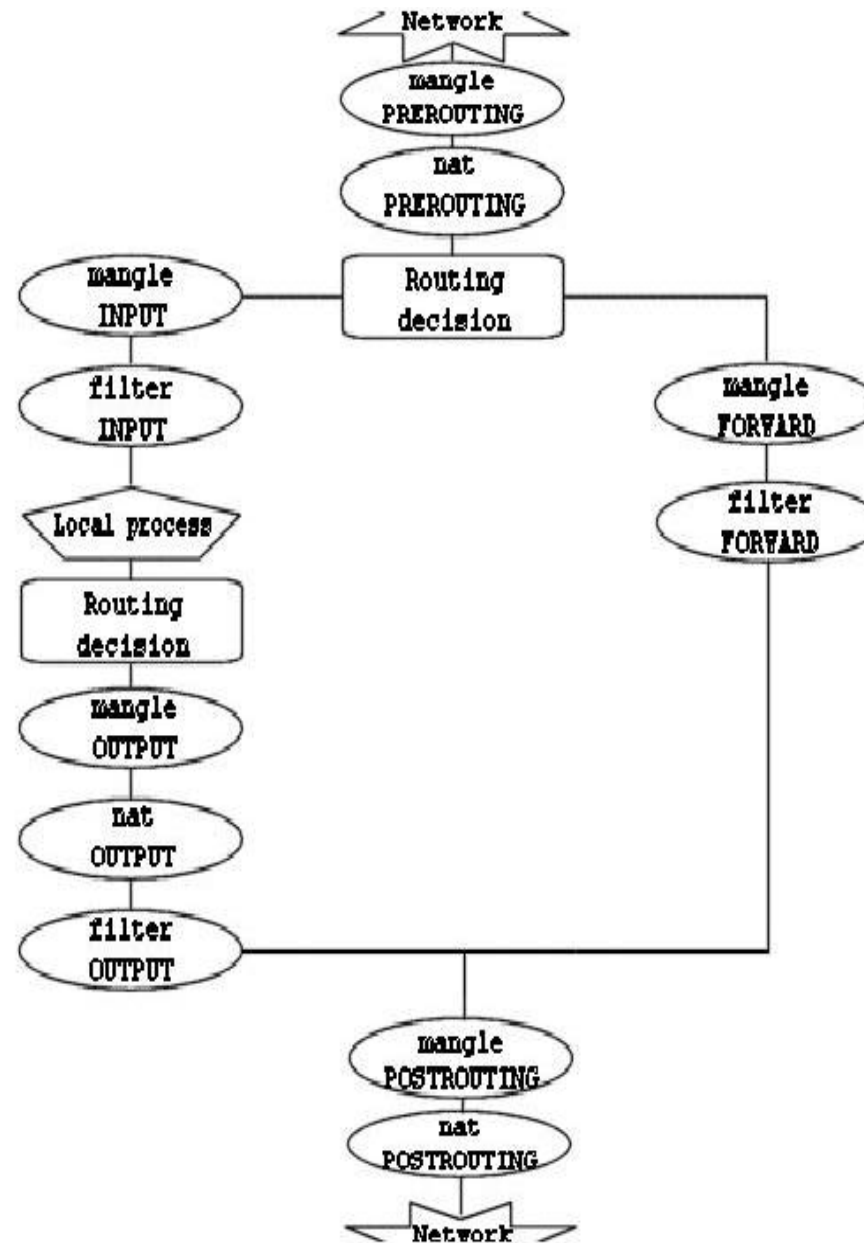
```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.11  
-j SNAT --to-source 202.52.255.5
```

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.12  
-j SNAT --to-source 202.52.255.6
```

The source IP of 192.168.10.11 will be converted to 202.52.255.5

The source IP of 192.168.10.12 will be converted to 202.52.255.6

# How packets flow thru IPtables?





# IPTables Lab

- Installation
- Using Iptables
  - To view all iptables command line options
  - To list all current default rules/chains
  - To set the default policies for all chains
  - To allow ping to work to/from your host to anywhere
  - To allow ping to work only across the firewall but not to/from it
  - To allow all internal users to access websites on the Internet
  - To allow some external users access to SSH, SMTP, POP, HTTP, DNS servers in your internal network
  - To save all rules/chains to /etc/sysconfig/iptables to make permanent
  - To manage iptables service
  - NAT configuration